

UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE PATENT TRIAL AND APPEAL BOARD

---

MICROSOFT CORPORATION,  
Petitioner,

v.

PARTEC AG,  
Patent Owner.

---

Case IPR2025-00318  
Patent 11,537,442

**DECLARATION OF DR. DAVID KAEI IN SUPPORT OF PATENT  
OWNER'S RESPONSE**

## TABLE OF CONTENTS

|       |                                                                                                                   |    |
|-------|-------------------------------------------------------------------------------------------------------------------|----|
| I.    | INTRODUCTION .....                                                                                                | 1  |
| II.   | PROFESSIONAL BACKGROUND AND QUALIFICATIONS.....                                                                   | 1  |
| III.  | MATERIALS REVIEWED.....                                                                                           | 4  |
| IV.   | LEGAL PRINCIPLES .....                                                                                            | 4  |
| A.    | Claim Construction.....                                                                                           | 5  |
| B.    | Anticipation .....                                                                                                | 6  |
| C.    | Obviousness.....                                                                                                  | 6  |
| D.    | Person of Ordinary Skill in the Art .....                                                                         | 10 |
| E.    | Enablement .....                                                                                                  | 10 |
| V.    | STATE OF THE ART .....                                                                                            | 11 |
| VI.   | THE 442 PATENT.....                                                                                               | 13 |
| VII.  | LEVEL OF ORDINARY SKILL IN THE ART .....                                                                          | 18 |
| VIII. | CLAIM CONSTRUCTION.....                                                                                           | 19 |
| IX.   | THE TEACHINGS OF THE REFERENCES .....                                                                             | 19 |
| A.    | Lippert .....                                                                                                     | 20 |
| B.    | Budenske .....                                                                                                    | 21 |
| C.    | Kambatla .....                                                                                                    | 23 |
| X.    | THE ASSERTED GROUNDS .....                                                                                        | 26 |
| A.    | Ground 1 .....                                                                                                    | 26 |
| 1.    | Claim 1 .....                                                                                                     | 26 |
| B.    | Claim 3 .....                                                                                                     | 40 |
| 1.    | Dynamically Alter Further Distribution .....                                                                      | 40 |
| C.    | Claim 5 .....                                                                                                     | 41 |
| 1.    | Computation Nodes Daemons and Booster Node Daemons .....                                                          | 41 |
| D.    | Claim 9 .....                                                                                                     | 43 |
| 1.    | Distribution of Sub-Tasks To Both Computation Nodes and Booster<br>Nodes in Each Claimed Computing Iteration..... | 43 |
| 2.    | Computation Node Daemons and Booster Node Daemons .....                                                           | 44 |

E. Ground 2.....45

1. Claim 3.....45

2. Claim 5.....46

3. Claim 9.....47

F. A POSITA would not have modified the references in the claimed  
manner47

XI. MISCELLANEOUS .....51

## **I. INTRODUCTION**

1. My name is David Kaeli. I am over eighteen years of age, of sound mind, and qualified to make the statements set forth in this Declaration. With the exception of the legal principles counsel has supplied to me, all the facts and statements contained herein are within my personal knowledge and are in all respects true and correct.

2. I have been asked by ParTec AG to submit this declaration in response to a petition (Paper 1, the “Petition”) filed by Microsoft Corporation and a declaration submitted by Darrel Long (EX1003, the “Long Declaration”) regarding U.S. Patent No. 11,537,442 (the “442 patent”).

## **II. PROFESSIONAL BACKGROUND AND QUALIFICATIONS**

3. I have summarized in this section some of my educational background, career history, publications, and other relevant qualifications. My full *curriculum vitae* is attached as Exhibit 2010 to this expert report. The list of cases that I have testified in over the past 5 years is attached as Exhibit 2011.

4. I am currently a Distinguished Full Professor of Electrical and Computer Engineering (ECE) at Northeastern University in Boston, Massachusetts. I have also served as the Associate Dean of Undergraduate Programs in the College of Engineering at Northeastern University. I have been on the faculty of the ECE Department since September 1993.

5. Prior to my thirty-one years on the faculty of the ECE Department, I spent twelve years working for IBM. From 1981-1986, I worked on the design and delivery of the IBM System/370 3090 mainframe system. I worked on the design and verification of the memory components of the system and later worked on the recovery design and reliability capabilities of the system.

6. From 1986-1993, I worked at the IBM T.J. Watson Research Center. During this time, I worked on the design and analysis of IBM Personal Systems 2 computers and associated software. I also worked on the design of a new Application System/400 multi-processor system. One part of my work on that system was building a memory tracing facility that interfaced with the memory bus and selectively acquired signals from the system bus for the purpose of memory address tracing.

7. I have also conducted research on and written about numerous aspects of computer architecture and high performance computing, including accelerators, graphics processing units, memory systems, and parallel/distributed computing systems, both while employed by IBM and as a faculty member at Northeastern University. I am presently the Director of the Northeastern University Computer Architecture Laboratory. My laboratory has been named a Center of Excellence in GPU Computing by industry. I have served as the co-Director of the Northeastern University Institute for Information Assurance and have pursued an active research

program in computer security and computer architecture fields for over thirty years.

I have published over 450 refereed journal and conference papers in the areas of computer architecture and computer security. I have also supervised over fifty Ph.D. theses in these areas.

8. I have worked on optimizing the performance of heterogeneous computing systems since I joined Northeastern University. From 1995-2010, I worked with Analog Devices on the SHARC and BlackFin DSP platforms. I have worked with NVIDIA and AMD on their GPUs for the past 20 years. I feel I am well positioned to evaluate the state-of-the-art in heterogeneous systems over the past 30 plus years.

9. I am a Fellow of the Association of Computing Machinery (ACM) and a Fellow of the Institute of Electrical and Electronic Engineers (IEEE) and a member of the Computer Society. I have served as chair of the Executive Committee of the IEEE Technical Committee on Computer Architecture. I am presently serving my second term as the Editor-in-Chief of ACM Transactions on Architecture and Code Optimization. I am also on the editorial board of the Elsevier Journal on Future Generation Computer Systems.

10. From 2005-2011 I served as a consultant for Akorri Networks, during which time my work resulted in at least three issued patents. In 2000, I served as a consultant for InCert Corporation. From 2011 to the present, I have served as Chief

Science Advisor of Apposable Software. From 2007-2011, I served as Chief Technology Officer of NUIC Technologies, where I worked on secure computing platforms.

11. Prior to my industry and academic careers, I obtained a Bachelor of Science degree in Electrical Engineering from Rutgers University, a Master of Science degree in Computer Engineering from Syracuse University, and a Ph.D. in Electrical Engineering from Rutgers University.

12. I have provided written expert opinions in fifteen prior patent cases in the areas of computer architecture, high performance computing, security and computer design.

### **III. MATERIALS REVIEWED**

13. In forming my opinions in this proceeding, I have reviewed each of the references cited in this Declaration including the Petition, the Long Declaration, and the references that the Petition and Long Declaration rely upon as invalidating prior art.

### **IV. LEGAL PRINCIPLES**

14. I am not a lawyer. In forming my opinions as set forth in this Declaration, I have relied upon legal principles supplied to me by attorneys for ParTec. In forming my opinions, I have not relied upon ParTec's attorneys for any purpose other than the following legal principles.

**A. Claim Construction**

15. I understand that determining whether a patent claim is valid requires a two-step analysis. First, the claim must be construed. Second, the properly construed claim must be compared to the prior art.

16. With respect to the first step, I understand that claims are construed from the perspective of a person of ordinary skill in the art (“POSITA”) at the time of the purported invention. I understand that in the context of an *inter partes* review proceeding, absent a disclaimer or express definition, claim terms are given their plain and ordinary meaning to a POSITA in view of the specification and file history and the knowledge of a POSITA. Unless otherwise noted, I have applied what I consider to be the plain and ordinary meaning of all terms as understood by a POSITA at the time of the invention, which I understand to be no later than January 23, 2018. I note that my opinions would not change if the time of invention was January 23, 2019.

17. I understand that a preamble is an introductory phrase of a claim. In general, a preamble limits the invention if it recites essential structure or steps, or if it is necessary to give life, meaning, and vitality to the claim. Conversely, a preamble is not limiting where a patentee defines a structurally complete invention in the claim body and uses the preamble only to state a purpose or intended use for the invention.

**B. Anticipation**

18. I understand that, although not expressly disclosed, subject matter may be inherently disclosed in a prior art reference where that subject matter is necessarily present in the subject matter disclosed and would be understood to be so by those of ordinary skill in the art. I understand that the fact that a certain result or characteristic may occur or be present in the prior art is not sufficient to establish the inherency of that result or characteristic.

19. I understand that a claim is invalid if each and every element of the claim is disclosed in a single prior art reference with the elements arranged as specified in the claim, either expressly or inherently. I understand that invalidity based on the disclosure of a single prior art reference in this manner is called anticipation.

**C. Obviousness**

20. I understand that a claim would have been obvious under 35 U.S.C. § 103 if one or more prior art references disclose, expressly or inherently, every claim limitation in such a way so as to render the claim, as a whole, obvious to a person of ordinary skill in the art at the time the purported invention was made. The relevant standard for obviousness is as follows:

A patent for a claimed invention may not be obtained, notwithstanding that the claimed invention is not identically disclosed as set forth in section 102, if the differences between the claimed invention and the

prior art are such that the claimed invention as a whole would have been obvious before the effective filing date of the claimed invention to a person having ordinary skill in the art to which the claimed invention pertains. Patentability shall not be negated by the manner in which the invention was made.

35 U.S.C. § 103 (AIA).

21. In determining whether or not a patented invention would have been obvious, the following factors should be considered: (a) the scope and content of the prior art; (b) the differences between the prior art and the claims at issue; (c) the level of ordinary skill in the art; and (d) whatever “secondary considerations” may be present.

22. I understand that certain “secondary considerations” may be relevant in determining whether or not an invention would have been obvious, and that these secondary considerations may include commercial success of a product using the invention, if that commercial success is due to the invention; long-felt but unsolved need for the invention; evidence of copying of the claimed invention by others; industry acceptance; initial skepticism of the invention; failure of others to solve the problem; praise of the invention by others; the taking of licenses under the patents by others; and simultaneous invention by others (which may be an indication that the claimed invention was the result of ordinary skill in the art).

23. I understand that a patent composed of several elements is not proved obvious merely by demonstrating that each of its elements was, independently, known in the prior art. While multiple prior art references or elements may, in some circumstances, be combined to render a patent claim obvious, I understand that I should consider whether there is an apparent reason to combine the prior art references or elements in the way the patent claims. To determine whether such an apparent reason exists to combine the prior art references or elements in the way a patent claims, it will often be necessary to look to the interrelated teaching of multiple patents, to the effects of demands known to the design community or present in the marketplace, and to the background knowledge possessed by a person having ordinary skill in the art.

24. I also understand that when the prior art teaches away from combining prior art references or certain known elements, discovery of a successful means of combining them is more likely to be non-obvious. A prior art reference may be said to teach away from a patent when a person of ordinary skill, upon reading the reference, would be discouraged from following the path set out in the patent or would be led in a direction divergent from the path that was taken by the patent.

25. I also understand that a combination of prior art references may not be obvious if there would not have been a reasonable expectation of success in

combining or modifying the prior art disclosures to meet the limitations of the claimed invention.

26. I also understand that it is not permissible to use hindsight in assessing whether a claimed invention is obvious. Rather, I understand that, to assess obviousness, you must place yourself in the shoes of a person having ordinary skill in the relevant field of technology at the time the inventions were made who is trying to address the issues or solve the problems faced by the inventor and ignore the knowledge you currently now have of the inventions.

27. I also understand that there are numerous ways in which to articulate the legal standard for obviousness, including: (1) combining prior art elements according to known methods to yield predictable results, (2) simple substitution of one known element for another to obtain predictable results, (3) use of a known technique to improve similar devices (methods, or products) in the same way, (4) applying a known technique to a known device, method, or product ready for improvement to yield predictable results, (5) choosing from a finite number of identified, predictable solutions with a reasonable expectation of success, (6) known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations are predictable to one of ordinary skill in the art, and (7) some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to

modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention.

**D. Person of Ordinary Skill in the Art**

28. I understand that the meaning of claim terms and a patent's validity are to be determined from the vantage point of one of ordinary skill in the art at the time of the invention. In determining who would be one of such ordinary skill, I understand it is appropriate to consider criteria such as: (a) the type of problems encountered in the art; (b) prior art solutions to those problems; (c) the rapidity with which innovations are made; (d) the sophistication of the technology; and (e) the education level of active workers in the field.

**E. Enablement**

29. I understand that a prior art disclosure must be an enabling disclosure in order to invalidate a challenged claim. I understand that a prior art reference is assumed to be enabling unless it appears to not be enabling on its face.

30. I understand that enablement relates to whether a disclosure describes an invention in such terms that one skilled in the art can make and use the invention. The standard for determining whether a disclosure is enabling is whether the experimentation needed to practice the invention based on that disclosure is undue or unreasonable. What is reasonable will depend on the nature of the invention and

the underlying art. I understand that several factors are often applied to determine whether experimentation is reasonable or undue, including:

- (A) The breadth of the claims;
- (B) The nature of the invention;
- (C) The state of the prior art;
- (D) The level of one of ordinary skill;
- (E) The level of predictability in the art;
- (F) The amount of direction provided by the inventor;
- (G) The existence of working examples; and
- (H) The quantity of experimentation needed to make or use the invention based on the content of the disclosure.

## **V. STATE OF THE ART**

31. If we take the state-of-the-art in cluster computing in the 1990's (e.g., the listed timeframe of Budenske), and compare it with advances in the 2018 timeframe (e.g., around the priority date of the 442 patent), we find major differences. In the 1990's, heterogeneous systems were just beginning to be discussed in the form of basic system integration of special purpose processors such as digital signal processors into computing systems.

32. If we analyze Ex. 1005 Budenske in light of the era it came from, we see the limits of its applicability. Budenske is focused on selecting a pre-computed

task mapping based on the content of the input data (e.g., the number of objects being tracked in the input image, Ex. 1005 at 393) – this seems appropriate, given the limited memory space and processor performance of heterogeneous systems in the 1990's. The task mappings were precomputed offline, and later used based on changes in the data input to the application. Computing tasks offline was a stopgap approach to getting around hardware limits of the time that provided a solution that the Budenske authors determined to be good enough for their limited theoretical applications. For example, Budenske at 394 discusses the likely necessity of pruning out parameter combinations that are less likely to appear to accommodate memory limitations, and 395 and 401 explain some of the tradeoffs that would be necessary to utilize this stopgap approach.

33. Another characteristic of heterogeneous systems during the time period of Budenske was that they were customized for a particular task (e.g., automatic target recognition). EX1005 at 387. The capabilities of the boosters was quite limited in this timeframe, which led to the need to focus on a particular limited task (and corresponding particular specialized processor). This is quite different than heterogeneous systems at the time of the 442 patent, which were expected to provide far more flexibility and be adaptable to many different computation tasks. This can be seen in the 442 patent disclosure, which is focused on distributing subtasks between different processor types dynamically, using information related to the

processing of the subtasks to redistribute the subtasks through multiple computing iterations. As a result, the 442 patent teaches us methods on how to best utilize heterogeneous systems around the 2018 timeframe.

## **VI. THE 442 PATENT**

34. I understand that the PCT application for the 442 patent was filed on January 23, 2019, and that it claims priority to a January 23, 2018 foreign patent application. The title of the 442 patent is “Application Runtime Determined Dynamical Allocation of Heterogeneous Compute Resources.” The “Field of the Disclosure” section states that the invention “relates to a mechanism for executing computation tasks within a computation environment, in particular a heterogeneous computation environment adapted for parallel processing of a computation task.” 442 patent at 1:14-19.

35. The 442 patent provides solutions relevant to a type of cluster computing systems called heterogeneous computing systems. Initially, homogenous computing clusters were constructed using large numbers of general purpose processors. EX2012 at 1. However, general purpose processors have drawbacks, including high power consumption and cost. EX2012 at 1. The fact that the processors were general purpose meant that there would be many tasks for which they were neither the fastest nor most efficient choice. General purpose CPUs are optimized to benefit from single-thread performance, devoting a significant percent

of die space to large caches and branch prediction hardware. In contrast, many accelerators (e.g., GPUs) are optimized for executing a massive number of threads, with a majority of die space devoted to a large number (thousands) of computing pipelines (over 4000 pipelines in 2018). EX2013. These booster devices (such as graphics processing units, or GPUs) can perform certain tasks far faster or more efficiently than a general-purpose CPU. This is especially true for today's demanding applications that perform the same operation on compound datatypes, such as matrices and vectors. Thus, certain tasks that are far more efficiently executed on a booster can be assigned to the booster rather than the CPU.

36. The 442 patent recognized that the types of tasks for which modern cluster computers are used are sometimes not so straightforward that they should simply be assigned solely to a single processor type. Computing jobs often comprise a number of tasks that may be repeated iteratively during execution of the job. EX1001 at 3:44-4:3. The 442 patent gives a simple example of Monte Carlo simulation which simulates an outcome using repeated random sampling. However, a POSITA would have understood that the disclosure could apply equally to a more complicated job, such as Deep Neural Network (DNN) training or inference. The 442 patent describes using a combination of different processor types to process sub-tasks—referred to in the context of “computation nodes” and “booster nodes”—during these computing iterations. *See* EX1001 at 3:24-38. It notes that computation

nodes “may be identical to one another or may have differing characteristics.”

EX1001 at 3:24-25. When discussing computation nodes, the specification gives the example of the Intel XeonE5-2680.

37. Historically, sub-tasks were not distributed between different processor types in this manner. The differences in architecture, scalability, execution speed, communications, etc. made such distributions difficult. Iterative processing jobs often have data dependencies (e.g., one sub-task must wait for another sub-task to finish before it can start because it needs the other’s sub-task’s results as an input to its processing) and certain scalability aspects that might make certain parts of the job better suited for certain processors. Efficiently using resources generally requires limiting idle time due to factors such as (1) data dependencies and (2) processing data on a processor not well-suited to the task type. It is far easier to balance such considerations in a homogeneous architecture, where the same types of processors will generally finish the same sub-task in about the same amount of time. It is far more difficult to manage when the various processors are not functioning using the same architecture.

38. For example, the 2024 IEEE article “Efficient Data-parallel Distributed DNN Training for Big Dataset under Heterogeneous GPU Cluster” wrestled with a similar topic in the context of heterogeneous GPU systems (i.e., systems where multiple types of GPUs are in use). EX2014. I note that the differences between

different types of GPUs is generally not nearly as drastic as the differences between GPUs and—for example—CPUs. Nonetheless, the article notes that the use of seven different GPU types (each GPU offers a different quantity of streaming multiprocessors and GPU memory), introduces a “straggler problem” which makes it difficult to predict when a job will complete. This uncertainty stemming from using different processor types at the same time makes it difficult to map processing jobs and “lower[s] GPU utilization.” EX2014 at 179-180. This is just one example of the types of issues that make it so difficult to map sub-tasks across different types of processors.

39. As another example, the 2021 IEEE article “Practice and Experience in using Parallel and Scalable Machine Learning with Heterogenous Modular Supercomputing Architectures” discusses “enabl[ing] users to take advantage of HPC systems that best suit their needs” and classifies user code into three categories:

- (1) “Users with low/medium-scalable codes with high data management benefit from a general purpose cluster (i.e., cluster module).” EX2015 at 3.
- (2) “Other users with highly scalable codes and more regular communication patterns benefit from a massively parallel and scalable HPC system (i.e., booster module).” EX2015 at 3.
- (3) “[T]he third type of users benefits from some **characteristics of both of these two architecture elements** requiring one single platform well interconnected,

as shown in Fig. 2. Those third type of users also use applications that take advantage of the above rather traditional architectures (i.e., general purpose cluster vs. highly scalable booster) **in combination with** innovative computing architectures such as those specifically designed for data analytics (i.e., large memory), quantum computing, or neuromorphic computing.” EX2015 at 3 (bolding and underlining added). The article notes that this third category of users introduces significantly more complexity than was seen under the other two historical models, “leading to a dissonance with these traditional HPC system workloads.” EX2015 at 1-2.

40. Thus, today it is well recognized that the types of problems addressed by the 442 patent (e.g., distributing sub-tasks between different processor types across computing iterations) were difficult to solve given the increased complexity of modern architectures and processing needs. The 442 patent teaches solutions to these problems using techniques related to collecting information relating to the processing of sub-tasks and updating distributions of sub-tasks between the different processor types during subsequent computing iterations.

41. The 442 patent explains that “[c]ertain sub-tasks which were assigned to a booster on the first iteration might in fact not be suitable for processing by the booster such that a processing of the sub-tasks by a computation node rather than a booster might optimize the computation of the task as a whole.” EX1001 at 3:62-66.

This might occur, for example, when the computing demands associated with each subtask can vary significantly. It is important to understand that problems of that type would not be resolved in the same way in a system where all processing will take place on a single processor type. The 442 patent discusses how to use dynamic factors such as scalability and data dependencies to continuously update sub-task distributions to adapt to the kinds of complicated data processing workloads that have become common place. EX1001 at 4:4-63. The 442 patent also discusses an architecture that enables dynamic assignment of computation nodes and booster nodes to each other, rather than a static mapping of such resources as might be found in a consumer computer (which often contains both a CPU and GPU that are statically linked). EX1001 at 2:51-57, 5:10-24.

## **VII. LEVEL OF ORDINARY SKILL IN THE ART**

42. In my opinion a POSITA at the time of invention would have had (1) a bachelor's degree in computer science or computer engineering and approximately two years of experience in the field of heterogeneous computing systems, or (2) a master's degree in computer science or computer engineering with a focus on heterogeneous computing systems.

43. I understand that Dr. Long identified the level of ordinary skill in the art as follows:

A person of ordinary skill in the art ("POSITA") would include someone who had, in January 2018, (i) a bachelor's degree in computer

science, electrical engineering, computer engineering, or a closely related field, as well as (ii) approximately three years of industry experience or research with distributed computing technologies. Additional education could substitute for the industry experience as it typically involves at least 2-4 years of additional study focused exclusively on computer science, as well as a thesis or project spanning several terms to implement a substantial project.

Long Declaration ¶¶ 31-33.

44. I note that my opinions would not change if I applied his definition of a POSITA instead of my own.

### **VIII. CLAIM CONSTRUCTION**

45. Unless otherwise noted, I have applied the plain and ordinary meaning of each claim term as it would have been understood to a POSITA at the time of invention of the 442 patent.

### **IX. THE TEACHINGS OF THE REFERENCES**

46. I understand that Microsoft has asserted two invalidity grounds. Ground 1 is based on a combination of Lippert (EX1004) and Budenske (EX1005) against Claims 1-10 of the 442 patent, and Ground 2 is based on a combination of Lippert, Budenske, and Kambatla (EX1006) against Claims 2-5 and 8-10 of the 442 patent. Petition at 2.

**A. Lippert**

47. Lippert's [EX1004] title is "COMPUTER CLUSTER ARRANGEMENT FOR PROCESSING A COMPUTATION TASK AND METHOD FOR OPERATION THEREOF." I understand that it claims priority to a October 13, 2010 filing. I also understand that one of the 442 patent inventors was also an inventor of Lippert, and that ParTec owns the patent that issued from the Lippert application.

48. Lippert describes a cluster-booster arrangement including computation nodes and boosters. It teaches the use of a dynamic assignment of boosters to computation nodes at runtime. EX1004 at [0007], [0016]-[0017]. It also teaches "an assignment metric is provided, which serves as a basis for the decision which booster is coupled with which computation node." EX1004 at [0017].

49. The 442 patent describes itself as a further improvement of the system described in Lippert. EX1001 at 1:23-27. I have reviewed the prosecution history of the 442 patent, and I noted that Lippert was discussed extensively during prosecution. The patent applicant argued that Lippert did not "recite that the plurality of sub-task[s] are processed in multiple computing iterations" and teach "redistribution based on information relating to the processing of the plurality of sub-tasks." EX1002 at 174-175. I understand that the application was allowed based

on Patent Owner's arguments and the related claim amendments. EX1002 at 183-185.

**B. Budenske**

50. Budenske [EX1005] appears to be an article from the October 1998 edition of the Journal of Supercomputing. The article is focused on what the authors refer to as a heterogeneous computing ("HC") kernel that can map certain tasks to hardware based on pre-determined heuristic mappings derived offline (i.e., not during execution of the task). EX1005 at 387-388.

51. The "heterogeneous" platform of Budenske in 1998 would have been very different from what we think of when we refer to heterogeneous computing today. Budenske gives the example of a system "developed by the Army Research Laboratory for a real-time ATR<sup>1</sup> relational template matching algorithm." EX1005 at 391. ATR uses could be quite different depending on the sensor information being processed, but based on the examples given a POSITA would have understood Budenske was focused on image processing. EX1005 at 393 (referring to image

---

<sup>1</sup> ATR stands for "automatic target recognition." EX1005 at 387. ATR is generally used to recognize objects based on sensor data. EX1005 at 390 ("Simply stated, an ATR system takes a set of images from a group of sensors and produces some type of description of the scene.").

frames). For that type of processing, Budenske gives the example of a platform with up to 64 total processors, some of which include SHARC DSP processors and PowerPC RISC processors. EX1005 at 391.

52. Budenske's discussion of its architecture confirms that, when remapping a task, all subtasks associated with the task will be remapped to the exact same kind of processor. Budenske states —“it is assumed that if an implementation of a given subtask uses multiple processors, all processors will be of the same type.” EX1005 at 392. Even if that was not sufficient, a POSITA would have understood the same thing from other parts of Budenske as well. For example, Budenske notes that the SHARC DSP processors physically share a DRAM. EX1005 at 392. This makes sense for an architecture where a specific task is assigned to just the SHARC DSP processors. They would all share the same working memory, and if five processors were needed instead of four for the next iteration (such as when the number of objects of interest in the last frame changed from four to five—*see* example on page 393), all DSP processors would have access to the same data set and could take over the extra processing load. The same would not be true of different processor types (such as the RISC processors) which would not have direct access to that shared memory pool. Also, on Budenske's system, all processors **of the same type** have symmetric and conflict-free intercommunication.

53. Mapping subtasks to a mixture of processor types would entail additional logic addressing things such as how different nodes would be selected in further iterations, what metrics and information would be needed to redistribute subtasks, as well as what interconnect configurations would be needed for interfacing computation nodes and booster nodes. Other language in Budenske identifies similar challenges. For example, Budenske notes that “the expected execution time of a particular multiprocessor implementation of a subtask is independent of which fixed-size subset of the processors of a given type are assigned to execute the subtask.” EX1005 at 392. A POSITA would understand that this could only be true if a subtask was assigned only to processors of the exact same type. Otherwise, there could be non-trivial differences in execution time for a task such as signal processing, which a DSP would process much faster than a RISC processor.

### **C. Kambatla**

54. Kambatla [EX1006] has the title “UTILIZATION - AWARE RESOURCE SCHEDULING IN A DISTRIBUTED COMPUTING CLUSTER” and claims priority to September 14, 2016. Kambatla relates generally to cluster computing resource management. Nothing in Kambatla relates to the more specific (and generally more complicated) topic of heterogeneous computing systems.

55. Dr. Long stated in his declaration that Kambatla “discloses a heterogeneous computing system that processes a task comprising a plurality of sub-

tasks.” EX1003 at ¶116. He cites paragraphs [0004], [0021], [0059] for that statement, however those paragraphs address generic cluster computing concepts. I see nothing in those paragraphs that would be specific to a heterogeneous computing system capable of using multiple processor types to process a task across computing iterations, and Dr. Long does not appear to provide any analysis that would support his conclusion. I understand that during his deposition Dr. Long also indicated that paragraphs 89 and 94 of Kambatla related to a heterogeneous computing system. EX2020 at 78:14-79:2. However, a POSITA would have understood that, at best, these paragraphs simply indicate what types of processors a cluster computing system might use, and not that they would be used at the same time for task processing. *See, e.g.*, EX1007 paragraph 89 (Using or instead of and when describing the processor: “[t]he processor may be, for example, a conventional microprocessor such as an Intel Pentium microprocessor or Motorola power PC microprocessor.”).

56. Dr. Long also stated in his declaration that “Kambatla explains that its teachings can be successfully implemented in heterogeneous computing systems comprising ‘CPU and memory resources’ (i.e., computation nodes) and ‘GPU[s] (graphical processing [units])’ (i.e., boosters).” EX1003 at ¶119 (citing EX1006 at [0020], [0023], [0059]). I first note that Kambatla does not refer to computation nodes or boosters—those appear to be assumptions that Dr. Long has made on his

own, and he does not provide any analysis to support those conclusions. The reference to “graphical processing” in Kambatla (Dr. Long added “units” to this, though “units” does not appear in the Kambatla patent) seems to be referring to the video display in Figure 8. Given that Kambatla does not discuss a booster or GPU used for computing, we are left to surmise that the GPU mentioned is for “graphical processing”, versus data processing. Furthermore, nothing that Dr. Long cites would suggest the use of a heterogeneous architecture. In the absence of any analysis from him I can only speculate, but he seems to assume that a mere recitation of “GPU” is enough to assume the existence of a heterogeneous system, where applications can run on both a CPU and a GPU. A POSITA would understand that in modern systems, GPUs can be used for both rendering 3-D graphics and data processing, and in older systems (e.g., around the time of Budenske) it is very unlikely that GPUs would ever be used for heterogeneous computing of tasks. Given that Figure 8 in Kambatla only shows a computer display, a POSITA would understand the GPU is being used for displaying graphics, and not for computation tasks, as described in the 442 patent. Even if I set all of that aside and assume for the sake of argument that Kambatla does address using a GPU for computing tasks, it still never addresses the concept of distributing subtasks between computation nodes and booster nodes.

## X. THE ASSERTED GROUNDS

57. I understand that Microsoft has asserted two invalidity grounds. Ground 1 is based on a combination of Lippert (EX1004) and Budenske (EX1005) against Claims 1-10 of the 442 patent, and Ground 2 is based on a combination of Lippert, Budenske, and Kambatla (EX1006) against Claims 2-5 and 8-10 of the 442 patent. Petition at 2. I have been asked to provide opinions on certain aspects of Microsoft's Petition grounds.

### A. Ground 1

#### 1. Claim 1

*a) Distribution of Sub-Tasks to Both Computation Nodes and Booster Nodes in Each Claimed Computing Iteration*

##### i. Claim Interpretation

58. First, a POSITA would have understood that the claims of the 442 patent require distributing sub-tasks to both computation nodes and booster nodes in each claimed computing iteration. There are several elements of the claim language that would lead a POSITA to that conclusion. I have highlighted some of this language in claim 1 below.

1. A method of operating a heterogeneous computing system comprising a **plurality of computation nodes** and a **plurality of booster nodes, at least one of the plurality of computation nodes and a plurality of booster nodes** being arranged to compute a

computation task, the computation task comprising a **plurality of sub-tasks**, the method comprising:

in a first computing iteration, assigning and processing the plurality of sub-tasks **by at least a portion of the plurality of computation nodes** and **at least a portion of the plurality of booster nodes** in a first distribution; and

generating, using information relating to the processing of the plurality of sub-tasks by **at least the portion of the plurality of computation nodes** and **at least the portion of the plurality of booster nodes**, a further **distribution** of the plurality of sub-tasks **between** the plurality of computation nodes and the plurality of booster nodes for processing thereby in a further computing iteration.

59. As shown above, claim 1 addresses processing multiple sub-tasks using multiple computation nodes and multiple booster nodes. These sub-tasks are processed by at least a portion of the computation nodes and at least a portion of the booster nodes. A POSITA would have understood a “portion” to be at least one of the plurality. EX2016 at 1412 (defining portion as “[a] section or quantity within a larger thing; a part of a whole.”).

60. My review of the 442 patent specification supports the same conclusion. The 442 patent addresses trying to improve the distribution between both computation nodes and booster nodes across iterations. EX1001 at 2:5-29, 2:45-50, 3:59-4:3, 4:9-13, 4:59-63.

61. A POSITA would not have understood the Petition Grounds to teach or render obvious distributing sub-tasks to both computation nodes and booster nodes in each claimed computing iteration. I have reviewed the Petition's analysis related to claim 1 Limitation [1.3], claim 3, and claim 9 Limitation [9.5] as applied to distribution of sub-tasks, and in my opinion a POSITA would not have understood any of the grounds to teach or render obvious distribution of sub-tasks between both computation nodes and booster nodes in further computing iterations. My understanding is that Microsoft is not relying on anything from Lippert as teaching or rendering obvious these limitations. I reviewed the Petition and Dr. Long's declaration and did not see anything in either citing Lippert regarding those limitations. *See* Petition at 29-32, 52; EX1003 at ¶¶77-79, 107. The fact that Microsoft did not cite anything from Lippert for that limitation is consistent with my own understanding of Lippert. Lippert does not address updating distributions of sub-tasks across multiple computing iterations in the manner claimed by the 442 patent—a fact that was documented during prosecution of the 442 patent. EX1002 at 173-175. A POSITA would have understood that the 442 patent was a further development of the Lippert disclosure, and that updating a distribution of sub-tasks between computation nodes and booster nodes in further computing iterations as claimed was one of the improvements added by the patent. *See* EX1001 at 1:23-27.

ii. Ground 1 Analysis

62. In my opinion, Budenske (either alone or in combination with Lippert) fails to teach or render obvious distributing sub-tasks to both computation nodes and booster nodes in a further computing iteration based on the claimed information. It is not difficult to see why, given that Budenske comes right out and states that when assigning a task to multiple processors, “all processors will be of the same type.” EX1005 at 392. A POSITA would have understood that such a limitation precludes distributing sub-tasks between processors of different types (e.g., computation nodes and booster nodes) as recited in the challenged claims.

63. As I explained above in Section IX.B, that limitation of the Budenske disclosure is really not surprising given the listed publication date of October 1998, a time in which heterogeneous systems in the context of cluster computing were in their infancy (and a time in which splitting a sub-task between different processor types would have been exceptionally unlikely). What was surprising to me is that I could not find anything in the Petition or in Dr. Long’s disclosure that addressed that statement from Budenske, or even the very different state of the art in the relevant field in 1998. A POSITA would have recognized that both would have presented significant hurdles to applying the teachings of Budenske to any distribution of sub-tasks between processor types.

64. As I noted in Section IX.B, there are several other indicators in Budenske that further confirm that it could not distribute sub-tasks to more than one processor type. These include (1) the fact that “the expected execution time of a particular multiprocessor implementation of a subtask is independent of which fixed-size subset of the processors of a given type are assigned to execute the subtask,” which a POSITA would recognize is only true if the “subset of processors” are all the same type (EX1005 at 392), and (2) the example given of the SHARC DSP processors which share a working memory only with the other SHARC DSP processors (EX1005 at 392).

65. Furthermore, I have reviewed the example “dynamic parameters” Budenske describes on page 393:

The task is the tracking of vehicles on the ground, and the Application Developer has chosen four dynamic parameters: (1) number of objects of interest being tracked in the last image frame, (2) average size of the objects currently being tracked (in terms of percent of the total number of pixels in the image), (3) whether the lighting conditions in the last image frame required requantization for better contrast for display, and (4) whether there was cloud cover detected in the last image frame that required special processing. The representative choices specified for the dynamic parameters are: (1) 0, 1, 2, 4, 8, 12, (2) 5%, 10%, 15%, 20%, 40%, (3) yes or no, and (4) yes or no.

66. This example further supports the fact the Budenske is simply disclosing assigning subtasks to one type of processor. Budenske is processing

image frames in this example, and I cannot think of any combination of parameters that would result in a subtask being moved from a DSP to RISC processor. For example, DSP processors are specifically designed to process array-based data, and typically include multiply/accumulate units that are well suited for image processing tasks. RISC processors typically are not optimized to run these tasks, and almost certainly would not have been optimized to run such tasks at the time Budenske was published.

67. I understand that Microsoft has taken the stance that EX1007 and EX1008 are “successful implementation of Budenske’s teachings.” Pet. at 22. Even taking that assertion at face value, that would further support my conclusion that Budenske does not teach distributing sub-tasks to both booster nodes and computation nodes in a further computing iteration. Both documents reinforce that their teachings were limited to assigning subtasks only to processors of the same type:

“For simplicity, it is assumed that if a data-parallel implementation of a given subtask uses a virtual machine of processors, all processors will be of the same type. Given this and the symmetry property of the inter-processor communications among processors **of the same type, the expected execution time of a particular multiprocessor implementation of a subtask is independent of which fixed-size subset of the processors of a given type are assigned to execute the subtask.**”

EX1007 at 81-82. As shown above, even when the exhibit refers to using a “virtual machine of processors” it still makes it clear that “all processors will be of the same type.”

“Each of these subtasks may allow the use of **multiple processors of the same type** for efficient execution.”

EX1008 at 156.

“[A]n off-line mapping heuristic (the enhanced GA described in Section 6.5.6) uses the following information to create a mapping, i.e., to transform a DAG into an MDDG: ...(2) the expected execution time of each subtask on a set of processors (**of the same type**) assigned, as a function of the type and number of processors...”

EX1008 at 158.

“Because each subtask can be mapped to multiple processors **of the same type**...”

EX1008 at 159.

68. Overall, there is nothing in Budenske that would suggest to a POSITA that it was disclosing distributing sub-tasks between different processor types across multiple iterations, and quite a few indications that the opposite was true (including express statements by the authors, architectural limitations that make such a distribution very unlikely, and the state of the art in 1998 and even up through the 2006 publication date of EX1007).

iii. Budenske Does not Enable Distribution of Sub-  
Tasks to Both Computation Nodes and Booster  
Nodes in a Further Computing Iteration

69. I was asked to consider whether Budenske is an enabling disclosure for the limitations for which the Petition relies on it. In my opinion, it is not.

70. First, Budenske makes several statements that would have led a POSITA to understand that it is not an enabling disclosure. For example, it states:

[T]his paper focuses on (1) the application and hardware platform characteristics that are needed to enable the use of the HC Kernel, (2) the techniques that comprise the HC Kernel, and (3) how the information needed for the HC Kernel to operate is collected. The IOS has not been implemented; such an implementation is a major undertaking and outside the scope of this paper, which is the design concepts for the HC Kernel.

EX1005 at 388-389.

71. The fact that an actual implementation would have been a “major undertaking” and “outside the scope of this paper” would certainly indicate to a POSITA that this paper was not enabling such a “major undertaking.” Furthermore, a POSITA would have understood that “design concepts” relate to high level ideas that would likely require extensive and undue experimentation to actually reduce to practice and implement. Researchers often come up with architectures that look attractive theoretically, but either cannot be implemented in reality, or are too

complex to be practical. The theories in the Budenske article would require extensive experimentation and likely the invention of new solutions in order to enable what I understand Microsoft to be relying on them for.

72. A POSITA would have also recognized that Budenske is not even addressing the full system necessary to implement its theoretical design concepts. For example, Budenske states “[t]his paper concentrates on the operation of the HC Kernel; the other components will be discussed only to the extent needed for describing the HC Kernel.” EX1005 at 388. Again, this would be a sign to a POSITA that Budenske was not actually enabling the practice of its theoretical concepts, and that even more experimentation might be necessary to address components other than the HC Kernel.

73. As I discussed above in Section IX.B, Budenske’s simplifications allow it to avoid solving the challenges that would be needed to distribute sub-tasks to different types of processors during the same computing iteration.

74. As I discussed in the previous section, EX1007 and EX1008 further demonstrate that Budenske did not disclose (much less enable) assignment of sub-tasks to both computation nodes and booster nodes in the same computing iteration.

75. I also note that EX1007 states “implementation of the On–Off methodology entails addressing the fundamental research issue of how to select representative mappings off-line for on-line use.” EX1007 at 80. I understand this

“On-Off methodology” to be what Microsoft refers to as Budenske’s “On-Off” approach.” Petition at 22.

76. “Fundamental research” is a term of art in the research community. It is sometimes referred to as “Basic Research,” but “basic” in this context does not mean simple, but rather that it relates to the building blocks on which further research and experimentation will be based. For example, the National Center for Science and Engineering Statistics describes “Basic Research,” excerpted below:

Basic research is systematic study directed toward greater knowledge or understanding of the fundamental aspects of phenomena and of observable facts without specific applications towards processes or products in mind. It includes all scientific study and experimentation directed toward increasing fundamental knowledge and understanding in those fields of the physical, engineering, environmental, and life sciences related to long-term national security needs. It is farsighted high payoff research that provides the basis for technological progress. Basic research may lead to: (a) subsequent applied research and advanced technology developments in Defense-related technologies, and (b) new and improved military functional capabilities in areas such as communications, detection, tracking, surveillance, propulsion, mobility, guidance and control, navigation, energy conversion, materials and structures, and personnel support.

EX2017 at 20.

77. The National Center for Science and Engineering Statistics also describes, “Applied Research” excerpted below:

Applied research is systematic study to understand the means to meet a recognized and specific need. It is a systematic expansion and application of knowledge to develop useful materials, devices, and systems or methods. It may be oriented, ultimately, toward the design, development, and improvement of prototypes and new processes to meet general mission area requirements. Applied research may translate promising basic research into solutions for broadly defined military needs, short of system development. This type of effort may vary from systematic mission-directed research beyond that in Budget Activity 1 to sophisticated breadboard hardware, study, programming and planning efforts that establish the initial feasibility and practicality of proposed solutions to technological challenges. It includes studies, investigations, and non-system specific technology efforts.

EX2017 at 20.

78. The fact that the authors of EX1007 considered the “On-Off methodology” of EX1007 to still be a “fundamental” or “basic” research topic would have meant a POSITA would have understood that additional experimentation would be necessary to get the theory to the stage where Applied Research could begin to determine how implementation would even be possible. A POSITA would have understood that undue experimentation would certainly have been needed to get the theoretical “On-Off methodology” idea past the fundamental and applied research stages to a point where a POSITA could make the invention as claimed in the 442 patent. The fact that “fundamental research” was still being performed on

the topic 7+ years after Budenske was published further reinforces my conclusion that the experimentation necessary to reach the implementation stage would have been extensive and thus undue.

79. EX1007 also confirms that “[n]either of the[] methods [suggested by Budenske for determining dynamic parameter values] was ever evaluated in any way, as this was not the focus of Budenske.” EX1007 at 79. This would also suggest to a POSITA that the Budenske disclosure had not even performed the minimum necessary experimentation to see if the dynamic parameter concept was viable.

80. Other aspects of EX1007 further reinforce this understanding. For example, EX1007 puts forward Equation (1), which is an equation for “Execution time of subtask  $s_i$  on virtual machine  $u$ .” EX1007 at 83. The equation given (which only has a single heterogeneity factor  $h_{iu}$ , a single coefficient for the parallel portion of subtask  $i$ ’s execution time  $a_i$ , a single coefficient for the parallelization overhead portion of subtask  $i$ ’s execution time  $b_i$ , and a single coefficient for the serial portion of subtask  $i$ ’s execution time  $c_i$ , could only be accurate if processing for subtasks was limited to a single processor type. The authors were proposing use of this equation to determine the optimal number of processors “ $p$ ” when using multiple processors **of the same type**. EX1007 at 83 (“By differentiating this equation and equating the derivative to zero, the optimal value of  $p$  that leads to the minimum execution time for a given subtask is  $p_{opt} = (a_i\alpha)/(b_i\beta)$ .”). A POSITA would understand the equation

could not have been used for computing a task with heterogeneous processors, at least because there would be a need to consider multiple derivatives at the same time along with other factors not accounted for in the equation. Even for the purpose of determining the number of processors of the same type to use, a POSITA would have viewed the disclosed formula as experimental and in need of further improvement before it could be reliably applied.

81. Furthermore, I gave examples in Section VI that demonstrate some of the difficulties that stem from trying to distribute sub-tasks between different types of processors, such as the straggler problem.

*b) Generating, Using Information Relating to the Processing of the Plurality of Sub-Tasks...a Further Distribution*

82. A POSITA would have understood that in 442 patent claim 1 the further distribution of sub-tasks is generated “using information relating to the processing of the plurality of sub-tasks by at least the portion of the plurality of computation nodes and at least the portion of the plurality of booster nodes.” EX1001 at 5:37-40. However, a POSITA would have understood that Budenske does not generate its distributions using information related to the processing of the plurality of sub-tasks. Budenske’s primary focus was the use of “Off-Line Derived Remappings” of tasks. EX1005 Title, 387. These off-line heuristics were generated using a “genetic algorithm.” EX1005 at 398. The heuristics were generated off-line because the Budenske authors determined that generating the mappings on the fly (as would be

needed to generate them based on information relating to the processing of the sub-tasks) would be too difficult or resource intensive. They theorized that the longer execution times allowed by offline heuristics might result in better mappings. EX1005 at 389-390. Based on this, a POSITA would have understood that the offline generated mappings of Budenske that were generated prior to any task processing were not generated using information relating to the processing of the plurality of sub-tasks.

*c) Information Relating to the Processing of the Plurality of Sub-Tasks*

83. A POSITA would have understood that Budenske's dynamic parameters could not teach or render obvious the "information relating to the processing of the plurality of sub-tasks" of 442 patent claim 1. EX1001 at 5:37-38. I reviewed the 442 patent specification, and the 442 patent specification makes a distinction between the results of execution of sub-tasks and the claimed information relating to the processing of the plurality of sub-tasks:

During a first iteration of the task, [1] the results of the execution of the sub-tasks are collected together with [2] the information from the daemons concerning the processing of the sub-tasks and the status of the nodes.

EX1001 at 4:23-27 (bracketed numbers added).

84. A POSITA would have understood from this that the results of the execution of the sub-tasks were not included in the information from the daemons concerning the processing of the sub-tasks and the status of the nodes.

85. Microsoft relies on Budenske's dynamic parameters for this limitation, but the dynamic parameters of Budenske that I understand them to be relying on are the results of the execution of the sub-tasks. For example, the examples of dynamic parameters that Budenske gives on page 393 are:

- (1) number of objects of interest being tracked in the last image frame,
- (2) average size of the objects currently being tracked (in terms of percent of the total number of pixels in the image), (3) whether the lighting conditions in the last image frame required requantization for better contrast for display, and (4) whether there was cloud cover detected in the last image frame that required special processing.

86. Each of those would be results of the ATR task described by Budenske. Each of these (number of objects in the last frame, average size of objects tracked, lighting conditions, and cloud cover) would be the results that came from processing the previous image frame.

## **B. Claim 3**

### **1. Dynamically Alter Further Distribution**

87. Claim 3 recites the following:

The method according to claim 2, wherein a resource manager determines the assignment of the plurality of sub-tasks to the plurality

of computation nodes and the plurality of booster nodes for the first computing iteration as a function of the computation task and wherein the application manager receives the information and processes the information as input to the resource manager such that the **resource manager dynamically alters further distribution during the computing of the computation task.**

EX1001 at 6:4-5 (emphasis added).

88. A POSITA would have understood that the “further distribution” being referred to here is the same “further distribution” of sub-tasks referred to in Claim 1. EX1001 at 5:40-44. The Petition grounds appear to rely on Lippert’s disclosure of the assignment of computation nodes and booster nodes to each other for this limitation. Pet. at 39 (citing EX1004 [0013]). As I discussed previously, a POSITA would have understood that Lippert does not teach altering further distribution of sub-tasks during the computing of the computation task as claimed by the 442 patent.

### **C. Claim 5**

#### **1. Computation Nodes Daemons and Booster Node Daemons**

89. 442 patent claim 5 recites “[t]he method according to claim 1, wherein daemons operate in the plurality of computation nodes and the plurality of booster nodes to generate the information.” EX1001 at 6:12-14. I have reviewed both Lippert and Budenske, and I do not see any discussion in either reference of daemons operating in computation nodes or daemons operating in booster nodes. At best, Lippert discloses in paragraph 14 that a resource manager may be formed by a

daemon, but a POSITA would not understand that a disclosure of a daemon forming one component would render obvious other types of daemons.

90. The term “daemon” is sometimes used to refer to as a background process, but a daemon operating in one context will not function the same (or be obvious based on) a daemon operating in another context. The context and operation of the nodes are important. In the context of claim 5, there is one set of daemons operating in the computation nodes and one set of daemons operating in the booster nodes. Those daemons generate the “information relating to the processing of the plurality of sub-tasks by at least the portion of the plurality of computation nodes and at least the portion of the plurality of booster nodes” recited in 442 patent claim 1. EX1001 at 5:37-40.

91. I also note that using daemons was not the only way to generate information in computation nodes and booster nodes. For example, dedicated hardware could be used to generate such information without the use of daemons operating in computation nodes and booster nodes. I understand that when asked to consider the same question, Dr. Long testified to the following:

Well, each application itself, instead of relying on a daemon for gathering information, could do its own instrumentation for the information back.

You don't need a daemon to collect and collate the information; you could have that done by, say, the application manager itself or the resource manager itself.

EX2020 at 128:15-21.

92. I agree that those options he identified (instrumentation by the application, or using the application manager or resource manager to collect the information) are also alternatives to the use of daemons operating in computation nodes and daemons operating in booster nodes as claimed.

**D. Claim 9**

**1. Distribution of Sub-Tasks To Both Computation Nodes and Booster Nodes in Each Claimed Computing Iteration**

93. Claim 9 is similar to Claim 1 in its distribution of sub-tasks. I have highlighted some of the language from claim 9 below (emphasis added).

9. A heterogeneous computing system comprising:

a **plurality of computation nodes** and a **plurality of booster nodes** for computing one or more **tasks comprising multiple sub-tasks**;

a communication interface connecting the plurality of computation nodes with each other and the plurality of booster nodes;

a resource manager for assigning **at least a portion of the plurality of booster nodes** and **at least a portion of the plurality of computation nodes** to each other for the computing of the one or more tasks in a first computing iteration; and

an application manager configured to receive information from daemons operating in **at least the portion of the plurality of computation nodes** and **at least the portion of the plurality of**

**booster nodes** to update a **distribution** of the multiple sub-tasks **between** the plurality of computation nodes and the plurality of booster nodes in a further computing iteration.

Again, a POSITA would have understood that this requires distributing sub-tasks between both computation nodes and booster nodes in the claimed computing iterations. A POSITA would have understood from this that simply assigning sub-tasks to just booster nodes or just computation nodes would not be sufficient to satisfy the claimed inventions of the 442 patent.

94. For the same reasons I discussed above in Section X.A.1.a, a POSITA would have understood that the Petition grounds do not teach or render obvious this limitation.

## 2. Computation Node Daemons and Booster Node Daemons

95. Similar to 442 patent claim 5, 442 patent claim 9 requires:

[R]eceiv[ing] information from **daemons operating in at least the portion of the plurality of computation nodes and at least the portion of the plurality of booster nodes** to update a distribution of the multiple sub-tasks between the plurality of computation nodes and the plurality of booster nodes in a further computing iteration.”

EX1001 at 6:38-43 (emphasis added).

96. For the same reasons I discussed above in Section X.C, a POSITA would have understood that the specific daemons operating in computation nodes and daemons operating in booster nodes were not taught by the Petition grounds.

**E. Ground 2**

**1. Claim 3**

*a) Dynamically Alter Further Distribution*

97. As I noted above for Ground 1, a POSITA would not have understood the Petition grounds to teach or render obvious Claim 3, which recites:

3. The method according to claim 2, wherein a resource manager determines the assignment of the plurality of sub-tasks to the plurality of computation nodes and the plurality of booster nodes for the first computing iteration as a function of the computation task and **wherein the application manager receives the information and processes the information as input to the resource manager such that the resource manager dynamically alters further distribution during the computing of the computation task.**

EX1001 at 5:47-6:5 (emphasis added).

98. I reviewed Dr. Long’s analysis for this limitation, EX1003 paragraph 126, and in my opinion it does not change my conclusions from Ground 1.

99. Dr. Long states that “Kambatla discloses that its central resource manager includes various modules configured to handle different tasks and feed information to other modules ‘to aid in resource allocation.’ EX1006, [0036], [0031]-[0035].” EX1003 paragraph 126. As I discussed previously, a POSITA would not have understood Kambatla to be teaching processing tasks using multiple different types of processors, therefore its generic disclosures regarding “modules

configured to handle different tasks” would not teach the specific limitations recited in the claim.

100. Dr. Long also states that “Kambatla also discloses that ‘more or fewer components, organized in various configurations ... remain[] within the scope’ of Kambatla. EX1006, [0030].” EX1003 paragraph 126. Again, a POSITA would have understood this to be no more than a generic statement that things could be done differently than Kambatla, which certainly would not teach or suggest any specific method of dynamically altering further distribution during the computing of the computation task as claimed.

## 2. Claim 5

### a) *Computation Nodes Daemons and Booster Node Daemons*

101. As I noted above for Ground 1, a POSITA would not have understood the Petition grounds to teach or render obvious Claim 5, which recites:

5. The method according to claim 1, wherein daemons operate in the plurality of computation nodes and the plurality of booster nodes to generate the information.

EX1001 at 6:11-14.

102. I reviewed Dr. Long’s analysis for this limitation, EX1003 paragraphs 128-129, and in my opinion it does not change my conclusions from Ground 1.

103. As I discussed above in Section IX.C, Kambatla does not even discuss the concept of heterogeneous computing for processing of tasks.

104. Kambatla discloses daemons in the abstract, but a POSITA would have understood that was not sufficient to disclose the specific claimed daemons operating in computation nodes and daemons operating in booster nodes.

### 3. Claim 9

#### *a) Distribution of Sub-Tasks To Both Computation Nodes and Booster Nodes in Each Computing Iteration*

105. I reviewed Dr. Long's ground 2 analysis for Claim 9, EX1003 paragraphs 132-137, and it does not change my analysis above for ground 1.

#### **F. A POSITA would not have modified the references in the claimed manner**

106. In addition to the Petition Grounds not teaching the limitations discussed above, it is also my opinion that a POSITA would not have been motivated to combine Budenske with either Lippert or Kambatla.

107. For the same reasons I discussed above about why Budenske is not an enabling disclosure in Section X.A.1.iii, a POSITA would not have been motivated to combine Lippert with Budenske and a POSITA would not have reasonably expected to have success combining Lippert with Budenske.

108. For background, I note that Budenske is a product of its era. As I discussed above, Budenske requires that when assigning subtasks to multiple processors, all processors must be the same type. EX1005 at 392. As I explained above in Sections IX.B and X.A, that limitation on the Budenske disclosure is really not surprising given the listed publication date of October 1998, a time in which

heterogeneous systems in the context of cluster computing were in their infancy (and a time in which splitting a sub-task between different processor types would have been exceptionally unlikely).

109. Budenske had some interesting ideas applicable to the very limited heterogeneous architectures available in 1998, but no POSITA trying to improve on Lippert (which came around 12 years later and addressed the much harder problem of dealing with a combination of processor types) would have looked to Budenske. Doing so would be like looking to an electric golf cart from the 1990's to improve on the design of electric cars made in the 2010's because both are electric vehicles and a lot of the terminology is the same (e.g., battery, electric motor, steering wheel). Even though Lippert and Budenske both relate to cluster computing and refer to heterogeneity, subtasks, and iterations in some capacity, a POSITA would have understood there was very little chance of applying the earlier/simpler concepts of Budenske to the much more complicated later applications.

110. Additionally, the other references I have reviewed (1) teach away from combining Budenske's dynamic parameters and subtask mapping to systems mapping subtasks to different processor types, and (2) suggest to a POSITA that such a combination would be unlikely to succeed.

111. First, the mere fact that Budenske expressly limits its mappings to a single type of processor strongly teaches away from trying to apply its teachings to

architectures that would need to be able to perform mapping of sub-tasks to multiple different processor types at the same time. EX1005 at 392. As I discussed in sections V, VI and IX.B, mapping sub-tasks across processor types is substantially more difficult than mapping sub-tasks to a single processor type, and the fact that the authors of Budenske felt it necessary to make it clear that their system was limited to a single processor type would have certainly discouraged a POSITA from trying to apply it to different types of processors. I incorporate my analysis from those sections by reference here.

112. Subsequent publications addressing Budenske would have confirmed that conclusion for a POSITA. I see that Howard Jay Siegel is one of the authors of both Budenske and EX1007, and based on that I believe he would have been very familiar with Budenske and its applicability. EX1007 notes that mapping tasks using the dynamic parameters of Budenske was not “ever evaluated in any way, as this was not the focus of Budenske,” and instead Budenske was focused on the ATR and HC Kernel design. EX1007 at 79. In general, a POSITA would tend not to rely on part of an article that the authors admit was not the focus and did not receive much attention. This would both teach away from using those teachings from Budenske, as well as suggest to a POSITA that trying to implement such teachings would not be likely to succeed.

113. I previously discussed the difficulty of mapping sub-tasks generally in Sections V, VI and IX.B, and especially mapping them to different types of processors, along with the significant changes in the state of the art between Budenske (1998), Lippert (2010), and the 442 patent (2018). Kambatla does not even address distributing subtasks in heterogeneous architectures (much less the difficulty in mapping sub-tasks between processors of different types). However, a POSITA would find that Kambatla confirms the difficulty of distributing tasks even in the context of a homogeneous architectures. For example, paragraph [0005] of Kambatla highlights that it is difficult to predict the amount of computing resources that will be needed to process a task even in the context of a homogeneous architectures. EX1006 at [0005]. It further explains that prediction is complicated by the fact that resource usage for a task varies over time and varies by input. In the context of a heterogeneous architecture, those problems would be significantly more difficult to overcome. For example, as I discussed previously certain processors will be able to process certain types of input far more efficiently than others. Thus, heterogeneous architecture introduces yet another variable to a problem that Kambatla acknowledged was difficult on even a homogeneous architecture. This again teaches away from trying to apply solutions that are focused on a single processor type to sub-tasks that must be split across different processor types.

114. Finally, as Kambatla fails to describe a class of heterogeneous architectures (e.g., where CPUs and GPUs are used for a single application for computation), a POSITA would not be compelled to look to Kambatla to combine with Lippert or Budenske.

115. Furthermore, as I discussed previously EX1007 and EX1008 showed that others had not succeeded in applying Budenske's theories regarding using dynamic parameters to assign subtasks. As discussed previously, EX1007 still considered Budenske's "On-Off" methodology to be a "fundamental research" question even years after Budenske was published.

## **XI. MISCELLANEOUS**

116. My work on this matter is being billed at an hourly rate of \$500 per hour and is not contingent on the outcome of this proceeding. I am also being reimbursed for reasonable expenses I incur in relation to my services provided for this proceeding.

117. All statements made herein of my own knowledge are true and all statements made on information and belief are believed to be true. I further understand that willful false statement and the like are punishable by fine or imprisonment, or both under Section 1001 of Title 18 of the United States Code. I declare under penalty of perjury that the foregoing is true and correct.

Date: September 30, 2025

By: 

Dr. David Kaeli