The Honorable Barbara J. Rothstein

UNITED STATES DISTRICT COURT
WESTERN DISTRICT OF WASHINGTON AT SEATTLE

THROUGHPUTER, INC., Plaintiff, v. MICROSOFT CORPORATION, Defendant.	Civ. No. 2:22-cv-00344-BJR Civ. No. 2:22-cv-00492-BJR FIRST AMENDED COMPLAINT FOR PATENT INFRINGEMENT JURY DEMANDED

ThroughPuter, Inc. ("ThroughPuter") hereby alleges for its First Amended Complaint ("Complaint") in these Consolidated Cases against Microsoft Corp. ("Microsoft") as follows.

INTRODUCTION

1. A Field Programmable Gate Array (FPGA) is a specific type of microprocessor that can be reconfigured based on the tasks to be performed by it. In some situations, this reconfiguration takes place in between the performance of tasks by the FPGA. For example, the FPGAs can be configured and reconfigured to provide hardware accelerators for data processing functions such as encoding/decoding, encryption/decryption or compression/decompression and then reconfigured to perform speech translation in addition to, *e.g.*, encryption and compression.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

MICROSOFT 1101

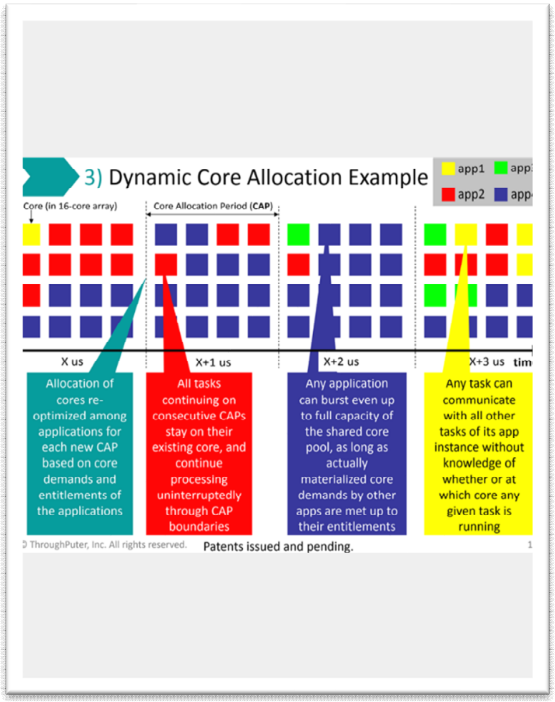
Starting in 2015, Microsoft started putting FPGA processors in each server for Microsoft Azure ("Azure"), which is the world's largest cloud computing platform-as-a-service (PaaS). Using these FPGA processors for the underlying dynamic parallel execution environment or architecture, Microsoft claims a 150-200 fold improvement in data throughput and a 50 fold improvement in energy efficiency. Ex. 17 at 91, Ex. 29. Latency has also lowered by about a factor of 10. Ex. 17 at 86, Ex. 29. The end result is a power savings in Microsoft and increased processing speeds for Microsoft and the users of applications running on the world's largest cloud computing platform.

2. In 2013, Plaintiff ThroughPuter disclosed a reconfigurable and dynamic parallel execution architecture running on FPGA processors in writing to Microsoft. Two years later, Microsoft filed a patent application on the same hardware-based fabric (failing to disclose to the Patent Office any information about ThroughPuter's technology or earlier patent filings). Microsoft was ultimately awarded several patents on this subject matter. For example, in October 2020, Microsoft was awarded U.S. Patent No. 10,819,657, which claims match almost exactly those of ThroughPuter's U.S. Patent No. 11,150,948, which is the basis for Count 1 of this complaint. In other words, Microsoft was awarded a patent on the same hardware-based fabric claimed in the ThroughPuter patents, wrongly suggesting that Microsoft had invented the architecture underlying Azure where ThroughPuter had already patented that architecture and disclosed it to Microsoft.

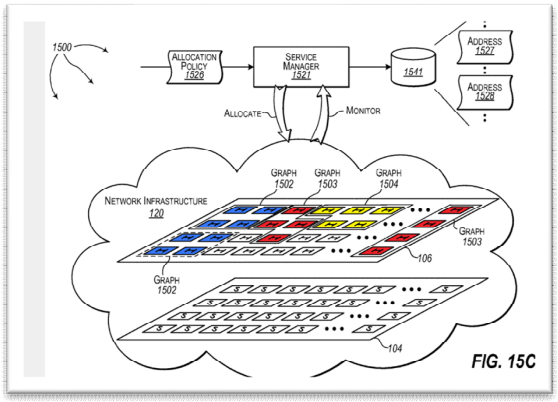
3. The figure below left is an excerpt from the materials disclosed by ThroughPuter to Microsoft in 2013, disclosing an application load and type adaptive "manycore fabric." The different colors show different reconfigurable cores (*e.g.*, logic blocks of FPGA processors) being assigned different tasks over time. The figure on the right is a colorized version of figures from

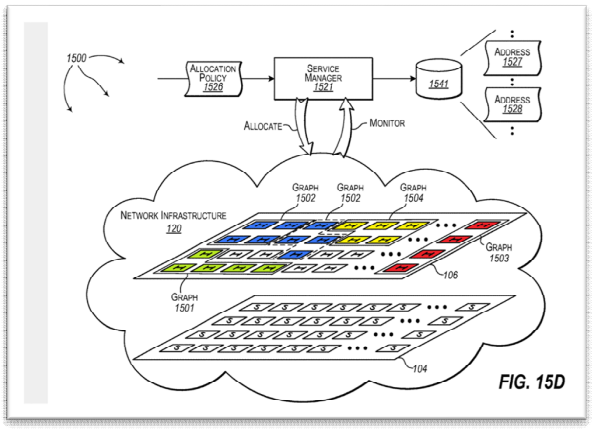
LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

Microsoft's 2015 patent filing. The Microsoft figures show the same reconfiguration of cores to different tasks over time.



Ex. 43 at 18





Ex. 13, FIGs. 15C and 15D (annotated)

4. Microsoft's copying of ThroughPuter's technology was deliberate and wanton. While ThroughPuter has managed to survive as a small company developing and offering for sale other innovative solutions enabled by the throughput improvements, which result from ThroughPuter's patented technology, Microsoft's copying and infringement of ThroughPuter's

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

intellectual property has irreparably damaged ThroughPuter's prospects of entering the PaaS market, which had grown from $3.8B in 2015 to $37.5B in 2019.[1]

5. On March 31, 2021, ThroughPuter filed suit against Microsoft in the United States District Court for the Eastern District of Virginia under Civil Action No. 3:21-cv-00216-MHL alleging infringement of related patents (the "Virginia Action"). ThroughPuter served the complaint in the Virginia Action on April 6, 2021. The Eastern District of Virginia transferred the Virginia Action to this Court by Order dated March 23, 2022. This Court assigned the transferred Virginia Action to Judge Rothstein as Civil Action No.2:22-cv-00344-BJR.

6. Prior to filing the Virginia Action, ThroughPuter repeatedly attempted to engage with Microsoft to resolve this matter outside of litigation. Microsoft rejected or ignored ThroughPuter's proposals and requests for business resolution of the matter.

7. Microsoft's decision to reject ThroughPuter's proposals and offers to collaborate created a highly inequitable situation by which Microsoft has used its tremendous market power to scale up the world's largest and most successful cloud computing platform. Microsoft did this based on the technology it copied from ThroughPuter: a small start-up that could have only competed based on its effort to protect and patent its innovations. Microsoft vaporized the competitive edge to which ThroughPuter was entitled by its decision to ignore ThroughPuter while generating billions of dollars in revenue per quarter based on its unlawful exploitation of ThroughPuter's technology.

---

[1] https://www.statista.com/statistics/505248/worldwide-platform-as-a-service-revenue/

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

8. In a 2016 presentation entitled "Catapult at Ignite Innovation Keynote," Microsoft's Chief Executive Officer, Mr. Satya Nadella, boasted of the numerous benefits Microsoft had achieved by incorporating FPGAs into its cloud computing offerings: "We now have FPGA support across every compute node of Azure. That means we have the ability[,] ***through the magic of the fabric*** that we have built, to distribute your machine learning tasks, your deep neural nets, to all of the silicon that is available, so that you can get that performance, that scale." Ex. 31 at 9 (emphasis added). The "magic of the fabric" did indeed allow Microsoft to scale up an FPGA-based cloud computing solution. But the "magic of the fabric" was not invented by Microsoft, it was copied from ThroughPuter.

9. According to Dr. Doug Burger (with whom ThroughPuter corresponded about its technology), the "magic of the fabric" referred to by Microsoft's CEO, use of FPGAs "allows us to do things on a scale that hasn't been done before." According to Dr. Burger: "It gives us the most powerful cloud, the most flexible cloud, and the most intelligent cloud." *Id.* at 30, 34.

10. The loss of ThroughPuter's technical competitive advantage through Microsoft's copying based infringement, which Microsoft admits gives it on the order of 100-fold performance and efficiency gain, has devastated ThroughPuter's business, including the ability to raise sufficient startup capital, gain collaborators, partners and initial customers, and to generally enter the market with a differentiated or more cost-efficient solution. ThroughPuter has survived, notwithstanding Microsoft's anticompetitive tactics, due to its ability to innovate in the fields of application of its core technology.

11. ThroughPuter now brings this action based on Microsoft's willful infringement of the patents that form the causes of action herein.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

NATURE OF THE ACTION

12. This is an action for patent infringement arising under the Patent Laws of the United States, 35 U.S.C. § 1 *et seq*., including 35 U.S.C. § 271.

13. ThroughPuter brings this action to halt Microsoft's infringement of its rights under the Patent Laws of the United States, 35 U.S.C. § 1 *et seq*., which arise under the following patents:

- U.S. Patent No. 11,50,948 ("the '948 patent"), which is attached hereto as Exhibit 1,
- U.S. Patent No. 11,036,556 ("the '556 patent"), which is attached hereto as Exhibit 2,
- U.S. Patent No. 10,963,306 ("the '306 patent"), which is attached hereto as Exhibit 3,
- U.S. Patent No. 10,620,998 ("the '998 patent"), which is attached hereto as Exhibit 4,
- U.S. Patent No. 10,437,644 ("the '644 patent"), which is attached hereto as Exhibit 5,
- U.S. Patent No. 10,430,242 ("the '242 patent"), which is attached hereto as Exhibit 6,
- U.S. Patent No. 10,318,353 ("the '353 patent"), which is attached hereto as Exhibit 7,
- U.S. Patent No. 10,310,902 ("the '902 patent"), which is attached hereto as Exhibit 8,
- U.S. Patent No. 10,133,599 ("the '599 patent"), which is attached hereto as Exhibit 9,

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

- U.S. Patent No. 9,632,833 ("the '833 patent"), which is attached hereto as Exhibit 10, and
- U.S. Patent No. 9,424,090 ("the '090 patent"), which is attached hereto as Exhibit 11.

THE PARTIES

14. Plaintiff ThroughPuter, Inc. is a Delaware corporation having a principal place of business at 249 Richmond Road, Williamsburg, VA 23185. Plaintiff owns over 50 issued domestic and foreign patents and pending applications protecting its products, services and technologies. ThroughPuter's President, Mark Sandstrom, is the named inventor on each of such patents and applications.

15. Plaintiff has developed and continues to develop various products and services, including i) Estimator™, a machine learning Application Specific Processor (ASP)-as-a-service offering of the ThroughPuter PaaS project, and ii) Grafword™, an artificial intelligence (AI) powered, graphical authentication service that is a pilot application of the Estimator™ machine learning microservice.

16. Estimator™ provides a streaming machine learning (ML) microservice, to support AI applications in unpredictably changing operating environments. Estimator™ allows its prediction models and logic parameters to be adjusted continuously while the microservice is in operation, such that its predictions will stay tuned-in to the prevailing reality of its operating environment, as that may evolve over time or even change abruptly. An International Search Report recently conducted by the International Search Authority under the Patent Cooperation Treaty concluded that this technology is patentable. A beta version of the Estimator™ application

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

programming interface is currently commercially available for 3rd party developer subscription at www.estimatorlab.com.

17. Grafword™ provides graphic based high-security password generation and authentication, such that the level of authentication challenge is adjusted according to a level of deviation of a given user's online session attributes from what is expected for the given username. Grafword™ thus provides both high security as well as, for the authentic users, convenience in online authentication. An International Search Report recently conducted by the International Search Authority under the Patent Cooperation Treaty also concluded that this technology is patentable. A beta version of Grafword™ is used for Estimator™ account creation and login: https://estimatorlab.com/landing.

18. Both Estimator™ and Grafword™ have the potential to change the space in which they are offered due to the advantages provided by ThroughPuter's claimed inventions such as increased throughput and latency.

19. Microsoft is a corporation organized and existing under the laws of the State of Washington with its principal place of business at One Microsoft Way, Redmond, WA 98052.

JURISDICTION AND VENUE

20. This is an action for patent infringement which arises under the Patent Laws of the United States, 35 U.S.C. § 1 et seq.

21. This Court has subject matter jurisdiction at least under 28 U.S.C. §§ 1331 and 1338.

22. Venue is proper in this District under 28 U.S.C. § 1400(b) because Microsoft has committed acts of infringement and has a regular and established place of business in this District.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

23. This Court has personal jurisdiction over Defendant Microsoft pursuant to due process and/or Washington's Long Arm Statute because Microsoft has committed and continues to commit acts of patent infringement, including acts giving rise to this action, within the State of Washington and this District, and because Microsoft recruits Washington residents, directly or through an intermediary located in this state, for employment inside or outside this state.

24. The Court's exercise of jurisdiction over Microsoft would not offend traditional notions of fair play and substantial justice because Microsoft has established at least the required minimum contacts with the forum.

25. Microsoft maintains regular and established places of business throughout Washington and in this District.

26. Microsoft has substantial business contacts within this District and has purposefully availed itself of the privileges and benefits of the laws of the State of Washington.

BACKGROUND

27. This case involves ThroughPuter's patented cloud computing, computing acceleration and related technologies, which were developed starting in 2010.

28. As of that time, advancements in computing technologies had generally fallen into two categories. First, in the field conventionally referred to as high performance computing, the main objective has been maximizing the processing speed of a given computationally intensive program running on dedicated hardware. In this field, speed was traditionally achieved by assigning a combination of separate parallel processors to all work on the same program simultaneously. Second, in the field conventionally referred to as utility or cloud computing, the main objective has been to most efficiently share a given pool of computing hardware resources among a large number of client application programs.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

29. Thus, in effect, one branch of computing innovation has been seeking to effectively use a large number of parallel processors to accelerate execution of a single application program by parallelizing its processing across a maximum possible number of processors. At the same time, another branch of computing innovation has been seeking to share a single pool of computing capacity among a large number of application programs to optimize utilization of processing capacity. The former efforts pursue maximizing processing speed of a single program. The latter efforts pursue maximizing utilization of processing capacity.

30. As of the time of ThroughPuter's pioneering patent filings starting in 2011, there had not been major synergies between the effort to increase processing speed of a single program on the one hand, and maximizing processing capacity utilization on the other. Indeed, pursuing one of these traditional objectives often happened at the expense of the other, placing the two objectives in tension with each other.

31. For instance, while dedicating an entire parallel processor based (super) computer to each individual application would increase processing speed of the individual programs, it would also cause severely sub-optimal computing resource utilization, as much of the capacity would be idle much of the time. On the other hand, while seeking to improve utilization of computing systems by sharing their processing capacity among a number of applications would lead to enhanced resource utilization, it also tended to slow down processing of individual programs. As such, the overall cost-efficiency of computing was not improving as much as improvements toward either of the two traditional objectives would imply: traditionally, increases in processing speed came at the expense of system utilization efficiency, while overall system utilization efficiency maximization came at the expense of individual application processing speed.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

32. The foregoing tension was exacerbated by the fact that even mainstream application performance requirements were increasingly exceeding the processing throughput achievable from a single CPU core, *e.g.*, due to the practical limits being reached on the CPU clock rates. This created an emerging requirement for intra-application parallel processing (at ever finer grades) even for mainstream programs in order to pursue satisfactory processing speeds, while these programs were to be increasingly hosted on cloud platforms where the processing resources would be shared among programs of multiple clients.

33. These internally parallelized and/or pipelined (concurrent) enterprise and web applications would ultimately be largely deployed on dynamically shared cloud computing infrastructure by entities such as Microsoft using the technologies patented, pioneered and promoted by ThroughPuter.

34. Given the foregoing, there existed a need as of 2011 for supporting a large number of concurrent applications on dynamically shared parallel processing resource pools. This then-existing need for a new parallel computing architecture could be met by a system that enabled increasing the speed of executing application programs (including through execution of a given application in parallel across multiple processor cores and/or using hardware accelerators) while at the same time improving the utilization of the available computing resources.

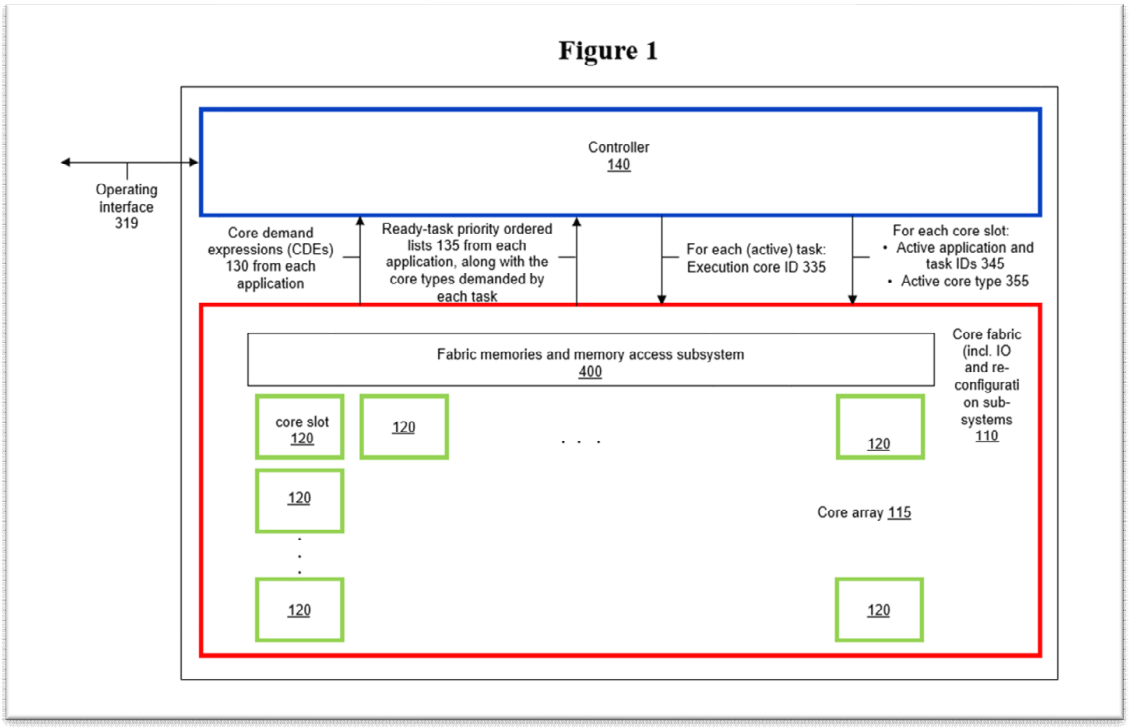
35. To address these problems, ThroughPuter developed hardware implemented dynamic resource management functionality including a scheduler, placer, inter-task communications and input/output system for use with multicore processor arrays dynamically shared among multiple concurrent applications, preferably to be deployed on FPGA processors. To that end, in this technology approach, the manycore processor array involves a fabric of reconfigurable cores that can be on-demand programmed to supply the needed mix or match of

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

hardware accelerators. ThroughPuter's technology provided a cloud computing solution that enables accelerated processing speeds across multiple application programs while at the same time optimizing processing resource utilization.

36. An exemplary embodiment of ThroughPuter's Dynamic Parallel Execution (DPE) Environment™ (sometimes referred to as "DPEE™" or "DPEE™ technology") is depicted in, for example, ThroughPuter's U.S. Patent No. 9,424,090. Ex. 11. Referring to the figures and tables of the representative '090 patent reproduced below, this preferred embodiment includes an **array of core slots 120**. The processors, or cores, are coordinated and coupled together with a **hardware-based manycore fabric managed by the controller**, which allows the client program tasks to be dynamically placed on processor cores of many-core arrays while communicating with each other directly and securely.

Figure 1



Ex. 11, Fig. 1 (annotated)

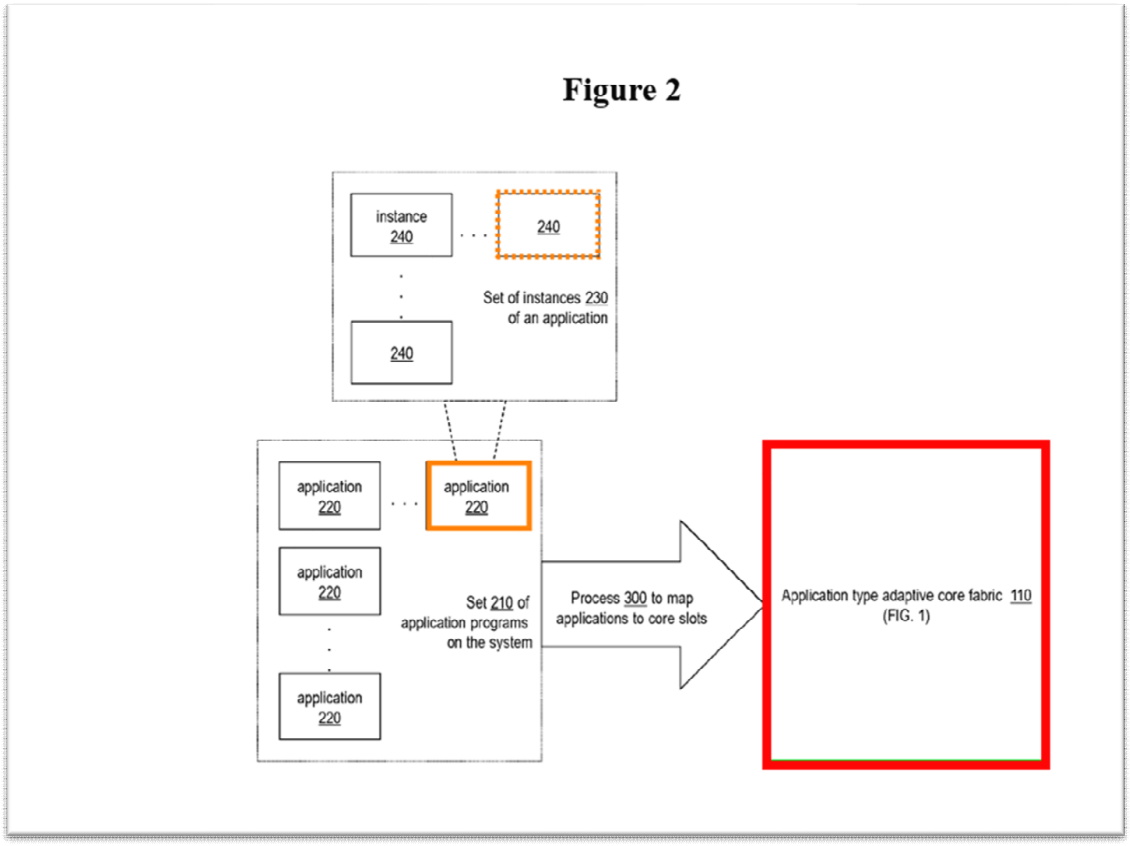
LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

37. ThroughPuter's hardware-based manycore fabric enables processing to be dynamically parallelized and hardware-accelerated, which results in optimized on-time processing throughput across the programs sharing an array of manycore processors. The effect for the client or end-user of an accelerated service is increased processing speed and reduced cost base for delivering the application service, such that it becomes economically feasible for cloud service providers to support a range of performance intensive applications even without charge to end-users.

38. In a preferred embodiment, the **controller 140** monitors the processing load for each program sharing **the manycore array 110**, so that periodically re-optimized sets of tasks from the programs can be periodically assigned for execution on the **pool of processor cores 120**. Such periodic reassignment allows for optimal utilization of a pool of processing resources across a number of potentially competing, concurrent applications. ThroughPuter's exemplary embodiment enables assigning time-variable sets of application tasks for execution on the fabric of reconfigurable cores.

39. A variety of **application tasks 240** are assigned to be run on the **core fabric**, which includes **an array of core slots** within the **core fabric**. Each **application task** may be assigned to run on a **single processor core** within the **core fabric**. The cross-connect, which connects the various components of **the hardware-based manycore fabric,** is used by the **controller** to repeatedly optimize assignment of **tasks** to the appropriate **processor core slots**.

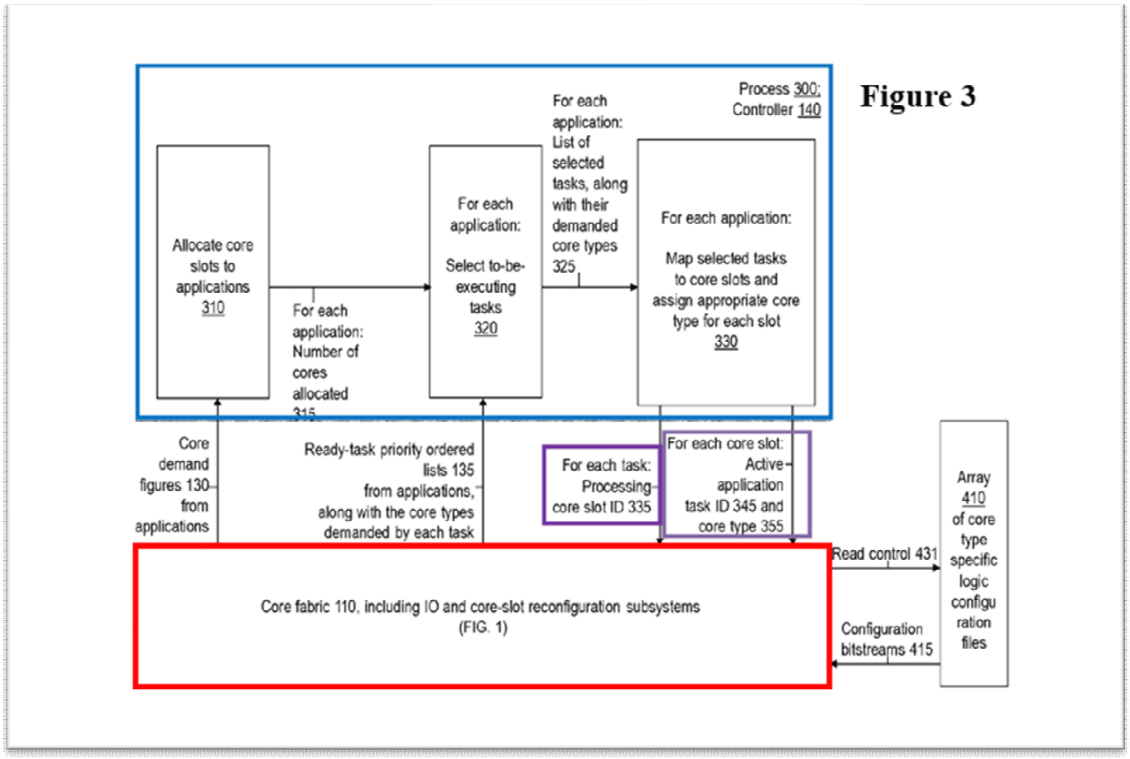
LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301



Ex. 11, Fig. 2 (annotated)

40. The **controller** manages the **mapping** of **application tasks** to **processor cores** within **the core fabric**. The result is a system which dynamically and in real-time adapts to varying application processing loads to provide scalable, secure, high-performance and resource-efficient parallel cloud computing.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301



Ex. 11, Fig. 3 (annotated)

TABLE 5

Core ID index	Application ID	Instance ID (within the application of column to the left)	Core type (e.g., 0 denotes CPU, 1 denotes DSP, 2 denotes GPU, 3 . . . 15 denotes an ASP for a particular function, etc.)
0	P	0	0
1	B	0	0
2	B	8	2
. . .	. . .	. . .	. . .
14	F	1	5
15	N	1	1

Ex. 11, Table 5 (annotated)

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

41. This novel application load and type adaptive manycore fabric permits tasks or applications to be managed at a high degree of granularity with minimized processing overhead while providing various advantages not previously attainable.

42. In recognition of ThroughPuter's innovative achievements, ThroughPuter's Mark Sandstrom was invited to speak at various high performance and cloud computing conferences starting in 2012. GigaOm selected ThroughPuter as one of eleven finalists to present at Launchpad 2012 in San Francisco, CA. That same year, ThroughPuter was invited to present its Dynamic Parallel Execution Environment (DPEE) based PaaS approach at the high performance computing start-up showcase at the Supercomputing 2012 conference ("SC12") in Provo, UT.

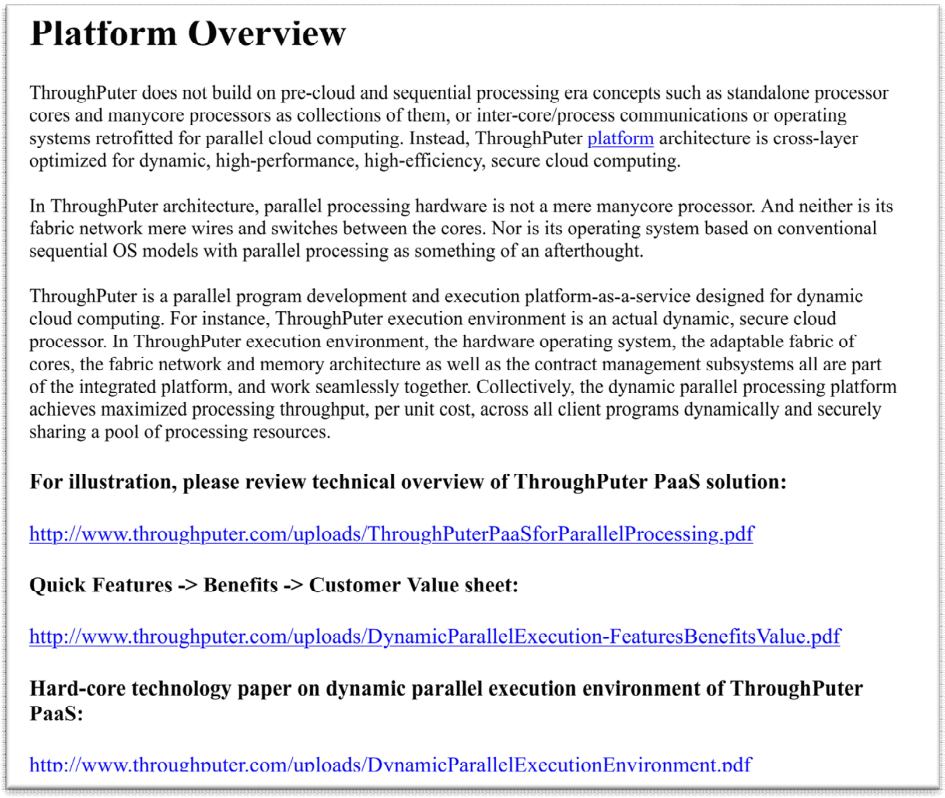
43. In February 2013, Mr. Sandstrom reached out to Dr. Doug Burger, Director of Client and Cloud Applications at Microsoft Corporation, via email indicating that ThroughPuter was looking for collaborators. The body of the email, which is attached as Exhibit 21, is reproduced below:

> Doug,
>
> I read at http://www.hpcwire.com/hpcwire/2013-02-01/kalray_produces_supercomputer-on-a-chip_for_industrial_applications.html your remarks at HiPEAC, concerning the need to modernize the computing architectures in view of the physical limits and future application requirements. In this context you may find ThroughPuter's cross-layer optimized platform architecture, based on dynamic parallel execution model, quite relevant -- please review http://www.throughputer.com/platform.html
>
> ThroughPuter is looking for collaborators for the effort to make the advanced dynamic parallel program execution capabilities available for users (application developers) via PaaS model; possibly Microsoft Azure business unit might be interested in exploring the collaboration opportunities.
>
> Feel welcome to share this call for collaboration among the appropriate parties at Microsoft, and naturally please get back to me for any questions etc. further discussions.

44. Dr. Burger responded: "Thanks Mark ... I appreciate the note. I'll forward to the right people in Azure." *Id.*

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

45. In February 2013, the webpage at http://www.throughputer.com/platform.html had the following content at the time of the email exchange:



Platform Overview

ThroughPuter does not build on pre-cloud and sequential processing era concepts such as standalone processor cores and manycore processors as collections of them, or inter-core/process communications or operating systems retrofitted for parallel cloud computing. Instead, ThroughPuter platform architecture is cross-layer optimized for dynamic, high-performance, high-efficiency, secure cloud computing.

In ThroughPuter architecture, parallel processing hardware is not a mere manycore processor. And neither is its fabric network mere wires and switches between the cores. Nor is its operating system based on conventional sequential OS models with parallel processing as something of an afterthought.

ThroughPuter is a parallel program development and execution platform-as-a-service designed for dynamic cloud computing. For instance, ThroughPuter execution environment is an actual dynamic, secure cloud processor. In ThroughPuter execution environment, the hardware operating system, the adaptable fabric of cores, the fabric network and memory architecture as well as the contract management subsystems all are part of the integrated platform, and work seamlessly together. Collectively, the dynamic parallel processing platform achieves maximized processing throughput, per unit cost, across all client programs dynamically and securely sharing a pool of processing resources.

For illustration, please review technical overview of ThroughPuter PaaS solution:

http://www.throughputer.com/uploads/ThroughPuterPaaSforParallelProcessing.pdf

Quick Features -> Benefits -> Customer Value sheet:

http://www.throughputer.com/uploads/DynamicParallelExecution-FeaturesBenefitsValue.pdf

Hard-core technology paper on dynamic parallel execution environment of ThroughPuter PaaS:

http://www.throughputer.com/uploads/DynamicParallelExecutionEnvironment.pdf

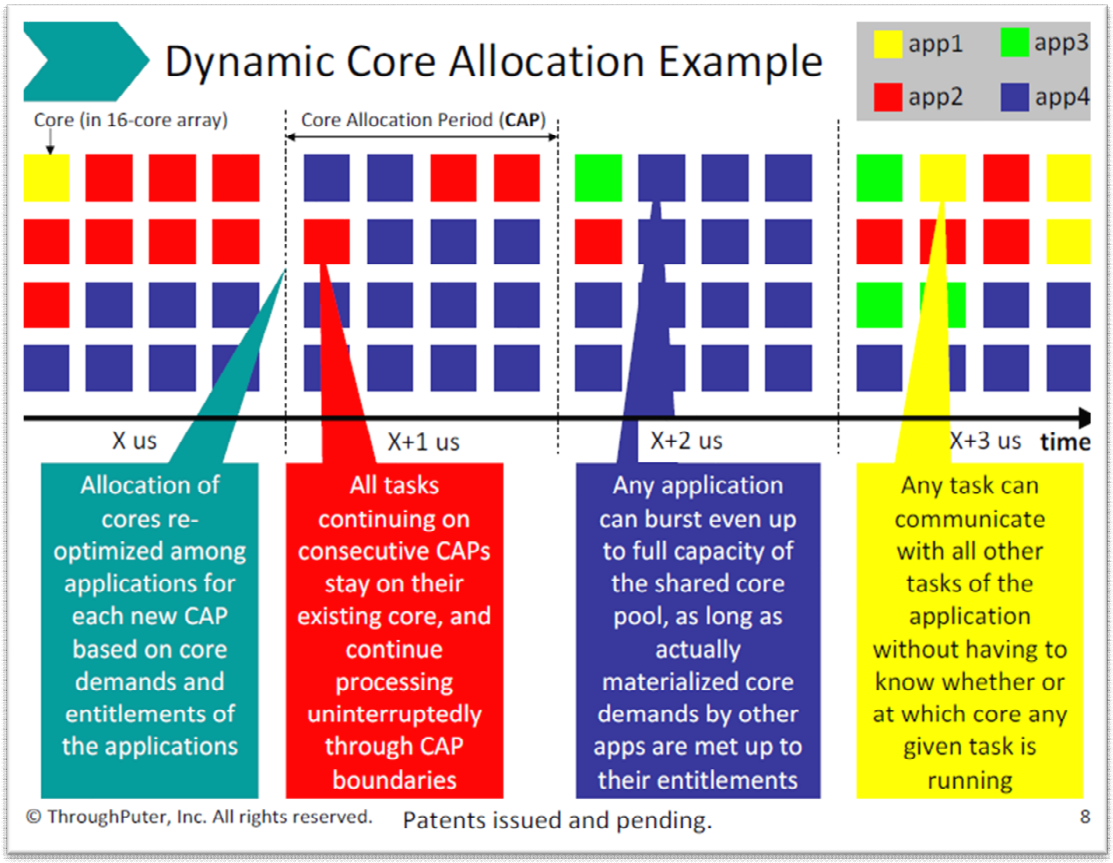
46. The third linked PDF entitled *Parallel Program Execution in a Dynamically Shared Adaptive Manycore Processor* is a document that closely follows the detailed description of ThroughPuter's '090 patent. Ex. 22.

47. The second linked PDF entitled *Dynamic Parallel Execution via PaaS* is a document that provides an overview of the innovations described in ThroughPuter's portfolio of patents and applications, including the application leading to the '090 patent, and the substantial advantages they provide. Ex. 23.

48. The first linked PDF entitled *Parallel Computing Development and Hosting Platform as a Service* is a presentation that provides an overview of ThroughPuter's DPEE.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

Ex. 24. As shown in one of ThroughPuter's slides, reproduced below, the cores are dynamically re-assigned to instances or tasks of different applications, which are coded by color.



Ex. 24 at 8

49. Repeatedly, the pool of (sixteen) processing cores are each allocated among the four different application programs. In an illustrative example, these may be application programs used for **speech translation (blue)**, **encryption (red)**, **search ranking (yellow)** and **data compression (green)**. In the first cycle or period, the search ranking **(yellow)** application program is allocated one core while the encryption **(red)** program application and speech translation **(blue)** application program are allocated eight and seven cores, respectively. For the next period, several of the cores are reassigned to the speech translation **(blue)** program application to accommodate

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

an increase in processing demand made by that application. In the third period, a further core is allocated to the speech translation **(blue)** program application and the data compression **(green)** program application is assigned one core, also in response to demand for those programs. In the fourth period, the cores are more evenly allocated across all four applications because the spike in demand for speech translation has abated, or because of demand spikes of the other applications. In this way, the processing throughput and latency across all the applications sharing the given manycore processor array are optimized. The end result for the application programs is faster processing compared to what would be achievable for equal cost base under non-adaptive capacity allocation or without on-demand acceleration.

50. ThroughPuter's novel manycore fabric led to industry recognition of ThroughPuter and its technology. For example, in January 2013, ThroughPuter was invited to publish an article in the Cloud Computing Journal, discussing the PaaS based on novel manycore fabric. Ex. 14.

51. In addition, in September 2014, Mr. Sandstrom presented at the FPGAworld conference in Stockholm, Sweden, on the topic of *Hardware Implemented Scheduler, Placer, Inter-Task Communications and IO System Functions for Manycore Processors Dynamically Shared among Multiple Applications*. Ex. 43.

52. By the time of the 2014 FPGAworld conference, ThroughPuter had already been granted at least a dozen U.S. and United Kingdom patents protecting techniques enabling the advantages of its Dynamic Parallel Execution Environment™ (DPEE).

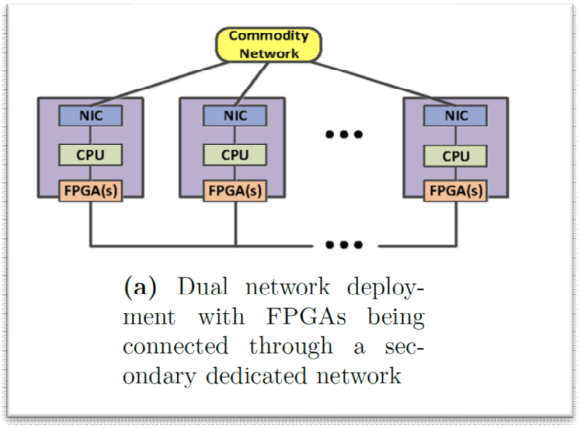
53. The following year, ThroughPuter was invited to present at the 2015 HPC Advisory Council Conference in Spain on the topic of executing multiple dynamically parallelized programs on dynamically shared cloud processors. A copy of the presentation is attached hereto as Exhibit 26.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

MICROSOFT'S INFRINGING CLOUD COMPUTING ARCHITECTURE

54. Microsoft's infringing cloud computing platform is known as Azure which is the technology service platform accused of infringement herein.

55. In April 2014, Microsoft introduced an early version of the Microsoft Azure reconfigurable hardware fabric in the whitepaper *A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services*, authored by Andrew Putnam, Adrian Caulfield, Eric Chung, Derek Chiou, Kypros Constantinides, John Demme, and seventeen additional Microsoft engineers. Ex. 15. Microsoft initially called this reconfigurable fabric Catapult™ and still uses that term today. *Id.* at 1. This early version of Catapult, depicted in the figure at right, was deployed on a 1,632 server testbed. *Id.*, *see also* Ex. 40 at 28-30.



(a) Dual network deployment with FPGAs being connected through a secondary dedicated network

56. The CPU and FPGA in each node communicate via PCIe. *Id.*, *see also* Ex. 35 *passim*, Ex. 40 at 28-30. *Id.* For external connectivity, dual network interfaces was used. *Id.* The first was Ethernet-based, which enables CPUs to communicate with each other over the commodity datacenter network. *Id.* The second was a specialized, low latency network implemented using 10Gb SAS cables, which connects FPGAs to each other. *Id.* The topology for the second network was a 2D torus. *Id.*

57. The figure below provides details of components within the FPGA. *Id.* Here, custom load logic was implemented within an FPGA fabric allocation called "Role," while the Shell logic was composed of:

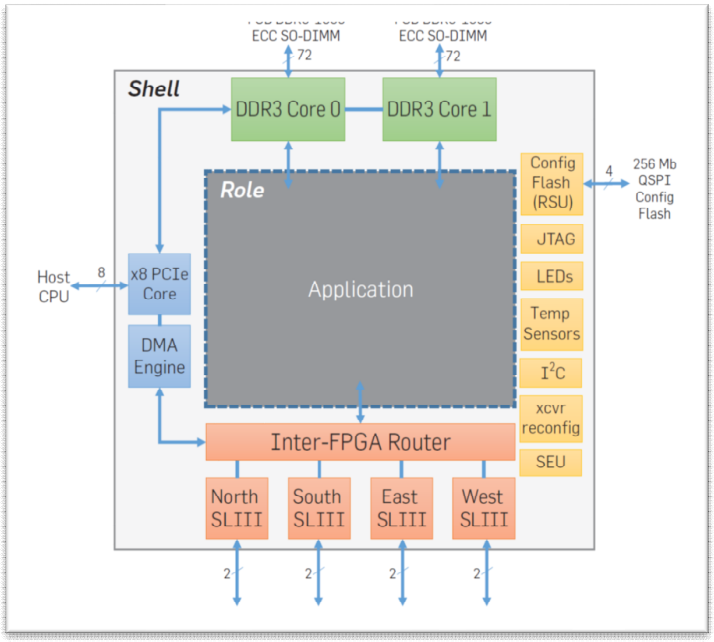
LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

i) PCIe Core: This was used to implement an interface between the host CPU and the corresponding FPGA.

ii) SerialLite III (SLIII): This was a lightweight protocol used for inter-FPGA communication over the SAS links. Since each FPGA connects to four other FPGAs (2D Torus), there were four SLIII blocks in the Shell.

iii) DRAM Controllers: These were used to provide on-chip memory access to the Role or host (for DMA over PCIe).

iv) Inter-FPGA Router: A crossbar connected the four SLIII blocks, PCIe Core and Role.



Ex. 34 at 4; Ex. 35 at 5

58. The foregoing version of Catapult will hereafter be referred to as Catapult I.

59. Circa 2015 Microsoft improved the Catapult I architecture to upgrade its initial scope of inter-FPGA connectivity, which permitted only communication between FPGAs in the same rack.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

60. Whereas Catapult I connected the FPGAs with "complex cabling" that "require[d] awareness of the machines' physical location [*i.e.*, each FPGA]" and provided only for inter-FPGA communication within a single rack (Ex. 16 at 53), the improved version includes a reconfigurable fabric enabling any FPGA in a datacenter to communicate with any other FPGA in the datacenter through super low latency multiplexer-switched connections. *See* Ex. 45(discussing the basics of how a multiplexer serves as a low latency, high speed switch), Ex. 29 at 148, 151.

61. This second version of Catapult, called Catapult II hereinafter, has been deployed on Azure servers since the launch of Catapult II circa 2015. Ex. 18 at 6 (indicating deployment in Azure "compute servers beginning in 2015."), Ex. 12 at 1 (indicating deployment at "hyperscale in Microsoft's production datacenters worldwide" by October 2016), Ex. 16 at 54.

62. At a high level, the improvement of Catapult II involved transforming the network from being a network of interconnected CPUs (some of which might have local, hardwired FGPA acceleration, sometimes called "bolt-on" acceleration) to a network of interconnected FPGAs (each of which can pass packets along to CPUs that are located "behind" the FPGAs). Stated differently, the network was transformed from being a network of CPUs (as was conventional) to being a network of FPGA processors/accelerators (as described in ThroughPuter's patent filings, which were disclosed to Microsoft years earlier).

63. In one of its whitepapers on the topic, Microsoft explained that "[a]lthough this change to the network design might seem minor" it is a "major advance" that made a "profound" "impact [on] the types of workloads that can be accelerated and the scalability of the [system]." Ex. 16 at 54, 58. Microsoft explained that this design improvement goes "far beyond just an improved network design" and "can be seen as a fundamental shift in the role of CPUs in the

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

datacenter." *Id*. at 58-59. Indeed, the network was changed from a *network of CPUs* with local FPGAs to a *network of FPGAs* with local CPUs. *Id. passim*.

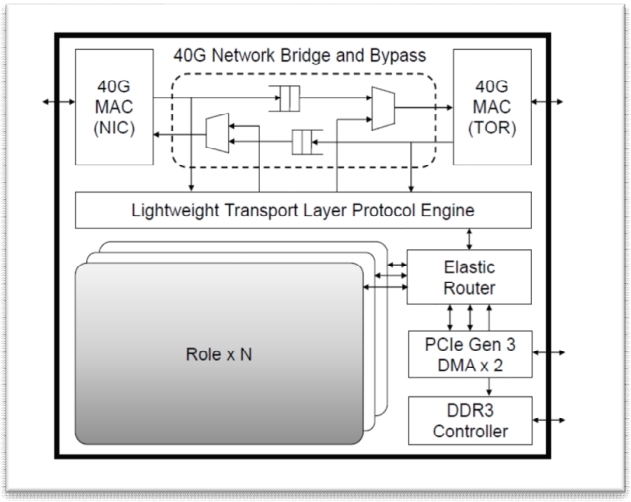
64. In a 2016 presentation entitled "Catapult at Ignite Innovation Keynote," Microsoft's Chief Executive Officer, Mr. Satya Nadella, boasted of the numerous benefits Microsoft had achieved by incorporating FPGAs into its cloud computing offerings: "We now have FPGA support across every compute node of Azure. That means we have the ability[,] ***through the magic of the fabric*** that we have built, to distribute your machine learning tasks, your deep neural nets, to all of the silicon that is available, so that you can get that performance, that scale." Ex. 31 at 9 (emphasis added). The "magic of the fabric" did indeed allow Microsoft to scale up an FPGA-based cloud computing solution. But the "magic of the fabric" was not invented by Microsoft, it was copied from ThroughPuter.

65. According to Dr. Doug Burger (with whom ThroughPuter corresponded about its technology), the "magic of the fabric" referred to by Microsoft's CEO, use of FPGAs "allows us to do things on a scale that hasn't been done before." According to Dr. Burger: "It gives us the most powerful cloud, the most flexible cloud, and the most intelligent cloud." *Id.* at 30, 34.

66. As will be described further below, since the launch of Catapult II, the system has continued to evolve in ways that gradually incorporate more and more of the finer details, features and functionality described in ThroughPuter's patents. Many functions that were performed on the CPUs in the early days of the Catapult II fabric have been gradually shifted to the FPGAs. The Azure fabric (sometimes referred to hereinafter as the "Catapult II network," the "current Catapult II network" or simply "Catapult II") strongly resembles or is nearly identical in relevant respects to the hardware-logic (*e.g.*, FPGA) fabric disclosed in ThroughPuter's patents.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

67. Returning to the "major advance" that is embodied in the Catapult II reconfigurable fabric, central to that advance was the decision to permit every FPGA in a datacenter to communicate with any other through a high-speed, low latency, multiplexer switched fabric as described in the ThroughPuter patents. Exs. 12, 17, 36 *passim*. The figure below shows the "shell" of the Catapult II FPGA. Exs. 12, 16. All of the depicted functionality is performed on the FPGA itself, providing high-speed, low latency interconnectivity. *Id.*, *see also* Ex. 46 *passim*, Ex. 18 at 5.



Ex. 12 at 4

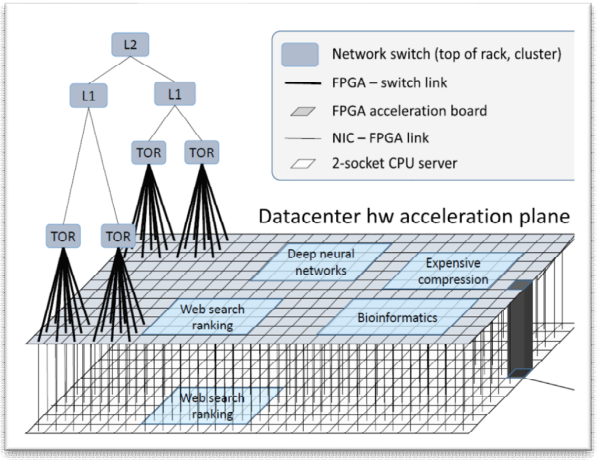
68. Via this shell, the Catapult II inter-FPGA communications and memory accesses occur through the high-speed, low latency multiplexer network inside the FPGA. *Id.* For example, the DDR3 controller comprises DMA multiplexers and demultiplexers. *Id.*, *see also* Ex. 36 at 3. As another example, the bridge between the Lightweight Transport Layer Protocol Engine and the ports for the NIC and TOR includes several functional multiplexer units.

69. In Catapult II, about 23% of each FPGA's logical circuitry (again, comprising logic units interconnected by high-speed, low latency multiplexer networks) is used to provide the new

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

functionality, including the elastic router, LTL engine, and bridge and the other adapters and controllers of Catapult II. *Compare* Ex. 16 at 4 to Ex. 35 at 5. Collectively, these components (hereinafter "Catapult II Shell Components") provide and facilitate FPGA fabric interconnectivity and memory access. Exs. 12, 16, 36 *passim*.

70. Each Azure rack in a datacenter is connected by a two-tier switching network L1/L2 as shown below. Exs. 12, 16, *passim*. The packet communications within each rack and to the top-of-rack switch (TOR) are facilitated by the FPGA via the high-speed, low latency multiplexer controlled connections that serve as the Catapult II Shell Components. *Id.*



Ex. 12, Fig. 1(a)

71. Upon information and belief, since at least 2017, the TOR switch and the L1/L2 switches have included multiplexer-switched network connections to reduce latency. Ex. 38 at § 4.1 ("4.1 Switch design . . . Our circuit switch operates at layer 1, *i.e.*, data traversing the switch is routed through the PHY block at the ingress and egress ports (Fig. 6). ... The control signals to these multiplexers are driven by *p* registers, one per multiplexer. . . Hence, the switch reconfiguration delay is simply the time it takes to update the registers, which can be done in one clock cycle."), Ex. 39 at § 4.1 ("4.1 Switch Design . . . .This is implemented using multiplexers

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

whose control signal is driven by registers storing the clock counter and timeslot. Therefore, the switch reconfiguration latency corresponds to the time required to reconfigure the multiplexers, which is just a few logic gate delays and well below one FPGA clock cycle."). In 2016, Microsoft published a paper entitled *A Cloud-Scale Acceleration Architecture* authored by, among others, Adrian Caulfield (hereafter "Cloud-Scale Acceleration Architecture"). Ex. 12.

72. Microsoft Azure includes or has included the functionality as described in Cloud-Scale Acceleration Architecture.

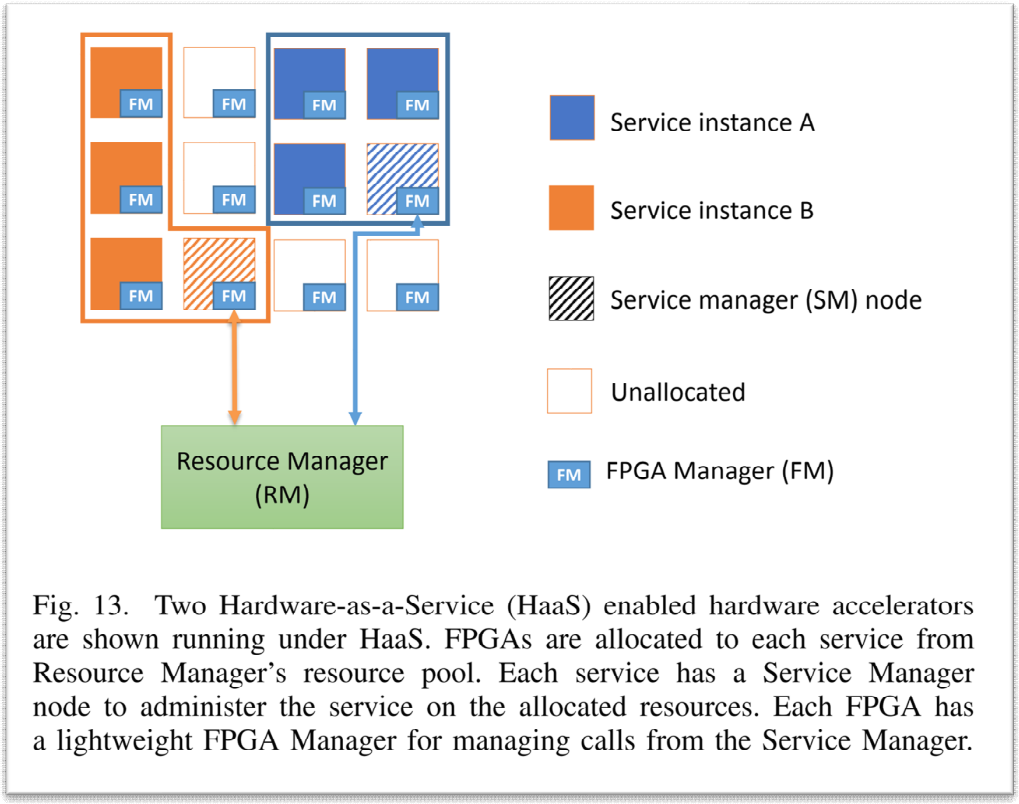


Fig. 13. Two Hardware-as-a-Service (HaaS) enabled hardware accelerators are shown running under HaaS. FPGAs are allocated to each service from Resource Manager's resource pool. Each service has a Service Manager node to administer the service on the allocated resources. Each FPGA has a lightweight FPGA Manager for managing calls from the Service Manager.

Ex. 12, Fig. 13

73. Each Service Manager (SM) described in Cloud-Scale Acceleration Architecture is associated with a given, dynamically sized and placed, FPGA processor group, each of which is referred to as a "service instance" in Cloud-Scale Acceleration Architecture.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

74. In Cloud-Scale Acceleration Architecture, Microsoft described a configurable core fabric similar to the configurable core fabric described in Microsoft's United States Patent Application Serial No. 14/752,800, now United States Patent No. 10,270,709, (hereinafter the" '800 application" or " '709 patent"), which is submitted herewith as Exhibit 13.

75. Fig. 13 of Cloud-Scale Acceleration Architecture illustrates a Resource Manager (RM) that, together with the Service (SM) and the FPGA Managers (FM), performs the same or similar function as the service manager 1521 described in Microsoft's '800 application.

76. Cloud-Scale Acceleration Architecture uses the phrase "service instance" to refer to the same or similar concept as the term "graph" in the '800 application.

77. Cloud-Scale Acceleration Architecture states that the functionality described therein is used to "accelerate . . . Azure infrastructure workloads," among other use cases. Ex. 12 at 12.

78. Microsoft's presentation entitled *Inside Microsoft's FPGA-Based Configurable Cloud* (Ex. 17) demonstrates that Azure embodies or has embodied the functionality described in Cloud-Scale Acceleration Architecture (Ex. 12) and Microsoft's '709 patent (Ex. 13).

79. As explained in the presentation, Microsoft Azure is a system comprised of a number of FPGA-accelerated servers, which process tasks from multiple application programs involving a controller that allocates FPGA processors from among a number of FPGA processors and then assigns application tasks to FPGA processors in order to pursue accelerated application processing while seeking efficient processing capacity usage.

80. A transcript and corresponding screenshots of Microsoft's presentation entitled *Inside Microsoft's FPGA-Based Configurable Cloud* are attached as Exhibit 17. The video is available at https://www.youtube.com/watch?v=v_4Ap1bjwgs.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301



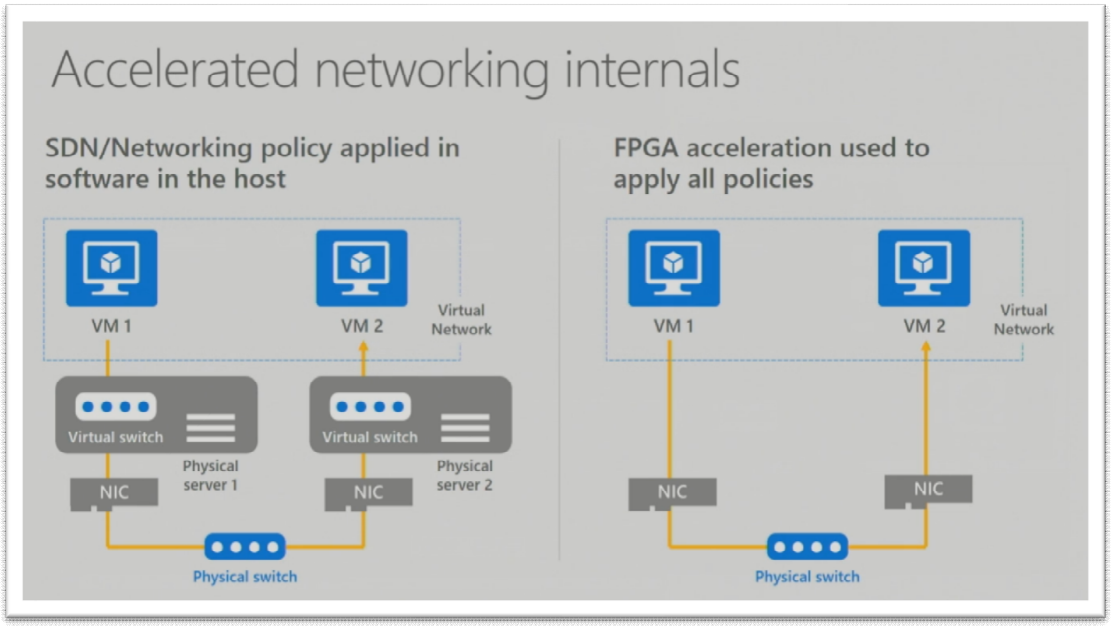
81. Microsoft Azure's RM interfaces with the Azure controller (AC) that applies policies to the data packets transmitted to and from the processors within Azure. Ex. 17 at 39-42. The RM configures a hardware switch to execute the policies set forth in the hardware access control lists (ACLs), which list grant access rights to a given application program. *Id.*

82. Under the direction of the AC, load balancing rules are practiced to pursue optimization of processing capacity by defining how tasks are distributed to FPGA Virtual Machines (*e.g.*, FPGA groupings or graphs in the parlance of Microsoft's '709 patent) (hereafter "FPGA Groupings"). *Id.* at 43-50.

83. The AC instructs the fabric how to route each data packet associated with an application program or service instance task to a given processor based on the type of task to be performed and the available processing capacity on the Azure resource pool. The AC sets up load balancing rules in the load balancing control table (LB NAT) and pushes them out to all servers hosting FPGA Groupings.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

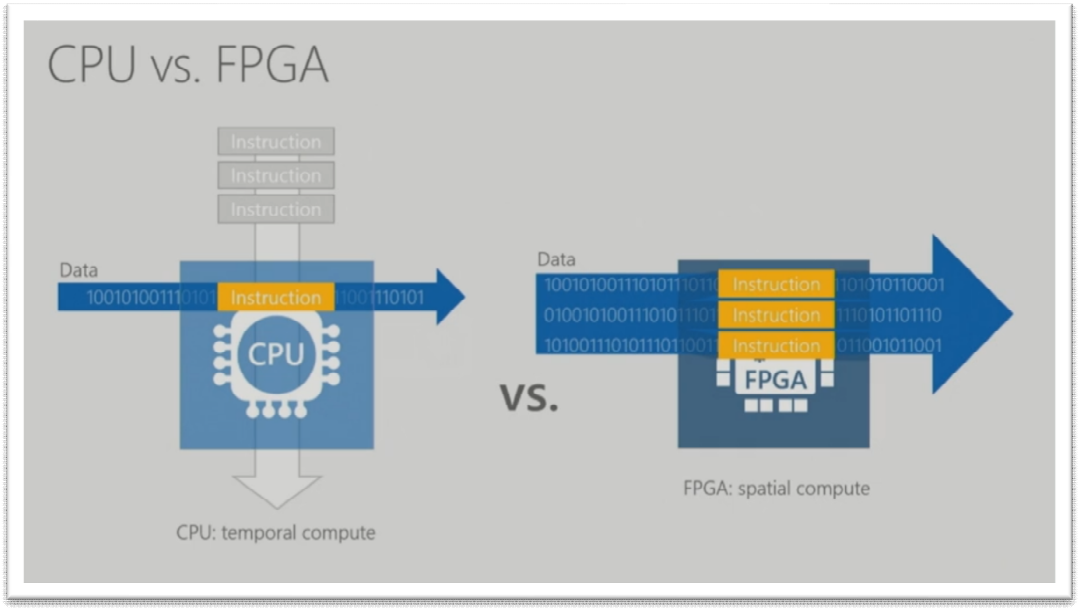
84. Under the direction of the AC, Microsoft Azure's physical switch connects directly to the network interface cards (NICs) to route packets (*e.g.*, messages) to application or service instance tasks running on the FPGA Grouping. *Id.* at 54-57. Each FPGA Grouping consists of several FPGA processors operating in a coordinated fashion under the control of the RM in conjunction with an SM, and a FPGA Manager or Node Manager (FM), as explained further below.

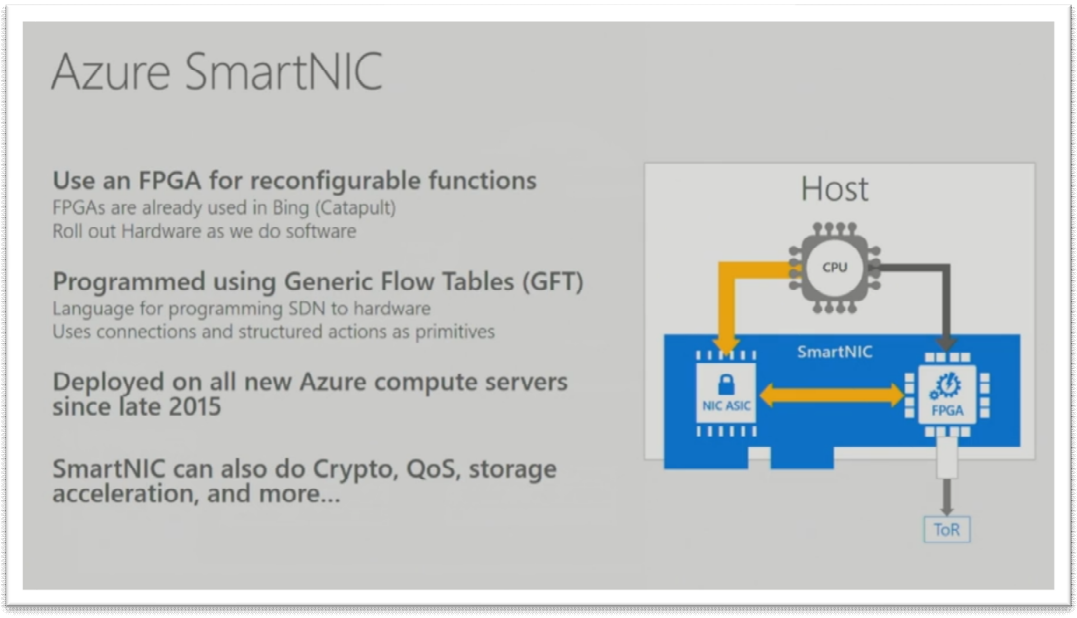


Id. at 57.

85. The FPGA processors are reconfigured to perform various functions for different application programs. *See, e.g.*, *Id.* at 19-24, 54. As explained further below, the FPGA processors are dynamically reconfigured to be suited for a given task (*e.g.*, deep neural network (DNN) application tasks, SQL database tasks, and so on).

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301



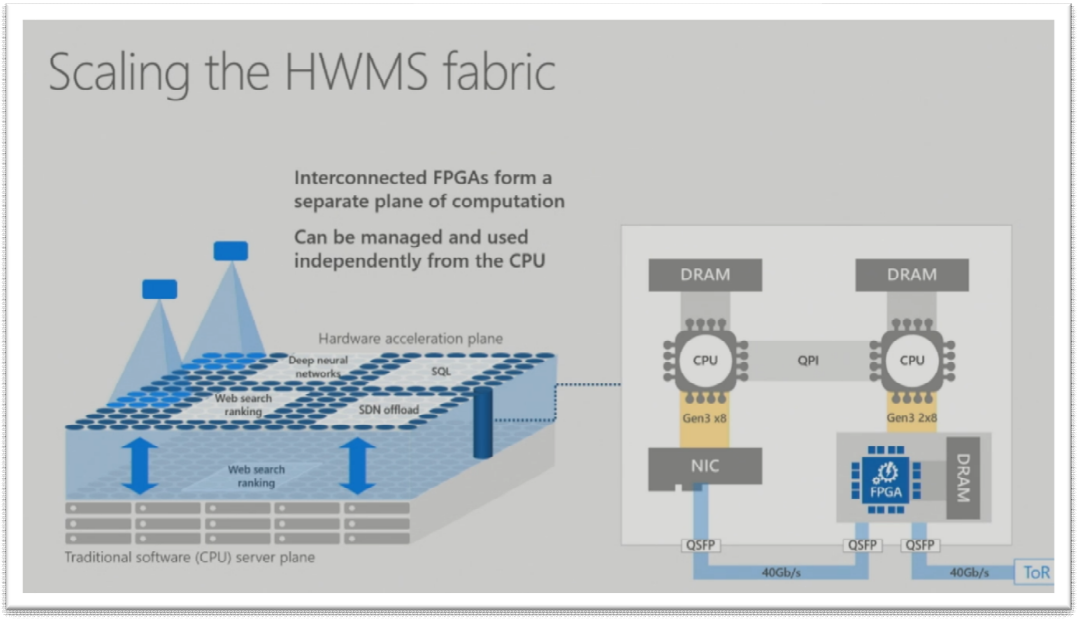


86. As shown in the above figure, an Azure Server includes an FPGA, a NIC, and a CPU. The FPGA is connected with the NIC and CPU, and the Azure Server is connected to other Azure Servers through a network, constituting a fabric of pooled FPGA accelerators. In an

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

implementation disclosed in Fig. 8 in the '709 patent, an Azure Server includes an FPGA subdivided into separate configurable domains. In this implementation, the separate configurable domains are locally connected to each other through on-chip/on-board hardware. This forms a fabric of accelerator units that may be, at least in part, internal to a subdivided FPGA (hereafter the fabrics referred to in this paragraph are referred to as "Azure Fabric").

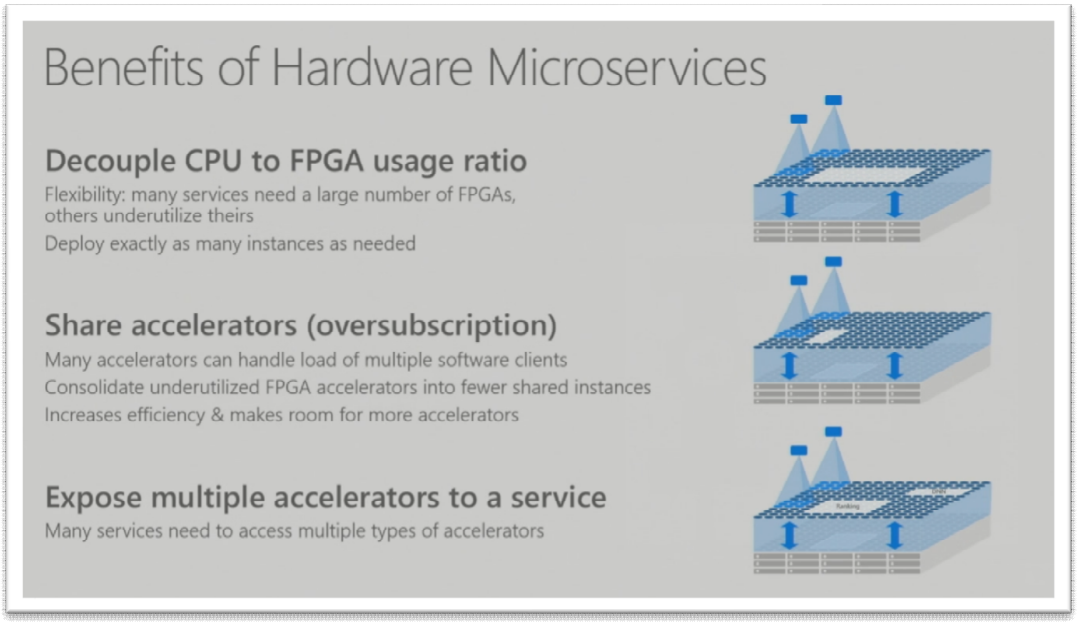
87. The FPGA processors are arranged in groups to execute tasks associated with various application programs. *Id.* at 93-98. For example, one FPGA service instance or graph, *e.g.*, an FPGA Grouping, (the blue rectangle with downwardly projecting triangle) executes deep neural network (DNN) application tasks. *Id.* Another executes SQL database tasks, yet another executes network protocol offload tasks, and the last executes web search ranking tasks. Each FPGA Grouping includes several individual FPGA processors (illustrated as blue ovals in the "hardware acceleration plane").



88. In this manner, the AC repeatedly and dynamically rearranges task assignment to

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

the array of processing units (*e.g.*, FPGA processors) while rearranging communication path connectivity for the array of processing units to optimize the application processing performance as well as the usage of processing capacity on the Azure system. *Id.* at 95-99. The FPGA Groupings (and thus the associated connectivity) changes over time in reaction to the processing demand of the various application programs.



89. Microsoft is now using this FPGA fabric architecture in every new server it deploys in its datacenters. Ex. 28 at 8. Using this architecture, Microsoft obtained 40-100 fold performance improvements in Microsoft Bing's machine learning algorithms. *Id.* Microsoft has also announced the availability of Brainwave, an FPGA-based system for ultra-low latency deep learning for Azure. *Id.*

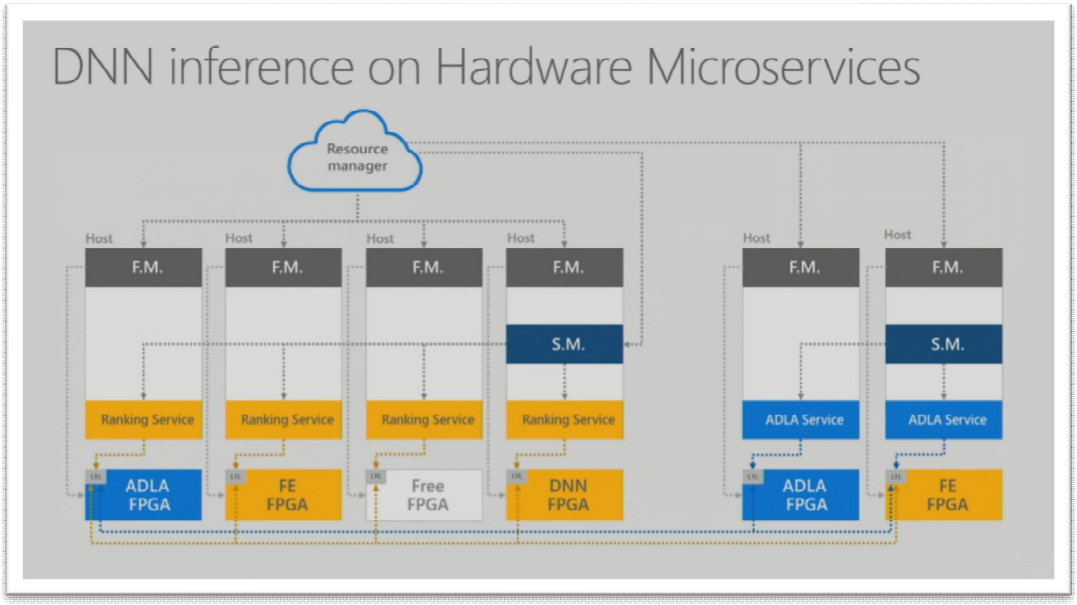
90. Using this architecture, Microsoft claims up to a 150-200 fold improvement in data processing throughput and up to a 50 fold improvement in energy efficiency. Ex. 17 at 91, Ex. 29. Latency has also lowered by about a factor of 10. Ex. 17 at 86, Ex. 29.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

91. In Azure, the RM works in conjunction with both (a) the FM on each host processing unit, which keeps track of resource allocations on the FPGA processors, and (b) the SM, which assigns computing tasks to the FPGA processors. Ex. 17 at 103-106.

92. In the implementation illustrated below, an SM in conjunction with the RM assigns one grouping of computing tasks of a ranking service application program (*e.g.*, Bing) to the FPGA processors that have been allocated. Another grouping of FPGA processors is performing tasks for the Azure Data Link Analytics (ADLA) application, with those tasks having been assigned by another SM in conjunction with the RM, and so forth. In this way, the RM, SM and FMs ensure that each FPGA processor is receiving data from the appropriate queue and executing a task corresponding to the proper application. The RM, SM and FMs also ensure that the output of each FPGA accelerator is communicated to the proper output queue. The load balancing table is utilized to allocate application loads among FPGA processors and FPGAs among loads in order to pursue optimal usage of available processing capacity. In the illustrated example, the Ranking Service is expressing higher processing demand and/or has higher prioritization resulting in it getting allocated four FPGA cores and/or FPGA Groupings, compared with *e.g.*, ADLA that is assigned two FPGA accelerators.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301



Ex. 17 at 104

93. Azure's queuing of data as it moves between memories and the FPGA processors is or has been accordant to descriptions in Microsoft's United States Patent No. 10,296,392 ("the '392 patent"). Ex. 47.

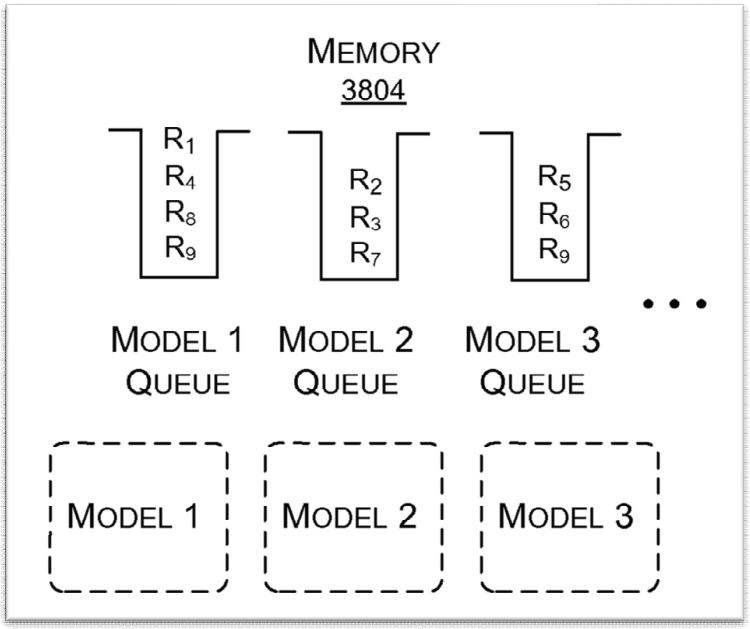
94. On information and belief, Microsoft Azure embodies or has embodied the functionality substantially as described in the '392 patent.

95. In connection with Fig. 38 (excerpted below), the '392 patent explains that different application programs such as a French language search engine, an English language search engine, and a German language search engine may have dedicated input buffers Model Queue 1, Model Queue 2 and Model Queue 3, respectively. Ex. 47 at 34:62-35:34; see also Fig. 38. Data is fed from the queues to the FPGA processors based on policies such as queue fullness. *Id*. Generally, the queue manager (in coordination with at least the SM) will prioritize the processing of queries in the queue with the most queries. As an example, shown below, Model 1 Queue has queued four

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

queries (R_1, R_4, R_8 and R_9) where Model 2 and Model 3 Queues have queued three queries. As such, Model 1 Queue would be prioritized above the other two.

96. On information and belief, Azure uses such queues to manage the transmission of data to, from and/or within the FPGA processors.



97. The distribution of tasks among the FPGA processors in Azure by the SM, RM, and FMs is also substantially described in the '392 patent.

98. In operations 1-6 illustrated in Fig. 2 of the '392 patent, each hardware acceleration component 208/210 (which may be an FPGA or a subdivision thereof) performs a task on the data and forwards the result to the next destination under the management of the head component of the FPGA Grouping (also referred to as the SM). The '392 patent further explains in connection with Fig. 36 that each acceleration component performs its respective function, such as mathematically combining feature values or compressing the data reflecting the feature values computed thus far.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

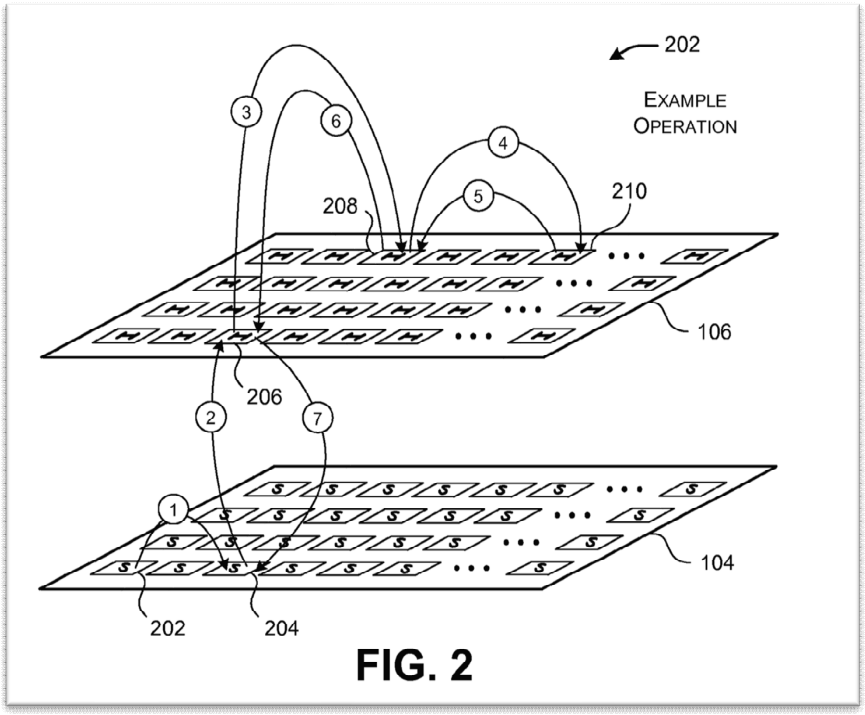


FIG. 2

99. The scheduling and placement of tasks by Azure through the RM, SMs and FMs is done taking into consideration a given task's readiness for execution. Ex. 47 at 35:20-25, Ex. 19 at 287-88.

100. On information and belief, Azure embodies or has embodied the technology described in Exhibit 19.

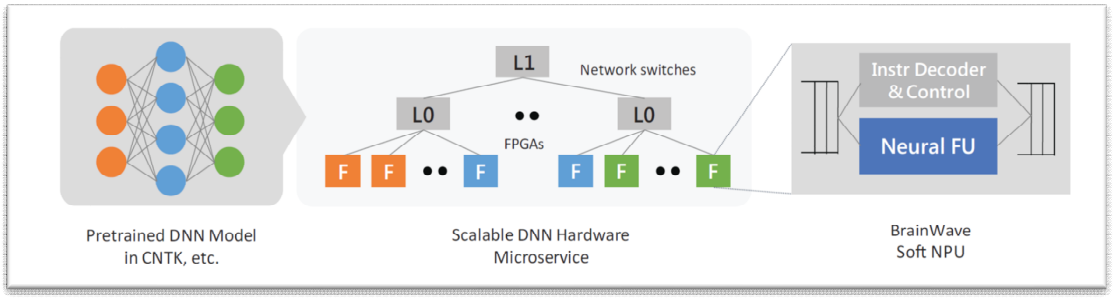
101. Microsoft Azure also supports assured resource allocations for a client's FPGA Grouping for an additional fee. *See* Ex. 48 at 3. Azure allocates FPGA processors based in part on the number of FPGA Groupings subscribed to by the client for its applications.

102. In late 2017, Microsoft began moving its deep neural networks (DNNs), which provide the backbone for their real time artificial intelligence (AI) applications, from CPU server networks to Catapult II FPGA server networks. Exs. 20, 37 *passim*, see also https://www.microsoft.com/en-us/research/ blog/microsoft-unveils-project-brainwave/. These

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

FPGA-enabled DNNs were used internally by Microsoft to support Bing and Azure in 2017 and became available to customers in 2018. Ex. 20 at 17. Microsoft uses the code-name Project Brainwave to refer to its platform that accelerates DNN inferencing, which runs onto of the Catapult servers in Azure.

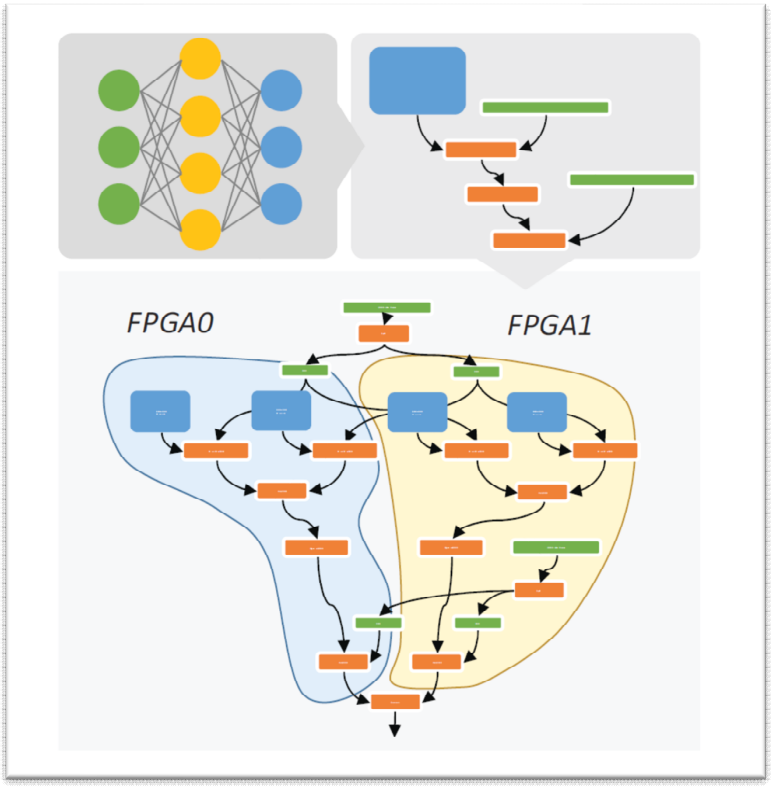
103. Microsoft's DNN models deployed on FPGA servers (*i.e.*, Brainwave deployed on the Catapult II FPGA network) utilize the multi-stage architecture described in various ThroughPuter patent applications, which were published and/or issued as patents more than three years earlier. See, *e.g.*, Ex. 20. As illustrated and explained in one of Microsoft's whitepapers on the topic (*Id.*), the DNNs are implemented as a hardware microservice having multiple stages. The first stage (illustrated in orange) is implemented on a flexible number of FPGA processing cores and passes its intermediate processing result to the second stage (illustrated in blue), which in turn passes its result on to the last stage (illustrated in green), which outputs the DNN hardware microservice result to the calling application.



Id., Fig. 2

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

104. The FPGA cores of the different FPGAs that are performing the same task constitute a stage. *Id.* In the illustration below (an excerpt of Ex. 20, Fig. 3), the tasks by the DNN model are separated into three types of tasks, illustrated in blue, orange and green. *Id.* Each of tasks is deployed on a FPGA processing core labelled FPGA0 and FPGA1. *Id.* The blue cores operate collectively to perform a first task in a first stage and pass the output to another a stage comprised of orange cores. *Id.* The green cores operate collectively to perform a second task in a second stage and pass the output to one of a set of stages comprised of orange cores. *Id.* The orange cores perform a third function and serve as the fourth, fifth and sixth stages, although those stages could be considered a single stage for certain purposes and in certain implementations. *Id.*



LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

105. In the Azure-implemented Project Brainwave, an arbitration controller or "network" manages data movement among the memory components: pipeline register files (MRF and VRFs), DRAM, and network I/O queues. Ex. 49 at 6. The arbitration network is deployed on the hardware logic of the FPGA fabric. *Id*. *passim*. This reduces observed latency by up to 90X. *Id*.

106. In 2018 or 2019, Microsoft began incorporating multiple FPGAs onto each server blade. Ex. 37 at Fig. 1, Ex. 38 at Fig. 1. In this iteration of Catapult II, on each server blade a single CPU is placed "behind" and supports several FGPAs and/or ASICs. This is yet another step in the evolution of Azure away from conventional CPUs and toward more widespread use of the FPGA network to provide an ever-increasing fraction of Azure's functions.

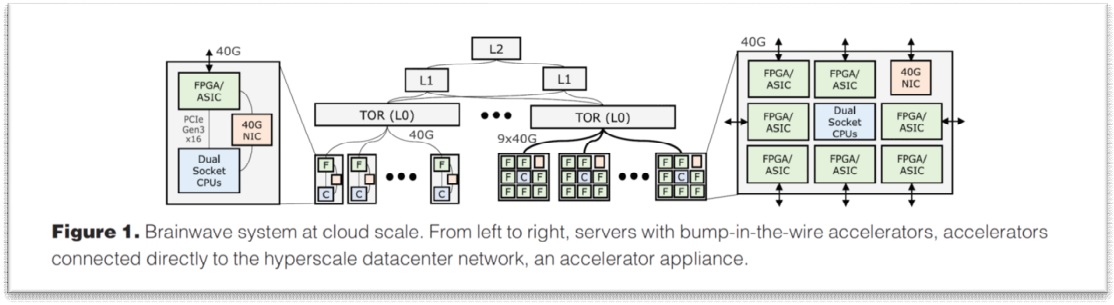


Figure 1. Brainwave system at cloud scale. From left to right, servers with bump-in-the-wire accelerators, accelerators connected directly to the hyperscale datacenter network, an accelerator appliance.

107. In August 2021, several months after the filing of the Virginia Action, one of the lead Azure engineers, Derek Chiou, published a paper describing a Persistent Recurring Neural Network (RNN) deployed on the FPGA-enabled Azure network. Ex. 25 at 6, 23. Dr. Chiou is an author on most of the whitepapers concerning Azure, including Exs. 12,15, 16, 18, 20 and 36. On information and belief, Microsoft is implementing the technology described in Exhibit 25 on at least some Azure servers.

108. RNNs are used for tasks such as speech recognition, text processing and translation. *Id*. Persistent RNNs can be considered a specialized type of DNN in which the inter-stage

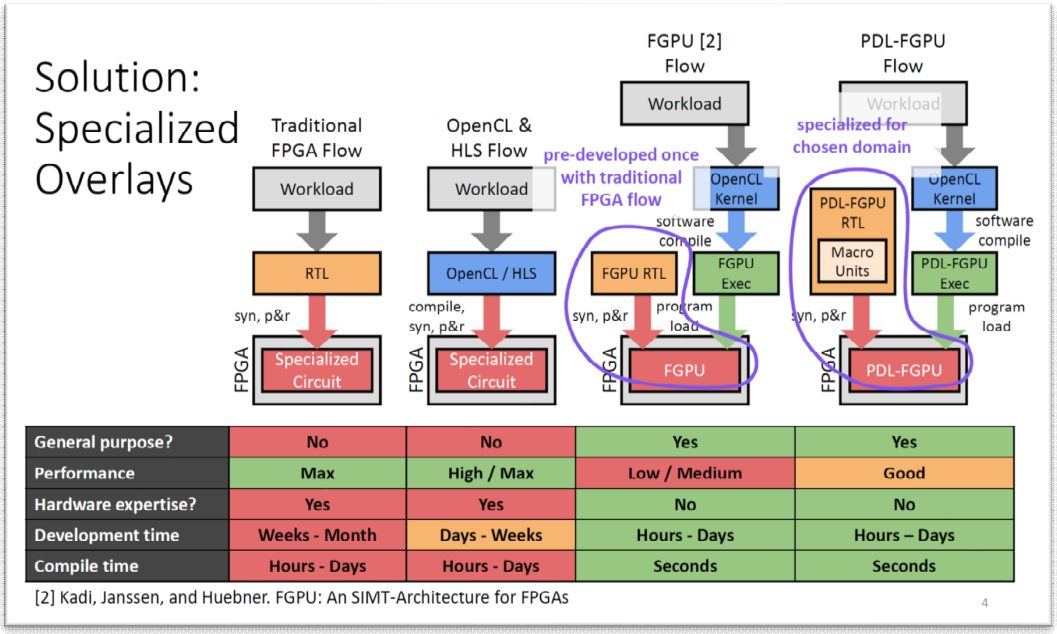
LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

connections are not static but rather change over time and operations may flow upstream, or backwards through the stages. The latter means that information persists because it can travel backwards through the stages and affect the AI model present at an ostensibly upstream node that is also processing data that was received later in time.

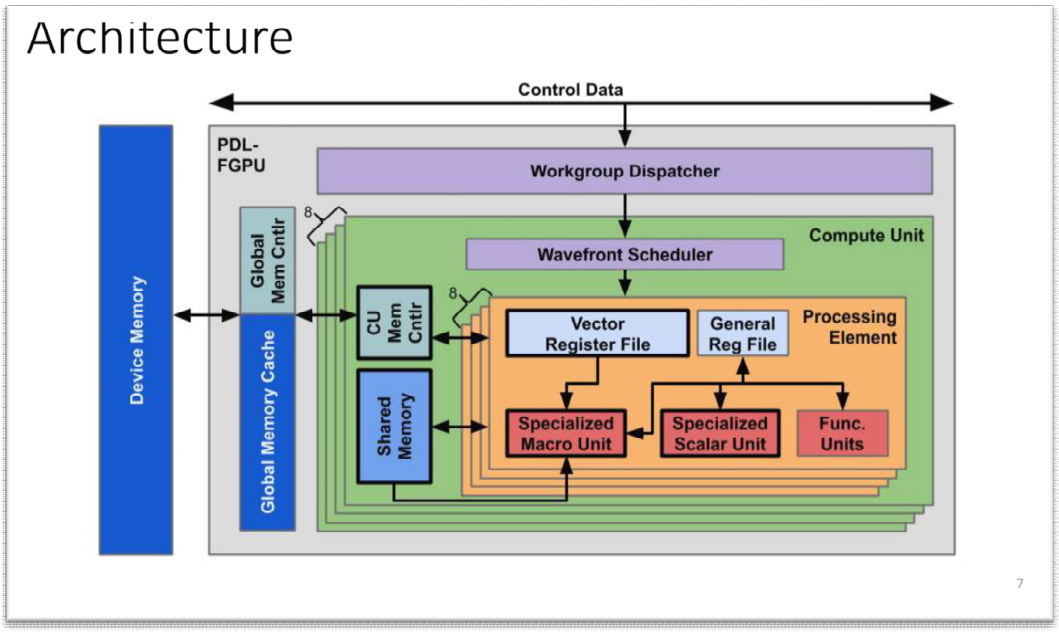
109. In its new implementation of the persistent RNN, Microsoft is using essentially a modified shell for the FGPA[2]. Microsoft's new RNN uses a graphical processing unit (GPU) instantiated on the FPGA. For this reason, it is referred to as an *FPGA general purpose Graphical Processing Unit*, or FGPU. Ex. 25 at 4, Ex. 44. Because it is used to perform persistent deep learning (PDL), it is termed a PDL-FGPU. On information and belief, Microsoft currently deploys PDL-FGPUs on some servers to provide Persistent RNNs on Catapult II.

[2] Recall that shells define the functionality that the FPGA's programmable network of multiplexer-switched ALUs are to perform.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301



110. The PDL-FGPU architecture or shell is shown below. As illustrated, the shell controls memory access, including both SRAM on each FPGA and DRAM on each server blade, and communication between each Processing Element (core) in the Compute Unit.



Ex. 25 at 8.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

111. On information and belief, the PDL-FGPU architecture or shell is deployed in conjunction with the Catapult II Shell Components. The PDL-FGPU rides "on top" of the Catapult II Shell Components, thereby allowing the Catapult II Shell-Components to mediate inter-FPGA communication and memory access. On information and belief, Microsoft currently uses this approach on at least some servers.

112. Deploying the RNNs on the FPGA-enabled Azure network via PDL-FGPUs provides a one to three order of magnitude improvement in throughput over Microsoft's internal baseline, which on information and belief is an implementation on a CPU network. See Ex. 25 at 13-17. On information and belief, Microsoft currently deploys PDL-FGPUs on at least some FPGA-enabled Azure network servers to provide Persistent RNNs.

113. In June 2015, Microsoft filed the "'800 application".

114. Microsoft's Azure cloud platform includes or has included the functionality described in the '800 application.

115. In its '800 application, Microsoft explains that "graphs" or groups of FPGA processors (each designated with an H) are logically grouped together by a hardware layer fabric 106.

116. Each group of processors in Microsoft's '800 application can be dynamically reallocated and reconfigured to different application programs, such as **speech translation (blue)**, **encryption (red)**, **search ranking (yellow)** or **compression (green)**, as seen in Microsoft Fig. 15D below).

117. These reconfigurations are facilitated by the fact that the fabric provides addresses 1527 that identify the physical locations of each processing unit and the processing group ("graph") to which a given processing unit is currently allocated.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

118. The reconfigurations are performed, for instance, to accommodate increased demand for or from tasks of a given application program. In the system disclosed in the '800 application, service manager 1521 will repeatedly reallocate and reconfigure the processor groups or "graphs". In one example, as shown in Fig. 15D below, the service manager reallocates processor cores that were being used by the **encryption (red)** application (executed by graph 1503) to the **speech translation (blue)** application (executed by graph 1502). *Compare* Fig. 15C above *with* Fig. 15D below.

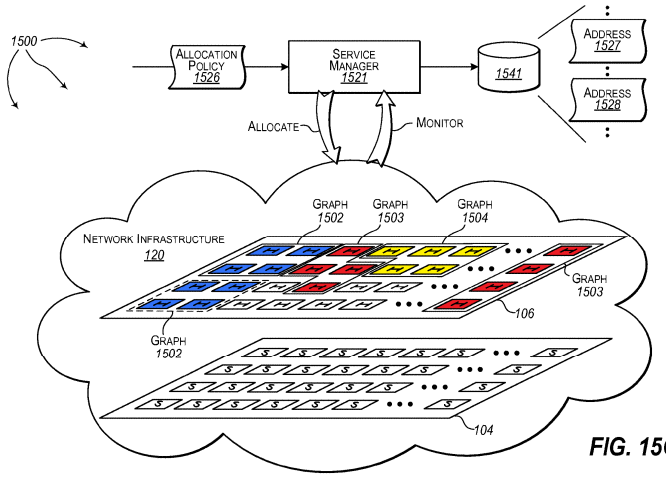


FIG. 15C

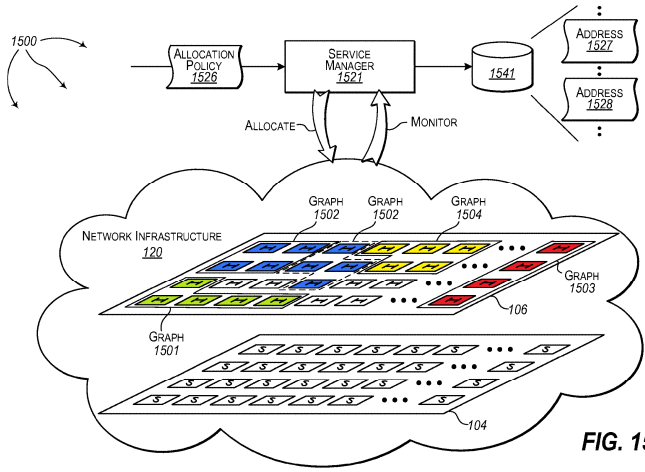


FIG. 15D

Ex. 13, Figs. 15C and 15D (annotated)

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

119. For this reallocation, the FPGA processor cores are reconfigured to have the functionality associated with the tasks being processed by the **speech translation (blue)** application or service (executed by graph 1502).

120. The service manager simultaneously reallocates FPGA cores from the speech translation group to accommodate increasing demand for a different processing task, for example **data compression (green, graph 1501)** as shown in the illustration above.

121. For this reallocation period, the processor groups handling, for example, **search ranking (yellow, graph 1504)** are not reallocated.

122. As illustrated above, the fabric addresses 1527 are updated to reflect the new processing core allocations.

123. As can be seen in claim 1 of the patent which issued from the '800 application (the '709 patent, attached as Exhibit 13), Microsoft claims the functionality discussed, *supra*, at paragraphs 67-74.

124. On information and belief, Azure currently uses the technology described in its '709 patent and the Microsoft publications cited herein to manage at least a portion of its Azure workloads.

125. On information and belief, at least a portion of those workloads are handled at one or more of Microsoft's datacenter complexes within this District.

126. Claim 1 of the '709 patent is directed to a system that includes at least two sets of hardware acceleration components with each one capable of accelerating a different task or service as well as a processor and memory that maintains the mapping of each physical core address and group assignment and reallocates the groupings based on variation in demand for a given group or core type.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

127. Microsoft's **application tasks (S) such as search ranking** serves the same or similar purpose as the **application tasks 240** in ThroughPuter's representative '090 patent, discussed above. That is, both represent tasks to be performed on a hardware based processing system.

128. Microsoft's **hardware accelerators (H)** achieve the same or similar function as the **reconfigurable core slots 120** in ThroughPuter's representative '090 patent. That is, both represent reconfigurable processors on which the tasks can be performed.

129. **Microsoft's service managers 1521** (in coordination with the Resource Manager and the FPGA Managers) perform the same or similar function as the **controller 140 and/or the process 300**, in an embodiment of ThroughPuter's representative '090 patent, discussed above. That is, both assign tasks to an allocated group of reconfigurable processors.

130. Microsoft's **addresses 1527** serves the same or similar purpose as the **mappings shown in Figure 3 and Tables 4-5** of ThroughPuter's representative '090 patent, discussed above. That is, both provide addresses for the allocated processors to which the tasks are assigned.

131. The substantial identity between Microsoft's '709 patent and ThroughPuter's invention can be further appreciated from a side-by-side comparison of the patent claims granted to Microsoft and ThroughPuter. The column on the left shows claim 1 of the '306 patent, which claims priority to applications dating back to 2012. The column on the right shows the text of dependent claim 5 (including the limitations of independent claim 1) of Microsoft's '709 patent, which claims priority to an application filed in 2015. As can be appreciated from this side-by-side comparison, Microsoft obtained a patent on substantially the same technology taught by ThroughPuter's patent applications. However, ThroughPuter's'306 patent is entitled to a priority date that is at least three years earlier than Microsoft's.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

ThroughPuter's U.S. Patent No. 10,936,306, Independent Claim 1	Microsoft's U.S. Patent No. 10,270,709, Dependent Claim 5 (including the limitations of Independent Claim 1)
A method for task-switching on a multi-user parallel processing hardware architecture comprising a plurality of reconfigurable logic-based processing units, the method comprising:	A method in a data center comprising a plurality of servers interconnected via a network, wherein each of the plurality of servers comprises at least one acceleration component, wherein the at least one acceleration component comprises a plurality of hardware logic blocks interconnected using reconfigurable interconnects, the method comprising:
linking, through a first set of inter-task communication paths of the multi-user parallel processing hardware architecture, a first set of reconfigurable logic-based processing units of the plurality of reconfigurable logic-based processing units into a first multi-stage program instance configured to execute a corresponding set of interdependent processing stages of a first program, wherein each of the first set of reconfigurable logic-based processing units is programmed to perform a respective task of a plurality of first tasks of the first program corresponding to a respective processing stage of the first multi-stage program instance;	linking a first set of acceleration components corresponding to a first set of the plurality of the servers into a first graph configured to accelerate a first service, wherein each of the first set of acceleration components is programmed to perform any of a first set of roles corresponding to the first service;
linking, through a second set of inter-task communication paths of the multi-user parallel processing hardware architecture, a second set of reconfigurable logic-based processing units of the plurality of reconfigurable logic-based processing units into a second multi-stage program instance configured to execute a corresponding set of interdependent processing stages of a second program, wherein each of the second set of reconfigurable logic-based processing units is programmed to perform a respective task of a plurality of second tasks of the second program corresponding to a respective processing stage of the second multi-stage program instance;	linking a second set of acceleration components corresponding to a second set of the plurality of the servers into a second graph configured to accelerate a second service, different from the first service, wherein each of the second set of acceleration components is programmed to perform any of a second set of roles corresponding to the second service;
maintaining, in a storage, a first location for a first reconfigurable logic-based processing unit of the first set of reconfigurable logic-based	maintaining in a storage a first address for the first graph such that the first service can request

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

processing units executing the first multi-stage program instance such that a first one or more users and/or programs are enabled to communicate directly with the first multi-stage program instance;	hardware acceleration from the first set of acceleration components;
maintaining, in the storage, a second location for a second reconfigurable logic-based processing unit of the second set of reconfigurable logic-based processing units executing the second multi-stage program instance such that a second one or more users and/or programs are enabled to communicate directly with the second multi-stage program instance; and	maintaining in the storage a second address for the second graph such that the second service can request hardware acceleration form the second set of acceleration components; and
in response to an increased demand for the second program, reallocating, by a controller comprising software and/or hardware logic configured to implement a load-adaptive allocation policy, at least one processing unit of the first set of reconfigurable logic-based processing units, the reallocating resulting in (1) switching the at least one of the first set of processing units from performing a task of the plurality of first tasks to performing one task of the plurality of second tasks, wherein switching comprises matching a first programming configuration of the at least one of the first set of processing units to a programming configuration demanded by the one task, and (2) adjusting, in storage, at least one of the first location or the second location to enable, through the respective location, direct communication to the other multi-stage program instance of the first multi-stage program instance or the second multi-stage program instance;	in response to an increased demand for hardware acceleration from the second graph, based on an allocation policy, a service manager comprising computer-executable instructions: (1) switching, using configuration data and without loading a new image file, at least one of the first set of acceleration components from performing any of the first set of roles to performing any of the second set of roles, and (2) adjusting at least one of the first address or the second address
wherein the load-adaptive allocation policy is configured to facilitate minimizing reconfiguring the plurality of reconfigurable logic-based processing units.	wherein each of the first set of acceleration components and the second set of acceleration components is included in a hardware plane, and wherein the allocation policy is configured to facilitate minimizing switching of roles among acceleration components in the hardware plane.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

132. On information and belief, Microsoft believed at the time of filing and still believes that the subject matter claimed in the '709 patent is patent eligible under 35 U.S.C. § 101.

133. On information and belief, Microsoft believed at the time of filing and still believes that this subject matter is novel under 35 U.S.C. § 102 in view of the references disclosed to and considered by the United States Patent & Trademark Office ("USPTO") during examination of their '800 application.

134. On information and belief, Microsoft believed at the time of filing and still believes that this subject matter is non-obvious under 35 U.S.C. § 103 in view of the references Microsoft disclosed to and considered by the USPTO during examination of their '800 application.

135. To summarize, since the introduction of Catapult II in 2015, Microsoft has transitioned a broad swath of workloads and the associated management responsibility from CPUs over to the low latency FPGA network fabric of Catapult II, per ThroughPuter's patents. In its current form, Catapult II functions as stand-alone, almost entirely independent FPGA fabric that handles the majority of the workload in each Azure datacenter. That independent FPGA network is one of the innovations described in ThroughPuter's patents that has been deliberately copied by Microsoft, including by filing patent applications wrongly claiming these advances as having been invented by Microsoft. In fact, they were disclosed to Microsoft by ThroughPuter years before Microsoft began filing its related patent applications.

136. However, Microsoft was right about one thing – this new FPGA fabric architecture is a "major advance" that has a "profound" "impact [on] the types of workloads that can be accelerated and the scalability of the [system]." Ex. 16 at 54, 58. This design improvement goes "far beyond just an improved network design" and "can be seen as a fundamental shift in the role of CPUs in the datacenter." *Id.* at 58-59.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

137. By using the new FPGA-network of Catapult II, as of 2017, Microsoft achieved up to a 150-200 fold improvement in data processing throughput and up to a 50 fold improvement in energy efficiency. Ex. 17 at 91. Also by that time aggregate latency had been lowered by about a factor of 10. *Id*. On information and belief, in the ensuing four years the evolutions described above (incorporating yet additional functionality described in the ThroughPuter patents) have improved each of these performance metrics by at least another order of magnitude.

138. It's no wonder that Microsoft's CEO described the new FPGA reconfigurable fabric as "magic." Ex. 31 at 9.

MICROSOFT'S KNOWLEDGE OF THROUGHPUTER AND ITS PATENTS

139. Starting in 2011, ThroughPuter has invested in development and patent protection of its intellectual property consistently, in order to protect the competitive advantage to which it is entitled under the law as the inventor of the technology at issue in this case. Recognizing the promise of its technology and the benefits it could provide to Microsoft, ThroughPuter made consistent overtures to Microsoft in an effort to collaborate.

140. For example, as discussed above, starting in 2013, Mr. Sandstrom corresponded with Microsoft's Director of Client and Cloud Applications (Dr. Burger: a named inventor on Microsoft's U.S. Patent No. 11,819,657, which is discussed further in Count 1 below) and others on Microsoft's cloud computing team concerning a potential collaboration between ThroughPuter and Microsoft. *See* Ex. 21.

141. At least through that correspondence, and no later than February 2013, Microsoft's cloud computing team was apprised of the technical details of ThroughPuter's Dynamic Parallel Execution Environment™ (DPEE).

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

142. At least through that correspondence, and no later than February 2013, Mr. Sandstrom provided Microsoft's cloud computing team a copy of the DPEE system description that corresponds with the preferred embodiments disclosed in ThroughPuter's patents. *See* Ex. 22.

143. From 2013 through 2016, ThroughPuter's President Mr. Sandstrom continued to correspond with Microsoft's cloud computing team concerning a potential collaboration between Microsoft and ThroughPuter. *See, e.g.*, Exhibit 32.

144. For example, in correspondence dated May 12, 2015, in response to a communication from ThroughPuter advising Microsoft of its patent portfolio, which at that time included 25 granted U.S. and UK patents, Dr. Burger stated that he had "forwarded [ThroughPuter's] note onto the relevant people looking at any IP and they will contact you directly if interested. Doug." Ex. 30.

145. Shortly thereafter Microsoft filed its '800 application (on June 26, 2015) directed to the same subject matter as ThroughPuter's technology naming Dr. Burger, to whom Mr. Sandstrom disclosed ThroughPuter's technology, as an inventor. Following the filing of that application, the leader of Microsoft's cloud computing team declined ThroughPuter's offers for collaboration, saying in an August 3, 2015 email to Mr. Sandstrom: "I don't think we can proceed … we already have our plates very full. Thanks and best wishes." Ex. 32.

146. Thereafter, Mr. Sandstrom reminded Microsoft several times that the solution he shared was the subject of several U.S. and foreign patents.

147. In 2015, Mr. Sandstrom had a discussion with Microsoft representative, Dr. Derek Chiou, during which conversation, Dr. Chiou made clear that despite Microsoft's knowledge of ThroughPuter's technology and its patent protection, Microsoft would not be willing to explore

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

any business relationship with ThroughPuter, including with respect to ThroughPuter's intellectual property rights.

148. On May 16, 2016, Mr. Sandstrom informed Arun Justus at Microsoft that "ThroughPuter owns the IP rights for the key techniques that will be necessary in realizing any scalable solution for this must-solve challenge." Ex. 33.

149. In addition, on May 19, 2016, Mr. Sandstrom wrote Aki Siponen at Microsoft: "[A]ll cloud service providers will be facing the fundamental scalability solution . . . and ThroughPuter holds the key patented techniques that will be needed in delivering an effective solution to this must-solve challenge." *Id*.

150. On May 17, 2017, via a LinkedIn message, Mr. Sandstrom provided notice to Dr. Burger of the issuance of the '833 and '090 patents, asserted herein under Counts 10 and 11. In response to Mr. Sandstrom's message, Dr. Burger, stated "Thanks Mark, appreciate the nice message and will definitely review."

151. In a 2017 Microsoft Research publication titled "The Feniks FPGA Operating System for Cloud Computing," Microsoft describes its experimental development of FPGA processors with a hardware based operating system in highly similar terms as ThroughPuter in the above-referenced, earlier patent disclosures as well as publications such as the one presented by Mr. Sandstrom at FPGAworld in 2014. *Compare e.g.,* Ex. 43 *with* Ex. 27.

152. In that 2017 publication, Dr. Chiou is credited with the "initial exploration which provides valuable experience for the system design." (Ex. 27 at 7, Acknowledgement). Dr. Chiou is the same individual with whom ThroughPuter spoke in 2015 after Microsoft had acknowledged ThroughPuter's technology and its patent protection.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

153. On January 2, 2018, Mr. Sandstrom sent a letter (the "January 2018 letter") to Kevin Scott, Chief Technology Officer of Microsoft Corporation, informing Microsoft of the existence of ThroughPuter's patent portfolio and invited a business discussion.

154. On September 1, 2018, outside patent counsel for ThroughPuter sent Microsoft Chairman John W. Thompson a letter ("the September 2018 letter") offering Microsoft the opportunity to license or acquire "ThroughPuter's Dynamic Parallel Execution (DPE) technology, protected by a portfolio of forty-eight (48) patents issued and pending worldwide ….". The September 2018 letter further explained that to the extent Microsoft's current or planned products were built on certain cloud and enterprise computing technologies, "Microsoft needs to obtain a license from ThroughPuter for continued as well as planned future usage of [ThroughPuter's] patents ….".

155. The September 2018 letter specifically brought to Microsoft's attention "claims 19 and 34 of U.S. Patent No. 9,632,833, claims 1 and 9 of U.S. Patent No. 9,424,090, and claims 1, 12 and 22 of U.S. Patent No. 10,061,615, along with a number of the patents' dependent claims," further noting "two Track 1 (expedited) patent applications, U.S. application serial nos. 16/014,658 and 16/014,674, each of which has received a notice of allowance."

156. On September 1, 2018, ThroughPuter's outside patent counsel sent the same letter referred to in the preceding paragraph to Microsoft's then-General Counsel.

157. Microsoft did not respond to any of the letters referred to in the four preceding paragraphs.

158. Prior to the filing of the Virginia Action, Microsoft had not communicated with ThroughPuter since ThroughPuter advised Microsoft of the existence of U.S. Patent Nos.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

9,424,090 and 9,632,833 in May 2017, which patents claim priority to patent applications filed in 2012.

159. On information and belief, Microsoft cut off communication with ThroughPuter in or around May 2017 because Microsoft realized that ThroughPuter had been issued patents with earlier priority dates on the technology approach underlying the Microsoft Azure PaaS and its development plans.

160. On information and belief, Dr. Burger and Microsoft's other personnel were instructed to stop communicating with ThroughPuter because Microsoft recognized the similarities between ThroughPuter's patent protected technology and the execution layer of the Azure PaaS and its development plans.

161. Rather than respect ThroughPuter's intellectual property rights, Microsoft made, upon information and belief, the deliberate decision to infringe ThroughPuter's patents. This conscious decision prevents ThroughPuter from being able to compete in the cloud computing space all while Microsoft used the enabling ThroughPuter technology to scale up the world's largest cloud-computing platform. ThroughPuter's patented technology allows Microsoft to claim up to 200-fold technical performance gains.

162. Microsoft recognized the novelty of ThroughPuter's technology by attempting to claim ThroughPuter's technology as its own. Specifically, as discussed herein, Microsoft sought and obtained patent protection on the fundamental technologies invented by ThroughPuter years earlier.

163. During the pendency of the patent application that matured into the '709 patent, Microsoft disclosed to the USPTO over two hundred fifty (250) documents that Microsoft

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

represented as possible "prior art" to its alleged invention, but failed to disclose any information concerning ThroughPuter or its patents to the USPTO despite knowledge of the same.

164. During prosecution of Microsoft's '709 patent, Microsoft did not disclose to the USPTO ThroughPuter's '090 patent.

165. During prosecution of Microsoft's '709 patent, Microsoft did not disclose to the USPTO ThroughPuter's '833 patent.

166. During prosecution of Microsoft's '709 patent, Microsoft did not disclose to the USPTO any of the other ThroughPuter patent assets referenced in the materials provided to Microsoft by ThroughPuter.

167. During prosecution of Microsoft's '709 patent, Microsoft did not disclose to the USPTO any information concerning ThroughPuter or its DPEE technology.

168. Upon information and belief, to date, Microsoft has not disclosed any information concerning ThroughPuter or its technologies to the USPTO in connection with any Microsoft patent application related to Azure.

169. On information and belief, Microsoft has been monitoring ThroughPuter's patent portfolio since at least 2017 and has had actual notice of all ThroughPuter patents that issued since that time.

170. On information and belief, Microsoft has been monitoring ThroughPuter's patent portfolio since at least being served with the complaint in the Virginia Action and has had actual notice of all ThroughPuter patents that issued since that time.

171. On information and belief, Microsoft has been monitoring ThroughPuter's patent portfolio since at least the time it was served with the complaint in the Virginia Action.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

172. Despite this knowledge, Microsoft has made the deliberate choice to refuse any business arrangement or resolution with ThroughPuter, opting instead to infringe ThroughPuter's patents, thereby preventing ThroughPuter from competing in the market ThroughPuter's technology has enabled.

173. On information and belief, Microsoft's infringement of the '090 and '833 patents has been continuous, deliberate and in willful disregard of ThroughPuter's patent rights.

174. Microsoft's ongoing infringement of the '090, '833, '948, '556, '306, '998, '644, '242, '353, '902, and '559 patents is continuous, deliberate, and in willful disregard of ThroughPuter's patent rights.

COUNT 1

(Infringement of U.S. Patent No. 11,150,948)

175. ThroughPuter repeats and realleges each and every allegation contained above as though fully set forth herein.

176. On October 19, 2021, the USPTO duly and legally issued the '948 patent, entitled "Managing Programmable Logic-Based Processing Unit Allocation on a Parallel Data Processing Platform." A copy of the '948 patent is attached as Exhibit 1.

177. Mark Sandstrom is the sole and true inventor of the '948 patent.

178. ThroughPuter, Inc. owns all right, title, and interest to and in the '948 patent.

179. Microsoft infringes claims 1-15 of the '948 patent.

180. As demonstrated in the side-by-side comparison below, claim 1 of ThroughPuter's '948 patent closely matches the claims of Microsoft's '657 patent. Ex. 42. In the chart below, the column on the left shows claim 1 of the '948 patent, which claims priority to applications dating back to 2012. The column on the right shows independent claim 8 of Microsoft's '657 patent,

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

which claims priority to an application filed in 2015. As can be appreciated from this side-by-side comparison, Microsoft obtained a patent on substantially the same technology taught by ThroughPuter's patent applications. However, ThroughPuter's '948 patent is entitled to a priority date that is over two years earlier than Microsoft's '657 patent.

ThroughPuter's U.S. Patent No. 11,150,948 Independent Claim 1	Microsoft's U.S. Patent No. 10,819,657, Independent Claim 8
A method on a parallel data processing platform comprising a plurality of programmable logic-based processing units for accelerating a service comprising at least a first program and a second program, different than the first program, the method comprising:	A method in a data center comprising a pool of acceleration components for accelerating the service comprising at least a first service functionality and a second service functionality, different from the first functionality, the method comprising:
forming a first set of inter-task communication paths connecting a first set of programmable logic-based processing units of the plurality of programmable logic-based processing units into a first multi-stage program instance, wherein the first multi-stage program instance is configured to accelerate the first program, and wherein the first set of programmable logic-based processing units are interconnected using reconfigurable cross-connects;	forming a first graph comprising a first set of interconnected acceleration components, wherein the first graph is configured to accelerate the first service functionality corresponding to the service, and wherein each of the first set of acceleration components comprises a programmable logic array comprising hardware logic blocks interconnected using reconfigurable interconnects;
forming a second set of inter-task communication paths connecting a second set of programmable logic-based processing units of the plurality of programmable logic-based processing units to a second multi-stage program instance, wherein the second multi-stage program instance is configured to accelerate the second program, and wherein the second set of programmable logic-based processing units are interconnected using reconfigurable cross-connects;	forming a second graph comprising a second set of interconnected acceleration components, wherein the second graph is configured to accelerate the second service functionality corresponding to the service, and wherein each of the second set of acceleration components comprises a programmable logic array comprising hardware logic blocks interconnected using reconfigurable interconnects;
in response to a first increased demand for the first multi-stage program instance determined by monitoring a first characteristic corresponding to the first set of programmable logic-based processing units, based on an adaptive optimized resource allocation policy, adding at least one more programmable logic-based processing unit to the first set of programmable logic-based processing units from the plurality of programmable logic-based processing units; and	in response to a first increased demand for hardware acceleration from the service determined by monitoring a first characteristic corresponding to the first set of interconnected acceleration components, based on an allocation policy, a service management component adding at least one more acceleration component to the first set of interconnected acceleration components from the pool of acceleration components;

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

in response to a second increased demand for the second multi-stage program instance determined by monitoring a second characteristic different from the first characteristic, corresponding to the second set of programmable logic-based processing units, based on the adaptive optimized resource allocation policy, adding at least one more programmable logic-based processing unit to the second set of programmable logic-based processing units from the plurality of programmable logic-based processing units.	and in response to a second increased demand for hardware acceleration from the service determined by monitoring a second characteristic, different from the first characteristic, corresponding to the second set of interconnected acceleration components, based on the allocation policy, the service management component adding at least one more acceleration component to the second set of interconnected acceleration components from the pool of acceleration components.
based on at least the determination, the given target programmable logic core redirects the request to another programmable logic core implementing the designated task.	based on at least the determination, the target hardware acceleration device redirects the request to another hardware acceleration device implementing the designated hardware accelerated service.
configuring the selected field-programmable gate arrays to process a respective processing stage of a respective requesting instance, and	configuring the selected field-programmable gate arrays of the plurality to perform the individual stages of the different functions; and
configuring certain selected field-programmable gate arrays to support communicating, by the task executing on the respective field-programmable gate array, final results to a requesting client over a network in the data processing system.	configuring certain selected field-programmable arrays to communicate final results of the different functions to the requesting server unit components, the certain selected field-programmable arrays communicating the final results to the requesting server unit components over a network in the data processing system.

181. On information and belief, Microsoft believed at the time of filing and still believes that the subject matter claimed in the ’657 patent is patent eligible under 35 U.S.C. § 101.

182. On information and belief, Microsoft believed at the time of filing and still believes that the subject matter claimed in the ’657 patent is novel under 35 U.S.C. § 102 in view of the references disclosed to and considered by the USPTO during examination of the underlying application.

183. On information and belief, Microsoft believed at the time of filing and still believes that the subject matter claimed in the ’657 patent is non-obvious under 35 U.S.C. § 103 in view of the references Microsoft disclosed to and considered by the USPTO during examination of the

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

underlying application.

184. Claim 1 of the ’948 patent is representative of the claims infringed by Microsoft and recites:

> 1. A method on a parallel data processing platform comprising a plurality of programmable logic-based processing units for accelerating a service comprising at least a first program and a second program, different than the first program, the method comprising:
>
> forming a first set of inter-task communication paths connecting a first set of programmable logic-based processing units of the plurality of programmable logic-based processing units into a first multi-stage program instance, wherein the first multi-stage program instance is configured to accelerate the first program, and wherein the first set of programmable logic-based processing units are interconnected using reconfigurable cross-connects;
>
> forming a second set of inter-task communication paths connecting a second set of programmable logic-based processing units of the plurality of programmable logic-based processing units to a second multi-stage program instance, wherein the second multi-stage program instance is configured to accelerate the second program, and wherein the second set of programmable logic-based processing units are interconnected using reconfigurable cross-connects;
>
> in response to a first increased demand for the first multi-stage program instance determined by monitoring a first characteristic corresponding to the first set of programmable logic-based processing units, based on an adaptive optimized resource allocation policy, adding at least one more programmable logic-based processing unit to the first set of programmable logic-based processing units from the plurality of programmable logic-based processing units; and
>
> in response to a second increased demand for the second multi-stage program instance determined by monitoring a second characteristic different from the first characteristic, corresponding to the second set of programmable logic-based processing units, based on the adaptive optimized resource allocation policy, adding at least one more programmable logic-based processing unit to the second set of programmable logic-based processing units from the plurality of programmable logic-based processing units.

185. On information and belief, Azure is implemented in a manner that meets each and every limitation of claim 1 of the ’948 patent.

186. On information and belief, Azure practices the invention recited in claim 8 of Microsoft’s ’657 patent.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

1. A method on a parallel data processing platform comprising a plurality of programmable logic-based processing units for accelerating a service comprising at least a first program and a second program, different than the first program, the method comprising:

187. Azure comprises a parallel data processing system with a plurality of programmable logic-based processing units (FPGA processors) for accelerating multiple applications such as DNNs, RNNs, compression, web search ranking, speech recognition, and SDN. In some applications, a single application or service (*e.g.*, speech recognition) may have multiple implementations (*e.g.*, English and German) associated with it. In Azure, applications or services are often broken down into smaller functional units that are executed (accelerated) separately.

forming a first set of inter-task communication paths connecting a first set of programmable logic-based processing units of the plurality of programmable logic-based processing units into a first multi-stage program instance, wherein the first multi-stage program instance is conf5igured to accelerate the first program, and wherein the first set of programmable logic-based processing units are interconnected using reconfigurable cross-connects;

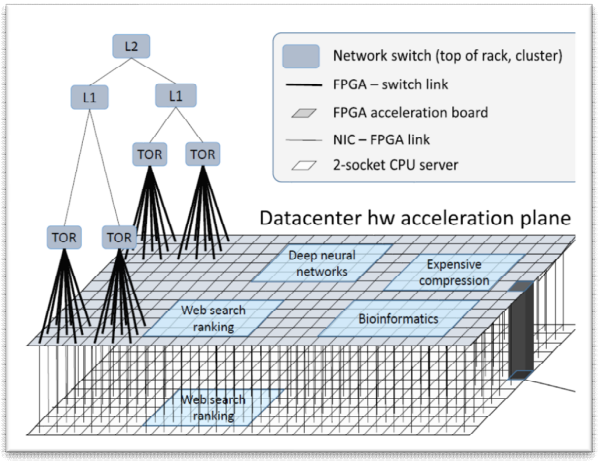
188. Microsoft uses Azure to form a first set of inter-task communication paths connecting a first set of programmable logic-based processing units of the plurality of programmable logic-based processing units into a first multi-stage program instance.

189. Azure includes multiple FPGAs, which are programmable logic-based processing units. Microsoft connects a first set of FPGAs into a multi-stage program instance comprised of multiple FPGA cores. Every FPGA in an Azure datacenter is able to communicate with every other FPGA in a datacenter through a high-speed, low latency, multiplexer switched fabric. Exs. 12, 16, 36 *passim*.

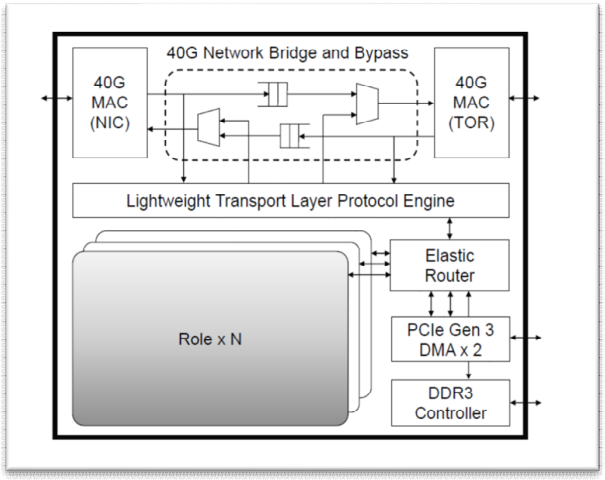
190. Each Azure rack in a datacenter is connected by a two-tier switching network L1/L2 as shown below. Exs. 12, 16, *passim*. The packet communications within each rack and to the

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

top-of-rack switch (TOR) are controlled by the FPGA via the high-speed, low latency multiplexer controlled connections that serve as the Catapult II Shell Components. *Id.*

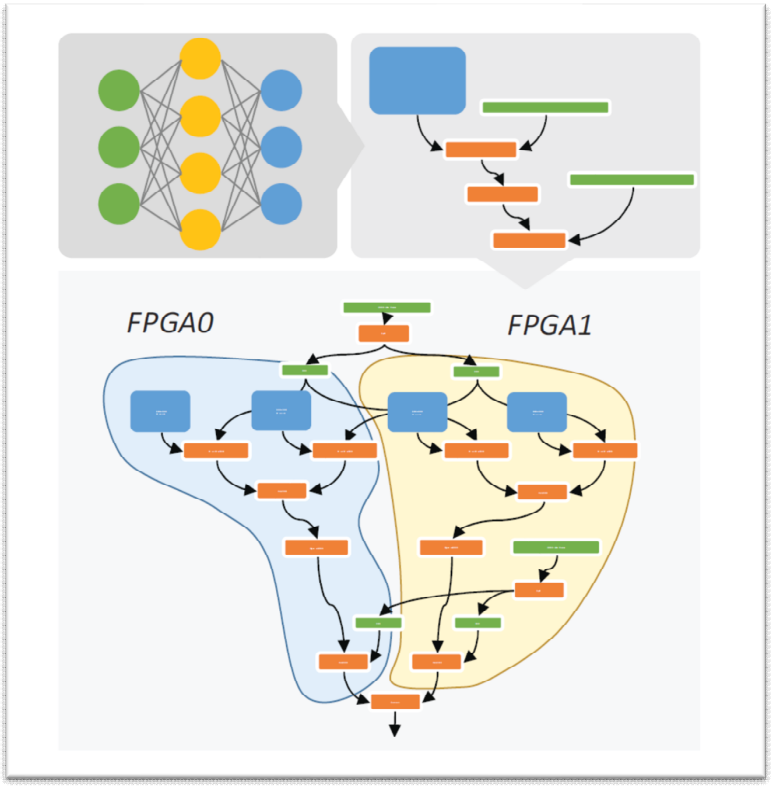


191. The hardware through which the FPGAs communicate constitutes a first set of inter-task communication paths connecting a first set of programmable logic-based processing units. The figure below shows the "shell" of the Catapult II FPGA. Exs. 12, 16. All of the depicted functionality is performed on the FPGA itself, which is essentially an array of multiplexers and wires that provide high-speed, low latency interconnectivity between logical units. *Id.*, *see also* Ex. 46 *passim*, Ex. 18 at 5.



LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

192. The FPGA cores for a given program flow can be spread among multiple FPGA stages. Ex. 20 at 10-12, Ex. 37 at 21-22. The group of FPGA cores that are performing the same task constitute a stage. *Id.* In the illustration below (an excerpt of Ex. 20, Fig. 3), the tasks by the DNN model are separated into three types of tasks, illustrated in blue, orange and green. *Id.* Each of the tasks is deployed on a FPGA processing core instantiated on one of two FPGA processing stages, labelled FPGA0 and FPGA1. *Id.* The blue cores operate collectively to perform a first task in a first stage and pass the output to another stage comprised of orange cores. *Id.* The green cores operate collectively to perform a second task in a second stage and pass the output to one of a set of stages comprised of orange cores. *Id.* The orange cores perform a third function and serve as the fourth, fifth and sixth stages, although those stages could be considered a single stage for certain purposes and in certain implementations. *Id.*



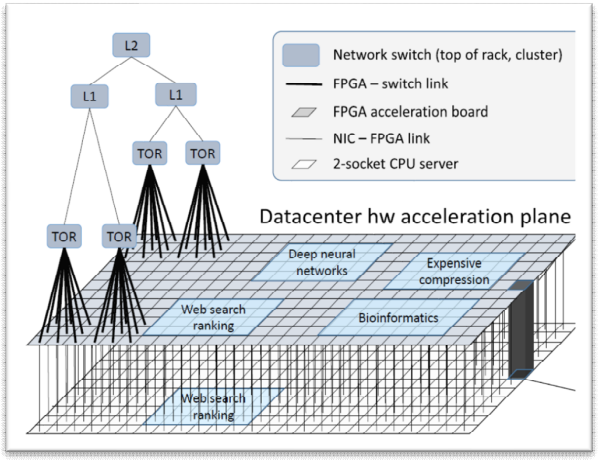
LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

forming a second set of inter-task communication paths connecting a second set of programmable logic-based processing units of the plurality of programmable logic-based processing units to a second multi-stage program instance, wherein the second multi-stage program instance is configured to accelerate the second program, and wherein the second set of programmable logic-based processing units are interconnected using reconfigurable cross-connects;

193. Microsoft uses Azure to form a second set of inter-task communication paths connecting a second set of programmable logic-based processing units of the plurality of programmable logic-based processing units into a first multi-stage program instance.

194. Azure includes multiple FPGA processors, which are programmable logic-based processing units. Microsoft connects a second set of FPGA processors to form a multi-stage program instance comprised of multiple FPGA cores. Every FPGA in an Azure datacenter is able to communicate with every other FPGA in a datacenter through a high-speed, low latency, multiplexer switched fabric. Exs. 12, 16, 36 *passim*.

195. Each Azure rack in a datacenter is connected by a two-tier switching network L1/L2 as shown below. Exs. 12, 16, *passim*. The packet communications within each rack and to the top-of-rack switch (TOR) are controlled by the FPGA via the high-speed, low latency multiplexer controlled connections that serve as the Catapult II Shell Components. *Id.*

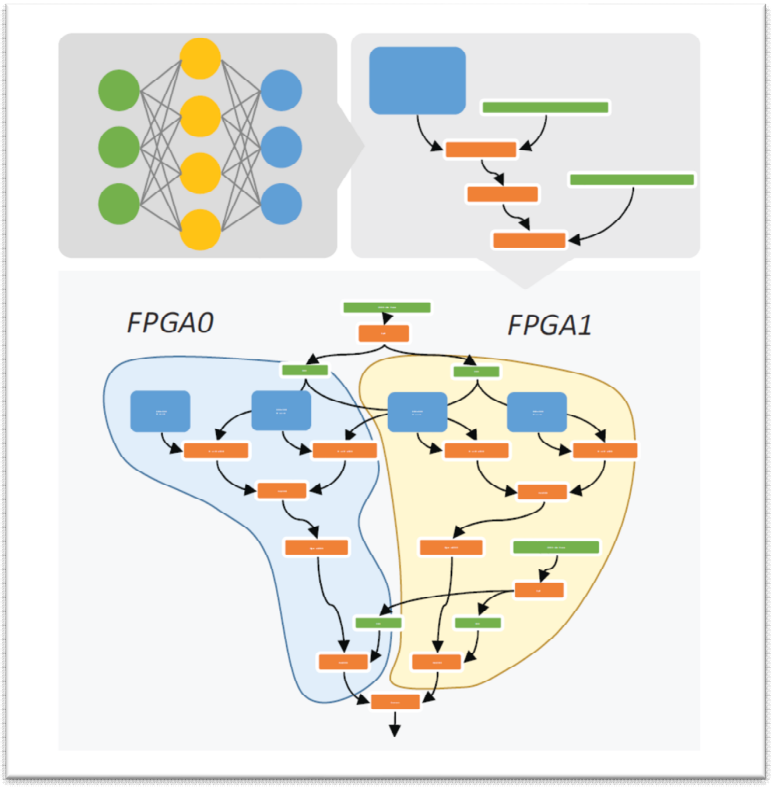


LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

196. The hardware through which the FPGAs communicate constitutes a first set of inter-task communication paths connecting a first set of programmable logic-based processing units.

197. The FPGA cores for a given program flow can be spread among multiple FPGA stages. Ex. 20 at 10-12, Ex. 37 at 21-22. The group of FPGA cores that are performing the same task constitute a stage. *Id.* In the illustration below (an excerpt of Ex. 20, Fig. 3), the tasks by the DNN model are separated into three types of tasks, illustrated in blue, orange and green. *Id.* Each of the tasks is deployed on a FPGA processing core instantiated on one of two FPGA processing stages, labelled FPGA0 and FPGA1. *Id.* The blue cores operate collectively to perform a first task in a first stage and pass the output to another stage comprised of orange cores. *Id.* The green cores operate collectively to perform a second task in a second stage and pass the output to one of a set of stages comprised of orange cores. *Id.* The orange cores perform a third function and serve as the fourth, fifth and sixth stages, although those stages could be considered a single stage for certain purposes and in certain implementations. *Id.*

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301



198. Azure simultaneously accelerates multiple multi-stage services or applications. At any given point in time, an Azure datacenter is accelerating multiple DNNs, RNNs, web search ranking, speech recognition, compression, computer vision, speech translation, and other programs. Many of these program or service instances are multi-stage.

in response to a first increased demand for the first multi-stage program instance determined by monitoring a first characteristic corresponding to the first set of programmable logic-based processing units, based on an adaptive optimized resource allocation policy, adding at least one more programmable logic-based processing unit to the first set of programmable logic-based processing units from the plurality of programmable logic-based processing units; and

199. In response to a first increased demand for the first multi-stage program instance determined by monitoring a first characteristic corresponding to the first set of programmable logic-based processing units, based on an adaptive optimized resource allocation policy, Azure

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

adds at least one more programmable logic-based processing unit to the first set of programmable logic-based processing units from the plurality of programmable logic-based processing units.

200. As Microsoft explained in a white paper discussing the Catapult II architecture, "[a]s demand for a service grows or shrinks, a global manager grows or shrinks the pools correspondingly." Ex. 12 at 2, see also Ex. 16 at 55. The pools of cores among and between different FPGAs assigned to each multi-stage program instance is controlled by the Catapult II Shell Components under the supervision and control of the Service Manager and Resource Manager. Ex. 12 at 10. As discussed above, load data, which is a first characteristic as claimed, is monitored and tracked at each FPGA core group. See Ex.41. That data is published to other FPGA core groups in the datacenter. Based on this monitoring, Azure expands and contracts the number of cores assigned to each group over time.

in response to a second increased demand for the second multi-stage program instance determined by monitoring a second characteristic different from the first characteristic, corresponding to the second set of programmable logic-based processing units, based on the adaptive optimized resource allocation policy, adding at least one more programmable logic-based processing unit to the second set of programmable logic-based processing units from the plurality of programmable logic-based processing units.

201. In response to a second increased demand for the second multi-stage program instance determined by monitoring a second characteristic different from the first characteristic, corresponding to the second set of programmable logic-based processing units, based on the adaptive optimized resource allocation policy, Azure adds at least one more programmable logic-based processing unit to the second set of programmable logic-based processing units from the plurality of programmable logic-based processing units.

202. Multiple instances of such multi-stage programs are executed in a similar manner as described above in connection with the first multi-stage program instance. That discussion is incorporated herein by reference.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

203. As described in Microsoft's own patent filings, "the load data [need] not [be] stored as a single discrete value. For example, the load data 414 may include a plurality of data types and values. Such as, for example, a number of queued requests for the target hardware acceleration device, a recent processing time of a previous request, an estimate based on the number and type of requests in the queue, a total size of the queued requests, a round trip time for receiving responses to requests, and any other suitable types of data that may indicate a load of the target hardware acceleration device." Ex.41, '472 Patent at 8:21-30. Each of the foregoing represents a second characteristic different than the first characteristic as claimed.

204. On information and belief, Azure uses a plurality of these different data types to reallocate cores among the multi-stage program instances in a given datacenter on any given day. Any two such multi-stage program instances being rebalanced or reallocated according to given load data types may constitute the first and second multi-stage program instances.

205. Upon information and belief, in accordance with 35 U.S.C. § 287, Microsoft has had actual notice and knowledge of the '948 patent no later than its issuance.

206. Microsoft continues without license to make, use, import, offer for sale, and/or sell in the United States services or products that infringe the '948 patent including specifically Microsoft Azure and its cloud computing functionalities.

207. Microsoft has directly infringed and continues to directly infringe the '948 patent by engaging in acts constituting infringement under 35 U.S.C. § 271(a) including but not necessarily limited to one or more of making, using, selling and offering to sell, in this District and elsewhere in the United States, and importing into the United States, the Microsoft Azure platform or components and services thereof.

208. Microsoft's continued infringement of the '948 patent is knowing, intentional, and

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

willful.

209. Microsoft has had knowledge of and notice of the chain of applications underlying the ’948 patent since at least May 2015, when Microsoft representatives acknowledged receipt of an email from ThroughPuter disclosing several of ThroughPuter’s U.S. patent applications and in any case no later than January 2018 when ThroughPuter brought ThroughPuter’s patent portfolio to the attention of Microsoft in the January 2018 letter, and despite this knowledge continues to commit the aforementioned infringing acts.

210. Microsoft has been actively monitoring ThroughPuter’s patent portfolio since at least the time it was served with the complaint in the Virginia Action and has gained actual knowledge of the patents asserted herein.

211. Microsoft’s continuing infringement of the ’948 patent is deliberate and willful and constitutes egregious misconduct. In developing and offering its products and services, Microsoft is, at the very least, willfully blind to this ongoing infringement.

212. Pursuant to 35 U.S.C. § 284, ThroughPuter is entitled to recover monetary damages for the injuries arising from Microsoft’s willful infringement in an amount to be determined at trial. Microsoft’s infringement of the ’948 patent has caused irreparable harm to ThroughPuter and will continue to cause such harm unless and until Microsoft’s infringing activities are enjoined by this Court.

213. Microsoft’s infringement of the ’948 patent is exceptional and entitles ThroughPuter to attorneys’ fees and costs incurred in prosecuting this action under 35 U.S.C. § 285.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

COUNT 2

(Infringement of U.S. Patent No. 11,036,556)

214. ThroughPuter repeats and realleges each and every allegation contained above as though fully set forth herein.

215. On June 15, 2021, the USPTO duly and legally issued the ’556 patent, entitled “Concurrent Program Execution Optimization.” A copy of the ’556 patent is attached as Exhibit 2.

216. Mark Sandstrom is the sole and true inventor of the ’556 patent.

217. ThroughPuter, Inc. owns all right, title and interest to and in the ’556 patent.

218. Microsoft infringes claims 1-8 of the ’556 patent.

219. As demonstrated in side-by-side comparison, claim 1 of ThroughPuter’s ’556 patent closely matches claim 17 of Microsoft’s ’392 patent. Ex. 47. The column on the left shows claim 1 of the ’556 patent, which claims priority to applications dating back to 2012. The column on the right shows independent claim 17 of Microsoft’s ’392 patent, which claims priority to an application filed in 2015. As can be appreciated from this side-by-side comparison, Microsoft obtained a patent on substantially the same technology taught by ThroughPuter’s patent applications. However, ThroughPuter’s ’556 patent is entitled to a priority date that is more than two years earlier than Microsoft’s.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

ThroughPuter's U.S. Patent No. 11,556,556, Independent Claim 1	Microsoft's U.S. Patent No. 10,296,392, Independent Claim 17
A method performed in a data processing system, the method comprising:	A method performed in a data processing system, the method comprising:
receiving, by hardware logic and/or software logic, requests to perform different tasks on behalf of instances of a plurality of programs managed by a data processing system;	receiving requests to perform different functions on behalf of tenant functionality that executes on requesting server unit components of the data processing system;
identifying, by the hardware logic and/or software logic for each of the instances, communication interdependencies between different processing stages of a set of processing stages of the respective instance;	parsing the different functions to identify programmatic calls between different stages of the different functions;
based on conditions in the data processing system, dynamically varying, by the hardware logic and/or software logic, structures of field-programmable gate arrays used to process different tasks of the instances of the plurality of programs, the structures being dynamically varied by	based at least on conditions in the data processing system, dynamically varying structures of field-programmable gate arrays used to implement different invocations of the different functions, the structures being dynamically varied by:
identifying available field-programmable gate arrays of the data processing system that are available to process different processing stages of requesting instances of respective programs,	identifying available field-programmable arrays of the data processing system that are available to implement the different stages of the different functions;
based at least on the conditions in the data processing system, identifying selected field-programmable gate arrays from the available field-programmable gate arrays to execute the different processing stages of the requesting instances of the respective programs,	based at least on the conditions in the data processing system, identifying selected field-programmable gate arrays from the available field-programmable gate arrays to perform individual stages of the different functions;
configuring the selected field-programmable gate arrays to process a respective processing stage of a respective requesting instance, and	configuring the selected field-programmable gate arrays of the plurality to perform the individual stages of the different functions; and
configuring certain selected field-programmable gate arrays to support communicating, by the task executing on the respective field-programmable gate array, final results to a requesting client over a network in the data processing system.	configuring certain selected field-programmable arrays to communicate final results of the different functions to the requesting server unit components, the certain selected field-programmable arrays communicating the final results to the requesting server unit components over a network in the data processing system.

220. On information and belief, Microsoft believed at the time of filing and still believes that the subject matter claimed in the '392 patent is patent eligible under 35 U.S.C. § 101.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

221. On information and belief, Microsoft believed at the time of filing and still believes that the subject matter claimed in the '392 patent is novel under 35 U.S.C. § 102 in view of the references disclosed to and considered by the USPTO during examination of the underlying application.

222. On information and belief, Microsoft believed at the time of filing and still believes that that the subject matter claimed in the '392 patent is non-obvious under 35 U.S.C. § 103 in view of the references Microsoft disclosed to and considered by the USPTO during examination of the underlying application.

223. Claim 1 of the '556 patent is representative of the claims infringed by Microsoft and recites:

> 1. A method performed in a data processing system, the method comprising:
>
> receiving, by hardware logic and/or software logic, requests to perform different tasks on behalf of instances of a plurality of programs managed by a data processing system;
>
> identifying, by the hardware logic and/or software logic for each of the instances, communication interdependencies between different processing stages of a set of processing stages of the respective instance;
>
> based on conditions in the data processing system, dynamically varying, by the hardware logic and/or software logic, structures of field-programmable gate arrays used to process different tasks of the instances of the plurality of programs, the structures being dynamically varied by
>
> identifying available field-programmable gate arrays of the data processing system that are available to process different processing stages of requesting instances of respective programs,
>
> based at least on the conditions in the data processing system, identifying selected field-programmable gate arrays from the available field-programmable gate arrays to execute the different processing stages of the requesting instances of the respective programs,
>
> configuring the selected field-programmable gate arrays to process a respective processing stage of a respective requesting instance, and

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

configuring certain selected field-programmable gate arrays to support communicating, by the task executing on the respective field-programmable gate array, final results to a requesting client over a network in the data processing system.

224. On information and belief, Azure is implemented in a manner that meets each and every limitation of claim 1 of the '556 patent.

225. On information and belief, Azure practices the invention recited in claim 17 of Microsoft's '392 patent (Ex. 47).

1. A method performed in a data processing system, the method comprising:

226. Azure is a data processing system.

receiving, by hardware logic and/or software logic, requests to perform different tasks on behalf of instances of a plurality of programs managed by a data processing system;

227. Azure receives by hardware logic/and or software logic, requests to perform different tasks on behalf of instances of a plurality of programs managed by a data processing system.

228. Azure, *e.g.*, via Catapult II Shell Components instantiated on FPGAs, receives requests to accelerate services from at least the CPUs by hardware logic and/or software logic that are positioned "behind" the FPGAs in the Catapult II network. Ex. 17 at 54. On information and belief, in Azure's current form a hardware microservice running on an FPGA may also initiate a request for acceleration.

identifying, by the hardware logic and/or software logic for each of the instances, communication interdependencies between different processing stages of a set of processing stages of the respective instance;

229. Azure identifies, by the hardware logic and/or software logic for each of the instances, communication interdependencies between processing stages of a set of processing stages of the respective instance.

230. In Azure, under the supervision of the RMs, the SM components instantiated on the

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

CPUs and/or FPGA shells determine how to break a requested service down into steps or sub-task components. Ex. 35 at 119, Ex. 20 at 11. Those sub-tasks are assigned to various processing cores on the FPGA fabric. As part of this process, the SM components determine what each sub-task will provide as inputs and outputs, which involves identification of communication interdependencies between the stages (*e.g.*, which stage needs to communicate with which other stage).

based on conditions in the data processing system, dynamically varying, by the hardware logic and/or software logic, structures of field-programmable gate arrays used to process different tasks of the instances of the plurality of programs,

231. Based on conditions in the data processing system, Azure dynamically varies, by the hardware logic and/or software logic, structures of field-programmable gate arrays used to process different tasks of the instances of the plurality of programs.

232. Azure's AC, RM, SMs and FMs works in conjunction to assign tasks of a multi-stage program instance to a group of reconfigurable FPGA processors. When a core is reallocated to perform a different task (for instance speech recognition instead of compression), the FPGA circuit is reconfigured accordingly by loading a corresponding image file or role. Ex. 12 at 7, Ex. 35 at 116. Such reconfiguration varies the structure of the logic blocks of the FPGA.

the structures being dynamically varied by identifying available field-programmable gate arrays of the data processing system that are available to process different processing stages of requesting instances of respective programs,

233. In Azure, the process of dynamically varying structures of available field-programmable gate arrays of the data processing system, includes identifying FPFAs that are available to process different processing stages of requesting instances of respective programs.

234. As discussed above in connection with previous counts and/or in the section of this complaint entitled Microsoft's Infringing Cloud Computing Architecture, Azure, *e.g.*, via the Catapult II system, monitors load data of various types and reallocates FPGA processing cores

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

accordingly. This reallocation includes identifying available FPGAs. For instance, unused or available cores are allocated to "join" groups that are experiencing particularly high demand.

based at least on the conditions in the data processing system, identifying selected field-programmable gate arrays from the available field-programmable gate arrays to execute the different processing stages of the requesting instances of the respective programs,

235. Azure, based at least on the conditions in the data processing system, identifies selected field-programmable gate arrays from the available field-programmable gate arrays to execute the different processing stages of the requesting instances of the respective programs.

236. As discussed above, after such a reallocation involving a multi-stage program instance (*e.g.*, an instance of a DNN service), the SM(s) in cooperation with the FMs determine which sub-tasks will be executed on which cores of which newly allocated FPGA blocks. These assignments are made based in part on the load data reported concerning processing tasks placed on each FPGA or block thereof, which data comprises the conditions in the data processing system.

configuring the selected field-programmable gate arrays to process a respective processing stage of a respective requesting instance, and

237. Azure configures the selected field-programmable gate arrays to process a respective processing stage of a respective requesting instance.

238. As discussed above, when cores are allocated for execution of a new function that requires execution of a new function, FPGA core blocks may be re-imaged as needed. That re-imaging may be partial (only some cores on the FPGA) or complete (all processing cores on the FPGA). Typically, the Catapult II Shell Components are not re-imaged during this process. Such reimaging constitutes configuring the FPGA gate arrays as claimed.

configuring certain selected field-programmable gate arrays to support communicating, by the task executing on the respective field-programmable gate array, final results to a requesting client over a network in the data processing system.

239. Azure configures selected field-programmable gate arrays to support

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

communicating, by the task executing on the respective field-programmable gate array, final results to a requesting client over a network in the data processing system.

240. The head component of each group of cores (each instantiated on an FGPA) is configured under the supervision of the RM and the SM components to communicate the final result of the acceleration request to the requesting client, hosted, *e.g.*, on the CPU which is local to the FPGA launching the acceleration request. On information and belief, in its current form Azure, *e.g.*, via Catapult II, permits requesting clients to reside on FPGAs (*e.g.*, as hardware accelerated microservices). Azure also allows external CPUs to communicate with FPGAs.

241. Upon information and belief, in accordance with 35 U.S.C. § 287, Microsoft has had actual notice and knowledge of the '556 patent no later than its issuance.

242. Microsoft has been actively monitoring ThroughPuter's patent portfolio since at least the time it was served with the complaint in the Virginia Action and has gained actual knowledge of the patents asserted herein.

243. Microsoft continues, without license, to make, use, offer for sale, import and/or sell in the United States services or products that infringe the '556 patent including specifically Microsoft Azure and its cloud computing functionalities.

244. Microsoft has directly infringed and continues to directly infringe the '556 patent by engaging in acts constituting patent infringement under 35 U.S.C. § 271(a) including but not necessarily limited to one or more of making, using, selling and offering to sell, in this District and elsewhere in the United States, and importing into the United States, the Microsoft Azure platform or components and services thereof.

245. Microsoft's continuing infringement of the '556 patent is knowing, intentional, and willful.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

246. Microsoft's infringement of the '556 patent has injured ThroughPuter in its business and property rights.

247. Microsoft's continued infringement of the '556 patent has been and is deliberate and willful and constitutes egregious misconduct. In developing and offering its products and services, Microsoft is, at the very least, willfully blind to this ongoing infringement.

248. Pursuant to 35 U.S.C. § 284, ThroughPuter is entitled to recover monetary damages for the injuries arising from Microsoft's willful infringement in an amount to be determined at trial. Microsoft's infringement of the '556 patent has caused irreparable harm to ThroughPuter and will continue to cause such harm unless and until Microsoft's infringing activities are enjoined by this Court.

249. Microsoft's infringement of the '556 patent is exceptional and entitles ThroughPuter to attorneys' fees and costs incurred in prosecuting this action under 35 U.S.C. § 285.

COUNT 3

(Infringement of U.S. Patent No. 10,963,306)

250. ThroughPuter repeats and realleges each and every allegation contained above as though fully set forth herein.

251. On March 30, 2021, the USPTO duly and legally issued the '306 patent, entitled "Managing Resource Sharing in a Multi-Core Data Processing Fabric." A copy of the '306 patent is attached as Exhibit 3.

252. Mark Sandstrom is the sole and true inventor of the '306 patent.

253. ThroughPuter, Inc. owns all right, title and interest to and in the '306 patent.

254. Microsoft infringes at least claims 1 and 4-7 of the '306 patent.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

255. As demonstrated below in a side-by-side comparison, claim 1 of ThroughPuter's '306 patent closely matches the claims of Microsoft's '709 patent. In other words, Microsoft later attempted to patent, and did obtain a patent on, the technology described by ThroughPuter's '306 patent disclosures, which were filed at least three years before the Microsoft '709 patent disclosure.

256. Claim 1 of the '306 patent is representative of the claims infringed by Microsoft and recites:

> 1. A method for task-switching on a multi-user parallel processing hardware architecture comprising a plurality of reconfigurable logic-based processing units, the method comprising:
>
> linking, through a first set of inter-task communication paths of the multi-user parallel processing hardware architecture, a first set of reconfigurable logic-based processing units of the plurality of reconfigurable logic-based processing units into a first multi-stage program instance configured to execute a corresponding set of interdependent processing stages of a first program, wherein each of the first set of reconfigurable logic-based processing units is programmed to perform a respective task of a plurality of first tasks of the first program corresponding to a respective processing stage of the first multi-stage program instance;
>
> linking, through a second set of inter-task communication paths of the multi-user parallel processing hardware architecture, a second set of reconfigurable logic-based processing units of the plurality of reconfigurable logic-based processing units into a second multi-stage program instance configured to execute a corresponding set of interdependent processing stages of a second program, wherein each of the second set of reconfigurable logic-based processing units is programmed to perform a respective task of a plurality of second tasks of the second program corresponding to a respective processing stage of the second multi-stage program instance;
>
> maintaining, in a storage, a first location for a first reconfigurable logic-based processing unit of the first set of reconfigurable logic-based processing units executing the first multi-stage program instance such that a first one or more users and/or programs are enabled to communicate directly with the first multi-stage program instance;
>
> maintaining, in the storage, a second location for a second reconfigurable logic-based processing unit of the second set of reconfigurable logic-based processing units executing the second multi-stage program instance such that a second one or more users and/or programs are enabled to communicate directly with the second multi-stage program instance; and

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

in response to an increased demand for the second program, reallocating, by a controller comprising software and/or hardware logic configured to implement a load-adaptive allocation policy, at least one processing unit of the first set of reconfigurable logic-based processing units, the reallocating resulting in

(1) switching the at least one of the first set of processing units from performing a task of the plurality of first tasks to performing one task of the plurality of second tasks, wherein switching comprises matching a first programming configuration of the at least one of the first set of processing units to a programming configuration demanded by the one task, and

(2) adjusting, in storage, at least one of the first location or the second location to enable, through the respective location, direct communication to the other multi-stage program instance of the first multi-stage program instance or the second multi-stage program instance;

wherein the load-adaptive allocation policy is configured to facilitate minimizing reconfiguring the plurality of reconfigurable logic-based processing units.

257. On information and belief, Azure is implemented in a manner that meets each and every limitation of claim 1 of the ’306 patent.

1. A method for task-switching on a multi-user parallel processing hardware architecture comprising a plurality of reconfigurable logic-based processing units, the method comprising:

258. The Microsoft Azure cloud platform manages execution of a plurality of software application tasks on an array of reconfigurable logic-based processing units shared amongst multiple client programs. Azure is configured to, and Microsoft uses Azure to, switch application tasks across multiple processors within a group of processors.

259. At least the Azure Controller (AC) in conjunction with Resource Manager (RM), Service Managers (SM), and FPGA Managers (FM), is used to manage the assignment of computing cores across multiple applications and multiple application tasks running, for instance, on multiple computing cores such as CPUs, GPUs and/or FPGA processors.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

linking, through a first set of inter-task communication paths of the multi-user parallel processing hardware architecture, a first set of reconfigurable logic-based processing units of the plurality of reconfigurable logic-based processing units into a first multi-stage program instance configured to execute a corresponding set of interdependent processing stages of a first program, wherein each of the first set of reconfigurable logic-based processing units is programmed to perform a respective task of a plurality of first tasks of the first program corresponding to a respective processing stage of the first multi-stage program instance;

260. Azure links, through a first set of inter-task communication paths of the multi-user parallel processing architecture, a first set of reconfigurable logic-based processing units of the plurality of reconfigurable logic-based processing units into a first multi-stage program instance configured to execute a corresponding set of interdependent processing stages of a first program. In Azure, each of the first set of reconfigurable logic-based processing units is programmed to perform a respective task of a plurality of first tasks of the first program corresponding to a respective processing stage of the first multi-stage program instance.

261. As described in Microsoft's '709 patent, the reconfigurable processing units (acceleration components) are dynamically connected through hardware based communication paths that include a dedicated three port switch that works in coordination with the bypass control, multiplexers (connected to FIFO buffers 714, 716), the TOR interface, and TOR switch.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

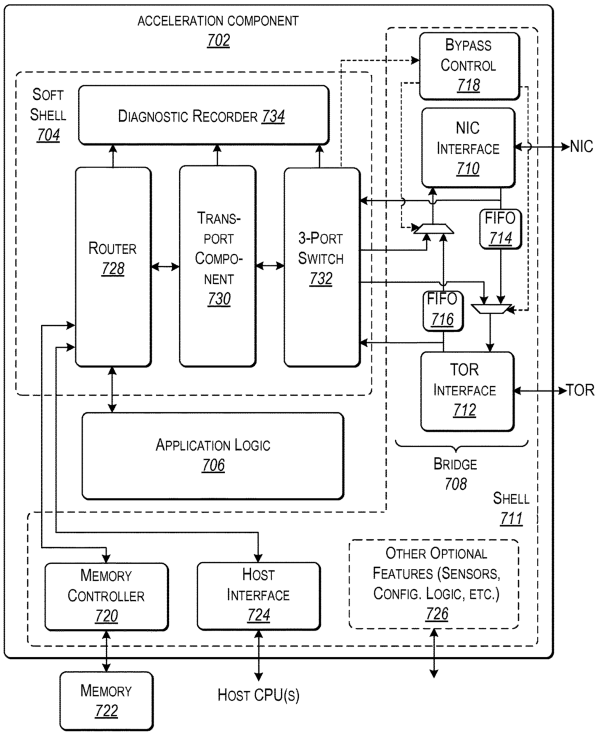


FIG. 7

Ex. 13, Fig. 7.

262. Azure's SM, in coordination with the RM and FM, assigns groups of reconfigurable processing units such that some of the processing groups collectively execute a multi-stage program instance (*e.g.*, linking a set of processors to a multi-stage program instance) and individual processing units of an FPGA Grouping perform a single task associated with a stage of the multi-stage program instance. As illustrated by Fig. 5 of Microsoft's '709 patent, individual reconfigurable processing units (acceleration components H) perform their respective tasks corresponding to one of the stages and then communicates the results to the next processing unit H, before the service or "graph" comprising said accelerators H outputs the result to the calling processing unit S.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

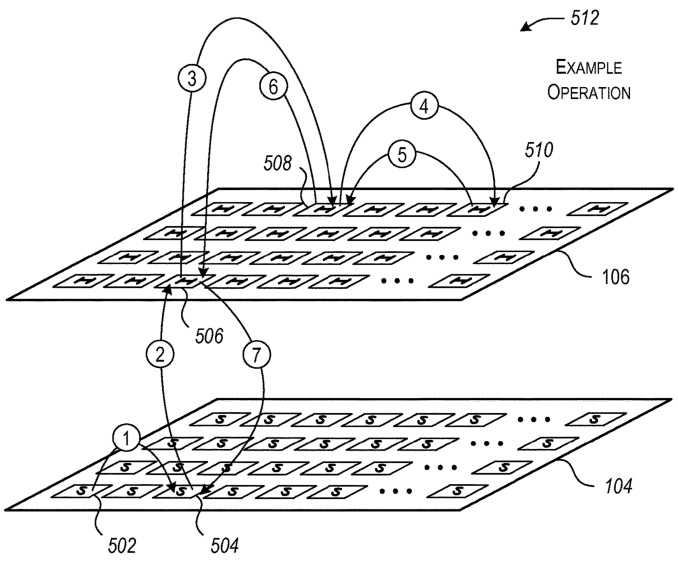


FIG. 5

Ex. 13, Fig. 5.

linking, through a second set of inter-task communication paths of the multi-user parallel processing hardware architecture, a second set of reconfigurable logic-based processing units of the plurality of reconfigurable logic-based processing units into a second multi-stage program instance configured to execute a corresponding set of interdependent processing stages of a second program, wherein each of the second set of reconfigurable logic-based processing units is programmed to perform a respective task of a plurality of second tasks of the second program corresponding to a respective processing stage of the second multi-stage program instance;

263. Azure links through a second set of inter-task communication paths of the multi-user parallel processing hardware architecture, a second set of reconfigurable logic-based processing units of the plurality of reconfigurable logic-based processing units into a second multi-stage program instance configured to execute a corresponding set of interdependent processing stages of a second program. In Azure, each of the second set of reconfigurable logic-based processing units is programmed to perform a respective task of a plurality of second tasks of the second program corresponding to a respective processing stage of the second multi-stage program

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

instance.

264. Azure's AC works in conjunction with the RM, SMs, and FM to repeat the basic process described immediately above to assign a set of tasks of a second multi-stage program instance to a second group of reconfigurable processors. This is illustrated in Microsoft's '709 patent, which shows that different groups of reconfigurable processors H execute different tasks of a different multi-stage program instance during a given time period, shown in different colors in the annotated version of Fig. 15C below. During a subsequent time period, the arrangement of processors H has been reconfigured such that different sets of processors are performing the respective tasks, shown in different colors in the annotated version of Fig. 15D below.

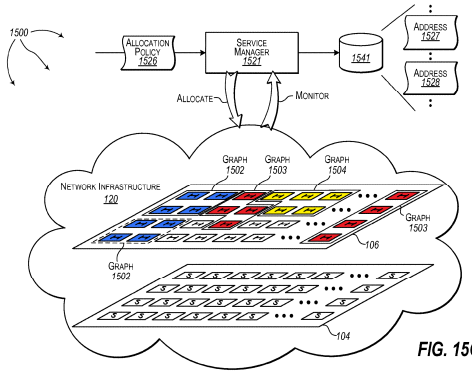


FIG. 15C

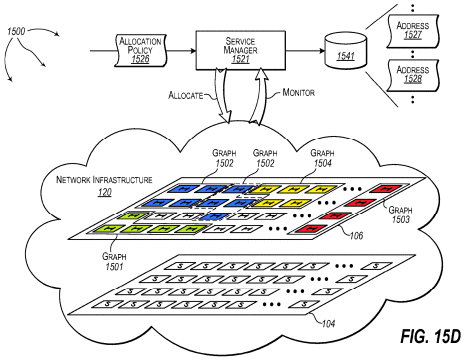


FIG. 15D

265. The inter-task communication pathways between processing units vary between FPGA groups as well as over allocation time intervals, including because those groups (for instance graph 1503 and 1504) include different processing units H and because the same processing group (for instance graph 1502) includes varying processing units over time.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

maintaining, in a storage, a first location for a first reconfigurable logic-based processing unit of the first set of reconfigurable logic-based processing units executing the first multi-stage program instance such that a first one or more users and/or programs are enabled to communicate directly with the first multi-stage program instance

266. Azure maintains, in a storage, a first location for a first reconfigurable logic-based processing unit of the first set of reconfigurable logic-based processing units executing the first multi-stage program instance such that a first one or more users and/or programs are enabled to communicate directly with the first multi-stage program instance.

267. Azure's RM and AC work in conjunction with at least the SMs to keep track of the addresses of each separately addressable FPGA processor group and maintain them in a storage that is accessible by the AC. According to Microsoft's '709 patent and as illustrated in Figs. 15C and 15D annotated below, the service manager 1521 (which corresponds to the RM) stores addresses 1527-1528 for graphs 1502-1504 in storage 1541.

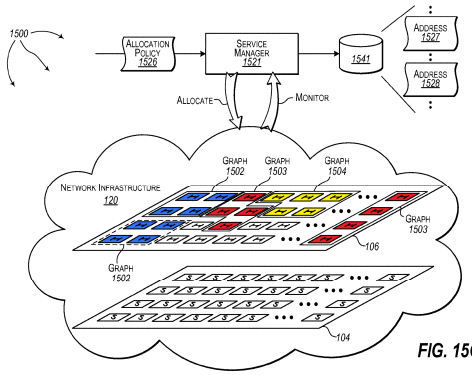


FIG. 15C

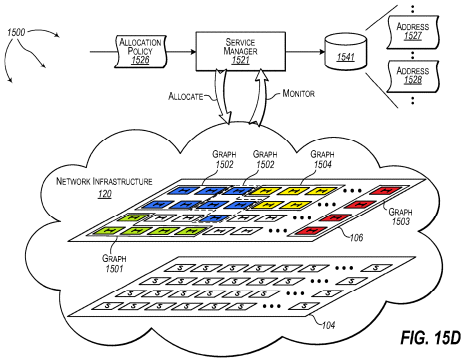


FIG. 15D

268. In Azure, client programs can communicate with the FPGA processor groups directly, which allow the client programs to communicate with the multi-stage program instance, including by contacting the RM and requesting that data be sent to a FPGA processor group and by receiving data output from a FPGA processor group. According to Microsoft's '709 patent,

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

"[a]ddresses 1527, 1528, etc. stored in storage 1541 can be used to direct requests for acceleration to the appropriate graphs." Ex. 13 at 19:57-59.

maintaining, in the storage, a second location for a second reconfigurable logic-based processing unit of the second set of reconfigurable logic-based processing units executing the second multi-stage program instance such that a second one or more users and/or programs are enabled to communicate directly with the second multi-stage program instance; and

269. Azure maintains in the storage, a second location for a second reconfigurable logic-based processing unit of the second set of reconfigurable logic-based processing units executing the second multi-stage program instance such that a second one or more users and/or programs are enabled to communicate directly with the second multi-stage program instance.

270. In Azure, the aforementioned processor group address storage and access functionality applies to and exists for each of the multiple FPGA processor groups or graphs.

in response to an increased demand for the second program, reallocating, by a controller comprising software and/or hardware logic configured to implement a load-adaptive allocation policy, at least one processing unit of the first set of reconfigurable logic-based processing units, the reallocating resulting in

271. In response to an increased demand for the second program, Azure reallocates by a controller comprising software and/or hardware logic configured to implement a load-adaptive allocation policy, at least one processing unit of the first set of reconfigurable logic-based processing units.

272. Azure's RM and AC work in conjunction with at least the SMs to reallocate processors among the processor groups. According to Microsoft's '709 patent and as illustrated in Figs. 15C and 15D annotated below, the service manager 1521 coordinates the reallocation of the processors in graph 1503 to graph 1502.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

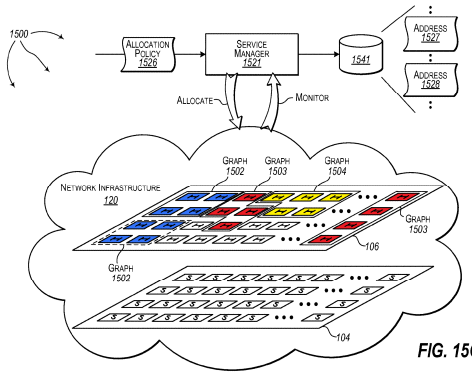


FIG. 15C

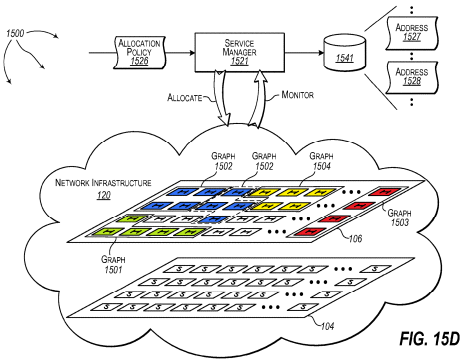


FIG. 15D

273. In Azure, this is done in response to a change in demand expressed by the application programs and/or their respective tasks, for instance by the accumulation of data at the input buffer for an application program and/or task. According to Microsoft's '709 patent, "[a]cceleration components [*i.e.*, reconfigurable processing units H] can be allocated in a manner that balances load in a hardware acceleration plane, minimizes role switching, and adapts to demand changes." Ex. 13 at Abstract. The '709 patent further explains that "[d]uring continued monitoring, service manager 1521 can detect an increased demand for graph 1502. Based on the current allocation of acceleration components in hardware plane 106 and in view of allocation policy 1526, service manager 1521 can (re)allocate acceleration components to meet the increased demand for graph 1502." Ex. 13 at 19:60-65.

1) switching the at least one of the first set of processing units from performing a task of the plurality of first tasks to performing one task of the plurality of second tasks, wherein switching comprises matching a first programming configuration of the at least one of the first set of processing units to a programming configuration demanded by the one task, and,

274. Azure switches the at least one of the first set of processing units from performing a task of the plurality of first tasks to performing one task of the plurality of second tasks, wherein switching comprises matching a first programming configuration of the at least one of the first set

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

of processing units to a programming configuration demanded by the one task.

275. Azure's RM and AC work in conjunction with at least the SMs to reallocate and accelerate processors among the accelerator groups such that the need to reconfigure the accelerator logic blocks is minimized. For example, consider the situation in which a first calling application (*e.g.*, encryption) and a second calling application (*e.g.*, data compression) both preferentially request or require the same type of math function accelerator processor. In Azure, when the data compression application expresses increased demand for processing units relative to the encryption application, the RM and AC work in connection with the SMs to ensure that an accelerator logic block (from the encryption service) that is already configured to function as the math function accelerator is added to the data compression group instead of, *e.g.*, a logic block presently configured to function as a search ranking accelerator. This reduces the need to reconfigure the accelerator logic blocks, which would take additional time and impede overall efficiency.

276. This functionality is described in Microsoft's '709 patent: "Configuration data can [be] used to switch between roles (or otherwise change behavior) at an acceleration component using the same underlying image [*i.e.*, processor configuration]. As such, roles (or other behavior) can be changed without having to load a new image file." Ex. 13 at 20:26-30.

2) adjusting, in storage, at least one of the first location or the second location to enable, through the respective location, direct communication to the other multi-stage program instance of the first multi-stage program instance or the second multi-stage program instance,

277. Microsoft adjusts in storage, at least one of the first location or the second location to enable, through the respective location, direct communication to the other multi-stage program instance of the first multi-stage program instance or the second multi-stage program instance.

278. As discussed above, Azure's RM and AC work in conjunction with at least the SMs

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

to keep track of the addresses of each separately addressable processing group and record them in a storage that is accessible by the AC. As processing groups are modified in the manner depicted in Figs. 15C and 15D of Microsoft's '709 patent (annotated below), their corresponding addresses 1527-28 are updated in storage 1541.

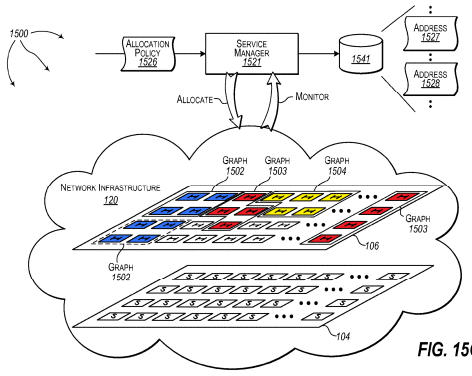


FIG. 15C

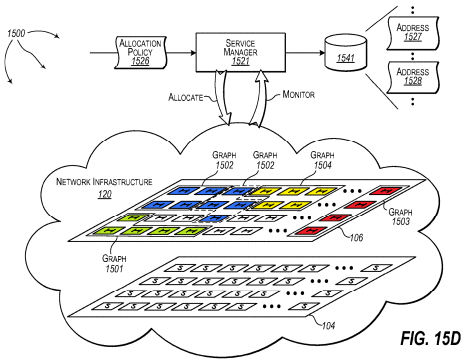


FIG. 15D

279. According to Microsoft's '709 patent, "[t]he service manager maintains [*i.e.*, keeps updated] an address for the graph so that the service can request hardware acceleration from the group of interoperating acceleration components." Ex. 13 at 1:44-47.

280. As discussed above, and as illustrated by Fig. 5 of Microsoft's '709 patent, in Azure each reconfigurable processing unit (acceleration component H) performs a task corresponding to one of the stages and then directly communicates the results to the next processing unit H, before the service or "graph" comprising said accelerators H outputs the result to the calling processing unit S.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

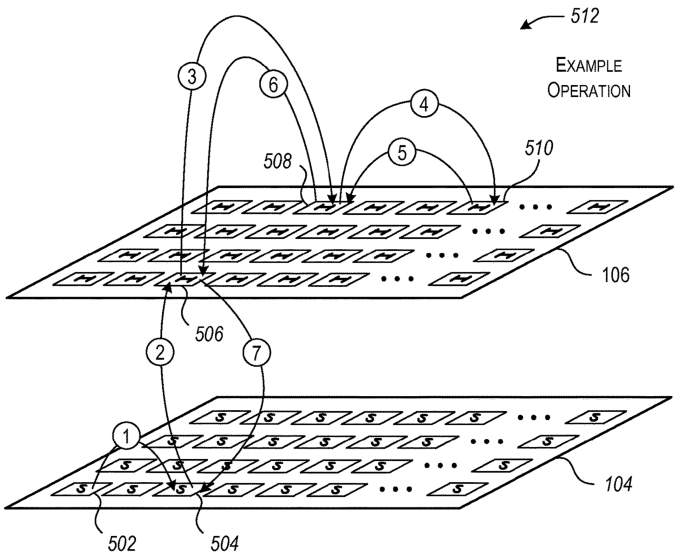


FIG. 5

wherein the load-adaptive allocation policy is configured to facilitate minimizing reconfiguring the plurality of reconfigurable logic-based processing units.

281. In Azure, the load-adaptive allocation policy is configured to facilitate minimizing reconfiguring the plurality of reconfigurable logic-based processing units.

282. As discussed above, Azure's RM and AC work in conjunction with at least the SMs to reallocate and reassign accelerator processors among the accelerator groups such that the need to reconfigure the accelerator logic blocks is minimized. For example, consider the situation in which a first calling application (*e.g.*, encryption) and a second calling application (*e.g.*, data compression) both preferentially request or require the same type of math function accelerator processor. In Azure, when the data compression application expresses increased demand for processing units relative to the encryption application, the RM and AC work in connection with the SMs to ensure that an accelerator logic block (from the encryption service) that is already configured to function as the demanded math accelerator is added to the data compression group

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

instead of, *e.g.*, a logic block presently configured to function as a search ranking accelerator. This reduces the need to reconfigure the accelerator logic blocks, which would take additional time and impede overall efficiency.

283. This functionality is described in Microsoft's '709 patent: "Configuration data can [be] used to switch between roles (or otherwise change behavior) at an acceleration component using the same underlying image [*i.e.*, processor configuration]. As such, roles (or other behavior) can be changed without having to load a new image file." Ex. 13 at 20:26-30. The '709 patent further explains that "[a] service manager can minimize reconfiguration as much as possible when allocating a group of interoperating acceleration components to accelerate a service." Ex. 13 at 7:14-17.

284. Upon information and belief, in accordance with 35 U.S.C. § 287, Microsoft has had actual notice and knowledge of the '306 patent no later than its issuance.

285. Microsoft continues, without license, to make, use, offer for sale, import and/or sell in the United States services or products that infringe the '306 patent including specifically Microsoft Azure and its cloud computing functionalities.

286. Microsoft has directly infringed and continues to directly infringe the '306 patent by engaging in acts constituting patent infringement under 35 U.S.C. § 271(a) including but not necessarily limited to one or more of making, using, selling and offering to sell, in this District and elsewhere in the United States, and importing into the United States, the Microsoft Azure platform or components and services thereof.

287. Microsoft's continuing infringement of the '306 patent is knowing, intentional, and willful.

288. Microsoft's infringement of the '306 patent has injured ThroughPuter in its

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

business and property rights.

289. Microsoft's continuing infringement of the '306 patent is deliberate and willful and constitutes egregious misconduct. Despite actual knowledge of the '306 patent, Microsoft continues to develop and offer its infringing products and services. In developing and offering its products and services, Microsoft is, at the very least, willfully blind to this ongoing infringement.

290. Pursuant to 35 U.S.C. § 284, ThroughPuter is entitled to recover monetary damages for the injuries arising from Microsoft's willful infringement in an amount to be determined at trial. Microsoft's infringement of the '306 patent has caused irreparable harm to ThroughPuter and will continue to cause such harm unless and until Microsoft's infringing activities are enjoined by this Court.

291. Microsoft's infringement of the '306 patent is exceptional and entitles ThroughPuter to attorneys' fees and costs incurred in prosecuting this action under 35 U.S.C. § 285.

COUNT 4

(Infringement of U.S. Patent No. 10,620,998)

292. ThroughPuter repeats and realleges each and every allegation contained above as though fully set forth herein.

293. On April 14, 2020, the USPTO duly and legally issued U.S. Patent No. 10,620,998 ("the '998 patent"), entitled "Task Switching and Inter-Task Communications for Coordination of Applications Executing on a Multi-User Parallel Processing Architecture." A copy of the '998 patent is attached as Exhibit 4.

294. Mark Sandstrom is the sole and true inventor of the '998 patent.

295. ThroughPuter, Inc. owns all right, title and interest to and in the '998 patent.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

296. Microsoft infringes at least claims 1-7 and 9-21 of the ’998 patent.

297. Claim 1 of the ’998 patent is representative of the claims infringed by Microsoft and recites:

> 1. A method for managing execution of a plurality of software applications on an array of processing units, the method comprising:
>
> providing, for each software application of the plurality of software applications, one or more input buffers of a plurality of input buffers, each input buffer being provided for buffering processing load input directed to a respective software application of the plurality of software applications and being dedicated to the respective software application; and
>
> repeatedly rearranging, by a controller comprising hardware logic and/or software logic, task assignment to the array of processing units and communication path connectivity for the array of processing units, wherein the array of processing units comprises a plurality of processing units of a reconfigurable type, each being configurable in two or more application specific configurations, and rearranging comprises, for each iteration of a plurality of iterations,
>
> allocating, to each software application of at least a portion of a plurality of software applications as a set of active software applications, a number of units of the array of processing units at least in part in accordance with a plurality of demand expressions, each demand expression of the plurality of demand expressions corresponding to a different software application of the plurality of software applications, wherein each demand expression of the plurality of demand expressions is based at least in part on a measure of an amount of a processing load input at the one or more input buffers of the respective software application,
>
> assigning, for each application of the set of active software applications, one or more task instances of the respective active application to one or more processing units of the plurality of processing units in accordance with the allocating, wherein
>
> at least one task instance of the one or more task instances is a ready-to-execute task instance of the respective software application for processing the data at the one or more input data buffers of the respective software application,
>
> each task instance of the one or more task instances is assigned to a different processing unit of the one or more processing units such that assigning results in each processing unit of the plurality of processing units being assigned only one respective task instance for

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

any given iteration of the plurality of iterations, and

assigning comprises, for at least one task instance of the one or more task instances of a given software application of the plurality of software applications, reconfiguring the respective processing unit to an application specific configuration of the two or more application specific configurations, wherein the application specific configuration is associated with the at least one task instance, and

causing connection, in accordance with the assigning, of the processing load input from each buffer of at least a portion of the plurality of input buffers to a different unit of the plurality of processing units to connect respective processing load input to the respective software application that the respective processing load input is directed to.

298. On information and belief, Azure is implemented in a manner that meets each and every limitation of claim 1 of the '998 patent.

1. A method for managing execution of a plurality of software applications on an array of processing units, the method comprising:

299. Microsoft Azure manages execution of a plurality of software applications on an array of processing units, *e.g.*, FPGA processors. Microsoft Azure is configured to, and Microsoft uses Azure to, manage and host a number of applications across multiple processors within a group of processors.

300. At least the AC in conjunction with RM, SMs FMs, is used to manage the assignment of computing cores across multiple applications and multiple application tasks running, for instance, on multiple computing cores such as CPUs, GPUs and/or FPGA processors.

providing, for each software application of the plurality of software applications, one or more input buffers of a plurality of input buffers, each input buffer being provided for buffering processing load input directed to a respective software application of the plurality of software applications and being dedicated to the respective software application; and

301. Azure provides, for each software application of the plurality of software applications, one or more input buffers of a plurality of input buffers in the form of queues.

302. A given input buffer is provided for buffering processing load input directed to a

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

respective software application of the plurality of software applications and being specific to usage by the respective software application.

303. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," in Azure, queues are associated with specific applications. Data from those queues are routed under the direction of the AC in conjunction with the RM, SMs and FMs to respective FPGA processors that have been at least temporarily or dynamically assigned to performing a task specific to an application program.

repeatedly rearranging, by a controller comprising hardware logic and/or software logic, task assignment to the array of processing units and communication path connectivity for the array of processing units, wherein the array of processing units comprises a plurality of processing units of a reconfigurable type, each being configurable in two or more application specific configurations, and rearranging comprises, for each iteration of a plurality of iterations

304. Microsoft Azure repeatedly rearranges, by a controller, which comprises hardware logic and/or software logic, task assignment to the array of processing units and communication path connectivity for the array of processing units, wherein the array of processing units comprises a plurality of processing units of a reconfigurable type, each being configurable in two or more application specific configurations, and rearranging comprises, for each iteration of a plurality of iterations.

305. Microsoft Azure includes a controller (*e.g.*, at least the AC in conjunction with the RM, SMs, and FMs) comprising hardware logic and/or software logic that repeatedly rearranges task assignment to the array of processing units for each iteration.

306. The controller (*e.g.*, at least the AC in conjunction with the RM, SMs, and FMs) also repeatedly rearranges communication path connectivity among the array of processing units.

307. As explained above in the section entitled "Microsoft's Infringing Cloud

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

Computing Architecture," Microsoft's Azure cloud platform includes an array of processing units that are of a reconfigurable type (*e.g.*, FPGA processors), each being configurable in two or more application specific configurations (*e.g.*, for DNN, SQL database programs, or a web search ranking algorithm or speech recognition, encryption or compression applications, demanding their respective forms of hardware acceleration).

> ***allocating, to each software application of at least a portion of a plurality of software applications as a set of active software applications, a number of units of the array of processing units at least in part in accordance with a plurality of demand expressions, each demand expression of the plurality of demand expressions corresponding to a different software application of the plurality of software applications, wherein each demand expression of the plurality of demand expressions is based at least in part on a measure of an amount of a processing load input at the one or more input buffers of the respective software application,***

308. Microsoft's Azure (through *e.g.*, the RM in coordination with SMs, and FMs, which operate in conjunction with the AC) also allocates processing units to application programs at least in part in accordance with a plurality of demand expressions made by Azure that correspond to a different software application (*e.g.*, a program or group of programs that demand encryption acceleration).

309. In Azure, this is done in response to a change in demand expressed by the application programs and/or their respective tasks, for instance by the accumulation of data at the input buffer for an application program and/or task. According to Microsoft's '709 patent, "[a]cceleration components [*i.e.*, reconfigurable processing units H] can be allocated in a manner that balances load in a hardware acceleration plane, minimizes role switching, and adapts to demand changes." Ex. 13 at Abstract. The '709 patent further explains that "[d]uring continued monitoring, service manager 1521 can detect an increased demand for graph 1502. Based on the current allocation of acceleration components in hardware plane 106 and in view of allocation policy 1526, service manager 1521 can (re)allocate acceleration components to meet the increased

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

demand for graph 1502." *Id.* at 19:60-65.

310. Each demand expression of the plurality of demand expressions is based at least in part on a measure of an amount of a processing load input at the one or more input buffers of the respective software application (*e.g.*, Azure allocates cores based on the measure of data to be processed for each software application such that the amount of processing capacity is allocated at least in part in accordance with the processing needs of the applications).

311. In Azure, each software application has its own processing needs, which are queued at the input buffers. The number of processor cores allocated to each application changes based on the applications' need for processing capacity as reflected in the demand expressions. In the illustrative example depicted above, the Ranking Service program is allocated twice the number of cores that are assigned to the ADLA program due to the fact that the Ranking Service needs more processing capacity.

> ***assigning, for each application of the set of active software applications, one or more task instances of the respective active application to one or more processing units of the plurality of processing units in accordance with the allocating, wherein at least one task instance of the one or more task instances is a ready-to-execute task instance of the respective software application for processing the data at the one or more input data buffers of the respective software application, each task instance of the one or more task instances is assigned to a different processing unit of the one or more processing units such that assigning results in each processing unit of the plurality of processing units being assigned only one respective task instance for any given iteration of the plurality of iterations, and***

312. Microsoft Azure assigns (*e.g.*, through the SM in coordination with the RM and FMs, which operate in conjunction with the AC), for each application of the set of active software applications, one or more task instances of the respective active application to one or more processing units of the plurality of processing units in accordance with the allocating of processors from among the array of processing units.

313. At least one task instance of the one or more task instances, which are assigned to

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

a processing unit, is a ready-to-execute task instance of the respective software application for processing the data at the one or more input data buffers of the respective software application.

314. In Azure, each task instance of the one or more task instances is assigned to a different processing unit of the one or more processing units such that assigning results in each processing unit of the plurality of processing units being assigned one respective task instance for any given iteration of the plurality of iterations.

assigning comprises, for at least one task instance of the one or more task instances of a given software application of the plurality of software applications, reconfiguring the respective processing unit to an application specific configuration of the two or more application specific configurations, wherein the application specific configuration is associated with the at least one task instance, and

315. Microsoft Azure's FPGA-enabled architecture performs assigning that includes, for at least one task instance of the one or more task instances of a given software application of the plurality of software applications, reconfiguring the respective processing unit to an application specific configuration of the two or more application specific configurations, wherein the application specific configuration is associated with the at least one task instance.

316. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," the FPGA processors underlying the processor groups or graphs are dynamically reconfigured over time, in response to demand expressed by various applications, to perform specialized application-specific tasks such as web search ranking, data compression, DNN, or SQL database operations.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

causing connection, in accordance with the assigning, of the processing load input from each buffer of at least a portion of the plurality of input buffers to a different unit of the plurality of processing units to connect respective processing load input to the respective software application that the respective processing load input is directed to.

317. Microsoft Azure causes connection, in accordance with the assigning, of the processing load input from each buffer of at least a portion of the plurality of input buffers to a different unit of the plurality of processing units to connect respective processing load input to the respective software application that the respective processing load input is directed to.

318. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," the queues such as message queues are associated with specific applications. Data from those queues are provided under the direction of the AC in conjunction with the RM, SMs, and/or FMs to respective FPGA processors that have been temporarily or dynamically assigned to perform a specific task for a specific application. Each FPGA processor performs a task such as data compression or web search ranking and returns the result back to the head unit or SM.

319. Upon information and belief, in accordance with 35 U.S.C. § 287, Microsoft has had actual notice and knowledge of the '998 patent no later than its issuance.

320. Microsoft continues, without license, to make, use, offer for sale, import and/or sell in the United States services or products that infringe the '998 patent including specifically Microsoft Azure and its cloud computing functionalities.

321. Microsoft has directly infringed and continues to directly infringe the '998 patent by engaging in acts constituting patent infringement under 35 U.S.C. § 271(a) including but not necessarily limited to one or more of making, using, selling and offering to sell, in this District and elsewhere in the United States, and importing into the United States, the Microsoft Azure platform or components and services thereof.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

322. Microsoft's continuing infringement of the '998 patent is knowing, intentional, and willful.

323. Microsoft's infringement of the '998 patent has injured ThroughPuter in its business and property rights.

324. Microsoft's continuing infringement of the '998 patent is deliberate and willful and constitutes egregious misconduct.. In developing and offering its products and services, Microsoft is, at the very least, willfully blind to this ongoing infringement.

325. Pursuant to 35 U.S.C. § 284, ThroughPuter is entitled to recover monetary damages for the injuries arising from Microsoft's willful infringement in an amount to be determined at trial. Microsoft's infringement of the '998 patent has caused irreparable harm to ThroughPuter and will continue to cause such harm unless and until Microsoft's infringing activities are enjoined by this Court.

326. Microsoft's infringement of the '998 patent is exceptional and entitles ThroughPuter to attorneys' fees and costs incurred in prosecuting this action under 35 U.S.C. § 285.

COUNT 5

(Infringement of U.S. Patent No. 10,437,644)

327. ThroughPuter repeats and realleges each and every allegation contained above as though fully set forth herein.

328. On October 8, 2019, the USPTO duly and legally issued the '644 patent, entitled "Task Switching and Inter-Task Communications for Coordination of Applications Executing on a Multi-User Parallel Processing Architecture." A copy of the '644 patent is attached as Exhibit 5.

329. Mark Sandstrom is the sole and true inventor of the '644 patent.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

330. ThroughPuter, Inc. owns all right, title, and interest to and in the ’644 patent.

331. Microsoft infringes at least claims 1-4, 6-7, 9-11, and 13-20 of the ’644 patent.

332. Claim 1 of the ’644 patent is representative of the claims infringed by Microsoft and recites:

> 1. A system for managing execution of a plurality of software applications on an array of processing units, the system comprising:
>
> a core fabric comprising
>
> the array of processing units, and
>
> a plurality of input buffers, each input buffer being provided for buffering processing load input directed to a respective software application of the plurality of software applications and being dedicated to the respective software application, wherein
>
> each buffer of the plurality of input buffers is deployed in the core fabric apart from the array of processing units, and each software application of the plurality of software applications is provided one or more input buffers of the plurality of input buffers; and
>
> a controller comprising hardware logic and/or software logic for performing operations for repeatedly reconfiguring task assignment to the array of processing units and communication path connectivity for the array of processing units, the operations comprising, for each iteration of a plurality of iterations,
>
> allocating, to each software application of at least a portion of the plurality of software applications as a plurality of active software applications, a number of units of the array of processing units at least in part in accordance with a plurality of demand expressions, each demand expression of the plurality of demand expressions corresponding to a different software application of the plurality of software applications, wherein each demand expression of the plurality of demand expressions is based at least in part on a measure of an amount of the processing load input at the one or more input buffers of the respective software application, wherein allocating results in, for at least some of the plurality of software applications, a respective one or more allocated processing units of the array of processing units,
>
> obtaining, for each software application of the plurality of active software applications, identification of one or more tasks of the respective software application,
>
> assigning a respective task of the one or more tasks of each application of the plurality of active software applications to at least one respective

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

processing unit of the respective one or more allocated processing units in accordance with the allocating, wherein

for at least a portion of the plurality of active software applications, the number of processing units allocated to the respective software application is less than a number of possible tasks of the respective software application available for assigning, and

assigning comprises, for each software application of at least a portion of the plurality of active software applications, identifying at least one activating task of the one or more tasks of the respective software application not assigned to the array of processing units for execution for a current iteration of the plurality of iterations, and

for each activating task of each software application of the portion of the plurality of active software applications, identifying at least one available unit of the array of processing units, each available unit corresponding to a respective deactivating task of at least one software application of the plurality of software applications, each deactivating task not assigned to the array of processing units for execution for a next iteration of the plurality of iterations, and assigning the at least one activating task to the at least one available unit, and

causing connection, in accordance with the assigning, of the processing load input from each buffer of at least a portion of the plurality of input buffers to a different unit of the respective one or more allocated processing units to connect respective processing load input to a task of the one or more tasks of the respective software application that the respective processing load input is directed to;

wherein, for consecutive iterations of the plurality of iterations and for at least one software application of the plurality of active software applications, a number of the one or more allocated processing units allocated to the respective software application varies with the measure of the processing load input at the one or more input buffers of the respective software application for the respective iteration, and the assigning comprises assigning a varying number of tasks of concurrent instances of the respective software application corresponding to variations in the number of processing units allocated to the respective software application between the consecutive iterations.

333. On information and belief, Azure is implemented in a manner that meets each and every limitation of claim 1 of the ’644 patent.

1. A system for managing execution of a plurality of software applications on an

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

array of processing units

334. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," Microsoft Azure comprises a system for managing execution of a plurality of software applications on an array of processing units. Microsoft Azure is configured to, and Microsoft uses Azure to, manage and host a number of applications across a pool of processing nodes. The AC in conjunction with the RM, FMs, and SMs is used to manage the assignment of computing cores across multiple applications and multiple application tasks running, for instance, on multiple computing cores such as CPUs, GPUs, and/or FPGA processors.

a core fabric comprising the array of processing units, and

335. Microsoft Azure includes a core fabric comprising an array of processing units. As discussed above, Azure includes the Azure Fabric, which is a parallel processing core fabric including an array of processing units.

a plurality of input buffers, each input buffer being provided for buffering processing load input directed to a respective software application of the plurality of software applications and being dedicated to the respective software application, wherein each buffer of the plurality of input buffers is deployed in the core fabric apart from the array of processing units, and each software application of the plurality of software applications is provided one or more input buffers of the plurality of input buffers; and

336. Microsoft Azure includes a plurality of input buffers, each input buffer being provided for buffering processing load input directed to a respective software application of the plurality of software applications and being dedicated to usage by a respective software application.

337. Each buffer of the plurality of input buffers is deployed in the core fabric apart from the array of processing units, and each software application of the plurality of software applications is provided with one or more input buffers of the plurality of input buffers.

338. As explained above in the section entitled "Microsoft's Infringing Cloud

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

Computing Architecture," in Azure, queues are associated with specific applications, and act as input buffers. Those input buffers are deployed in the Azure Fabric and are separate from the processors with which the buffers interface.

a controller comprising hardware logic and/or software logic for performing operations for repeatedly reconfiguring task assignment to the array of processing units and communication path connectivity for the array of processing units, the operations comprising, for each iteration of a plurality of iterations,

339. Microsoft Azure includes a controller comprising hardware logic and/or software logic for performing operations for repeatedly reconfiguring both task assignment to the array of processing units and communication path connectivity for the array of processing units.

340. Microsoft Azure includes a controller (*e.g.*, at least the AC in conjunction with the RM, SMs, and FMs) comprising hardware logic and/or software logic that repeatedly rearranges task assignment to the array of processing units.

341. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," the controller also repeatedly rearranges communication path connectivity through the Azure Fabric among the array of processing units.

allocating, to each software application of at least a portion of the plurality of software applications as a plurality of active software applications, a number of units of the array of processing units at least in part in accordance with a plurality of demand expressions, each demand expression of the plurality of demand expressions corresponding to a different software application of the plurality of software applications, wherein each demand expression of the plurality of demand expressions is based at least in part on a measure of an amount of the processing load input at the one or more input buffers of the respective software application,

342. For each iteration, of a plurality of iterations, Microsoft Azure allocates, to each software application of at least a portion of the plurality of software applications as a plurality of active software applications, a number of units of the array of processing units at least in part in

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

accordance with a plurality of demand expressions, each demand expression of the plurality of demand expressions corresponding to a different software application of the plurality of software applications.

343. Microsoft's Azure cloud platform (through *e.g.*, the RM in coordination with the SMs and FMs, which work in conjunction with the AC) also allocates processing units to application programs at least in part in accordance with a plurality of demand expressions that correspond to the needs of a different software application (*e.g.*, a program for encryption services).

344. Each demand expression of the plurality of demand expressions is based at least in part on a measure of an amount of a processing load input at the one or more input buffers of the respective software application (*e.g.*, Azure allocates cores based on the measure of data to be processed for each software application such that the amount of processing capacity is allocated taking into consideration the respective processing needs of the software applications).

345. In Azure, each software application has its own processing needs, which are queued at the input buffers. The number of processor cores allocated to each application changes based on the applications' need for processing capacity as reflected in the demand expressions. In the illustrative example depicted above, the Ranking Service program is allocated twice the number of cores that are assigned to the ADLA program due to the fact that the Ranking Service needs more processing capacity.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

wherein allocating results in, for at least some of the plurality of software applications, a respective one or more allocated processing units of the array of processing units, obtaining, for each software application of the plurality of active software applications, identification of one or more tasks of the respective software application

346. The allocating performed by Microsoft Azure results in, for at least some of the plurality of software applications, a respective one or more allocated processing units of the array of processing units, obtaining, for each software application of the plurality of active software applications, identification of one or more tasks of the respective software application.

347. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," in Azure, a given application comprises a number of concurrent tasks to be executable by the processing units, which results in Azure identifying one or more tasks of the respective software applications.

assigning a respective task of the one or more tasks of each application of the plurality of active software applications to at least one respective processing unit of the respective one or more allocated processing units in accordance with the allocating, wherein

for at least a portion of the plurality of active software applications, the number of processing units allocated to the respective software application is less than a number of possible tasks of the respective software application available for assigning, and

348. Microsoft Azure assigns (*e.g.*, at least the SMs in coordination with the RM and FMs, which work in conjunction with the AC) a respective task of the one or more tasks of each application of the plurality of active software applications to at least one respective processing unit of the respective one or more allocated processing units in accordance with the allocating, wherein for at least a portion of the plurality of active software applications, the number of processing units allocated to the respective software application is less than a number of possible tasks of the respective software application available for assigning.

349. As explained above in the section entitled "Microsoft's Infringing Cloud

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

Computing Architecture," at least the SMs in coordination with the RM and FMs, which work in conjunction with the AC, assign a number of processing cores based on the demand expressed by each application. In Azure, application programs are sometimes allocated fewer processing units than the number of queued tasks, including when other applications express higher demand.

> ***assigning comprises, for each software application of at least a portion of the plurality of active software applications, identifying at least one activating task of the one or more tasks of the respective software application not assigned to the array of processing units for execution for a current iteration of the plurality of iterations, and for each activating task of each software application of the portion of the plurality of active software applications, identifying at least one available unit of the array of processing units, each available unit corresponding to a respective deactivating task of at least one software application of the plurality of software applications, each deactivating task not assigned to the array of processing units for execution for a next iteration of the plurality of iterations, and assigning the at least one activating task to the at least one available unit, and***

350. Microsoft Azure (*e.g.*, at least the SMs in coordination with the RM and FMs, which work in conjunction with the AC) identifies, for each software application of at least a portion of the plurality of active software applications, at least one activating task of the one or more tasks of a respective software application that was not assigned to the array of processing units for execution for a current iteration of the plurality of iterations.

351. For each activating task, Microsoft Azure identifies at least one available processing unit corresponding to a respective deactivating task of at least one software application of the plurality of software applications, with each such deactivating task being not assigned to the array of processing units for execution for a next iteration of the plurality of iterations.

352. Microsoft Azure assigns the at least one activating task to the at least one available unit.

353. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," Azure changes the graphs or groupings of processors dynamically and

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

periodically (*e.g.*, by the RM in coordination with the SMs and FMs, which work in conjunction with the AC). In some situations, such as the scenarios described in the '709 patent, in a given reconfiguration some of the running tasks are deactivated from their group or application, such that a new task from potentially a different application can be activated and placed on an available processing unit.

> ***causing connection, in accordance with the assigning, of the processing load input from each buffer of at least a portion of the plurality of input buffers to a different unit of the respective one or more allocated processing units to connect respective processing load input to a task of the one or more tasks of the respective software application that the respective processing load input is directed to;***

354. Microsoft Azure causes connection, in accordance with the assigning, of the processing load input from each buffer of at least a portion of the plurality of input buffers to a different unit of the respective one or more allocated processing units to connect respective processing load input to a task among the one or more tasks of the respective software application that the respective processing load input is directed to.

355. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," the queues such as message queues are associated with specific applications. Data from those queues are provided under the direction of the AC in conjunction with the RM, SMs, and FMs to respective processing units that have been temporarily or dynamically assigned to perform a task specific to an application program. When the data from one of the queues is connected to an FPGA processor of an FPGA Grouping, it may be a different processing unit than was previously connected to the queue.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

wherein, for consecutive iterations of the plurality of iterations and for at least one software application of the plurality of active software applications, a number of the one or more allocated processing units allocated to the respective software application varies with the measure of the processing load input at the one or more input buffers of the respective software application for the respective iteration, and the assigning comprises assigning a varying number of tasks of concurrent instances of the respective software application corresponding to variations in the number of processing units allocated to the respective software application between the consecutive iterations.

356. In Azure, these steps are repeated and, for consecutive iterations of the plurality of iterations and for at least one software application of the plurality of active software applications, a number of the one or more allocated processing units allocated to the respective software application varies with the measure of the processing load input at the one or more input buffers of the respective software application for the respective iteration, and the assigning comprises assigning a varying number of tasks of concurrent instances of the respective software application corresponding to variations in the number of processing units allocated to the respective software application between the consecutive iterations.

357. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," the RM in coordination with the SMs and FMs, which work in conjunction with the AC, allocates a number of processing cores based on the processing needs of each application. One task is assigned to each allocated processing core for a given allocation interval. During each periodic reallocation, the number of processing cores allocated to each application varies based on the processing needs of the various hosted applications.

358. On information and belief, in accordance with 35 U.S.C. § 287, Microsoft has had actual notice and knowledge of the '644 patent no later than its issuance.

359. Microsoft has had knowledge of and notice of the chain of applications underlying the '644 patent since at least January 2018 when ThroughPuter brought the '090 and '833 patents as well as its portfolio, including a number of pending patent applications, to the attention of

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

Microsoft in the January 2018 letter.

360. On information and belief, Microsoft continues without license to make, use, import, market, offer for sale, and/or sell in the United States services or products that infringe the '644 patent.

361. Microsoft has directly infringed and continues to directly infringe the '644 patent by engaging in acts constituting infringement under 35 U.S.C. § 271(a), including but not necessarily limited to one or more of making, using, selling and offering to sell, in this District and elsewhere in the United States, and importing into the United States, the Microsoft Azure platform or components and services thereof.

362. Microsoft's continuing infringement of the '644 patent is knowing, intentional, and willful.

363. Microsoft's infringement of the '644 patent has injured ThroughPuter in its business and property rights. Microsoft's continued infringement of the '644 patent is deliberate and willful and constitutes egregious misconduct. In developing and offering its products and services, Microsoft is, at the very least, willfully blind to this ongoing infringement.

364. ThroughPuter is entitled to recover monetary damages for the injuries arising from Microsoft's willful infringement pursuant to 35 U.S.C. § 284 in an amount to be determined at trial. Microsoft's infringement of the '644 patent has caused irreparable harm to ThroughPuter and will continue to cause such harm unless and until Microsoft's infringing activities are enjoined by this Court.

365. Microsoft's infringement of the '644 patent is exceptional and entitles ThroughPuter to attorneys' fees and costs incurred in prosecuting this action under 35 U.S.C. § 285.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

COUNT 6

(Infringement of U.S. Patent No. 10,430,242)

366. ThroughPuter repeats and realleges each and every allegation contained above as though fully set forth herein.

367. On October 1, 2019, the USPTO duly and legally issued the '242 patent, entitled "Task Switching and Inter-Task Communications for Coordination of Applications Executing on a Multi-User Parallel Processing Architecture." A copy of the '242 patent is attached as Exhibit 6.

368. Mark Sandstrom is the sole and true inventor of the '242 patent.

369. ThroughPuter, Inc. owns all right, title, and interest to and in the '242 patent.

370. Microsoft infringes at least claims 1-3, 6-7, 9-10, 12, 14-20, and 23 of the '242 patent.

371. Claim 1 of the '242 patent is representative of the claims infringed by Microsoft and recites:

> 1. A system for managing execution of a plurality of software applications on an array of processing units, the system comprising:
>
> a core fabric comprising
>
> the array of processing units, and
>
> a plurality of input data buffers, each input data buffer being provided for buffering input data directed to a respective software application of the plurality of software applications and being dedicated to the respective software application, wherein
>
> each buffer of the plurality of input data buffers is deployed in the core fabric apart from the array of processing units, and each software application of the plurality of software applications is provided one or more input data buffers of the plurality of input data buffers; and
>
> a controller comprising hardware logic and/or software logic for performing operations for repeatedly reconfiguring task assignment to the array of processing units, the operations comprising, for each iteration of a plurality of iterations,
>
> identifying, for each software application of at least a portion of the plurality of software applications, an amount of input data at one or more input data buffers of the plurality of input data buffers buffering data for the respective software application,

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

allocating, to each software application of the portion of the plurality of software applications, a number of processing units of the array of processing units based at least in part on the amount of input data buffered for the respective software application, and for each software application of the portion,

i) assigning one or more task instances of the respective software application for concurrent processing of the amount of input data to the number of processing units allocated to the respective software application by the allocating as one or more assigned instances, and

ii) adjusting, based at least in part on a change in a count of units between the number of processing units allocated to the respective software application and a number of previously allocated processing units allocated to the respective software application during a previous iteration of the plurality of iterations, a relative portion of the amount of input data to be processed by at least one assigned instance of the one or more assigned instances;

wherein, for one or more iterations of the plurality of iterations where a current number of the one or more processing units allocated to any given software application of the portion of the plurality of software applications is increased by the allocating, adjusting comprises relatively decreasing the portion of the amount of input data to be processed by at least one instance of the one or more assigned instances of the respective software application; and

wherein for one or more other iterations of the plurality of iterations where a present number of the one or more processing units allocated to any given software application of the portion of the plurality of software applications is decreased by the allocating, adjusting comprises relatively increasing the portion of the amount of input data to be processed by at least one instance of the one or more assigned instances of the respective software application.

372. On information and belief, Azure is implemented in a manner that meets each and every limitation of claim 1 of the ’242 patent.

1. A system for managing execution of a plurality of software applications on an array of processing units, the system comprising:

373. As explained above in the section entitled “Microsoft’s Infringing Cloud Computing Architecture,” Microsoft Azure comprises a system for managing execution of a plurality of software applications on an array of processing units. Microsoft Azure is configured to, and Microsoft uses Azure to, manage and host a number of applications across a pool of processing nodes. The AC in conjunction with the RM, SMs, and FMs is used to manage the

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

assignment of computing cores across multiple applications and multiple application tasks running, for instance, on multiple computing cores such as CPUs, GPUs, and/or FPGA processors.

a core fabric comprising the array of processing units, and

374. Microsoft Azure includes the Azure Fabric, which is a core fabric comprising an array of processing units.

a plurality of input data buffers, each input data buffer being provided for buffering input data directed to a respective software application of the plurality of software applications and being dedicated to the respective software application, wherein
each buffer of the plurality of input data buffers is deployed in the core fabric apart from the array of processing units, and each software application of the plurality of software applications is provided one or more input data buffers of the plurality of input data buffers; and

375. Microsoft Azure includes a plurality of input data buffers, each input data buffer being provided for buffering input data directed to a respective software application of the plurality of software applications and being dedicated by the respective software application, wherein each buffer of the plurality of input data buffers is deployed in the core fabric apart from the array of processing units, and each software application of the plurality of software applications is provided one or more input data buffers of the plurality of input data buffers.

376. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," in Azure, queues are associated with specific applications, which act as input buffers. Those input buffers are deployed as part of the core fabric and separate from the processors with which the buffers interface.

a controller comprising hardware logic and/or software logic for performing operations for repeatedly reconfiguring task assignment to the array of processing units, the operations comprising, for each iteration of a plurality of iterations,

377. As explained above in the section entitled "Microsoft's Infringing Cloud

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

Computing Architecture," Microsoft Azure includes a controller (*e.g.*, at least the AC in conjunction with the RM, SMs, and FMs) comprising hardware logic and/or software logic for performing operations for repeatedly reconfiguring task assignment to the array of processing units.

378. The controller also repeatedly rearranges communication path connectivity among the array of processing units through the Azure Fabric.

identifying, for each software application of at least a portion of the plurality of software applications, an amount of input data at one or more input data buffers of the plurality of input data buffers buffering data for the respective software application,

379. For each iteration of a plurality of iterations, Microsoft Azure identifies, for each software application of at least a portion of the plurality of software applications, an amount of input data at one or more input data buffers of the plurality of input data buffers buffering data for the respective software application.

380. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," Azure identifies data at the data queues that is specific to a respective software application.

allocating, to each software application of the portion of the plurality of software applications, a number of processing units of the array of processing units based at least in part on the amount of input data buffered for the respective software application, and

381. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," the data queues act as input buffers, while the RM in coordination with the SMs and FMs, which work in conjunction with the AC, allocates processing units based at least in part on the amount of application program data stored at a queue.

382. Microsoft Azure allocates, to each software application of the portion of the plurality of software applications, a number of processing units of the array of processing units

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

based at least in part on the amount of input data buffered for the respective software application.

383. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," the controller (*e.g.*, at least the AC in conjunction with the RM, SMs, and FMs) allocates accelerators (*e.g.*, FPGA processors) to software applications. Each software application requires processing capacity as reflected by the data load buffered in the queues. The number of processor cores allocated to each application is adjustable based on the application program's processing load. In the illustrative example depicted above, the Ranking Service program is allocated twice the number of cores that are assigned to the ADLA program due to the processing needs of the Ranking Service as compared to the ADLA program.

for each software application of the portion,

i) assigning one or more task instances of the respective software application for concurrent processing of the amount of input data to the number of processing units allocated to the respective software application by the allocating as one or more assigned instances, and

384. Microsoft Azure assigns one or more task instances of the respective software application for concurrent processing of the amount of input data to the number of processing units allocated to the respective software application by the allocating as one or more assigned instances.

385. In Azure, one or more task instances is assigned (*e.g.*, through the SM in coordination with the RM and FMs, which work in conjunction with the AC) to different processing units of the one or more processing units such that assigning results in individual processing units of the plurality of processing units being assigned their respective task instances for concurrent processing of the input load for the respective applications.

ii) adjusting, based at least in part on a change in a count of units between the number of processing units allocated to the respective software application and a number of previously allocated processing units allocated to the respective software application during a previous iteration of the plurality of iterations, a relative portion of the amount of input data to be processed by at least one assigned instance of the one or more assigned instances;

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

386. Microsoft Azure adjusts, based at least in part on a change in a count of units between the number of processing units allocated to the respective software application and a number of previously allocated processing units allocated to the respective software application during a previous iteration of the plurality of iterations, a relative portion of the amount of input data to be processed by at least one assigned instance of the one or more assigned instances.

387. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," in Azure, a given software application has its own processing load at the input buffers. The number of processor cores allocated to any given application is adjustable based on the application's processing needs. In the illustrative example depicted above, the Ranking Service program is allocated twice the number of cores than are allocated to the ADLA due to the relative processing needs between the two programs. In that scenario, the number of executing instances assigned per a given program may be adjusted due to the reallocation of cores.

> ***wherein, for one or more iterations of the plurality of iterations where a current number of the one or more processing units allocated to any given software application of the portion of the plurality of software applications is increased by the allocating, adjusting comprises relatively decreasing the portion of the amount of input data to be processed by at least one instance of the one or more assigned instances of the respective software application; and***
>
> ***wherein for one or more other iterations of the plurality of iterations where a present number of the one or more processing units allocated to any given software application of the portion of the plurality of software applications is decreased by the allocating, adjusting comprises relatively increasing the portion of the amount of input data to be processed by at least one instance of the one or more assigned instances of the respective software application.***

388. The adjusting performed by Microsoft Azure comprises, for one or more iterations of the plurality of iterations where a current number of the one or more processing units allocated to any given software application of the portion of the plurality of software applications is increased by the allocating, *inter alia*, relatively decreasing the portion of the amount of input data

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

to be processed by at least one instance of the one or more assigned instances of the respective software application. For one or more other iterations of the plurality of iterations where a present number of the one or more processing units allocated to any given software application of the portion of the plurality of software applications is decreased by the allocating, the adjusting comprises relatively increasing the portion of the amount of input data to be processed by at least one instance of the one or more assigned instances of the respective software application.

389. As discussed above, Azure changes the graphs or groupings of processing units dynamically and periodically (*e.g.*, at least through the AC in conjunction with the RM, SMs, and FMs). In some situations, such as the scenarios described in the '709 patent, in a given reallocation, some of the processing units are added to the group of processors performing a task (*e.g.*, mathematical algorithm) for an application (*e.g.*, search ranking). For a given input load level, the amount of input data sent to each processing unit in that group decreases since there are more processor units performing the mathematical algorithm on a given amount of input data to be processed, and vice versa in case fewer processing units were allocated for that given workload.

390. On information and belief, in accordance with 35 U.S.C. § 287, Microsoft has had actual notice and knowledge of the '242 patent no later than its issuance.

391. Microsoft has had knowledge of and notice of the chain of applications underlying the '242 patent since at least January 2018 when ThroughPuter brought the '090 and '833 patents as well as its portfolio, including a number of pending patent applications, to the attention of Microsoft in the January 2018 letter.

392. On information and belief, Microsoft continues without license to make, use, import, market, offer for sale, and/or sell in the United States services or products that infringe the '242 patent.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

393. Microsoft has directly infringed and continues to directly infringe the ’242 patent by engaging in acts constituting infringement under 35 U.S.C. § 271(a), including but not necessarily limited to one or more of making, using, selling and offering to sell, in this District and elsewhere in the United States, and importing into the United States, the Microsoft Azure platform or components and services thereof.

394. Microsoft’s continuing infringement of the ’242 patent is knowing, intentional, and willful.

395. Microsoft’s infringement of the ’242 patent has injured ThroughPuter in its business and property rights. Microsoft’s continuing infringement of the ’242 patent is deliberate and willful and constitutes egregious misconduct. In developing and offering its products and services, Microsoft is, at the very least, willfully blind to this ongoing infringement.

396. ThroughPuter is entitled to recover monetary damages for the injuries arising from Microsoft’s willful infringement pursuant to 35 U.S.C. § 284 in an amount to be determined at trial. Microsoft’s infringement of the ’242 patent has caused irreparable harm to ThroughPuter and will continue to cause such harm unless and until Microsoft’s infringing activities are enjoined by this Court.

397. Microsoft’s infringement of the ’242 patent is exceptional and entitles ThroughPuter to attorneys’ fees and costs incurred in prosecuting this action under 35 U.S.C. § 285.

COUNT 7

(Infringement of U.S. Patent No. 10,318,353)

398. ThroughPuter repeats and realleges each and every allegation contained above as though fully set forth herein.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

399. On June 11, 2019, the USPTO duly and legally issued the '353 patent, entitled "Concurrent Program Execution Optimization." A copy of the '353 patent is attached as Exhibit 7.

400. Mark Sandstrom is the sole and true inventor of the '353 patent.

401. ThroughPuter, Inc. owns all right, title, and interest to and in the '353 patent.

402. Microsoft infringes at least claims 1-2 and 8-14 of the '353 patent.

403. Claim 1 of the '353 patent is representative of the claims infringed by Microsoft and recites:

> 1. A system for processing a set of computer program instances, comprising:
>
> a plurality of processing stages, at least one of the plurality of processing stages comprising multiple processing cores, wherein,
>
> each given task of a plurality of tasks of a given program instance of the set of program instances is hosted at a different stage of the plurality of processing stages as a local task of the given program instance at the respective stage, and
>
> for at least one of the multiple processing cores of a given processing stage of the plurality of processing stages, a local task of one of the program instances is assigned as an active task instance for execution for a period of time; and
>
> a group of multiplexers each connecting inter-task communications (ITC) data to a respective stage of the plurality of processing stages, wherein at least one multiplexer of the group of multiplexers is a hardware resource dedicated to the local task, wherein
>
> the at least one multiplexer is configured to connect ITC data to any processing core of the multiple processing cores to which the local task is assigned for execution for the period of time.

404. On information and belief, Azure is implemented in a manner that meets each and every limitation of claim 1 of the '353 patent.

1. A system for processing a set of computer program instances, comprising:

405. As explained above in the section entitled, "Microsoft's Infringing Cloud Computing Architecture," Microsoft Azure comprises a system for processing a set of computer program instances.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

a plurality of processing stages, at least one of the plurality of processing stages comprising multiple processing cores, wherein, each given task of a plurality of tasks of a given program instance of the set of program instances is hosted at a different stage of the plurality of processing stages as a local task of the given program instance at the respective stage, and

406. Microsoft Azure includes a plurality of processing stages where at least one of the plurality of processing stages comprises multiple processing cores, wherein each given task of a plurality of tasks of a given program instance of the set of program instances is hosted at a different stage of the plurality of processing stages as a local task of the given program instance at the respective stage.

407. Microsoft Azure is configured to, and Microsoft uses Azure to, manage and host a number of applications across a pool of processing nodes. The AC in conjunction with the RM, SMs, and FMs is used to manage the assignment of computing cores across multiple applications and multiple application tasks running, for instance, on multiple computing cores such as CPUs, GPUs, and/or FPGA processors.

408. Azure performs a requested task by breaking the requested functionality or computation into various tasks, each of which is executed in a separate processing stage. Each processing stage comprises at least one computing core. As illustrated in Fig. 2 of Microsoft's '392 patent (Ex. 18), Azure assigns application instances for execution on a pipelined series of stages, each stage comprising at least one reconfigurable processing element H. Each processing element executes its assigned local task.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

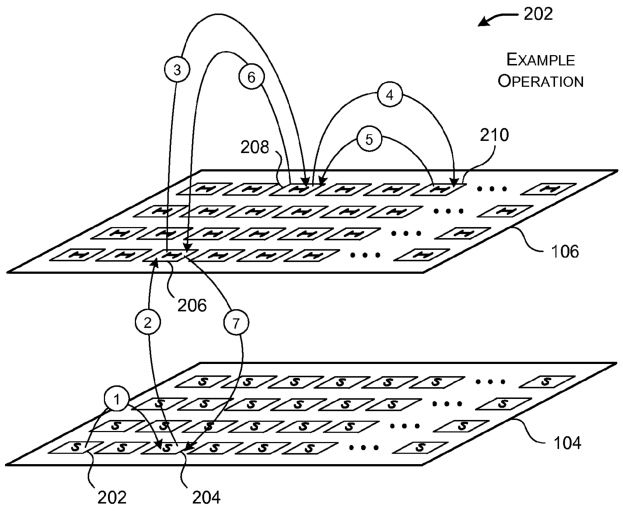


FIG. 2

Ex. 18, Fig. 2.

409. When a certain processing stage or task is particularly computationally intensive and/or parallelizable, the stage or task may be executed by more than one computing core. This functionality is explained, for example, in Microsoft's '392 patent:

> In another scenario, assume that the function has plural stages that the function parsing component 3220 maps to different acceleration components. But assume that one stage is more labor intensive than the others. To avoid a bottleneck in processing associated with this stage, the function parsing component 3220 can allocate two or more acceleration components that operate in parallel for this stage.

Ex. 47 at 32:40-47.

> ***for at least one of the multiple processing cores of a given processing stage of the plurality of processing stages, a local task of one of the program instances is assigned as an active task instance for execution for a period of time; and***

410. In Microsoft Azure, for at least one of the multiple processing cores of a given processing stage of the plurality of processing stages, a local task of one of the program instances is assigned as an active task instance for execution for a period of time.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

411. In Azure, a local task to be executed by a processing stage is assigned as an active task instance for execution on a processing core for a period of time.

> ***a group of multiplexers each connecting inter-task communications (ITC) data to a respective stage of the plurality of processing stages, wherein at least one multiplexer of the group of multiplexers is a hardware resource dedicated to the local task, wherein the at least one multiplexer is configured to connect ITC data to any processing core of the multiple processing cores to which the local task is assigned for execution for the period of time.***

412. Microsoft Azure includes a group of multiplexers each connecting inter-task communications (ITC) data to a respective stage of the plurality of processing stages, wherein at least one multiplexer of the group of multiplexers is a hardware resource dedicated to the local task, wherein the at least one multiplexer is configured to connect ITC data to any processing core of the multiple processing cores to which the local task is assigned for execution for the period of time.

413. As explained in the above and further discussed in Microsoft's '392 patent, each of the acceleration components H (2502 in the figure below) communicates with other acceleration components through **multiport switch 2532**, **multiplexers**, and additional switches and **multiplexers** in the top-of-rack (TOR) switch (not shown).

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

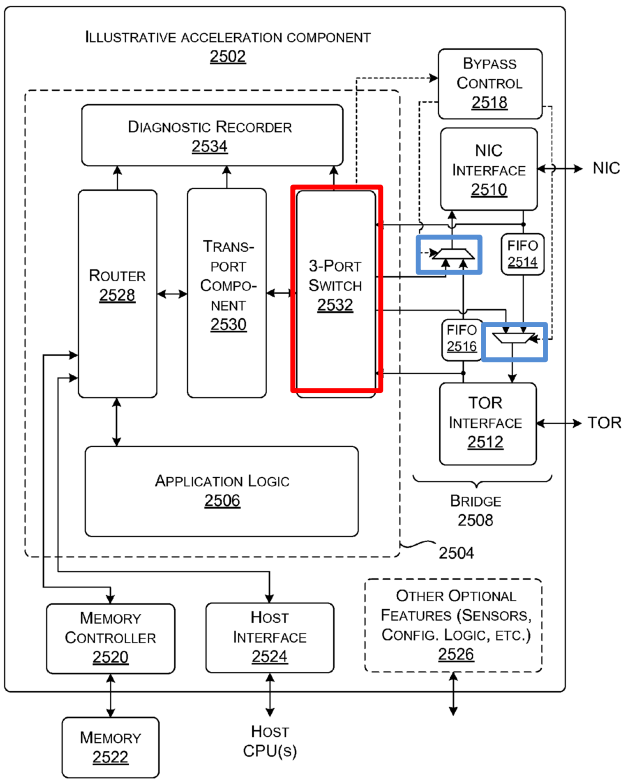


FIG. 25

Ex. 18, Fig. 25 (annotated).

414. When one acceleration component needs to send inter-task communications data to another of the pipelined stages executing on a remote acceleration component, the data is sent through **multiport switch 2532** (which is an example of a hardware resource dedicated to the local task), **multiplexers**, and additional switches and **multiplexers** in the TOR switch, through the TOR interface on the remote acceleration component, and to the **multiport switch 2532** on the remote acceleration component.

415. When one acceleration component needs to receive data from another of the pipelined stages executing on a different acceleration component, the data is sent through **multiport switch 2532** on the remote acceleration component, through the **multiplexers** on the

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

remote acceleration component, through the additional switches and **multiplexers** in the TOR switch, through the TOR interface 2512 on the local acceleration component, and to the **multiport switch 2532** on the local acceleration component.

416. In accordance with 35 U.S.C. § 287, Microsoft has had actual notice and knowledge of the '353 patent no later than its issuance.

417. On information and belief, Microsoft continues without license to make, use, import, market, offer for sale, and/or sell in the United States services or products that infringe the '353 patent.

418. Microsoft has directly infringed and continues to directly infringe the '353 patent by engaging in acts constituting infringement under 35 U.S.C. § 271(a), including but not necessarily limited to one or more of making, using, selling and offering to sell, in this District and elsewhere in the United States, and importing into the United States, the Microsoft Azure platform or components and services thereof.

419. Microsoft's continued infringement of the '353 patent is knowing, intentional, and willful.

420. Microsoft's infringement of the '353 patent has injured ThroughPuter in its business and property rights. Microsoft's continued infringement of the '353 patent is deliberate and willful and constitutes egregious misconduct. In developing and offering its products and services, Microsoft is, at the very least willfully blind to this ongoing infringement.

421. ThroughPuter is entitled to recover monetary damages for the injuries arising from Microsoft's willful infringement pursuant to 35 U.S.C. § 284 in an amount to be determined at trial. Microsoft's infringement of the '353 patent has caused irreparable harm to ThroughPuter and will continue to cause such harm unless and until Microsoft's infringing activities are enjoined by

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

this Court.

422. Microsoft's infringement of the '353 patent is exceptional and entitles ThroughPuter to attorneys' fees and costs incurred in prosecuting this action under 35 U.S.C. § 285.

COUNT 8

(Infringement of U.S. Patent No. 10,310,902)

423. ThroughPuter repeats and realleges each and every allegation contained above as though fully set forth herein.

424. On June 4, 2019, the USPTO duly and legally issued the '902 patent, entitled "System and Method for Input Data Load Adaptive Parallel Processing." A copy of the '902 patent is attached as Exhibit 8.

425. Mark Sandstrom is the sole and true inventor of the '902 patent.

426. ThroughPuter, Inc. owns all right, title, and interest to and in the '902 patent.

427. Microsoft infringes at least claims 1-2 and 4-16 of the '902 patent.

428. Claim 1 of the '902 patent is representative of the claims infringed by Microsoft and recites:

> 1. A system, executing on at least one of hardware logic and software logic executing on a plurality of processors, for hosting a plurality of application programs, the system comprising:
>
> a plurality of processing data input buffers, each input buffer of the plurality of processing data input buffers queuing data for a corresponding instance of one program of the plurality of application programs, wherein each program of the plurality of programs comprises a plurality of instances;
>
> an array of cores of computing capacity;
>
> a first subsystem configured to allocate the array of cores of computing capacity among the plurality of application programs, wherein

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

allocating comprises allocating the array of cores of computing capacity based at least in part on a respective volume of processing data at each buffer of the plurality of processing data input buffers, and a respective processing quota of each application program of at least a portion of the plurality of application programs, and allocating comprises allocating more than one core of the array of cores of computing capacity to at least one of the plurality of application programs;

a second subsystem configured to assign, for each program of the plurality of application programs, each core allocated to the respective program to a different instance of the plurality of instances of the respective program, wherein

the assigning results in assignment of a plurality of selected instances of the plurality of instances of the plurality of application programs, the plurality of instances comprising one or more executable instances of each program of the plurality of application programs, wherein

a number of the plurality of selected instances is fewer than a maximum number of the plurality of instances, and the plurality of selected instances is selected based at least in part on respective volumes of processing data available for each instance of the plurality of instances of the respective program at the portion of the plurality of data input buffers queuing data for the respective program, and according to the assigning, control connectivity between the plurality of processing data input buffers and the array of cores; and

a third subsystem configured, according to the controlling, to establish direct data access from each input buffer of at least a subset of the plurality of processing data input buffers to the respective core of the array of cores that is assigned to a given corresponding instance of the program for which the respective input buffer is queuing data;

wherein the array of cores is periodically allocated by the first subsystem and assigned by the second subsystem based at least in part on changes in respective volumes of processing data associated with each program of the plurality of application programs.

429. On information and belief, Azure is implemented in a manner that meets each and every limitation of claim 1 of the '902 patent.

1. A system, executing on at least one of hardware logic and software logic executing on a plurality of processors, for hosting a plurality of application programs, the system comprising:

430. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," Microsoft Azure comprises a system, executing on at least one of

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

hardware logic and software logic executing on a plurality of processors, for hosting a plurality of application programs. Microsoft Azure is configured to, and Microsoft uses Azure to, manage and host a number of applications across a pool of processing nodes. The AC in conjunction with the RM, SMs, and FMs is used to manage the assignment of computing cores across multiple applications and multiple application tasks running, for instance, on multiple computing cores such as CPUs, GPUs, and/or FPGA processors.

a plurality of processing data input buffers, each input buffer of the plurality of processing data input buffers queuing data for a corresponding instance of one program of the plurality of application programs, wherein each program of the plurality of programs comprises a plurality of instances;

431. Microsoft Azure includes a plurality of processing data input buffers, each input buffer of the plurality of processing data input buffers queuing data for a corresponding instance of one program of the plurality of application programs, wherein each program of the plurality of programs comprises a plurality of instances.

432. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," in Azure, queues are associated with specific applications, which act as input buffers for queuing data provided by application programs.

an array of cores of computing capacity;

433. Microsoft Azure includes an array of cores of computing capacity.

434. As discussed above, Azure includes an array of processing cores (*e.g.*, FPGA processors).

a first subsystem configured to allocate the array of cores of computing capacity among the plurality of application programs, wherein allocating comprises allocating the array of cores of computing capacity based at least in part on a respective volume of processing data at each buffer of the plurality of processing data input buffers, and a respective processing quota of each application program of at least a portion of the plurality of application programs, and allocating

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

comprises allocating more than one core of the array of cores of computing capacity to at least one of the plurality of application programs;

435. Microsoft Azure includes a first subsystem configured to allocate the array of cores of computing capacity among the plurality of application programs, wherein allocating comprises allocating the array of cores of computing capacity based at least in part on a respective volume of processing data at each buffer of the plurality of processing data input buffers, and a respective processing quota of each application program of at least a portion of the plurality of application programs, and allocating comprises allocating more than one core of the array of cores of computing capacity to at least one of the plurality of application programs.

436. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," the RM in conjunction with the SMs and FMs, which work in coordination with the AC, periodically reallocates processing cores among various application programs (*e.g.*, web search ranking, DNN, SQL, SDN offload) based on the processing needs of each application.

437. Microsoft Azure also supports subscriptions for assured processing resource access levels for an additional fee. The RM in conjunction with the SMs and FMs, which work in coordination with the AC, allocates processing resources based in part on the amount of resources subscribed to by a given client for its applications.

a second subsystem configured to assign, for each program of the plurality of application programs, each core allocated to the respective program to a different instance of the plurality of instances of the respective program, wherein the assigning results in assignment of a plurality of selected instances of the plurality of instances of the plurality of application programs, the plurality of instances comprising one or more executable instances of each program of the plurality of application programs, wherein a number of the plurality of selected instances is fewer than a maximum number of the plurality of instances, and the plurality of selected instances is selected based at least in part on respective volumes of processing data available for each instance of the plurality of instances of the respective program at the portion of the plurality of data input buffers queuing data for the respective program, and according to the assigning, control

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

connectivity between the plurality of processing data input buffers and the array of cores; and

438. Microsoft Azure includes a second subsystem configured to assign, for each program of the plurality of application programs, each core allocated to the respective program to a different instance of the plurality of instances of the respective program.

439. The assigning results in assignment of a plurality of selected instances of the plurality of instances of the plurality of application programs, the plurality of instances comprising one or more executable instances of each program of the plurality of application programs.

440. A number of the plurality of selected instances is fewer than a maximum number of the plurality of instances, and the plurality of selected instances is selected based at least in part on respective volumes of processing data available for each instance of the plurality of instances of the respective program at the portion of the plurality of data input buffers queuing data for the respective program, and according to the assigning, control connectivity between the plurality of processing data input buffers and the array of cores.

441. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," the SMs in coordination with the RM and FMs, which work in conjunction with the AC, select tasks to be assigned to, and executed by, each processing core in a group. The selection and assignment of the respective application instances is a function of the number of cores in the group, the readiness of the application task instances for execution, and the processing needs of each application instance as manifested at the input buffers or queues for each program. A given application will often not have all of its possible tasks processed all the time.

442. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," the controller through the Azure Fabric controls connectivity between the processing cores.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

a third subsystem configured, according to the controlling, to establish direct data access from each input buffer of at least a subset of the plurality of processing data input buffers to the respective core of the array of cores that is assigned to a given corresponding instance of the program for which the respective input buffer is queuing data; wherein the array of cores is periodically allocated by the first subsystem and assigned by the second subsystem based at least in part on changes in respective volumes of processing data associated with each program of the plurality of application programs.

443. Microsoft Azure includes a third subsystem configured, according to the controlling, to establish direct data access from each input buffer of at least a subset of the plurality of processing data input buffers to the respective core of the array of cores that is assigned to a given corresponding instance of the program for which the respective input buffer is queuing data.

444. Azure's FMs in coordination with SMs and RM, which work in conjunction with the AC, establish direct data input access from the queues to the appropriate processing cores.

445. In Microsoft Azure, the array of cores is periodically allocated by the first subsystem and assigned by the second subsystem based at least in part on changes in respective volumes of processing data associated with each program of the plurality of application programs.

446. As discussed above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," the RM in coordination with the SMs and FMs (which work in conjunction with the AC) allocates and assigns processors based on changes in processing volume. In some situations, such as the scenarios described in the '709 patent, in a given reallocation some of the processor cores are added to the group of processors performing a task (*e.g.*, mathematical algorithm) for an application (*e.g.*, search). An impact of this is that the amount of input data sent to individual cores in that group decreases since there are more cores performing the mathematical algorithm per a given volume of input data to be processed (*e.g.*, searched for key terms).

447. Upon information and belief, in accordance with 35 U.S.C. § 287, Microsoft has

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

had actual notice and knowledge of the ’902 patent no later than its issuance.

448. Microsoft has had knowledge of and notice of the chain of applications underlying the ’902 patent since at least January 2018 when ThroughPuter brought the ’090 and ’833 patents as well as its portfolio, including a number of pending patent applications, to the attention of Microsoft in the January 2018 letter.

449. On information and belief, Microsoft continues without license to make, use, import, market, offer for sale, and/or sell in the United States services or products that infringe the ’902 patent.

450. Microsoft has directly infringed and continues to directly infringe the ’902 patent by engaging in acts constituting infringement under 35 U.S.C. § 271(a), including but not necessarily limited to one or more of making, using, selling and offering to sell, in this District and elsewhere in the United States, and importing into the United States, the Microsoft Azure platform or components and services thereof.

451. Microsoft’s continuing infringement of the ’902 patent is knowing, intentional, and willful.

452. Microsoft’s infringement of the ’902 patent has injured ThroughPuter in its business and property rights. Microsoft’s continuing infringement of the ’902 patent is deliberate and willful and constitutes egregious misconduct. In developing and offering its products and services, Microsoft is, at the very least, willfully blind to this ongoing infringement.

453. ThroughPuter is entitled to recover monetary damages for the injuries arising from Microsoft’s willful infringement pursuant to 35 U.S.C. § 284 in an amount to be determined at trial. Microsoft’s infringement of the ’902 patent has caused irreparable harm to ThroughPuter and will continue to cause such harm unless and until Microsoft’s infringing activities are enjoined by

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

this Court.

454. Microsoft's infringement of the '902 patent is exceptional and entitles ThroughPuter to attorneys' fees and costs incurred in prosecuting this action under 35 U.S.C. § 285.

COUNT 9

(Infringement of U.S. Patent No. 10,133,599)

455. ThroughPuter repeats and realleges each and every allegation contained above as though fully set forth herein.

456. On November 20, 2018, the USPTO duly and legally issued the '599 patent, entitled "Application Load Adaptive Multi-Stage Parallel Data Processing Architecture." A copy of the '599 patent is attached as Exhibit 9.

457. Mark Sandstrom is the sole and true inventor of the '599 patent.

458. ThroughPuter, Inc. owns all right, title, and interest to and in the '599 patent.

459. Microsoft infringes at least claims 1-8, 10-11, 13-15, and 17-20 of the '599 patent.

460. Claim 1 of the '599 patent is representative of the claims infringed by Microsoft and recites:

> 1. A system including processors and comprising a plurality of subsystems, implemented on hardware logic and/or software logic executing on the processors, for dynamic resource management of a pool of processing resources on behalf of application programs, the system comprising:
>
> a first subsystem configured to periodically allocate a plurality of processing units of the pool of processing resources among a plurality of application programs over time, wherein
>
> the plurality of processing units comprises units of at least two types of processing units and/or units of a reconfigurable type, and
>
> allocating the plurality of processing units among the plurality of application programs is based at least in part on i) a respective processing demand of each of the plurality of application programs, and ii) a respective processing resource quota of each program of a subset of the plurality of application

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

programs;

a second subsystem configured to, for each program of the plurality of application programs, select a set of highest priority instances of one or more instances of a respective program, wherein

a number of instances in the set of instances corresponds to a number of processing units allocated to the respective application program by the first subsystem during a current allocation period, and each instance of the set of highest priority instances is selected based at least in part on a) the number of processing units allocated to the respective program and b) a relative readiness for execution among the one or more instances of the respective application program; and

a third subsystem configured to assign, for each of the plurality of application programs, the respective set of highest priority instances to a subset of the processing units allocated by the first subsystem for the respective application program for execution during an upcoming allocation period, wherein, for at least a portion of instances of at least a portion of the plurality of application programs,

a respective instance of the respective set of highest priority instances is associated with a processing unit type or configuration, wherein assigning comprises placing the respective instance to a respective processing unit of the plurality of processing units based at least in part on the processing unit type or configuration associated with the respective instance, and prioritizing placement of the respective instance to the particular type or configuration processing unit demanded;

wherein the system is further configured to periodically adjust allocations of the plurality of processing units of the pool of processing resources over time, wherein, for each periodic adjustment, the adjusting is followed by the selecting and the assigning.

461. On information and belief, Azure is implemented in a manner that meets each and every limitation of claim 1 of the '599 patent.

1. A system including processors and comprising a plurality of subsystems, implemented on hardware logic and/or software logic executing on the processors, for dynamic resource management of a pool of processing resources on behalf of application programs, the system comprising:

462. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," Microsoft Azure comprises a system including processors and comprising a plurality of subsystems, implemented on hardware logic and/or software logic

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

executing on the processors, for dynamic resource management of a pool of processing resources on behalf of application programs.

463. Microsoft Azure is configured to, and Microsoft uses Azure to, manage and host a number of applications across a pool of processing nodes. The Azure RM (in coordination with the SMs and FMs) is used to manage the assignment of computing cores across multiple applications and multiple application tasks running, for instance, on multiple computing cores such as CPUs, GPUs, and/or FPGA processors.

> ***a first subsystem configured to periodically allocate a plurality of processing units of the pool of processing resources among a plurality of application programs over time, wherein***
>
> > ***the plurality of processing units comprises units of at least two types of processing units and/or units of a reconfigurable type, and***
> >
> > ***allocating the plurality of processing units among the plurality of application programs is based at least in part on i) a respective processing demand of each of the plurality of application programs, and ii) a respective processing resource quota of each program of a subset of the plurality of application programs;***

464. Microsoft Azure includes a first subsystem configured to periodically allocate a plurality of processing units of the pool of processing resources among a plurality of application programs over time, wherein the plurality of processing units comprises units of at least two types of processing units and/or units of a reconfigurable type, and allocating the plurality of processing units among the plurality of application programs is based at least in part on i) a respective processing demand of each of the plurality of application programs, and ii) a respective processing resource quota of each program of a subset of the plurality of application programs.

465. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," the RM in coordination with the SMs and FMs, which work in conjunction with the AC, periodically reallocates processing units (*e.g.*, CPUs, reconfigurable FPGA cores, and/or other core types) among various application programs (*e.g.*, web search

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

ranking, DNN, SQL, SDN offload) based on the processing demand of each application.

466. Microsoft Azure also supports subscribing for assured resources such as FPGA Groupings for an additional fee. In such scenarios, the RM in coordination with the SMs and FMs, which work in conjunction with the AC, allocates processors based in part on the number of FPGA Groupings subscribed by the client for its application.

a second subsystem configured to, for each program of the plurality of application programs, select a set of highest priority instances of one or more instances of a respective program, wherein

a number of instances in the set of instances corresponds to a number of processing units allocated to the respective application program by the first subsystem during a current allocation period, and each instance of the set of highest priority instances is selected based at least in part on a) the number of processing units allocated to the respective program and b) a relative readiness for execution among the one or more instances of the respective application program; and

467. Microsoft Azure includes a second subsystem configured to, for each program of the plurality of application programs, select a set of highest priority instances of one or more instances of a respective program, wherein a number of instances in the set of instances corresponds to a number of processing units allocated to the respective application program by the first subsystem for a current allocation period, and each instance of the set of highest priority instances is selected based at least in part on a) the number of processing units allocated to the respective program and b) a relative readiness for execution among the one or more instances of the respective application program.

468. As discussed above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," Azure schedules tasks based on their relative readiness for execution. Azure prioritizes tasks based on, *e.g.*, whether a task is awaiting input data and whether data sufficient to execute the task exists in the queue.

469. As explained above in the section entitled "Microsoft's Infringing Cloud

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

Computing Architecture," the AC working in conjunction with the RM, SMs, and FMs selects tasks to be assigned to, and executed by, each FPGA or other processor core in a group. The number and selection of the respective tasks is a function of at least the number of processor cores in the group and the respective readiness of the tasks for execution.

> ***a third subsystem configured to assign, for each of the plurality of application programs, the respective set of highest priority instances to a subset of the processing units allocated by the first subsystem for the respective application program for execution during an upcoming allocation period, wherein, for at least a portion of instances of at least a portion of the plurality of application programs,***
>
> > ***a respective instance of the respective set of highest priority instances is associated with a processing unit type or configuration, wherein assigning comprises placing the respective instance to a respective processing unit of the plurality of processing units based at least in part on the processing unit type or configuration associated with the respective instance, and prioritizing placement of the respective instance to the particular type or configuration processing unit demanded;***

470. Microsoft Azure includes a third subsystem configured to assign, for each of the plurality of application programs, the respective set of highest priority instances to a subset of the processing units allocated by the first subsystem for the respective application program for execution during an upcoming allocation period, wherein, for at least a portion of instances of at least a portion of the plurality of application programs, a respective instance of the respective set of highest priority instances is associated with a processing unit type or configuration, wherein assigning comprises placing the respective instance to a respective processing unit of the plurality of processing units based at least in part on the processing unit type or configuration associated with the respective instance, and prioritizing placement of the respective instance to the particular type or configuration processing unit demanded.

471. The SMs in coordination with the FMs and RM, which work in conjunction with the AC, assign the selected tasks for execution on the individual FPGA or other processor cores in

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

the group.

472. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," the FPGA cores in each group are configured or reconfigured to optimally perform a certain task such as encryption or performing a mathematical algorithm associated with web search ranking. In the use cases, illustrated in the materials discussed above, every task is assigned to a FPGA core specially configured for the task.

wherein the system is further configured to periodically adjust allocations of the plurality of processing units of the pool of processing resources over time, wherein, for each periodic adjustment, the adjusting is followed by the selecting and the assigning.

473. Microsoft Azure is further configured to periodically adjust allocations of the plurality of processing units of the pool of processing resources over time, wherein, for each periodic adjustment, the adjusting is followed by the selecting and the assigning.

474. As discussed above, the RM in coordination with the SMs and FMs, allocates a number of FPGA cores among the applications and periodically adjusts that allocation while the SMs (which work in conjunction with the AC, RM, and FMs), select tasks for execution by the appropriately configured FPGA cores.

475. Microsoft has had knowledge of and notice of the chain of applications underlying the '599 patent since at least January 2018 when ThroughPuter brought the '090 and '833 patents as well as the portfolio, including a number of pending patent applications, to the attention of Microsoft in the January 2018 letter.

476. Microsoft has had knowledge of and notice of the application that led to the issuance of the '599 patent since September 2018 when ThroughPuter brought the application to the attention of Microsoft in the September 2018 letter.

477. On information and belief, in accordance with 35 U.S.C. § 287, Microsoft has had

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

actual notice and knowledge of the '599 patent no later than its issuance.

478. On information and belief, Microsoft continues without license to make, use, import, market, offer for sale, and/or sell in the United States services or products that infringe the '599 patent.

479. Microsoft has directly infringed and continues to directly infringe the '599 patent by engaging in acts constituting infringement under 35 U.S.C. § 271(a), including but not necessarily limited to one or more of making, using, selling and offering to sell, in this District and elsewhere in the United States, and importing into the United States, the Microsoft Azure platform or components and services thereof.

480. Microsoft's continuing infringement of the '599 patent is knowing, intentional, and willful.

481. Microsoft's infringement of the '599 patent has injured ThroughPuter in its business and property rights. Microsoft's continuing infringement of the '599 patent is deliberate and willful and constitutes egregious misconduct. In developing and offering its products and services, Microsoft is, at the very least, willfully blind to this ongoing infringement.

482. ThroughPuter is entitled to recover monetary damages for the injuries arising from Microsoft's willful infringement pursuant to 35 U.S.C. § 284 in an amount to be determined at trial. Microsoft's infringement of the '599 patent has caused irreparable harm to ThroughPuter and will continue to cause such harm unless and until Microsoft's infringing activities are enjoined by this Court.

483. Microsoft's infringement of the '599 patent is exceptional and entitles ThroughPuter to attorneys' fees and costs incurred in prosecuting this action under 35 U.S.C. § 285.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

COUNT 10

(Infringement of U.S. Patent No. 9,632,833)

484. ThroughPuter repeats and realleges each and every allegation contained above as though fully set forth herein.

485. On April 25, 2017, the USPTO duly and legally issued the '833 patent, entitled "Scheduling Application Instances to Processor Cores Over Consecutive Allocation Periods Based on Application Requirements." A copy of the '833 patent is attached as Exhibit 10.

486. Mark Sandstrom is the sole and true inventor of the '833 patent.

487. ThroughPuter, Inc. owns all right, title, and interest to and in the '833 patent.

488. Microsoft infringes at least claims 10, 13-15, 17-19, 21-22, 24-25, 27, 29, 32-34, and 37 of the '833 patent.

489. Claim 34 of the '833 patent is representative of the claims infringed by Microsoft and recites:

> 34. A system for assigning instances of software programs to an array of processor cores comprising:
>
> a first hardware or software logic subsystem configured to, for each of a series of successive core allocation periods (CAPs), select, from a group of executable instances of a set of software programs, a subset of the executable instances, referred to as selected instances, for execution on the cores of the array for an upcoming CAP, wherein the selection of the selected instances is based, at least in part, on a respective capacity demand indication of each of the set of software programs, with said indication of a given program (a) being based at least in part on a number of its executable instances that presently have input data available for processing and (b) indicating processor core types demanded by its executable instances;
>
> a second hardware or software logic subsystem configured to assign each of the selected instances for execution on a processor core in the array of processor cores based, at least in part, on matching the respective demanded processor core types associated with the selected instances with types of processor cores available for assignment; and

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

a third subsystem, comprising the array of processor cores, configured to execute the selected instances on their assigned cores over the next CAP, at least in part, to process the input data.

490. On information and belief, Azure is implemented in a manner that meets each and every limitation of claim 34 of the '833 patent.

34. A system for assigning instances of software programs to an array of processor cores comprising:

491. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," Microsoft Azure comprises a system for assigning instances of software programs to an array of processor cores.

492. Microsoft Azure is configured to, and Microsoft uses Azure to assign instances of software programs to an array of processor cores.

a first hardware or software logic subsystem configured to, for each of a series of successive core allocation periods (CAPs), select, from a group of executable instances of a set of software programs, a subset of the executable instances, referred to as selected instances, for execution on the cores of the array for an upcoming CAP, wherein the selection of the selected instances is based, at least in part, on a respective capacity demand indication of each of the set of software programs, with said indication of a given program (a) being based at least in part on a number of its executable instances that presently have input data available for processing and (b) indicating processor core types demanded by its executable instances;

493. Microsoft Azure includes a first hardware or software logic subsystem configured to, for each of a series of successive core allocation periods (CAPs), select, from a group of executable instances of a set of software programs, a subset of the executable instances, referred to as selected instances, for execution on the cores of the array for an upcoming CAP, wherein the selection of the selected instances is based, at least in part, on a respective capacity demand indication of each of the set of software programs, with said indication of a given program (a) being based at least in part on a number of its executable instances that presently have input data available for processing and (b) indicating processor core types demanded by its executable

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

instances.

494. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," the AC, which works in conjunction with the RM, SMs, and FMs, selects a subset of available executable instances for execution on the FPGA processors. The executable instances are selected based on demand indications of respective software programs with such indications being based in part on the number of instances of the respective program ready for execution and the type of processor core needed for a given instance. For example, the demand indication of a speech translation software will depend on how many tasks can be executed concurrently and the type of processor core(s) that are best-suited to execute these tasks.

a second hardware or software logic subsystem configured to assign each of the selected instances for execution on a processor core in the array of processor cores based, at least in part, on matching the respective demanded processor core types associated with the selected instances with types of processor cores available for assignment; and

495. Microsoft Azure includes a second hardware or software logic subsystem configured to assign each of the selected instances for execution on a processor core in the array of processor cores based, at least in part, on matching the respective demanded processor core types associated with the selected instances with types of processor cores available for assignment.

496. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," the SMs in coordination with the RM and FMs, which work in conjunction with the AC, assign the selected instances for execution on an FPGA or FPGA Groupings at least in part by matching the selected instance with a processor core suited to performing the task.

a third subsystem, comprising the array of processor cores, configured to execute the selected instances on their assigned cores over the next CAP, at least in part, to process the input data.

497. Microsoft Azure includes a third subsystem, comprising the array of processor

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

cores, configured to execute the selected instances on their assigned cores over the next CAP, at least in part, to process the input data.

498. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," Azure includes the Azure Fabric, which includes an array of FPGA cores that execute selected instances to process data received from the queues.

499. In accordance with 35 U.S.C. § 287, Microsoft has had actual notice and knowledge of the '833 patent no later than the time when ThroughPuter informed Microsoft of its existence as set forth, *supra*.

500. On information and belief, Microsoft continues without license to make, use, import, market, offer for sale, and/or sell in the United States services or products that infringe the '833 patent.

501. Microsoft has directly infringed and continues to directly infringe the '833 patent by engaging in acts constituting infringement under 35 U.S.C. § 271(a), including but not necessarily limited to one or more of making, using, selling and offering to sell, in this District and elsewhere in the United States, and importing into the United States, the Microsoft Azure platform or components and services thereof.

502. Microsoft's infringement of the '833 patent has been, and continues to be knowing, intentional, and willful. Microsoft has had knowledge of and notice of the application underlying the '833 patent since at least May 2015 and despite this knowledge continues to commit the aforementioned infringing acts. For at least the reasons stated in this paragraph and above, this infringement has been willful.

503. Microsoft's infringement of the '833 patent has injured ThroughPuter in its business and property rights. Microsoft's infringement of the '833 patent has been and is deliberate

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

and willful and constitutes egregious misconduct. Despite actual knowledge of the '833 patent and numerous related patents and applications since at least January 2018, Microsoft continued to develop and offer its infringing products and services. In developing and offering its products and services, Microsoft has been willfully blind to this ongoing infringement.

504. ThroughPuter is entitled to recover monetary damages for the injuries arising from Microsoft's willful infringement pursuant to 35 U.S.C. § 284 in an amount to be determined at trial. Microsoft's infringement of the '833 patent has caused irreparable harm to ThroughPuter and will continue to cause such harm unless and until Microsoft's infringing activities are enjoined by this Court.

505. Microsoft's infringement of the '833 patent is exceptional and entitles ThroughPuter to attorneys' fees and costs incurred in prosecuting this action under 35 U.S.C. § 285.

COUNT 11

(Infringement of U.S. Patent No. 9,424,090)

506. ThroughPuter repeats and realleges each and every allegation contained above as though fully set forth herein.

507. On August 23, 2016, the USPTO duly and legally issued the '090 patent, entitled "Scheduling Tasks to Configurable Processing Cores Based on Task Requirements and Specification." A copy of the '090 patent is attached as Exhibit 11.

508. Mark Sandstrom is the sole and true inventor of the '090 patent.

509. ThroughPuter, Inc. owns all right, title, and interest to and in the '090 patent.

510. Microsoft infringes at least claims 1-7 and 10-15 of the '090 patent.

511. Claim 1 of the '090 patent is representative of the claims infringed by Microsoft

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

and recites:

> 1. A method for assigning a set of processing tasks to an array of processor cores of configurable types, the method comprising:
>
> executing time variable subsets of the processing tasks of differing types on their assigned processor cores of matching types, wherein the matching type for the assigned processor core for a given processing task of the set corresponds to a type of a processor core demanded by the given processing task;
>
> for each of a series of core allocation periods (CAPs), selecting, from the set of processing tasks, specific tasks, referred to as selected tasks, for execution on the processor cores for a next CAP at least in part based on core capacity demand expressions associated with the processing tasks;
>
> assigning the selected tasks for execution on the processor cores for the next CAP in a manner to maximize, within the array, a number of processor cores whose assigned processing tasks for a present and the next CAP demands the same type of processor core; and
>
> configuring the array such that the type of any given processor core in the array matches the type of processing task assigned for execution on the given processor core for the next CAP.

512. On information and belief, Azure is implemented in a manner that meets each and every limitation of claim 1 of the '090 patent.

> ***1. A method for assigning a set of processing tasks to an array of processor cores of configurable types, the method comprising:***

513. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," Microsoft Azure assigns a set of processing tasks to an array of processor cores of configurable types.

> ***executing time variable subsets of the processing tasks of differing types on their assigned processor cores of matching types, wherein the matching type for the assigned processor core for a given processing task of the set corresponds to a type of a processor core demanded by the given processing task;***

514. Microsoft Azure executes time variable subsets of the processing tasks of differing types on their assigned processor cores of matching types, wherein the matching type for the assigned processor core for a given processing task of the set corresponds to a type of a processor core demanded by the given processing task.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

515. As explained above in the section entitled "Microsoft's Infringing Cloud Computing Architecture," the FPGA accelerator core blocks in each group are repeatedly configured or reconfigured to optimally perform tasks, of varying duration, such as encryption or performing a mathematical algorithm associated with web search ranking. In the use cases, illustrated in the materials discussed above, a given task is assigned to an FPGA accelerator configured for optimal execution of the given the task.

for each of a series of core allocation periods (CAPs), selecting, from the set of processing tasks, specific tasks, referred to as selected tasks, for execution on the processor cores for a next CAP at least in part based on core capacity demand expressions associated with the processing tasks;

516. For each of a series of core allocation periods (CAPs), Microsoft Azure selects, from the set of processing tasks, specific tasks for execution on the processor cores for a next CAP at least in part based on core capacity demand expressions associated with the processing tasks.

517. As discussed above, in Azure, the number of FPGA processor core blocks allocated to each application changes based on the demand expressions as observed by Azure's resource management systems. In the illustrative example depicted above, the Ranking Service program is allocated twice the number of cores that are assigned to the ADLA program due to the higher processing needs of the former.

assigning the selected tasks for execution on the processor cores for the next CAP in a manner to maximize, within the array, a number of processor cores whose assigned processing tasks for a present and the next CAP demands the same type of processor core; and

518. Microsoft Azure assigns the selected tasks for execution on the processor cores for the next CAP in a manner to maximize, within the array, a number of processor cores whose assigned processing tasks for a present and the next CAP demands the same type of processor core.

519. As discussed above, the SMs in coordination with the RM and FMs, which work in conjunction with the AC, assigns the selected tasks to FPGA processors for execution, as

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

illustrated in Microsoft's '709 patent, such that the FPGA cores are not reassigned between tasks demanding different core types if doing so is not necessary. In Azure, FPGA core assignments, at least across tasks demanding distinct core types, are held constant from one cycle to the next to the extent permitted by the variations in selections of tasks being assigned for execution.

configuring the array such that the type of any given processor core in the array matches the type of processing task assigned for execution on the given processor core for the next CAP.

520. Microsoft Azure configures the FPGA core array such that the type of any given processor core in the array matches the type of processing task assigned for execution on the given processor core for the next CAP. Azure's SMs in coordination with the FMs and RM, which work in conjunction with the AC, assign the selected tasks for execution on appropriate individual FPGA cores in the group.

521. As discussed above, the FPGA cores in each group are configured or reconfigured to optimally perform a certain task such as encryption or performing a mathematical algorithm associated with web search ranking. In the use cases, illustrated in the materials discussed above, every task is assigned to an FPGA core specially configured for the task.

522. In accordance with 35 U.S.C. § 287, Microsoft has had actual notice and knowledge of the '090 patent no later than the time when ThroughPuter informed Microsoft of its existence as set forth, *supra*.

523. On information and belief, Microsoft continues without license to make, use, import, market, offer for sale, and/or sell in the United States services or products that infringe the '090 patent.

524. Microsoft has directly infringed and continues to directly infringe the '090 patent by engaging in acts constituting infringement under 35 U.S.C. § 271(a), including but not necessarily limited to one or more of making, using, selling and offering to sell, in this District and

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

elsewhere in the United States, and importing into the United States, the Microsoft Azure platform or components and services thereof.

525. Microsoft's infringement of the '090 patent has been, and continues to be knowing, intentional, and willful. Microsoft has had knowledge of and notice of the application underlying the '090 patent since at least May 2015 and despite this knowledge continues to commit the aforementioned infringing acts. For at least the reasons stated in this paragraph and above, this infringement has been willful.

526. Microsoft's infringement of the '090 patent has injured ThroughPuter in its business and property rights. Microsoft's infringement of the '090 patent has been and is deliberate and willful and constitutes egregious misconduct. Despite actual knowledge of the '090 patent and numerous related patents and applications since at least January 2018, Microsoft continued to develop and offer its infringing products and services. In developing and offering its products and services, Microsoft has been willfully blind to this ongoing infringement.

527. ThroughPuter is entitled to recover monetary damages for the injuries arising from Microsoft's willful infringement pursuant to 35 U.S.C. § 284 in an amount to be determined at trial. Microsoft's infringement of the '090 patent has caused irreparable harm to ThroughPuter and will continue to cause such harm unless and until Microsoft's infringing activities are enjoined by this Court.

528. Microsoft's infringement of the '090 patent is exceptional and entitles ThroughPuter to attorneys' fees and costs incurred in prosecuting this action under 35 U.S.C. § 285.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

PRAYER FOR RELIEF

WHEREFORE, ThroughPuter respectfully requests that the Court enter judgment against Microsoft as follows:

A. An adjudication that Microsoft has infringed one or more claims of the ’948, ’556, ’306, ’599, ’902, ’242, ’644, ’998, ’353, ’090, and ’833 patents;

B. An order permanently enjoining Microsoft from further infringement of the ’948,’556, ’306, ’599, ’902, ’242, ’644, ’998, ’353, ’090, and ’833 patents;

C. An award of damages pursuant to 35 U.S.C. § 284;

D. An order that the damages award be increased up to three times the actual amount assessed, pursuant to 35 U.S.C. § 284;

E. An award to ThroughPuter of its costs, pre- and post-judgment interest, and reasonable expenses to the fullest extent permitted by law;

F. A declaration that this case is exceptional pursuant to 35 U.S.C. § 285, and an award of attorneys’ fees and costs; and

G. An award to ThroughPuter of such other and further relief as this Court deems just and proper.

DEMAND FOR JURY TRIAL

Pursuant to Rule 38(b) of the Federal Rules of Civil Procedure, ThroughPuter hereby demands a trial by jury on all issues so triable.

LOWE GRAHAM JONES PLLC
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
206.381.3300 • F: 206.381.3301

DATED this 3rd day of June, 2022.

LOWE GRAHAM JONES PLLC

s/Lawrence D. Graham
Lawrence D. Graham, WSBA No. 25402
Graham@LoweGrahamJones.com
1325 Fourth Avenue, Suite 1130
Seattle, Washington 98101
T: 206.381.3300
F: 206.381.3301

Gardella Grace P.A.

W. Cook Alciati (admitted *pro hac vice*)
80 M Street SE, 1st Floor
Washington D.C. 20003
T: 703-721-8379
calciati@gardellagrace.com

Michael Dorfman (admitted *pro hac vice*)
2502 North Clark Street, Suite 222
Chicago, Illinois 60614
T: 703-415-9344
mdorfman@gardellagrace.com

Attorneys for Plaintiff ThroughPuter, Inc.