

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.

Case No. IPR2025-00785

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

PATENT OWNER'S PRELIMINARY RESPONSE

Table of Contents

PATENT OWNER’S EXHIBIT LIST	v
I. INTRODUCTION	1
II. THE SPECIFICATION OF THE ‘582 PATENT	3
III. THE CLAIMS OF THE ‘582 PATENT	6
IV. CLAIM CONSTRUCTION	8
V. APPLICABLE LEGAL STANDARDS	12
VI. A PERSON HAVING ORDINARY SKILL IN THE ART	13
VII. PRIOR ART RELIED ON IN PETITION	13
A. Chow	13
B. Reiffin	17
C. Kurowski	18
VIII. GROUND 1 OF THE PETITION FAILS TO DEMONSTRATE A REASONABLE LIKELIHOOD OF SUCCESS	19
A. The Petition Fails To Demonstrate That Chow Alone Meets All Of The Limitations Of Claim 1 Of The ‘582 Patent	19
1. The Petition Erroneously Attempts To Combine Aspects Of Systems Disparaged By Chow With Aspects of Chow’s Proposed Solution In Attempting To Map Element [1a] To Chow	20
2. The Petition Presents Grossly Inconsistent Mappings Of Aspects Of Claim 1 To Chow	22
3. The Petition Has Failed To Explain How Chow Distributes Descriptions Of All Of Said Partitions To The Subtask Processors As Recited In Element [1b]	23
4. The Petition Has Failed To Explain How Chow Meets The Sequential Requirements Of Claim 1 of the ‘582 Patent ...	25
B. The Petition Fails To Demonstrate That Reiffin Alone Meets All Of The Limitations Of Claim 1 Of The ‘582 Patent	28
1. The Petition Fails To Explain How Reiffin Logically Subdivides Records Of A Preexisting Input File Into A Plurality Of Partitions As Recited In Element [1a]	28

2.	The Petition Fails To Explain How Reiffin Distributes Descriptions Of All Of Said Partitions To The Subtask Processors As Recited In Element [1b]	30
C.	The Petition Fails To Demonstrate That Claim 1 Is Obvious Over The Combination Of Chow And Reiffin	32
1.	Both Chow And Reiffin Are Deficient With Respect To Elements [1a] And [1b]	32
2.	The Petition’s Purported Motivation For Combining Chow And Reiffin Is Deficient	33
a)	The Fact That References Relate To The Same Field Standing Alone Is Insufficient To Establish A Motivation To Combine	35
b)	The Lack Of A Non-Conclusory Explanation “Why” Chow And Reiffin Would Have Been Combined Is Fatal	36
c)	A Reasonable Expectation Of Success Is Insufficient To Establish A Motivation To Combine.....	38
d)	The Testimony Of Petitioners’ Declarant Parrots The Words Of The Petition And Provides No Additional Explanation Or Citations Supporting The Alleged Motivation To Combine	39
3.	There Is No Nexus Between The Purported Motivation To Combine Chow And Reiffin And Each Of The Numerous Ways The Petition Alleges The References Could Be Combined.....	39
D.	Ground 1 Is Deficient.....	41
IX.	GROUND 2 OF THE PETITION FAILS TO DEMONSTRATE A REASONABLE LIKELIHOOD OF SUCCESS.....	41
A.	The Petition Fails To Demonstrate That Kurowski Alone Meets All Of The Limitations Of Claim 1 Of The ‘582 Patent.....	42
1.	The Petition Fails To Explain How Kurowski Logically Subdivides Records Of A Preexisting Input File Into A Plurality Of Partitions As Recited In Element [1a].....	42

2.	The Petition Fails To Explain How Kurowski Distributes Descriptions Of All Of Said Partitions To The Subtask Processors As Recited In Element [1b]	44
B.	The Petition Fails To Demonstrate That Claim 2 Is Obvious Over The Combination Of Chow, Reiffin, And Kurowski	45
1.	Both Chow And Reiffin Are Deficient With Respect To Elements [1a] And [1b]	45
2.	The Petition’s Purported Motivation For Combining Chow And Reiffin Is Deficient	46
a)	The Fact That Chow And Kurowski Are Similar Is Insufficient To Establish A Motivation To Combine...49	
b)	The Lack Of A Non-Conclusory Explanation Of “Why” Chow And Reiffin Would Have Been Combined Is Fatal	50
c)	The Testimony Of Petitioners’ Declarant Parrots The Words Of The Petition And Provides No Additional Explanation Or Citations Supporting The Alleged Motivation To Combine	51
3.	There Is No Nexus Between The Purported Motivation To Combine Kurowski With Chow and Reiffin And Each Of The Numerous Ways The Petition Alleges The References Could Be Combined.....	52
C.	Ground 2 Is Deficient.....	54
X.	THE DECLARATION OF PETITIONERS’ EXPERT IS ENTITLED TO LITTLE IF ANY WEIGHT.....	54
XI.	CONCLUSION.....	56

PATENT OWNER'S EXHIBIT LIST

Ex. 2001	https://www.unifiedpatents.com/faq
Ex. 2002	https://www.unifiedpatents.com/members
Ex. 2003	Reexamination Request challenging U.S. Patent 7,257,582 filed by Unified Patents (No. 90/015,337)
Ex. 2004	Reexamination Request challenging U.S. Patent 8,332,844 filed by Unified Patents (No. 90/019,938)
Ex. 2005	Petition filed in IPR2025-00931
Ex. 2006	Docket from <i>Intellectual Ventures I LLC et al. v. Southwest Airlines, Co.</i> , 7:24-cv-277 (WDTX)
Ex. 2007	www.txwd.uscourts.gov/for-attorneys/judge-albright-courtroom-faq/
Ex. 2008	Docket from <i>Intellectual Ventures I LLC et al. v. American Airlines, Inc.</i> , 4:24-cv-980 (EDTX)
Ex. 2009	Defendant American Airlines, Inc.'s P.R. 3-3 Invalidity Contentions in <i>Intellectual Ventures I LLC et al. v. American Airlines, Inc.</i> , 4:24-cv-980 (EDTX)
Ex. 2010	Defendant Southwest Airlines Co.'s Preliminary Invalidity Contentions in <i>Intellectual Ventures I LLC et al. v. Southwest Airlines, Co.</i> , 7:24-cv-277 (WDTX)
Ex. 2011	Agreed Scheduling Order <i>Intellectual Ventures I LLC et al. v. Southwest Airlines, Co.</i> , 7:24-cv-277 (WDTX)
Ex. 2012	Order Denying Motion to Sever and Stay Claims Against Viasat's In-Flight Connectivity Systems by Defendant American Airlines <i>Intellectual Ventures I LLC et al. v. American Airlines, Inc.</i> , 4:24-cv-980 (EDTX)
Ex. 2013	Declaration of Dr. Daniel Abadi
Ex. 2014	CV of Dr. Daniel Abadi

I. INTRODUCTION

The Petition challenges claims 1-14 of U.S. Patent No. 7,257,582 (the ‘582 Patent) under two grounds. Institution should be denied for at least the following reasons:

First, the Petition’s mappings of Chow to claim 1 in Ground 1 are grossly deficient, and fail to explain how Chow meets elements [1a], [1b] and [1c] of claim 1. The mappings fail to explain how Chow discloses the claimed elements “arranged as in the claim,” and inconsistently map the “partitions” recited throughout the claim to different disclosures in Chow. In addition, the Petition fails to explain how Chow meets the sequence of operations required by claim 1 in which the subdividing of an input file into a number of partitions greater than the number of subtask processors occurs before executing any of the subtasks on the partitions.

Second, the Petition’s mappings of Reiffin to claim 1 in Ground 1 are also deficient. In purporting to map Reiffin to Element [1a] which expressly refers to logically subdividing a preexisting input file, the Petition relies solely on Reiffin’s division of tasks and subtasks, but fails to offer any explanation whatsoever why, in the context of the claim language which clearly distinguishes between a file and subtasks, Reiffin’s division of tasks and subtasks equates to the logical the partitioning of the “preexisting input file” as recited in the claim language. Moreover, in purporting to map Reiffin to Element [1b], the Petition relies on one

of Reiffin's computers accessing "the next subtask identifier" waiting in a queue, but fails to explain how a computer accessing "the next subtask identifier waiting in the queue" equates with distribution of descriptions of all of said partitions to each processor, as claimed.

Third, the Petition's mappings of Kurowski to claim 1 in Ground 2 suffer from similar deficiencies. Again, rather than addressing the language of Element [1a] which expressly refers to logically subdividing a preexisting input file, the Petition simply points to Kurowski's division of "a very large computing problem into" tasks, and fails to offer any explanation whatsoever why Kurowski's division of "a very large computing problem" into tasks equates to the logical the partitioning of the "preexisting input file." In connection with Element [1b] which requires the distribution of descriptions of all of the partitions of the preexisting input file to subtask processors, the Petition points to Kurowski's web interface that allows users to view summary reports but fails to explain how such a user-interface equates to the machine-level distribution of descriptions of all partitions to the subtask processors as claimed.

Finally, the purported motivations set forth in the Petition for combining the references in Grounds 1 and 2 are limited to conclusory allegations that fall short of what is required to establish obviousness. Both Grounds 1 and 2 lack a non-conclusory explanation as to how or why the references would be combined to

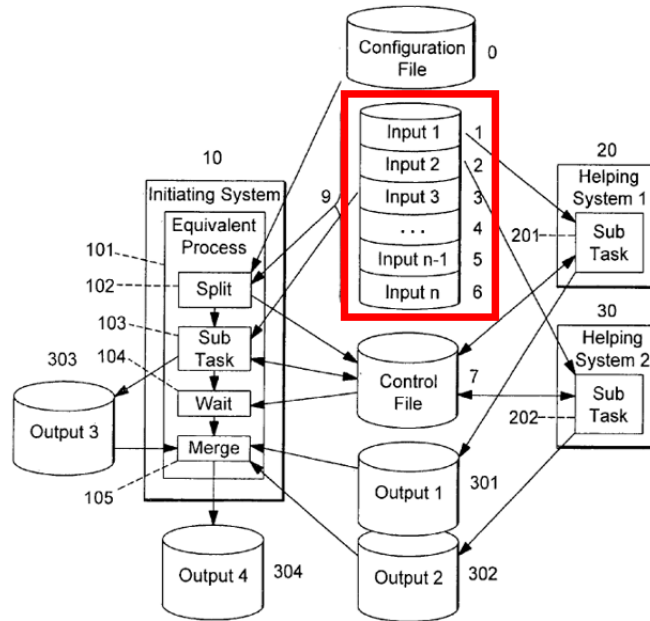
produce the claimed invention.

For all of these reasons, and as explained more fully below, institution should be denied.

II. THE SPECIFICATION OF THE ‘582 PATENT

The Summary of the ‘582 Patent describes “a method which distributes the load of a process that normally reads an input file sequentially and processes its records one by one between a plurality of potentially heterogeneous processors through the logical partition of the input and the activation of a plurality of sub tasks in said plurality of processors, each said sub task processing the partitions defined by said logical partition in a first come first serve basis.” Ex. 1001, 1:62-2:2. As explained in the specification and claims of the patent, the “input file” and the “sub tasks” are different. Ex. 2013, ¶ 69. More specifically, the input file includes records that are logically partitioned, while the sub tasks process the logical partitions.

An example of the logical partitioning of input files is depicted in Figure 1 of the ‘582 Patent, reproduced below with colored annotations added by Patent Owner. In Figure 1, “[i]nput files 9 are, thus, logically partitioned into n logical partitions where n is a number the split step 102 has decided upon. In FIG. 1, these logical partitions [red box] are numbered as 1 to 6.”



The specification explains that, in the content of the ‘582 Patent, a logical partition is a well-defined part of the input file:

A logical partition in this context is a well-defined part of the input or output. A very simple way to define a partition, which in many cases is the most efficient, would be to define the partitions as consecutive ranges on the input or output, ranging from one relative byte address to another relative byte address or from one relative track address to another. ...

Note that the logical partition does not rely on actual reading of the file. The actual reading of the file is reserved to the subtasks which read the partitions allocated to them. A splitting function may be allocating offset 0 to offset 100000 to a first partition, offset 100001 to 200000 to the second partition, offset 200001 to 300000 to the third partition and 300001 to the end of the file to a fourth partition. *Id.*, 3:36-59 (emphasis added).

Thus, in the parlance of the ‘582 Patent, the logical partitions correspond to well-defined portions of an input file, and the logical partitions are different from the sub tasks which, *inter alia*, read and process the logical partitions.

The specification also explains that the activation of the sub tasks occurs only after the logical partitioning of the input file (referred to in the ‘582 Patent as “the split step”) is finished. In this regard, the specification provides:

When the split step terminates, the various *Sub Tasks* (in this case: 103, 202,203) can be activated. This activation can be initiated by the split step 102 itself or by an external scheduler. *Id.*, 3:64-67 (emphasis added).

As explained more fully below, the requirement that the logical partitioning of the input file into partitions occur before any of the subtasks are activated is expressly reflected in the claims of the ‘582 Patent. Ex.2013, ¶¶ 31-32.

After being activated, each sub task is provided with a description of the subdivisions of the input file (reflecting the logical partitions) and uses such information “to allocate for itself and then process, an input partition that has not been allocated yet.” *Id.*, 3:67-4:3.

After the sub tasks have terminated, a merge step merges “the outputs of the various Sub Tasks into the result output file or files.” *Id.*, 4:40-61.

III. THE CLAIMS OF THE ‘582 PATENT

Claim 1 of the ‘582 Patent is the only independent claim, and recites as follows:

1. A method of effecting on a preexisting *input file* a computer-executable process comprised of a *plurality of subtasks*, the method comprising the steps of:
 - (a) automatically determining file allocation and *logically subdividing records of said input file into a plurality of partitions*;
 - (b) *distributing descriptions of all of said partitions* to each of a plurality of subtask processors
 - (c) simultaneously *executing at least a respective one of the subtasks* of the computer-executable process *in each of at least some of said processors on a respective one of the partitions with each subtask reading and processing the respective partition* so as to process the respective partition and produce respective subtask output and;
 - (d) *thereafter repeating step (c) in at least some of the subtask processors each with another unprocessed partition* on a first-come/first-served basis; and
 - (e) generating at least one output combining all of the subtask outputs and reflecting the processing of all of said subtasks.

Several aspects of the plain and ordinary meaning of claim 1 are worthy of note.

First, the preamble separately introduces an “input file” and “subtasks,” indicating that the two are *different*. The body of the claim goes on to explain the differences between the two. More specifically, step (a) of the claim recites that *the input file includes records that are logically partitioned*, while step (c) recites that *the subtasks process the logical partitions*. Moreover, in the parlance of the ‘582

Patent, the logical partitions correspond to “well-defined” portions of the preexisting input file such as “ranging from one relative byte address to another.” Ex. 1001, 3:36-41.

Second, step (a) of the claim recites “logically subdividing records” of “said input file” into “a plurality of partitions.”

Third, step (b) of the claim introduces “a plurality of subtask processors” which are used in steps (c) and (d) to execute subtasks on “the partitions.”

Fourth, in step (c) each of the subtask processors simultaneously executes a subtask on “a respective one of the partitions.”

Fifth, step (d) specifies “thereafter repeating step (c)” with “at least some of the subtask processors” on “unprocessed” partitions. Because it is a repetition of step (c), step (d) repeats the requirement of step (c) that some of the subtask processors each simultaneously execute a subtask on “a respective one of the partitions.”

Based on the above, the claim language requires that “the partitions” initially subdivided from the input file in step (a) be **greater in number than the number of subtask processors**. This result must hold because in the first iteration of step (c) **each of the subtask processors** simultaneously executes a subtask on “a respective one of the partitions” and in the repetition of step (c) called for by step (d) **some of the subtask processors** each simultaneously execute a subtask on “a respective one

of the partitions” (“unprocessed” partitions). Thus, in order for there to be sufficient partitions to meet the processing recited in steps (c) and (d), “the partitions” initially subdivided from the input file in step (a) must be greater in number than the number of subtask processors. See also, Ex.2013, ¶¶31, 33.

Finally, the claim requires that the subdividing of the input file in step (a) into a number of partitions greater in number than the number of subtask processors occur before executing any of the subtasks on the partitions in steps (c) and (d). The claim language requires this sequence because the “executing” of the subtasks in steps (c) and (d) is performed on “the partitions” that resulted from the subdividing recited earlier in step (a). This aspect of the claim language is consistent with the specification which discloses that “[w]hen the split step terminates, the various Sub Tasks (in this case: 103, 202, 203) can be activated.” *Id.*, 3:64-67; Ex. 2013, ¶ 32.

IV. CLAIM CONSTRUCTION

The claim limitations at issue must be construed under *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005). Claim language is not read in isolation. *Phillips*, 415 F.3d at 1315. Rather, the claim language, when read in light of the specification and the prosecution history, elucidates the meaning of the claim language.

As explained in the previous section, the plain and ordinary meaning of the claim language makes clear that a “preexisting input file” and “subtasks” are different. In this regard, the claim specifies that the input file includes records that

are logically partitioned, while the subtasks process the logical partitions. In addition, as explained in the previous section, the plain and ordinary meaning of the claim language requires that the subdividing of an input file into a number of partitions greater than the number of subtask processors occurs before executing any of the subtasks on the partitions. See, Section III, *supra*.

Under 37 C.F.R. § 42.104(b)(3), a petitioner must provide a clear articulation of “[h]ow the challenged claim is to be construed.” The Petitioners asserted that they “do not believe explicit construction is needed to resolve this petition.” Petition, 6. However, as explained more fully below, the Petition fails to explain how the relied on references meet the plain and ordinary meaning of the aspects of the claim language emphasized above. Even worse, the Petition is predicated on claim construction positions that cannot be reconciled with those advanced by Petitioners in the parallel district court litigations. In the district courts, Petitioners have taken the position that key claim terms of the ‘582 Patent are “indefinite under 35 U.S.C. § 112, ¶ 2” because they “fail to inform, with reasonable certainty, those skilled in the art about the scope of the invention.” Ex. 2009, 36-37 and Ex. 2010, 200-201. Yet, in this IPR, the Petitioners have abandoned those positions in favor of interpretations intended to support invalidity arguments in this IPR—without acknowledging, let alone justifying, the inconsistency.

As detailed below, Petitioners indefiniteness allegations in the district courts cannot be reconciled with Petitioners' position here that, under *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005), "Petitioners do not believe explicit construction is needed to resolve this petition." Petition, 6 (emphasis added). Under *Phillips*, claim terms are typically given their "ordinary and customary meaning" which "is the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention." *Id.*, at 1313. Petitioners' indefiniteness positions in the district courts are irreconcilable with those taken by Petitioners in this IPR because a claim term cannot be both "indefinite under 35 U.S.C. § 112, ¶ 2" because it fails "to inform, with reasonable certainty, those skilled in the art about the scope of the invention" and, at the same time, have an "ordinary and customary meaning" as understood by "a person of ordinary skill in the art in question at the time of the invention."

By way of example, in the district court, Petitioner Southwest Airlines alleges that "automatically determining file allocation" in claim 1 "is vague and undefined" because the criteria that "define "automatic" allocation" is unclear, and raises questions such as whether the allocation is "based on file size, content, metadata, or another heuristic," whether "manual intervention" can "be involved at any stage and still meet this limitation." Ex. 2010, 200-201. Petitioner American Airlines asserts that the same phrase is indefinite in the district court. Ex. 2009, 37. Similarly, in the

district court, Petitioner Southwest Airlines alleges that the phrase “logically subdividing records” is “also ambiguous,” and raises questions such as “[w]hat constitutes a logical subdivision,” “[d]oes the subdivision require a specific algorithm or data structure,” and [i]f different methods (e.g., fixed-size vs. content-based) could be used, which ones fall under the claim scope.” Ex. 2010, 200-201. Again, Petitioner American Airlines asserts that the same phrase is indefinite in the district court. By contrast, in this IPR, Petitioners purport to convey their “belief” that, under *Phillips*, no explicit construction is necessary because the very same claim terms can be given their “ordinary and customary meaning.” *Phillips*, 415 F.3d 1303, 1313. These positions are diametrically opposed. A claim term that is “vague and undefined” does not have any “ordinary and customary meaning” as understood by a POSITA. Congress did not create the IPR process to provide an accused infringer with a mechanism for simultaneously advancing irreconcilable positions in different forums.

Petitioner’s indefiniteness positions in the district court are not limited to the terms described above but extend to other terms in claim 1 of the ‘582 Patent. More specifically, in the district courts, Petitioners contend that a POSITA would not have been able to understand the meaning of (i) “simultaneously executing at least a respective one of the subtasks of the computer-executable process in each of at least some of said processors on a respective one of the partitions with each subtask

reading and processing the respective partition so as to process the respective partition and produce respective subtask output” and (ii) “repeating step (c) in at least some of the subtask processors each with another unprocessed partition on a first-come/first-served basis.” Ex. 2009, 36-37 and Ex. 2010, 200-201. Again, these distinct claim construction positions are diametrically opposed to ones taken by Petitioners in this IPR, and provide a further basis for denial of institution.

Under 37 C.F.R. § 42.104(b)(3), a petitioner must provide a clear articulation of “[h]ow the challenged claim is to be construed.” Here, the Petitioners failed to meet this burden by basing the Petition on claim constructions that Petitioners believe are incorrect. These aspects of the Petition favor denial of institution.

V. APPLICABLE LEGAL STANDARDS

The Board may only grant a petition for IPR where “the information presented in the petition ... shows that there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition.” 35 U.S.C. § 314(a); 37 C.F.R. § 42.108(c). Accordingly, the Petitioner bears the burden of showing this statutory threshold has been met. *See* Office Patent Trial Practice Guide, 77 Fed. Reg. 48,756, 48,756 (Aug. 14, 2012) (“The Board ... may institute a trial where the petitioner establishes that the standards for instituting the requested trial are met...”).

VI. A PERSON HAVING ORDINARY SKILL IN THE ART

The Petition alleges that a POSITA would have had a bachelor's degree in computer science, computer engineering, electrical engineering, or related field, and at least two years of experience in computer networking and parallel computing, or a person with a master's degree in one of the foregoing and at least one year of experience in the aforementioned fields. Additional education could substitute for professional experience, and vice-versa. Petition, 7. For purposes of this Patent Owner Preliminary Response only, Patent Owner does not contest Petitioner's proposed definition of a POSITA. However, Patent Owner reserves the right to contest Petitioner's proposed definition of a POSITA should this challenge be instituted.

VII. PRIOR ART RELIED ON IN PETITION

Ground 1 of the Petition alleges that claims 1-14 of the '582 Patent are "[o]bvious over Chow (Ex. 1003) in view of Reiffin (Ex. 1004)." Petition, 7. Ground 2 alleges that claims 1-14 of the '582 Patent are "[o]bvious over Chow (Ex. 1003) in view of Reiffin (Ex. 1004) and further in view of Kurowski (Ex. 1005)." *Id.* Each of these references is discussed below.

A. Chow

Chow begins with a "Description of the Related Art." There, Chow disparages systems that partition data by logically dividing the data *a priori* so that

each thread can process a separate portion of the data. Chow explained that such systems resulted in a “significant problem” known as “workload skew.” From Chow:

Reducing task completion time is especially challenging in “multiprocessing environments” ... Typically, workload is apportioned by somehow logically dividing the data to be processed. For example, a block of data might be divided evenly into a number of parts equal to the number of available threads, so that each thread can independently process a separate portion of the data. This is called “static” or “a priori” partitioning.

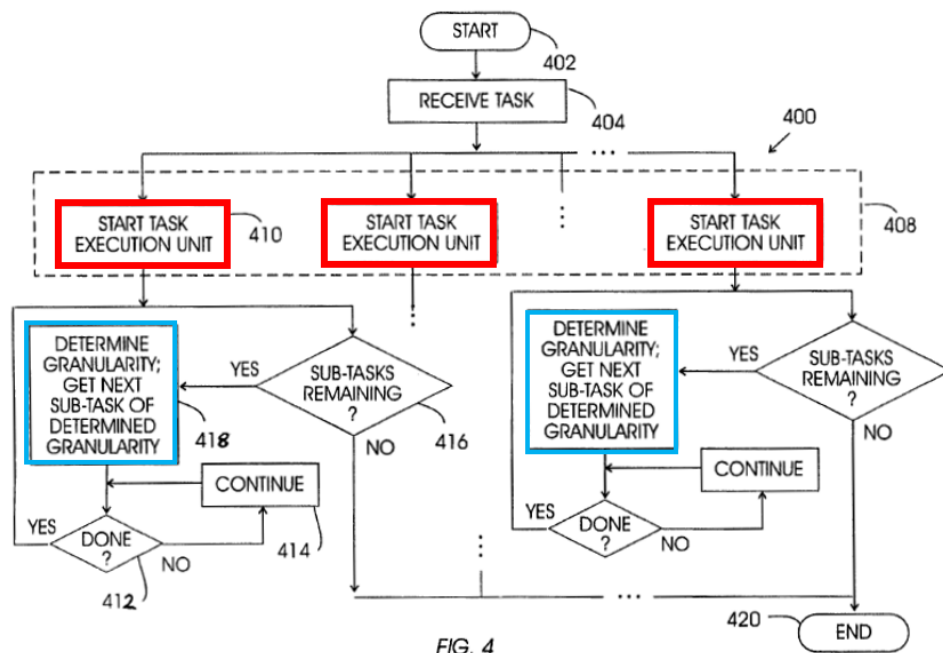
Such apportionment of work is rarely completely balanced. One reason is that the extent of the data may be difficult to ascertain, or the data’s processing time may be otherwise difficult to predict. Furthermore, even equisized data subparts may require comparatively different processing times. As a result of this lack of balance, some threads finish early, and other threads labor for longer periods. Still, the overall job is not complete until all threads have finished, including the last to finish its respective task. Thus, even though some threads complete their work early, other threads linger on, delaying completion of the ultimate task. One name for this phenomenon is called “workload skew.” For users seeking faster execution of their multi-thread application programs, workload skew is a significant problem area. Ex. 1003, 1:26-58 (emphasis added).

In contrast to systems that partition data by logically dividing the data *a priori*, Chow proposed a “multiprocessing system [that] operates like multiple people drinking water from a common glass through individual straws—although each drinker works independently, all finish simultaneously, thus completing the aggregate task as expeditiously as possible.” *Id.*, 2:15-20. Among other advantages,

Chow explained that this approach resulted in “minimum workload skew among independently operating processing elements.” *Id.*, 2:44-45.

Rather than logically dividing all of the data to be processed before execution of the subtasks begins, in Chow’s solution the data is divided into subtasks after processing by the task execution units commences. As Chow explains, after a task execution unit completes a subtask, “the task execution unit 200 in step 418 determines a unit of granularity for the next sub-task to be processed, and then, using the determined granularity, obtains the sub-task from the aggregate task. ... The size of the next sub-task to be processed is dictated by a corresponding granularity unit, also determined [by the task execution unit 200] in step 418.” *Id.*, 5:29-37 (emphasis added); *see also*, Ex. 2013, ¶ 43.

Chow’s division of an aggregate task into subtasks after processing by the task execution units commences is depicted in Figure 4 of Chow (reproduced below with colored annotations added by Patent Owner).



Chow explains that the process of Figure 4 is initiated in task 402, and “step 404 receives the aggregate task.” *Id.*, 4:7-8. Each task execution unit “starts in step 410 [red boxes].” *Id.*, 4:24. Significantly, no logical division of the aggregate task is shown between receipt of the aggregate task in step 404 and the starting of the task execution units in step 410. Rather, after the task execution units start in step 410, “step 416 asks whether any work from the aggregate task remains to be processed” and “if step 416 finds work remaining,” the task execution unit 200 in step 418 [blue boxes] determines a unit of granularity for the next sub-task to be processed, and then, using the determined granularity, obtains the sub-task from the aggregate task.” *Id.*, 4:24- 33. Thus, in Chow’s solution, *partitioning of each*

subsequent sub-task from the aggregate task occurs after each task execution unit completes a prior sub-task.

Simply put, Chow's proposed solution does not logically divide a "preexisting input file" into all of its partitions before (or "*a priori*") execution of the subtasks. On the contrary, Chow eschewed such an approach because "the data's processing time may be ... difficult to predict" and "even equisized data subparts may require comparatively different processing times." Ex. 1003, 1:44-48. Instead, Chow performs partitioning of each subsequent sub-task from the aggregate task only after each task execution unit completes a prior sub-task. Ex. 2013, ¶¶ 43, 65-68, 70.

B. Reiffin

Reiffin discloses that "[i]f 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. Reiffin's "subtasks" let an idle workstation fetch a self-contained code module, run it, and return results. Nothing in Reiffin discloses logically subdividing records of "a preexisting input file" into "a plurality of partitions" wherein each partition is a "well-defined" part of the input file such as "ranging from one relative byte address to another relative byte address." Ex. 1001, 3:36-41; Ex. 2013, ¶¶ 47, 73.

C. Kurowski

Kurowski discloses a system “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.” Ex. 1005, Abstract. “The idle CPU time of each of these thousands of client computers is used to perform these computations by running custom application modules in a low priority. A task server keeps track of information associated with each of the clients and uses the information to assign one or more tasks associated with a computing problem to each client computer.” *Id.*

Nothing in Kurowski discloses logically subdividing records of “a preexisting input file” into “a plurality of partitions” wherein each partition is a “well-defined” part of the input file such as “ranging from one relative byte address to another relative byte address.” Ex. 1001, 3:36-41; Ex. 2013, ¶¶ 81, 83. Also absent from Kurowski is the distribution of descriptions of all of the partitions of such a preexisting input file to subtask processors. Instead of distributing all of such descriptions to Kurowski’s thousands of processors, Kurowski simply assigns one task at a time to each client and provides it with the code and input needed to complete that individual computation. The Petition’s reliance on paragraph [0188] of Kurowski—which describes a web interface that allows users to view summary

reports — has no relevance to the machine-level distribution of descriptions of all partitions to the subtask *processors* as required by claim 1 of the ‘582 Patent.

VIII. GROUND 1 OF THE PETITION FAILS TO DEMONSTRATE A REASONABLE LIKELIHOOD OF SUCCESS

In identifying the “Grounds Raised,” the Petition describes Ground 1 as alleging that claims 1-14 of the ‘582 Patent are “[o]bvious over Chow (Ex. 1003) *in view of* Reiffin (Ex. 1004).” Petition, 7 (emphasis added). However, at the end of Ground 1, the Petition alleges that claim 1 is “obvious over *Chow and Reiffin separately* and *over their combination*.” Petition, 19 (emphasis added). Thus, in actuality, Ground 1 is not a single ground, but rather comprises multiple Grounds separately alleging that: (i) claim 1 is obvious over Chow alone, (ii) claim 1 is obvious over Reiffin alone, and (iii) claim 1 is obvious over some combination of Chow and Reiffin. As explained below, the Petition has failed to demonstrate a likelihood of success with respect to any of these theories.

A. The Petition Fails To Demonstrate That Chow Alone Meets All Of The Limitations Of Claim 1 Of The ‘582 Patent

The Petition’s mappings of claim 1 to Chow alone in Ground 1 are deficient for at least four reasons. First, the Petition’s mappings of claim 1 to Chow fail to explain how Chow discloses the claimed elements “arranged as in the claim.” Second, the Petition’s mappings of claim 1 to Chow are plagued with gross inconsistencies. Third, the Petition’s mappings fail to explain how Chow discloses

“distributing descriptions of all of said partitions” to the processors as recited in Element [1b] of claim 1. Fourth, the Petition fails to explain how Chow discloses the sequence recited in claim 1 which requires that the subdividing of an input file into a number of partitions greater than the number of subtask processors occur before executing any of the subtasks on the partitions.

1. The Petition Erroneously Attempts To Combine Aspects Of Systems Disparaged By Chow With Aspects of Chow’s Proposed Solution In Attempting To Map Element [1a] To Chow

In an attempt to map Chow to Element [1a], the Petition erroneously attempts to link statements from Chow’s “Description of the Related Art” where Chow disparages systems that partition data by logically dividing the data before (or “*a priori*”) execution of the subtasks begins, with aspects of Chow’s solution which performs partitioning of each subsequent sub-task only after each task execution unit completes a prior sub-task. However, in connection with the Petition’s allegation that Chow alone meets the limitations of claim 1, it was Petitioner’s burden to explain how Chow discloses the claim elements “arranged as in the claim.” *See, e.g., Lindemann Maschinenfabrik GMBH v. Am. Hoist & Derrick Co.*, 730 F.2d 1452, 1458, 1459 (Fed. Cir. 1984) (explaining that the prior art elements themselves must be “arranged as in the claim”).

More specifically, in purporting to map Chow to Element [1a] which recites “logically subdividing records of said input file into a plurality of partitions”, the Petition attempts to link aspects of (i) Chow’s description of “static” or “a priori” partitioning systems (disparaged by Chow) where “a block of data might be divided evenly into a number of parts equal to the number of available threads” and (ii) aspects of Chow’s disclosed solution which divide an aggregate task into sub-tasks. Petition, 13, citing Ex. 1003, 1:38-42 (disparaging “static” or “a priori” partitioning systems) and Petition, 14, citing Ex. 1003, 1:66-2:15 (summarizing aspects of Chow’s proposed solution). As explained above, in the ‘582 Patent, an “input file” and “subtasks” are different. In purporting to bridge this difference and allege that Chow alone meets Element [1a], the Petition erroneously attempts to link Chow’s discussion of dividing a block of data (which the Petition appears to analogize to an “input file”) in the systems disparaged by Chow with aspects of Chow’s disclosed solution that divide an aggregate task into subtasks. However, the Petition has failed to explain how these disparate aspects of Chow are arranged as claimed. Moreover, other than relying on an erroneous linkage between the “static” or “a priori” partitioning systems disparaged by Chow and Chow’s proposed solution, the Petition fails to offer any explanation whatsoever how Chow’s disclosure of dividing an aggregate task into subtasks equates to “logically subdividing records of said input file into a plurality of partitions” as recited in Element [1a] of claim 1.

For this reason alone, the Petition’s mapping of Chow alone to claim 1 is deficient.

2. The Petition Presents Grossly Inconsistent Mappings Of Aspects Of Claim 1 To Chow

In addition, the Petition fails to present a consistent mapping to Chow for the “partitions” recited throughout claim 1 of the ‘582 Patent. For example, in mapping Element [1a] to Chow, the Petition purports to equate Chow’s discussion of dividing a block of data (in the systems disparaged by Chow) “into a number of parts equal to the number of available threads” to “logically subdividing records of said input file into a plurality of partitions.” Thus, in mapping Element [1a] to Chow, the Petition analogizes “the number of parts” of the “block of data” (in the systems disparaged by Chow) to the “partitions” recited in the claim element. However, in mapping “said partitions” referenced in Element [1b] to Chow, the Petition does not reference back to the “partitions” identified in connection with the mapping of Element [1a] to Chow (e.g., “the number of parts” of the “block of data” in the systems disparaged by Chow) but instead appears to map “said partitions” in Element [1b] to entirely different disclosures in Chow, namely, Chow’s “common partition” or Chow’s division of an aggregate task into subtasks. Petition, 15-16.

In Element [1c], the inconsistencies in the Petition’s mappings of the claimed “partitions” get even worse. More specifically, in Element [1c], the Petition states:

... Chow discloses that “[g]enerally, the method sequence 400 is used to perform an aggregate database processing task, by breaking the task down into smaller units, and executing the units with multiple concurrently operating *task execution units*.” Ex.1003 (5:1-5). *A POSITA would understand Chow’s “units” to be the claimed “partitions.”* Ex.1022 ¶ 153. Petition, 16-17 (emphasis added).

Thus, in mapping “the partitions” referenced in Element [1c] to Chow, the Petition does not reference back to its mapping of “partitions” in Element [1a] to “the number of parts” of the “block of data” (in the systems disparaged by Chow), or its mapping of “said partitions” in Element [1b] to Chow’s “common partition” or Chow’s division of an aggregate task into subtasks, but instead identifies yet another aspect of Chow, namely, Chow’s execution units, as the claimed partitions. The Petition’s grossly inconsistent mappings of the “partitions” recited in claim 1 to Chow are further evidence that the Petition has failed to demonstrate that Chow alone meets all of the elements of claim 1.

3. The Petition Has Failed To Explain How Chow Distributes Descriptions Of All Of Said Partitions To The Subtask Processors As Recited In Element [1b]

In purporting to map the requirement in Element [1b] that “descriptions of all of said partitions” be “distributed” to the subtask processors to Chow, the Petition states only the following:

Chow teaches or suggests claim [1b]. Ex.1022 ¶ 149. Chow discloses that partitions are accessible by all subtask processors. For example, Chow discloses that “[t]he common partition is accessible by all task execution units 101-104.” Ex.1003 (9:37-38). As such, each of

the task execution units have access to information about the partition.” Ex.1022 ¶ 149. Chow further discloses: “[a] command is received to perform a machine-executed task divisible into multiple sub-tasks, i.e., an ‘aggregate task’. Then, multiple task execution units are independently operated on-demand to sequentially self-allocate and execute sub-tasks of the aggregate task. Ex.1003 (1:66-2:15). Petition, 15-16.

These statements fail to explain how the claim language is met.

Starting with the Petition’s references to “the common partition” set forth above, the language of claim 1 makes abundantly clear that a plurality of partitions are defined in Element [1a] and descriptions “of all of said partitions” (plural) are distributed in to the processors in Element [1b]. The Petition fails to explain how Chow’s disclosure of a common partition (singular) or providing task execution units access to information about only a single partition meets the claim language, which explicitly requires distribution of information describing a plurality of partitions.

Turning next to the Petition’s reference to Chow’s disclosure that “[a] command is received to perform a machine-executed task divisible into multiple sub-tasks, i.e., an ‘aggregate task’. Then, multiple task execution units are independently operated on-demand to sequentially self-allocate and execute sub-tasks of the aggregate task.” The Petition fails to explain how this disclosure equates to “distributing descriptions of all of said partitions” to the processors as recited in Element [1b]. The Petition fails to point to any express disclosure in Chow that

Chow's execution units are provided with "descriptions of all of said partitions" in connection with their sequential self-allocation of sub-tasks; nor does the Petition allege, let alone demonstrate, that this feature is inherent in Chow.

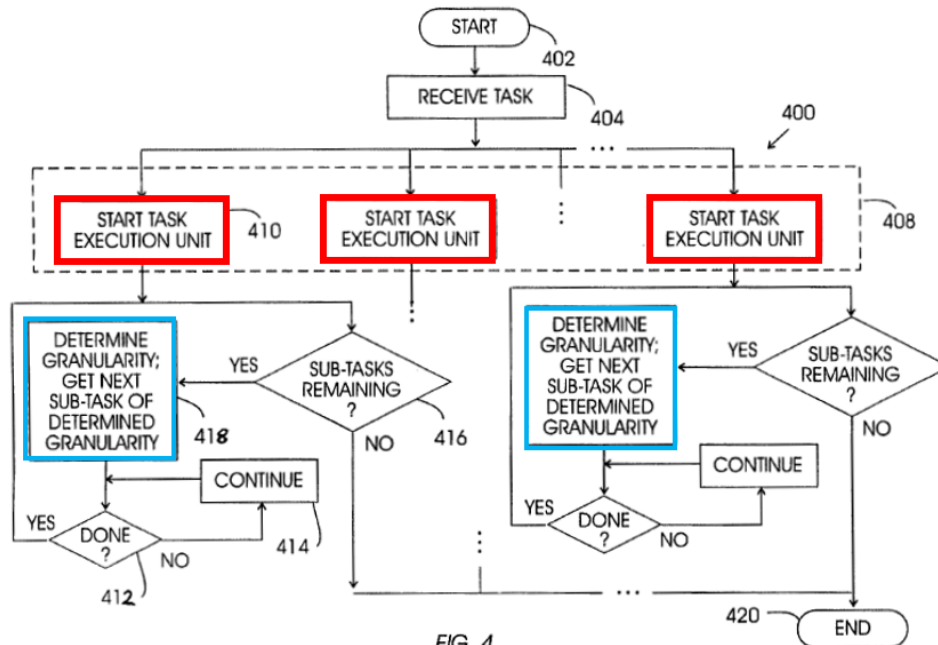
For this further reason, the Petition's mapping of Chow alone to claim 1 is deficient.

4. The Petition Has Failed To Explain How Chow Meets The Sequential Requirements Of Claim 1 of the '582 Patent

As explained in Sections III and IV, *supra*, the plain and ordinary meaning of the language of claim 1 requires that the subdividing of an input file into a number of partitions greater than the number of subtask processors occurs before executing any of the subtasks on the partitions. The Petition fails to explain how Chow discloses this sequence of operations. In purporting to map Chow to Element [1c] of claim 1, the Petition relies on Chow's "method sequence 400" and states only the following:

Chow discloses that "[g]enerally, the method sequence 400 is used to perform an aggregate database processing task, by breaking the task down into smaller units, and executing the units with multiple concurrently operating task execution units." Ex.1003 (5:1-5). A POSITA would understand Chow's "units" to be the claimed "partitions." Ex.1022 ¶ 153. Chow describes the concurrent execution of subtasks to produce a resulting output. Ex.1022 ¶ 153; Ex.1003 (7:46-48). Petition, 16-17.

Thus, the Petition fails to even allege -- let alone demonstrate -- that, in Chow, the subdividing of an input file into a number of partitions greater than the number of subtask processors occurs before executing any of the subtasks on the partitions. In any event, Chow's "method sequence 400" (relied on by the Petition to map Chow to Element [1c]) fails to meet the sequence of operations required by the claim language. As explained in Section VII, *supra*, rather than logically dividing all of the data to be processed before execution of the subtasks begins, in Chow's solution the data is divided into subtasks after processing by the task execution units commences. Ex. 2013, ¶¶ 43, 65-68, 70. Chow's "method sequence 400" is depicted in Figure 4 of Chow, which was discussed previously and is reproduced again below with colored annotations added by Patent Owner:



As Chow explains, each task execution unit “starts in step 410 [red boxes].” Ex. 1003, 4:24. As shown, “method sequence 400” lacks any logical division of the aggregate task prior to the starting of the task execution units in step 410. Rather, after the task execution units start in step 410, “step 416 asks whether any work from the aggregate task remains to be processed” and “if step 416 finds work remaining,” the task execution unit 200 in step 418 [blue boxes] determines a unit of granularity for the next sub-task to be processed, and then, using the determined granularity, obtains the sub-task from the aggregate task.” *Id.*, 4:24- 33. Thus, Chow performs partitioning of each subsequent sub-task from the aggregate task only after each task execution unit completes a prior sub-task. As a result, Chow’s “method sequence 400” (relied on by the Petition to map Chow to Element [1c]) fails to meet the sequence of operations required by claim 1 in which a preexisting input file is logically subdivided into partitions greater in number than the number of subtask processors before execution of the subtasks on the partitions occurs in steps [1c] and [1d].¹ Ex. 2013, ¶¶ 43, 65-68, 70.

For this further reason, the Petition’s mapping of Chow alone to claim 1 is deficient.

¹ Moreover, it is clear that Chow’s “p-sort” or “round robin sort” do not meet the sequence of operations required by Elements [1c]-[1d] because they rely on the same straw model discussed above in connection with Figure 4. Ex. 2013, ¶ 44.

B. The Petition Fails To Demonstrate That Reiffin Alone Meets All Of The Limitations Of Claim 1 Of The ‘582 Patent

The Petition’s mappings of claim 1 to Reiffin alone in Ground 1 are deficient for at least two reasons. First, in purporting to map Reiffin to Element [1a] which expressly refers to logically subdividing a preexisting input file, the Petition relies solely on Reiffin’s division of tasks and subtasks. The Petition fails to offer any explanation whatsoever why, in the context of the claim language which clearly distinguishes between a file and subtasks, Reiffin’s division of tasks and subtasks equates to the logical partitioning of the “preexisting input file” as recited in Element [1a]. Second, in purporting to map Reiffin to Element [1b], the Petition relies on one of Reiffin’s computers accessing “the next subtask identifier waiting in the queue.” However, the Petition fails to explain how a computer accessing “the next subtask identifier waiting in the queue” equates with distribution of descriptions of all of said partitions” to the processors as claimed.

1. The Petition Fails To Explain How Reiffin Logically Subdivides Records Of A Preexisting Input File Into A Plurality Of Partitions As Recited In Element [1a]

As explained in Section III, *supra*, the preamble of claim 1 of the ‘582 Patent separately introduces a “preexisting input file” and “subtasks,” indicating that the two are different. The body of the claim goes on to explain the differences between the two. More specifically, Element (a) of the claim recites that the input file includes records that are logically partitioned, while Element (c) the subtasks

process the logical partitions. Moreover, in the parlance of the ‘582 Patent, the logical partitions correspond to “well-defined” portions of the preexisting input file such as “ranging from one relative byte address to another.” Ex. 1001, 3:36-41.

Claim Element [1a] is directed to the logical partitioning of the “preexisting input file” introduced in the preamble, and recites as follows:

(a) automatically determining file allocation and logically subdividing records of said input file into a plurality of partitions

The Petition’s attempt to map Reiffin to Element [1a] is limited to the following discussion of tasks and subtasks:

Reiffin also discloses claim [1a]. Ex.1022 ¶ . (sic) 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.

Rather than addressing the language of Element [1a] which expressly refers to logically subdividing a preexisting input file, the Petition simply points to

Reiffin's division of tasks and subtasks followed by the conclusory assertion that Reiffin meets the claim language. The Petition fails to offer any explanation whatsoever why, in the context of the claim language which clearly distinguishes between a file and subtasks, Reiffin's division of tasks and subtasks equates to the logical the partitioning of the "preexisting input file" as recited in Element [1a]. Moreover, the specification of the '582 Patent makes clear that the file partitions resulting from Element [1a] correspond to "well-defined" portions of the preexisting input file such as "ranging from one relative byte address to another." Ex. 1001, 3:36-41. Completely absent from the Petition's purported mapping of Reiffin to Element [1a] is any explanation as to how Reiffin's subtasks correspond to portions of a preexisting input file such as "ranging from one relative byte address to another." Ex. 2013, ¶¶ 72-73.

For this reason alone, the Petition's mapping of Reiffin alone to claim 1 is deficient.

2. The Petition Fails To Explain How Reiffin Distributes Descriptions Of All Of Said Partitions To The Subtask Processors As Recited In Element [1b]

Element [1b] recites "distributing descriptions of all of said partitions" to the processors. In purporting to map Reiffin to Element [1b]:

Further, Reiffin teaches or suggests claim [1b]. Ex.1022 ¶ 150. Reiffin discloses that "[i]f 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. One of the network disk drives has a directory constituting a pool for storing all of the subtask identifiers, preferably in the form of a queue.” Ex.1004 (2:19-25). Reiffin therefore discloses providing descriptions of the partitions to each of the subtask processors. Ex.1022 ¶ 150. Petition, 16 (emphasis added).

Omitted from the Petition’s mapping is a more complete explanation of how Reiffin’s pool actually functions. Reiffin explains that “[i]f no remote subtask is being concurrently executed then the computer searches the network directory pool for the next subtask identifier waiting in the queue, copies from the originating computer the subtask program corresponding to said next identifier, and then proceeds with the execution of this remote subtask. Ex. 1004, 2:52-58 (emphasis added). Thus, in Reiffin, the only “description” of a “partition” accessed by the computer is “the next subtask identifier waiting in the queue.” *Id.*

The Petition fails to explain how a computer accessing “the next subtask identifier waiting in the queue” meets the plain language of Element [1b] which requires distribution of descriptions of all of said partitions to the processors. Moreover, there is no explanation why in Reiffin’s system it would even be useful for the computers to have distributed to them descriptions of all partitions. For example, there is no disclosure in Reiffin that each computer is performing any analysis of all of the “partitions” in the pool and then selecting the “partition” that is best suited for execution on the computer.

Simply put, the Petition fails to point to any express disclosure in Reiffin that Reiffin's computers are provided with descriptions of all of the "partitions." Nor does the Petition allege, let alone demonstrate, that this feature is inherent in Reiffin.

For this additional reason, the Petition's mapping of Reiffin alone to claim 1 is deficient.

C. The Petition Fails To Demonstrate That Claim 1 Is Obvious Over The Combination Of Chow And Reiffin

Turning now to the portion of Ground 1 alleging that claim 1 is obvious over the combination of Chow and Reiffin, this aspect of Ground 1 is deficient for several reasons. First, since both Chow and Reiffin are deficient with respect to Elements [1a] and [1b] of claim 1, the combination of Chow and Reiffin is necessarily deficient with respect to those claim elements. Second, the Petition's motivation to combine argument for Chow and Reiffin is deficient. Third, there is no nexus between the purported motivations to combine Chow and Reiffin and each of the numerous variations in which the Petition alleges that the references could be combined.

1. Both Chow And Reiffin Are Deficient With Respect To Elements [1a] And [1b]

As explained in Sections VIII(A)(1)-(3), the Petition has failed to explain how Chow meets the requirements of Elements [1a] and [1b]. Similarly, as explained in Sections VIII(B)(1)-(2), the Petition has failed to explain how Reiffin meets the

requirements of Elements [1a] and [1b]. Since the Petition has failed to show that either of the relied on references satisfies Elements [1a] and [1b], the Petition fails to demonstrate that claim 1 is obvious over the combination of Chow and Reiffin.

2. The Petition's Purported Motivation For Combining Chow And Reiffin Is Deficient

The Petition's argument purporting to establish a motivation for combining Chow and Reiffin is deficient. After alleging that Chow and Reiffin are "in the same field of endeavor" (Petition, 8) and "share" the same goal (Petition, 11), the Petition makes only limited and conclusory allegations with respect to any motivation to combine the references. The totality of the Petition's motivation to combine argument for Chow and Reiffin is limited to the following conclusory statements:

Chow is ready for improvement. Ex. 1022 ¶ 134. Chow focuses on multiprocessing systems consisting of a single computer with multiple processors executing programs in parallel. *Id.* Reiffin expands this concept to multiple networked computers. *Id.* A POSITA would have recognized that benefit of combining Reiffin with Chow would be increased computational power and the ability to solve larger problems. *Id.*

Combining Chow's multi-processing system with Reiffin's teachings of a distributed parallel processing network would have been the simple combination of known elements to obtain a predictable result and thus obvious to a POSITA. Ex.1022 ¶ 135.

It would have been routine to add the features of Reiffin to the system of Chow. Ex.1022 ¶ 136. A POSITA would have thus been motivated to combine Chow and Reiffin with a reasonable expectation of success. *Id.* Petition, 10-11.

The Petition's conclusory statements fail to establish the necessary motivation to combine the references. Instead of providing reason and evidence of *why* a POSITA would be motivated to combine Chow and Reiffin, the Petition offers only a conclusory allegation that "Chow is ready for improvement" -- without any evidence or explanation to support that conclusion or show *why* a POSITA would have considered Chow "ready for improvement." Petitioners' next two sentences which allege that "Chow focuses on multiprocessing systems consisting of a single computer with multiple processors executing programs in parallel" while "Reiffin expands this concept to multiple networked computers," is factually incorrect. Petition, 10-11 (emphasis added). Rather, Chow clearly disclosed "'multiprocessing environments'... where multiple computers, digital data processing units, threads, or other processes perform concurrent tasks." Ex. 1003, 1:27-30 (emphasis added). Given that Chow was already directed to the use of multiple computers to perform concurrent tasks, why would a POSITA have looked to Reiffin for this same concept? The Petition doesn't say.

Petitioners' allegations that the references each supposedly belong to the same field is insufficient to establish a motivation to combine. In sum, the Petition offers nothing more than unsupported conclusions regarding motivation to combine, which fail to add up to a reasonable likelihood of prevailing. *See Johns Manville Corp. v. Knauf Insulation, Inc.*, Case IPR2018-00827, Paper 9, at 10-13 (PTAB Oct. 16,

2018) (designated informative) (rejecting a similar same-field, beneficial-combination rationale and denying institution); *In re Magnum Oil Tools Int’l, Ltd.*, 829 F.3d 1364, 1380 (Fed. Cir. 2016) (a petitioner cannot satisfy its burden of proving obviousness by employing “mere conclusory statements,” but “must instead articulate specific reasoning, based on evidence of record, to support the legal conclusion of obviousness”).

a) The Fact That References Relate To The Same Field Standing Alone Is Insufficient To Establish A Motivation To Combine

Petitioners’ contentions that Chow and Reiffin purportedly relate to the same field or goal (Petition at 8, 11) do not by themselves establish a reason to combine the references. *Securus Techs., Inc. v Glob. Tel*Link Corp.*, 701 F. App’x 971, 976-77 (Fed. Cir. 2017) (non-precedential) (holding that “[s]uch short-cut logic would lead to the conclusion that any and all combinations of elements known in [a] broad field would automatically be obvious, without the need for further analysis”); *Microsoft Corp. v. Enfish, LLC*, 662 F. App’x 981, 990 (Fed. Cir. 2016) (non-precedential) (holding that “the Board correctly concluded” that the petitioner “did not articulate a sufficient motivation to combine” where the only reason given was “that the references were directed to the same art or same techniques”); *Johns Manville*, IPR2018-00827, Paper 9 at 10-11 (rejecting the argument that two references were similar and belonged to the same field as an insufficient reason to

combine those references). Petitioners' same field/same goal argument is "simply too conclusory" to show that a POSITA would have combined the references in the manner claimed. *Securus Techs., Inc.*, 701 F. App'x at 976.

b) The Lack Of A Non-Conclusory Explanation "Why" Chow And Reiffin Would Have Been Combined Is Fatal

The remainder of Petitioners' arguments for combining Chow and Reiffin fares no better. The Petition (at 11) makes the conclusory assertion that "[a] POSITA would have recognized that [the] benefit of combining Reiffin with Chow would be increased computational power and the ability to solve larger problems" -- but provides no evidence or explanation demonstrating how or why a POSITA would have recognized such a benefit. For example, Chow was already directed to "multiprocessing environments'... where multiple computers, digital data processing units, threads, or other processes perform concurrent tasks." Ex. 1003, 1:27-30. Given that combining multiple computers to increase computational power was a concept already present in Chow, why would a POSITA have looked to Reiffin for this same concept? The Petition doesn't say. Similarly, what "larger problems" existed that required computational power beyond what Chow already provided, and how would the supposed combination of Chow and Reiffin be any better at solving such "larger problems" than Chow alone? Again, the Petition doesn't say. *TP-Link Corp. Ltd. v. Netgear*, IPR2023-01429, Paper #10, 2024 Pat.

App. LEXIS 1168 (non-precedential) (finding the petition lacked sufficient explanation “why the integration would have been an improvement, let alone why Sinha was ready for improvement”).

In the present case, the lack of a “why” is dispositive: a determination of obviousness cannot be reached where the record lacks an “explanation as to how or why the references would be combined to produce the claimed invention.” *TriVascular, Inc. v. Samuels*, 812 F.3d 1056, 1066 (Fed. Cir. 2016); *Virtek Vision Int’l ULC, Inc.*, 97 F.4th at 886 (“These disclosures, however, do not provide any reason why a skilled artisan would use 3D coordinates instead of angular directions in a system”).

Petitioners’ conclusory assertion that “[c]ombining Chow’s multi-processing system with Reiffin’s teachings of a distributed parallel processing network would have been the simple combination of known elements to obtain a predictable result and thus obvious to a POSITA” (Petition, 11-12) also lacks supporting evidence or an explanation demonstrating how or why a POSITA would have combined the references. Given that Chow already disclosed an implementation using parallel computing, why would a POSITA have looked to Reiffin for “distributed” parallel computing? Again, the Petition doesn’t say. In addition, the Petition fails to explain how Chow’s parallel computers would be combined with Reiffin’s distributed parallel computers to solve “larger problems” that Chow alone would have been

unable to solve. For example, what aspects of Reiffin would be substituted into Chow in order to solve “larger problems” that Chow alone could not solve? Again, the Petition doesn’t say. All the Petition includes is the conclusory statement that Chow and Reiffin represented a “simple combination of known elements” that could be combined “to obtain a predictable result” -- which is too conclusory to support a motivation to combine.

c) A Reasonable Expectation Of Success Is Insufficient To Establish A Motivation To Combine

The Petition’s conclusory assertion that “[a] POSITA would have thus been motivated to combine Chow and Reiffin with a reasonable expectation of success” (Petition, 12) addresses the wrong issue and is also totally conclusory. “The obviousness inquiry does not merely ask whether a skilled artisan could combine the references, but instead asks whether ‘they would have been motivated to do so.’” *Adidas AG v. Nike, Inc.*, 963 F.3d 1355, 1359 (Fed. Cir. 2020) (quoting *InTouch Techs., Inc. v. VGO Commc’ns, Inc.*, 751 F.3d 1327, 1352 (Fed. Cir. 2014)). In other words, a reasonable expectation of success (“could combine references”) does not establish a motivation to combine (“would have been motivated to do so”). An expectation of success cannot cure the deficiencies of Petitioners’ conclusory allegations that a POSITA would have been motivated to combine Chow and Reiffin.

d) The Testimony Of Petitioners’ Declarant Parrots The Words Of The Petition And Provides No Additional Explanation Or Citations Supporting The Alleged Motivation To Combine

The cited testimony from Petitioner’s declarant, Dr. Shamos, does not cure the Petition’s deficiencies. Ex.1022 ¶¶ 134-136. The cited testimony largely repeats (word-for-word) the petition’s conclusory assertions. *Compare* Pet. at 11-12 with Ex. 1022, ¶¶134-136. For that reason, the testimony of Dr. Shamos is entitled to little if any weight. *Xerox Corp. v. Bytemark, Inc.*, IPR2022-00624, Paper 9 at 15-17 (PTAB Aug. 24, 2022) (designated precedential) (according “little weight” to declaration testimony that contains a verbatim restatement of a petition’s conclusory assertions without additional supporting evidence or reasoning).

In summary, the Petition’s argument purporting to establish a motivation for combining Chow and Reiffin is deficient.

3. There Is No Nexus Between The Purported Motivation To Combine Chow And Reiffin And Each Of The Numerous Ways The Petition Alleges The References Could Be Combined

In Ground 1, the Petition purports to map each of Elements [1pre], [1a], [1b], [1c], [1d] and [1e] to both Chow and Reiffin, and seems to argue that if any element is not met by one of the references, then the element is met by the other reference, and vice versa. This approach to Ground 1 encompasses at least 64 possible ways of combining the references. Just like the fact that flipping a coin six times can result

in 2^6 or 64 possible outcomes, Ground 1 purports to advance at least 64 different possibilities for combining Chow and Reiffin:

2 possibilities (Chow or Reiffin) for 1[pre] ×
2 possibilities (Chow or Reiffin) for 1[a] ×
2 possibilities (Chow or Reiffin) for 1[b] ×
2 possibilities (Chow or Reiffin) for 1[c] ×
2 possibilities (Chow or Reiffin) for 1[d] ×
2 possibilities (Chow or Reiffin) for 1[e] =

2^6 or 64 possible combinations

This “spaghetti on the wall” approach fails explain how Petitioners’ purported motivation for combining Chow and Reiffin fits with each of the at least 64 different possibilities for combining Chow and Reiffin.

For example, consider an exemplary permutation where disclosures from Chow are purportedly mapped to elements [1 pre], [1a] and [1c] and disclosures from Reiffin are purportedly mapped to elements [1b] and [1d]-[1e]. The Petition provides no explanation why a POSITA would have been motivated to combine these *specific* aspects of each of the references with one another. The Petition’s conclusory assertion that “A POSITA would have recognized that (sic) benefit of combining Reiffin with Chow would be increased computational power and the ability to solve larger problems” is not tied to this specific combination. For example, there is no explanation why or how taking aspects of Chow purportedly mapped to [1 pre], [1a] and [1c] and combining them with disclosures from Reiffin

purportedly mapped to elements [1b] and [1d]-[1e] would result in increased computational power or the ability to solve larger problems.

The failure of the Petition to articulate specific motivations to combine that are particularized for each of the at least 64 different possibilities for combining Chow and Reiffin advanced in Ground 1 renders the Petition's "spaghetti on the wall" approach for combining Chow and Reiffin deficient.

D. Ground 1 Is Deficient

In summary, for at least the reasons set forth above, Ground 1 fails to demonstrate a reasonable likelihood of success with respect to independent claim 1 of the '582 Patent because the Petition fails to demonstrate that: (i) claim 1 is obvious over Chow alone, (ii) claim 1 is obvious over Reiffin alone, or (iii) claim 1 is obvious over the combination of Chow and Reiffin. Since Ground 1 fails to demonstrate a likelihood of success with respect to independent claim 1, the Petition similarly fails to demonstrate a likelihood of success in Ground 1 with respect to claims 2-14, all of which depend directly or indirectly from independent claim 1.

IX. GROUND 2 OF THE PETITION FAILS TO DEMONSTRATE A REASONABLE LIKELIHOOD OF SUCCESS

In identifying the "Grounds Raised," the Petition describes Ground 2 as alleging that claims 1-14 of the '582 Patent are "[o]bvious over Chow (Ex. 1003) in view of Reiffin (Ex. 1004) and further in view of Kurowski (Ex. 1005)." Petition, 7 (emphasis added). However, within Ground 2, the Petition alleges that every

element of claim 1 is met by Kurowski alone.” Petition, 37-42. Thus, in actuality, Ground 2 is not a single ground, but rather comprises multiple Grounds separately alleging that: (i) claim 1 is obvious over Kurowski alone, and (ii) claim 1 is obvious over some combination of Chow, Reiffin and Kurowski. As explained below, the Petition has failed to demonstrate a likelihood of success with respect to any of these allegations.

A. The Petition Fails To Demonstrate That Kurowski Alone Meets All Of The Limitations Of Claim 1 Of The ‘582 Patent

The Petition’s mappings of claim 1 to Kurowski alone in Ground 2 are deficient because the mappings fail to explain how Kurowski meets Elements [1a] and [1b].

1. The Petition Fails To Explain How Kurowski Logically Subdivides Records Of A Preexisting Input File Into A Plurality Of Partitions As Recited In Element [1a]

As explained in Section III, *supra*, claim 1 of the ‘582 Patent differentiates a “preexisting input *file*” from “*subtasks*.” For example, the claim explains that *the input file includes records that are logically partitioned*, while *the subtasks process the logical partitions*. Claim Element [1a] is directed to the logical partitioning of the “preexisting input *file*” introduced in the preamble, and recites as follows:

(a) automatically determining file allocation and logically subdividing records of said input *file* into a plurality of partitions

The Petition's attempt to map Kurowski to Element [1a] is limited to the following discussion of tasks:

... 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 (emphasis added).

Rather than addressing the language of Element [1a] which expressly refers to logically subdividing a preexisting input file, the Petition simply points to Kurowski's division of “a very large computing problem” into tasks. The Petition fails to offer any explanation whatsoever why, in the context of the claim language which clearly distinguishes between a file and subtasks, Kurowski's division of “a very large computing problem” into tasks equates to the logical the partitioning of the “preexisting input file” as recited in Element [1a].

Moreover, the specification of the ‘582 Patent makes clear that the file partitions resulting from Element [1a] correspond to “well-defined” portions of the preexisting input file such as “ranging from one relative byte address to another.” Ex. 1001, 3:36-41. Completely absent from the Petition's purported mapping of

Kurowski to Element [1a] is any explanation as to how Kurowski's tasks correspond to well-defined portions of a preexisting input file such as "ranging from one relative byte address to another." Ex. 2013, ¶¶80-83.

For this reason alone, the Petition's mapping of Kurowski alone to claim 1 is deficient.

2. The Petition Fails To Explain How Kurowski Distributes Descriptions Of All Of Said Partitions To The Subtask Processors As Recited In Element [1b]

Element [1b] recites "distributing descriptions of all of said partitions" to the subtask processors. In purporting to map Kurowski to Element [1b]:

... Kurowski teaches or suggests claim [1b]. Ex.1022 ¶¶ 262-263. Kurowski discloses that "[s]pecifically, all machine, user, team, and task information, summaries of such information, or subsets of such information, may be made available through reports and interactive pages on an "integrated" web site." Ex.1005 ([0188]). As such, partition descriptions are distributed and made available to all of the processors. Ex.1022 ¶ 263. Petition, 39 (emphasis added).

The Petition fails to explain how Kurowski discloses the distribution of descriptions of all of the partitions of the input file to subtask processors. The Petition's reliance on Kurowski's web interface that allows users to view summary reports has no relevance to the machine-level distribution of descriptions of all partitions to each of the subtask processors as required by Element [1b].

The Petition fails to point to any express disclosure in Kurowski that Reiffin's computers are provided with descriptions of all of the "partitions." Nor does the Petition allege, let alone demonstrate, that this feature is inherent in Kurowski.

For this additional reason, the Petition's mapping of Kurowski alone to claim 1 is deficient.

B. The Petition Fails To Demonstrate That Claim 2 Is Obvious Over The Combination Of Chow, Reiffin, And Kurowski

Turning now to the portion of Ground 2 alleging that claim 1 is obvious over the combination of Chow, Reiffin, and Kurowski, this aspect of Ground 1 is deficient for several reasons. First, since Chow, Reiffin, and Kurowski are each deficient with respect to Elements [1a] and [1b] of claim 1, the combination of Chow, Reiffin, and Kurowski is necessarily deficient with respect to those claim elements. Second, the Petition's motivation to combine argument for Chow, Reiffin, and Kurowski is deficient. Third, there is no nexus between the purported motivations to combine Chow, Reiffin, and Kurowski and each of the numerous variations in which the Petition alleges that the references could be combined.

1. Both Chow And Reiffin Are Deficient With Respect To Elements [1a] And [1b]

As explained in Sections VIII(A)(1)-(3), the Petition has failed to explain how Chow meets the requirements of Elements [1a] and [1b]. Similarly, as explained in Sections VIII(B)(1)-(2), the Petition has failed to explain how Reiffin meets the

requirements of Elements [1a] and [1b]. Finally, as explained in Sections IX(A)(1)-(2), the Petition has failed to explain how Kurowski meets the requirements of Elements [1a] and [1b]. Since the Petition has failed to show that any of the relied on references satisfies Elements [1a] and [1b], the Petition fails to demonstrate that claim 1 is obvious over the combination of Chow, Reiffin, and Kurowski in Ground 2.

2. The Petition’s Purported Motivation For Combining Chow And Reiffin Is Deficient

The Petition’s argument purporting to establish a motivation for combining Chow, Reiffin and Kurowski is deficient. In arguing that it would have been obvious to combine all three references, the Petition relies on its reasoning in Ground 1 for combining Chow and Reiffin. Petition, 34-35 (“For all of the reasons previously stated it would have been simple to combine Chow’s multi-processing system with Reiffin’s teachings of a distributed parallel processing network ... “) (emphasis added). As explained in Section VIII(C)(2), *supra*, the Petition’s argument purporting to establish a motivation for combining Chow and Reiffin is deficient. Since the Petition fails to establish a motivation for initially combining Chow and Reiffin, the Petition necessarily fails to establish that it would have been obvious to add a further reference (i.e., Kurowski) to Chow and Reiffin.

Moreover, after alleging that Kurowski is “directed to the ‘582 Patent’s goal”

(Petition, 35), the Petition proffers nothing beyond the statement underlined below even purporting to explain why a POSITA would have been motivated to combine Kurowski with Chow and Reiffin:

Both Chow and Reiffin were ready for improvement. Ex.1022 ¶ 249. Chow focuses on multiprocessor systems consisting of multiple processors. *Id.* Reiffin focuses on multiprocessor systems consisting of multiple separate computers connetce (sic) via a local area network. *Id.* *Kurowski expands the concept to “thousands of clients running on a variety of computers across the Internet.”* *Id.* Petition, 35 (emphasis added).

The Petition’s conclusory statement that “Kurowski expands the concept” fails to establish the necessary motivation to combine the references. Even if it were true that “Kurowski expands the concept” (to be clear, that is not true), this fact does not explain *why* a POSITA would be motivated to combine Kurowski with Chow and Reiffin.

Petitioners’ allegation that Kurowski “expands” the concepts in Chow and Reiffin to more computers is also factually incorrect. Chow disclosed ““multiprocessing environments’... where multiple computers, digital data processing units, threads, or other processes perform concurrent tasks.” Ex. 1003, 1:27-30 (emphasis added). Chow did not place any limit on the number of computers deployed in its multiprocessing environment. Moreover, according to the Petition, Reiffin already “expands” Chow’s “concept to multiple networked computers” resulting in “increased computational power and the ability to solve larger

problems.” Petition, 10-11. Again, Reiffin did not place any limit on the number of computers deployed.

Given that Chow was already directed to the use of multiple computers to perform concurrent tasks and Reiffin already “expanded Chow’s “concept to multiple networked computers” resulting in “increased computational power and the ability to solve larger problems,” why would a POSITA have looked to Kurowski for this same concept? Petition, 11. The Petition doesn’t say.

Moreover, Petitioners’ allegations that “the architectures of Chow and Kurowski are very similar” (Petition, 36) is not only wrong,² but fails to explain *why* a POSITA would be motivated to combine Kurowski with Chow and Reiffin. The fact that references are similar is insufficient to establish a motivation to combine.

² The Petition (at 36) attempts to argue that Chow and Kurowski have similar architectures by comparing Figure 1 from Chow with Figure 3 from Kurowski. Aside from having rectangles and cylinders in approximately the place, 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 they are 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. 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. Ex. 2013, ¶¶ 78-79.

In sum, the Petition offers nothing more than unsupported conclusions regarding motivation to combine, which fail to add up to a reasonable likelihood of prevailing. *See Johns Manville Corp.*, Case IPR2018-00827, Paper 9, at 10-13 (designated informative) (rejecting a similar same-field, beneficial-combination rationale and denying institution); *In re Magnum Oil Tools Int’l, Ltd.*, 829 F.3d 1364, 1380 (a petitioner cannot satisfy its burden of proving obviousness by employing “mere conclusory statements,” but “must instead articulate specific reasoning, based on evidence of record, to support the legal conclusion of obviousness”).

a) The Fact That Chow And Kurowski Are Similar Is Insufficient To Establish A Motivation To Combine

Petitioners’ contention that Chow and Kurowski “are very similar” (Petition, 36) is not a sufficient reason to combine the references. *Securus Techs.*, 701 F. App’x 971, 976-77 (non-precedential) (holding that “[s]uch short-cut logic would lead to the conclusion that any and all combinations of elements known in [a] broad field would automatically be obvious, without the need for further analysis”); *Microsoft Corp.*, 662 F. App’x 981, 990 (non-precedential) (holding that “the Board correctly concluded” that the petitioner “did not articulate a sufficient motivation to combine” where the only reason given was “that the references were directed to the same art or same techniques”); *Johns Manville*, IPR2018-00827, Paper 9 at 10-11 (rejecting an argument that two references were similar and belonged to the same field as an

insufficient reason to combine those references). Petitioners’ assertion that “the architectures of Chow and Kurowski are very similar” is “simply too conclusory” to show that a POSITA would have combined the references in the manner claimed. *Securus Techs., Inc.*, 701 F. App’x at 976.

b) The Lack Of A Non-Conclusory Explanation Of “Why” Chow And Reiffin Would Have Been Combined Is Fatal

The remainder of Petitioners’ purported motivations for combining Chow, Reiffin, and Kurowski fare no better. After making the conclusory statement that “Kurowski expands the concept...,” the Petition includes the following additional conclusory remarks:

It would have been routine to add the features of Reiffin to the system of Chow. Ex.1022 ¶ 250. It would have also been routine to add the features of Kurowski to Chow and Reiffin. *Id.* Petition, 35.

Even if the Petition had established that it would have “been routine to add the features of Kurowski to Chow and Reiffin,” (to be clear, it did not) this fact would fail to explain *why* a POSITA would be motivated to combine Kurowski with Chow/Reiffin. Again, the lack of a “why” is dispositive: a determination of obviousness cannot be reached where the record lacks an “explanation as to how or why the references would be combined to produce the claimed invention.” *TriVascular*, 812 F.3d 1056, 1066; *Virtek Vision*, 97 F.4th at 886 (“These

disclosures, however, do not provide any reason why a skilled artisan would use 3D coordinates instead of angular directions in a system”).

The Petition’s conclusory assertion that “[a] POSITA would have thus been motivated to combine Chow, Reiffin, and Kurowski with a reasonable expectation of success” (Petition, 37) addresses the wrong issue and is also totally conclusory. “The obviousness inquiry does not merely ask whether a skilled artisan could combine the references, but instead asks whether ‘they would have been motivated to do so.’” *Adidas AG*, 963 F.3d 1355, 1359. A reasonable expectation of success (“could combine references”) does not establish a motivation to combine (“would have been motivated to do so”). An expectation of success cannot cure the deficiencies of Petitioners’ conclusory allegations that a POSITA would have been motivated to combine Chow and Reiffin.

c) The Testimony Of Petitioners’ Declarant Parrots The Words Of The Petition And Provides No Additional Explanation Or Citations Supporting The Alleged Motivation To Combine

The cited testimony from Petitioner’s declarant, Dr. Shamos, does not cure the Petition’s deficiencies. Ex.1022 ¶¶ 248-252. The cited testimony largely repeats (word-for-word) the petition’s conclusory assertions. *Compare* Pet. at 34-37 with Ex. 1022, ¶¶248-252. For that reason, the testimony of Dr. Shamos is entitled to little if any weight. *Xerox Corp. v. Bytemark, Inc.*, IPR2022-00624, Paper 9 at 15-17

(PTAB Aug. 24, 2022) (designated precedential) (according “little weight” to declaration testimony that contains a verbatim restatement of a petition’s conclusory assertions without additional supporting evidence or reasoning).

In summary, the Petition’s argument purporting to establish a motivation for combining Kurowski with Chow and Reiffin is deficient.

3. There Is No Nexus Between The Purported Motivation To Combine Kurowski With Chow and Reiffin And Each Of The Numerous Ways The Petition Alleges The References Could Be Combined

In Ground 2, the Petition purports to map each of Elements [1pre], [1a], [1b], [1c], [1d] and [1e] to Chow, Reiffin, and Kurowski, and seems to argue that if any element is not met by either Chow or Reiffin, then the element is met by Kurowski. This approach to Ground 2 encompasses at least 729 possible ways of combining the references. Just like the fact that flipping a three-headed coin six times can result in 3^6 or 729 possible outcomes, Ground 2 purports to advance at least the same number of different possibilities for combining Chow, Reiffin, and Kurowski:

3 possibilities (Chow, Reiffin, or Kurowski) for 1[pre] ×
3 possibilities (Chow, Reiffin, or Kurowski) for 1[a] ×
3 possibilities (Chow, Reiffin, or Kurowski) for 1[b] ×
3 possibilities (Chow, Reiffin, or Kurowski) for 1[c] ×
3 possibilities (Chow, Reiffin, or Kurowski) for 1[d] ×
3 possibilities (Chow, Reiffin, or Kurowski) for 1[e] =

 3^6 or 729 possible combinations

As was the case in Ground 1, the Petition fails explain how Petitioners' purported motivation for combining Kurowski with Chow/Reiffin fits with each of the at least 729 different possibilities for combining the references.

For example, consider an exemplary permutation where disclosures from Chow are purportedly mapped to elements [1 pre], [1a] and [1c], disclosures from Reiffin are purportedly mapped to element [1b] and disclosures from Kurowski are purportedly mapped to elements [1d]-[1e]. The Petition provides no explanation why a POSITA would have been motivated to combine these *specific* aspects of each of the references with one another. The Petition's conclusory assertions that "A POSITA would have recognized that benefit of combining Reiffin with Chow would be increased computational power and the ability to solve larger problems" and that "Kurowski expands the concept ..." are not tied to this specific combination. For example, there is no explanation why or how taking aspects of Chow purportedly mapped to [1 pre], [1a] and [1c] and combining them with disclosures from Reiffin purportedly mapped to elements [1b] and [1d]-[1e] would result in increased computational power or the ability to solve larger problems.

The failure of the Petition to articulate specific motivations to combine that are *particularized* for each of the at least 729 different possibilities for combining Kurowski with Chow and Reiffin advanced in Ground 2 renders the Petition's approach for combining Kurowski with Chow and Reiffin deficient.

C. Ground 2 Is Deficient

In summary, for at least the reasons set forth above, Ground 2 fails to demonstrate a reasonable likelihood of success with respect to independent claim 1 of the ‘582 Patent because the Petition fails to demonstrate that: (i) claim 1 is obvious over Kurowski alone, or (ii) claim 1 is obvious over Kurowski in combination with Chow/Reiffin. Since Ground 2 fails to demonstrate a likelihood of success with respect to independent claim 1, the Petition similarly fails to demonstrate a likelihood of success in Ground 2 with respect to claims 2-14, all of which depend directly or indirectly from independent claim 1.

X. THE DECLARATION OF PETITIONERS’ EXPERT IS ENTITLED TO LITTLE IF ANY WEIGHT

The Petition extensively relies on a 96-page expert declaration that largely parrots *verbatim* the wording of the Petition. For example, as set forth below, the following paragraphs of the expert declaration essentially parrot the Petition word-for-word:

<u>Argument In Petition</u>	Argument Repeated Almost Verbatim In Expert Declaration (Ex. 1022)
Ground 1 - Motivation to combine Chow and Reiffin (Pet., 11-12).	¶¶133-136.
Ground 1 - Discussion of [1pre] (Pet., 12).	¶¶140-142.
Ground 1 - Discussion of [1a] (Pet., 13-15).	¶¶144-146.

Ground 1 - Discussion of [1b] (Pet., 15-16).	¶¶ 149-151
Ground 1 - Discussion of [1c] (Pet., 16-17).	¶¶ 152-155
Ground 1 - Discussion of [1d] (Pet., 17-18).	¶¶ 156-159
Ground 1 - Discussion of [1e] (Pet., 19).	¶¶ 161-165
Ground 1 - Discussion of Claim 2 (Pet., 20-21).	¶¶ 168-171
Ground 1 - Discussion of Claim 3 (Pet., 21-22).	¶¶ 172-176
Ground 1 - Discussion of Claim 4 (Pet., 22-23).	¶¶ 177, 179-181
Ground 1 - Discussion of Claim 5 (Pet., 23-24).	¶¶ 188-190
Ground 1 - Discussion of Claim 6 (Pet., 24-25).	¶¶ 191, 193-196
Ground 1 - Discussion of Claim 7 (Pet., 25-26).	¶¶ 197, 199-201
Ground 1 - Discussion of Claim 8 (Pet., 26-27).	¶¶ 202, 204-208
Ground 1 - Discussion of Claim 9 (Pet., 25-26).	¶¶ 209, 211-214
Ground 1 - Discussion of Claim 10 (Pet., 28-29).	¶¶ 215, 217-228
Ground 1 - Discussion of Claim 11 (Pet., 29-30).	¶¶ 221-227
Ground 1 - Discussion of Claim 12 (Pet., 30-32).	¶¶ 228, 230-234
Ground 1 - Discussion of Claim 13 (Pet., 32-33).	¶¶ 235-240
Ground 1 - Discussion of Claim 14 (Pet., 33-34).	¶¶ 241-246
Ground 2 (Pet. 34-37).	¶¶ 247-252
Ground 2 - Discussion of [1pre] (Pet., 37-38).	¶¶ 253, 255-256
Ground 2 - Discussion of [1a] (Pet., 38-39).	¶¶ 257, 259-260
Ground 2 - Discussion of [1b] (Pet., 39).	¶¶ 262-264
Ground 2 - Discussion of [1c] (Pet., 40).	¶¶ 265, 267-268

Ground 2 - Discussion of [1d] (Pet., 41).	¶¶ 269, 271-272
Ground 2 - Discussion of [1e] (Pet., 42).	¶¶ 273, 275-276
Ground 2 - Claim 2 (Pet., 43-44).	¶¶ 277-278, 281-282
Ground 2 - Claims 3-14 (Pet., 44).	¶¶ 283-285

Since the expert declaration contains essentially a verbatim restatement of the allegations in the Petition, the declaration relied on by the Petition is entitled to little if any weight. *Xerox Corp. v. Bytemark, Inc.*, IPR2022-00624, Paper 9 at 15-17 (PTAB August 24, 2022) (designated precedential) (according “little weight” to declaration testimony that contains a verbatim restatement of a petition’s conclusory assertions without additional supporting evidence or reasoning); *Facebook v. Windy City Innovations, LLC*, 973 F.3d 1321, 1340-41 (Fed. Cir. 2020) (affirming the Board’s holding that a claim was not unpatentable when “the Board considered Facebook’s expert’s testimony but determined that it did not add materially to Petitioner’s unpersuasive attorney argument,” because it merely repeated Petitioner’s argument, nearly verbatim, without citation to the basis for his testimony”).

XI. CONCLUSION

For the reasons set forth above, the Petition does not come close to meeting the statutory threshold for institution under 35 U.S.C. § 314(a). The Petition’s two asserted grounds are each predicated on multiple legal and factual deficiencies that

cannot be cured by further argument or expert testimony. In Ground 1, the Petition fails to show that Chow alone, Reiffin alone, or the combination of Chow and Reiffin meets all limitations of claim 1 as arranged in the claim. The cited disclosures are not only incomplete, but also internally inconsistent with respect to critical elements such as [1a] and [1b]. The Petition does not explain how Chow's post-execution subdivision of an aggregate task could satisfy the claim's requirement for logically subdividing a preexisting input file into more partitions than subtask processors before any subtasks are executed. Nor does the Petition explain how Reiffin's next-in-queue subtask fetching could satisfy the requirement to distribute descriptions of all partitions to the subtask processors. These are central deficiencies that undermine the entirety of Ground 1.

In Ground 2, the Petition fares no better. The attempt to rely on Kurowski either alone or in combination with Chow and Reiffin suffers from the same fundamental gap: the failure to identify any disclosure of logically subdividing a preexisting input file into well-defined partitions, and the failure to identify any disclosure of distributing descriptions of all such partitions to the subtask processors. Petitioners' reliance on a web-based reporting interface in Kurowski is misplaced, as such functionality is not relevant to the machine-level distribution required by the claims.

The Petition also fails to establish any non-conclusory reason why a person of ordinary skill in the art would have been motivated to combine the cited references in the specific manners advanced. Petitioners rely on broad statements that the references are in the same field, pursue similar goals, or could be combined to achieve greater computational capacity. These statements are unsupported by evidence, fail to address the particular elements at issue, and do not explain why such combinations would have been pursued by a skilled artisan. Federal Circuit and Board precedent makes clear that mere conclusory assertions of a possible benefit or an alleged similarity between references are insufficient to satisfy the burden of proving a motivation to combine.

Petitioners' approach compounds these deficiencies by presenting a "mix and match" theory that implicitly advances dozens or even hundreds of different permutations of the references without articulating a tailored reason to combine for any specific permutation. This scattershot method does not satisfy the requirement for a clear, particularized explanation of how and why a skilled artisan would have made the proposed combination to arrive at the claimed invention.

The Petition is further undermined by Petitioners' contradictory positions on claim construction. In parallel district court litigation, Petitioners contend that key claim terms are indefinite and cannot be understood by a person of ordinary skill in the art. Here, Petitioners assert that no explicit construction is required because those

same terms have an ordinary and customary meaning. These positions cannot be reconciled, and they weigh against granting institution.

Finally, the Petition's heavy reliance on an expert declaration that merely repeats the Petition's assertions without additional reasoning or evidentiary support offers no substantive reinforcement of Petitioners' arguments. The Board has repeatedly held that such parroting testimony is entitled to little, if any, weight.

Taken together, these deficiencies demonstrate that Petitioners have not shown, and cannot show, a reasonable likelihood of prevailing with respect to any challenged claim. Because neither Ground 1 nor Ground 2 meets the statutory standard, and because both are undermined by the same fundamental factual and legal flaws, the Petition should be denied.

Respectfully submitted,

Dated: August 11, 2025

By: /Daniel H. Golub/
Daniel H. Golub
Registration No. 33,701

*Counsel for Patent Owner
Intellectual Ventures I LLC*

CERTIFICATE OF COMPLIANCE WITH WORD COUNT

Pursuant to 37 C.F.R. § 42.24(d), I certify that this **PATENT OWNER'S PRELIMINARY RESPONSE** complies with the type-volume limits set forth in the Memorandum and 37 C.F.R. § 42.24 because it contains 13,627 words, excluding the parts of the Brief that are exempted by 37 C.F.R. § 42.24(a), according to the word processing system used to prepare this Brief.

Respectfully submitted,

Dated: August 11, 2025

By: /Daniel H. Golub/
Daniel H. Golub
Registration No. 33,701

*Counsel for Patent Owner
Intellectual Ventures I LLC*

CERTIFICATE OF SERVICE

The undersigned certifies that pursuant to 37 C.F.R. § 42.6(e), a copy of the foregoing **PATENT OWNER'S PRELIMINARY RESPONSE** was served via email (as consented to by counsel) on August 11, 2025 to lead and backup counsel of record for Petitioner as follows:

John B. Campbell
jcampbell@McKoolSmith.com
MCKOOL SMITH, P.C.
303 Colorado Street, Suite 2100
Austin, TX 78701

Alan Block
ablock@mckoolsmith.com
McKool Smith, P.C.
300 South Grand Avenue, Suite 2900
Los Angeles, California 90071

Casey Shomaker
cshomaker@mckoolsmith.com
McKool Smith, P.C.
300 Crescent Court, Suite 1200
Dallas, TX 75201

Keith D. Harden
kharden@munckwilson.com
Munck Wilson Mandala, LLP
2000 McKinney Ave., Ste. 1900
Dallas, Texas 75201

S. Wallace Dunwoody
wdunwoody@munckwilson.com
Munck Wilson Mandala, LLP
2000 McKinney Ave., Ste. 1900
Dallas, Texas 75201

Michael C. Wilson
mwilson@munckwilson.com
Munck Wilson Mandala, LLP
2000 McKinney Ave., Ste. 1900
Dallas, Texas 75201

AA_Intellectual_Ventures@mckoolsmith.com

SWA-IV@munckwilson.com

Dated: August 11, 2025

By: /Daniel H. Golub/
Daniel H. Golub
Registration No. 33,701

*Counsel for Patent Owner
Intellectual Ventures I LLC*