

UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE PATENT TRIAL AND APPEAL BOARD

---

AMERICAN AIRLINES, INC.

AND

SOUTHWEST AIRLINES CO.

Petitioners

v.

INTELLECTUAL VENTURES I, LLC,

Patent Owner

---

IPR2025-00785

U.S. Patent No. 7,257,582 B2

**DECLARATION OF DR. DANIEL ABADI  
IN SUPPORT OF PATENT OWNER'S PRELIMINARY RESPONSE**

## TABLE OF CONTENTS

|                                                                                                                                                 |    |
|-------------------------------------------------------------------------------------------------------------------------------------------------|----|
| I. INTRODUCTION AND QUALIFICATIONS .....                                                                                                        | 1  |
| II. RELEVANT LEGAL STANDARDS .....                                                                                                              | 6  |
| III. THE ‘582 PATENT .....                                                                                                                      | 9  |
| IV. PRIOR ART RELIED ON IN THE PETITION .....                                                                                                   | 10 |
| A. Chow (EX1003).....                                                                                                                           | 11 |
| B. Reiffin (EX1004).....                                                                                                                        | 13 |
| C. Kurowski (EX1004).....                                                                                                                       | 14 |
| V. GROUND 1: CLAIM 1 IS NOT OBVIOUS IN VIEW OF CHOW; NOT BY<br>ITSELF, NOR IN COMBINATION WITH REIFFIN .....                                    | 15 |
| A. Aspects of Chow focus on a single computer system .....                                                                                      | 16 |
| B. The ‘582 patent is designed for a distributed computer system consisting of<br>many computers .....                                          | 18 |
| C. Distributed systems pay communication costs to distribute information<br>across computers in the system.....                                 | 19 |
| D. The ‘582 patent explicitly tries to reduce required communication between<br>computers .....                                                 | 20 |
| E. The petition appears to misunderstand the basic meaning of the word<br>“record” in data processing systems .....                             | 21 |
| F. The petition makes the same misunderstanding when discussing Reiffin ....                                                                    | 22 |
| VI. GROUND 2: CLAIM 1 IS NOT OBVIOUS IN VIEW OF CHOW; NOT BY<br>ITSELF, NOR IN COMBINATION WITH REIFFIN AND KUROWSKI ..                         | 23 |
| A. The petition makes the same misunderstanding regarding claim 1a when<br>discussing Kurowski as it did when discussing Chow and Reiffin ..... | 25 |
| VII. CONCLUSION .....                                                                                                                           | 26 |
| VIII. DECLARATION.....                                                                                                                          | 27 |

## **I. INTRODUCTION AND QUALIFICATIONS**

1. I, Daniel Abadi, submit this declaration on behalf of Intellectual Ventures I LLC, (“Patent Owner”) to provide my opinions concerning the validity of the claims of U.S. Patent No. 7,257,582 (the “ ‘582 patent,” Ex.1001), in support of Patent Owner’s Preliminary Response to the Petitioner for Inter Partes review of the ‘582 patent, which I understand was filed by American Airlines Inc., (“AA”) and Southwest Airlines, (the “Petitioners”) and accompanied by a Declaration of Dr. Shamos (the “Shamos Declaration”), among other exhibits.

2. I am more than eighteen years of age, and I am a citizen of the United States, currently residing in the State of Maryland.

3. In formulating my opinions, I have relied upon my knowledge, training, and experience in relevant art. My qualifications, along with a more detailed summary of my background and publications, are stated more fully in my curriculum vitae, which has been provided as Exhibit 2012. Here, I provide a concise summary of my qualifications.

4. I am a Professor in the Department of Computer Science at the University of Maryland in College Park. I was hired as the Darnell-Kanal Chair, so technically I am the Darnell-Kanal Professor of Computer Science at the University of Maryland, College Park.

5. In addition to my two decades in academia, I have also been involved in several start-ups. My PhD thesis was commercialized by a company called Vertica for which I served as a consultant to help with the technology transfer. In addition, the HadoopDB project which was developed in my lab at Yale University (back when I was on the computer science faculty there) was commercialized by a company called Hadapt. I served as co-founder and Chief Scientist at Hadapt during the company's entire existence until it was acquired by Teradata.

6. Vertica built one of the first commercially successful column-stores in history. Column-stores store database tables on disk in columns instead of rows. This makes read analysis much faster (since irrelevant columns can be ignored) at the cost of increasing the time it takes to insert records (rows) into tables. My work on C-Store introduced mechanisms for decreasing the write overhead while further accelerating the read benefits through compression, introduction of operators that know how to directly process column-oriented compressed data, delaying the stitching of columns into rows and other new operators including a faster column-oriented join algorithm.

7. My work on C-Store and Vertica made tremendous impact in the industry. Today, almost every commercial database system vendor implements column-store options for their customers. Consequently, this work won the test of time award at VLDB --- the most prestigious award at one of the two leading yearly database community conferences.

8. The HadoopDB project introduced the first high-performance SQL layer on top of data stored in Hadoop (it was concurrent with Hive that had a competing goal, but was significantly slower). It used a combination of traditional database operators and MapReduce operations to process database queries.

9. The HadoopDB project had a similar impact as the C-Store project mentioned above. Now there are dozens of companies that offer SQL layers to data stored in Hadoop. Consequently, this work won yet another test of time award at VLDB.

10. My work on streaming database systems --- including Aurora and Borealis also has been impactful. Today many companies use the streaming technology to process data as it generated using ideas introduced by the Aurora and Borealis projects. The Borealis project won yet another test of time award --- this time at CIDR --- 20 years after the seminal paper was published there.

11. I am unaware of any other member of my technical community that has won three test of time awards (though the data of who has won these awards does not appear to be maintained in a central location, so I have not been able to verify whether I am the only one or not). At the very least, this is an extremely rare feat.

12. I was named an ACM Fellow in 2020, an elite recognition of outstanding science and scholarship that is bestowed upon less than one percent of the organization's 100,000 members.

13. I have also won several other awards for my research and contributions to the database systems community. Perhaps the four most notable of these awards are the VLDB 2007 best paper award, the 2008 Jim Gray Doctoral Dissertation Award, the 2013 VLDB Early Career Researcher Award, and the Yale Provost's Teaching Prize.

14. I have consistently advised undergraduate students, masters students, PhD students, and postdocs throughout my career. Two of my PhD students won the Jim Gray Doctoral Dissertation Award --- the highest award for a PhD dissertation given by the database systems community. The name of these two students are Alexander Thomson and Jose Faleiro.

15. My research publications have received 25000+ citations with an h-index of 55 (based on Google Scholar).

16. I am a named inventor on several patents corresponding to the areas of my professional work. Accordingly, I have a general understanding of the U.S. patent prosecution process and of the novelty and non-obviousness requirements for patentability. Example patents include:

- US Patent 10,868,865 System and apparatus to manage data using a peer-to-peer network and the blockchain (2020)
- US Patent 9,934,276 Systems and methods for fault tolerant, adaptive execution of arbitrary queries at low latency (2018)

- US Patent 9,495,427 Processing of data using a database system in communication with a data processing framework (2016)
- US Patent 9,471,711 Schema-less access to stored data (2016)
- US Patent 9,336,263 Data loading systems and methods (2016)
- US Patent 8,935,232 Query execution systems and methods (2015)
- US Patent 8,886,631 Query execution systems and methods (2014)
- US Patent 8,700,563 Deterministic database systems (2014)

17. I believe that my extensive industry experience and educational background qualify me as an expert in the relevant field of parallel and distributed data processing systems. Almost the entirety of my career has been spent working in this area, including the Borealis, C-Store, and HadoopDB projects mentioned above.

18. I am knowledgeable of the relevant skill set that would have been possessed by a hypothetical person of ordinary skill in the art (“POSITA”) at the time of the claimed invention of the ‘582 patent, and accept for purposes of this declaration only the relevant skill level set forth in the Petition. I have the necessary expertise to provide my informed opinions in this IPR, and I am an expert in the art of the ‘582 Patent due to all of the reasons mentioned above, and especially from my work on HadoopDB and Hadapt which have a very strong overlap with the ‘582 patent (they both use specifically MapReduce techniques to process data in parallel, and the ‘582 patent explicitly describes database queries as a use case).

## II. RELEVANT LEGAL STANDARDS

19. The opinions I am expressing in this report involve the application of my knowledge and experience to evaluate certain references with respect to the ‘582 patent. My formal knowledge of patent law is no different than that of any lay person. Therefore, I have requested the attorneys from Volpe Koenig, who represent Intellectual Ventures, to provide me with guidance as to the applicable patent law in this matter. The paragraphs below express my understanding of how I must apply current principles related to patent validity to my analysis.

20. I understand that an IPR is a proceeding before the USPTO for evaluating the patentability of an issued patent’s claims based on prior-art patents and printed publications.

21. I understand that the claims of a patent are reviewed from the point of view of a hypothetical person of ordinary skill in the art (“POSITA”) at the time of the filing of the patent. The “art” is the field of technology to which a patent relates.

22. I understand that in determining whether a patent claim is anticipated or obvious in view of the prior art, the Patent Office must apply the *Phillips* standard to construe the claims.

23. I understand that, in this proceeding, Petitioners have the burden of proving that the challenged claims of the ‘582 Patent are unpatentable by a

preponderance of the evidence. I understand that “preponderance of the evidence” means that a fact or conclusion is more likely true than not true.

24. I understand that there are two ways in which prior art may render a patent claim unpatentable. First, the prior art can “anticipate” the claim. Second, the prior art can render the claim “obvious” to a POSITA. I understand that for an invention claimed in a patent to be patentable, it must not be anticipated and must not be obvious based on what was known before the invention was made. For purposes of this Declaration only, I have applied the definition of a POSITA proposed in the Petition.

25. I understand that a patent claim may be invalid as “anticipated” under 35 U.S.C. § 102 if each and every limitation of the claim as construed by the court is found, either expressly or inherently, in a single prior art reference, such as a publication or patent, that predates the claimed subject matter. I also understand that a patent claim may be invalid as obvious under 35 U.S.C. § 103 if the differences between the subject matter claimed and the prior art are such that the claimed subject matter as a whole would have been obvious to a POSITA at the time the invention was made. I also understand that several factual inquiries underlie a determination of obviousness. These inquiries include (1) the scope and content of the prior art, (2) the level of ordinary skill in the field of the invention, (3) the differences between the

claimed invention and the prior art, and (4) any objective evidence of non-obviousness (i.e., “secondary considerations”).

26. I also understand that, when a party alleges obviousness based on a combination of references, that party should identify a reason why a POSITA would have been motivated to combine the asserted references in the manner recited in the claims and to explain why a POSITA would have had a reasonable expectation of success in making such combinations.

27. I understand that an obviousness analysis permits the application of “common sense” in examining whether a claimed invention would have been obvious to a POSITA. For example, I understand that combining familiar elements according to known methods and in a predictable way may suggest obviousness when such a combination would yield nothing more than predictable results. I also understand a party asserting obviousness should still provide a specific motivation to combine the references as recited in the claims and explain why one would have reasonably expected to succeed in doing so.

28. I understand that one indicator of non-obviousness is when a prior art reference leads in a different direction or discourages that combination, recommends steps that would not likely lead to the patent’s result, or otherwise indicates that a seemingly inoperative device would be produced.

### III. THE '582 PATENT

29. The '582 patent discusses processing records in a file. It discusses several different examples of why a practitioner would want to process records in a file, including data sorting, batch data processing, and querying of data. All of these examples involved reading through (potentially many) records, performing processing actions on them, and then returning results or report outputs.

30. Fundamentally, the '582 patent deals with data-centric processing. The system is designed to be able to read and process large amounts of data, while maximizing the ability for a set of computers running in parallel to share in the effort of doing this reading and processing.

31. Another important contribution is that the '582 patent performs dynamic load balancing across all of the computers performing data processing. The claims require dividing an input file into many different partitions (more partitions than the available computers to process it).

32. Only after these partitions have been defined and identified, data processing can begin ("When the split step terminates, the various Sub Tasks can be activated." EX1001, 3:64). Each computer can run a "Sub Task" that takes responsibility for processing a partition that has been previously identified.

33. Different computers maintain load balance by only accepting responsibility for one partition at a time on a first-come first-served basis. Only after

it completes processing of one partition does it take responsibility for the next one. Since there are more partitions than there are computers, the load is balanced naturally. The faster computers will end up requesting responsibility for more partitions over time as they process them at a fast rate, while the slower computers will end up requesting responsibility for fewer partitions. All computers are constantly working on their data processing tasks until there are no more partitions to process.

34. The patent also discusses how to partition records in a file so that records do not get split across partition boundaries, how to define partition ranges using byte ranges, and how to perform this partitioning process without actually having to perform an extra scan of the input file while partitioning it. It also discusses that the partitioning process may take into account various additional properties of the computers that will be participating in the data processing effort.

35. The end of the patent discusses several special case embodiments for particular uses cases, including data sorting, tasks for which the expected output size is small, and tasks for which data output order does not matter. In each of these cases, optimizations can be made to the basic description described earlier in the patent in order to achieve better performance.

#### **IV. PRIOR ART RELIED ON IN THE PETITION**

36. The petition discusses three other patents that are used in its obviousness claims with respect to the '582 patent. In this section I will describe each one, in turn.

## **A. Chow (EX1003)**

37. In the 1997 time frame that the Chow patent was filed, it was becoming more common for computers to be manufactured with multiple processors. The trend has only continued until today, where the vast majority of computers are now manufactured with multiple processors.

38. Computers that have multiple processors typically also contain some memory (RAM) that all these processors are able to read or write to. It may also contain a disk that all these processors are able to read and write to. Such systems are called shared-memory systems. The vast majority of laptops, desktops, and servers built today fit into this category.

39. Chow discusses how to perform data processing in a shared-memory system (see Figure 1 of Chow which is also analyzed in a later section of this document). The goal of Chow is to allow each processor of the shared-memory system to work in parallel to complete a data processing task.

40. Chow uses a straw-model analogy to describe how it processes data. The way the straw model works is that there is an initial shared pool of data that needs to be processed. This is the glass of water in the straw-model analogy. Each processor on the fly allocates to itself a chunk of data to process from this shared pool. This corresponds to drinking of the water from a straw in the straw-model analogy. (“Thus,

the multiprocessing system operates like multiple people drinking water from a common glass through individual straws” EX1003, 2:17).

41. In Chow, the shared pool of data (the “water”) sits in shared-memory. All the processors drink from it (i.e. process it) like from a straw. Each processor has freedom to decide how much or little it wants to process (drink). It decides on this amount on the fly, as it processes it. (“As mentioned above, each task execution unit may dynamically self-allocate Sub-tasks by determining a granularity unit anew for each Sub-task, each Sub-task thus having a different size. As one example, granularity may vary with time, starting with a large granularity which decreases with time. This helps to ensure that all task execution units finish substantially concurrently.” EX1003, 6:1-7)

42. The above cited quote implies that not only does each processor (which runs a task execution unit) have full control over how much data in total it will process; it even has control over how much data it will process in each iteration, and that it can decide on these amounts on the fly.

43. Thus, unlike other systems which partition data in advance, the partitioning process in Chow is performed by the data processors, during the data processing step itself. There are advantages and disadvantages of this approach. One advantage is that it might allow more heterogeneity across the processors. However,

one disadvantage is that partitions cannot be easily referred to via globally agreed upon identifiers.

44. After an initial step of processing, the Chow patent discusses how subsequent steps of processing may be done, using a psort model for redistributing data across processors (EX1003, 8:9-52), the round robin model for redistributing data across processors (EX1003, 8:53 – 9:29) and the Shared Sort model (EX1003, 9:31-9:62). The first two of these models for data redistribution do not use the straw model, while the third one does. However, all occur after an initial step of processing that uses the straw model (“the multi-level processing paradigm of the invention uses the straw model to allocate a first level of work among the task execution units. Then, results of the first-level work are pooled and redistributed among the task execution units” EX1003, 7:38).

#### **B. Reiffin (EX1004)**

45. The Reiffin patent has an earlier filing date (1994) and indeed is designed for an antiquated computer architecture in which each computer only has a single processor. The primary innovation of Reiffin is how to allow that single processor to give a user who is interacting with the system on a foreground task the impression that they have the full attention of the single processor, while in the background the processor is also working on a separate compute-intensive task. The goal is for the

user to be entirely unaware of the background task and observe no degradation in experience when interacting with the multi-tasking processor.

46. Reiffin achieves this goal by breaking down the compute-intensive task into smaller sub-tasks that can be processed by many different computers, and then further breaking down the sub-task into smaller slices that can run during short time slices when it has access to the CPU as it alternates with a foreground task. In aggregate, the compute-intensive task can eventually complete, even though it is only running in short time CPU slices in the background of the computers participating in the effort.

47. Reiffin does not discuss processing records from a table. Its focus is on compute-intensive tasks rather than data-intensive tasks. The goal is to avoid allowing a CPU to sit idle while a foreground task stalls for any reason. It does not specify how files containing records would be input to these tasks. Instead, it gives the example of processing a matrix of floating point numbers (EX1004, 3:15) which are well known to be a compute-intensive task, and not generally done in record-based systems.

### **C. Kurowski (EX1004)**

48. The Kurowski patent is designed to be a distributed computing system that focuses on computation intensive tasks (see first line of Abstract: “A distributed computing system achieves a highly distributed environment where very large computation intensive tasks are broken down into thousands of sub-tasks and then

distributed to thousands of clients running on a variety of computers across the Internet”).

49. The basic idea is that these large, computation-intensive tasks are broken into many small sub-tasks such that over time, as all of these sub-tasks complete, the final result will be identical to the result of processing the original large computational-task.

50. Kurowski included a task server that keeps track of all of the computers who are currently participating in the distributed system and assigned sub-tasks to them, a file server that provides application modules to these participating computers and an application server that helps provide state information for these modules to the participating computers and receive outputs from them. Having servers dedicated to these functions helps to control the chaos that is anticipated with computers currently joining and leaving the network of participating computers in Kurowski.

51. The Kurowski patent also discusses embodiments that track performance and status information to ensure proper functionality of the system.

## **V. GROUND 1: CLAIM 1 IS NOT OBVIOUS IN VIEW OF CHOW; NOT BY ITSELF, NOR IN COMBINATION WITH REIFFIN**

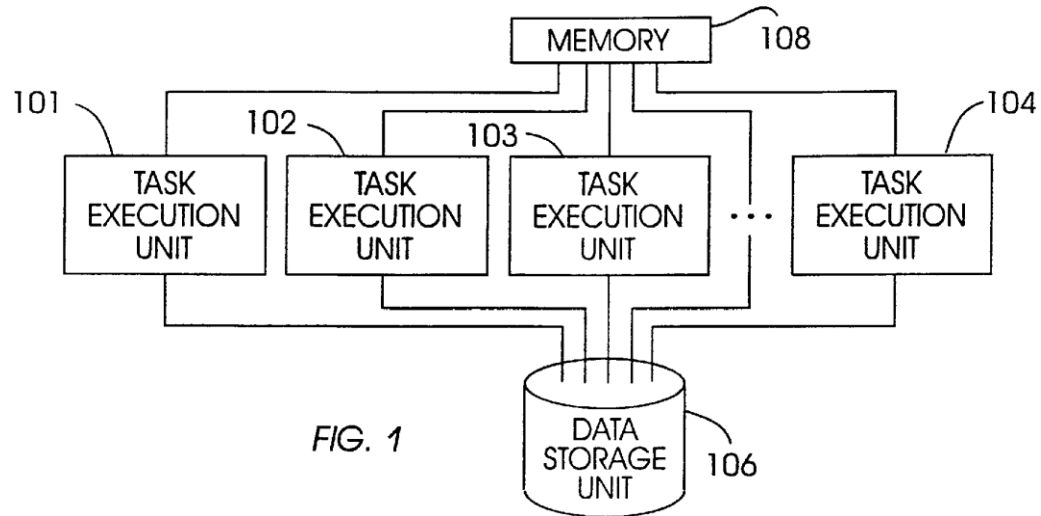
52. Chow was designed for a completely different type of computing setup than the ‘582 patent. It therefore makes some technical assumptions that are invalid in the context of the ‘582 patent. Attempting to apply Chow in the context that the

‘582 patent targets would yield a poorly functioning system that nobody would want to use, and certainly would not succeed in achieving the claims of the ‘582 patent.

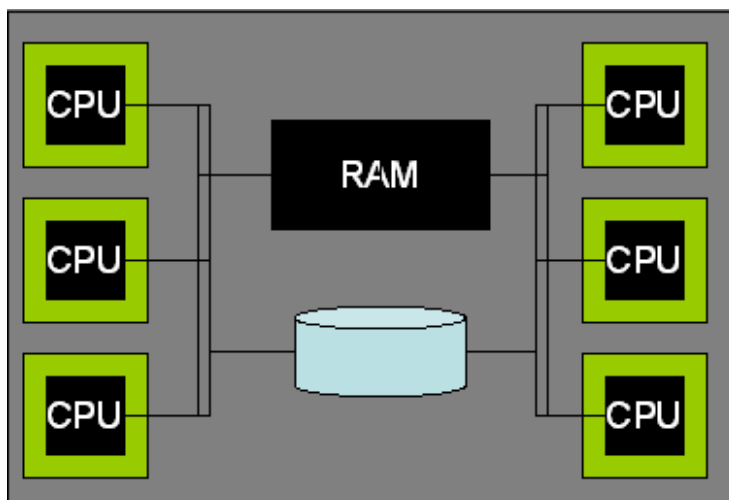
53. The context of the ‘582 patent is clear in the wording of its claims. For example, claim 1 uses key phrases such as “input file”, “subdividing records ... into ... partitions”, and “distributing descriptions.” I will explain how these phrases make the context of the ‘582 patent clear and why Chow fails in this context shortly. However, first I will explain the computing setup that Chow was designed for in more detail.

**A. Aspects of Chow focus on a single computer system**

54. The petition makes many errors and shows a gross ignorance of the definition of basic, widely known terms in computer science. Some aspects of Chow focus on multiprocessing systems consisting of a single computer with multiple processors executing programs in parallel. For example, the very first figure in the patent which gives a high level overview of the architecture of such a system:



55. This figure makes it clear that every task execution unit has direct access to the same memory (at the top of the figure). Each computer generally has its own memory (RAM). The fact that they are all connected to the same memory provides a strong indication that they are running on the same machine. This architecture is called a shared-memory architecture, and is explained well by Hellerstein et. al. in their “Architecture of a Database System” reading used in many computer science classes across the world (including the ones that I teach):



56. Hellerstein et al. explain in Chapter 3 of the cited reading, which includes the above figure, that the CPUs in the figure are processors, RAM is the memory that they all share, and the blue cylinder represents the place where the data is stored for the database.

57. Note that Chow also uses a cylinder to represent the data storage unit (labeled as 106 in their figure which was copied above). The CPU processors in Hellerstein correspond to the task execution units in Chow (labeled 101 through 104). And the RAM from Hellerstein corresponds to the memory from the Chow figure (labelled 108). Note that in both figures, the processors all share (and are directly connected to) the same memory and storage.

58. In summary, it is obvious that aspects of Chow are designed for shared-memory systems running on a single computer. Based on the wording in the petition, it does not appear that either side in the case disputes this fact.

**B. The ‘582 patent is designed for a distributed computer system consisting of many computers**

59. On the very first page of the ‘582 patent, there is a section entitled the OBJECTS OF THE INVENTION. The very first sentence of that section states: “The principal object of the present invention is to enable the decomposition of a certain type of linear processes that currently use a single computer, into equivalent parallel processes that can efficiently use any number of potentially heterogeneous computers,

taking the available capacity of each of these computers into account while optimizing execution.”

60. This sentence, without any ambiguity, makes it very clear that the entire purpose of the ‘582 invention is to allow multiple computers to process a task in parallel with each other.

61. Again, the petition does not appear to dispute this very obvious point. The petition states clearly that the ‘582 patent is in the field of “distributed computing systems and methods.”

**C. Distributed systems pay communication costs to distribute information across computers in the system**

62. Each computer in a distributed system generally has its own memory (and often also its own disk). If a computer in the distributed system wants to know information in the memory (or disk) of a different computer, it generally needs to send a message to the first computer to ask for this information, and wait for a response back in the form of a message with the designed information. The length of time this takes depends on the speed of the network connection between the two computers, and is not instantaneous.

63. In contrast, if a processor within a computer wants to get information stored in its own memory, this is typically much faster, often by orders of magnitude (especially in the time frame of the patents under discussion).

64. Therefore, throughout the history of computer science, distributed data processing systems have always been designed with this basic fact in mind --- that sending messages between computers can be slow. The goal in distributed database processing has always been to maximize the amount of work that can be done across the different computers in parallel, while minimizing slow and costly messages across machines that interrupt the otherwise rapid processing that a computer is able to perform.

**D. The ‘582 patent explicitly tries to reduce required communication between computers**

65. Since the ‘582 patent is focused on distributed systems, it introduces ways to reduce communication between computers. One of these ways, as stated clearly right at the beginning of its claim 1, is that it “logically subdivide[s] records ... into a plurality of partitions” and then “distribut[es] descriptions of all of said partitions to each of a plurality of subtask processors.”

66. The ‘582 patent purposefully creates data partitions in advance so that it can succinctly distribute descriptions of these pre-existing partitions to other processors and reduce communication cost.

67. In contrast, Chow’s straw model does not create partitions in advance. Rather each processor dynamically chooses a size (granularity) chunk of data to work on based on the current conditions at that processor. As stated above, this would yield additional communication overhead in a distributed system.

**E. The petition appears to misunderstand the basic meaning of the word “record” in data processing systems**

68. The ‘582 patent clearly creates data partitions in advance. The Chow straw model clearly does not. Neither party appears to dispute this basic fact. Nonetheless, the petition argues that Chow fulfills the element 1a of “logically subdividing records of said input file into a plurality of partitions” by quoting a section from the “related art” section from page 1 which is distinct from what Chow actually does and then another section from Chow that describes partitioning a task into sub-tasks.

69. A task is an action, not a data item. Yet claim 1a clearly requires partitioning of data. This is indicated by the phrase “records of said input file.” The term record is a fundamental term in the field of database systems that is taught on day 1 of just about every undergraduate class on database systems, including my own class: CMSC 424 at UMD. A record refers to a row in a database table (or relation). It is fundamentally a data item. It is shocking that the petition would fail to understand such a basic term.

70. Furthermore, it cannot be argued that when Chow talks about partitioning tasks, they really mean to say that they are partitioning data. This is obviously not true since Chow creates the subtasks in advance, and the whole point of Chow’s straw model is to partition data later, on the fly, based on the correct granularity determined during processing.

71. In short, Chow not only fails to disclose claim 1a, but rather the whole point of Chow's straw model is to do the exact opposite!

**F. The petition makes the same misunderstanding when discussing Reiffin**

72. The petition's discussion of Reiffin by claim 1a makes a similar mistake as its discussion of Chow. It states:

Reiffin also discloses claim [1a]. Ex.1022 ¶ . For example, Reiffin discloses that "[t]his invention relates to computer networks, and more particularly, to a distributed parallel processing network system wherein a large compute-intensive task may be partitioned into subtasks." Ex.1004 (1:8-10). Reiffin further discloses: "If a computer has a task which is both large and compute-intensive, the task is partitioned into a plurality of smaller subtasks to be executed simultaneously in parallel by the other computers of the network. Each of these subtasks is assigned its respective identifier" (Ex.1004 (2:19-23)) and "[t]he workstations or personal computer originally given such a task may partition the data and program code into a plurality of subtasks. A plurality of subtask identifiers corresponding to the partitioned subtasks are stored in a network directory of a disk drive of one of the computers to constitute a pool or queue identifying remote subtasks to be executed" (Ex.1004 (4:64-5:3)). Reiffin, therefore, discloses determining file allocation and logically subdividing records of said input file into a plurality of partitions. Ex.1022 ¶ 146.

Petition, 14-15

73. This quote from the petition again clearly focuses the partition of tasks into subtasks. Although it does make a brief mention of partitioning of data in the above quote, this is actually the only place in the entire patent where the word data appears. And the data here is not in the format of records. Rather as inputs to a

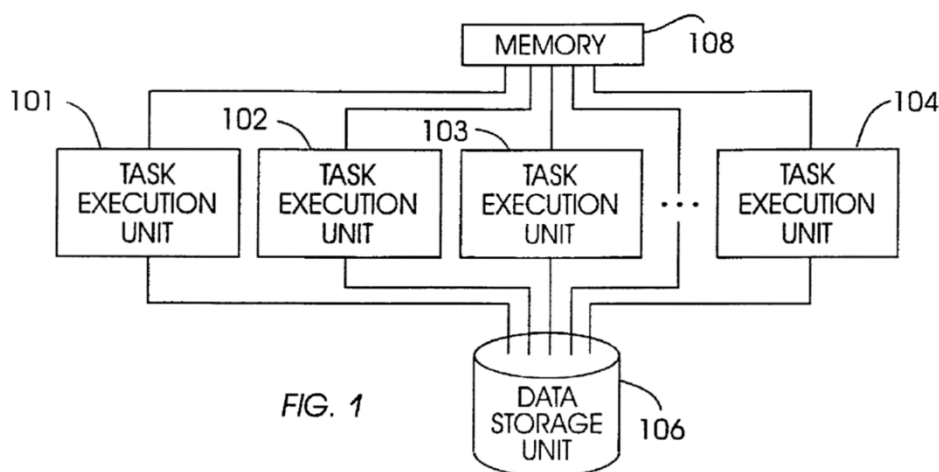
computational task such as a matrix of floating point numbers (the example from 3:15). No mention is made of an input file or records from a database table because Reiffin is focused on partitioning computing tasks.

**VI. GROUND 2: CLAIM 1 IS NOT OBVIOUS IN VIEW OF CHOW; NOT BY ITSELF, NOR IN COMBINATION WITH REIFFIN AND KUROWSKI**

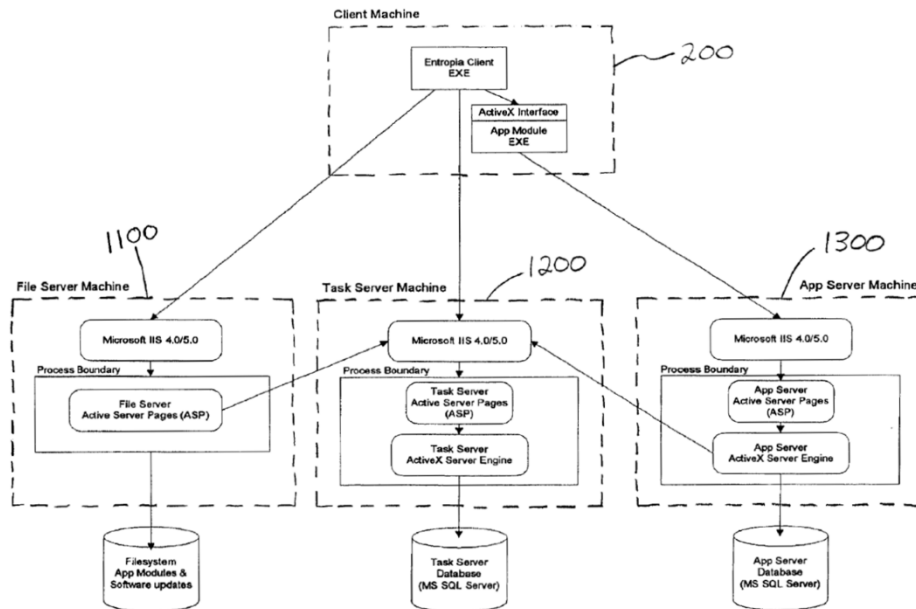
74. Ground 2 is identical to ground 1, except that now a practitioner would have to combine three different types of systems: Chow, Reiffin and Kurowski.

75. The petition attempts to argue that the combination of these systems are achievable because “architectures of Chow and Kurowski are very similar.” This statement borders on the absurd.

76. The petition attempts to argue that Chow and Kurowski have similar architectures by comparing figure 1 from Chow with figure 3 from Kurowski, reproduced below.



EX1003, Fig. 1



77.

FIG. 3

EX1005, Fig. 3.

78. Aside from having rectangles and cylinders in approximately the place in the figure, these figures could not be more different. The middle row of rectangles in the Chow figure are all equal task execution units, working in parallel to perform a task. The middle row of rectangles in the Kurowski figure are not equals --- rather three different components running fundamentally different tasks --- a file server, and task server, and an app server. Furthermore, they are running on the same computer in Chow, and explicitly labeled as separate machines in Kurowski. The top rectangle in Chow is a shared-memory that every task execution unit can read from. The top

rectangle in Kurowski is a separate client machine that is not directly accessible from any of the other rectangles in the figure.

79. In short, the figure in Chow is showing a traditional shared memory architecture. The figure in Kurowski is showing a shared-nothing architecture --- the extreme opposite of a shared memory architecture.

**A. The petition makes the same misunderstanding regarding claim 1a when discussing Kurowski as it did when discussing Chow and Reiffin**

80. As discussed above, the ‘582 patent requires creating partitions of records in advance of parallel processing. Records are data, and the patent thus requires data partitions to be created in advance.

81. The petition misunderstands the definition of the word record, and appears to understand the partitioning of tasks to meet the ‘582 patent’s claim of partitioning of records. As described above, it made this mistake when discussing Chow, and made the same mistake when discussing Reiffin.

82. It makes the same exact mistake when discussing Kurowski. When discussing claim 1a, the petition states:

To the extent the Board disagrees that this limitation is disclosed by the Chow/Reiffin combination, Kurowski discloses claim [1a]. Ex.1022 ¶ 259. For example, Kurowski discloses that “[t]he client 200 is one of the pieces in the larger picture of how the distributed computing system 100 described herein provides a highly distributed environment for large scale computing problems. One goal of all these different pieces is to enable the system to take on a very large computing problem, break it down into thousands of tasks and distribute it to tens of thousands of computers across the internet so this large computation could be

performed by tens of thousands of parallel sub-computations and the results brought back from all these different clients 200.” Ex.1005 ([0192]). Kurowski, therefore, discloses automatically identifying a large computing task and partitioning it.

Petition, 38-39

83. This text is clearly only talking about partitioning “a large computing task,” rather than partitioning data.

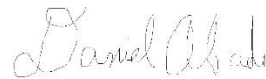
## **VII. CONCLUSION**

84. For the foregoing reasons, it is my technical opinion that the claims of the ‘582 Patent would not have been obvious over the cited references and that the Petition fails to demonstrate a reasonable likelihood of prevailing on at least Grounds 1 and 2 of the Petition.

## VIII. DECLARATION

85. I, Dr. Daniel Abadi, declare that all statements made herein on my own knowledge are true, and all statements made on information and belief are believed to be true. Further, I am aware that these statements are made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001.h. I declare under penalty of perjury that the foregoing is true and correct.

Executed on August 11, 2025, in College Park, Maryland.



---

Dr. , Daniel Abadi