

In the Matter Of:

Microsoft vs

Partec AG

DARELL LONG, PH.D.

September 17, 2025



UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE PATENT TRIAL AND APPEAL BOARD

IN RE MATTER OF:)
MICROSOFT CORPORATION,)
Petitioner,)
vs.) CASE NO. IPR2025-00318
PARTEC AG,)
Patent Owner.)

DEPOSITION OF DARELL D.E. LONG, Ph.D.

REMOTE VIA VIDEOCONFERENCE

Wednesday, September 17, 2025

Stenographically Reported by:

HEATHER J. BAUTISTA, CSR, CRR, RPR, CLR
Realtime Systems Administrator
California CSR License #11600
Oregon CSR License #21-0005
Washington License #21009491
Texas CSR License #10725

DEPOSITION of DARELL D.E. LONG, Ph.D., taken
before Heather J. Bautista, CSR No. 11600, a Certified
Shorthand Reporter for the state of California, with
principal office in the county of Santa Clara,
commencing on Wednesday, September 17, 2025, 8:02 a.m.,
Pacific, remotely via videoconference.

APPEARANCES OF COUNSEL:

For Petitioner:

Klarquist Sparkman LLP
BY: CAMERON D. CLAWSON, ESQ.
FRANK MORTON-PARK, ESQ.
One World Trade Center
121 S.W. Salmon Street
Suite 1600
Portland, Oregon 97204
Phone: (503) 595-5300 / Fax: (503) 595-5301
cameron.clawson@klarquist.com
frank.morton-park@klarquist.com

For Patent Owner:

Heim Payne & Chorush LLP
BY: CHRIS LIMBACHER, ESQ.
609 Main Street
Suite 3200
Houston, Texas 77002
Phone: (713) 221-2013
climbacher@hpcllp.com

1		3
2	INDEX OF EXAMINATION	
3		PAGE
4	DARELL D.E. LONG, Ph.D.	
5	EXAMINATION BY MR. LIMBACHER	7
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		

INDEX OF EXHIBITS

Exhibit No.	Description	Page
Exhibit A	U.S. Patent No. 11,537,442; 1001 in IPR	12
Exhibit B	Declaration of Darell Long; 1003 in IPR	13
Exhibit C	U.S. Patent Publication No. 2013/0282787; 1004 in IPR	13
Exhibit D	Article titled, "A Method for the Online Use of Offline Derived Remappings of Iterative Automatic Target Recognition Tasks onto a Particular Class of Heterogeneous Parallel Platforms"; 1005 in IPR	14
Exhibit E	U.S. Patent Publication No. 2018/0074855; 1006 in IPR	14
Exhibit F	Petition for Inter Partes Review of U.S. Patent No. 11,537,442	14
Exhibit G	Article titled, "A Semi-Static Approach to Mapping Dynamic Iterative Tasks Onto Heterogeneous Computing Systems"; 1007 in IPR	116
Exhibit H	Article titled, "Mapping Tasks onto Distributed Heterogeneous Computing Systems Using a Genetic Algorithm Approach"; 1008 in IPR	125
Exhibit I	Article titled, "Efficient Data-Parallel Distributed DNN Training For Big Dataset Under Heterogeneous GPU Cluster"; 2014 in IPR	162
Exhibit J	Paper entitled, "Disc failure prediction in heterogeneous environments"	169
Exhibit K	Article entitled, "RESAR, Reliable Storage and Exabyte Scale"	175

INDEX OF EXHIBITS - CONTINUED

Exhibit No.	Description	Page
Exhibit L	Paper titled, "Duplicate Data Elimination in a SAN File System"	179
Exhibit M	Paper titled, "Compare Results"; 2008 in IPR	183

1 Wednesday, September 17, 2025

2 8:02 a.m., Pacific

3 --000--

4 THE STENOGRAPHER: Good morning. We are on the
5 record. Today's date is Wednesday, September 17, 2025,
6 and the time is 8:02 a.m. Pacific.

7 This is the deposition of Darell Long, in the
8 matter of Microsoft Corporation v. Partec AG, Case No.
9 IPR2020-0318 [sic], venued in the United States Patent
10 and Trademark Office.

11 My name is Heather Bautista, and I am a
12 Certified Shorthand Reporter licensed by the State of
13 California. My license number is 11600.

14 This deposition and any transcript produced
15 therefrom will be handled pursuant to California Code of
16 Civil Procedure Section 2025.

17 As the deposition officer, I will be retaining
18 my duties and responsibilities under the Code.

19 At this time, I'll ask all counsel to state
20 your appearance for the record, starting with the
21 noticing attorney, please.

22 MR. LIMBACHER: This is Chris Limbacher from
23 Heim Payne and Chorush on behalf of Patent Owner Partec.

24 MR. CLAWSON: This is Cameron Clawson with
25 Klarquist Sparkman, on behalf of Petitioner Microsoft

1 Corporation. And also with me from Klarquist is Frank
2 Morton-Park.

3 DARELL D.E. LONG, Ph.D.,
4 having been first duly sworn, was examined and testified
5 as follows:

6 THE WITNESS: I do.

7 THE STENOGRAPHER: Thank you.

8 Please state your full name for the record.

9 THE WITNESS: Darell Don Earl Long. It's two
10 middle names.

11 THE STENOGRAPHER: Thank you.

12 Counsel, you can begin.

13 MR. LIMBACHER: Thank you.

14 Before I start, I just wanted to correct a
15 couple of things. This is for IPR2025-00318, and it
16 will be governed by the Rules of the United States
17 Patent and Trademark Office.

18 DIRECT EXAMINATION

19 BY MR. LIMBACHER:

20 Q. Good morning, Dr. Long. My name is Chris
21 Limbacher. I am counsel for Patent Owner Partec.

22 Will you please -- I know you already did this,
23 but will you please state your full name for the record.

24 A. Darryl Don Earl Long.

25 Q. Have you been deposed before?

1 A. I have.

2 Q. How many times?

3 A. Many. Honestly do not recall. More than ten.

4 Q. More than ten?

5 A. Yes, sir.

6 (Stenographer clarification.)

7 MR. LIMBACHER: My apologies. I asked him if
8 it had been more than 50.

9 THE WITNESS: And I said -- was no.

10 Q. (By Mr. Limbacher) About how many of those
11 were for inter partes review proceedings?

12 A. Thinking back, there was a lot of IPRs during
13 COVID, because that seemed to be the thing to do. Let's
14 say a dozen.

15 Q. Is there any reason, such as a medication or
16 illness, that would prevent you from giving your best
17 testimony today?

18 A. No. I will ask to get up about once an hour
19 since I've had back surgery, so sitting for long periods
20 of time is not pleasant.

21 Q. That's not a problem. We can take breaks
22 throughout the day. I just ask that you try to give me
23 about five minutes of notice before you need a break so
24 I can just wrap up whatever question I'm on, and then we
25 can call it there, and you can go take a break as

1 needed.

2 A. Okay.

3 Q. Do you understand that your testimony today is
4 under oath and that the oath you've taken has the same
5 significance as the oath you would take if you were to
6 testify in court?

7 A. I do.

8 Q. If you do not understand a question, will you
9 let me know so that I can clarify it?

10 A. I will.

11 Q. Thank you.

12 About how many hours have you spent on this
13 matter so far?

14 A. Total?

15 Q. Yes, sir.

16 A. About 60.

17 Q. How much time did you spend preparing to
18 testify today?

19 A. In the 20s; about 20 hours.

20 Q. How did you prepare to testify?

21 A. Say it again, please.

22 Q. How did you prepare to testify?

23 A. I think I -- I think I heard you say how; is
24 that correct?

25 Q. Yes, sir.

1 A. I reread documents that I'd read almost a year
2 ago, and I met with Cameron and with Frank, and we
3 talked -- we talked things through.

4 Q. Other than the attorneys with whom you prepared
5 for this deposition, have you spoken with anyone else
6 regarding your deposition today?

7 A. I told my wife I didn't want to get up this
8 early, but other than that, no.

9 Q. Have you reviewed any materials from any other
10 proceedings related to the '442 patent?

11 A. No, I don't believe I have.

12 Q. Did you rely on any materials from your
13 attorneys or anyone else in forming your opinions?

14 MR. CLAWSON: Objection. Form.

15 THE WITNESS: I guess I don't know what that
16 means.

17 Q. (By Mr. Limbacher) Did anyone -- did anyone
18 give you any materials that you based your opinion on;
19 as in you used those materials, and they formed your
20 opinion?

21 A. I see. So you're asking me whether or not I
22 gathered all the materials myself?

23 Q. Not necessarily. It could be analysis from
24 someone else that you reviewed and that you're basing
25 your opinion on that analysis.

1 A. Okay.

2 I think those are -- I think those are
3 separable issues. No, I'm not basing my analysis on
4 anybody else's analysis, so we can be clear about that.

5 Where did I get all of the materials?

6 I suggested a large number of the exhibits. I
7 think that the attorneys probably found a couple other
8 of the exhibits. I would have to -- if I still have my
9 notes in California, which I probably don't, would be,
10 Hey, we should look at this, and we should look at that;
11 you know, those kinds of things. I spent some hours on
12 Google Scholar, for example, back in whenever this
13 started.

14 Q. Did you do your own search for any prior art in
15 this proceeding?

16 A. I do. I do my own search, and the attorneys
17 also do their own searches, I suppose, and they say,
18 "What do you think about this?"

19 And I say, "Well, what do you think about
20 that?"

21 Q. Okay.

22 I'm now going to introduce several exhibits.
23 These are all documents that were previously filed in
24 the IPR, so if you already have these documents
25 downloaded locally, feel free to refer to your local

1 versions.

2 These exhibits --

3 A. Hold on. I'm -- I'm on a Windows computer,
4 which is not my computer. You had a Dropbox link that
5 you sent, I guess, yesterday. Is that what we're
6 referring to, or is it something else?

7 Q. Yes, sir. These will be documents that are in
8 the Dropbox.

9 A. Okay.

10 Q. So if it's easier for you, also feel free to
11 pull them up there; just whatever makes it easier for
12 you to do your review and answer the questions.

13 A. Thank you.

14 Just give me a second here.

15 Dropbox is being difficult here. It wants me
16 to log in, in order to download. Yeah. Okay.

17 This is -- the first one is the Lippert patent.

18 Q. So I'd like to -- let me go ahead and read off
19 all the ones that are being introduced right now, and
20 then we can work through making sure you have access.

21 A. Okay.

22 MR. LIMBACHER: I'm going to introduce U.S.
23 Patent No. 11,537,442 as Exhibit A to this deposition.
24 This is Exhibit 1001 in the IPR.

25 (Exhibit A was marked for identification.)

1 THE WITNESS: Okay.

2 Q. (By Mr. Limbacher) If I refer to this exhibit
3 as the "'442 patent," will you understand --

4 A. Got it.

5 Q. -- what I'm referring to?

6 A. Yes.

7 (Stenographer clarification.)

8 MR. LIMBACHER: I'm going to introduce the
9 declaration of Darell Long as Exhibit B to this
10 deposition. This is Exhibit 1003 in the IPR.

11 (Exhibit B was marked for identification.)

12 Q. (By Mr. Limbacher) If I refer to this exhibit
13 as "your declaration" or "your report," will you
14 understand what I am referring to?

15 A. I will.

16 MR. LIMBACHER: I'm going to introduce U.S.
17 Patent Publication No. 2013/0282787 as Exhibit C to this
18 deposition. This is Exhibit 1004 in the IPR.

19 (Exhibit C was marked for identification.)

20 Q. (By Mr. Limbacher) If I refer to this exhibit
21 as "Lippert," will you understand what I am referring
22 to?

23 A. I will.

24 MR. LIMBACHER: I'm going to introduce an
25 article titled, "A Method for the Online Use of Offline

1 Derived Remappings of Iterative Automatic Target
2 Recognition Tasks onto a Particular Class of
3 Heterogeneous Parallel Platforms," as Exhibit D to this
4 deposition. This is Exhibit 1005 in the IPR.

5 (Exhibit D was marked for identification.)

6 Q. (By Mr. Limbacher) If I refer to this exhibit
7 as "Budenske," will you understand what I am referring
8 to?

9 A. I will.

10 MR. LIMBACHER: I'm going to introduce U.S.
11 Patent Publication No. 2018/0074855 as Exhibit E to this
12 deposition. This is Exhibit 1006 in the IPR.

13 (Exhibit E was marked for identification.)

14 Q. (By Mr. Limbacher) If I refer to this exhibit
15 as "Kambatla," will you understand what I'm referring
16 to?

17 A. I will.

18 MR. LIMBACHER: I'm going to introduce the
19 petition for inter partes review of U.S. Patent No.
20 11,537,442 from IPR2025-00318, as Exhibit F to this
21 deposition. This is the petition in the IPR.

22 (Exhibit F was marked for identification.)

23 Q. (By Mr. Limbacher) If I refer to this exhibit
24 as "the petition," will you understand what I am
25 referring to?

1 A. I will.

2 Q. Thank you.

3 THE WITNESS: If the attorneys on my side are
4 listening, if they could send the IT guy over to log in
5 Dropbox, it would make things easier. If not, I can
6 work around this issue.

7 MR. CLAWSON: Chris, do you mind if we go off
8 record?

9 MR. LIMBACHER: Yes, let's go off the record
10 for now.

11 THE STENOGRAPHER: We're off the record. The
12 time is 8:14 a.m.

13 (Recess taken from 8:14 a.m. to 8:17 a.m.)

14 THE STENOGRAPHER: We're back on the record.
15 The time is 8:17 a.m.

16 Q. (By Mr. Limbacher) Okay.

17 Dr. Long, can you please turn to your
18 declaration. This should be Exhibit B to the deposition
19 and I believe Exhibit 1003 in the IPR.

20 A. Yes, sir.

21 Q. And then please turn to Paragraph 48 of your
22 declaration and read the first sentence of that
23 paragraph to yourself.

24 A. Scroll, scroll, scroll.

25 Okay.

1 Q. Did you consider the plain and ordinary meaning
2 of each claim term that you applied?

3 A. Yes, I did.

4 Q. Okay.

5 Please turn to the '442 patent. This is
6 Exhibit A to the deposition and Exhibit 1001 in the IPR.

7 A. Find the mouse. Okay.

8 Q. So please scroll down to Columns 5 and 6.

9 A. Okay.

10 Q. So please look at Column 5, Line 26.

11 What is the plain and ordinary meaning of a
12 heterogeneous computing system?

13 MR. CLAWSON: Objection. Form.

14 THE WITNESS: In what year are we? 2017?
15 2017, a heterogeneous computing system would be a
16 computing system with more than one type of processor.

17 Q. (By Mr. Limbacher) Okay.

18 Has that meaning changed over time?

19 A. Not in a substantive sense, other than every
20 computer, at this point, is a heterogeneous computer
21 system.

22 Q. So Claim 1 of the '442 patent recites
23 computation nodes and booster nodes; is that correct?

24 A. That's correct.

25 Q. Is reciting computation nodes and booster nodes

1 enough to indicate that something is a heterogeneous
2 computing system?

3 A. Well, in the context of the Lippert patent,
4 this is '442, there is a discussion of the Xeon Phi.

5 Q. Might be --

6 A. Okay.

7 So -- yeah, here we go.

8 If we look at Column 3, around Line 35, there
9 is discussion of the Xeon Phi. And if we look -- it's
10 formatted different here.

11 If we look around Line 56, it says, "The system
12 is flexible in all directions, also permitting joint
13 processing of sub-tasks by all nodes addressed herein,
14 as well as reshuffling of processing between the nodes."

15 It's interesting that the Xeon Phi, just like
16 the Xeon processor, was an x86 architecture. So in a
17 strict sense, the booster nodes and the compute nodes
18 actually have the same architecture as described in --
19 in the '442 patent.

20 Q. So do you disagree with the '442 patent's
21 characterization of those processors?

22 MR. CLAWSON: Objection. Form.

23 THE WITNESS: Which processors, Mr. Limbacher?

24 Q. (By Mr. Limbacher) So the paragraph you were
25 referring to recites an Intel XeonE5-2680 processor; is

1 that correct?

2 A. Correct. That's a fairly generic Intel Xeon
3 processor of the 2017 time frame.

4 Q. Would you consider that a computation node?

5 A. That would typically be called a computation
6 node.

7 Q. Would you consider it to be a booster node?

8 A. If it had the AVX or other instructions, people
9 would also call it a booster node, because it had
10 boosting instructions, but I don't think that's where
11 you're trying to go with this.

12 Q. So that paragraph also mentions the Intel Xeon
13 Phi processor; correct?

14 A. Correct.

15 MR. LIMBACHER: And just for the record, Phi is
16 spelled P-h-i.

17 Q. (By Mr. Limbacher) Do you --

18 A. It's after the Greek letter Phi, yes.

19 Q. Do you consider the Intel Xeon Phi processor to
20 be a computation node?

21 MR. CLAWSON: Objection. Form.

22 THE WITNESS: In the context of the patent,
23 they are talking about it as a booster node, but as a
24 person who had a Xeon Phi and can tell you that the Xeon
25 Phi is an x86 processor based on the Pentium III core

1 with a 512-bit vector accelerator, it is both a
2 computation node; and, in this context, they're calling
3 it a booster node.

4 Q. (By Mr. Limbacher) Do you believe that the
5 '442 patent is using the term "booster node" differently
6 than it is known in the art?

7 MR. CLAWSON: Objection. Form.

8 THE WITNESS: I think it is picking a fairly
9 unique processor and deciding to call it a booster node.
10 So the Xeon -- if we could talk about the history of the
11 Xeon Phi, if that's what you'd like to do. But it's an
12 interesting choice, is what I'm saying, to call that a
13 booster node.

14 Q. (By Mr. Limbacher) So if you were going to
15 classify the Intel Xeon Phi processor as either a
16 computation node or booster node, based on how those
17 terms are used in the art, separate from the
18 '442 patent, how would you classify it?

19 MR. CLAWSON: Objection. Form.

20 Objection. Scope.

21 THE WITNESS: I don't think you couldn't simply
22 classify it that way. And the reason is that the Xeon
23 Phi was capable, as I said, based on a Pentium III core,
24 and was capable of running Linux on each of the nodes,
25 so it was a full computational node. It had, in

1 addition, a 512-bit vector unit associated with each of
2 those cores.

3 Q. (By Mr. Limbacher) So moving back to Column 5,
4 Claim 1, if a system contains both computation nodes and
5 booster nodes, is that enough to indicate to you that it
6 is a heterogeneous computing system?

7 MR. CLAWSON: Objection. Form.

8 THE WITNESS: It would indicate to me that the
9 author of the claim meant to indicate that these nodes
10 were different. And them being different, I'm happy to
11 accept the word "heterogeneous."

12 Q. (By Mr. Limbacher) Okay.

13 With that, if you look at Column 5, Lines 26 to
14 27, does the recitation of heterogeneous computing
15 system alter the scope of the claim?

16 MR. CLAWSON: Objection. Form.

17 THE WITNESS: Recall I'm not an attorney, so
18 tell me what you mean by "alter the scope of the claim."

19 Q. (By Mr. Limbacher) If the words "a
20 heterogeneous computing system" were removed from this
21 claim, would it change what infringes this claim or what
22 invalidates this claim?

23 MR. CLAWSON: Objection. Form.

24 THE WITNESS: So I just removed the words
25 "heterogeneous computing system comprising" and would

1 say, "A method of operating a plurality of computation
2 nodes and a plurality of booster nodes."

3 Is that what you want me to do?

4 Q. (By Mr. Limbacher) Sure.

5 A. It certainly opens up a lot of questions. For
6 example, how are these nodes connected? If it's not a
7 computing system, how do they -- how do they interact?

8 It would be hard -- there would need to be more
9 detail in order to answer the question. I'd have to
10 say, "What is this thing if there's no system involved?"

11 Q. So look at Claim 9; this is Column 6, Line 26.
12 If this claim simply recited, "A system comprising,"
13 rather than "a heterogeneous system comprising," would
14 that change the scope of that claim?

15 MR. CLAWSON: Objection. Form.

16 Objection. Scope.

17 THE WITNESS: Well, if we removed
18 "heterogeneous," then we would have, "A system
19 comprising plurality of computation nodes and plurality
20 of booster nodes," and then they would ask, "Well, what
21 differentiates a booster node from a computation node?"
22 And that would be the question to be asked.

23 Q. (By Mr. Limbacher) Do you know what a claim
24 preamble is?

25 A. Not --

1 MR. CLAWSON: Objection. Form.

2 THE WITNESS: Not from a legal point of view,
3 but I know what a preamble is.

4 Q. (By Mr. Limbacher) So you were not advised on
5 the meaning of "preamble" before issuing your opinions;
6 is that correct?

7 MR. CLAWSON: Objection. Form.

8 THE WITNESS: No, that's not correct. I'm just
9 saying that I'm not an attorney, and I don't remember
10 the definition of preamble.

11 Q. (By Mr. Limbacher) Okay.

12 What is the plain and ordinary meaning of the
13 term "computation node" as used in Claim 1?

14 MR. CLAWSON: Objection. Form.

15 THE WITNESS: Computation node would be a node
16 or processor. It's interesting, because in computer
17 science, one typically talks about a node being more
18 than just the CPU.

19 So it would be a processor, some memory,
20 various support chips around it that interconnect that
21 is primarily designed or intended to do what I would
22 call general purpose computation.

23 Q. (By Mr. Limbacher) Is the term "computation
24 node" a term of art?

25 A. No.

1 Q. What plain and ordinary meaning did you apply
2 for the word "booster node" as used in Claim 1?

3 MR. CLAWSON: Objection. Form.

4 THE WITNESS: A booster node is something,
5 based on my reading, that is intended to perform some
6 kind of computation more rapidly or more efficiently
7 than a computation node would.

8 Q. (By Mr. Limbacher) Is the term "booster node"
9 a term of art?

10 MR. CLAWSON: Objection. Form.

11 THE WITNESS: Not so much a term of art, but a
12 computer scientist would understand it if it appeared in
13 context.

14 Q. (By Mr. Limbacher) You had mentioned, when you
15 first started answering this question, that you were
16 basing this, I believe, off of what you were reading; is
17 that correct?

18 A. Correct.

19 Q. What reading are you referring to there?

20 A. Well, we have, for example, the '442 patent
21 that uses boost- -- that uses "booster node," as an
22 example.

23 Q. Is there anything else, other than the
24 '442 patent, that you consulted for the meaning of
25 "booster node"?

1 A. I'd have to go and look and see what other
2 terms were used. Give me a few minutes.

3 Sorry. Windows is not my native habitat. The
4 mouse is moving.

5 Without searching everything, I don't see
6 anything other than '442 that uses the term "booster
7 node," although I didn't look at Lippert. Probably
8 does.

9 Q. Did you apply the same plain and ordinary
10 meaning for the term "computation node" for Claim 1 as
11 you did for Claim 9?

12 A. Yes.

13 Q. Did you apply the same plain and ordinary
14 meaning for the term "booster node" in Claim 1 as you
15 did in Claim 9?

16 A. Yes.

17 Q. What plain and ordinary meaning did you apply
18 for the term "sub-task" as recited in Claim 1?

19 MR. CLAWSON: Objection. Form.

20 THE WITNESS: Give me a moment to reopen here.
21 Sub-task is a portion of a task. There's a loose
22 taxonomy that computer science -- computer scientists
23 use. We describe things as jobs, and jobs are made up
24 of tasks, and tasks are made up of sub-tasks, and
25 sub-tasks are made up of sub-sub-tasks, and you can keep

1 going as far as you want, depending on the level of
2 granularity.

3 Q. Is "sub-task" a term of art?

4 A. "Sub-task" would be a term of art.

5 Q. Is it used the same in the '442 patent as it is
6 used in the art?

7 A. Other than the word "taskifying," I've never
8 seen taskifying used in the art, but yes, I would say
9 that it is.

10 Q. What makes up a sub-task?

11 A. That's somewhat subjective. A sub-task would
12 be something that could be done -- I won't say
13 independently of a task, but could be done, for
14 example -- are you familiar with the term "sub-routine"?

15 Q. Yes, sir.

16 A. So a sub-task is rather like a sub-routine,
17 except that it could be executed on a different
18 processor or concurrently. In a modern language like
19 Go, something like a "go" routine could be considered a
20 sub-task.

21 So it's something that -- not -- does not
22 necessarily have to be, but could be executed
23 concurrently. If we think of -- oh, I don't know --
24 threads, IP threads, or sub -- trying to think what
25 they're called in JAVA.

1 Most languages have some way of executing
2 sub-tasks concurrently.

3 Q. Does a sub-task include data?

4 MR. CLAWSON: Objection. Form.

5 THE WITNESS: If you're asking whether a
6 sub-task has data or manipulates data, then the answer
7 is yes; if you're saying is a sub-task just data, then
8 the answer is no.

9 Q. (By Mr. Limbacher) Is the plain and ordinary
10 meaning of the term sub-task you applied in Claim 1 the
11 same as you applied in Claim 9?

12 A. Yes.

13 Q. What plain and ordinary meaning did you apply
14 for the term "computing iteration" in Claim 1?

15 MR. CLAWSON: Objection. Form.

16 THE WITNESS: The plain and ordinary meaning
17 for iteration in this is a repetition of a task being
18 done. And I think that a "Monte Carlo" simulation was
19 an example that was given in the '442.

20 Q. (By Mr. Limbacher) Are computing iterations
21 processed serially?

22 MR. CLAWSON: Objection. Form.

23 THE WITNESS: Not necessarily.

24 Q. (By Mr. Limbacher) Can you give me an example
25 of one that would not be processed serially.

1 A. Sure. You can map a task across many nodes,
2 and each iteration -- for example, Monte-Carlo
3 simulation, each iteration could be processed in
4 parallel on a separate node. This is very commonly done
5 in high-performance computing. It's performed in
6 Google. Google does it with its map reduce. Hadoop
7 does it. It's very, very common.

8 (Stenographer clarification.)

9 THE WITNESS: Hadoop. H-a-d-o-o-p. Computer
10 scientists make up silly words.

11 Q. (By Mr. Limbacher) In Claim 1, are multiple
12 sub-tasks processed during a single computing iteration?

13 MR. CLAWSON: Objection. Form.

14 THE WITNESS: Are you asking whether that's
15 required that that be done?

16 Q. (By Mr. Limbacher) Sure.

17 Is that required by Claim 1?

18 MR. CLAWSON: Same objection.

19 THE WITNESS: So, again, this is my -- my
20 computer scientist mind versus legal mind. Plurality,
21 for me, means more than one.

22 So I think what you just asked me is whether
23 more than one sub-task happens during an iteration. It
24 seems to me that it requires more than one sub-task,
25 but --

1 Q. (By Mr. Limbacher) So how can you tell when
2 one computing iteration ends and another computing
3 iteration starts?

4 MR. CLAWSON: Objection. Form.

5 THE WITNESS: That depends on -- oh, there's a
6 typo -- oh. Okay.

7 (Stenographer clarification.)

8 THE WITNESS: That depends on the concurrency
9 model being used. It's common, for example, and I think
10 in some of the references, they talk about things such
11 as barrier synchronization.

12 So if you were to do a parallel Monte Carlo
13 simulation and you wanted multiple parallel instances,
14 you could have them all run in parallel, and you would
15 know you were done when they had all reached the barrier
16 synchronization, but you would know that one of them had
17 finished when it had reached the synchronization
18 barrier.

19 Q. (By Mr. Limbacher) In the context of Claim 1,
20 is it a requirement that every sub-task is completed in
21 every computing iteration?

22 MR. CLAWSON: Objection. Form.

23 THE WITNESS: Want to ask that again?

24 Q. (By Mr. Limbacher) Sure.

25 In the context of Claim 1, is it a requirement

1 that every sub-task is completed in every computing
2 iteration?

3 MR. CLAWSON: Objection. Form.

4 THE WITNESS: I don't see that requirement
5 there.

6 Q. (By Mr. Limbacher) Did you apply the same
7 plain and ordinary meaning for the term "computing
8 iteration" in Claim 1 as you did in Claim 9?

9 A. Yes.

10 Q. What is a distribution, as recited in Claim
11 1 -- or, I'm sorry. Scratch that.

12 What plain and ordinary meaning did you apply
13 for the word "distribution" in Claim 1?

14 MR. CLAWSON: Objection. Form.

15 THE WITNESS: When you read my report, you
16 realize that "distribution" is an overloaded term in
17 computer science, and I make it clear that I'm not
18 talking about a statistical distribution, but a -- an
19 allocation for assignment of, in this case, boost --
20 computation nodes and booster nodes. So that means an
21 allocation or an assignment of these nodes, as in
22 distributing cookies.

23 Q. (By Mr. Limbacher) So you say an allocation or
24 assignment of the nodes -- I'm sorry. Why don't you
25 repeat back to me your explanation so I can make sure I

1 understand it, please.

2 A. What it says is the first computing iteration
3 assigning and processing the plurality of sub-tasks by
4 at least the portion of the plurality of computation
5 nodes and at least a portion of the plurality of booster
6 nodes in a first distribution. That first distribution
7 is the assignment of those nodes.

8 Q. When you say "the assignment of those nodes,"
9 do you mean the assignment of sub-tasks to those nodes?

10 A. There's the -- there is -- there's two levels
11 of assignment. There is deciding which parts of the
12 application go to computation nodes, and there's which
13 parts go to booster nodes. So that's one level of
14 distribution or assignment.

15 Then of the plurality of computation nodes and
16 booster nodes, which of those computation nodes and
17 which of those booster nodes are going to get utilized,
18 and which of those booster nodes are -- of course, this
19 is architecturally dependent -- are assigned or
20 connected or affiliated with the computation nodes that
21 are performing those -- or coordinating those sub-tasks.

22 Q. So all of the elements you just mentioned in
23 your answer, those are required by Claim 1; is that
24 correct?

25 MR. CLAWSON: Objection. Form.

1 THE WITNESS: I think I was explaining what
2 I -- you asked me what I meant, and now you're asking me
3 if these are required.

4 It is required, yes, that these nodes are
5 assigned in this first distribution.

6 Q. (By Mr. Limbacher) What I'm asking is -- I
7 understand you to be talking about two levels of
8 assignment; is that correct?

9 A. Correct.

10 Q. On one level, you are assigning sub-tasks to
11 computation nodes and booster nodes; is that correct?

12 A. Correct.

13 Q. And on another level, you are choosing which
14 computation nodes and booster nodes to use; is that
15 correct?

16 A. Correct.

17 Q. Did you apply the same plain and ordinary
18 meaning for the word "distribution" in Claim 1 as you
19 did in Claim 9?

20 MR. CLAWSON: Objection. Form.

21 THE WITNESS: Yes.

22 Q. (By Mr. Limbacher) Do you need to take a break
23 yet, or would you like to keep going?

24 A. That would be great. I was going to ask. I'm
25 going to get some more coffee and walk around for about

1 two, three minutes. Okay?

2 MR. LIMBACHER: Okay.

3 Let's go off the record.

4 THE STENOGRAPHER: We're off the record. The
5 time is 8:50 a.m.

6 (Recess taken from 8:50 a.m. to 8:56 a.m.)

7 THE STENOGRAPHER: We're back on the record.
8 The time is 8:56 a.m.

9 Q. (By Mr. Limbacher) Dr. Long, did you speak
10 with anyone during the break?

11 A. Somebody in the breakroom, we talked about the
12 price of electricity in Oregon, and they complained and
13 I laughed, because it's 40 cents a kilowatt hour in
14 California.

15 Q. Did you speak with anyone about this proceeding
16 or this deposition?

17 A. Not at all.

18 Q. Okay.

19 I'm going to read a limitation from '442 patent
20 Claim 1. This is at Column 5, Lines 33 to 36.

21 "In a first computing iteration, assigning and
22 processing the plurality of sub-tasks by at least a
23 portion of the plurality of computation nodes and at
24 least a portion of the plurality of booster nodes in a
25 first distribution."

1 Did I read that correctly?

2 A. You did.

3 Q. Can that limitation be satisfied by a system
4 that includes only computation nodes?

5 MR. CLAWSON: Objection. Form.

6 THE WITNESS: So it depends on definition of
7 "plurality." Remind me, "plurality" means one or more
8 in legalese?

9 Q. (By Mr. Limbacher) That's actually a question
10 for you, sir.

11 What definition did you apply to the word
12 "plurality"?

13 A. I would say one or more, so that would require,
14 in my mind, at least one computation node and at least
15 one booster node.

16 Q. So to go back to my previous question, could
17 that limitation be satisfied by a system that included
18 only computation nodes?

19 MR. CLAWSON: Objection. Form.

20 THE WITNESS: That's tough, because it goes to
21 the nature of what is a booster node and can a booster
22 node do what a computation node can do.

23 If a booster node can do everything a
24 computation node can do, just faster, then it could.
25 But if we assume that a booster node is some kind of

1 special purpose node with some kind of special
2 capability, then you would need a booster node that is
3 different than a computation node.

4 Q. (By Mr. Limbacher) So if we assume that a
5 booster node must be different than a computation node,
6 could that limitation be satisfied by a system that
7 contains only computation nodes?

8 MR. CLAWSON: Objection. Form.

9 THE WITNESS: We have to be careful on the
10 definition of "different" and the definition of
11 "computation node."

12 The reason that I say that is, very often, you
13 have a heterogeneous computing system full of different
14 generations and different models of compatible but
15 different nodes, and some of them have different
16 capabilities than other ones. And those cases, I would
17 be very happy to call some of those booster nodes.

18 So I think that the -- the distinction between
19 computation nodes and booster nodes is a little squishy.

20 Q. (By Mr. Limbacher) So can you define that
21 distinction between computation nodes and booster nodes?

22 A. Well, if we go back to -- give me a moment
23 here -- Column 3, Line 56, this is, I think, the root of
24 the thing. "The system is flexible in all directions,
25 also permitting joint processing of sub-tasks by all

1 nodes addressed herein, as well as reshuffling of
2 processing between the nodes."

3 So what this is saying is that any node can run
4 any sub-task. And that turns out to be a fairly unique
5 property of the Xeon Phi processor.

6 Q. So the question to you, sir, is what is the
7 distinction between a booster node and a computation
8 node?

9 MR. CLAWSON: Objection. Form.

10 THE WITNESS: Well, to a computer scientist, a
11 booster node is something that would make it go faster.
12 And I offered earlier to talk about the architecture of
13 the Xeon Phi, which we can do if you'd like.

14 Q. (By Mr. Limbacher) I'm not looking to go into
15 the Xeon Phi specifically; what I'm trying to address
16 is -- is your report says that you applied the plain and
17 ordinary meaning of these terms, and I'm trying to
18 understand what plain and ordinary meaning you applied
19 for "computation node."

20 So let's go back to that -- what is the plain
21 and ordinary meaning that you applied for "computation
22 node"?

23 (Stenographer clarification.)

24 MR. CLAWSON: I said, "Objection to form."

25 THE WITNESS: We've talked about this before,

1 but I'm happy to do it again. A computation node, first
2 we take the word "node." A node is something that is
3 more than just a CPU. It will have -- it will have
4 memory, it will have some data, it will have a CPU, it
5 will have the various support chips that are around the
6 CPU, and it will be able to perform general purpose
7 computation.

8 Q. (By Mr. Limbacher) If I had an Intel -- well,
9 let me stop.

10 Do you know what an Intel x86 processor is?

11 A. Yes, I do.

12 Q. If I had an Intel x86 processor that did not
13 include a GPU, would you consider that to be a
14 computation node?

15 MR. CLAWSON: Objection. Form.

16 THE WITNESS: Which generation x86 processor?

17 Q. (By Mr. Limbacher) Let's call it a Pentium.

18 A. Oh, that's an old one.

19 Q. Yeah, well, I'm old, so that's what I'm going
20 with.

21 A. Yeah. I would have to look that one up,
22 whether or not that had any special purpose graphics
23 processing instructions in it; would need to know the
24 generation of the Pentium.

25 Q. Is there any type of processor that you would

1 consider to be a computation node?

2 MR. CLAWSON: Objection. Form.

3 THE WITNESS: Well, the Pentium would certainly
4 be a computation node, but that doesn't exclude it from
5 being something else either.

6 Q. (By Mr. Limbacher) Okay.

7 What would exclude it from being a booster
8 node?

9 MR. CLAWSON: Objection. Form.

10 THE WITNESS: If it were standing alone and it
11 were not being used to make some other computation go
12 faster.

13 Q. (By Mr. Limbacher) In your view, is there any
14 processor that would only be a computation node and not
15 a booster node?

16 MR. CLAWSON: Objection. Form.

17 THE WITNESS: It will be the same answer. If I
18 have a processor in isolation that can perform general
19 purpose computation; in other words, it is --

20 (Stenographer clarification.)

21 THE WITNESS: -- turing complete,
22 t-u-r-i-n-g -- then I would consider it a computation
23 node.

24 If, on the other hand, I have two processors
25 and one processor has a capability that the first

1 processor doesn't have, then that second processor could
2 be a booster node to the first processor because it has
3 a unique capability.

4 Q. (By Mr. Limbacher) Is it your view that there
5 are no existing processors that you're aware of that are
6 computation nodes and not booster nodes?

7 MR. CLAWSON: Objection. Form.

8 THE WITNESS: I think you're mischaracterizing
9 what I said.

10 Q. (By Mr. Limbacher) I'm asking you a question.
11 Do you believe there are any processors in existence
12 that are computation nodes but not booster nodes?

13 MR. CLAWSON: Objection. Form.

14 THE WITNESS: What I've said is that if I have
15 two processors, and one processor is faster than the
16 other, the second processor could be used as a booster
17 node for the first slower processor.

18 So if we reduce the slowest possible processor,
19 I suppose, then we could say this is a booster for
20 nothing, but that's kind of a ridiculous argument.

21 In general, if I have all the same kind of
22 processor in my computer, then I would not call any of
23 those booster nodes.

24 If I have a processor that has a faster
25 floating point unit, for example, then I could use that

1 as a booster. This is -- you were talking about the
2 Pentium. The Pentium, this was the -- actually the
3 x386. Before the 486 came out, they had a separate
4 floating point unit. There was no floating point unit
5 in 386. So a 486, the floating point unit would be a
6 booster node if used in combination with a 386.

7 Q. (By Mr. Limbacher) So what I take from all
8 that is that the only way you can have a processor that
9 is a computation node but not a booster node is if it is
10 the slowest possible processor for all tasks; is that a
11 correct characterization?

12 MR. CLAWSON: Objection. Form.

13 THE WITNESS: No. It would also be if it were
14 the only processor being used or if -- of a set of
15 processors being used, that set of processors was
16 homogeneous.

17 Q. (By Mr. Limbacher) Okay.

18 If you had a system that had four Intel Pentium
19 processors, would those processors be computation nodes?

20 A. Are they all the same?

21 Q. Let's say they are all identical and there are
22 no other processors in that system.

23 MR. CLAWSON: Objection. Form.

24 THE WITNESS: 2017?

25 Q. (By Mr. Limbacher) Sure.

1 A. Mr. Limbacher?

2 Q. Sure.

3 A. Even in 2017, those wouldn't be the only
4 processors in the system. But in terms of running
5 applications, aside from the GPU that would necessarily
6 be present in 2017, so if -- let's -- you want to
7 exclude the GPU?

8 Q. Sure. Let's exclude the GPU. I said there's
9 only -- so let's stop for a second.

10 Can you go to your -- your declaration. This
11 is Exhibit B in the deposition, Exhibit 1003 in the IPR.

12 A. Okay.

13 Q. And go to Paragraph 68.

14 A. Sorry. It takes a while.

15 Q. Actually, I'm sorry. Go to Paragraph 119
16 instead.

17 A. Okay.

18 Q. And at the top of paragraph -- or top of Page
19 73, please read that paragraph at the top of Page 73,
20 along with whatever else you need for context.

21 A. Okay.

22 Q. Do you see where you state that, "CPU and
23 memory resources, i.e., computation nodes and GPUs,
24 graphical processing units, i.e., boosters"?

25 A. I do.

1 Q. Do you disagree with that now?

2 A. I do not.

3 Q. Do you agree that CPUs are computation nodes?

4 MR. CLAWSON: Objection. Form.

5 THE WITNESS: Well, a GPU also has a CPU in it,
6 so a CPU forms part of a computation node.

7 Q. (By Mr. Limbacher) Can you say that again. I
8 think I got lost.

9 A. Okay.

10 A CPU forms part of a computation node. A CPU,
11 in and of itself, is not a computation node. Even here.
12 CPU and memory resources.

13 A GPU is a booster, but GPUs also have CPUs in
14 them.

15 Q. Can you give me an example of any processor on
16 the market that would be just a booster node or just a
17 booster and not a computation node? I'm sorry. Scratch
18 that. Let me do a cleaner version of that.

19 Can you give me an example of any processor on
20 the market that would be a booster node and not a
21 computation node?

22 MR. CLAWSON: Objection. Form.

23 THE WITNESS: In 2025 or 2017?

24 Q. (By Mr. Limbacher) Any year ever.

25 A. Any year ever. In the 1980s, a floating point

1 system, floating point unit, for the Solarity computer
2 system; the -- for the 8386, the floating point unit,
3 which numbers escapes me, that you would plug into a
4 separate socket would be a booster for the 386. There
5 was a similar one for the 486 until IBM integrated one
6 into its 486 processor.

7 Various digital signal processing processors
8 dating up through the early 2000s did not have CPUs.
9 FPGAs of various types. ASICs of various types.

10 Q. Okay.

11 Can you give me an example of a processor
12 that's ever existed that would just be a computation
13 node and not a booster node?

14 MR. CLAWSON: Objection. Form.

15 THE WITNESS: I've done this multiple times.
16 I'll do it again. If I have a CPU and even with
17 floating point in it, if it is the only one or it is a
18 collection, a homogeneous collection of these nodes,
19 then this would be a computation node, but not a booster
20 node because it's not boosting anything.

21 Q. (By Mr. Limbacher) So is it your view that
22 there's no processor you could point to standing alone
23 and say that is a computation node and it's entirely
24 dependent on what other processors it's been bundled
25 with?

1 MR. CLAWSON: Objection. Form.

2 THE WITNESS: I'm sorry, Mr. Limbacher. I
3 don't understand what you just said.

4 Q. (By Mr. Limbacher) So I gave you the example
5 previously of a Pentium processor, and your response
6 was, "Well, it depends on what other processors are in
7 the system with it."

8 A. Yes.

9 Q. So your view of computation node is that it is
10 entirely dependent on what other processors are in the
11 system with a given processor?

12 MR. CLAWSON: Objection. Form.

13 THE WITNESS: I said it before; I'll say it
14 again: If I have a CPU with memory and it's alone, it
15 has a computational capability. If I have five of these
16 and they're all the same, they have the same
17 computational capability, none of them boosts the
18 performance. If I have four Pentium IIIs and I have one
19 Pentium V, then I would consider the Pentium V a booster
20 node because it's much faster.

21 Q. (By Mr. Limbacher) Please turn back to the
22 '442 patent. I'm again addressing the limitation I read
23 previously that's on Column 5, Lines 33 to 36.

24 You mentioned there are various DSPs up through
25 the early 2000s that you would consider to be booster

1 nodes; is that correct?

2 A. Correct.

3 Q. If a system included only those DSPs, could it
4 meet that limitation?

5 A. Only those --

6 MR. CLAWSON: Objection. Form.

7 Q. (By Mr. Limbacher) Only those DSPs?

8 A. They would not be computation nodes.

9 Q. So if the system included only DSPs, they could
10 not meet this limitation; is that correct?

11 MR. CLAWSON: Objection. Form.

12 THE WITNESS: You would have to look at the
13 specific DSPs, but let's pause at the dumbest possible
14 TIDSP from the 1990s, then it would likely not meet
15 requirements of Claim 1.

16 Q. (By Mr. Limbacher) So why would it not meet
17 the requirements of Claim 1?

18 A. Without looking at the model of the DSP, I'm
19 assuming that the DSP is not turing complete; in other
20 words, it can't compute all --

21 (Stenographer clarification.)

22 THE WITNESS: -- partial recursive functions.
23 It's a question of universality.

24 Q. (By Mr. Limbacher) I'm sorry. Are you
25 still --

1 A. No, I'm -- I'm done. I'm saying if you don't
2 have a processor that's universal, in other words, it's
3 not turing complete or can't compute partial recursive
4 functions, then you would not consider it a
5 computational node.

6 Something that just does a 48 transform, for
7 example, would be it -- a -- it does a computation, but
8 it can't do general purpose computation.

9 Q. Referring again to the same limitation --
10 again, this is Column 5, Lines 33 to 36 -- if you
11 assigned sub-tasks only to booster nodes, would this
12 limitation be met?

13 MR. CLAWSON: Objection. Form.

14 THE WITNESS: Who's doing the assigning?

15 Q. (By Mr. Limbacher) You tell me. Under your
16 interpretation of this claim, who's doing the assigning?

17 MR. CLAWSON: Objection. Form.

18 THE WITNESS: This nice little diagram here.

19 We've got application managers and resource
20 managers. They could do the assigning.

21 Q. (By Mr. Limbacher) Okay.

22 So, again, referring to the limitation at
23 Column 5, Lines 33 to 36.

24 A. So I need to ignore Claim 2 and Claim 3, then;
25 is that correct?

1 Q. Why do you need to ignore Claim 2 and Claim 3?

2 A. I'm just asking you. You're referring me to
3 certain lines in Claim 1, and I haven't gotten to Claim
4 2 or Claim 3 yet, which mention the application manager
5 and the resource manager.

6 Q. Well, sir, you can refer to them if it helps
7 you answer the question, but what I'm asking is --
8 specifically is Column 5, Lines 33 to 36, if sub-tasks
9 are only assigned to booster nodes, is that limitation
10 satisfied?

11 MR. CLAWSON: Objection. Form.

12 THE WITNESS: I can certainly think of
13 computations that I only want done by booster nodes.
14 Probably want one computation node to coordinate the
15 task.

16 Q. (By Mr. Limbacher) Well, sir, that's not what
17 I asked.

18 Under this hypothetical, if sub-tasks are
19 assigned only to booster nodes, would this limitation be
20 met? And by this "limitation," I'm referring to the
21 limitation at Column 5, Lines 33 to 36.

22 A. Right. And I think we're stuck on the word
23 "plurality." I think the word plurality requires me to
24 say I need at least one computation node.

25 Q. So you're saying the limitation would not be

1 met in that situation?

2 MR. CLAWSON: Objection. Form.

3 THE WITNESS: I'm saying I think that the
4 wording requires me to use at least one computation
5 node, even if I didn't need it.

6 Q. (By Mr. Limbacher) Okay.

7 You see in this limitation where it refers to a
8 portion of the plurality of booster nodes?

9 A. Yes.

10 Q. What do you interpret "a portion" as being?

11 MR. CLAWSON: Objection. Form.

12 THE WITNESS: A subset.

13 Q. (By Mr. Limbacher) Can the subset be empty?

14 A. A portion of the plurality. I think a POSITA
15 reading this would think that you need to have a booster
16 node being used. They probably wouldn't be thinking
17 about strict set theory.

18 Q. Is it your view that "a portion" means one or
19 more in this situation?

20 MR. CLAWSON: Objection. Form.

21 THE WITNESS: I think taken together, "at least
22 a portion," the "at least" means you got to have some.
23 So for me, it means one or more.

24 Q. (By Mr. Limbacher) Is the -- is the term
25 "portion," as used in Claim 9 -- sorry. Scratch that.

1 Would the term "portion," as used in Claim 9,
2 also require one or more?

3 MR. CLAWSON: Objection. Form.

4 THE WITNESS: The phrase "at least a portion"
5 would, in my mind, require one or more.

6 Q. (By Mr. Limbacher) Okay.

7 Now I'm going to read another limitation from
8 Claim 1. This is Column 5, Lines 37 to 40. "Generating
9 using information relating to the processing of the
10 plurality of sub-tasks by at least the portion of the
11 plurality of computation nodes and at least a portion of
12 the plurality of booster nodes."

13 Did I read that correctly?

14 A. You did.

15 Q. What plain and ordinary meaning did you apply
16 for "information relating to the processing of the
17 plurality of sub-tasks"?

18 MR. CLAWSON: Objection. Form.

19 THE WITNESS: It's informed by some of the
20 words in the body of the patent. And also in Lippert,
21 which I believe is incorporated -- we're talking about
22 performance information.

23 So the simple plain and ordinary meaning is how
24 well did it perform? The details aren't given here, but
25 you can -- if you want to ask what a POSITA would look

1 for, I can tell you.

2 Q. (By Mr. Limbacher) Do you see for Claim 9 on
3 Column 6, Line 37, where it recites information?

4 A. 37. 37. 35. 36, 37. The line that begins,
5 "An application manager"?

6 Q. Yes, sir.

7 A. Yeah, I see it.

8 Q. Did you apply the same plain and ordinary
9 meaning for information for this -- for this Claim 9 as
10 you did for Claim 1?

11 A. Yes.

12 Q. Do you see on that same line where it recites
13 an application manager?

14 A. Yes.

15 Q. What plain and ordinary meaning did you apply
16 for "application manager"?

17 MR. CLAWSON: Objection. Form.

18 THE WITNESS: Sorry. We're having search
19 issues here.

20 The application manager is the component that
21 decides which sub-tasks go to computation nodes and
22 which go to booster nodes.

23 Q. (By Mr. Limbacher) Is it a term of art?

24 A. It's a term that would be understood as
25 something that manages applications.

1 Q. So please look at Column 6, Line -- this is 33,
2 where it says, "A resource manager."

3 Do you see that?

4 A. Yes.

5 Q. What plain and ordinary meaning did you apply
6 for the term "resource manager"?

7 MR. CLAWSON: Objection. Form.

8 THE WITNESS: The resource manager is the
9 entity that assigns the actual physical hardware, the
10 computation nodes and the booster nodes.

11 Q. (By Mr. Limbacher) Is it a term of art?

12 A. It is. Just as "application manager" is
13 something that would be understood in context, "the
14 resource manager" would be understood in context.

15 Q. Do you see where Claims 3 and 4 also recite a
16 resource manager?

17 A. Yes.

18 Q. Did you apply the same plain and ordinary
19 meaning for resource manager to Claim 3 and Claim 4 as
20 you did for Claim 9?

21 A. Yes.

22 Q. Do you see where Claim 3 recites an application
23 manager? It should be --

24 A. Yes, I see it.

25 Q. Did you apply the same plain and ordinary

1 meaning for application manager in Claim 3 as you did in
2 Claim 9?

3 A. Yes.

4 Q. Do you see for Claim 3, this is Column 6, Line,
5 looks like, 4 to 5, where it says, "The resource manager
6 dynamically" --

7 (Stenographer clarification.)

8 Q. (By Mr. Limbacher) -- where "The resource
9 manager dynamically alters further distribution during
10 the computing of the computation task"?

11 A. What line are you on? I've read that before,
12 but I'm not seeing which line you're on.

13 Q. So I'm on Column 6, Lines, looks like, 3 to
14 5-ish?

15 A. Okay. Okay.

16 Yes, go ahead.

17 Q. What is the further distribution there
18 referring to?

19 MR. CLAWSON: Objection. Form.

20 THE WITNESS: It means that the resource
21 manager, based on the information that was received, the
22 application manager indicates that some reallocation of
23 resources would hopefully be beneficial and the resource
24 manager does that reallocation of resources.

25 Q. (By Mr. Limbacher) What does it mean for

1 altering a distribution to be dynamic, as recited in
2 Claim 3?

3 MR. CLAWSON: Objection. Form.

4 THE WITNESS: "Dynamically," in computer
5 science, typically means its's done in response to
6 something, as opposed to statically, which means it's
7 done once.

8 Q. (By Mr. Limbacher) I'm going to refer you back
9 to Claim 9. This is Column 6, Line 39. There is a word
10 that is spelled D-a-e-m-o-n-s and I'm going to ask you
11 how you typically pronounce that word.

12 A. Daemon.

13 Q. Okay.

14 That's the pronunciation I'm going to use then.

15 A. Not "dee-mon," but "day-mon." But yeah, it's
16 pronounced all over the place.

17 Q. Okay.

18 Just to make sure I got it.

19 And you pronounce it Daemon; correct?

20 A. If you say Daemon or Daemon, either one is okay
21 with me. Computer scientists use both. And probably
22 have since Maxwell.

23 Q. Okay.

24 What plain and ordinary meaning did you apply
25 for the term "daemons" as recited in Claim 9?

1 A. Daemon is a helper process in general and
2 that's what it would be understood to be. To any
3 computer scientist in this context, it's something that
4 gathers information and, if I recall correctly, also
5 does some actions, but maybe that's Lippert that I'm
6 thinking of. It's a helper process.

7 Q. Is it a term of art?

8 A. It is a term of art.

9 Q. What differences are there between -- scratch
10 that.

11 Do you see where Claim 9 recites, "Daemons
12 operating in at least the portion of the plurality of
13 the computation nodes and at least a portion of the
14 plurality of the booster nodes"?

15 A. I do.

16 Q. Would the daemons in the computation nodes be
17 different than the daemons in the booster nodes?

18 MR. CLAWSON: Objection. Form.

19 THE WITNESS: The answer is maybe. In the case
20 of the Xeon Phi, probably not. In the case of a truly
21 heterogeneous system, almost certainly yes.

22 Q. (By Mr. Limbacher) What does it mean for a
23 daemon to operate in a computation node? And I'm
24 emphasizing the word "in."

25 A. In.

1 MR. CLAWSON: Objection. Form.

2 THE WITNESS: Then we need to discuss operating
3 systems theory for a moment, then.

4 Q. (By Mr. Limbacher) Okay.

5 A. Okay.

6 A processor, CPU, memory, various circuits
7 around it, has an abstraction and that abstraction is
8 called "process." A process provides the illusion that
9 it is in possession of that CPU and that memory and it
10 provides isolation from other processes that run on that
11 CPU.

12 We clear so far?

13 Q. Yes.

14 A. Good.

15 So we say that that process operates in or on
16 that CPU. So when we say that a daemon is in or on the
17 CPU or a node, all we're saying is it's a process that's
18 executing on that node.

19 A daemon is just a certain kind of process.

20 Q. Do you see in Claim 8 where it recites, "A
21 daemon operating at a node"?

22 A. I do.

23 Q. What is the difference between a daemon
24 operating at a node and a daemon operating in a node?

25 MR. CLAWSON: Objection. Form.

1 THE WITNESS: I think a computer scientist
2 would read them as the same thing.

3 Q. (By Mr. Limbacher) Please look at Claim 6.
4 This is in Column 6, from Lines 15-ish to 17-ish.

5 A. Okay.

6 Q. Do you see it refers to a rating provided in
7 source code?

8 A. Right.

9 Q. What plain and ordinary meaning did you apply
10 for "rating provided in source code"?

11 MR. CLAWSON: Objection. Form.

12 THE WITNESS: There was a discussion in
13 Lippert, as I recall, discussing the use of pragmas, and
14 if you want, we can discuss what a pragma is.

15 Pragma is a compiler directive. There's a lot
16 of baggage in that word, the wording here, which we can
17 unpack.

18 "Rating," the plain and ordinary meaning is
19 that there's some indication calling it a rating in the
20 source code for each sub-task. If you read Lippert,
21 which is included by reference, it discusses pragmas.
22 Pragmas are something that is not unique to, but is
23 characteristic of a certain version of the C-programming
24 language and is indicated to use things outside of the
25 language.

1 And in this case, it would say something. We
2 didn't get examples, but it would say something like,
3 This sub-task would be really good to run on a booster
4 node. This is an annotation.

5 So the plain and ordinary meaning is that there
6 was some form of annotation being done in the source
7 code for each sub-task.

8 Q. Is "rating provided in source code" a term of
9 art?

10 A. That particular phrasing is not a term of art.

11 Q. Can you turn to Lippert, which is Deposition
12 Exhibit C, and it's Exhibit 1004 in the IPR.

13 A. Just a second. Okay. There's Lippert.

14 Q. Do you know where in Lippert it discusses
15 pragmas?

16 A. This is -- let me look. Pragma. Is this not
17 Lippert? Where did I see pragma?

18 Pragma is in Exhibit A in the '442 patent, Line
19 49, Column 1. Wrong Lippert. Sorry.

20 Q. No problem. Thank you.

21 Can you turn back to the '442 patent. This is
22 Deposition Exhibit A. This is Exhibit 1001 in the IPR.

23 A. Um-hum.

24 Q. If you're looking at Column 6, this is Claim 7.
25 What plain and ordinary meaning did you apply

1 for the term "grouping of sub-tasks"?

2 MR. CLAWSON: Objection. Form.

3 THE WITNESS: Okay. Let me try to define this
4 as you're going to group things together based on some
5 characteristics wherein the information is used, the
6 information is not really specified, it's vague, but
7 you're -- you would group sub-tasks together that are
8 related in some way.

9 An example would be if I have three tasks that
10 require -- are required to do a --

11 (Stenographer clarification.)

12 THE WITNESS: -- Fourier transform, F-o-u-r --
13 -o-u-i-e-r.

14 Q. (By Mr. Limbacher) Is "grouping of sub-tasks"
15 a term of art?

16 A. Something that would be understood to a
17 computer scientist. If I say I'm grouping things, I
18 would understand what that means. I would ask you, why
19 am I grouping them; what is the grouping metric?

20 Q. Do you see the term "measure of a loading" in
21 '442 patent, Claim 8?

22 A. Yes.

23 Q. What plain and ordinary meaning did you apply
24 for that term?

25 MR. CLAWSON: Objection. Form.

1 THE WITNESS: I would -- first of all, I would
2 fuss about the grammar of that, but normally one would
3 say a measure of the load of a node. But load on a node
4 is typically measured according to several -- several
5 metrics. One is CPU utilization; one is memory
6 utilization, for example.

7 So when it says "a measure," the measure is not
8 specified, but it would be one of the typical measures
9 of load on a node.

10 Q. (By Mr. Limbacher) Okay.

11 Is -- is "measure of a loading" a term of art?

12 A. "Measure of a loading" is not a term of art;
13 "Measure of the load" is a term of art.

14 Q. You need a break now or would you like to keep
15 going?

16 A. Break would be great. We're doing about once
17 an hour. I'll be fine. I'll be back in five minutes.

18 MR. LIMBACHER: Off the record, please.

19 THE STENOGRAPHER: We're off the record. The
20 time is 9:49 a.m.

21 (Recess taken from 9:49 a.m. to 9:56 a.m.)

22 THE STENOGRAPHER: We're back on the record.
23 The time is 9:56 a.m.

24 Q. (By Mr. Limbacher) Dr. Long, did you speak
25 with anyone during the break about this proceeding or

1 this deposition?

2 A. I did not.

3 Q. Can you turn to the -- it's the '442 patent.
4 This is Deposition Exhibit A. It's Exhibit 1001 in the
5 IPR. I'm on Column 6, looking at Claim 9. This is
6 Column 6, Lines 33 to 37.

7 Do you see where it -- it's reciting, "A
8 resource manager for assigning at least a portion of the
9 plurality of booster nodes and at least a portion of the
10 plurality of computation nodes to each other"?

11 A. I do.

12 Q. What does it mean to assign computation nodes
13 and booster nodes to each other?

14 MR. CLAWSON: Objection. Form.

15 THE WITNESS: Well, we have to read some things
16 into this. I would say that it means to logically
17 assign, rather than physically assign, because that
18 would imply things about the interconnection
19 architecture, which is unspecified.

20 So if I'm performing some computation and one
21 of the sub-tasks is to be executed on a booster node,
22 the sub-task, in a sense, belongs to or is controlled by
23 the larger part of the computation in the computation
24 node.

25 So what we say in computer science is I "farmed

1 that out," so I'm farming out some computation to a
2 booster node. So that booster node, for those purposes,
3 is assigned to me for the task that I'm farming out.
4 That's my understanding of this.

5 Q. (By Mr. Limbacher) Okay.

6 Would you please turn to your declaration.
7 This is Deposition Exhibit B. It's Exhibit 1003 in the
8 IPR.

9 A. Okay.

10 Q. And then go to Paragraph 4.

11 So Paragraph 4 -- sorry. Let me know when
12 you're there.

13 A. It's driving with the mouse and it's -- every
14 time you say something, the mouse jumps over to the
15 other screen.

16 Okay. I'm on Paragraph 4.

17 Q. You state that you are an expert in the field
18 of distributed computing systems; correct?

19 A. That's correct.

20 Q. What are distributed computing systems?

21 A. Distributed computing systems -- well, this
22 point in life, most computers are distributed, but
23 distributed computing systems are ones that have more
24 than one processor, more than one process, and are
25 connected by some form of communication network.

1 (Stenographer clarification.)

2 THE WITNESS: Distributed.

3 Q. (By Mr. Limbacher) Is a distributed computing
4 system the same thing as a heterogeneous computing
5 system?

6 A. You're asking whether they're isomorphic, the
7 answer is no, but almost every distributed computing
8 system is a heterogeneous computing system.

9 Q. So looking back at the '442 patent, this is
10 Deposition Exhibit A, Exhibit 1001 in the IPR, I'm
11 looking at Claims 1 and Claim 9 on Columns 5 and 6.

12 A. Um-hum.

13 Q. Would you consider the systems recited by these
14 claims to be distributed computing system?

15 A. Sure. Looks like it to me.

16 Q. What, in the claims, tells you that they are
17 distributed computing system?

18 A. I would just -- even just looking at the
19 diagrams. I have different types of nodes. Each of
20 these nodes is going to have daemons and operating
21 systems. These are all the characteristics of a
22 distributed computing systems.

23 THE STENOGRAPHER: We're off the record. The
24 time is 10:02 a.m.

25 (Brief recess taken to address stenographer's

1 equipment issue from 10:02 a.m. to 10:04 a.m.)

2 THE STENOGRAPHER: We're back on the record.

3 The time is 10:04 a.m.

4 Q. (By Mr. Limbacher) All right.

5 If you can look at Claim 1, this is in Column 5
6 of the '442 patent, what elements in Claim 1 tell you
7 that it's referring to a distributed computing system?

8 A. A plurality of computation nodes.

9 Q. Okay.

10 Is that -- is that, by itself, sufficient to
11 make it a distributed computing system?

12 A. Almost certainly.

13 Q. Can you go back to your declaration. This is
14 Deposition Exhibit B, Exhibit 1003 in the IPR.

15 A. Paragraph?

16 Q. Paragraph 19.

17 And then read that to yourself, please.

18 A. This is refusing to scroll. Scrolling. Excuse
19 me. Computer is going insane. I got to 19, then it
20 magically started scrolling backwards slowly.

21 This is, "Not aware of any allegations," that
22 paragraph, 19?

23 Q. Yes, sir.

24 A. Okay.

25 Q. What did you review to reach that conclusion?

1 MR. CLAWSON: Objection. Form.

2 THE WITNESS: Let's see. I don't recall off
3 the top of my head, all of the secondary considerations.
4 But let me go back to my understanding of the law.

5 One of them is an unmet long-felt need. That's
6 not there. And one is significant praise from others in
7 the field, which I also did not see.

8 Q. (By Mr. Limbacher) Did you review any
9 documents in reaching the conclusion in Paragraph 19?

10 A. I would say more that it's the lack of
11 documents.

12 Q. Did you do any research?

13 A. I did some. I looked around. I said, "Who are
14 these guys?" And some professors from Germany who wrote
15 a few papers, good. It was many hundreds of papers in
16 this area.

17 Q. So are you familiar with any systems that they
18 had implemented?

19 MR. CLAWSON: Objection. Form.

20 THE WITNESS: You mean actual deployments
21 outside of the German government? No.

22 Q. (By Mr. Limbacher) Can you turn to Paragraph
23 26 of your declaration, and then read the first sentence
24 to yourself.

25 A. Okay.

1 Q. How does one achieve high overall efficiency in
2 a multi-processor computing system?

3 MR. CLAWSON: Objection. Form.

4 THE WITNESS: How does one do that? Well, in
5 the case of Livermore and Los Alamos, one spends
6 billions of dollars. I was on the advisory board for
7 Livermore and Los Alamos for their high-performance
8 computing system acquisitions and designs, and there are
9 many, many aspects. Which aspect would you like to
10 explore?

11 Q. (By Mr. Limbacher) Well, can you give me some
12 background? You referred to the -- was it the Livermore
13 and Los Alamos project?

14 A. Livermore and Los Alamos are national
15 laboratories. They are leaders in high-performance
16 computing. And routinely, between Livermore and Los
17 Alamos and Oak Ridge have one -- place one or place two
18 on the top 500 fastest computers in the world.

19 Q. Okay.

20 Do you have an example of one of their systems
21 that you worked on?

22 A. I was -- Roadrunner was one that I was on the
23 advisory board when they did.

24 (Stenographer clarification.)

25 THE WITNESS: There was one ASCI Red came out

1 of Sandia. Roadrunner, I was on the advisory board when
2 Roadrunner was being built by IBM, for example.

3 Q. (By Mr. Limbacher) Okay.

4 And what was Roadrunner?

5 A. Roadrunner was an interesting computer because
6 it used the IBM PowerPC as using the nomenclature of our
7 patents as a computation node and the IBM cell processor
8 as the booster nodes. It was the first high-performance
9 computer that was considered a heterogeneous cluster
10 computer. It was early 2000s.

11 Q. So what types of things did you consult on for
12 that project?

13 A. I was one of the reviewers for it, actually. I
14 was on the -- I was on the science and technology panel.

15 (Stenographer clarification.)

16 THE WITNESS: I was on the science and
17 technology panel. And I was on the -- I believe still
18 the president's council for the national labs on that
19 one.

20 So I've -- I participated in -- I didn't do the
21 design, of course, that was IBM's design. I did
22 multiple reviews of the design for the national lab.

23 Q. (By Mr. Limbacher) And can you give me
24 examples of the types of challenges faced that you
25 consulted on?

1 MR. CLAWSON: Objection. Form.

2 THE WITNESS: Like I said, we did reviews of
3 the design. And issues that we discussed were memory
4 bandwidth issues. There were issues of how quickly you
5 could do checkpoints. There were issues of how one
6 programs the cell processors. The cell processors were
7 basically the same processor, closely related to the
8 accelerator that was in the Sony PS3, interestingly
9 enough.

10 So mainly discussions centered around
11 programmability, how one programs things.

12 Q. (By Mr. Limbacher) Okay.

13 And were the solutions to those problems
14 obvious to you?

15 MR. CLAWSON: Objection. Form.

16 THE WITNESS: Were they obvious? Yeah, I would
17 say they're -- they were difficult, but fairly obvious.
18 They centered around restructuring the program into
19 sub-tasks and which sub-tasks would be assigned. It was
20 different than previous -- previous machines that the
21 labs acquired.

22 Q. (By Mr. Limbacher) Can you please turn to
23 Paragraph 31 of your declaration.

24 A. Okay.

25 Q. Read that to yourself, and let me know when

1 you're ready.

2 A. Okay.

3 Q. I'm going to use the word POSITA, P-O-S-I-T-A.

4 How did you arrive at this definition for a
5 POSITA?

6 A. I thought about what a person in this time
7 frame would -- typically their education would be, and
8 thought about the undergraduates that I taught during
9 that time frame, and undergraduates in that time frame
10 would have been familiar with using both CPUs and also
11 DSPs and FPGAs and other, what we're calling booster
12 nodes, to do the computation. So someone with a
13 bachelor's degree and some experience would understand
14 how this works.

15 Q. Why did you focus on distributed computing
16 technologies as the -- as the industry -- well, first
17 let me walk back.

18 Based on this paragraph, are you saying that
19 the proper industry is distributed computing
20 technologies?

21 MR. CLAWSON: Objection. Form.

22 THE WITNESS: I'm sorry. You're saying?

23 Q. (By Mr. Limbacher) Do you --

24 A. Does the word distributed computing
25 technologies in this paragraph -- it is not in this

1 paragraph.

2 Q. I'm looking about halfway down. It says,

3 "Approximately three years of" --

4 (Unreportable cross-talk.)

5 (Stenographer clarification.)

6 Q. (By Mr. Limbacher) So I'm going to repeat
7 this: "Approximately three years of industry experience
8 or research with distributed computing technologies."
9 And this is from Paragraph 31 of Dr. Long's declaration.

10 A. I see.

11 Can you ask the question? I think I misheard
12 you the first time.

13 Q. Yes, sir.

14 Why did you choose to focus on distributed
15 computing technologies for this POSITA definition?

16 A. Because this is not embedded computing; it's
17 distributed computing. We have multiple nodes, multiple
18 types. That's almost a classic definition of
19 distributed computing.

20 Q. So why distributed computing and not
21 heterogeneous computing?

22 A. Because almost all distributed computing is
23 heterogeneous at this point.

24 Q. Turn to Paragraph 32 of your declaration.

25 A. Okay.

1 Q. The last sentence there, you stated, "A POSITA
2 would have known of the basic design principles of fault
3 tolerance and efficient scheduling/task assignments."

4 Did I read that correctly?

5 A. Yes.

6 Q. What are the basic design principles of fault
7 tolerance?

8 A. What are the basic design principles of fault
9 tolerance?

10 Q. Yes.

11 A. Well, this is something I typically would --
12 taught in my junior year operating systems course.
13 Fault tolerance is done multiple ways. Typically
14 through redundancy. You have more than one copy of
15 something so that if there's a failure, they are done
16 through verification, testing, and so forth and so on.
17 You want to make sure that your code is as reliable as
18 possible and bug free and so forth and so on.

19 But primarily through redundancy, in the case
20 of data storage, it's done through --

21 (Stenographer clarification.)

22 THE WITNESS: -- erasure codes or
23 error-correcting codes.

24 In terms of computational redundancy, you
25 can -- the classic example, of course, is the Space

1 Shuttle where they had three copies of every computation
2 running and they would vote to see which one to do.
3 That's still done in spacecraft today. It's also still
4 done in some aircraft.

5 So typically fault tolerance is going to be
6 done through some kind of redundancy so that if one or
7 more of the modules, processors, memories, whatever,
8 fails, it can -- the error can either be corrected or
9 the process restarted.

10 Another method is what's called checkpoint
11 restart, which is done in high-performance computing.
12 Because when we have a very large number of processors,
13 you can approximate the failure rate of processors,
14 assuming exponential distribution, by just dividing by
15 the number of processors.

16 So if you have a processor that is going to
17 crash once a year, and you have a thousand processors,
18 that means you have a thousand crashes a year, which
19 means three times a day, a processor is going to crash.
20 Approximately. That's just the math.

21 And so what is typically done is several times
22 a day, the memory of the processor will be dumped to
23 storage, and if it crashes, it can restart from where
24 the crash happened.

25 Q. (By Mr. Limbacher) Okay.

1 Is it your opinion that the claims of the
2 '442 patent relate to fault tolerance?

3 MR. CLAWSON: Objection. Form.

4 THE WITNESS: I would have to go back and look
5 and see if any of them mention fault tolerance.

6 Q. (By Mr. Limbacher) Please take all the time
7 you need to review.

8 A. Go ahead.

9 Q. So where in the claims of the '442 patent --
10 scratch that.

11 What parts of the claims of the '442 patent
12 relate to fault tolerance?

13 MR. CLAWSON: Objection. Form.

14 THE WITNESS: So when I read that, the dynamic
15 reassignment enables fault tolerance. It's actually
16 kind of cool. It doesn't require it though. There's no
17 requirement for fault tolerance. But if I have one of
18 the, say, booster nodes fail, then I could do a dynamic
19 reassignment.

20 Q. (By Mr. Limbacher) Turning back to this, is
21 your declaration, Deposition Exhibit B, Exhibit 1003 in
22 the IPR still, Paragraph 32, what are the basic design
23 principles of efficient scheduling/task assignments?

24 A. Well, there is an enormous literature on
25 scheduling theory. I think -- I don't know if we gave

1 you a copy of the old book from the 1970s called
2 Operating Systems Theory.

3 Efficient scheduling means -- well, let me take
4 a step back.

5 Scheduling comes from operations research,
6 which is a branch of mathematics developed, I think,
7 around World War I/World War II for job scheduling
8 and --

9 (Stenographer clarification.)

10 THE WITNESS: -- factory floors.

11 And this includes the allocation of resources,
12 which -- which things go to which machine for making
13 things.

14 So that's task assignment. That's a kind of
15 scheduling. Your task assignment.

16 Computer scientists, particularly in the 1960s
17 and 1970s, developed an extensive theory for assigning
18 resources. They abstract these things as things like
19 the knapsack problem, the bin-packing problem, the
20 graph-coloring problem --

21 (Stenographer clarification.)

22 THE WITNESS: The graph-coloring problem.

23 And, again, there's enormous literature on
24 scheduling is whether long-term scheduling or short-term
25 scheduling.

1 Long-term scheduling is deciding which process,
2 for example, runs on which computer. Short-term
3 scheduling typically involves how long a task or process
4 runs on a processor. But there's other issues,
5 including processor affinity, whether you move a process
6 from one processor to another and so forth. But, again,
7 this is -- this is Undergraduate Operating Systems,
8 Chapter 3, I think.

9 Q. (By Mr. Limbacher) Okay.

10 If there's enormous literature on the topic of
11 scheduling theory, would that tend to indicate that it
12 is a complicated topic?

13 A. It would indicate --

14 MR. CLAWSON: Objection. Form.

15 THE WITNESS: It would indicate, at this point,
16 that it is a well understood topic. Nobody works on it
17 anymore. It's done.

18 Q. (By Mr. Limbacher) So there's no longer any
19 improvements in scheduling theory?

20 MR. CLAWSON: Objection. Form.

21 THE WITNESS: We would have to go look at the
22 publications, but I have not seen a significant advance
23 in scheduling theory in many years.

24 Q. (By Mr. Limbacher) Would a scheduler
25 implementation be the same for a desktop computer as it

1 would be for a cluster computer?

2 MR. CLAWSON: Objection. Form.

3 THE WITNESS: Happy to answer that. The
4 scheduler that runs on a Linux desktop computer is the
5 same scheduler that runs on a cluster computer. They
6 all run Linux.

7 The schedule that runs on the Windows computer
8 that I'm using right now is similar to, but not exactly
9 the same as a Linux scheduler. A --

10 (Stenographer clarification.)

11 THE WITNESS: Correct. I said the Windows
12 computer that I'm looking at right now has a different
13 scheduler. My recollection is that it's more derived
14 from the --

15 (Stenographer clarification.)

16 THE WITNESS: -- VAX VMS and Windows NT
17 scheduler, but it may have been replaced, but it uses
18 the same scheduling principles. We would have to look
19 up the details.

20 (Stenographer clarification.)

21 THE WITNESS: The FreeBSD scheduler,
22 F-R-E-E-B-S-D -- one word -- scheduler is more priority
23 based the last time I looked, but, again, the same set
24 of principles.

25 There are such things as real-time schedulers.

1 There is a real-time scheduler in a special version of
2 Linux that is not widely used. There are real-time
3 operating systems that have real-time schedulers. These
4 include both hard real-time and soft real-time
5 scheduling.

6 Q. (By Mr. Limbacher) So you are referring to
7 kernel-level schedulers; correct?

8 A. In the last few minutes, those are kernel-level
9 schedulers. Higher level schedulers are the tasks
10 that -- typically the task assignment schedulers that
11 admit tasks into the system, those tend to be simpler.
12 And assigned at a much coarser granularity.

13 Those are the ones that you would see in things
14 like Hadoop or in Google's -- whatever Google is calling
15 their system this time, or what's being done in
16 cloud-era-based systems where we're assigning entire
17 processes across things.

18 There's other scheduling algorithms that
19 schedule virtual machines across physical machines.
20 Again, these are coarser-grained decisions.

21 Q. (By Mr. Limbacher) Can you turn to Paragraph
22 52 of your declaration and read it to yourself.

23 A. Okay.

24 Q. You see where you refer -- well, first of all,
25 this paragraph is discussing Lippert; is that correct?

1 A. 104, I believe, is Lippert. Let me check.

2 Yes.

3 Q. Okay.

4 In Paragraph 52, you refer to each part of the
5 computation task.

6 Do you see that?

7 A. Yes.

8 Q. What is a part of a computation task?

9 A. Sub-task.

10 Q. So your view is that's the same as a sub-task?

11 A. Well, when we're talking about it going to a
12 booster node or a computation node, I would call that a
13 sub-task.

14 Q. Okay.

15 What are you basing that on?

16 A. Well, I can't -- again, except for the Xeon
17 Phi, which is fairly unique, I can't run code that's
18 designed for a DSP on a Pentium.

19 So if I have code that's designed for a booster
20 node, it's going to have to run on a booster node. And
21 I would call that a sub-task.

22 Q. Please turn to Paragraph 53 of your
23 declaration, and read it to yourself.

24 A. Okay.

25 Q. You state that, "The Journal of Supercomputing

1 is a well-regarded journal in the art"; is that correct?

2 A. That's correct.

3 Q. What are you basing that on?

4 A. It's where a lot of my colleagues who publish
5 in supercomputing publish their work, and I have high
6 regard for their work. The citation index seems to be
7 pretty good. The reviewing seems to be pretty good,
8 considering I was one of the reviewers, and I'm a pretty
9 harsh reviewer.

10 Q. Have you published anything in that journal?

11 A. I typically publish in conferences, so I
12 publish in the supercomputing conference, not in the
13 Journal of Supercomputing.

14 Q. How frequently do you read the Journal of
15 Supercomputing?

16 A. I skim it -- well, not now, since I retired. I
17 would skim it maybe quarterly, every half year, I'd see
18 if there's anything interesting with what I was working
19 on at the time. My grad students read it more often
20 than I do.

21 Q. Can you please turn to Paragraph 116 of your
22 declaration and read it to yourself.

23 A. Okay.

24 Q. Is it your opinion that Kambatla discloses a
25 heterogeneous computing system?

1 A. Yes.

2 Q. What are you basing that on?

3 A. The clusters that Kambatla runs on, including
4 some in my lab when we were running, were heterogeneous.
5 They included Intel of various types, as well as AMD
6 processors with different capabilities, memory sizes,
7 vector units, and so forth.

8 Q. I'm sorry. Did you refer to your lab?

9 A. Yeah.

10 Q. Is your lab discussed in the Kambatla
11 disclosure?

12 A. No. We used -- we used YARN, which is what the
13 Kambatla is based on, for running our Hadoop cluster.

14 Q. Where in the Kambatla disclosure does it
15 indicate that it relates to heterogeneous computing
16 systems?

17 A. I will have to go and look at Kambatla for you.

18 Q. Okay.

19 Kambatla is Deposition Exhibit E. It is
20 Exhibit 1006 in the IPR.

21 A. Okay. Paragraph 89. Keep going. Paragraph
22 94. All right. Let's talk about those.

23 Q. Okay.

24 What in Paragraph 89 tells you that it relates
25 to heterogeneous computing system?

1 A. It describes two different kinds of processors,
2 a PowerPC and a Intel processor.

3 Q. Okay.

4 Where does it say that both will be used in the
5 system?

6 MR. CLAWSON: Objection. Form.

7 THE WITNESS: Doesn't say they won't. That
8 would be easy to do.

9 Q. (By Mr. Limbacher) But you --

10 A. On the right -- run the right binary on the
11 right processor. It's easy enough to do.

12 Q. So you're saying it could be done, but Kambatla
13 doesn't disclose it.

14 A. It discloses two different kinds of processors,
15 include any type of device --

16 (Stenographer clarification.)

17 THE WITNESS: -- that is accessible by the
18 processor.

19 For me, this says that I can have multiple
20 kinds of processors on there. And even if I don't, I
21 can have multiple types of Intel processors on here.
22 This is just giving examples.

23 Q. (By Mr. Limbacher) So what's the first
24 sentence of Paragraph 89?

25 A. The processor may be, for example, a

1 conventional microprocessor, such as Intel Pentium --
2 which nobody uses anymore -- or a Motorola PowerPC
3 processor.

4 Q. This is referring to processor, singular;
5 correct?

6 A. I think you're -- that would not be a cluster
7 if there were only one processor. They're talking about
8 a processor; what kind of processor can you use?

9 To say there was only one processor would make
10 this nonsensical.

11 Q. That's referring to Figure 8; correct?

12 A. Let me look.

13 MR. CLAWSON: Objection. Form.

14 THE WITNESS: What page is Figure 8 on?

15 Q. (By Mr. Limbacher) Figure 8 is on page -- it's
16 PDF Page 10.

17 A. This is one node. There is a processor with a
18 memory. But we're talking about scheduling clusters.
19 That's Figure 5. So a node is represented by Figure 8.

20 That's a high school representation of a
21 computer.

22 Q. (By Mr. Limbacher) Are you still looking or
23 are you ready?

24 A. No. No. I looked at 8, and that's the high
25 school representation of one single node.

1 Q. Do you agree there is no express disclosure in
2 Kambatla of using two different types of processors in
3 the same system?

4 MR. CLAWSON: Objection. Form.

5 THE WITNESS: I would have to take the time and
6 reread Kambatla.

7 Q. (By Mr. Limbacher) Did you do that prior to
8 drafting your opinion in this matter?

9 A. I did, but that was --

10 MR. CLAWSON: Objection. Form.

11 THE WITNESS: I did, but that was almost a year
12 ago.

13 Q. (By Mr. Limbacher) So sitting here today,
14 without rereading the entire document, you're not aware
15 of any express disclosure of using two different types
16 of processors at the same time -- or, I'm sorry -- in
17 the same system in Kambatla?

18 MR. CLAWSON: Objection. Form.

19 THE WITNESS: I'm going to push back and ask
20 you to define processor types.

21 Q. (By Mr. Limbacher) Well, how do you interpret
22 the term?

23 A. Well, if I have an Intel Core i7 and a Core i9
24 in my cluster, which is very common, then they are
25 different processor types. If I have AMD and Intel in

1 my cluster, they are different -- very commonly, they
2 are different processor types.

3 Q. Does Kambatla disclose either of those
4 scenarios?

5 A. It says, for example, you could have an Intel
6 processor or a power PC processor. It's giving you an
7 example. It does not require that the processors be
8 different.

9 Q. Okay.

10 But sitting here today, you don't know of
11 anything -- anywhere that it discloses that they will be
12 different types?

13 MR. CLAWSON: Objection. Form.

14 THE WITNESS: I do not recall seeing anything
15 that requires them to be different processor types.

16 Q. (By Mr. Limbacher) Okay.

17 Can you please turn to Paragraph 119 of your
18 declaration and read it to yourself. And your
19 declaration is Deposition Exhibit B, Exhibit 1003 in the
20 IPR.

21 A. Okay. Go ahead.

22 Q. What minor software code would be needed to
23 implement the known resource management techniques you
24 refer to in this paragraph?

25 MR. CLAWSON: Objection. Form.

1 THE WITNESS: Correct. What's the question?

2 Q. (By Mr. Limbacher) I'm asking what minor
3 software code would be needed to implement the known
4 resource management techniques you refer to?

5 MR. CLAWSON: Same objection.

6 THE WITNESS: I would need to have Git
7 repository for Kambatla to show you or I can't answer
8 that question for you that easily.

9 Q. (By Mr. Limbacher) Did you review that
10 software before writing your opinion on this?

11 A. I reviewed --

12 MR. CLAWSON: Objection. Form.

13 THE WITNESS: I reviewed the documentation and
14 YARN says that it makes it significantly simpler and it,
15 in Kambatla, discloses doing this for both computational
16 nodes and GPU nodes, which are, in fact, heterogeneous,
17 to address your earlier question.

18 It's just a question of whether or not I
19 repeatedly apply the metrics, and Kambatla does that. I
20 don't see a big change as being necessary, but we would
21 have to review the source code to answer the question
22 specifically.

23 Q. (By Mr. Limbacher) But you did not do that
24 before writing your opinion that it would be minor;
25 correct?

1 MR. CLAWSON: Objection. Form.

2 THE WITNESS: I did not review the source code
3 for Kambatla.

4 Q. (By Mr. Limbacher) So you just made the
5 assumption that it would be minor software code;
6 correct?

7 MR. CLAWSON: Objection. Form.

8 THE WITNESS: I took them at their word that it
9 would make it simpler to do. That's their express --
10 that's the expressed goal of it.

11 Q. (By Mr. Limbacher) Does Kambatla say that it
12 would be minor software code?

13 A. Well --

14 MR. CLAWSON: Objection. Form.

15 THE WITNESS: YARN says. So that's Exhibit
16 112. Kambatla is a scheduler modification of YARN.

17 Q. (By Mr. Limbacher) I'm sorry. I didn't
18 follow.

19 A. Okay. So YARN is part of the Apache Hadoop
20 project, which is for scheduling tasks across very large
21 clusters of computers, hundreds/thousands of computers.
22 And Kambatla is a purported improvement on the
23 scheduling algorithm for YARN. YARN explicitly states
24 that it makes the effort of building distributed
25 computing systems significantly simpler because it

1 provides a scheduling infrastructure. Changing that
2 scheduling algorithm should not be difficult.

3 Q. But, again, sitting here today, you didn't
4 verify what software code would be necessary before
5 writing that opinion; correct?

6 MR. CLAWSON: Objection. Form.

7 THE WITNESS: I did not review the source code
8 for YARN or Kambatla.

9 Q. (By Mr. Limbacher) Can you please read
10 Paragraph 125 of your declaration to yourself.

11 A. Go ahead.

12 Q. How would combining the resource manager of
13 Lippert Budenske with the central resource manager of
14 Kambatla enable efficient scheduling of multiple tasks
15 rather than one computational task?

16 MR. CLAWSON: Objection. Form.

17 THE WITNESS: I'm -- yeah. Say that again.
18 I'm struggling to understand what you're asking.

19 Q. (By Mr. Limbacher) So -- so top of Page 77,
20 this is the first full sentence, you say, "A POSITA
21 implementing Lippert-Budenske-Kambatla would have
22 combined the resource of Lippert-Budenske with the
23 central resource manager of Kambatla"; is that correct?

24 A. Correct.

25 Q. And then in the last sentence, you state, "The

1 POSITA would have done so to gain additional advantages
2 of Kambatla, namely to enable official scheduling of
3 multiple tasks, rather than one computational task";
4 correct?

5 A. Correct.

6 Q. How would that combination enable efficient
7 scheduling of multiple tasks rather than one
8 computational task?

9 A. That's what YARN does; YARN lets you schedule
10 lots of tasks.

11 Q. Is it your view that --

12 (Stenographer clarification.)

13 Q. (By Mr. Limbacher) Is your opinion that
14 Lippert-Budenske, without YARN or Kambatla, is not
15 capable of scheduling multiple tasks?

16 MR. CLAWSON: Objection. Form.

17 THE WITNESS: I think that's a
18 mischaracterization. What we're talking about here is
19 concurrently scheduling multiple tasks. I think Lippert
20 and Budenske are silent about whether the tasks are
21 going to be scheduled concurrently or sequentially.

22 Q. (By Mr. Limbacher) So when you're referring to
23 Kambatla, you keep referring to YARN; correct?

24 A. Correct.

25 Q. Are the aspects of YARN that you're relying on

1 disclosed in Kambatla?

2 MR. CLAWSON: Objection. Form.

3 THE WITNESS: I would have to study Kambatla to
4 make sure everything is in there.

5 Q. (By Mr. Limbacher) Did you do that prior to
6 drafting your opinion?

7 MR. CLAWSON: Objection. Form.

8 THE WITNESS: I did, about a year ago, but,
9 again, it's been a long time.

10 Q. (By Mr. Limbacher) Now a good time for a break
11 or would you like to keep going?

12 A. Now is a good time for a break. It's been
13 about an hour.

14 MR. LIMBACHER: Let's go off the record,
15 please.

16 THE STENOGRAPHER: We're off the record. The
17 time is 10:54 a.m.

18 (Recess taken from 10:54 a.m. to 11:03 a.m.)

19 THE STENOGRAPHER: We're back on the record.
20 The time is 11:03 a.m.

21 Q. (By Mr. Limbacher) Just 'cause I have to ask:
22 Did you speak with anyone about the deposition or the
23 IPR during the break?

24 A. I did not.

25 Q. We're still in your declaration. This is

1 Exhibit B, Exhibit 1003 in the IPR.

2 Can you turn to Paragraph 57 of your
3 declaration and read it to yourself.

4 (Stenographer clarification.)

5 THE WITNESS: Okay.

6 Q. (By Mr. Limbacher) So I'm looking at the top
7 of Page 28, the first filled-in black bullet point says,
8 "A resource manager module implemented as a daemon, as
9 taught by Lippert, e.g. EX1004, Paragraph 14."

10 Is that correct?

11 A. That's correct.

12 Q. Where in that paragraph in Lippert does it
13 teach a resource manager module implemented as a daemon?

14 MR. CLAWSON: Objection. Form.

15 Q. (By Mr. Limbacher) So this is referring to
16 Deposition Exhibit C, Exhibit 1004 in the IPR, Lippert.

17 A. Page 2, top of Column 1, at the last word,
18 virtualized hardware component or a daemon.

19 Q. That is referring to the resource manager;
20 correct?

21 A. That's how it appears to me, especially, "The
22 resource manager may be formed of" -- "may be formed by
23 any one of microprocessor or hardware component,
24 virtualized component or a daemon."

25 Q. Okay.

1 Please turn to Paragraph 59 of your declaration
2 and read it to yourself.

3 A. Okay.

4 Q. I'm focused on the last two lines.

5 You say, "The combination would reduce
6 execution time of computing iterations"; is that
7 correct?

8 A. That's correct.

9 Q. How would it reduce execution time of computing
10 iterations?

11 A. So this is related to each iteration, you get
12 some information, some feedback, about how the previous
13 iteration computed.

14 And by doing the dynamic redistribution, the
15 hope is that the next iteration will be faster, and
16 taken together, the application, the task, and in the
17 case of Budenske, it's identifying targets, is going to
18 be done more rapidly than if you had not reassigned
19 those resources.

20 Q. Does Lippert teach using computing iterations?

21 A. Lippert, the original Lippert?

22 Q. Yes. The original Lippert, that would be --
23 make sure I have the right one.

24 A. I think it does. It hints at it.

25 Q. Okay.

1 Where does it hint at it?

2 A. That's going to be me taking a minute to look
3 here.

4 Q. Okay.

5 A. IT department needs to get me a better
6 computer. Paragraph 93, Figures 4 and 5, "A person" --
7 (Stenographer clarification.)

8 THE WITNESS: Paragraph 93, referring to
9 Figures 4 and 5, "Persons skilled in the art appreciates
10 that any of the steps that can be performed
11 iterative" -- "any of the steps can be performed
12 iteratively in a different order and may comprise
13 further familiar sub-steps. For instance, Step 102 may
14 be performed before Step 101, which results in a
15 computation of a first part of the computation task, and
16 assignment of one booster to one computation node, and
17 finally the computation of the second part of the
18 computation task, et cetera, et cetera. Hence, the
19 Booster B returns to the computed" -- "returns the
20 computed result back to the computation of CN, and the
21 computation nodes CN may use a return" --

22 (Stenographer clarification.)

23 THE WITNESS: CN, all caps.

24 -- "may use the return value for computation of
25 further computation tasks and may again forward at least

1 a further part of a computation task to at least one of
2 the boosters."

3 Q. (By Mr. Limbacher) So if Paragraph 93, let's
4 say for the sake of argument, was not referring to
5 computing iterations, would you agree that the -- that
6 the mode of --

7 (Stenographer clarification.)

8 Q. (By Mr. Limbacher) Let me start over.

9 Okay.

10 So let's go back to Paragraph 59 of your
11 declaration. This is Deposition Exhibit B.

12 A. Um-hum.

13 Q. Exhibit 1003 from the IPR.

14 And so as we discussed, your point is that this
15 combination would reduce the execution -- sorry --
16 reduce execution time of computing iterations as taught
17 by Lippert; is that correct?

18 A. Correct.

19 Q. So if the paragraph we just discussed in
20 Lippert, that Paragraph 93, if that was not referring to
21 computing iterations, is there anything else in Lippert
22 you see that you think teaches computing iterations?

23 MR. CLAWSON: Objection. Form.

24 THE WITNESS: Let me go back to Lippert.

25 You guys hear that funny ding ding noise every

1 once in a while. Might be my headset.

2 MR. LIMBACHER: Let me mute and see if it goes
3 away. It might be my AC.

4 THE WITNESS: It did it again.

5 Q. (By Mr. Limbacher) Okay. If it did it again,
6 it wasn't me.

7 A. It just did it again.

8 Well, in the abstract, the assignment takes
9 place dynamically at runtime.

10 Q. What are you referring to?

11 A. Lippert abstract, it says, "The assignment
12 takes dynamically place" -- grammar -- "at runtime."

13 Q. Okay.

14 Is it your position that that teaches computing
15 iterations?

16 A. Dynamically doing things at runtime, that's --
17 would tell a computer scientist that things are getting
18 repeated. And Paragraph 93 tells me that things happen
19 iteratively and more than once.

20 Q. Okay.

21 A. And, again, of course we have Budenske, which
22 does explicitly teach iteration.

23 Q. Let's go back to your declaration. This is
24 Deposition Exhibit B, Exhibit 1003 in the IPR. Please
25 turn to Paragraph 61.

1 And this is on Page 31, the last couple
2 sentences, you'll see where it says, "A POSITA would
3 have recognized that Lippert and Budenske follow the
4 same principles of efficient processing of a task."

5 Did I read that correctly?

6 A. You did.

7 Q. What are the principles of efficient processing
8 of a task that you are referring to?

9 A. Assigning sub-tasks that would benefit from
10 what we were calling booster nodes. So if we have
11 something that can be done faster using a DSP or FPGA or
12 commonly we do it with encryption these days, assign it
13 to a special purpose processor to do that task, we
14 would. And both Budenske and Lippert teach that.

15 Q. Please turn to Paragraph 63 of your
16 declaration.

17 Do you see where it refers to "code blocks"?

18 A. I do.

19 Q. What is a code block?

20 A. Code block is -- in some cases, could be
21 considered a term of art, but what we're talking about
22 here are the sub-tasks. Code block is code that is
23 performing a well-defined task.

24 Q. Are all code blocks sub-tasks?

25 A. No.

1 Q. Are all sub-tasks code blocks?

2 A. All sub-tasks are made up of code blocks.

3 Q. Okay.

4 Do sub-tasks include things other than code
5 blocks?

6 MR. CLAWSON: Objection. Form.

7 THE WITNESS: Interesting question. A sub-task
8 could, I suppose, have metadata associated with it
9 that's not executable code that describes it.

10 This would be, for example, in an operating
11 system context of the loader and the metadata would
12 include where in memory to load code, how much memory to
13 allocate with the code, the size of the stack, the size
14 of the heap, whether there's any reallocate --
15 relocation information that needs to be done and that
16 metadata would not be code, but it would be used to
17 inform the placement and allocation of the code.

18 Q. (By Mr. Limbacher) Do you see a little bit
19 further down where it says, "Lippert recognizes that
20 each code has highly scaleable and less scaleable
21 complex elements"?

22 A. Yes.

23 Q. Will that code change between iterations?

24 MR. CLAWSON: Objection. Form.

25 THE WITNESS: Does the code itself change

1 between iterations? The answer is no.

2 Q. (By Mr. Limbacher) Will the scalability of
3 that code change between iterations?

4 MR. CLAWSON: Objection. Form.

5 THE WITNESS: We're saying code and not code
6 block here.

7 The part of the code that you're executing
8 would change between iterations, which means you are
9 using the less scaleable or the more scaleable parts of
10 the code.

11 Q. (By Mr. Limbacher) So you're saying sub-tasks
12 will include sections of code, some of which are more
13 scaleable and some of which are less scaleable; is that
14 correct?

15 A. I'm reading my quote here, which says, "Each
16 code." And the code is going to have multiple code
17 blocks in it; not saying a sub-task here, saying each
18 code has scaleable and less scaleable code elements.
19 That's from Lippert.

20 The complex or the highly scaleable code is
21 going to go to different processors. All right?

22 I mean, if it's highly scaleable, it's -- in
23 the case of Lippert, it means it's parallellizeable and
24 can be used in the --

25 (Stenographer clarification.)

1 THE WITNESS: -- in the 512-bit vector units of
2 the Xeon Phi processor. It's vectorizable, is what
3 they're saying there. And the less scaleable is going
4 to be stuff that has to be run on the normal computed
5 nodes.

6 But this is the code. This isn't a sub-task.
7 The sub-task is what's going to get farmed out to a
8 booster node.

9 Q. (By Mr. Limbacher) So is it your position that
10 this code is different from a code block?

11 A. Yes.

12 Q. And that this code is different from a
13 sub-task?

14 A. Yes.

15 Q. So even though it's not referring to a code
16 block or a sub-task, you still feel that that would
17 provide a motivation to combine -- I'm sorry, a
18 motivation to change which sub-tasks are assigned to a
19 first part and a second part?

20 A. Yes.

21 MR. CLAWSON: Objection. Form.

22 Q. (By Mr. Limbacher) How are you making that
23 leap?

24 A. Each code has highly scaleable and less
25 scaleable complex elements. That tells me that there's

1 multiple types. That's not a leap. That's very clear.

2 Q. So the first part of that sentence, you're
3 referring to code. The second part of that sentence,
4 you're referring to sub-tasks.

5 You're telling me code is different from
6 sub-tasks, so how are you getting from talking about the
7 scalability of the code to changing where sub-tasks are
8 assigned?

9 MR. CLAWSON: Objection. Form.

10 THE WITNESS: Okay. We need to get into
11 terminology.

12 Q. (By Mr. Limbacher) Okay.

13 A. A code, in high-performance computing, is a
14 program. Program can be thousands or millions of lines
15 long. That code is made up of many, many, many code
16 blocks. Those code blocks can be grouped into
17 sub-tasks. Those sub-tasks, some of them, will benefit
18 from, say, vectorization or from a DSP or from a GPU.
19 Other parts of the code will not benefit from that.

20 So what we're saying here is each code has
21 highly scaleable and less scaleable complex elements.
22 So the code is a big thing. And that big thing can be
23 subdivided into many small things, and those smaller
24 things can be assigned to the booster nodes to increase
25 performance.

1 Q. Please turn to Paragraph 65 of your declaration
2 and read the first sentence to yourself.

3 A. Okay.

4 Q. What software code would be needed to combine
5 Lippert and Budenske?

6 A. Yes.

7 Q. I'm asking what software code would be needed
8 to combine Lippert and Budenske?

9 A. Are you asking me to write that code?

10 Q. I'm asking did you consider what code would be
11 needed when you made that statement?

12 A. I did. I didn't write the lines of code that
13 would need to be written because I have neither the
14 source code to Lippert, nor Budenske, and they're
15 written in different programming languages.

16 Q. Okay.

17 You said you considered what the code would be,
18 so can you tell me what software -- what source code
19 would be? Give me an overview.

20 MR. CLAWSON: Objection. Form.

21 THE WITNESS: Sure. If I have Lippert, and in
22 light of Budenske, and it says, Hey, you can do this
23 assignment more than once, and Lippert has a mechanism
24 for doing the initial assignment, and it has a mechanism
25 for receiving information, then I can repeat, it's a

1 simple "do it again" loop, the code that's already in
2 Lippert for doing the initial assignment.

3 If I decide that's not good enough because
4 perhaps Lippert's initial assignment wasn't very good,
5 then I look at Budenske, and Budenske breaks things up
6 into DAGs. I think they call them something else, DDGs.
7 A DDG is --

8 (Stenographer clarification.)

9 THE WITNESS: A DDG, we also call them DAG,
10 it's a directed acyclic graph where the nodes of the
11 graph represent sub-tasks, and the edges of the graph --
12 graph represent dependencies. These are temporal
13 dependencies among these things, which means that you
14 can place on that graph what's called a partial order,
15 and the partial order defines the degree of concurrency
16 that's possible.

17 Anyhow, so now if I have Budenske, and I've
18 decided that Lippert's assignment wasn't good enough,
19 Budenske has another assignment mechanism, which because
20 it's 1998 and it was fashionable at the time, they
21 compute a bunch of different layouts, a bunch of
22 different graphs using what's called a genetic
23 algorithm, and each -- after each iteration completes,
24 they get the information back and they go, Okay, how was
25 that? Well, maybe I can make that better with a

1 different DDG, and it chooses a different DDG and does a
2 different assignment.

3 Q. (By Mr. Limbacher) Would the modifications
4 you're referring to work for any type of processor?

5 MR. CLAWSON: Objection. Form.

6 THE WITNESS: Do you mean any type of
7 computation node? Any type of -- certainly Budenske has
8 the flexibility to work with any number of types of
9 processors.

10 Because it specifically defines
11 heterogeneous -- this heterogeneous architecture more
12 rigorously than Lippert.

13 Q. (By Mr. Limbacher) I apologize. Give me one
14 second. Sorry about that disruption. Something on my
15 desk came loose.

16 So what about communication between -- let's
17 say you have to transfer a sub-task from a computation
18 node to a booster node. What sort of communication
19 changes would be needed in order to implement that?

20 MR. CLAWSON: Objection. Form.

21 THE WITNESS: Say it one more time, since you
22 broke up a little bit.

23 Q. (By Mr. Limbacher) So my understanding is that
24 you're saying this -- this combination in a further
25 computing iteration would be able to take a sub-task

1 away from, say, a computation node and put it onto a
2 booster node in a further iteration; is that correct?

3 A. Correct.

4 Q. Okay.

5 What changes to the communication systems would
6 be necessary to accommodate that?

7 MR. CLAWSON: Objection. Form.

8 THE WITNESS: You shouldn't need to change the
9 communication system. Existing communication system
10 should work.

11 Q. (By Mr. Limbacher) Does Budenske provide
12 sufficient communication infrastructure between the
13 different types of processes?

14 MR. CLAWSON: Objection. Form.

15 THE WITNESS: Budenske describes the
16 architecture for the specific instance they're doing.
17 As I recall, it is a VMEbus, which is, interestingly,
18 still used by the military, and that will provide
19 communication between the cards. So yeah, the -- it
20 will work.

21 Q. (By Mr. Limbacher) Is the communication
22 between -- so let's take a step back.

23 What are the types of processors that Budenske
24 is using?

25 A. Budenske had -- I'll have to go open it up.

1 It's Number 17.

2 It had PowerPC RISC process, as an example, it
3 says. "For example, two types are" -- "may be SHARC DSP
4 and the PowerPC RISC processor" -- heterogeneous
5 parallel architecture. Those were -- "communications
6 among these processors are assumed to be symmetric,"
7 which is nice. That's because they're in a VMEbus. And
8 conflict-free, which is also nice.

9 Q. Where are you reading from right now?

10 A. Page 392 at this point.

11 Q. Can you please reread that sentence and make
12 sure you read all the words.

13 A. Okay. Sure.

14 Top of Page 392: "All the processors of all
15 types will have communications, paths, to one another,"
16 which means everything can talk to everything.

17 Q. Then the next sentence, please.

18 A. "Communications among processors of the same
19 type are assumed to be symmetric in the sense that the
20 conflict-free time for any pair of processors to
21 communicate is the same."

22 Q. Where are only processors of the same -- why is
23 communication only symmetric for processors of the same
24 type?

25 A. It says it's symmetric in the sense that they

1 are conflict-free. I assume it is a property of the
2 Mercury Daughter board, but I don't have a diagram of
3 the board to tell you.

4 Q. Okay.

5 So you don't know why Budenske has that
6 limitation?

7 MR. CLAWSON: Objection. Form.

8 THE WITNESS: I'm not sure that's a limitation.
9 It's a feature of this particular implementation.

10 Q. (By Mr. Limbacher) Well, the --
11 (Stenographer clarification.)

12 Q. (By Mr. Limbacher) -- the SHARC, S-H-A-R-C,
13 DSP processors of Budenske, will they have the same
14 bandwidth between each other as they have with the
15 PowerPC RISC, R-I-S-C, processors?

16 MR. CLAWSON: Objection. Form.

17 THE WITNESS: I do not know. I don't know if
18 that data is in this paper.

19 Q. (By Mr. Limbacher) Would that affect how you
20 would implement sub-task scheduling?

21 MR. CLAWSON: Objection. Form.

22 THE WITNESS: The bandwidth? Probably not.
23 Sorry. Something on my desk just fell apart.

24 For sub-task scheduling, this kind of
25 application, the bandwidth and the latency are not going

1 to be the dominant factors.

2 Q. (By Mr. Limbacher) You say this kind of
3 application.

4 Would it be a dominant factor in other
5 applications?

6 MR. CLAWSON: Objection. Form.

7 THE WITNESS: One can always make up an
8 application where latency or bandwidth are the dominant
9 things.

10 Q. (By Mr. Limbacher) Can you give me one,
11 please.

12 A. Networking is where I care most about latency.
13 If I'm doing real time -- real time things, latency is
14 what I care about the most.

15 If I'm doing large data transfers, then
16 bandwidth is what I care about the most.

17 Q. So if you were processing large amounts of
18 data, using a system like Budenske's, would that
19 bandwidth limitation affect how you would perform the
20 sub-task scheduling?

21 MR. CLAWSON: Objection. Form.

22 THE WITNESS: No, not that this -- not in this
23 case. Because what I'm doing is using DSPs. I'm doing
24 a lot of computation on what I would consider a medium
25 amount of data. DSPs are going to be doing Fourier

1 transforms, which means a lot of matrix math, and that's
2 going to be the dominant computational task, not
3 bandwidth.

4 Q. (By Mr. Limbacher) What if you were performing
5 real-time image recognition?

6 MR. CLAWSON: Objection. Form.

7 Q. (By Mr. Limbacher) Would bandwidth limitations
8 affect how you would assign sub-tasks in that situation?

9 MR. CLAWSON: Same objection.

10 THE WITNESS: The bandwidth here is going to
11 be -- I mean, a megapixel image is going to take a
12 fraction of a second. No, I don't see that as the
13 issue. I think it's going to be the computational load.
14 That's why we use a DSP.

15 Q. (By Mr. Limbacher) So the Budenske solution
16 you're saying is good enough as long as there's not a
17 lot of data?

18 A. That's not what I said. What we're talking
19 about Budenske is there's a solution and then there is a
20 specific instance of the hardware that was available to
21 them in 1998, which was 64 processors, it says, four
22 different types of processors, so forth and so on.

23 They were able to do what they wanted to do in
24 1998. It's the software's architecture that we're
25 talking about, and software architecture is reassigning

1 tasks to booster nodes.

2 We don't have data on the bandwidth or the
3 latency between the PowerPC and the VSP. But my iPhone
4 is a lot faster than all the computers in the United
5 States when I was in college, so it's kind of a
6 non-sequitur here.

7 Q. So is your -- circling back to the proposed
8 combination between Lippert and Budenske, is it your
9 position that bandwidth would not be a consideration in
10 implementing that?

11 MR. CLAWSON: Objection. Form.

12 THE WITNESS: I don't understand how bandwidth
13 plays into deciding whether I assign something to a
14 booster node or not unless bandwidth is -- there is an
15 equation that we could write down here. And deciding
16 not to assign it to a booster node would have to have
17 bandwidth so severely limited that it's faster to do the
18 task on the compute node. And it beggars the
19 imagination that one would build a computer system like
20 that.

21 Q. (By Mr. Limbacher) Is there overhead involved
22 in moving a sub-task, say, from a computation node to a
23 booster node?

24 MR. CLAWSON: Objection. Form.

25 THE WITNESS: Not in the sense that maybe you

1 are proposing it. The code is probably going to be
2 loaded onto the booster node before the data to be
3 processed.

4 So the data to be processed is going to be
5 brought in, and it could be given over the VMEbus, by
6 the way. It could be given to the computational node or
7 the booster node, so the rate at which the data is
8 delivered to either node should be the same.

9 The loading of the code can be done before the
10 task even starts, so there's no overhead in that sense
11 there.

12 Q. (By Mr. Limbacher) Look back at, this is
13 Budenske, Deposition Exhibit D, Exhibit 1005 in the IPR,
14 Page 392.

15 A. Are you asking a question?

16 Q. I'm just waiting for you to get to the page.

17 A. That's where I am.

18 Q. Do you see where it says, The six SHARC
19 processors physically share a DRAM"?

20 A. Um-hum.

21 Q. What does that mean?

22 A. That means they share DRAM. DRAM is the
23 dynamic memory. That means the SHARC processors are
24 going to be able to talk to each other very fast.

25 Q. Is that where the data being processed would be

1 stored for sub-tasks that were running on the SHARC
2 processors?

3 A. That's where part of it's going to be stored.
4 Before it loaded into DRAM, it's going to be in a hard
5 drive or brought in from some video system. The SHARC
6 processors are going to have their own memories as well,
7 but they share DRAM, so communication between SHARC
8 processors is very fast.

9 Q. If a sub-task was going to be moved from the
10 SHARC processor to the RISC processor, would the data
11 being processed by that sub-task also need to be moved?

12 MR. CLAWSON: Objection. Form.

13 THE WITNESS: I would need to see a diagram
14 of this to know whether the DRAM is accessible over the
15 VMEbus or not.

16 Q. (By Mr. Limbacher) Does Budenske say anywhere
17 that the DRAM -- I'm sorry -- that the DRAM is
18 accessible over the VMEbus?

19 A. I don't have a -- I don't have that information
20 available to me.

21 Q. So in doing your analysis, you didn't consider
22 any of these things specifically; correct?

23 MR. CLAWSON: Objection. Form.

24 THE WITNESS: The questions you're asking of
25 the specific architecture of Budenske?

1 Q. (By Mr. Limbacher) Correct.

2 A. Not the ones which I consider irrelevant to the
3 question, which is the mapping of tasks and the
4 iterative processing. I did not consider the latency
5 across a VMEbus.

6 Q. So do you consider transfer of data to be
7 relevant to mapping of tasks?

8 MR. CLAWSON: Objection. Form.

9 THE WITNESS: The ability to transfer data is
10 going to be important. The nature of the task is going
11 to define the requirement, how rapidly you can transfer
12 data.

13 Q. (By Mr. Limbacher) So handling certain tasks
14 could be more complicated than others?

15 MR. CLAWSON: Objection. Form.

16 THE WITNESS: Certainly some tasks are more
17 complicated than others. Yeah. That's a truism.

18 Q. (By Mr. Limbacher) You have written software
19 code in the past?

20 A. A lot of it.

21 Q. Do you still write software code today?

22 A. When is the last code I wrote? Last week.

23 Q. Are all software code implementations obvious?

24 MR. CLAWSON: Objection. Form. Objection.

25 Scope.

1 THE WITNESS: Are all software code
2 implementations obvious? If the AI utopians are to be
3 believed, of course the answer is yes, but I don't
4 believe them.

5 Some software is very well known. There are
6 very well known patterns and one would consider doing
7 that obvious. Write a quick sort. Obvious. Do a
8 binary tree. Obvious. Implement a novel algorithm that
9 no one's done before or a difficult algorithm, it's not
10 going to be obvious.

11 Q. (By Mr. Limbacher) Have you ever had to go
12 back and redesign a project after you had started
13 writing the code because the original design was not
14 workable?

15 MR. CLAWSON: Objection. Form. Objection.
16 Scope.

17 THE WITNESS: Let me think back over 45 years.
18 No, no. I always planned it out pretty carefully before
19 I got started. I found bugs and fixed them, but the
20 design -- if you design things properly by thinking them
21 through, you don't have to go back and completely
22 rearchitect.

23 Q. (By Mr. Limbacher) Are there any software
24 products available on the market today that you've
25 written?

1 MR. CLAWSON: Objection. Form. Objection.
2 Relevance.

3 THE WITNESS: I think you can still buy IBM
4 total storage/SAN.

5 (Stenographer clarification.)

6 THE WITNESS: Storage/SAN. You can -- it's in
7 the Linux kernel, it is the Ceph file system, which
8 was -- a major part of that was my design. It's
9 available on every Linux computer in the world. Let's
10 see. There is -- I don't know if you can still buy it
11 from IBM. It was called Tivoli Storage Manager.
12 There's a good chunk of my code in there.

13 Q. (By Mr. Limbacher) Let's start with IBM's
14 total storage/SAN.

15 Did you write that entire application?

16 A. No. It was written by a team; me, Robert
17 Reese, Wayne Hineman, Randal Burns. Bob and Randall and
18 I were the primary architects. We also had Brian. Oh,
19 I can't remember his name. And I think Ralph worked on
20 it for a while before IBM handed development off to a
21 team in Portland. Several million lines of code.

22 Q. Did you draft design documents before code
23 development began?

24 A. Absolutely.

25 Q. Did the final implementation look exactly like

1 those design documents?

2 A. It looked pretty darn close.

3 Q. What's "pretty darn close"?

4 A. Well, I can't tell you what IBM released in the
5 end because I said they handed it off to Portland. I
6 can tell you that our original design had something
7 called opportunistic leases in it, and we had those
8 working, and it was very fast, and it went to the group
9 in Portland and they didn't understand opportunistic
10 leases, replaced it with 30-second timeout and suddenly
11 it went slow and they complained. And so the code was
12 sent back to us and we reimplemented opportunistic
13 leases and it went fast again.

14 Opportunistic leases are actually described in
15 one of the papers on my CV. I think maybe even one of
16 the patents.

17 Q. Have you ever written any task scheduling code?

18 MR. CLAWSON: Objection. Form.

19 THE WITNESS: Task scheduling code? You mean
20 low-level or high-level code?

21 Q. (By Mr. Limbacher) Either. Let's start with
22 low-level.

23 A. Low-level code, I've done it all the time,
24 especially when I taught the operating systems, of
25 course, for 30 years to modify the scheduler.

1 Q. Are you familiar with the Linux scheduler?

2 A. Which version?

3 Q. Let's start with 2000.

4 A. I would have to go back and look. Linux likes
5 to redo its scheduler frequently.

6 Q. So why do they like to redo their scheduler
7 frequently?

8 A. It's the Linux people. The original scheduler
9 for Linux was very simple, and it's an open-source
10 project, and people get new ideas for doing it. And
11 whoever owns the scheduler gets to implement their
12 ideas.

13 But Linux scheduler's gone through several
14 significant -- significant updates. So asking me what
15 it looked like 25 years ago, I can't tell you. I'd have
16 to go and look.

17 Q. Do you recall an update where it first moved
18 to -- we'll call this Big O of one efficiency?

19 MR. CLAWSON: Objection. Form.

20 THE WITNESS: There's -- that's somewhat of a
21 misstatement. Which thing is Big O of one that you're
22 attributing that efficiency to the selection of the
23 task?

24 Q. (By Mr. Limbacher) Yeah. Selection of the
25 task.

1 A. Selection of the task. Big O of one efficiency
2 was, in fact, the first version. Because it just pulled
3 the thing off the front of the queue. Then there's been
4 multiple revisions.

5 So when you talk about the efficiency of
6 choosing which thing to run, Big O of one means constant
7 time. And if you always pick the first thing from the
8 first queue, that's going to be Big O of one. If you
9 have to do any sorting of the queue, then it's going to
10 add some complexity to that. Not sure why this is
11 relevant, but it's fun. Let's go ahead.

12 Q. I'm just getting -- is it your view that all of
13 the Linux scheduling implementations would have been
14 obvious?

15 A. All the Linux scheduler?

16 MR. CLAWSON: Objection. Form.

17 Q. (By Mr. Limbacher) Yes.

18 A. I would have to see them all. Most of them
19 that I've seen rely heavily on existing scheduling
20 theory largely developed in the 1970s.

21 Q. All right.

22 So let's go back to you said you had written
23 some high-level task scheduling code; is that correct?

24 A. Yes.

25 Q. Can you give me an example of that.

1 A. It was a long time ago. This high-level task
2 scheduling for discreet event simulators. I did this in
3 1986 through '89. I reimplemented that again in '91 for
4 another project. '93 for an -- did make replication
5 scheme. It's probably been 30 years since I've had to
6 write any high-level --

7 (Stenographer clarification.)

8 THE WITNESS: -- task scheduling code. Last
9 few times I've done it, I've just reused my code from
10 the previous time.

11 Q. (By Mr. Limbacher) I think I'm about to get to
12 a new topic. How about we go off the record.

13 THE STENOGRAPHER: We're off the record. The
14 time is 11:54 a.m.

15 (Lunch recess taken 11:54 a.m. to 12:30 p.m.)

16 THE STENOGRAPHER: We're back on the record.
17 The time is 12:30 p.m.

18 Q. (By Mr. Limbacher) Dr. Long, during the break,
19 did you speak to anyone about this deposition or this
20 proceeding?

21 A. No, sir, I did not.

22 Q. Okay.

23 Please turn to Paragraph 65 of your
24 declaration.

25 A. Okay.

1 Q. Do you see about halfway down, it says, "Other
2 prior" --

3 (Stenographer clarification.)

4 Q. (By Mr. Limbacher) "Other prior art references
5 available to the POSITA disclosed successful
6 implementations of Budenske's teachings"?

7 A. Yes.

8 Q. Okay.

9 And then you cite Exhibit 1007 and 1008; is
10 that correct?

11 A. 1007 and 1008, that's right.

12 Q. Okay.

13 I'm going to introduce an article titled, "A
14 Semi-Static Approach to Mapping Dynamic Iterative Tasks
15 Onto Heterogeneous Computing Systems" as Exhibit G to
16 this deposition. This is Exhibit 1007 in the IPR.

17 (Exhibit G was marked for identification.)

18 Q. (By Mr. Limbacher) Can you please turn to that
19 exhibit.

20 A. I see it.

21 Q. And then turn to Page 81.

22 A. Okay.

23 Q. And read the first two paragraphs of Section 3,
24 the section titled, "System Model."

25 A. How many paragraphs?

1 Q. The first two paragraphs of Section 3.

2 A. Okay.

3 Q. All right.

4 At the bottom of Page 81, what does it mean for
5 communications among processors of the same type to be
6 symmetric?

7 A. It should mean that sending -- how do I say
8 this easily?

9 If A sends to B and it takes a --

10 (Stenographer clarification.)

11 THE WITNESS: If A sends to B and it takes a
12 certain amount of time, then it should -- B sending to A
13 should take the same amount of time. Latency should be
14 the same. Bandwidth should be the same.

15 Q. And what is conflict-free time?

16 A. That means that presumably that there is no
17 queuing delays, probably because they have a crossbar --

18 (Stenographer clarification.)

19 THE WITNESS: -- crossbar switch.

20 Q. (By Mr. Limbacher) Do you see the last
21 sentence at the bottom of Page 81, it refers to the
22 symmetry property of the interprocessor communications
23 among processors of the same type?

24 A. Yes.

25 Q. Why is it emphasizing that they are processors

1 of the same type?

2 MR. CLAWSON: Objection. Form.

3 THE WITNESS: No idea why they decided to do
4 that. Read it a little bit more.

5 Q. (By Mr. Limbacher) Okay.

6 Just let me know when you're done.

7 A. Apparently they just did it for simplicity.

8 Q. So why would doing that increase simplicity?

9 MR. CLAWSON: Objection. Form.

10 THE WITNESS: Computer science, we always make
11 simplifying assumptions. If you're doing modeling,
12 apparently they're doing modeling, the more parameters
13 to the model, the more complicated it is.

14 So assuming symmetry, that's somehow a much
15 simpler equation.

16 Q. (By Mr. Limbacher) So simplifying things makes
17 them less difficult?

18 A. That's not what I said.

19 Q. Well, would you agree with that statement?

20 MR. CLAWSON: Objection. Form.

21 THE WITNESS: No, I would not.

22 Q. (By Mr. Limbacher) So you disagree that
23 simplifying things makes them less difficult?

24 MR. CLAWSON: Objection. Form.

25 THE WITNESS: Simplifying things does not

1 always make them less difficult. Sometimes it makes
2 them wrong. So what I'm saying here is that they are
3 talking about their modeling something, they have some
4 equations, and by assuming symmetry, the equations
5 become simpler.

6 Q. (By Mr. Limbacher) Can you go to the bottom of
7 Page 83.

8 A. Sure.

9 Q. On the left column, there's a formula labeled
10 as Formula 1?

11 A. Yes.

12 Q. What is formula?

13 A. No idea. I have to read it.

14 Q. Okay.

15 Please do so.

16 A. Okay. Okay. At a high level, it is a model of
17 the execution time of some sub-task $S(i)$ on a virtual
18 machine. Virtual machine is identified by you. There
19 is a heterogeneous -- heterogeneity factor indicating
20 the relative speed of the sub-task on this type of
21 processor used in the virtual machine.

22 There is a fraction of the time that is
23 parallel and fraction that is serial and then there is a
24 parallelization overhead represented by $BI \beta \log P$.
25 It's not clear to me yet why we're using a algorithm

1 here.

2 Q. There's only a single heterogeneity factor in
3 the equation; is that correct?

4 A. As called out as a heterogeneity factor, but
5 there's a parallel and there's a serial fraction, which
6 is also going to be part of that calculation. You can't
7 just pull that -- that thing, that heterogeneity factor
8 distributes over all of the terms in the rest of the
9 equation, so it affects every one of those terms.

10 Q. Okay.

11 So let's talk about the rest of those terms.

12 So $A(i)$, would you agree that is the
13 coefficient for the parallel portion of sub-task (i) 's
14 execution time?

15 A. Let me look.

16 MR. CLAWSON: Objection. Form.

17 THE WITNESS: Number of process and three
18 coefficients, $A(i)$, $B(i)$, $C(i)$, the parallel fraction is
19 $A(i)$ alpha divided by P , where P is the number of
20 processors.

21 And you wanted me to agree to $A(i)$ being what?

22 Q. So I'm referencing the definition in Table 1.

23 A. Thank you.

24 So coefficient with the parallel portion of
25 sub-task (i) 's execution time.

1 Q. Would you expect that coefficient to be
2 different for different types of processors?

3 MR. CLAWSON: Objection. Form.

4 THE WITNESS: I think so. It will be
5 different. Based on -- thus, in general, different
6 sub-tasks will have affinities for different machines.
7 These are -- what you're asking me about is a
8 dimensionalist quantity. It's just a coefficient, so --

9 Q. (By Mr. Limbacher) The heterogeneity factor,
10 would you expect that to be different for different
11 processor types?

12 MR. CLAWSON: Objection. Form. Objection.
13 Scope.

14 THE WITNESS: You're asking me to speculate on
15 somebody else's model. Yeah, for different types of
16 machines, yeah, they talk about things like clock rate,
17 number and levels of caches and cache sizes. So that
18 should be different, based on the different kinds of
19 machine you have.

20 Q. (By Mr. Limbacher) So what about the parameter
21 B(i)? Would you expect that to be different for
22 different types of processors?

23 MR. CLAWSON: Objection. Form. Objection.
24 Scope.

25 THE WITNESS: For different types of

1 processors, you said?

2 Q. (By Mr. Limbacher) Yes, sir.

3 A. That's a coefficient, so it's dimensionalist.
4 It means it's going to depend on the different type of
5 processor and the type of sub-task that you're
6 executing, so I would expect it to be different.

7 Q. Okay.

8 What about for $C(i)$? Would you expect that to
9 be different for different types of processors?

10 MR. CLAWSON: Same objections.

11 THE WITNESS: Probably going to be different
12 because serial portions can be handled by different
13 processors, maybe have faster energy units or faster
14 floating point units or whatever. But, again, it's
15 dimensionalist quantities, so it's hard to say.

16 Q. (By Mr. Limbacher) Okay.

17 Please turn to Page 80 and read the first full
18 paragraph.

19 A. Okay.

20 Q. What is a fundamental research issue?

21 MR. CLAWSON: Objection. Form.

22 Q. (By Mr. Limbacher) This is recited in the
23 first sentence of that paragraph.

24 MR. CLAWSON: Same objection.

25 THE WITNESS: It's a term of art, I suppose.

1 They are saying that they are addressing an
2 interesting research question of how to select
3 representative mappings offline for online use.

4 Q. (By Mr. Limbacher) Okay.

5 And what makes it fundamental?

6 MR. CLAWSON: Objection. Form.

7 THE WITNESS: It's -- let me think without
8 circular reasoning. It's the goal of this research to
9 find offline mappings for online use, so it's
10 fundamental to find a way to do that, and the way
11 they've chosen is using these genetic algorithms.

12 Q. (By Mr. Limbacher) Have you ever performed any
13 research that you would consider to be fundamental
14 research?

15 MR. CLAWSON: Objection. Form.

16 THE WITNESS: I would say most of my research
17 is considered fundamental research. And according to
18 the NSF definition, it's also basic research.

19 Q. (By Mr. Limbacher) What's the NSF definition?

20 A. It's -- the short version is curiosity-driven
21 research for unsolved problems. If you already have a
22 solution, then it's not really considered fundamental.

23 Now, you can find a better solution to an
24 existing problem, but you really need to find something
25 that works better according to some metric.

1 Q. Does that research typically entail
2 experimentation?

3 MR. CLAWSON: Objection.

4 THE WITNESS: Doesn't have to. There's -- we
5 call them theoreticians. Theoreticians prove their
6 theorems and their theorems show that this behavior or
7 that behavior is better than anything before. And
8 they -- theoreticians are the ones that actually, I
9 guess, answer the most fundamental of the fundamental
10 questions, and then experimentalists determine whether
11 the theoreticians' theories are practical or not.

12 Q. (By Mr. Limbacher) So this article, Exhibit
13 1007 in the IPR, do you consider it as being -- as being
14 written by experimentalists or theoreticians?

15 MR. CLAWSON: Objection. Form.

16 THE WITNESS: This appears to be
17 experimentalists. I have actual numbers in it. It
18 looks like they're trying to fit a model to experimental
19 results. Section 5, the performance section.

20 Q. (By Mr. Limbacher) So would you understand
21 from this that experimentation was likely needed for
22 them to reach their conclusions in this paper and draft
23 the paper?

24 MR. CLAWSON: Objection. Form.

25 THE WITNESS: For the paper in this form with

1 experimental results, they needed to run experiments,
2 yes.

3 Q. (By Mr. Limbacher) Do you think the outcome of
4 those experiments was predictable before they started
5 them?

6 MR. CLAWSON: Objection. Form.

7 THE. WITNESS: I think the shape of the
8 results was. I doubt that it was predictable to some
9 number of decimal places because there's a lot of
10 overheads and other little factors that go into
11 implementation details that are not predicted by theory.

12 As we discussed earlier with the Big O, the
13 constant factors are generally not known, and those are
14 what are exposed through experimentation.

15 Q. (By Mr. Limbacher) I'm going to introduce an
16 article titled, "Mapping Tasks onto Distributed
17 Heterogeneous Computing Systems Using a Genetic
18 Algorithm Approach," as Exhibit H to this deposition.
19 This is Exhibit 1008 in the IPR.

20 (Exhibit H was marked for identification.)

21 THE WITNESS: Okay.

22 Q. (By Mr. Limbacher) So when you have that up,
23 please turn to Page 156.

24 A. Okay.

25 Q. Please read the first paragraph of Section

1 6.5.2 to yourself.

2 A. Okay.

3 Q. Do you see where it says each of these
4 sub-tasks may allow use of multiple processors of the
5 same type for efficient execution?

6 A. Yes.

7 Q. Why did they limit it to processors of the same
8 type?

9 MR. CLAWSON: Objection. Form.

10 THE WITNESS: Because they're doing the same
11 thing.

12 Q. (By Mr. Limbacher) Okay.

13 Is there anything in this paper that you're
14 aware of that discusses efficient processing of
15 sub-tasks using multiple processors of different types?

16 MR. CLAWSON: Objection. Form.

17 THE WITNESS: I think you're trying to say
18 something that's not what they're trying to say.

19 They're saying that if I have a sub-task, all
20 the things in the sub-task are doing the same thing, in
21 which case, using the same type of processor makes
22 sense.

23 If you're asking if there's anything in the
24 paper that teaches away from that, where doing the same
25 thing should be done with multiple types of processors,

1 I'm unaware of it. But I would have to read the paper
2 again.

3 Q. (By Mr. Limbacher) Just to confirm, you did
4 review this paper before submitting your declaration;
5 correct?

6 A. Almost a year ago.

7 Q. Did you review this paper in preparation for
8 the deposition today?

9 A. This one, no.

10 Q. So we're going to go back to your declaration.
11 This is Exhibit B to the deposition. This is Exhibit
12 1003 in the IPR, and please turn to Paragraph 90.

13 Let me know when you're there.

14 A. I'm reading it now.

15 Q. Okay.

16 A. Okay.

17 Question?

18 Q. Yeah. So I'm looking at Page 54, and the first
19 full sentence, you state, "In order to send the dynamic
20 parameter values taught by Budenske, as well as
21 information regarding computing capacities, as taught by
22 Lippert, from each computation node and each booster
23 node to a central manager to determine a remapping of
24 sub-tasks, a POSITA thus would have implemented
25 Lippert-Budenske with daemons at each computation node

1 and each booster node configured to collect and send
2 this information."

3 Did I read that correctly?

4 A. Yes.

5 Q. Would it be possible to implement the same
6 thing without using daemons?

7 A. Sure.

8 MR. CLAWSON: Objection. Form.

9 Q. (By Mr. Limbacher) Can you give an example of
10 a way to do so without using daemons?

11 MR. CLAWSON: Objection. Form.

12 THE WITNESS: Without using daemons. Which
13 part would you like to do first?

14 Q. (By Mr. Limbacher) I'll leave that up to you.

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

18 You don't need a daemon to collect and collate
19 the information; you could have that done by, say, the
20 application manager itself or the resource manager
21 itself. Daemons are a nice way to structure code.

22 Q. Can you please turn to Paragraph 77 of your
23 declaration and read it to yourself.

24 A. Okay.

25 Q. Are you asserting that the dynamic parameters

1 of Budenske are the information relating to the
2 processing of the sub-tasks in the claim?

3 MR. CLAWSON: Objection. Form.

4 THE WITNESS: Which -- I need you to say that
5 again. That was?

6 Q. (By Mr. Limbacher) So I'm looking at the last
7 sentence in Paragraph 77.

8 A. "Budenske thus," that one?

9 Q. Correct.

10 And my question is: Are you mapping the
11 dynamic parameters of Budenske to the information
12 relating to the processing of sub-tasks recited in the
13 claim?

14 A. Yes.

15 Q. What are the dynamic parameters of Budenske?

16 A. Let's go back to Budenske.

17 Q. For the record, Budenske is Exhibit D to this
18 deposition, and Exhibit 1005 in the IPR.

19 A. Some of the dynamic parameters, from the
20 bottom, Page 393, may give it just as a hypothetical
21 example, number of vehicles to be tracked and so forth.

22 Q. Okay.

23 So I'm going to read off what I think you're
24 referring to and you can tell me if I'm correct.

25 A. Okay.

1 Q. I'm at the last full photograph of Page 393,
2 and it states, "The application developer has chosen
3 four dynamic parameters. One, number of objects of
4 interest being tracked in the last image frame; two,
5 average size of the objects currently being tracked in
6 terms of percent of the total number of pixels in the
7 image; three, whether the lighting conditions in the
8 last image frame required requantization for better
9 contrast for display; and four, whether there was cloud
10 cover detected in the last image frame that required
11 special processing."

12 A. Right.

13 Q. Those are the dynamic parameters you're
14 referring to; is that correct?

15 A. That's one of -- that's the example they give
16 in Budenske, yes.

17 Q. Okay.

18 So these dynamic parameters appear to refer to
19 the data being processed; is that correct?

20 A. Right. And that's going to affect the
21 performance of the booster node and the computation node
22 and may require dynamic remapping.

23 Q. Okay.

24 Are there any other dynamic parameters that
25 you're relying on from Budenske?

1 MR. CLAWSON: Objection. Form.

2 THE WITNESS: Are you asking whether there are
3 other types of dynamic parameters or whether the dynamic
4 parameters are handled in a different way or by a
5 different entity?

6 Q. (By Mr. Limbacher) So to -- to reiterate,
7 you're relying on the dynamic parameters as the
8 information relating to the processing of the sub-tasks
9 in your analysis; is that correct?

10 A. Correct.

11 Q. And I'm just trying to make sure I understand
12 what specific dynamic parameters you're saying are that
13 information relating to the -- relating to the
14 processing of the sub-tasks.

15 A. So in Budenske, there was an example given, but
16 it was very clear there was an example and that there
17 could be other parameters. For example, they gave cloud
18 cover. Cloud cover is lighting, number of vehicles, and
19 so forth.

20 Q. Okay.

21 A. It later says that the HC monitor monitors the
22 runtime values, the dynamic parameters at the end of
23 each iteration to make a heuristically-based decision
24 whether to continue with the current mapping or to
25 select and instantiate a new mapping.

1 So you make a guess as to what reasonable
2 parameters are; that instantiates a DAG. The HC monitor
3 monitors those and sees if those parameters fit. If
4 those parameters didn't fit, is there something else
5 that fits better and instantiates a new DAG. More
6 vehicles arrive, the cloud cover disappears, whatever
7 the parameters happen to be.

8 Q. Okay.

9 A. I don't see anything that says those are the
10 only parameters. I saw those examples saying those are
11 the parameters.

12 Q. Sitting here today, are there any other dynamic
13 parameters that you can think of that you are relying
14 on?

15 A. That I'm relying on?

16 MR. CLAWSON: Objection. Form.

17 (Stenographer clarification.)

18 THE WITNESS: I'm not relying on specific
19 values of dynamic parameters; I'm relying on the
20 existence of dynamic parameters.

21 Q. (By Mr. Limbacher) So looking at this example
22 on the bottom of Page 393, I believe they're referring
23 to an automatic task recognition.

24 What sort of task are they performing there?

25 A. Target -- automatic target recognition.

1 Q. Okay.

2 A. This is the Army. I think they're looking for
3 tanks or something.

4 Q. Okay.

5 So what is automatic target recognition?

6 A. It means to capture an image and look for
7 targets. In the modern context, we might be looking for
8 drones, for example, and tracking -- and tracking the
9 drones. Something like you would use on a Patriot
10 Missile System would do this.

11 Q. So the -- the dynamic parameters that are
12 referred to at the bottom of Page 393, do you know how
13 those are being generated in Budenske?

14 MR. CLAWSON: Objection. Form.

15 THE WITNESS: I'm guessing that Budenske and
16 Jay and the other authors said, These seem like
17 reasonable parameters for detecting tanks. And like I
18 said, they gave them as examples. There may be others.

19 Q. (By Mr. Limbacher) So let's look at the first
20 dynamic parameter, number of objects of interest being
21 tracked in the last image frame.

22 A. Yep.

23 Q. Can you walk me through how that value would be
24 generated.

25 MR. CLAWSON: Objection. Form.

1 THE WITNESS: How the value would be generated?

2 Q. (By Mr. Limbacher) Yes, sir. I would assume
3 that dynamic parameter value would be generated in some
4 form; is that correct?

5 MR. CLAWSON: Objection. Form.

6 THE WITNESS: I'm -- I'm having trouble with
7 the word "generated." You mean "computed."

8 Q. (By Mr. Limbacher) Sure. Let's say computed.

9 A. It's not going to be computed. It's going to
10 be -- well, the image will be computed, but when the
11 DDGs -- DDGs or whatever they're called -- are created,
12 there's going to be one tank, two tank, three tank, four
13 tank, some number of tanks. This cloud cover, that
14 cloud cover, so forth and so on.

15 All these parameters are going to go into the
16 genetic algorithm and it's going to create a bunch of
17 these DDGs. These are mappings of the ATR task onto the
18 processors.

19 When the -- an iteration of the task happens,
20 the HC monitor calculates whether these parameters were
21 accurate. You start off maybe there's one tank, and the
22 HC monitor says, Well, we detected two tanks.

23 Then the next time, if it's advantageous, then
24 they're going to load a different DDG so they can detect
25 more tanks because, well, maybe there's a bunch of tanks

1 arriving for a tank battle.

2 But instead of starting off with all the tanks,
3 they start off with maybe one or two tanks, try it with
4 that. If there's more, then we need to add more
5 computational resources to detect more tanks.

6 Q. (By Mr. Limbacher) So would there be a
7 sub-task responsible for counting the number of tanks in
8 the image frame?

9 MR. CLAWSON: Objection. Form.

10 THE WITNESS: That could be one of the possible
11 things that the SHARC processor does.

12 Q. (By Mr. Limbacher) Okay.

13 Is it your opinion -- sorry. Go ahead.

14 A. I'm -- it's unclear. The algorithm's not laid
15 out who's doing the actual counting.

16 Q. Okay.

17 So of the tasks -- I'm sorry.

18 Of the dynamic parameters numbered one, two,
19 three, and four, do you know which processor would be
20 more likely to be used to identify --

21 MR. CLAWSON: Objection. Form.

22 THE WITNESS: Let's see.

23 (Stenographer clarification.)

24 Q. (By Mr. Limbacher) I believe I said that
25 information, but let me ask it again.

1 I'm going ask it, and then we're all going to
2 pause for Cameron to object to it again, and then we'll
3 get you, Dr. Long.

4 Of the dynamic parameters numbered one, two,
5 three, and four, do you know which type of processor
6 would be -- more likely be used to calculate those
7 values?

8 MR. CLAWSON: Objection. Form.

9 THE WITNESS: Number one, counting the number
10 of objects, I would -- you're asking me if I were
11 writing the code.

12 If I were writing the code, then I would use
13 the computational node. For number two, the average
14 size of the objects being tracked, I would probably use
15 the SHARC. The lighting conditions are going to be
16 something that's calculated, probably the SHARC. Cloud
17 cover, that may be a function of lighting. Again, that
18 may be the SHARC.

19 So really depends on how these things are done.
20 A lot of these are calculated values that are then going
21 to be matched by the computational node to see how well
22 they fit.

23 Q. (By Mr. Limbacher) So given the example here
24 in Budenske of these dynamic parameters, one, two,
25 three, and four, can you think of a change in value that

1 would likely result in moving a sub-task from one
2 processor type to the other processor type?

3 MR. CLAWSON: Objection. Form.

4 THE WITNESS: Not for this task.

5 Q. (By Mr. Limbacher) Okay.

6 Please go back to your declaration. This is
7 Deposition Exhibit B. It's Exhibit 1003 in the IPR.

8 A. Okay.

9 Q. And then go to Paragraph 78, and then read
10 until you see the word "parallelism."

11 It would be on Page 43, about seven lines down.

12 A. I do not see the word "parallelism."

13 Q. Are you on Page 43?

14 A. Yeah. Okay. It was in parenthesis.

15 Q. Correct.

16 A. "At each iteration, there can be different
17 mapping of the DDG (parallelism) onto the accelerators."
18 Yep.

19 Q. So why do you have the word parallelism in
20 parenthesis after the DDG?

21 A. As I explained earlier, when we have a DAG, it
22 represents partial ordering. And the -- how do I say
23 this?

24 When you look at a partial ordering, the number
25 of choices that you have at each step in the partial

1 ordering that are in order with respect to each other
2 represent the degree of parallelism in the graph.

3 THE WITNESS: Can we take a quick break? I
4 want to go to the restroom.

5 MR. LIMBACHER: Yes. Actually, this is a good
6 stopping point, so let's go off the record.

7 THE STENOGRAPHER: We're off the record. The
8 time is 1:20 p.m.

9 (Recess taken from 1:20 p.m. to 1:28 p.m.)

10 THE STENOGRAPHER: We're back on the record.
11 The time is 1:28 p.m.

12 Q. (By Mr. Limbacher) All right.

13 Dr. Long, did you speak with anyone about your
14 deposition or this proceeding during the break?

15 A. I did not. We talked about bees.

16 Q. Oh, okay.

17 A. Somebody put a honey hive outside the building
18 and there were no bees.

19 Q. Please turn to your declaration. This is
20 Deposition Exhibit B and Exhibit 1003 in the IPR.

21 A. Okay.

22 Q. Turn to Paragraph 88.

23 A. Okay.

24 Q. So the first sentence of this, you state that,
25 "The resource manager dynamically alters further

1 distribution, as recited in this claim element
2 encompasses the resource manager dynamically alters
3 assignment of plurality of computation nodes and the
4 plurality of booster nodes to each other"; is that
5 correct?

6 A. Correct.

7 Q. What does that mean?

8 A. That's not how I would write it, so -- how do I
9 say this and not sound like a computer scientist saying,
10 It does the thing.

11 It's included in, I suppose.

12 Q. Are you asserting -- well, strike that.

13 What are the differences between what the
14 resource manager is doing in Claim 3 and what it's doing
15 in Claim 4?

16 MR. CLAWSON: Objection. Form.

17 THE WITNESS: Yeah, it's Exhibit A?

18 Q. (By Mr. Limbacher) So the '442 patent would be
19 Exhibit A, which is Exhibit 1001 in the IPR.

20 A. Okay. Got it. Thanks.

21 Well, in 3, the application manager receives
22 the information, processes it, and gives it to the
23 resource manager. In 4, it looks like the resource
24 manager gets the information.

25 Q. So how does that mean, though, that one

1 encompasses the other?

2 MR. CLAWSON: Objection. Form.

3 THE WITNESS: Go back to that one.

4 For me, what I'm trying to say here is
5 basically they're doing the same thing. There's really
6 not any -- can you still hear me?

7 Q. (By Mr. Limbacher) I can hear you.

8 A. Okay. Yeah. This thing is making a bonk
9 noise.

10 But there's really not a substantial difference
11 here. In one case, it goes to the application manager
12 and then to the resource manager, and the other one, it
13 goes to the resource manager. It really doesn't make
14 any difference. One is pretty much the same as the
15 other.

16 Q. So can you pull up the '442 patent for me.
17 Again, this is Exhibit A.

18 A. Got it.

19 Q. I'm looking in Column 6. The last line of
20 Claim 3 says, "The resource manager" -- I'm sorry. Let
21 me just start again.

22 In Column 6, Lines 3 to 5, it says, "The
23 resource manager dynamically alters further distribution
24 during the computation task."

25 A. Um-hum.

1 Q. And then Claim 4, Lines, looks like, about 7 to
2 11, says, "The resource manager dynamically alters
3 assignment of the plurality of computation nodes and a
4 plurality of booster nodes to each other during the
5 computation of the computation task."

6 Are you saying the limitation I read from Claim
7 3 is essentially the same as the limitation I read from
8 Claim 4?

9 A. Pretty much.

10 MR. CLAWSON: Objection. Form.

11 THE WITNESS: The distribution -- the
12 distribution -- those seem like different ways of saying
13 pretty much the same thing.

14 Q. (By Mr. Limbacher) Okay.

15 Going back to your declaration, your
16 declaration is Deposition Exhibit B, IPR Exhibit 1003,
17 please turn to Paragraph 92 and read the first sentence.

18 A. Out loud?

19 Q. You can just read it to yourself.

20 A. Okay.

21 Okay.

22 Q. Is it your assertion that a rating is -- or I'm
23 sorry -- that a rating -- let me back up.

24 What is the relationship between a rating and
25 the priority information, complexity information, and

1 scalability information that you mention here?

2 MR. CLAWSON: Objection. Form.

3 THE WITNESS: I cited to Paragraph 20 of
4 Lippert, "Specific computation tasks may comprise
5 priority information, which indicates how urgently the
6 specific computation task has to be completed." That's
7 an example.

8 Go back and look at Lippert for rating. Do you
9 want to continue with priority or do you want to go on
10 to the rest of the pragmas?

11 Q. (By Mr. Limbacher) My main question for right
12 now, to be a rating, does something need to have all
13 three of those, priority, complexity, and scalability
14 information?

15 A. No. You could do -- you could have any number
16 of things that you wanted to use as a rating for
17 deciding whether or not to map this onto a booster.

18 Q. Is there anything else other than those three
19 types of information that you're aware of that would be
20 a rating?

21 MR. CLAWSON: Objection. Form.

22 THE WITNESS: I can imagine any number of
23 things. Are you asking for things that are disclosed in
24 Lippert or are you asking in general?

25 Q. (By Mr. Limbacher) I'm just -- can you give me

1 one example beyond those three that you would consider
2 to be a rating?

3 A. Vectorizeability.

4 Q. Okay.

5 Where does Lippert state that the information
6 you're relying on is provided in the source code?

7 MR. CLAWSON: Objection. Form.

8 THE WITNESS: I got to go back here. We did
9 this one before.

10 The '442 patent does, and that is Line 49 of
11 Column 1, Lippert, old Lippert. Sorry. Lots of things
12 open here.

13 Paragraph 84.

14 Q. (By Mr. Limbacher) And you're referring to
15 Deposition Exhibit C, Exhibit 1004 Lippert; is that
16 correct?

17 A. That's right.

18 Q. All right.

19 So can you read off the line you're relying on.

20 A. "In the following, the distinction between
21 highly scaleable and complex is made on the level of
22 code blocks may introduce the notion of Exascale Code
23 Blocks and Complex Code Blocks, CCB. Obviously there is
24 no purely highly scaleable code and there is no strictly
25 complex code as well. Each code has" --

1 (Stenographer clarification.)

2 THE WITNESS: -- "has highly scaleable and less
3 scaleable complex elements."

4 Q. (By Mr. Limbacher) So what, in what you just
5 read to me, is the rating?

6 A. Exascale Code Block or Complex Code Block.

7 Q. So you're saying the rating is the code block
8 itself?

9 A. Is the type of code block.

10 Q. Okay.

11 Is that provided in the source code?

12 A. I can't think of any other place it would be
13 provided.

14 Q. Does Lippert say that it's provided in the
15 source code?

16 A. Let me see. It says that that's -- the code
17 block -- no, it doesn't say that that is in the source
18 code.

19 Q. Are there other places that a rating could be
20 located, such as maybe a separate metadata file?

21 A. One could.

22 Q. Is there any other places you can think of
23 where such a rating could be -- could be provided?

24 MR. CLAWSON: Objection. Form.

25 THE WITNESS: You're asking me to invent, so --

1 '442 says it's pragmas, which makes sense. If the code
2 is written in or for a different processor type, then
3 that would indicate that it goes to a booster instead of
4 a computational node.

5 I believe, but I've not proven a theorem, that
6 it is a decidable problem. Or you could just annotate
7 the code; the annotation could be in the source code
8 with -- as a pragma or it could be the name of the file,
9 which I suppose you could call metadata, or it could be
10 in, as we do in more modern things like macOS and
11 Windows, I have it as part of the file metadata.

12 Q. Okay.

13 Can you please turn to Budenske. This is
14 Deposition Exhibit D, Exhibit 1005 in the IPR.

15 A. Okay. I think this is it. Hold on. No, it's
16 a different -- okay.

17 Q. What is a heuristic?

18 A. What is a heuristic?

19 MR. CLAWSON: Objection. Form.

20 Q. (By Mr. Limbacher) So let me make this more
21 concrete. Can you go to Page 387. It's the first page
22 of the article, and the abstract refers to heuristics.

23 A. Okay.

24 Q. What is a heuristic?

25 A. It a little -- it's -- go ahead.

1 MR. CLAWSON: I said same objection.
2 Objection. Form.

3 THE WITNESS: Okay. Of course, we can always
4 look in a dictionary for the specific notion of what a
5 heuristic is. Because some people will sometimes say a
6 heuristic algorithm. But most people will say an
7 algorithm computes something, and provides a -- well,
8 that's where it gets sketchy.

9 A heuristic is -- I hate this turn of phrase --
10 but a rule of thumb. For example, a greedy algorithm is
11 a heuristic. And a greedy algorithm is you always pick
12 the biggest thing, and then you pick the next biggest
13 thing, and so forth. That's a heuristic.

14 In general, it will give you a pretty good
15 answer, but it may not give you the best answer. And
16 the reason you use a heuristic is because computing an
17 exact solution may be computationally intractable.

18 So hill climbing, for example, is a heuristic.
19 Genetic algorithms are heuristics. Some randomized
20 algorithms use heuristics.

21 Q. (By Mr. Limbacher) Okay.

22 Budenske refers to an offline heuristic.

23 What is an offline heuristic as referred to by
24 Budenske?

25 A. It's a genetic algorithm.

1 Q. Okay.

2 And then also refers to -- one second.

3 So, for example, on Page 389, it also refers to
4 online heuristics.

5 What would an online heuristic be in that
6 context?

7 MR. CLAWSON: Objection. Form.

8 THE WITNESS: Okay. So in this case, an
9 offline heuristic is the genetic algorithm. And since
10 it's done offline, it doesn't count against the
11 computation time for the task. You can let this run all
12 weekend or a month or however long you want to come up
13 with a bunch of possible mappings that you can choose.

14 An online heuristic is one you would have to do
15 on the fly during computation. This would be -- I found
16 that there was one more tank than I thought there was.
17 Gee, I better throw another booster node at it. That
18 would be a heuristic. An educated guess.

19 Is that going to work as well as one of the
20 results from the genetic algorithm? Probably not, but
21 it can be done online.

22 This goes back to what we were discussing
23 earlier about Big O. If I'm doing something offline,
24 I'm willing to do something that's more complicated
25 maybe. We go --

1 (Stenographer clarification.)

2 THE WITNESS: Maybe Big O in squared, and Big O
3 in cubed might be a reasonable thing to do. If I'm
4 doing something online, I'm going to want to do Big O of
5 one or Big O of login because I need to do it quickly.
6 So my heuristic has to be fast for an online.

7 Q. (By Mr. Limbacher) Are you familiar with the
8 very long instruction word or VLIW architecture?

9 A. I am. I remember Alex Aiken, what was his --
10 Nikolau were working on those at Yale back in the '80s.

11 Q. How would task scheduling for a VLIW
12 architecture be different than for, say, an x86
13 architecture?

14 MR. CLAWSON: Objection. Form.

15 THE WITNESS: Hold on. I dropped something.

16 Well, Alexander Nikolau wrote his Ph.D. on this
17 topic. When you have a VLIW, depending on the VLIW,
18 whether it is homogeneous or not, typically you are
19 doing something that's like a SIMD processor.

20 (Stenographer clarification.)

21 THE WITNESS: So it's often treated as a form
22 of vectorization. The x86 processor has the ADX
23 instruction, for example, that does some of these sort
24 of VLIW things. As did the -- I can't remember -- the
25 one that was joint project with HP and Intel back in the

1 day.

2 The scheduling is really not scheduling in the
3 sense of process scheduling; it's instruction
4 scheduling, so that it's a compiler operation where you
5 want to have multiple instructions that get packed into
6 a very long instruction word. So all of those things
7 happen at once.

8 So you shouldn't think of it as an online --
9 it's not an online scheduling algorithm; it's an offline
10 scheduling algorithm. So it's what's called instruction
11 scheduling.

12 Q. (By Mr. Limbacher) Can you please turn to Page
13 392 of Budenske. And I'm looking at the second
14 paragraph on that page. It states, "For simplicity, it
15 is assumed that if an implementation of a given sub-task
16 uses multiple processors, all processors will be of the
17 same type."

18 A. See that. We've talked about this.

19 Q. Do you know why they made this simplification?

20 MR. CLAWSON: Objection. Form.

21 THE WITNESS: Well, I can easily justify it.
22 I'm not sure what was in their mind when they did it,
23 but if I'm doing the same thing, apropos VLIW, as you
24 were discussing, then I would -- I would expect and want
25 those things to all complete at about the same time.

1 So in the case of VLIW, if part of the VLIW
2 instruction finished, everything in the VLIW instruction
3 has to wait for the last thing to finish.

4 So if things are different, some are slower,
5 some are faster, then you end up waiting a lot. So if
6 you have all the same type, then they should all finish
7 at about the same time. So it's a nice simplifying
8 assumption.

9 Q. (By Mr. Limbacher) So if they don't all finish
10 at the same time, does that make it more difficult to
11 efficiently use your processors to their maximum
12 capacity?

13 MR. CLAWSON: Objection. Form.

14 THE WITNESS: It's not clear to me that even if
15 they finish all at the same time, that you get the
16 maximum capacity. Because you're not doing what's
17 called a global optimum.

18 You might actually be able to do better if they
19 didn't finish at the same time, but, again, it's going
20 to be an incomplete problem to found that global
21 optimum.

22 (Stenographer clarification.)

23 THE WITNESS: Optimum.

24 Q. (By Mr. Limbacher) If you're doing data
25 processing, do you generally try to avoid processor down

1 time?

2 MR. CLAWSON: Objection. Form.

3 THE WITNESS: I don't know what you're asking.
4 Do I want my processors not to crash? Yes, I'd like my
5 processors not to crash.

6 Q. (By Mr. Limbacher) Let me rephrase.

7 Do you try to minimize the amount of time that
8 your processors sit idle.

9 A. We used to worry about that quite a lot and
10 then quit worrying about that in the 1980s for the most
11 part, except for very specialized applications.

12 Q. What are those very specialized applications?

13 A. Well, it's usually not the processor that's the
14 issue; it's usually memory that's the issue. Memory
15 takes the most power and is the slowest. So it's
16 feeding the processor. That's -- that's the issue.

17 When a processor -- what I'm trying to say is
18 the processor is not the most precious resource anymore.
19 It used to be, but it's not anymore.

20 So in a modern processor, most of the time,
21 most of the processor is actually turned off and not
22 being used; it's completely idle. We turn on the parts
23 that we need and we leave the other parts turned off.

24 Q. So just -- so to be clear, it's your view that
25 processors idling during computation tasks is not a

1 problem?

2 MR. CLAWSON: Objection. Form.

3 THE WITNESS: We generally don't idle them. We
4 put them to sleep these days.

5 So the old days of -- of idle spin loops are
6 long gone. Processors have processor states and we
7 power them down to various lower-powered states.

8 Q. (By Mr. Limbacher) By idling, I just mean not
9 processing something.

10 Is a processor that is sitting there not
11 processing something, is that a problem if you're trying
12 to perform a computation task?

13 MR. CLAWSON: Objection. Form.

14 THE WITNESS: We would always like to be able
15 to use our processors to their full capacity, but it's
16 not the processor problem at this point; it's memory
17 that's the problem with feeding the processors. For the
18 most part.

19 Q. (By Mr. Limbacher) So managing the memory and
20 essentially having the data in the correct spot at the
21 correct time is a problem with processing computation
22 tasks -- scratch that.

23 So when you're talking about memory, is the
24 main thing you're concerned about moving into that
25 memory typically data or is it typically source code or

1 both?

2 A. Well --

3 MR. CLAWSON: Objection. Form.

4 THE WITNESS: It's not source code; it's binary
5 code. You don't execute source code; you execute binary
6 code.

7 There are caches and there are cache effects.
8 If the code is made up of tight loops, then the code
9 will fit into the cache, and typically the code will not
10 be an issue. If the code is random, or won't fit into
11 the cache, or sequential, then you will not get good
12 performance. Your cache performance will be bad.

13 The same is true of data. If there's locality
14 in the data, then you'll do well because it will be in
15 the cache. If there's not locality in the data, you
16 will do very poorly because it will not be in the cache.

17 DRAM is typically ten to a hundred times slower
18 than CPU. Those are, of course, approximate numbers.

19 Q. (By Mr. Limbacher) So if you limit processing
20 of a task to a single type of processor, does that make
21 it simpler to get the needed data in a place that's
22 accessible to those processors?

23 MR. CLAWSON: Objection. Form.

24 THE WITNESS: Make it simpler.

25 It might. It's not a --

1 (Stenographer clarification.)

2 THE WITNESS: -- autology.

3 Q. (By Mr. Limbacher) So in the example of
4 Budenske where it's discussing the six SHARC processors
5 that share a -- physically share a DRAM, would them
6 physically sharing a DRAM make it easier for them to get
7 access to data as long as the processing of that data is
8 limited to just the SHARC processors?

9 MR. CLAWSON: Objection. Form.

10 THE WITNESS: I honestly don't know if the
11 SHARC processors had a cache or not. It's been too
12 long. Certainly it's going to make it easier in the
13 sense that they won't have to go over the different bus,
14 so it may make it a bit quicker. It should make memory
15 coherence issues simpler.

16 Q. (By Mr. Limbacher) Are you aware of any
17 working examples of the system described by Budenske?

18 MR. CLAWSON: Objection. Form.

19 THE WITNESS: Working examples? Well, there
20 are published work by Kwok. This mouse is not
21 cooperating.

22 So there's Kwok, which I have no reason to
23 believe didn't work. They have numbers, performance
24 numbers, so they got it working.

25 Q. (By Mr. Limbacher) When you say Kwok, are you

1 referring to Deposition Exhibit G?

2 (Stenographer clarification.)

3 MR. LIMBACHER: Exhibit 1007 in the IPR.

4 THE WITNESS: Let me look. I'm sorry. I got
5 like a hundred tabs open. It's called --

6 (Stenographer clarification.)

7 THE WITNESS: -- "A Semi-Static Approach to
8 Mapping Dynamic Iterative Tasks on the Heterogeneous
9 Computing Systems." My tab is too small. It's Exhibit
10 G. There we go. I just have to highlight it. There's
11 Exhibit H. It also appears to describe a working
12 system. Real nice thing would be to see the reports
13 they sent to the Army.

14 Q. (By Mr. Limbacher) Are you aware of any
15 evidence that they actually implemented an ATR system by
16 use by the Army?

17 A. I have not talked to the Army about that. They
18 do have experimental numbers, so they implemented
19 something. They have measured execution times.

20 Q. Can you turn back to Budenske. This is
21 Deposition Exhibit D, Exhibit 1005 in the IPR, and then
22 go to Page 388.

23 A. Okay.

24 Q. So the first sentence on this page says, "This
25 high-level operating system approach, which includes the

1 capability for the online use of offline computed
2 mappings, is called the IOS intelligent operating
3 system."

4 Did I read that correctly?

5 A. You did.

6 Q. What is the intelligent operating system?

7 A. This -- this whole thing here. HER kernel, the
8 HCR kernel, the algorithm database. It's the way we
9 would call operating system as we would Linux or
10 Windows.

11 Q. Can you please turn to Page 389. The last
12 sentence of the first paragraph says, "The IOS has not
13 been implemented. Such an implementation is a major
14 undertaking and outside the scope of this paper, which
15 is the design concepts for the HC kernel."

16 Do you agree that Budenske is stating that the
17 intelligent operating system was not implemented?

18 A. I agree it was not implemented in 1998, yes.

19 Q. Do you have any reason to doubt the statement
20 in Budenske that such an implementation would be a major
21 undertaking?

22 MR. CLAWSON: Objection. Form.

23 THE WITNESS: I think it's a qualitative
24 statement. I know that they worked on this project for
25 a number of years, so clearly it was not trivial.

1 Q. (By Mr. Limbacher) Do you believe that
2 experimentation would have been necessary to implement
3 the intelligent operating system?

4 MR. CLAWSON: Objection. Form.

5 THE WITNESS: I think it depends on what you
6 mean by experimentation.

7 Q. (By Mr. Limbacher) Well, what do you
8 understand the word to mean?

9 A. Well, it means that to two experimentalists,
10 that you have a falsified hypothesis and so -- but to
11 other people, it would be, well, I have to try something
12 and see if it works.

13 I think that reading Budenske, I'm fairly
14 convinced that they've got most of it sorted out. They
15 have their genetic algorithm that they're going to use.
16 They had some parameters. I think we listed four as an
17 example. They might have done some experiments and
18 found out that five or six dynamic parameters were more
19 interesting.

20 I don't think that experimentation was needed
21 to develop the basic kernel, for example, or the
22 knowledge base, or the algorithm database, or the
23 monitoring. But deciding what the best parameters are,
24 probably some experimentation would be good.

25 Q. So we recently discussed Deposition Exhibit G,

1 which is Exhibit 1007 in the IPR.

2 A. Okay. Hold on. Yes.

3 Q. Did this paper perform any experimentation?

4 MR. CLAWSON: Objection. Form.

5 THE WITNESS: Different kind of experiment, but
6 yes.

7 Q. (By Mr. Limbacher) Okay.

8 Please turn to Deposition H, Exhibit 1008 in
9 the IPR.

10 A. Okay.

11 Q. Was there any experimentation performed for
12 this paper?

13 MR. CLAWSON: Objection. Form.

14 THE WITNESS: Yes, there is.

15 Q. (By Mr. Limbacher) So it seems like for
16 several years after Budenske, they were still
17 experimenting with the concept?

18 A. No, I think they were -- I think you're
19 conflating the word "experiment" to mean they didn't
20 know how to do things. If I'm looking here at H, they
21 tried -- Page 172, for example, they ran a nice
22 experiment, tracking execution time using different
23 heuristics. Fast, greedy --

24 (Stenographer clarification.)

25 THE WITNESS: Different heuristics, OLDB, fast,

1 greedy, min min, min max, greedy GA, which I assume is
2 genetic algorithm. SA, I'm not sure what that is. GSA,
3 taboo.

4 That's the kind of experiment where you say,
5 I've implemented something and I've implemented another
6 thing; which thing is better? It's not a question of
7 does the thing work at all, which thing is better?

8 Q. (By Mr. Limbacher) What do you see in
9 Exhibit -- or Deposition Exhibit H, Exhibit 1008, that
10 tells you that it worked and they were just trying to
11 improve it?

12 MR. CLAWSON: Objection. Form.

13 THE WITNESS: Well, Figure 6.14. They're
14 showing ten different ways of -- ten different ways in
15 getting different execution times. Not hugely
16 different, actually. In fact, within a factor of two,
17 which is actually quite good, even for something as dumb
18 as a greedy algorithm.

19 Q. (By Mr. Limbacher) I'm sorry. Can you tell me
20 what page you're on.

21 A. 172.

22 (Stenographer clarification.)

23 THE WITNESS: A greedy algorithm.

24 Q. (By Mr. Limbacher) Okay.

25 What in here tells you that they were actually

1 able to identify what they were trying to identify with
2 the ATR system?

3 MR. CLAWSON: Objection. Form.

4 THE WITNESS: I have to go back and read the
5 paper again. Sorry.

6 Q. (By Mr. Limbacher) So sitting here today,
7 you're not aware of anything that would suggest that the
8 system functioned. And by "the system," I'm referring
9 to the ATR system Budenske was describing.

10 MR. CLAWSON: Objection. Form.

11 THE WITNESS: I'm not willing to stipulate that
12 it doesn't work, if that's what you're trying to get me
13 to do.

14 Q. (By Mr. Limbacher) I'm just asking if you can
15 point me to anything that shows that it does work or did
16 work?

17 MR. CLAWSON: Objection. Form.

18 THE WITNESS: Now, considering that's not what
19 this paper is about, I can't find anything that says,
20 yes, it worked. It's assumed it worked based on the
21 other results.

22 Q. (By Mr. Limbacher) Can you please go to
23 Deposition Exhibit D. This is Exhibit 1005 in the IPR.
24 This is Budenske.

25 A. Yes. Okay.

1 Q. On Pages 388 and 389, there are three diagrams.

2 A. Okay.

3 Q. Do you agree that these are block diagrams?

4 A. That's not what I would call them, but sure, if
5 you like.

6 Q. What would you call them?

7 A. It's more of a flow chart, flow diagram, but --

8 Q. Would you agree that Budenske is primarily
9 describing the HC kernel?

10 MR. CLAWSON: Objection. Form.

11 THE WITNESS: No.

12 Q. (By Mr. Limbacher) Well, I'm going to point
13 you -- Page 388, the second paragraph, the first
14 sentence of it.

15 A. The first sentence or --

16 Q. The first sentence of the second paragraph on
17 Page 388.

18 A. Okay.

19 Q. So that line says, "This paper concentrates on
20 the operation of the HC kernel. The other components
21 will be discussed only to the extent needed for
22 describing the HC kernel."

23 A. Yes, I agree with that.

24 Q. Okay.

25 Okay.

1 I'm going to introduce an article titled,
2 "Efficient Data-Parallel Distributed DNN Training For
3 Big Dataset Under Heterogeneous GPU Cluster," as Exhibit
4 I to this deposition.

5 A. Okay.

6 Q. This is Exhibit 2014 in the IPR.

7 A. Okay.

8 (Exhibit I was marked for identification.)

9 Q. (By Mr. Limbacher) Have you read this article
10 before?

11 A. I have. And I breezed through it a few days
12 ago.

13 Q. So this system or this paper describes a system
14 with heterogeneous GPUs.

15 Do you agree with that?

16 MR. CLAWSON: Objection. Form.

17 THE WITNESS: Hold on.

18 Yes. Okay.

19 Q. (By Mr. Limbacher) Would you consider a system
20 with heterogeneous GPUs to be a heterogeneous computing
21 system?

22 MR. CLAWSON: Objection. Form.

23 THE WITNESS: I consider a computing system
24 with homogenous GPUs to be a heterogeneous computing
25 system. The more heterogeneity, the more heterogeneous

1 it is.

2 Q. (By Mr. Limbacher) Okay.

3 So just to make sure the record is clear: You
4 do consider a system with heterogeneous GPUs to be a
5 heterogeneous computing system; is that correct?

6 A. Correct.

7 Q. If you look on -- this is Page 179, the first
8 page of the article, the right column, the last sentence
9 of the first paragraph describes the -- what are
10 referred to as "straggler problems."

11 Do you know what straggler problems?

12 MR. CLAWSON: Objection. Form.

13 THE WITNESS: Need to read a bit more context.

14 Q. (By Mr. Limbacher) That's fine.

15 A. Okay.

16 How do I say what this is?

17 It's talking about PyTorch, for example, is one
18 of the -- tensor flow is an exception, apparently. That
19 was Martin --

20 (Stenographer clarification.)

21 THE WITNESS: That was Martin Abadi. He worked
22 on tensor flow at Google I think.

23 They're talking about homogeneous GPU for
24 training of deep neural networks. And the straggler
25 problem would be something competing or completing later

1 than other things. It would be a straggler.

2 And that would probably result from either it
3 being more complicated task, but more often, it would be
4 on a slower GPU. And since synchronous training is
5 what's required in a lot of these systems, we talked
6 about barrier synchronization before, okay, a step,
7 everybody has to finish that step and then go on to the
8 next step.

9 What that means is that the straggler problem
10 means that the training steps are as slow as the slowest
11 node.

12 Q. (By Mr. Limbacher) Would you consider this to
13 be a task distribution problem?

14 MR. CLAWSON: Objection. Form.

15 THE WITNESS: Task distribution problem.

16 I suppose you could call it a task distribution
17 problem.

18 Q. (By Mr. Limbacher) There's a sentence in the
19 abstract that says, "Many researchers have struggled to
20 efficiently conduct distributed training on
21 heterogeneous GPU clusters owing to the straggler
22 problem"?

23 A. Right.

24 Q. Do you disagree with that statement?

25 A. No, I'm not sure "many" have struggled. Could

1 be five; it could be two. That's typical abstract
2 statement, but the straggler problem is a real thing
3 because it's going to represent an issue if you're doing
4 synchronous training.

5 If you get rid of synchronous training, then
6 you don't have to worry about the straggler problem.

7 Q. This paper, on its face, appears to have been
8 published in 2024.

9 Do you agree?

10 A. Looks like it.

11 Q. Did this indicate to you that task distribution
12 problems still existed all the way up through at least
13 2024?

14 MR. CLAWSON: Objection. Form.

15 THE WITNESS: Task distribution problems in
16 general? I think task distribution problems will always
17 be an issue because doing the --

18 (Stenographer clarification.)

19 THE WITNESS: Doing tasks distribution
20 optimally is NP complete, so it's always going to exist.

21 Q. (By Mr. Limbacher) And just for the record,
22 can you explain why a problem being NP complete makes it
23 more difficult to solve?

24 A. Because as far as we know, and we have 60 years
25 of work on this, doing an exact or optimal solution to

1 an NP complete problem requires exponential time, which
2 makes it computationally intractable.

3 Q. Okay.

4 I think I'm at a point where I'm about to move
5 to a new topic, and then I should be able to wrap it up
6 in the next round.

7 Is now a good time for a break?

8 A. Sure. Good time for a walk-around.

9 THE STENOGRAPHER: We're off the record. The
10 time is 2:24 p.m.

11 (Recess taken from 2:24 p.m. to 2:31 p.m.)

12 THE STENOGRAPHER: We're back on the record.
13 The time is 2:31 p.m.

14 MR. LIMBACHER: Cameron there?

15 MR. CLAWSON: Yes.

16 THE WITNESS: Okay.

17 Q. (By Mr. Limbacher) All right.

18 Dr. Long, during the break, did you speak to
19 anyone about your deposition or this proceeding?

20 A. Not a single human being was present.

21 Q. Okay.

22 About how many papers have you published?

23 A. 220, 250, something like that.

24 Q. How many papers have you published on the topic
25 of distribution of tasks and sub-tasks?

1 A. Tasks and sub-tasks.

2 Video scheduling papers are all task
3 scheduling. Probably about ten of those. There was
4 disk data layouts, I don't know how many of those.

5 I don't know. Depends on how you -- how you
6 count a task and how you count layouts and scheduling.

7 Q. So for the first one, did you say video
8 scheduling?

9 A. Yeah. Yeah. I did a lot of -- in the late
10 '90s, I did a lot of video demand scheduling. Video on
11 demand scheduling, I should say.

12 Q. Would you consider those to relate to
13 distributed computing?

14 A. They were absolutely distributed computing.

15 Q. Okay.

16 Would you consider them to relate to
17 heterogeneous computing?

18 A. In fact they are heterogeneous computing. They
19 included set-top boxes and servers.

20 (Stenographer clarification.)

21 THE WITNESS: Included set-top boxes and
22 servers.

23 Q. (By Mr. Limbacher) Is it fair to say that the
24 majority of your work is focused more on data storage
25 rather than task distribution?

1 MR. CLAWSON: Objection. Form.

2 THE WITNESS: Task distribution. I think I've
3 probably written more papers on data storage than
4 anything else.

5 Q. (By Mr. Limbacher) What would you say is your
6 current research focus?

7 A. I'm retired. My research focus right now is
8 looking into physics and number theory.

9 Q. Okay.

10 A. That's not entirely accurate. I'm doing some
11 stuff on anonymity and computer security, but that's
12 low-key.

13 Q. Do you believe your prior publications on data
14 storage are relevant to the -- the claims of the
15 '442 patent?

16 A. The data storage publications?

17 Q. Well, I understood that to be the bulk of your
18 work; yes?

19 A. I said it's the largest number. No. But
20 teaching operating systems for 30 years, which involves
21 scheduling and task scheduling and placement certainly
22 is.

23 Q. I'm going to introduce as Exhibit J, a paper
24 titled, disk failure prediction in heterogeneous
25 environments."

1 A. Okay.

2 MR. CLAWSON: Objection to this, this exhibit,
3 on foundation grounds.

4 MR. LIMBACHER: Okay.

5 Q. (By Mr. Limbacher) Dr. Long, are you one of
6 the authors of this paper?

7 A. I am.

8 Q. Is there anything that would suggest to you
9 this is not an authentic copy of your paper?

10 A. No.

11 Q. Okay.

12 MR. LIMBACHER: So just for the record,
13 Cameron, I disagree with your objection that there's no
14 foundation for this. It's his paper on a relevant
15 topic.

16 MR. CLAWSON: Fair enough. I'll retract the
17 objection.

18 MR. LIMBACHER: Okay.

19 (Exhibit J was marked for identification.)

20 Q. (By Mr. Limbacher) Dr. Long, do you -- do you
21 recall this paper?

22 A. A little. It's pretty old. Some work that I
23 did with my old thesis advisor and his
24 at-that-time-current Ph.D. student, Carlos Rincon, using
25 data from Backblaze as a predictor for when disks might

1 fail.

2 Q. Okay.

3 Can you turn to PDF Page 2. In the left
4 column, there's two paragraphs. The first one starts
5 with Zhu, Z-h-u, et al., and the one after it starts
6 with "other studies."

7 Can you please read those two paragraphs to
8 yourself.

9 A. Okay.

10 Okay.

11 Q. So in that first paragraph I asked you to read,
12 the one that starts with Zhu --

13 A. Um-hum.

14 Q. -- it is discussing a data set, but it says the
15 results are not applicable to heterogeneous
16 environments.

17 Do you agree with that?

18 MR. CLAWSON: Objection. Form.

19 THE WITNESS: I'm sorry. Are you asking
20 whether I agree with that?

21 Q. (By Mr. Limbacher) Well, let's start with is
22 that -- I'm going to read the last sentence. It says,
23 "Even though the authors used a real-world dataset, the
24 distribution of the disks in the dataset, only one
25 model, makes these results not applicable to

1 heterogeneous environments."

2 So first, did I read that correctly?

3 A. You did.

4 Q. And second, do you agree that the
5 homogeneous -- presumably homogeneous dataset there
6 would not be applicable to heterogeneous environments?

7 MR. CLAWSON: Objection. Form.

8 THE WITNESS: We're asking a statistical
9 question. I'd have to go look at Zhu again, but what
10 we're saying is that since all of the disks come from
11 one model, the assumption here is that they have a
12 common failure mode, and consequently would not have the
13 same statistics if we had a heterogeneous set of disks.

14 Not sure where you're going with this, but this
15 is, again, a fairly well-studied area.

16 Q. (By Mr. Limbacher) So is it only the failure
17 mode difference that you consider relevant here?

18 A. Well --

19 MR. CLAWSON: Objection. Form.

20 THE WITNESS: When I say failure mode, I mean
21 the modality of failure. I can't say that; there's an
22 NDA.

23 Hard drives are mechanical devices; they're not
24 electronic devices, as you know, and they tend to
25 exhibit very specific statistics.

1 There was a guy named -- he used to work
2 for NetApp -- that ended up modeling these things with
3 some very fancy statistics because of the unique failure
4 modes of hard drives.

5 You know, they -- they exhibit broadly
6 something called a bathtub curve, which means they have
7 high infant mortality, a long life, and then a high
8 death rate towards the end.

9 Although some common failure modes do occur, if
10 you have ten drives and one of them dies, there's a high
11 probability that another one's going to die very soon
12 afterwards because of the commonality of the failure
13 modes, which is not the same that you would find in
14 electronics.

15 Q. (By Mr. Limbacher) So would you agree that
16 analyzing a heterogeneous disk population is more
17 complicated than analyzing a homogeneous disk
18 population?

19 MR. CLAWSON: Objection. Form.

20 THE WITNESS: No, I think it's actually
21 preferable. You get much better statistics. That's
22 kind of the point of this paragraph, is these guys did a
23 homogeneous disk population and it gave what I would
24 consider a poor model for the prediction of failure.

25 Q. (By Mr. Limbacher) If you're building a model

1 for prediction of failure, is it easier to build one for
2 a homogeneous population or a heterogeneous population?

3 A. Is it easier?

4 Q. Let me scratch that.

5 Is it easier to -- if your goal is maximizing
6 accuracy, are you more likely to be able to do so for a
7 homogeneous population of disks or a heterogeneous
8 population of disks?

9 MR. CLAWSON: Objection. Form.

10 THE WITNESS: This is a -- this is a fun
11 question because it really depends on the degree of
12 homogeneity.

13 There's such a thing as Friday-afternoon disks.
14 So if you mean homogeneous and my serial numbers are
15 within a certain range, which I'm very unlikely to get,
16 then I probably will be able to be very accurate because
17 I'll probably be able to say, These were made on late
18 Friday afternoon and everybody wanted to go home. This
19 is actually a real thing.

20 If it's just they're all model X made in 2023,
21 then there's going to be sufficient variability and it
22 will be a larger common failure mode, if there is a
23 common failure mode that will not be that easy to get.
24 It might actually be harder to tease out. In the law of
25 large numbers, it's actually a true thing.

1 Q. (By Mr. Limbacher) When I read these two
2 paragraphs, they seem to suggest to me that homogeneous
3 disk data is less ideal for building a predictive model.
4 Am I misinterpreting that?

5 A. What we're saying is that if you have a
6 homogeneous population, you can make a model and I don't
7 care. Because it's very rare for you to have a
8 completely homogeneous system. So it's better to do
9 some higher powered statistics to try to come up with a
10 better model.

11 Q. Can you turn to PDF Page 7 and read the last
12 paragraph.

13 A. There's no numbers on the page.

14 Q. It's just the final page in the file.

15 A. Okay. And you want the last the conclusion?

16 Q. Yes. Just read that to yourself, please, that
17 final paragraph of the conclusion.

18 A. Okay.

19 Q. So is this final paragraph suggesting
20 additional experimentation that was planned?

21 A. Carlos promised to do it and he never did it.

22 Q. Okay.

23 That does away with my other questions on that,
24 so let me go to the next thing.

25 Going to introduce as Exhibit K, a paper

1 titled, "RESAR, Reliable Storage and Exabyte Scale"?

2 (Exhibit K was marked for identification.)

3 THE WITNESS: Ahh, yes. RESAR, which means
4 something in Spanish. I don't remember what Thomas --
5 the acronym was for that.

6 Q. (By Mr. Limbacher) Are you one of the authors
7 of this paper?

8 A. I am.

9 Q. Are you familiar with this paper?

10 A. I am.

11 Q. Is there anything that would suggest to you
12 that this is not an authentic copy of your paper?

13 A. We wrote a couple papers on this. Which one is
14 this? This is 2016 mascots. Okay. 2016 paper. We
15 wrote a couple other papers on this topic.

16 Q. Okay.

17 Are you still reviewing or?

18 A. I'm -- I'm having sort of fond memories of ten
19 years ago, when we actually spun up a million computers
20 to run this file, this storage system, Sandia National
21 Lab. That's just why Kroger's listed on there. We had
22 a million virtual machines running this thing.

23 Q. Sounds like a fun project.

24 A. Yeah.

25 Q. So is there -- is there anything you see in

1 here that would indicate to you that this is not an
2 authentic copy of your paper?

3 A. No, it looks like it.

4 Q. Okay.

5 In the -- on the first -- let's see here. This
6 is Page 1.

7 A. Um-hum.

8 Q. On the right column, this is the third
9 paragraph down, it says, "By introducing a disklet layer
10 into the organization of a disk array, we need a
11 metadata server that manages the mapping of disk but
12 obtain the following advantages," and then listed off
13 several advantages presumably of the disklet layer."

14 (Stenographer clarification.)

15 THE WITNESS: D-i-s-k-l-e-t.

16 Q. (By Mr. Limbacher) So what is the disklet
17 layer?

18 A. The disklet layer, if you're familiar with
19 storage systems and disk striping?

20 Q. Grade?

21 A. Like grade, sure. A stripe is an example. If
22 you look at Butter FS, Butter FS has a certain
23 allocation size. Most modern file systems has a certain
24 allocation system, megabyte, gigabyte, something, It
25 depends on which one you're doing.

1 And what we did here is that it was very
2 unlikely that you're going to find a million hard drives
3 that are exactly the same size, even if they're the same
4 model. Because some of them will have bad blocks and
5 some of them won't.

6 So what we did is we abstracted and came up
7 with a disklet, which is a small thing, and then we
8 would map the small -- many of the small things. Just
9 like you'd have many stripes in arrayed array, you'd
10 have many disklets per actual hard drive.

11 Q. Did the concept of a disklet layer exist before
12 your work on this?

13 A. Certainly the idea of allocating in thick size
14 chunks existed before this. I'm not sure if it was
15 called a disklet or not. We're not claiming the idea of
16 a disklet. This is a small fixed size thing.

17 I think it existed -- this is 2016. Oh, yeah,
18 yeah, yeah. The file system in Linux had a one gigabyte
19 allocation unit, which was taken advantage of by Luster.
20 So, yeah. It's been -- the idea of abstracting away
21 from the physical disk in smaller, but still large
22 chunks is not new.

23 Q. So what would you say -- or would you say there
24 was a novel aspect to this project?

25 A. Well, there's the graph layout; the graph

1 theoretic layout of the parity code.

2 Q. Okay.

3 So you don't attribute the novelty the -- to
4 the disklet layer itself?

5 A. No, that's just to make things fit onto a
6 heterogeneous set of disks.

7 Q. Okay.

8 Why was that needed to make things fit onto a
9 heterogeneous set of disks?

10 A. I have a really big disk and some small disks
11 and I want it to fit, then I need to chop it up into
12 chunks that divide into the size of the disks.

13 It's just -- it's just numbers; right? And if
14 I have some ten gigabyte disks and some hundred gigabyte
15 disks, it's 2016 after all, then one of those can have
16 ten disklets and the other one can have a hundred
17 disklets. And in order to make this work properly, I'm
18 going to have to make sure that these things don't
19 overlap each other because then they have a common
20 failure mode and so forth.

21 Q. So that was a problem you had to solve in order
22 to enable the use of heterogeneous disks in this
23 application; is that correct?

24 A. Right.

25 Q. And that was more complicated than if you were

1 just going to homogeneous disks that were all the same
2 size?

3 MR. CLAWSON: Objection. Form.

4 THE WITNESS: Mildly. Mildly. I mean, we
5 would have done it anyway. You would never use -- you
6 would never use an entire disk. Just like you would
7 never use an entire disk for an array strike.

8 Q. (By Mr. Limbacher) All right. I'm going to
9 introduce as Exhibit L, a paper titled, "Duplicate Data
10 Elimination in a SAN File System."

11 (Exhibit L was marked for identification.)

12 THE WITNESS: Yep. Oh, yeah. 21 years ago. I
13 remember this.

14 Q. (By Mr. Limbacher) Are you one of the authors
15 of this paper?

16 A. I am.

17 Q. Are you familiar with this paper?

18 A. Barely. It's 21 years ago.

19 Q. Fair enough.

20 A. It's when I was still doing data deduplication.

21 Q. Is there anything that would suggest to you
22 that this is not an authentic copy of your paper?

23 A. As far as I remember, it looks like the paper.
24 There's how you spell Hineman. I was right.
25 H-i-n-e-m-a-n. There's Wayne. Right there.

1 Q. Can I get you to turn to Page 312.

2 A. Okay.

3 Q. And read Section 8 entitled, "Future Work."

4 A. Storage Tank, by the way, became IBM Total
5 Storage SA-n.

6 Q. Okay.

7 A. Storage Tank was the internal name.

8 Make this a little bit bigger. Yep.

9 Okay.

10 Q. What are file and block access patterns?

11 A. What about them?

12 Q. What are they?

13 A. File access pattern is the time and order in
14 which you access a file, and the order in which the
15 pattern in which you access the contents of the file.

16 Files are made out of blocks. Blocks,
17 therefore, because of file access patterns, can block
18 access patterns, which can be sequential, they can be
19 random, they can be strided, they can be any number of
20 different things.

21 Q. So I'm looking at Paragraph 3 in Section 8, and
22 it says, "This policy is suboptimal in terms of
23 efficiency, although we have considered file and block
24 access patterns in our design, we do not, for
25 simplicity, specifically elaborate policies" --

1 (Stenographer clarification.)

2 Q. (By Mr. Limbacher) -- "explicitly elaborate
3 policies that favor sequential fingerprint probing and
4 matching under certain block access patterns.
5 Therefore, further research on such policies is needed."

6 Why did you make that choice for simplicity for
7 purposes of this paper?

8 MR. CLAWSON: Objection. Form. Objection
9 relevance.

10 THE WITNESS: Twenty-one years ago, why did we
11 do that? Because what we were trying to prove was
12 deduplication would work in the storage system we
13 weren't trying to optimize for performance.

14 Q. (By Mr. Limbacher) Okay.

15 What is the further research on such policies
16 that you reference?

17 A. Well, Beau went -- left IBM and went to Pure
18 Storage and became a rich guy, moved to China. You'd
19 have to ask him.

20 I don't think we followed up this paper. When
21 you write a future research section like this, often
22 you're giving hints to people, other people, to do it in
23 the future. And that's one in this case.

24 And what we said is you could, for example,
25 take the expected access pattern into account. Now, the

1 fact that we use solid state and things instead of hard
2 drives, most of this doesn't matter anymore.

3 If we have a file system where the access
4 pattern is sequential, then, for example, we mention
5 coalescing, we could be more aggressive when we
6 deduplicate and coalesce, so that sequential access is
7 better performing. We didn't bother to do that.

8 Q. Do you think the further research would have
9 required further experimentation?

10 MR. CLAWSON: Objection. Form.

11 THE WITNESS: You and I have a different
12 definition of experimentation. Your definition is will
13 it work; my definition of experiment is I have a
14 hypothesis and I want to know if it's true or not.

15 I would have, for example, said, Well, let's
16 try sorting the fingerprint tables. It's a stupid idea,
17 but I -- and see if that helps. Turns out it won't
18 help, but we could try.

19 How would we coalesce, we would probably
20 coalesce on actual block addresses versus -- versus
21 fingerprints and that would give us a high degree of
22 spatial localities.

23 Q. (By Mr. Limbacher) How -- sorry.

24 A. And I would be willing to bet a lot of money
25 that that would work, and then I would run an experiment

1 to see if I was right.

2 Q. How predictable do you think those experiments
3 would have been before you ran them?

4 MR. CLAWSON: Objection. Form.

5 THE WITNESS: Pretty predictable because I
6 would have done the math first.

7 Q. (By Mr. Limbacher) Dr. Long, who wrote the
8 first draft of your declaration?

9 MR. CLAWSON: Objection. Form.

10 THE WITNESS: The first draft would have been
11 me.

12 Q. (By Mr. Limbacher) I'm going to introduce a
13 document titled, "Compare results," as Exhibit M to this
14 deposition. This is Exhibit 2008 in the IPR.

15 (Stenographer clarification.)

16 (Exhibit M was marked for identification.)

17 Q. (By Mr. Limbacher) Dr. Long, I'm going to
18 represent to you that this was a -- a computer-generated
19 comparison of Sections 1.3 and 9.5 between your
20 declaration and the petition. The highlighting
21 illustrates the differences.

22 Are you surprised that there is so much overlap
23 between them?

24 A. No.

25 MR. CLAWSON: Objection. Form.

1 THE WITNESS: No. Because I helped with the
2 petition.

3 Q. (By Mr. Limbacher) Was the text in the
4 petition taken from the declaration or vice versa?

5 MR. CLAWSON: Objection. Form.

6 THE WITNESS: I honestly don't remember how
7 that worked out.

8 Q. (By Mr. Limbacher) Does your declaration add
9 to the analysis in the petition for these limitations?

10 MR. CLAWSON: Objection. Form.

11 THE WITNESS: I have to -- I would want to do a
12 larger difference. I think you're saying that -- which
13 thing is here? Does it add to? Yes. Does it add a
14 lot? Probably not a lot.

15 Q. (By Mr. Limbacher) How many times have you
16 testified as an expert witness in the past?

17 A. I wish I had my computer in front of me. On
18 the order of 50.

19 Q. Okay.

20 Were all of those patent cases?

21 A. No, there have been -- what do you call them?
22 There was one for Tehama County. There was -- mostly
23 patent cases. Some trademark cases. And one -- one in
24 England that never went anywhere.

25 Q. For the patent cases, about what percentage of

1 those did you testify on behalf of the patent owner?

2 A. Majority of them.

3 Q. Okay.

4 Have you ever turned down a patent challenger
5 case because you did not agree the patent was invalid?

6 A. Say that again very carefully.

7 Q. Have you ever turned down an offer to work for
8 a patent challenger because you did not think the patent
9 was invalid?

10 A. Yes.

11 Q. Has that happened more than once?

12 A. Yes.

13 Q. All right.

14 I need to make one more quick review, but I
15 think I am probably done.

16 Is it okay if we go off the record for five
17 minutes while I do that?

18 A. Sure.

19 THE STENOGRAPHER: We're off the record. The
20 time is 3:03 p.m.

21 (Recess taken from 3:03 p.m. to 3:05 p.m.)

22 THE STENOGRAPHER: We're back on the record.
23 The time is 3:05 p.m.

24 MR. LIMBACHER: Dr. Long, I have no further
25 questions for you at this time. I really appreciate

1 your time and effort to be here today. So with that, I
2 will pass the witness.

3 (Stenographer clarification.)

4 MR. CLAWSON: Can we go off the record, please.

5 THE STENOGRAPHER: We're off the record. The
6 time is 3:05 p.m.

7 (Recess taken from 3:05 p.m. to 3:12 p.m.)

8 THE STENOGRAPHER: We're back on the record.
9 The time is 3:12 p.m.

10 MR. CLAWSON: Okay. Yeah. We have no
11 questions on redirect. We'll thank Dr. Long for his
12 time here today, and we'll end it there.

13 THE STENOGRAPHER: Counsel, would you like to
14 order a copy of today's transcript?

15 MR. CLAWSON: Let's see. We'll do standard
16 timing.

17 THE STENOGRAPHER: Would you also like to order
18 a rough draft?

19 MR. CLAWSON: Let's see. Yeah, we don't --

20 THE WITNESS: I'm going to go, guys, and get
21 back to this board meeting. Okay?

22 THE STENOGRAPHER: We're off the record. The
23 time is 3:12 p.m. This concludes today's deposition.

24 (Whereupon, the deposition of DARRELL D.E. LONG
25 concluded at 3:12 p.m.)

1 I, HEATHER J. BAUTISTA, CSR No. 11600, Certified
2 Shorthand Reporter, certify:

3 That the foregoing proceedings were taken before
4 me at the time and place therein set forth, at which
5 time the witness declared under penalty of perjury; that
6 the testimony of the witness and all objections made at
7 the time of the examination were recorded
8 stenographically by me and were thereafter transcribed
9 under my direction and supervision;

10 That the foregoing is a full, true, and correct
11 transcript of my shorthand notes so taken and of the
12 testimony so given;

13 () Reading and signing was requested/offered.

14 (XX) Reading and signing was not requested/offered.

15 () Reading and signing was waived.

16 I further certify that I am not financially
17 interested in the action, and I am not a relative or
18 employee of any attorney of the parties, nor of any of
19 the parties.

20 I declare under penalty of perjury under the laws
21 of California that the foregoing is true and correct.
22

23 Dated: September 22, 2025

24 

25 HEATHER J. BAUTISTA, CSR, CRR, RPR, CLR

((i)'s 120:13,25 - --ooo-- 6:3 -o-u-i-e-r 57:13 1	11:54 115:14,15 125 85:10 12:30 115:15,17 14 88:9 15-ish 55:4 156 125:23 17 6:1,5 102:1 17-ish 55:4 172 158:21 159:21 179 163:7 19 62:16,19,22 63:9 1960s 72:16 1970s 72:1,17 114:20 1980s 41:25 151:10 1986 115:3 1990s 44:14 1998 99:20 105:21,24 156:18 1:20 138:8,9 1:28 138:9,11	25 113:15 250 166:23 26 16:10 20:13 21:11 63:23 27 20:14 28 88:7 2:24 166:10,11 2:31 166:11,13
1 16:22 20:4 22:13 23:2 24:10,14,18 26:10,14 27:11,17 28:19,25 29:8,11, 13 30:23 31:18 32:20 44:15,17 46:3 48:8 49:10 56:19 61:11 62:5,6 88:17 119:10 120:22 143:11 176:6 1.3 183:19 10 80:16 1001 12:24 16:6 56:22 59:4 61:10 139:19 1003 13:10 15:19 40:11 60:7 62:14 71:21 82:19 88:1 91:13 92:24 127:12 137:7 138:20 141:16 1004 13:18 56:12 88:16 143:15 1005 14:4 107:13 129:18 145:14 155:21 160:23 1006 14:12 78:20 1007 116:9,11,16 124:13 155:3 158:1 1008 116:9,11 125:19 158:8 159:9 101 90:14 102 90:13 104 76:1 10:02 61:24 62:1 10:04 62:1,3 10:54 87:17,18 11 141:2 11,537,442 12:23 14:20 112 84:16 116 77:21 11600 6:13 119 40:15 82:17 11:03 87:18,20	2 2 45:24 46:1,4 88:17 170:3 20 9:19 142:3 2000 113:3 2000s 42:8 43:25 65:10 2008 183:14 2013/0282787 13:17 2014 162:6 2016 175:14 177:17 178:15 2017 16:14,15 18:3 39:24 40:3,6 41:23 2018/0074855 14:11 2023 173:20 2024 165:8,13 2025 6:1,5,16 41:23 20s 9:19 21 179:12,18 220 166:23	3 3 17:8 34:23 45:24 46:1,4 50:15,19,22 51:1,4,13 52:2 73:8 116:23 117:1 139:14,21 140:20,22 141:7 180:21 30 112:25 115:5 168:20 30-second 112:10 31 66:23 68:9 93:1 312 180:1 32 68:24 71:22 33 32:20 43:23 45:10,23 46:8,21 50:1 59:6 35 17:8 49:4 36 32:20 43:23 45:10,23 46:8,21 49:4 37 48:8 49:3,4 59:6 386 39:5,6 42:4 387 145:21 388 155:22 161:1,13,17 389 147:3 156:11 161:1 39 52:9 392 102:10,14 107:14 149:13 393 129:20 130:1 132:22 133:12 3:03 185:20,21 3:05 185:21,23 4 4 50:15,19 51:5 60:10,11,16 90:6,9 139:15,23 141:1,8 40 32:13 48:8 43 137:11,13 442 10:10 13:3 16:5,22 17:4,19,20 19:5,18 23:20,24 24:6 25:5 26:19 32:19 43:22 56:18,21 57:21 59:3 61:9

62:6 71:2,9,11 139:18 140:16 143:10 145:1 168:15		
45 110:17		
48 15:21 45:6		
486 39:3,5 42:5,6		
49 56:19 143:10		
5	8	absolutely 111:24 167:14
		abstract 72:18 92:8,11 145:22 164:19 165:1
5 16:8,10 20:3,13 32:20 43:23 45:10, 23 46:8,21 48:8 51:5 61:11 62:5 80:19 90:6,9 124:19 140:22	8 54:20 57:21 80:11,14,15,19,24 180:3,21	abstracted 177:6
5-ish 51:14	80 122:17	abstracting 177:20
50 8:8 184:18	80s 148:10	abstraction 54:7
500 64:18	81 116:21 117:4,21	AC 92:3
512-bit 19:1 20:1 96:1	83 119:7	accelerator 19:1 66:8
52 75:22 76:4	8386 42:2	accelerators 137:17
53 76:22	84 143:13	accept 20:11
54 127:18	88 138:22	access 12:20 154:7 180:10,13,14,15, 17,18,24 181:4,25 182:3,6
56 17:11 34:23	89 78:21,24 79:24 115:3	accessible 79:17 108:14,18 153:22
57 88:2	8:02 6:2,6	accommodate 101:6
59 89:1 91:10	8:14 15:12,13	account 181:25
	8:17 15:13,15	accuracy 173:6
6	8:50 32:5,6	accurate 134:21 168:10 173:16
	8:56 32:6,8	achieve 64:1
6 16:8 21:11 49:3 50:1 51:4,13 52:9 55:3,4 56:24 59:5,6 61:11 140:19,22	9	acquired 66:21
6.14 159:13	9 21:11 24:11,15 26:11 29:8 31:19 47:25 48:1 49:2,9 50:20 51:2 52:9,25 53:11 59:5 61:11	acquisitions 64:8
6.5.2 126:1	9.5 183:19	acronym 175:5
60 9:16 165:24	90 127:12	actions 53:5
61 92:25	90s 167:10	actual 50:9 63:20 124:17 135:15 177:10 182:20
63 93:15	91 115:3	acyclic 99:10
64 105:21	92 141:17	add 114:10 135:4 184:8,13
65 98:1 115:23	93 90:6,8 91:3,20 92:18 115:4	addition 20:1
68 40:13	94 78:22	additional 86:1 174:20
	9:49 58:20,21	address 35:15 61:25 83:17
7	9:56 58:21,23	addressed 17:13 35:1
	A	addresses 182:20
7 56:24 141:1 174:11	A(i) 120:12,18,19,21	addressing 43:22 123:1
73 40:19	a.m. 6:2,6 15:12,13,15 32:5,6,8 58:20,21,23 61:24 62:1,3 87:17,18,20 115:14,15	admit 75:11
77 85:19 128:22 129:7	Abadi 163:21	advance 73:22
78 137:9	ability 109:9	advantage 177:19
		advantageous 134:23
		advantages 86:1 176:12,13
		advised 22:4
		advisor 169:23

advisory 64:6,23 65:1 ADX 148:22 affect 103:19 104:19 105:8 130:20 affects 120:9 affiliated 30:20 affinities 121:6 affinity 73:5 afternoon 173:18 AG 6:8 aggressive 182:5 agree 41:3 81:1 91:5 118:19 120:12, 21 156:16,18 161:3,8,23 162:15 165:9 170:17,20 171:4 172:15 185:5 ahead 12:18 51:16 71:8 82:21 85:11 114:11 135:13 145:25 Ahh 175:3 AI 110:2 Aiken 148:9 aircraft 70:4 Alamos 64:5,7,13,14,17 Alex 148:9 Alexander 148:16 algorithm 84:23 85:2 99:23 110:8,9 119:25 125:18 134:16 146:6,7,10,11, 25 147:9,20 149:9,10 156:8 157:15, 22 159:2,18,23 algorithm's 135:14 algorithms 75:18 123:11 146:19,20 allegations 62:21 allocate 94:13 allocating 177:13 allocation 29:19,21,23 72:11 94:17 176:23,24 177:19 alpha 120:19 alter 20:15,18 altering 52:1 alters 51:9 138:25 139:2 140:23 141:2 AMD 78:5 81:25 amount 104:25 117:12,13 151:7 amounts 104:17	analysis 10:23,25 11:3,4 108:21 131:9 184:9 analyzing 172:16,17 annotate 145:6 annotation 56:4,6 145:7 anonymity 168:11 answering 23:15 anymore 73:17 80:2 151:18,19 182:2 Apache 84:19 apologies 8:7 apologize 100:13 apparently 118:7,12 163:18 appearance 6:20 appeared 23:12 appears 88:21 124:16 155:11 165:7 applicable 170:15,25 171:6 application 30:12 45:19 46:4 49:5, 13,16,20 50:12,22 51:1,22 89:16 103:25 104:3,8 111:15 128:15,20 130:2 139:21 140:11 178:23 applications 40:5 49:25 104:5 151:11,12 applied 16:2 26:10,11 35:16,18,21 apply 23:1 24:9,13,17 26:13 29:6,12 31:17 33:11 48:15 49:8,15 50:5,18,25 52:24 55:9 56:25 57:23 83:19 appreciates 90:9 approach 116:14 125:18 155:7,25 approximate 70:13 153:18 Approximately 68:3,7 70:20 apropos 149:23 architects 111:18 architecturally 30:19 architecture 17:16,18 35:12 59:19 100:11 101:16 102:5 105:24,25 108:25 148:8,12,13 area 63:16 171:15 argument 38:20 91:4 Army 133:2 155:13,16,17 array 176:10 177:9 179:7	arrayed 177:9 arrive 67:4 132:6 arriving 135:1 art 11:14 19:6,17 22:24 23:9,11 25:3, 4,6,8 49:23 50:11 53:7,8 56:9,10 57:15 58:11,12,13 77:1 90:9 93:21 116:4 122:25 article 13:25 116:13 124:12 125:16 145:22 162:1,9 163:8 ASCI 64:25 ASICS 42:9 aspect 64:9 177:24 aspects 64:9 86:25 asserting 128:25 139:12 assertion 141:22 assign 59:12,17 93:12 105:8 106:13, 16 assigned 30:19 31:5 45:11 46:9,19 60:3 66:19 75:12 96:18 97:8,24 assigning 30:3 31:10 32:21 45:14, 16,20 59:8 72:17 75:16 93:9 assignment 29:19,21,24 30:7,8,9, 11,14 31:8 72:14,15 75:10 90:16 92:8,11 98:23,24 99:2,4,18,19 100:2 139:3 141:3 assignments 69:3 71:23 assigns 50:9 assume 33:25 34:4 103:1 134:2 159:1 assumed 102:6,19 149:15 160:20 assuming 44:19 70:14 118:14 119:4 assumption 84:5 150:8 171:11 assumptions 118:11 at-that-time-current 169:24 ATR 134:17 155:15 160:2,9 attorney 6:21 20:17 22:9 attorneys 10:4,13 11:7,16 15:3 attribute 178:3 attributing 113:22 authentic 169:9 175:12 176:2 179:22 author 20:9
---	--	---

authors 133:16 169:6 170:23 175:6 179:14 autology 154:2 automatic 14:1 132:23,25 133:5 average 130:5 136:13 avoid 150:25 AVX 18:8 aware 38:5 62:21 81:14 126:14 142:19 154:16 155:14 160:7	Bautista 6:11 Beau 181:17 bees 138:15,18 began 111:23 beggars 106:18 begin 7:12 begins 49:4 behalf 6:23,25 185:1 behavior 124:6,7 believed 110:3 belongs 59:22 beneficial 51:23 benefit 93:9 97:17,19 bet 182:24 — beta 119:24 BI 119:24 — big 83:20 97:22 113:18,21 114:1,6,8 125:12 147:23 148:2,4,5 162:3 178:10 bigger 180:8 biggest 146:12 billions 64:6 bin-packing 72:19 binary 79:10 110:8 153:4,5 bit 94:18 100:22 118:4 154:14 163:13 180:8 black 88:7 block 93:19,20,22 95:6 96:10,16 144:6,7,9,17 161:3 180:10,17,23 181:4 182:20 blocks 93:17,24 94:1,2,5 95:17 97:16 143:22,23 177:4 180:16 board 64:6,23 65:1 103:2,3 Bob 111:17 body 48:20 bonk 140:8 book 72:1 boost 29:19 boost- 23:21 booster 16:23,25 17:17 18:7,9,23 19:3,5,9,13,16 20:5 21:2,20,21 23:2,	4,8,21,25 24:6,14 29:20 30:5,13,16, 17,18 31:11,14 32:24 33:15,21,23,25 34:2,5,17,19,21 35:7,11 37:7,15 38:2, 6,12,16,19,23 39:1,6,9 41:13,16,17, 20 42:4,13,19 43:19,25 45:11 46:9, 13,19 47:8,15 48:12 49:22 50:10 53:14,17 56:3 59:9,13,21 60:2 65:8 67:11 71:18 76:12,19,20 90:16,19 93:10 96:8 97:24 100:18 101:2 106:1, 14,16,23 107:2,7 127:22 128:1 130:21 139:4 141:4 142:17 145:3 147:17 boosters 40:24 91:2 boosting 18:10 42:20 boosts 43:17 bother 182:7 bottom 117:4,21 119:6 129:20 132:22 133:12 boxes 167:19,21 branch 72:6 break 8:23,25 31:22 32:10 58:14,16, 25 87:10,12,23 115:18 138:3,14 166:7,18 breakroom 32:11 breaks 8:21 99:5 breezed 162:11 Brian 111:18 broadly 172:5 broke 100:22 brought 107:5 108:5 Budenske 14:7 85:13 86:20 89:17 92:21 93:3,14 98:5,8,14,22 99:5,17, 19 100:7 101:11,15,23,25 103:5,13 105:15,19 106:8 107:13 108:16,25 127:20 129:1,8,11,15,16,17 130:16, 25 131:15 133:13,15 136:24 145:13 146:22,24 149:13 154:4,17 155:20 156:16,20 157:13 158:16 160:9,24 161:8 Budenske's 104:18 116:6 bug 69:18 bugs 110:19 build 106:19 173:1 building 84:24 138:17 172:25 174:3 built 65:2
B		
B(i) 120:18 121:21 bachelor's 67:13 back 8:12,19 11:12 15:14 20:3 29:25 32:7 33:16 34:22 35:20 43:21 52:8 56:21 58:17,22 61:9 62:2,13 63:4 67:17 71:4,20 72:4 81:19 87:19 90:20 91:10,24 92:23 99:24 101:22 106:7 107:12 110:12,17,21 112:12 113:4 114:22 115:16 127:10 128:17 129:16 137:6 138:10 140:3 141:15,23 142:8 143:8 147:22 148:10,25 155:20 160:4 166:12 185:22 Backblaze 169:25 background 64:12 backwards 62:20 bad 153:12 177:4 baggage 55:16 bandwidth 66:4 103:14,22,25 104:8, 16,19 105:3,7,10 106:2,9,12,14,17 117:14 Barely 179:18 barrier 28:11,15,18 164:6 base 157:22 based 10:18 18:25 19:16,23 23:5 51:21 57:4 67:18 74:23 78:13 121:5, 18 160:20 basic 69:2,6,8 71:22 123:18 157:21 basically 66:7 140:5 basing 10:24 11:3 23:16 76:15 77:3 78:2 bathtub 172:6 battle 135:1		

bulk 168:17 bullet 88:7 bunch 99:21 134:16,25 147:13 bundled 42:24 Burns 111:17 bus 154:13 Butter 176:22 buy 111:3,10	careful 34:9 carefully 110:18 185:6 Carlo 26:18 28:12 Carlos 169:24 174:21 case 6:8 29:19 53:19,20 56:1 64:5 69:19 89:17 95:23 104:23 126:21 140:11 147:8 150:1 181:23 185:5 cases 34:16 93:20 184:20,23,25 CCB 143:23 cell 65:7 66:6 centered 66:10,18 central 85:13,23 127:23 cents 32:13 Ceph 111:7 Certified 6:12 cetera 90:18 challenger 185:4,8 challenges 65:24 change 20:21 21:14 83:20 94:23,25 95:3,8 96:18 101:8 136:25 changed 16:18 changing 85:1 97:7 Chapter 73:8 characteristic 55:23 characteristics 57:5 61:21 characterization 17:21 39:11 chart 161:7 check 76:1 checkpoint 70:10 checkpoints 66:5 China 181:18 chips 22:20 36:5 choice 19:12 181:6 choices 137:25 choose 68:14 147:13 chooses 100:1 choosing 31:13 114:6 chop 178:11 Chorush 6:23	chosen 123:11 130:2 Chris 6:22 7:20 15:7 chunk 111:12 chunks 177:14,22 178:12 circling 106:7 circuits 54:6 circular 123:8 citation 77:6 cite 116:9 cited 142:3 Civil 6:16 claim 16:2,22 20:4,9,15,18,21,22 21:11,12,14,23 22:13 23:2 24:10,11, 14,15,18 26:10,11,14 27:11,17 28:19, 25 29:8,10,13 30:23 31:18,19 32:20 44:15,17 45:16,24 46:1,3,4 47:25 48:1,8 49:2,9,10 50:19,20,22 51:1,2,4 52:2,9,25 53:11 54:20 55:3 56:24 57:21 59:5 61:11 62:5,6 129:2,13 139:1,14,15 140:20 141:1,6,8 claiming 177:15 claims 50:15 61:11,14,16 71:1,9,11 168:14 clarification 8:6 13:7 27:8 28:7 35:23 37:20 44:21 51:7 57:11 61:1 64:24 65:15 68:5 69:21 72:9,21 74:10,15,20 79:16 86:12 88:4 90:7,22 91:7 95:25 99:8 103:11 111:5 115:7 116:3 117:10,18 132:17 135:23 144:1 148:1,20 150:22 154:1 155:2,6 158:24 159:22 163:20 165:18 167:20 176:14 181:1 183:15 clarify 9:9 Class 14:2 classic 68:18 69:25 classify 19:15,18,22 Clawson 6:24 10:14 15:7 16:13 17:22 18:21 19:7,19 20:7,16,23 21:15 22:1,7,14 23:3,10 24:19 26:4,15,22 27:13,18 28:4,22 29:3,14 30:25 31:20 33:5,19 34:8 35:9,24 36:15 37:2,9,16 38:7,13 39:12,23 41:4,22 42:14 43:1, 12 44:6,11 45:13,17 46:11 47:2,11,20 48:3,18 49:17 50:7 51:19 52:3 53:18 54:1,25 55:11 57:2,25 59:14 63:1,19 64:3 66:1,15 67:21 71:3,13 73:14,20 74:2 79:6 80:13 81:4,10,18 82:13,25 83:5,12 84:1,7,14 85:6,16 86:16 87:2,
C		
C(i) 120:18 122:8 C-PROGRAMMING 55:23 cache 121:17 153:7,9,11,12,15,16 154:11 caches 121:17 153:7 calculate 136:6 calculated 136:16,20 calculates 134:20 calculation 120:6 California 6:13,15 11:9 32:14 call 8:25 18:9 19:9,12 22:22 34:17 36:17 38:22 76:12,21 99:6,9 113:18 124:5 145:9 156:9 161:4,6 164:16 184:21 called 18:5 25:25 54:8 70:10 72:1 99:14,22 111:11 112:7 120:4 134:11 149:10 150:17 155:5 156:2 172:6 177:15 calling 19:2 55:19 67:11 75:14 93:10 Cameron 6:24 10:2 136:2 166:14 169:13 capabilities 34:16 78:6 capability 34:2 37:25 38:3 43:15,17 156:1 capable 19:23,24 86:15 capacities 127:21 capacity 150:12,16 152:15 caps 90:23 capture 133:6 cards 101:19 care 104:12,14,16 174:7		

7 88:14 91:23 94:6,24 95:4 96:21 97:9 98:20 100:5,20 101:7,14 103:7, 16,21 104:6,21 105:6,9 106:11,24 108:12,23 109:8,15,24 110:15 111:1 112:18 113:19 114:16 118:2,9,20,24 120:16 121:3,12,23 122:10,21,24 123:6,15 124:3,15,24 125:6 126:9,16 128:8,11 129:3 131:1 132:16 133:14, 25 134:5 135:9,21 136:8 137:3 139:16 140:2 141:10 142:2,21 143:7 144:24 145:19 146:1 147:7 148:14 149:20 150:13 151:2 152:2,13 153:3, 23 154:9,18 156:22 157:4 158:4,13 159:12 160:3,10,17 161:10 162:16,22 163:12 164:14 165:14 166:15 168:1 169:2,16 170:18 171:7,19 172:19 173:9 179:3 181:8 182:10 183:4,9,25 184:5,10 cleaner 41:18 clear 11:4 29:17 54:12 97:1 119:25 131:16 150:14 151:24 163:3 climbing 146:18 clock 121:16 close 112:2,3 closely 66:7 cloud 130:9 131:17,18 132:6 134:13, 14 136:16 cloud-era-based 75:16 cluster 65:9 74:1,5 78:13 80:6 81:24 82:1 162:3 clusters 78:3 80:18 84:21 164:21 CN 90:20,21,23 coalesce 182:6,19,20 coalescing 182:5 coarser 75:12 coarser-grained 75:20 code 6:15,18 55:7,10,20 56:7,8 69:17 76:17,19 82:22 83:3,21 84:2,5,12 85:4,7 93:17,19,20,22,24 94:1,2,4,9, 12,13,16,17,20,23,25 95:3,5,7,10,12, 16,18,20 96:6,10,12,15,24 97:3,5,7, 13,15,16,19,20,22 98:4,7,9,10,12,14, 17,18 99:1 107:1,9 109:19,21,22,23 110:1,13 111:12,21,22 112:11,17,19, 20,23 114:23 115:8,9 128:21 136:11, 12 143:6,22,23,24,25 144:6,7,9,11, 15,16,18 145:1,7 152:25 153:4,5,6,8, 9,10 178:1 codes 69:22,23	coefficient 120:13,24 121:1,8 122:3 coefficients 120:18 coffee 31:25 coherence 154:15 collate 128:18 colleagues 77:4 collect 128:1,18 collection 42:18 college 106:5 column 16:10 17:8 20:3,13 21:11 32:20 34:23 43:23 45:10,23 46:8,21 48:8 49:3 50:1 51:4,13 52:9 55:4 56:19,24 59:5,6 62:5 88:17 119:9 140:19,22 143:11 163:8 170:4 176:8 Columns 16:8 61:11 combination 39:6 86:6 89:5 91:15 100:24 106:8 combine 96:17 98:4,8 combined 85:22 combining 85:12 common 27:7 28:9 81:24 171:12 172:9 173:22,23 178:19 commonality 172:12 commonly 27:4 82:1 93:12 communicate 102:21 communication 60:25 100:16,18 101:5,9,12,19,21 102:23 108:7 communications 102:5,15,18 117:5,22 Compare 183:13 comparison 183:19 compatible 34:14 competing 163:25 compiler 55:15 149:4 complained 32:12 112:11 complete 37:21 44:19 45:3 149:25 165:20,22 166:1 completed 28:20 29:1 142:6 completely 110:21 151:22 174:8 completes 99:23 completing 163:25	complex 94:21 95:20 96:25 97:21 143:21,23,25 144:3,6 complexity 114:10 141:25 142:13 complicated 73:12 109:14,17 118:13 147:24 164:3 172:17 178:25 component 49:20 88:18,23,24 components 161:20 comprise 90:12 142:4 comprising 20:25 21:12,13,19 computation 16:23,25 18:4,5,20 19:2,16 20:4 21:1,19,21 22:13,15,22, 23 23:6,7 24:10 29:20 30:4,12,15,16, 20 31:11,14 32:23 33:4,14,18,22,24 34:3,5,7,11,19,21 35:7,19,21 36:1,7, 14 37:1,4,11,14,19,22 38:6,12 39:9, 19 40:23 41:3,6,10,11,17,21 42:12, 19,23 43:9 44:8 45:7,8 46:14,24 47:4 48:11 49:21 50:10 51:10 53:13,16,23 59:10,12,20,23 60:1 62:8 65:7 67:12 70:1 76:5,8,12 90:15,16,17,18,20,21, 24,25 91:1 100:7,17 101:1 104:24 106:22 127:22,25 130:21 139:3 140:24 141:3,5 142:4,6 147:11,15 151:25 152:12,21 computational 19:25 43:15,17 45:5 69:24 83:15 85:15 86:3,8 105:2,13 107:6 135:5 136:13,21 145:4 computationally 146:17 166:2 computations 46:13 compute 17:17 44:20 45:3 99:21 106:18 computed 89:13 90:19,20 96:4 134:7,8,9,10 156:1 computer 12:3,4 16:20 22:16 23:12 24:22 27:9,20 29:17 35:10 38:22 42:1 52:4,21 53:3 55:1 57:17 59:25 62:19 65:5,9,10 72:16 73:2,25 74:1,4,5,7,12 80:21 90:6 92:17 106:19 111:9 118:10 139:9 168:11 184:17 computer-generated 183:18 computers 60:22 64:18 84:21 106:4 175:19 computes 146:7 computing 16:12,15,16 17:2 20:6, 14,20,25 21:7 26:14,20 27:5,12 28:2, 21 29:1,7 30:2 32:21 34:13 51:10 60:18,20,21,23 61:3,4,7,8,14,17,22 62:7,11 64:2,8,16 67:15,19,24 68:8, 15,16,17,19,20,21,22 70:11 77:25
---	--	--

78:15,25 84:25 89:6,9,20 91:5,16,21, 22 92:14 97:13 100:25 116:15 125:17 127:21 146:16 155:9 162:20,23,24 163:5 167:13,14,17,18 concentrates 161:19 concept 158:17 177:11 concepts 156:15 concerned 152:24 conclusion 62:25 63:9 174:15,17 conclusions 124:22 concrete 145:21 concurrency 28:8 99:15 concurrently 25:18,23 26:2 86:19, 21 conditions 130:7 136:15 conduct 164:20 conference 77:12 conferences 77:11 configured 128:1 confirm 127:3 conflating 158:19 conflict-free 102:8,20 103:1 117:15 connected 21:6 30:20 60:25 consideration 106:9 considerations 63:3 considered 25:19 65:9 93:21 98:17 123:17,22 180:23 constant 114:6 125:13 consult 65:11 consulted 23:24 65:25 contents 180:15 context 17:3 18:22 19:2 23:13 28:19, 25 40:20 50:13,14 53:3 94:11 133:7 147:6 163:13 continue 131:24 142:9 contrast 130:9 controlled 59:22 conventional 80:1 convinced 157:14 cookies 29:22	cool 71:16 cooperating 154:21 coordinate 46:14 coordinating 30:21 copies 70:1 copy 69:14 72:1 169:9 175:12 176:2 179:22 core 18:25 19:23 81:23 cores 20:2 Corporation 6:8 7:1 correct 7:14 9:24 16:23,24 18:1,2, 13,14 22:6,8 23:17,18 30:24 31:8,9, 11,12,15,16 39:11 44:1,2,10 45:25 52:19 60:18,19 74:11 75:7,25 77:1,2 80:5,11 83:1,25 84:6 85:5,23,24 86:4, 5,23,24 88:10,11,20 89:7,8 91:17,18 95:14 101:2,3 108:22 109:1 114:23 116:10 120:3 127:5 129:9,24 130:14, 19 131:9,10 134:4 137:15 139:5,6 143:16 152:20,21 163:5,6 178:23 corrected 70:8 correctly 33:1 48:13 53:4 69:4 93:5 128:3 156:4 171:2 council 65:18 counsel 6:19 7:12,21 count 147:10 167:6 counting 135:7,15 136:9 County 184:22 couple 7:15 11:7 93:1 175:13,15 court 9:6 cover 130:10 131:18 132:6 134:13, 14 136:17 COVID 8:13 CPU 22:18 36:3,4,6 40:22 41:5,6,10, 12 42:16 43:14 54:6,9,11,16,17 58:5 153:18 CPUS 41:3,13 42:8 67:10 crash 70:17,19,24 151:4,5 crashes 70:18,23 create 134:16 created 134:11 cross-talk 68:4 crossbar 117:17,19	cubed 148:3 curiosity-driven 123:20 current 131:24 168:6 curve 172:6 CV 112:15
D		
D-A-E-M-O-N-S 52:10 D-I-S-K-L-E-T 176:15 D.E. 7:3 daemon 52:12,19,20 53:1,23 54:16, 19,21,23,24 88:8,13,18,24 128:16,18 daemons 52:25 53:11,16,17 61:20 127:25 128:6,10,12,21 DAG 99:9 132:2,5 137:21 DAGS 99:6 Darell 6:7 7:3,9 13:9 darn 112:2,3 Darryl 7:24 data 26:3,6,7 36:4 69:20 103:18 104:15,18,25 105:17 106:2 107:2,4,7, 25 108:10 109:6,9,12 130:19 150:24 152:20,25 153:13,14,15,21 154:7 167:4,24 168:3,13,16 169:25 170:14 174:3 179:9,20 Data-parallel 162:2 database 156:8 157:22 dataset 162:3 170:23,24 171:5 date 6:5 dating 42:8 Daughter 103:2 day 8:22 70:19,22 149:1 day-mon 52:15 days 93:12 152:4,5 162:11 DDG 99:7,9 100:1 134:24 137:17,20 DDGS 99:6 134:11,17 death 172:8 decidable 145:6 decide 99:3 decided 99:18 118:3		

decides 49:21 deciding 19:9 30:11 73:1 106:13,15 142:17 157:23 decimal 125:9 decision 131:23 decisions 75:20 declaration 13:9,13 15:18,22 40:10 60:6 62:13 63:23 66:23 68:9,24 71:21 75:22 76:23 77:22 82:18,19 85:10 87:25 88:3 89:1 91:11 92:23 93:16 98:1 115:24 127:4,10 128:23 137:6 138:19 141:15,16 183:8,20 184:4,8 deduplicate 182:6 deduplication 179:20 181:12 dee-mon 52:15 deep 163:24 define 34:20 57:3 81:20 109:11 defines 99:15 100:10 definition 22:10 33:6,11 34:10 67:4 68:15,18 120:22 123:18,19 182:12,13 degree 67:13 99:15 138:2 173:11 182:21 delays 117:17 delivered 107:8 demand 167:10,11 department 90:5 depend 122:4 dependencies 99:12,13 dependent 30:19 42:24 43:10 depending 25:1 148:17 depends 28:5,8 33:6 43:6 136:19 157:5 167:5 173:11 176:25 deployments 63:20 deposed 7:25 deposition 6:7,14,17 10:5,6 12:23 13:10,18 14:4,12,21 15:18 16:6 32:16 40:11 56:11,22 59:1,4 60:7 61:10 62:14 71:21 78:19 82:19 87:22 88:16 91:11 92:24 107:13 115:19 116:16 125:18 127:8,11 129:18 137:7 138:14,20 141:16 143:15 145:14 155:1,21 157:25 158:8 159:9 160:23 162:4 166:19 183:14 derived 14:1 74:13	describe 24:23 155:11 describes 79:1 94:9 101:15 162:13 163:9 describing 160:9 161:9,22 design 65:21,22 66:3 69:2,6,8 71:22 110:13,20 111:8,22 112:1,6 156:15 180:24 designed 22:21 76:18,19 designs 64:8 desk 100:15 103:23 desktop 73:25 74:4 detail 21:9 details 48:24 74:19 125:11 detect 134:24 135:5 detected 130:10 134:22 detecting 133:17 determine 124:10 127:23 develop 157:21 developed 72:6,17 114:20 developer 130:2 development 111:20,23 device 79:15 devices 171:23,24 diagram 45:18 103:2 108:13 161:7 diagrams 61:19 161:1,3 dictionary 146:4 die 172:11 dies 172:10 difference 54:23 140:10,14 171:17 184:12 differences 53:9 139:13 183:21 differentiates 21:21 differently 19:5 difficult 12:15 66:17 85:2 110:9 118:17,23 119:1 150:10 165:23 digital 42:7 dimensionalist 121:8 122:3,15 ding 91:25 DIRECT 7:18 directed 99:10	directions 17:12 34:24 directive 55:15 disagree 17:20 41:1 118:22 164:24 169:13 disappears 132:6 disclose 79:13 82:3 disclosed 87:1 116:5 142:23 discloses 77:24 79:14 82:11 83:15 disclosure 78:11,14 81:1,15 discreet 115:2 discuss 54:2 55:14 discussed 66:3 78:10 91:14,19 125:12 157:25 161:21 discusses 55:21 56:14 126:14 discussing 55:13 75:25 147:22 149:24 154:4 170:14 discussion 17:4,9 55:12 discussions 66:10 disk 167:4 168:24 172:16,17,23 174:3 176:10,11,19 177:21 178:10 179:6,7 disklet 176:9,13,16,18 177:7,11,15, 16 178:4 disklets 177:10 178:16,17 disks 169:25 170:24 171:10,13 173:7,8,13 178:6,9,10,12,14,15,22 179:1 display 130:9 disruption 100:14 distinction 34:18,21 35:7 143:20 distributed 60:18,20,21,22,23 61:2, 3,7,14,17,22 62:7,11 67:15,19,24 68:8,14,17,19,20,22 84:24 125:16 162:2 164:20 167:13,14 distributes 120:8 distributing 29:22 distribution 29:10,13,16,18 30:6,14 31:5,18 32:25 51:9,17 52:1 70:14 139:1 140:23 141:11,12 164:13,15,16 165:11,15,16,19 166:25 167:25 168:2 170:24 divide 178:12 divided 120:19
--	---	---

dividing 70:14 DNN 162:2 document 81:14 183:13 documentation 83:13 documents 10:1 11:23,24 12:7 63:9, 11 111:22 112:1 dollars 64:6 dominant 104:1,4,8 105:2 Don 7:9,24 doubt 125:8 156:19 download 12:16 downloaded 11:25 dozen 8:14 draft 111:22 124:22 183:8,10 drafting 81:8 87:6 DRAM 107:19,22 108:4,7,14,17 153:17 154:5,6 drive 108:5 177:10 drives 171:23 172:4,10 177:2 182:2 driving 60:13 drones 133:8,9 Dropbox 12:4,8,15 15:5 dropped 148:15 DSP 44:18,19 76:18 93:11 97:18 102:3 103:13 105:14 DSPS 43:24 44:3,7,9,13 67:11 104:23,25 duly 7:4 dumb 159:17 dumbest 44:13 dumped 70:22 Duplicate 179:9 duties 6:18 dynamic 52:1 71:14,18 89:14 107:23 116:14 127:19 128:25 129:11,15,19 130:3,13,18,22,24 131:3,7,12,22 132:12,19,20 133:11,20 134:3 135:18 136:4,24 155:8 157:18 dynamically 51:6,9 52:4 92:9,12,16 138:25 139:2 140:23 141:2	E e.g. 88:9 Earl 7:9,24 earlier 35:12 83:17 125:12 137:21 147:23 early 10:8 42:8 43:25 65:10 easier 12:10,11 15:5 154:6,12 173:1, 3,5 easily 83:8 117:8 149:21 easy 79:8,11 173:23 edges 99:11 educated 147:18 education 67:7 effects 153:7 efficiency 64:1 113:18,22 114:1,5 180:23 efficient 69:3 71:23 72:3 85:14 86:6 93:4,7 126:5,14 162:2 efficiently 23:6 150:11 164:20 effort 84:24 elaborate 180:25 181:2 electricity 32:12 electronic 171:24 electronics 172:14 element 139:1 elements 30:22 62:6 94:21 95:18 96:25 97:21 144:3 Elimination 179:10 else's 11:4 121:15 embedded 68:16 emphasizing 53:24 117:25 empty 47:13 enable 85:14 86:2,6 178:22 enables 71:15 encompasses 139:2 140:1 encryption 93:12 end 112:5 131:22 150:5 172:8 ended 172:2	ends 28:2 energy 122:13 England 184:24 enormous 71:24 72:23 73:10 entail 124:1 entire 75:16 81:14 111:15 179:6,7 entitled 180:3 entity 50:9 131:5 environments 168:25 170:16 171:1, 6 equation 106:15 118:15 120:3,9 equations 119:4 equipment 62:1 erasure 69:22 error 70:8 error-correcting 69:23 escapes 42:3 essentially 141:7 152:20 et al 170:5 event 115:2 evidence 155:15 EX1004 88:9 Exabyte 175:1 exact 146:17 165:25 EXAMINATION 7:18 examined 7:4 examples 56:2 65:24 79:22 132:10 133:18 154:17,19 Exascale 143:22 144:6 exception 163:18 exclude 37:4,7 40:7,8 Excuse 62:18 executable 94:9 execute 153:5 executed 25:17,22 59:21 executing 26:1 54:18 95:7 122:6 execution 89:6,9 91:15,16 119:17 120:14,25 126:5 155:19 158:22 159:15
--	---	--

exhibit 12:23,24,25 13:2,9,10,11,12, 17,18,19,20 14:3,4,5,6,11,12,13,14, 20,22,23 15:18,19 16:6 40:11 56:12, 18,22 59:4 60:7 61:10 62:14 71:21 78:19,20 82:19 84:15 88:1,16 91:11, 13 92:24 107:13 116:9,15,16,17,19 124:12 125:18,19,20 127:11 129:17, 18 137:7 138:20 139:17,19 140:17 141:16 143:15 145:14 155:1,3,9,11, 21 157:25 158:1,8 159:9 160:23 162:3,6,8 168:23 169:2,19 171:25 172:5 174:25 175:2 179:9,11 183:13, 14,16 exhibits 11:6,8,22 12:2 exist 165:20 177:11 existed 42:12 165:12 177:14,17 existence 38:11 132:20 existing 38:5 101:9 114:19 123:24 expect 121:1,10,21 122:6,8 149:24 expected 181:25 experience 67:13 68:7 experiment 158:5,19,22 159:4 182:13,25 experimental 124:18 125:1 155:18 experimentalists 124:10,14,17 157:9 experimentation 124:2,21 125:14 157:2,6,20,24 158:3,11 174:20 182:9, 12 experimenting 158:17 experiments 125:1,4 157:17 183:2 expert 60:17 184:16 explain 165:22 explained 137:21 explaining 31:1 explanation 29:25 explicitly 84:23 92:22 181:2 explore 64:10 exponential 70:14 166:1 exposed 125:14 express 81:1,15 84:9 expressed 84:10 extensive 72:17	extent 161:21 F F-O-U-R 57:12 F-R-E-E-B-S-D 74:22 face 165:7 faced 65:24 fact 83:16 114:2 159:16 167:18 182:1 factor 104:4 119:19 120:2,4,7 121:9 159:16 factors 104:1 125:10,13 factory 72:10 fail 71:18 170:1 fails 70:8 — failure 69:15 70:13 168:24 171:12, 16,20,21 172:3,9,12,24 173:1,22,23 178:20 fair 167:23 169:16 179:19 fairly 18:2 19:8 35:4 66:17 76:17 157:13 171:15 falsified 157:10 familiar 25:14 63:17 67:10 90:13 113:1 148:7 175:9 176:18 179:17 fancy 172:3 farmed 59:25 96:7 farming 60:1,3 fashionable 99:20 fast 107:24 108:8 112:8,13 148:6 158:23,25 faster 33:24 35:11 37:12 38:15,24 43:20 89:15 93:11 106:4,17 122:13 150:5 fastest 64:18 fault 69:2,6,8,13 70:5 71:2,5,12,15,17 favor 181:3 feature 103:9 feedback 89:12 feeding 151:16 152:17 feel 11:25 12:10 96:16 fell 103:23	field 60:17 63:7 Figure 80:11,14,15,19 159:13 Figures 90:6,9 file 111:7 144:20 145:8,11 174:14 175:20 176:23 177:18 179:10 180:10, 13,14,15,17,23 182:3 filed 11:23 Files 180:16 filled-in 88:7 final 111:25 174:14,17,19 finally 90:17 find 16:7 123:9,10,23,24 160:19 172:13 177:2 fine 58:17 163:14 fingerprint 181:3 182:16 fingerprints 182:21 finish 150:3,6,9,15,19 164:7 finished 28:17 150:2 fit 124:18 132:3,4 136:22 153:9,10 178:5,8,11 fits 132:5 fixed 110:19 177:16 flexibility 100:8 flexible 17:12 34:24 floating 38:25 39:4,5 41:25 42:1,2,17 122:14 floors 72:10 flow 161:7 163:18,22 fly 147:15 focus 67:15 68:14 168:6,7 focused 89:4 167:24 follow 84:18 93:3 fond 175:18 form 10:14 16:13 17:22 18:21 19:7, 19 20:7,16,23 21:15 22:1,7,14 23:3, 10 24:19 26:4,15,22 27:13 28:4,22 29:3,14 30:25 31:20 33:5,19 34:8 35:9,24 36:15 37:2,9,16 38:7,13 39:12,23 41:4,22 42:14 43:1,12 44:6, 11 45:13,17 46:11 47:2,11,20 48:3,18 49:17 50:7 51:19 52:3 53:18 54:1,25 55:11 56:6 57:2,25 59:14 60:25 63:1, 19 64:3 66:1,15 67:21 71:3,13 73:14,
---	---	--

20 74:2 79:6 80:13 81:4,10,18 82:13, 25 83:12 84:1,7,14 85:6,16 86:16 87:2,7 88:14 91:23 94:6,24 95:4 96:21 97:9 98:20 100:5,20 101:7,14 103:7,16,21 104:6,21 105:6 106:11, 24 108:12,23 109:8,15,24 110:15 111:1 112:18 113:19 114:16 118:2,9, 20,24 120:16 121:3,12,23 122:21 123:6,15 124:15,24,25 125:6 126:9, 16 128:8,11 129:3 131:1 132:16 133:14,25 134:4,5 135:9,21 136:8 137:3 139:16 140:2 141:10 142:2,21 143:7 144:24 145:19 146:2 147:7 148:14,21 149:20 150:13 151:2 152:2,13 153:3,23 154:9,18 156:22 157:4 158:4,13 159:12 160:3,10,17 161:10 162:16,22 163:12 164:14 165:14 168:1 170:18 171:7,19 172:19 173:9 179:3 181:8 182:10 183:4,9,25 184:5,10 formatted 17:10 formed 10:19 88:22 forming 10:13 forms 41:6,10 formula 119:9,10,12 forward 90:25 found 11:7 110:19 147:15 150:20 157:18 foundation 169:3,14 Fourier 57:12 104:25 FPGA 93:11 FPGAS 42:9 67:11 fraction 105:12 119:22,23 120:5,18 frame 18:3 67:7,9 130:4,8,10 133:21 135:8 Frank 7:1 10:2 free 11:25 12:10 69:18 Freebsd 74:21 frequently 77:14 113:5,7 Friday 173:18 Friday-afternoon 173:13 front 114:3 184:17 FS 176:22 full 7:8,23 19:25 34:13 85:20 122:17 127:19 130:1 152:15	fun 114:11 173:10 175:23 function 136:17 functioned 160:8 functions 44:22 45:4 fundamental 122:20 123:5,10,13, 17,22 124:9 funny 91:25 fuss 58:2 future 180:3 181:21,23 G GA 159:1 gain 86:1 gathered 10:22 gathering 128:16 gathers 53:4 gave 43:4 71:25 131:17 133:18 172:23 Gee 147:17 general 22:22 36:6 37:18 38:21 45:8 53:1 121:5 142:24 146:14 165:16 generally 125:13 150:25 152:3 generated 133:13,24 134:1,3,7 Generating 48:8 generation 36:16,24 generations 34:14 generic 18:2 genetic 99:22 123:11 125:17 134:16 146:19,25 147:9,20 157:15 159:2 German 63:21 Germany 63:14 get all 11:5 gigabyte 176:24 177:18 178:14 Git 83:6 give 8:22 10:18 12:14 24:2,20 26:24 34:22 41:15,19 42:11 64:11 65:23 98:19 100:13 104:10 114:25 128:9 129:20 130:15 142:25 146:14,15 182:21 giving 8:16 79:22 82:6 181:22	global 150:17,20 goal 84:10 123:8 173:5 good 6:4 7:20 54:14 56:3 63:15 77:7 87:10,12 99:3,4,18 105:16 111:12 138:5 146:14 153:11 157:24 159:17 166:7,8 Google 11:12 27:6 75:14 163:22 Google's 75:14 governed 7:16 government 63:21 GPU 36:13 40:5,7,8 41:5,13 83:16 97:18 162:3 163:23 164:4,21 GPUS 40:23 41:13 162:14,20,24 163:4 grad 77:19 grade 176:20,21 grammar 58:2 92:12 granularity 25:2 75:12 graph 99:10,11,12,14 138:2 177:25 graph-coloring 72:20,22 graphical 40:24 graphics 36:22 graphs 99:22 great 31:24 58:16 greedy 146:10,11 158:23 159:1,18, 23 Greek 18:18 grounds 169:3 group 57:4,7 112:8 grouped 97:16 grouping 57:1,14,17,19 GSA 159:2 guess 10:15 12:5 124:9 132:1 147:18 guessing 133:15 guy 15:4 172:1 181:18 guys 63:14 91:25 172:22 H H-A-D-O-O-P 27:9
---	--	---

H-I-N-E-M-A-N 179:25 habitat 24:3 Hadoop 27:6,9 75:14 78:13 84:19 half 77:17 halfway 68:2 116:1 hand 37:24 handed 111:20 112:5 handled 6:15 122:12 131:4 handling 109:13 happen 92:18 132:7 149:7 happened 70:24 185:11 happy 20:10 34:17 36:1 74:3 hard 21:8 75:4 108:4 122:15 171:23 172:4 177:2,10 182:1 harder 173:24 hardware 50:9 88:18,23 105:20 harsh 77:9 hate 146:9 HC 131:21 132:2 134:20,22 156:15 161:9,20,22 HCR 156:8 head 63:3 headset 92:1 heap 94:14 hear 91:25 140:6,7 heard 9:23 Heather 6:11 heavily 114:19 Heim 6:23 helped 184:1 helper 53:1,6 helps 46:6 182:17 heterogeneity 119:19 120:2,4,7 121:9 162:25 heterogeneous 14:3 16:12,15,20 17:1 20:6,11,14,20,25 21:13,18 34:13 53:21 61:4,8 65:9 68:21,23 77:25 78:4,15,25 83:16 100:11 102:4 116:15 119:19 125:17 155:8 162:3, 14,20,24,25 163:4,5 164:21 167:17, 18 168:24 170:15 171:1,6,13 172:16 173:2,7 178:6,9,22	heuristic 145:17,18,24 146:5,6,9,11, 13,16,18,22,23 147:5,9,14,18 148:6 heuristically-based 131:23 heuristics 145:22 146:19,20 147:4 158:23,25 Hey 11:10 98:22 high 64:1 77:5 80:20,24 119:16 172:7,10 182:21 high-level 112:20 114:23 115:1,6 155:25 high-performance 27:5 64:7,15 65:8 70:11 97:13 higher 75:9 174:9 highlight 155:10 highlighting 183:20 highly 94:20 95:20,22 96:24 97:21 143:21,24 144:2 hill 146:18 Hineman 111:17 179:24 hint 90:1 hints 89:24 181:22 history 19:10 hive 138:17 Hold 12:3 145:15 148:15 158:2 162:17 home 173:18 homogeneity 173:12 homogeneous 39:16 42:18 148:18 163:23 171:5 172:17,23 173:2,7,14 174:2,6,8 179:1 homogenous 162:24 honestly 8:3 154:10 184:6 honey 138:17 hope 89:15 hour 8:18 32:13 58:17 87:13 hours 9:12,19 11:11 HP 148:25 hugely 159:15 human 166:20 hundred 153:17 155:5 178:14,16 hundreds 63:15	hundreds/thousands 84:21 hypothesis 157:10 182:14 hypothetical 46:18 129:20 I i.e. 40:23,24 I/world 72:7 i7 81:23 i9 81:23 IBM 42:5 65:2,6,7 111:3,11,20 112:4 180:4 181:17 IBM's 65:21 111:13 idea 118:3 119:13 177:13,15,20 182:16 ideal 174:3 ideas 113:10,12 identical 39:21 identification 12:25 13:11,19 14:5, 13,22 116:17 125:20 162:8 169:19 175:2 179:11 183:16 identified 119:18 identify 135:20 160:1 identifying 89:17 idle 151:8,22 152:3,5 idling 151:25 152:8 ignore 45:24 46:1 II 72:7 III 18:25 19:23 IIIS 43:18 illness 8:16 illusion 54:8 illustrates 183:21 image 105:5,11 130:4,7,8,10 133:6, 21 134:10 135:8 imagination 106:19 imagine 142:22 implement 82:23 83:3 100:19 103:20 110:8 113:11 128:5 157:2 implementation 73:25 103:9 111:25 125:11 149:15 156:13,20
--	---	--

implementations 109:23 110:2 114:13 116:6 implemented 63:18 88:8,13 127:24 155:15,18 156:13,17,18 159:5 implementing 85:21 106:10 imply 59:18 important 109:10 improve 159:11 improvement 84:22 improvements 73:19 include 26:3 36:13 75:4 79:15 94:4, 12 95:12 included 33:17 44:3,9 55:21 78:5 139:11 167:19,21 includes 33:4 72:11 155:25 including 73:5 78:3 incomplete 150:20 incorporated 48:21 increase 97:24 118:8 independently 25:13 index 77:6 indicating 119:19 indication 55:19 industry 67:16,19 68:7 infant 172:7 inform 94:17 information 48:9,16,22 49:3,9 51:21 53:4 57:5,6 89:12 94:15 98:25 99:24 108:19 127:21 128:2,16,17,19 129:1, 11 131:8,13 135:25 139:22,24 141:25 142:1,5,14,19 143:5 informed 48:19 infrastructure 85:1 101:12 infringes 20:21 initial 98:24 99:2,4 insane 62:19 instance 90:13 101:16 105:20 instances 28:13 instantiate 131:25 instantiates 132:2,5 instruction 148:8,23 149:3,6,10	150:2 instructions 18:8,10 36:23 149:5 instrumentation 128:17 integrated 42:5 Intel 17:25 18:2,12,19 19:15 36:8,10, 12 39:18 78:5 79:2,21 80:1 81:23,25 82:5 148:25 intelligent 156:2,6,17 157:3 intended 22:21 23:5 inter 8:11 14:19 interact 21:7 interconnect 22:20 interconnection 59:18 interest 130:4 133:20 interesting 17:15 19:12 22:16 65:5 77:18 94:7 123:2 157:19 interestingly 66:8 101:17 internal 180:7 interpret 47:10 81:21 interpretation 45:16 interprocessor 117:22 intractable 146:17 166:2 introduce 11:22 12:22 13:8,16,24 14:10,18 116:13 125:15 143:22 162:1 168:23 174:25 179:9 183:12 introduced 12:19 introducing 176:9 invalid 185:5,9 invalidates 20:22 invent 144:25 involved 21:10 106:21 involves 73:3 168:20 IOS 156:2,12 IP 25:24 iphone 106:3 IPR 11:24 12:24 13:10,18 14:4,12,21 15:19 16:6 40:11 56:12,22 59:5 60:8 61:10 62:14 71:22 78:20 82:20 87:23 88:1,16 91:13 92:24 107:13 116:16 124:13 125:19 127:12 129:18 137:7 138:20 139:19 141:16 145:14 155:3, 21 158:1,9 160:23 162:6 183:14	IPR2020-0318 6:9 IPR2025-00318 7:15 14:20 IPRS 8:12 irrelevant 109:2 isolation 37:18 54:10 isomorphic 61:6 issue 15:6 62:1 105:13 122:20 151:14,16 153:10 165:3,17 issues 11:3 49:19 66:3,4,5 73:4 154:15 issuing 22:5 iteration 26:14,17 27:2,3,12,23 28:2, 3,21 29:2,8 30:2 32:21 89:11,13,15 92:22 99:23 100:25 101:2 131:23 134:19 137:16 iterations 26:20 89:6,10,20 91:5,16, 21,22 92:15 94:23 95:1,3,8 iterative 14:1 90:11 109:4 116:14 155:8 iteratively 90:12 92:19 its's 52:5
J		
JAVA 25:25 Jay 133:16 job 72:7 jobs 24:23 joint 17:12 34:25 148:25 journal 76:25 77:1,10,13,14 jumps 60:14 junior 69:12 justify 149:21		
K		
Kambatla 14:15 77:24 78:3,10,13, 14,17,19 79:12 81:2,6,17 82:3 83:7, 15,19 84:3,11,16,22 85:8,14,23 86:2, 14,23 87:1,3 kernel 111:7 156:7,8,15 157:21 161:9,20,22 kernel-level 75:7,8		

kilowatt 32:13 kind 23:6 33:25 34:1 38:20,21 54:19 70:6 71:16 72:14 80:8 103:24 104:2 106:5 158:5 159:4 172:22 kinds 11:11 79:1,14,20 121:18 Klarquist 6:25 7:1 knapsack 72:19 knowledge 157:22 Kroger's 175:21 Kwok 154:20,22,25	legalese 33:8 lets 86:9 letter 18:18 level 25:1 30:13 31:10,13 75:9 119:16 143:21 levels 30:10 31:7 121:17 license 6:13 licensed 6:12 life 60:22 172:7 light 98:22 lighting 130:7 131:18 136:15,17 likes 113:4 Limbacher 6:22 7:13,19,21 8:7,10 10:17 12:22 13:2,8,12,16,20,24 14:6, 10,14,18,23 15:9,16 16:17 17:23,24 18:15,17 19:4,14 20:3,12,19 21:4,23 22:4,11,23 23:8,14 26:9,20,24 27:11, 16 28:1,19,24 29:6,23 31:6,22 32:2,9 33:9 34:4,20 35:14 36:8,17 37:6,13 38:4,10 39:7,17,25 40:1 41:7,24 42:21 43:2,4,21 44:7,16,24 45:15,21 46:16 47:6,13,24 48:6 49:2,23 50:11 51:8,25 52:8 53:22 54:4 55:3 57:14 58:10,18,24 60:5 61:3 62:4 63:8,22 64:11 65:3,23 66:12,22 67:23 68:6 70:25 71:6,20 73:9,18,24 75:6,21 79:9,23 80:15,22 81:7,13,21 82:16 83:2,9,23 84:4,11,17 85:9,19 86:13, 22 87:5,10,14,21 88:6,15 91:3,8 92:2, 5 94:18 95:2,11 96:9,22 97:12 100:3, 13,23 101:11,21 103:10,12,19 104:2, 10 105:4,7,15 106:21 107:12 108:16 109:1,13,18 110:11,23 111:13 112:21 113:24 114:17 115:11,18 116:4,18 117:20 118:5,16,22 119:6 121:9,20 122:2,16,22 123:4,12,19 124:12,20 125:3,15,22 126:12 127:3 128:9,14 129:6 131:6 132:21 133:19 134:2,8 135:6,12,24 136:23 137:5 138:5,12 139:18 140:7 141:14 142:11,25 143:14 144:4 145:20 146:21 148:7 149:12 150:9,24 151:6 152:8,19 153:19 154:3,16,25 155:3,14 157:1,7 158:7,15 159:8,19,24 160:6,14,22 161:12 162:9,19 163:2,14 164:12,18 165:21 166:14,17 167:23 168:5 169:4,5,12,18,20 170:21 171:16 172:15,25 174:1 175:6 176:16 179:8, 14 181:2,14 182:23 183:7,12,17 184:3,8,15 185:24 limit 126:7 153:19	limitation 32:19 33:3,17 34:6 43:22 44:4,10 45:9,12,22 46:9,19,20,21,25 47:7 48:7 103:6,8 104:19 141:6,7 limitations 105:7 184:9 limited 106:17 154:8 lines 20:13 32:20 43:23 45:10,23 46:3,8,21 48:8 51:13 55:4 59:6 89:4 97:14 98:12 111:21 137:11 140:22 141:1 link 12:4 Linux 19:24 74:4,6,9 75:2 111:7,9 113:1,4,8,9,13 114:13,15 156:9 177:18 Lippert 12:17 13:21 17:3 24:7 48:20 53:5 55:13,20 56:11,13,14,17,19 75:25 76:1 85:13 86:19 88:9,12,16 89:20,21,22 91:17,20,21,24 92:11 93:3,14 94:19 95:19,23 98:5,8,14,21, 23 99:2 100:12 106:8 127:22 142:4,8, 24 143:5,11,15 144:14 Lippert's 99:4,18 Lippert-budenske 85:22 86:14 127:25 Lippert-budenske-kambatla 85:21 listed 157:16 175:21 176:12 listening 15:4 literature 71:24 72:23 73:10 Livermore 64:5,7,12,14,16 load 58:3,9,13 94:12 105:13 134:24 loaded 107:2 108:4 loader 94:11 loading 57:20 58:11,12 107:9 local 11:25 localities 182:22 locality 153:13,15 locally 11:25 located 144:20 log 12:16 15:4 119:24 logically 59:16 login 148:5 long 6:7 7:3,9,20,24 8:19 13:9 15:17 32:9 58:24 73:3 87:9 97:15 105:16 115:1,18 136:3 138:13 147:12 148:8
L		
lab 65:22 78:4,8,10 175:21 labeled 119:9 laboratories 64:15 labs 65:18 66:21 lack 63:10 laid 135:14 language 25:18 55:24,25 languages 26:1 98:15 large 11:6 70:12 84:20 104:15,17 173:25 177:21 largely 114:20 larger 59:23 173:22 184:12 largest 168:19 late 167:9 173:17 latency 103:25 104:8,12,13 106:3 109:4 117:13 laughed 32:13 law 63:4 173:24 layer 176:9,13,17,18 177:11 178:4 layout 177:25 178:1 layouts 99:21 167:4,6 leaders 64:15 leap 96:23 97:1 leases 112:7,10,13,14 leave 128:14 151:23 left 119:9 170:3 181:17 legal 22:2 27:20		

149:6 152:6 154:7,12 166:18 169:5, 20 172:7 183:7,17 185:24 Long's 68:9 long-felt 63:5 long-term 72:24 73:1 longer 73:18 looked 63:13 74:23 80:24 112:2 113:15 loop 99:1 loops 152:5 153:8 loose 24:21 100:15 Los 64:5,7,13,14,16 lost 41:8 lot 8:12 21:5 55:15 77:4 104:24 105:1,17 106:4 109:20 125:9 136:20 150:5 151:9 164:5 167:9,10 182:24 184:14 lots 86:10 143:11 loud 141:18 low-key 168:12 low-level 112:20,22,23 lower-powered 152:7 lunch 115:15 Luster 177:19	181:6 185:14 makes 12:11 25:10 83:14 84:24 118:16,23 119:1 123:5 126:21 145:1 165:22 166:2 170:25 making 12:20 72:12 96:22 140:8 management 82:23 83:4 manager 46:4,5 49:5,13,16,20 50:2, 6,8,12,14,16,19,23 51:1,5,9,21,22,24 59:8 85:12,13,23 88:8,13,19,22 111:11 127:23 128:20 138:25 139:2, 14,21,23,24 140:11,12,13,20,23 141:2 managers 45:19,20 manages 49:25 176:11 managing 152:19 manipulates 26:6 map 27:1,6 142:17 177:8 mapping 109:3,7 116:14 125:16 129:10 131:24,25 137:17 155:8 176:11 mappings 123:3,9 134:17 147:13 156:2 marked 12:25 13:11,19 14:5,13,22 116:17 125:20 162:8 169:19 175:2 179:11 183:16 market 41:16,20 110:24 Martin 163:19,21 mascots 175:14 matched 136:21 matching 181:4 materials 10:9,12,18,19,22 11:5 math 70:20 105:1 183:6 mathematics 72:6 matrix 105:1 matter 6:8 9:13 81:8 182:2 max 159:1 maximizing 173:5 maximum 150:11,16 Maxwell 52:22 meaning 16:1,11,18 22:5,12 23:1,24 24:10,14,17 26:10,13,16 29:7,12 31:18 35:17,18,21 48:15,23 49:9,15 50:5,19 51:1 52:24 55:9,18 56:5,25 57:23	means 10:16 27:21 29:20 33:7 47:18,22,23 51:20 52:5,6 57:18 59:16 70:18,19 72:3 95:8,23 99:13 102:16 105:1 107:22,23 114:6 117:16 122:4 133:6 157:9 164:9,10 172:6 175:3 meant 20:9 31:2 measure 57:20 58:3,7,11,12,13 measured 58:4 155:19 measures 58:8 mechanical 171:23 mechanism 98:23,24 99:19 medication 8:15 medium 104:24 meet 44:4,10,14,16 megabyte 176:24 megapixel 105:11 memories 70:7 108:6 175:18 memory 22:19 36:4 40:23 41:12 43:14 54:6,9 58:5 66:3 70:22 78:6 80:18 94:12 107:23 151:14 152:16, 19,23,25 154:14 mention 46:4 71:5 142:1 182:4 mentioned 23:14 30:22 43:24 mentions 18:12 Mercury 103:2 met 10:2 45:12 46:20 47:1 metadata 94:8,11,16 144:20 145:9, 11 176:11 method 13:25 21:1 70:10 metric 57:19 123:25 metrics 58:5 83:19 microprocessor 80:1 88:23 Microsoft 6:8,25 middle 7:10 Mildly 179:4 military 101:18 million 111:21 175:19,22 177:2 millions 97:14 min 159:1 mind 15:7 27:20 33:14 48:5 149:22 minimize 151:7
M		
machine 72:12 119:18,21 121:19 machines 66:20 75:19 121:6,16 175:22 macos 145:10 made 24:23,24,25 84:4 94:2 97:15 98:11 143:21 149:19 153:8 173:17,20 180:16 magically 62:20 main 142:11 152:24 major 111:8 156:13,20 majority 167:24 185:2 make 15:5 27:10 29:17,25 35:11 37:11 52:18 62:11 69:17 80:9 84:9 87:4 89:23 99:25 102:11 104:7 115:4 118:10 119:1 131:11,23 132:1 140:13 145:20 150:10 153:20,24 154:6,12,14 163:3 174:6 178:5,8,17,18 180:8		

minor 82:22 83:2,24 84:5,12 minute 90:2 minutes 8:23 24:2 32:1 58:17 75:8 185:17 mischaracterization 86:18 mischaracterizing 38:8 misheard 68:11 misinterpreting 174:4 Missile 133:10 misstatement 113:21 modality 171:21 mode 91:6 171:12,17,20 173:22,23 178:20 model 28:9 44:18 116:24 118:13 119:16 121:15 124:18 170:25 171:11 172:24,25 173:20 174:3,6,10 177:4 modeling 118:11,12 119:3 172:2 models 34:14 modern 25:18 133:7 145:10 151:20 176:23 modes 172:4,9,13 modification 84:16 modifications 100:3 modify 112:25 module 88:8,13 modules 70:7 moment 24:20 34:22 54:3 money 182:24 monitor 131:21 132:2 134:20,22 monitoring 157:23 monitors 131:21 132:3 Monte 26:18 28:12 Monte-carlo 27:2 month 147:12 morning 6:4 7:20 mortality 172:7 Morton-park 7:2 motivation 96:17,18 Motorola 80:2 mouse 16:7 24:4 60:13,14 154:20	move 73:5 166:4 moved 108:9,11 113:17 181:18 moving 20:3 24:4 106:22 137:1 152:24 multi-processor 64:2 multiple 27:11 28:13 42:15 65:22 68:17 69:13 79:19,21 85:14 86:3,7, 15,19 95:16 97:1 114:4 126:4,15,25 149:5,16 mute 92:2 N named 172:1 names 7:10 national 64:14 65:18,22 175:20 native 24:3 nature 33:21 109:10 NDA 171:22 necessarily 10:23 25:22 26:23 40:5 needed 9:1 82:22 83:3 98:4,7,11 100:19 124:21 125:1 153:21 157:20 161:21 178:8 181:5 Netapp 172:2 network 60:25 Networking 104:12 networks 163:24 neural 163:24 nice 45:18 102:7,8 128:21 150:7 155:12 158:21 Nikolau 148:10,16 node 18:4,6,7,9,20,23 19:2,3,5,9,13, 16,25 21:21 22:13,15,17,24 23:2,4,7, 8,21,25 24:7,10,14 27:4 33:14,15,21, 22,23,24,25 34:1,2,3,5,11 35:3,7,8, 11,19,22 36:1,2,14 37:1,4,8,14,15,23 38:2,17 39:6,9 41:6,10,11,16,17,20, 21 42:13,19,20,23 43:9,20 45:5 46:14,24 47:5,16 53:23 54:17,18,21, 24 56:4 58:3,9 59:21,24 60:2 65:7 76:12,20 80:17,19,25 90:16 96:8 100:7,18 101:1,2 106:14,16,18,22,23 107:2,6,7,8 127:22,23,25 128:1 130:21 136:13,21 145:4 147:17 164:11 nodes 16:23,25 17:13,14,17 19:24	20:4,5,9 21:2,6,19,20 27:1 29:20,21, 24 30:5,6,7,8,9,12,13,15,16,17,18,20 31:4,11,14 32:23,24 33:4,18 34:7,15, 17,19,21 35:1,2 38:6,12,23 39:19 40:23 41:3 42:18 44:1,8 45:11 46:9, 13,19 47:8 48:11,12 49:21,22 50:10 53:13,14,16,17 59:9,10,12,13 61:19, 20 62:8 65:8 67:12 68:17 71:18 83:16 90:21 93:10 96:5 97:24 99:10 106:1 139:3,4 141:3,4 noise 91:25 140:9 nomenclature 65:6 non-sequitur 106:6 nonsensical 80:10 normal 96:4 notes 11:9 notice 8:23 noticing 6:21 notion 143:22 146:4 novelty 178:3 NP 165:20,22 166:1 NSF 123:18,19 NT 74:16 number 6:13 11:6 70:12,15 100:8 102:1 120:17,19 121:17 125:9 129:21 130:3,6 131:18 133:20 134:13 135:7 136:9,13 137:24 142:15,22 156:25 168:8,19 180:19 numbered 135:18 136:4 numbers 42:3 124:17 153:18 154:23,24 155:18 173:14,25 174:13 178:13 O Oak 64:17 oath 9:4,5 object 136:2 objection 10:14 16:13 17:22 18:21 19:7,19,20 20:7,16,23 21:15,16 22:1, 7,14 23:3,10 24:19 26:4,15,22 27:13, 18 28:4,22 29:3,14 30:25 31:20 33:5, 19 34:8 35:9,24 36:15 37:2,9,16 38:7, 13 39:12,23 41:4,22 42:14 43:1,12 44:6,11 45:13,17 46:11 47:2,11,20 48:3,18 49:17 50:7 51:19 52:3 53:18 54:1,25 55:11 57:2,25 59:14 63:1,19
---	---	---

64:3 66:1,15 67:21 71:3,13 73:14,20 74:2 79:6 80:13 81:4,10,18 82:13,25 83:5,12 84:1,7,14 85:6,16 86:16 87:2, 7 88:14 91:23 94:6,24 95:4 96:21 97:9 98:20 100:5,20 101:7,14 103:7, 16,21 104:6,21 105:6,9 106:11,24 108:12,23 109:8,15,24 110:15 111:1 112:18 113:19 114:16 118:2,9,20,24 120:16 121:3,12,23 122:21,24 123:6, 15 124:3,15,24 125:6 126:9,16 128:8, 11 129:3 131:1 132:16 133:14,25 134:5 135:9,21 136:8 137:3 139:16 140:2 141:10 142:2,21 143:7 144:24 145:19 146:1,2 147:7 148:14 149:20 150:13 151:2 152:2,13 153:3,23 154:9,18 156:22 157:4 158:4,13 159:12 160:3,10,17 161:10 162:16,22 163:12 164:14 165:14 168:1 169:2, 13,17 170:18 171:7,19 172:19 173:9 179:3 181:8 182:10 183:4,9,25 184:5, 10 objections 122:10 objects 130:3,5 133:20 136:10,14 obtain 176:12 obvious 66:14,16,17 109:23 110:2,7, 8,10 114:14 occur 172:9 offer 185:7 offered 35:12 Office 6:10 7:17 officer 6:17 official 86:2 offline 13:25 123:3,9 146:22,23 147:9,10,23 149:9 156:1 OLDB 158:25 one's 110:9 172:11 online 13:25 123:3,9 147:4,5,14,21 148:4,6 149:8,9 156:1 open 101:25 143:12 155:5 open-source 113:9 opens 21:5 operate 53:23 operates 54:15 operating 21:1 53:12 54:2,21,24 61:20 69:12 72:2 73:7 75:3 94:10 112:24 155:25 156:2,6,9,17 157:3 168:20	operation 149:4 161:20 operations 72:5 opinion 10:18,20,25 71:1 77:24 81:8 83:10,24 85:5 86:13 87:6 135:13 opinions 10:13 22:5 opportunistic 112:7,9,12,14 opposed 52:6 optimal 165:25 optimally 165:20 optimize 181:13 optimum 150:17,21,23 order 12:16 21:9 90:12 99:14,15 100:19 127:19 138:1 178:17,21 180:13,14 184:18 ordering 137:22,24 138:1 ordinary 16:1,11 22:12 23:1 24:9,13, 17 26:9,13,16 29:7,12 31:17 35:17, 18,21 48:15,23 49:8,15 50:5,18,25 52:24 55:9,18 56:5,25 57:23 Oregon 32:12 organization 176:10 original 89:21,22 110:13 112:6 113:8 outcome 125:3 overhead 106:21 107:10 119:24 overheads 125:10 overlap 178:19 183:22 overloaded 29:16 overview 98:19 owing 164:21 owner 6:23 7:21 185:1 owns 113:11 P P-H-I 18:16 P-O-S-I-T-A 67:3 p.m. 115:15,17 138:8,9,11 166:10,11, 13 185:20,21,23 Pacific 6:2,6 packed 149:5 Pages 161:1	pair 102:20 panel 65:14,17 paper 103:18 124:22,23,25 126:13, 24 127:1,4,7 156:14 158:3,12 160:5, 19 161:19 162:13 165:7 168:23 169:6,9,14,21 174:25 175:7,9,12,14 176:2 179:9,15,17,22,23 181:7,20 papers 63:15 112:15 166:22,24 167:2 168:3 175:13,15 paragraph 15:21,23 17:24 18:12 40:13,15,18,19 60:10,11,16 62:15,16, 22 63:9,22 66:23 67:18,25 68:1,9,24 71:22 75:21,25 76:4,22 77:21 78:21, 24 79:24 82:17,24 85:10 88:2,9,12 89:1 90:6,8 91:3,10,19,20 92:18,25 93:15 98:1 115:23 122:18,23 125:25 127:12 128:22 129:7 137:9 138:22 141:17 142:3 143:13 149:14 156:12 161:13,16 163:9 170:11 172:22 174:12,17,19 176:9 180:21 paragraphs 116:23,25 117:1 170:4, 7 174:2 parallel 14:3 27:4 28:12,13,14 102:5 119:23 120:5,13,18,24 parallelism 137:10,12,17,19 138:2 parallelization 119:24 parallelizeable 95:23 parameter 121:20 127:20 133:20 134:3 parameters 118:12 128:25 129:11, 15,19 130:3,13,18,24 131:3,4,7,12, 17,22 132:2,3,4,7,10,11,13,19,20 133:11,17 134:15,20 135:18 136:4,24 157:16,18,23 parenthesis 137:14,20 parity 178:1 part 41:6,10 59:23 76:4,8 84:19 90:15,17 91:1 95:7 96:19 97:2,3 108:3 111:8 120:6 128:13 145:11 150:1 151:11 152:18 Partec 6:8,23 7:21 partes 8:11 14:19 partial 44:22 45:3 99:14,15 137:22, 24,25 participated 65:20 parts 30:11,13 71:11 95:9 97:19 151:22,23
---	---	---

past 109:19 184:16 patent 6:9,23 7:17,21 10:10 12:17,23 13:3,17 14:11,19 16:5,22 17:3,19 18:22 19:5,18 23:20,24 25:5 32:19 43:22 48:20 56:18,21 57:21 59:3 61:9 62:6 71:2,9,11 139:18 140:16 143:10 168:15 184:20,23,25 185:1,4,5,8 patent's 17:20 patents 65:7 112:16 paths 102:15 Patriot 133:9 pattern 180:13,15 181:25 182:4 patterns 110:6 180:10,17,18,24 181:4 pause 44:13 136:2 Payne 6:23 PC 82:6 PDF 80:16 170:3 174:11 Pentium 18:25 19:23 36:17,24 37:3 39:2,18 43:5,18,19 76:18 80:1 people 18:8 113:8,10 146:5,6 157:11 181:22 percent 130:6 percentage 184:25 perform 23:5 36:6 37:18 48:24 104:19 152:12 158:3 performance 43:18 48:22 97:25 124:19 130:21 153:12 154:23 181:13 performed 27:5 90:10,11,14 123:12 158:11 performing 30:21 59:20 93:23 105:4 132:24 182:7 periods 8:19 permitting 17:12 34:25 person 18:24 67:6 90:6 Persons 90:9 petition 14:19,21,24 183:20 184:2,4,9 Petitioner 6:25 Ph.d. 7:3 148:16 169:24 Phi 17:4,9,15 18:13,15,18,19,24,25 19:11,15,23 35:5,13,15 53:20 76:17 96:2	photograph 130:1 phrase 48:4 146:9 phrasing 56:10 physical 50:9 75:19 177:21 physically 59:17 107:19 154:5,6 physics 168:8 pick 114:7 146:11,12 picking 19:8 pixels 130:6 place 52:16 64:17 92:9,12 99:14 144:12 153:21 placement 94:17 168:21 places 125:9 144:19,22 plain 16:1,11 22:12 23:1 24:9,13,17 26:9,13,16 29:7,12 31:17 35:16,18,20 48:15,23 49:8,15 50:5,18,25 52:24 55:9,18 56:5,25 57:23 planned 110:18 174:20 Platforms 14:3 plays 106:13 pleasant 8:20 plug 42:3 plurality 21:1,2,19 27:20 30:3,4,5,15 32:22,23,24 33:7,12 46:23 47:8,14 48:10,11,12,17 53:12,14 59:9,10 62:8 139:3,4 141:3,4 point 16:20 22:2 38:25 39:4,5 41:25 42:1,2,17,22 60:22 68:23 73:15 88:7 91:14 102:10 122:14 138:6 152:16 160:15 161:12 166:4 172:22 policies 180:25 181:3,5,15 policy 180:22 poor 172:24 poorly 153:16 population 172:16,18,23 173:2,7,8 174:6 portion 24:21 30:4,5 32:23,24 47:8,10,14,18,22,25 48:1,4,10,11 53:12,13 59:8,9 120:13,24 portions 122:12 Portland 111:21 112:5,9 POSITA 47:14 48:25 67:3,5 68:15 69:1 85:20 86:1 93:2 116:5 127:24	position 92:14 96:9 106:9 possession 54:9 power 82:6 151:15 152:7 powered 174:9 Powerpc 65:6 79:2 80:2 102:2,4 103:15 106:3 practical 124:11 pragma 55:14,15 56:16,17,18 145:8 pragmas 55:13,21,22 56:15 142:10 145:1 praise 63:6 preamble 21:24 22:3,5,10 precious 151:18 predictable 125:4,8 183:2,5 predicted 125:11 prediction 168:24 172:24 173:1 predictive 174:3 predictor 169:25 preferable 172:21 preparation 127:7 prepare 9:20,22 prepared 10:4 preparing 9:17 present 40:6 166:20 president's 65:18 pretty 77:7,8 110:18 112:2,3 140:14 141:9,13 146:14 169:22 183:5 prevent 8:16 previous 33:16 66:20 89:12 115:10 previously 11:23 43:5,23 price 32:12 primarily 22:21 69:19 161:8 primary 111:18 principles 69:2,6,8 71:23 74:18,24 93:4,7 prior 11:14 81:7 87:5 116:2,4 168:13 priority 74:22 141:25 142:5,9,13 probability 172:11 probing 181:3
---	---	--

problem 8:21 56:20 72:19,20,22 123:24 145:6 150:20 152:1,11,16,17, 21 163:25 164:9,13,15,17,22 165:2,6, 22 166:1 178:21 problems 66:13 123:21 163:10,11 165:12,15,16 Procedure 6:16 proceeding 11:15 32:15 58:25 115:20 138:14 166:19 proceedings 8:11 10:10 process 53:1,6 54:8,15,17,19 60:24 70:9 73:1,3,5 102:2 120:17 149:3 processed 26:21,25 27:3,12 107:3, 4,25 108:11 130:19 processes 54:10 75:17 101:13 139:22 processing 17:13,14 30:3 32:22 34:25 35:2 36:23 40:24 42:7 48:9,16 93:4,7 104:17 109:4 126:14 129:2,12 130:11 131:8,14 150:25 152:9,11,21 153:19 154:7 processor 16:16 17:16,25 18:3,13, 19,25 19:9,15 22:16,19 25:18 35:5 36:10,12,16,25 37:14,18,25 38:1,2, 15,16,17,18,22,24 39:8,10,14 41:15, 19 42:6,11,22 43:5,11 45:2 54:6 60:24 65:7 66:7 70:16,19,22 73:4,5,6 79:2,11,18,25 80:3,4,7,8,9,17 81:20, 25 82:2,6,15 93:13 96:2 100:4 102:4 108:10 119:21 121:11 122:5 126:21 135:11,19 136:5 137:2 145:2 148:19, 22 150:25 151:13,16,17,18,20,21 152:6,10,16 153:20 processors 17:21,23 37:24 38:5,11, 15 39:15,19,22 40:4 42:7,24 43:6,10 66:6 70:7,12,13,15,17 78:6 79:1,14, 20,21 81:2,16 82:7 95:21 100:9 101:23 102:6,14,18,20,22,23 103:13, 15 105:21,22 107:19,23 108:2,6,8 117:5,23,25 120:20 121:2,22 122:1,9, 13 126:4,7,15,25 134:18 149:16 150:11 151:4,5,8,25 152:6,15,17 153:22 154:4,8,11 produced 6:14 products 110:24 professors 63:14 program 66:18 97:14 programmability 66:11 programming 98:15	programs 66:6,11 project 64:13 65:12 84:20 110:12 113:10 115:4 148:25 156:24 175:23 177:24 promised 174:21 pronounce 52:11,19 pronounced 52:16 pronunciation 52:14 proper 67:19 properly 110:20 178:17 property 35:5 103:1 117:22 proposed 106:7 proposing 107:1 prove 124:5 181:11 proven 145:5 provide 96:17 101:11,18 provided 55:6,10 56:8 143:6 144:11, 13,14,23 PS3 66:8 Publication 13:17 14:11 publications 73:22 168:13,16 publish 77:4,5,11,12 published 77:10 154:20 165:8 166:22,24 pull 12:11 120:7 140:16 pulled 114:2 Pure 181:17 purely 143:24 purported 84:22 purpose 22:22 34:1 36:6,22 37:19 45:8 93:13 purposes 60:2 181:7 pursuant 6:15 push 81:19 put 101:1 138:17 152:4 Pytorch 163:17	quantities 122:15 quantity 121:8 quarterly 77:17 question 8:24 9:8 21:9,22 23:15 33:9,16 35:6 38:10 44:23 46:7 68:11 83:1,8,17,18,21 94:7 107:15 109:3 123:2 127:17 129:10 142:11 159:6 171:9 173:11 questions 12:12 21:5 108:24 124:10 174:23 185:25 queue 114:3,8,9 queuing 117:17 quick 110:7 138:3 185:14 quicker 154:14 quickly 66:4 148:5 quit 151:10 quote 95:15
R		
R-I-S-C 103:15 Ralph 111:19 ran 158:21 183:3 Randal 111:17 Randall 111:17 random 153:10 180:19 randomized 146:19 range 173:15 rapidly 23:6 89:18 109:11 rare 174:7 rate 70:13 107:7 121:16 172:8 rating 55:6,10,18,19 56:8 141:22,23, 24 142:8,12,16,20 143:2 144:5,7,19, 23 reach 62:25 124:22 reached 28:15,17 reaching 63:9 read 10:1 12:18 15:22 29:15 32:19 33:1 40:19 43:22 48:7,13 51:11 55:2, 20 59:15 62:17 63:23 66:25 69:4 71:14 75:22 76:23 77:14,19,22 82:18 85:9 88:3 89:2 93:5 98:2 102:12 116:23 118:4 119:13 122:17 125:25 127:1 128:3,23 129:23 137:9 141:6,7,		
Q		
qualitative 156:23		

17,19 143:19 144:5 156:4 160:4 162:9 163:13 170:7,11,22 171:2 174:1,11,16 180:3 reading 23:5,16,19 47:15 95:15 102:9 127:14 157:13 ready 67:1 80:23 real 104:13 155:12 165:2 173:19 real-time 74:25 75:1,2,3,4 105:5 real-world 170:23 realize 29:16 reallocate 94:14 reallocation 51:22,24 rearchitect 110:22 reason 8:15 19:22 34:12 146:16 154:22 156:19 reasonable 132:1 133:17 148:3 reasoning 123:8 reassigned 89:18 reassigning 105:25 reassignment 71:15,19 recall 8:3 20:17 53:4 55:13 63:2 82:14 101:17 113:17 169:21 received 51:21 receives 139:21 receiving 98:25 recently 157:25 recess 15:13 32:6 58:21 61:25 87:18 115:15 138:9 166:11 185:21 recitation 20:14 recite 50:15 recited 21:12 24:18 29:10 52:1,25 61:13 122:22 129:12 139:1 recites 16:22 17:25 49:3,12 50:22 53:11 54:20 reciting 16:25 59:7 recognition 14:2 105:5 132:23,25 133:5 recognized 93:3 recognizes 94:19 recollection 74:13 record 6:5,20 7:8,23 15:8,9,11,14 18:15 32:3,4,7 58:18,19,22 61:23	62:2 87:14,16,19 115:12,13,16 129:17 138:6,7,10 163:3 165:21 166:9,12 169:12 185:16,19,22 recursive 44:22 45:3 Red 64:25 redesign 110:12 redistribution 89:14 redo 113:5,6 reduce 27:6 38:18 89:5,9 91:15,16 redundancy 69:14,19,24 70:6 Reese 111:17 refer 11:25 13:2,12,20 14:6,14,23 46:6 52:8 75:24 76:4 78:8 82:24 83:4 130:18 reference 55:21 181:16 references 28:10 116:4 referencing 120:22 referred 64:12 133:12 146:23 163:10 referring 12:6 13:5,14,21 14:7,15,25 17:25 23:19 45:9,22 46:2,20 51:18 62:7 75:6 80:4,11 86:22,23 88:15,19 90:8 91:4,20 92:10 93:8 96:15 97:3,4 100:4 129:24 130:14 132:22 143:14 155:1 160:8 refers 47:7 55:6 93:17 117:21 145:22 146:22 147:2,3 refusing 62:18 regard 77:6 reimplemented 112:12 115:3 reiterate 131:6 relate 71:2,12 167:12,16 related 10:10 57:8 66:7 89:11 relates 78:15,24 relating 48:9,16 129:1,12 131:8,13 relationship 141:24 relative 119:20 released 112:4 relevance 111:2 181:9 relevant 109:7 114:11 168:14 169:14 171:17 reliable 69:17 175:1 relocation 94:15	rely 10:12 114:19 relying 86:25 128:16 130:25 131:7 132:13,15,18,19 143:6,19 remapping 127:23 130:22 Remappings 14:1 remember 22:9 111:19 148:9,24 175:4 179:13,23 184:6 Remind 33:7 removed 20:20,24 21:17 reopen 24:20 repeat 29:25 68:6 98:25 repeated 92:18 repeatedly 83:19 repetition 26:17 rephrase 151:6 replaced 74:17 112:10 replication 115:4 report 13:13 29:15 35:16 Reporter 6:12 reports 155:12 repository 83:7 represent 99:11,12 138:2 165:3 183:18 representation 80:20,25 representative 123:3 represented 80:19 119:24 represents 137:22 requantization 130:8 require 33:13 48:2,5 57:10 71:16 82:7 130:22 required 27:15,17 30:23 31:3,4 57:10 130:8,10 164:5 182:9 requirement 28:20,25 29:4 71:17 109:11 requirements 44:15,17 requires 27:24 46:23 47:4 82:15 166:1 reread 10:1 81:6 102:11 rereading 81:14 RESAR 175:1,3
---	--	---

research 63:12 68:8 72:5 122:20 123:2,8,13,14,16,17,18,21 124:1 168:6,7 181:5,15,21 182:8 researchers 164:19 reshuffling 17:14 35:1 resource 45:19 46:5 50:2,6,8,14,16, 19 51:5,8,20,23 59:8 82:23 83:4 85:12,13,22,23 88:8,13,19,22 128:20 138:25 139:2,14,23 140:12,13,20,23 141:2 151:18 resources 40:23 41:12 51:23,24 72:11,18 89:19 135:5 respect 138:1 response 43:5 52:5 responsibilities 6:18 responsible 135:7 rest 120:8,11 142:10 restart 70:11,23 restarted 70:9 restroom 138:4 restructuring 66:18 result 90:20 137:1 164:2 results 90:14 124:19 125:1,8 147:20 160:21 170:15,25 183:13 retaining 6:17 retired 77:16 168:7 retract 169:16 return 90:21,24 returns 90:19 reused 115:9 review 8:11 12:12 14:19 62:25 63:8 71:7 83:9,21 84:2 85:7 127:4,7 185:14 reviewed 10:9,24 83:11,13 reviewer 77:9 reviewers 65:13 77:8 reviewing 77:7 175:17 reviews 65:22 66:2 revisions 114:4 rich 181:18 rid 165:5	Ridge 64:17 ridiculous 38:20 rigorously 100:12 Rincon 169:24 RISC 102:2,4 103:15 108:10 Roadrunner 64:22 65:1,2,4,5 Robert 111:16 root 34:23 round 166:6 routine 25:19 routinely 64:16 rule 146:10 Rules 7:16 run 28:14 35:3 54:10 56:3 74:6 76:17, 20 79:10 96:4 114:6 125:1 147:11 175:20 182:25 running 19:24 40:4 70:2 78:4,13 108:1 175:22 runs 73:2,4 74:4,5,7 78:3 runtime 92:9,12,16 131:22	scheduler's 113:13 schedulers 74:25 75:3,7,9,10 scheduling 71:25 72:3,5,7,15,24,25 73:1,3,11,19,23 74:18 75:5,18 80:18 84:20,23 85:1,2,14 86:2,7,15,19 103:20,24 104:20 112:17,19 114:13, 19,23 115:2,8 148:11 149:2,3,4,9,10, 11 167:2,3,6,8,10,11 168:21 scheduling/task 69:3 71:23 scheme 115:5 Scholar 11:12 school 80:20,25 science 22:17 24:22 29:17 52:5 59:25 65:14,16 118:10 scientist 23:12 27:20 35:10 53:3 55:1 57:17 92:17 139:9 scientists 24:22 27:10 52:21 72:16 scope 19:20 20:15,18 21:14,16 109:25 110:16 121:13,24 156:14 scratch 29:11 41:17 47:25 53:9 71:10 152:22 173:4 screen 60:15 scroll 15:24 16:8 62:18 scrolling 62:18,20 search 11:14,16 49:18 searches 11:17 searching 24:5 secondary 63:3 section 6:16 116:23,24 117:1 124:19 125:25 180:3,21 181:21 sections 95:12 183:19 security 168:11 sees 132:3 select 123:2 131:25 selection 113:22,24 114:1 Semi-static 116:14 155:7 send 15:4 127:19 128:1 sending 117:7,12 sends 117:9,11 sense 16:19 17:17 59:22 102:19,25 106:25 107:10 126:22 145:1 149:3 154:13
S		
retaining 6:17 retired 77:16 168:7 retract 169:16 return 90:21,24 returns 90:19 reused 115:9 review 8:11 12:12 14:19 62:25 63:8 71:7 83:9,21 84:2 85:7 127:4,7 185:14 reviewed 10:9,24 83:11,13 reviewer 77:9 reviewers 65:13 77:8 reviewing 77:7 175:17 reviews 65:22 66:2 revisions 114:4 rich 181:18 rid 165:5	Ridge 64:17 ridiculous 38:20 rigorously 100:12 Rincon 169:24 RISC 102:2,4 103:15 108:10 Roadrunner 64:22 65:1,2,4,5 Robert 111:16 root 34:23 round 166:6 routine 25:19 routinely 64:16 rule 146:10 Rules 7:16 run 28:14 35:3 54:10 56:3 74:6 76:17, 20 79:10 96:4 114:6 125:1 147:11 175:20 182:25 running 19:24 40:4 70:2 78:4,13 108:1 175:22 runs 73:2,4 74:4,5,7 78:3 runtime 92:9,12,16 131:22	scheduler's 113:13 schedulers 74:25 75:3,7,9,10 scheduling 71:25 72:3,5,7,15,24,25 73:1,3,11,19,23 74:18 75:5,18 80:18 84:20,23 85:1,2,14 86:2,7,15,19 103:20,24 104:20 112:17,19 114:13, 19,23 115:2,8 148:11 149:2,3,4,9,10, 11 167:2,3,6,8,10,11 168:21 scheduling/task 69:3 71:23 scheme 115:5 Scholar 11:12 school 80:20,25 science 22:17 24:22 29:17 52:5 59:25 65:14,16 118:10 scientist 23:12 27:20 35:10 53:3 55:1 57:17 92:17 139:9 scientists 24:22 27:10 52:21 72:16 scope 19:20 20:15,18 21:14,16 109:25 110:16 121:13,24 156:14 scratch 29:11 41:17 47:25 53:9 71:10 152:22 173:4 screen 60:15 scroll 15:24 16:8 62:18 scrolling 62:18,20 search 11:14,16 49:18 searches 11:17 searching 24:5 secondary 63:3 section 6:16 116:23,24 117:1 124:19 125:25 180:3,21 181:21 sections 95:12 183:19 security 168:11 sees 132:3 select 123:2 131:25 selection 113:22,24 114:1 Semi-static 116:14 155:7 send 15:4 127:19 128:1 sending 117:7,12 sends 117:9,11 sense 16:19 17:17 59:22 102:19,25 106:25 107:10 126:22 145:1 149:3 154:13

sentence 15:22 63:23 69:1 79:24 85:20,25 97:2,3 98:2 102:11,17 117:21 122:23 127:19 129:7 138:24 141:17 155:24 156:12 161:14,15,16 163:8 164:18 170:22 sentences 93:2 separable 11:3 separate 19:17 27:4 39:3 42:4 144:20 September 6:1,5 sequential 153:11 180:18 181:3 182:4,6 sequentially 86:21 serial 119:23 120:5 122:12 173:14 serially 26:21,25 server 176:11 servers 167:19,22 set 39:14,15 47:17 74:23 170:14 171:13 178:6,9 set-top 167:19,21 severely 106:17 shape 125:7 SHARC 102:3 103:12 107:18,23 108:1,5,7,10 135:11 136:15,16,18 154:4,8,11 share 107:19,22 108:7 154:5 sharing 154:6 short 123:20 short-term 72:24 73:2 Shorthand 6:12 show 83:7 124:6 showing 159:14 shows 160:15 Shuttle 70:1 sic 6:9 side 15:3 signal 42:7 significance 9:5 significant 63:6 73:22 113:14 significantly 83:14 84:25 silent 86:20	silly 27:10 SIMD 148:19 similar 42:5 74:8 simple 48:23 99:1 113:9 simpler 75:11 83:14 84:9,25 118:15 119:5 153:21,24 154:15 simplicity 118:7,8 149:14 180:25 181:6 simplification 149:19 simplifying 118:11,16,23,25 150:7 simply 19:21 21:12 simulation 26:18 27:3 28:13 simulators 115:2 single 27:12 80:25 120:2 153:20 166:20 singular 80:4 sir 8:5 9:15,25 12:7 15:20 25:15 33:10 35:6 46:6,16 49:6 62:23 68:13 115:21 122:2 134:2 sit 151:8 sitting 8:19 81:13 82:10 85:3 132:12 152:10 160:6 situation 47:1,19 105:8 size 94:13 130:5 136:14 176:23 177:3,13,16 178:12 179:2 sizes 78:6 121:17 sketchy 146:8 skilled 90:9 skim 77:16,17 sleep 152:4 slow 112:11 164:10 slower 38:17 150:4 153:17 164:4 slowest 38:18 39:10 151:15 164:10 slowly 62:20 small 97:23 155:9 177:7,8,16 178:10 smaller 97:23 177:21 socket 42:4 soft 75:4 software 82:22 83:3,10 84:5,12 85:4 98:4,7,18 105:25 109:18,21,23 110:1, 5,23	software's 105:24 Solarity 42:1 solid 182:1 solution 105:15,19 123:22,23 146:17 165:25 solutions 66:13 solve 165:23 178:21 Sony 66:8 sort 100:18 110:7 132:24 148:23 175:18 sorted 157:14 sorting 114:9 182:16 sound 139:9 Sounds 175:23 source 55:7,10,20 56:6,8 83:21 84:2 85:7 98:14,18 143:6 144:11,15,17 145:7 152:25 153:4,5 Space 69:25 spacecraft 70:3 Spanish 175:4 Sparkman 6:25 spatial 182:22 speak 32:9,15 58:24 87:22 115:19 138:13 166:18 special 34:1 36:22 75:1 93:13 130:11 specialized 151:11,12 specific 44:13 101:16 105:20 108:25 131:12 132:18 142:4,6 146:4 171:25 specifically 35:15 46:8 83:22 100:10 108:22 180:25 speculate 121:14 speed 119:20 spell 179:24 spelled 18:16 52:10 spend 9:17 spends 64:5 spent 9:12 11:11 spin 152:5 spoken 10:5 spot 152:20
--	---	---

spun 175:19 squared 148:2 squishy 34:19 stack 94:13 standing 37:10 42:22 start 7:14 91:8 111:13 112:21 113:3 134:21 135:3 140:21 170:21 started 11:13 23:15 62:20 110:12,19 125:4 starting 6:20 135:2 starts 28:3 107:10 170:4,5,12 state 6:12,19 7:8,23 40:22 60:17 76:25 85:25 127:19 138:24 143:5 182:1 stated 69:1 statement 98:11 118:19 156:19,24 164:24 165:2 states 6:9 7:16 84:23 106:5 130:2 149:14 152:6,7 statically 52:6 stating 156:16 statistical 29:18 171:8 statistics 171:13,25 172:3,21 174:9 stenographer 6:4 7:7,11 8:6 13:7 15:11,14 27:8 28:7 32:4,7 35:23 37:20 44:21 51:7 57:11 58:19,22 61:1,23 62:2 64:24 65:15 68:5 69:21 72:9,21 74:10,15,20 79:16 86:12 87:16,19 88:4 90:7,22 91:7 95:25 99:8 103:11 111:5 115:7,13,16 116:3 117:10,18 132:17 135:23 138:7,10 144:1 148:1,20 150:22 154:1 155:2,6 158:24 159:22 163:20 165:18 166:9, 12 167:20 176:14 181:1 183:15 185:19,22 stenographer's 61:25 step 72:4 90:13,14 101:22 137:25 164:6,7,8 steps 90:10,11 164:10 stipulate 160:11 stop 36:9 40:9 stopping 138:6 storage 69:20 70:23 111:11 167:24 168:3,14,16 175:1,20 176:19 180:4,5, 7 181:12,18	storage/san 111:4,6,14 stored 108:1,3 straggler 163:10,11,24 164:1,9,21 165:2,6 strict 17:17 47:17 strictly 143:24 strided 180:19 strike 139:12 176:21 179:7 stripes 177:9 striping 176:19 structure 128:21 struggled 164:19,25 struggling 85:18 stuck 46:22 student 169:24 students 77:19 studies 170:6 study 87:3 stuff 96:4 168:11 stupid 182:16 sub-routine 25:14,16 sub-steps 90:13 sub-sub-tasks 24:25 sub-task 24:18,21 25:3,4,10,11,16, 20 26:3,6,7,10 27:23,24 28:20 29:1 35:4 55:20 56:3,7 59:22 76:9,10,13, 21 94:7 95:17 96:6,7,13,16 100:17,25 103:20,24 104:20 106:22 108:9,11 119:17,20 120:13,25 122:5 126:19,20 135:7 137:1 149:15 sub-tasks 17:13 24:24,25 26:2 27:12 30:3,9,21 31:10 32:22 34:25 45:11 46:8,18 48:10,17 49:21 57:1,7, 14 59:21 66:19 93:9,22,24 94:1,2,4 95:11 96:18 97:4,6,7,17 99:11 105:8 108:1 121:6 126:4,15 127:24 129:2, 12 131:8,14 166:25 167:1 subdivided 97:23 subjective 25:11 submitting 127:4 suboptimal 180:22 subset 47:12,13	substantial 140:10 substantive 16:19 successful 116:5 suddenly 112:10 sufficient 62:10 101:12 173:21 suggest 160:7 169:8 174:2 175:11 179:21 suggested 11:6 suggesting 174:19 supercomputing 76:25 77:5,12,13, 15 support 22:20 36:5 suppose 11:17 38:19 94:8 122:25 139:11 145:9 164:16 surgery 8:19 surprised 183:22 switch 117:19 sworn 7:4 symmetric 102:6,19,23,25 117:6 symmetry 117:22 118:14 119:4 synchronization 28:11,16,17 164:6 synchronous 164:4 165:4,5 system 16:12,15,16,21 17:2,11 20:4, 6,15,20,25 21:7,10,12,13,18 33:3,17 34:6,13,24 39:18,22 40:4 42:1,2 43:7, 11 44:3,9 53:21 61:4,5,8,14,17 62:7, 11 64:2,8 75:11,15 77:25 78:25 79:5 81:3,17 94:11 101:9 104:18 106:19 108:5 111:7 116:24 133:10 154:17 155:12,15,25 156:3,6,9,17 157:3 160:2,8,9 162:13,19,21,23,25 163:4,5 174:8 175:20 176:24 177:18 179:10 181:12 182:3 systems 54:3 60:18,20,21,23 61:13, 21,22 63:17 64:20 69:12 72:2 73:7 75:3,16 78:16 84:25 101:5 112:24 116:15 125:17 155:9 164:5 168:20 176:19,23
T		
t-u-r-i-n-g 37:22		
tab 155:9		
Table 120:22		
tables 182:16		

taboo 159:3 tabs 155:5 takes 40:14 92:8,12 117:9,11 151:15 taking 90:2 talk 19:10 28:10 35:12 78:22 102:16 107:24 114:5 120:11 121:16 talked 10:3 32:11 35:25 138:15 149:18 155:17 164:5 talking 18:23 29:18 31:7 39:1 48:21 76:11 80:7,18 86:18 93:21 97:6 105:18,25 119:3 152:23 163:17,23 talks 22:17 tank 134:12,13,21 135:1 147:16 180:4,7 tanks 133:3,17 134:13,22,25 135:2,3, 5,7 target 14:1 132:25 133:5 targets 89:17 133:7 task 24:21 25:13 26:17 27:1 46:15 51:10 60:3 72:14,15 73:3 75:10 76:5, 8 85:15 86:3,8 89:16 90:15,18 91:1 93:4,8,13,23 105:2 106:18 107:10 109:10 112:17,19 113:23,25 114:1,23 115:1,8 132:23,24 134:17,19 137:4 140:24 141:5 142:6 147:11 148:11 152:12 153:20 164:3,13,15,16 165:11,15,16 167:2,6,25 168:2,21 taskifying 25:7,8 tasks 14:2 24:24 39:10 57:9 75:9,11 84:20 85:14 86:3,7,10,15,19,20 90:25 106:1 109:3,7,13,16 116:14 125:16 135:17 142:4 151:25 152:22 155:8 165:19 166:25 167:1 taught 67:8 69:12 88:9 91:16 112:24 127:20,21 taxonomy 24:22 teach 88:13 89:20 92:22 93:14 teaches 91:22 92:14 126:24 teaching 168:20 teachings 116:6 team 111:16,21 tease 173:24 techniques 82:23 83:4 technologies 67:16,20,25 68:8,15	technology 65:14,17 Tehama 184:22 telling 97:5 tells 61:16 78:24 92:18 96:25 159:10, 25 temporal 99:12 ten 8:3,4 153:17 159:14 167:3 172:10 175:18 178:14,16 tend 73:11 75:11 171:24 tensor 163:18,22 term 16:2 19:5 22:13,23,24 23:8,9,11 24:6,10,14,18 25:3,4,14 26:10,14 29:7,16 47:24 48:1 49:23,24 50:6,11 52:25 53:7,8 56:8,10 57:1,15,20,24 58:11,12,13 81:22 93:21 122:25 terminology 97:11 terms 19:17 24:2 35:17 40:4 69:24 120:8,9,11 130:6 180:22 testified 7:4 184:16 testify 9:6,18,20,22 185:1 testimony 8:17 9:3 testing 69:16 text 184:3 theorem 145:5 theorems 124:6 theoretic 178:1 theoreticians 124:5,8,14 theoreticians' 124:11 theories 124:11 theory 47:17 54:3 71:25 72:2,17 73:11,19,23 114:20 125:11 168:8 therefrom 6:15 thesis 169:23 thick 177:13 thing 8:13 21:10 34:24 55:2 61:4 97:22 113:21 114:3,6,7 120:7 126:11, 20,25 128:6 139:10 140:5,8 141:13 146:12,13 148:3 149:23 150:3 152:24 155:12 156:7 159:6,7 165:2 173:13, 19,25 174:24 175:22 177:7,16 184:13 things 7:15 10:3 11:11 15:5 24:23 28:10 55:24 57:4,17 59:15,18 65:11 66:11 72:12,13,18 74:25 75:13,17 92:16,17,18 94:4 97:23,24 99:5,13	104:9,13 108:22 110:20 118:16,23,25 121:16 126:20 135:11 136:19 142:16, 23 143:11 145:10 148:24 149:6,25 150:4 158:20 164:1 172:2 177:8 178:5,8,18 180:20 182:1 thinking 8:12 47:16 53:6 110:20 Thomas 175:4 thought 67:6,8 147:16 thousand 70:17,18 thousands 97:14 threads 25:24 throw 147:17 thumb 146:10 TIDSP 44:14 tight 153:8 time 6:6,19 8:20 9:17 15:12,15 16:18 18:3 32:5,8 58:20,23 60:14 61:24 62:3 67:6,9 68:12 71:6 74:23 75:15 77:19 81:5,16 87:9,10,12,17,20 89:6, 9 91:16 99:20 100:21 102:20 104:13 112:23 114:7 115:1,10,14,17 117:12, 13,15 119:17,22 120:14,25 134:23 138:8,11 147:11 149:25 150:7,10,15, 19 151:1,7,20 152:21 158:22 166:1,7, 8,10,13 180:13 185:20,23,25 timeout 112:10 times 8:2 42:15 70:19,21 115:9 153:17 155:19 159:15 184:15 titled 13:25 116:13,24 125:16 162:1 168:24 175:1 179:9 183:13 Tivoli 111:11 today 8:17 9:3,18 10:6 70:3 81:13 82:10 85:3 109:21 110:24 127:8 132:12 160:6 Today's 6:5 told 10:7 tolerance 69:3,7,9,13 70:5 71:2,5, 12,15,17 top 40:18,19 63:3 64:18 85:19 88:6, 17 102:14 topic 73:10,12,16 115:12 148:17 166:5,24 169:15 175:15 total 9:14 111:4,14 130:6 180:4 tough 33:20 tracked 129:21 130:4,5 133:21
---	---	--

136:14 tracking 133:8 158:22 trademark 6:10 7:17 184:23 training 162:2 163:24 164:4,10,20 165:4,5 transcript 6:14 transfer 100:17 109:6,9,11 transfers 104:15 transform 45:6 57:12 transforms 105:1 treated 148:21 tree 110:8 trivial 156:25 trouble 134:6 true 153:13 173:25 182:14 truism 109:17 turing 37:21 44:19 45:3 turn 15:17,21 16:5 43:21 56:11,21 59:3 60:6 63:22 66:22 68:24 75:21 76:22 77:21 82:17 88:2 89:1 92:25 93:15 98:1 115:23 116:18,21 122:17 125:23 127:12 128:22 138:19,22 141:17 145:13 146:9 149:12 151:22 155:20 156:11 158:8 170:3 174:11 180:1 turned 151:21,23 185:4,7 Turning 71:20 turns 35:4 182:17 Twenty-one 181:10 type 16:16 36:25 79:15 100:4,6,7 102:19,24 117:5,23 118:1 119:20 122:4,5 126:5,8,21 136:5 137:2 144:9 145:2 149:17 150:6 153:20 types 42:9 61:19 65:11,24 68:18 78:5 79:21 81:2,15,20,25 82:2,12,15 97:1 100:8 101:13,23 102:3,15 105:22 121:2,11,15,22,25 122:9 126:15,25 131:3 142:19 typical 58:8 165:1 typically 18:5 22:17 52:5,11 58:4 67:7 69:11,13 70:5,21 73:3 75:10 77:11 124:1 148:18 152:25 153:9,17 typo 28:6	U U.S. 12:22 13:16 14:10,19 Um-hum 56:23 61:12 91:12 107:20 140:25 170:13 176:7 unaware 127:1 unclear 135:14 Undergraduate 73:7 undergraduates 67:8,9 understand 9:3,8 13:3,14,21 14:7, 15,24 23:12 30:1 31:7 35:18 43:3 57:18 67:13 85:18 106:12 112:9 124:20 131:11 157:8 understanding 60:4 63:4 100:23 understood 49:24 50:13,14 53:2 57:16 73:16 168:17 undertaking 156:14,21 unique 19:9 35:4 38:3 55:22 76:17 172:3 unit 20:1 38:25 39:4,5 42:1,2 177:19 United 6:9 7:16 106:4 units 40:24 78:7 96:1 122:13,14 universal 45:2 universality 44:23 unmet 63:5 unpack 55:17 unreportable 68:4 unsolved 123:21 unspecified 59:19 update 113:17 updates 113:14 urgently 142:5 utilization 58:5,6 utilized 30:17 utopians 110:2 V vague 57:6 values 127:20 131:22 132:19 136:7, 20	variability 173:21 VAX 74:16 vector 19:1 20:1 78:7 96:1 vectorization 97:18 148:22 Vectorizeability 143:3 vectorrizeable 96:2 vehicles 129:21 131:18 132:6 venued 6:9 verification 69:16 verify 85:4 versa 184:4 version 41:18 55:23 75:1 113:2 114:2 123:20 versions 12:1 versus 27:20 182:20 vice 184:4 video 108:5 167:2,7,10 view 22:2 37:13 38:4 42:21 43:9 47:18 76:10 86:11 114:12 151:24 virtual 75:19 119:17,18,21 175:22 virtualized 88:18,24 VLIW 148:8,11,17,24 149:23 150:1,2 VMEBUS 101:17 102:7 107:5 108:15,18 109:5 VMS 74:16 vote 70:2 VSP 106:3 W wait 150:3 waiting 107:16 150:5 walk 31:25 67:17 133:23 walk-around 166:8 wanted 7:14 28:13 105:23 120:21 142:16 173:18 War 72:7 Wayne 111:17 179:25 ways 69:13 141:12 159:14 Wednesday 6:1,5
--	--	---

week 109:22 weekend 147:12 well-defined 93:23 well-regarded 77:1 well-studied 171:15 widely 75:2 wife 10:7 Windows 12:3 24:3 74:7,11,16 145:11 156:10 word 20:11 23:2 25:7 29:13 31:18 33:11 36:2 46:22,23 52:9,11 53:24 55:16 67:3,24 74:22 84:8 88:17 134:7 137:10,12,19 148:8 149:6 157:8 158:19 wording 47:4 55:16 words 20:19,24 27:10 37:19 44:20 45:2 48:20 102:12 work 12:20 15:6 77:5,6 100:4,8 101:10,20 147:19 154:20,23 159:7 160:12,15,16 165:25 167:24 168:18 169:22 172:1 177:12 178:17 180:3 181:12 182:13,25 185:7 workable 110:14 worked 64:21 111:19 156:24 159:10 160:20 163:21 184:7 working 77:18 112:8 148:10 154:17, 19,24 155:11 works 67:14 73:16 123:25 157:12 world 64:18 72:7 111:9 worry 151:9 165:6 worrying 151:10 wrap 8:24 166:5 write 98:9,12 106:15 109:21 110:7 111:15 115:6 139:8 181:21 writing 83:10,24 85:5 110:13 136:11, 12 written 98:13,15 109:18 110:25 111:16 112:17 114:22 124:14 145:2 168:3 wrong 56:19 119:2 wrote 63:14 109:22 148:16 175:13, 15 183:7	X x386 39:3 x86 17:16 18:25 36:10,12,16 148:12, 22 Xeon 17:4,9,15,16 18:2,12,19,24 19:10,11,15,22 35:5,13,15 53:20 76:16 96:2 Xeone5-2680 17:25 Y Yale 148:10 YARN 78:12 83:14 84:15,16,19,23 85:8 86:9,14,23,25 year 10:1 16:14 41:24,25 69:12 70:17,18 77:17 81:11 87:8 127:6 years 68:3,7 73:23 110:17 112:25 113:15 115:5 156:25 158:16 165:24 168:20 175:19 179:12,18 181:10 yesterday 12:5 Z Z-H-U 170:5 Zhu 170:5,12 171:9	
--	---	--

ERRATA SHEET

CASE NAME: Microsoft Corporation v. ParTec AG, IPR2025-00318

DATE OF DEPOSITION: September 17, 2025

DEPONENT NAME: Darrell Long, Ph.D.

<u>PAGE #</u>	<u>LINE #</u>	<u>CORRECTION FROM</u>	<u>CORRECTION TO</u>	<u>REASON</u>
1	12	Darell	Darrell	Misspelling
2	1	Darell	Darrell	Misspelling
3	3	Darell	Darrell	Misspelling
4	4-5	Darell	Darrell	Misspelling
6	7	Darell	Darrell	Misspelling
7	3	Darell	Darrell	Misspelling
7	9	Darell	Darrell	Misspelling
7	24	Darryl	Darrell	Misspelling
13	9	Darell	Darrell	Misspelling
27	6	map reduce	MapReduce	Mistranscription
37	21	turing complete	Turing-complete	Mistranscription
41	1	Solarity	Celerity	Mistranscription
44	14	TIDSP	TI DSP	Mistranscription
44	19	turing complete	Turing-complete	Mistranscription
45	3	turing complete	Turing-complete	Mistranscription
45	6	48	Fourier	Mistranscription
52	5	its's	it's	Mistranscription
52	20	daemon or daemon	daemon or demon	Mistranscription
65	7	cell	Cell	Mistranscription
66	6	cell	Cell	Mistranscription
74	7	schedule	scheduler	Mistranscription
83	6	Git	the Git	mistranscription

<u>PAGE #</u>	<u>LINE #</u>	<u>CORRECTION FROM</u>	<u>CORRECTION TO</u>	<u>REASON</u>
94	20	scaleable	scalable	Misspelling
95	9	scaleable	scalable	Misspelling
95	13	scaleable	scalable	Misspelling
95	18	scaleable	scalable	Misspelling
95	20	scaleable	scalable	Misspelling
95	22	scaleable	scalable	Misspelling
95	23	parallelizeable	parallelizable	Misspelling
96	2	vectorrizeable	vectorizable	Misspelling
96	3	scaleable	scalable	Misspelling
96	24	scaleable	scalable	Misspelling
97	21	scaleable	scalable	Misspelling
106	3	VSP	DSP	Misspelling
115	2	discreet	discrete	Misspelling
143	3	Vectorizeability	Vectorizability	Misspelling
143	21	scaleable	scalable	Misspelling
143	24	scaleable	scalable	Misspelling
144	2-3	scalable	scalable	Misspelling
156	7	HER	ATR	Mistranscription
156	8	HCR	HC	Mistranscription
175	20	file	file system	Mistranscription
181	17	Beau	Bo	Misspelling
185	2	Majority	Minority	Misspoke

I DECLARE UNDER PENALTY OF PERJURY THAT THE FOREGOING IS TRUE AND CORRECT.


 Name

29 September 2025
 DATE