UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
14/150,635	01/08/2014	Wilson Parkhurst Snyder II	CVM-014US	9657

106719 7590 09/21/2017

Pavel Kalousek
22262 Crestline Road
P.O. Box 245
Palomar Mountain, CA 92060

EXAMINER	
ESMAEILIAN, MAJID	

ART UNIT	PAPER NUMBER
2477	

NOTIFICATION DATE	DELIVERY MODE
09/21/2017	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

pkalousek.ip@gmail.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte WILSON PARKHURST SNYDER II and DANIEL ADAM KATZ

Appeal 2017-004918
Application 14/150,635[1]
Technology Center 2400

Before LINZY T. McCARTNEY, NATHAN A. ENGELS, and JAMES W. DEJMEK, *Administrative Patent Judges*.

ENGELS, *Administrative Patent Judge*.

DECISION ON APPEAL

STATEMENT OF THE CASE[2]

Appellants appeal under 35 U.S.C. § 134(a) from a Final Rejection of claims 1–18, 20, and 21. Claim 19 stands subject to an objection. Final

---

[1] Appellants identify Cavium, Inc. as the real party in interest. Appeal Br. 1.
[2] Although not identified by Appellants as a related appeal, Appellants filed an appeal involving similar claims, prior art, and issues. *Compare* Appeal Br., Appeal No. 2017-004918 (filed Sept. 16, 2016), *with* Appeal Br., Appeal No. 2017-004917 (filed June 29, 2016). We remind Appellants that 37 C.F.R. § 41.37(c)(1)(ii) requires that Appellants identify related appeals, interferences, and trials in their appeal briefs.

Act. 16. We have jurisdiction over the pending claims under 35 U.S.C. § 6(b).

We affirm-in-part.

REPRESENTATIVE CLAIM

Claims 1, 13, 20, and 21 are independent claims. Claim 1, reproduced below, is representative of the claimed subject matter:

> 1. A method for parsing network packets via a plurality of clusters configured to parse network packets, the method comprising:
>
> receiving one or more packets to be parsed;
>
> determining a candidate cluster of the plurality of clusters for parsing the one or more packets;
>
> transmitting the one or more packets to the candidate cluster;
>
> launching the candidate cluster to parse the one or more packets when a launch condition is met; and
>
> receiving parse results for the one or more packets from the candidate cluster.

THE REJECTIONS[3]

Claims 1–18, 20, and 21 stand rejected under 35 U.S.C. § 103 as

[3] The Final Rejection states that claims 1–18, 20, and 21 are rejected under pre-AIA 35 U.S.C. § 103(a). Final Act. 4. Because the application was filed after March 16, 2013, however, the statements of rejection should have referenced AIA 35 U.S.C. § 103, not pre-AIA 35 U.S.C. § 103(a). Nevertheless, the body of the rejection includes findings from Kumar and/or Panigrahy for each limitation, along with a rationale to combine the cited references in accordance with AIA 35 U.S.C. § 103. Final Act. 4–15. Accordingly, we understand claims 1–18, 20, and 21 to stand rejected under

unpatentable over Kumar et al. (US 2013/0120168 A1; May 16, 2013) ("Kumar") and Panigrahy (US 2005/0238022 A1; Oct. 27, 2005). Final Act. 4–15.

ANALYSIS

Independent Claim 1

Appellants argue the Examiner's rejection of claim 1 (i) fails to demonstrate that Kumar and Panigrahy show each and every element of claim 1, (ii) improperly advances contradictory interpretations of claim terms, and (iii) fails to respond to Appellants' traversals of the Examiner's rejection. Appeal Br. 4–12; Reply Br. 2–10. For the reasons explained below, we disagree with Appellants.

In rejecting claim 1, the Examiner found "Kumar teaches a multi-core system wherein network traffic is distributed across one or more packet processing engines for packet processing purposes according to a flow-bases [sic] scheme." Final Act. 5 (citing Kumar ¶¶ 88, 168). More specifically, the Examiner found the multi-core system depicted in Kumar's Figure 5B and related disclosures teaches or suggests each limitation of claim 1 except receiving one or more packets to be parsed and receiving parse results for the one or more packets from the candidate cluster, for which the Examiner relied on Panigrahy. Final Act. 4–6 (citing Kumar Fig. 5B, ¶¶ 107, 187); Ans. 16–19 (citing, among other things, Kumar ¶¶ 210, 215).

AIA 35 U.S.C. § 103 as unpatentable over Kumar and Panigrahy and treat this oversight to be a typographical error.

Figure 5B of Kumar is reproduced below:

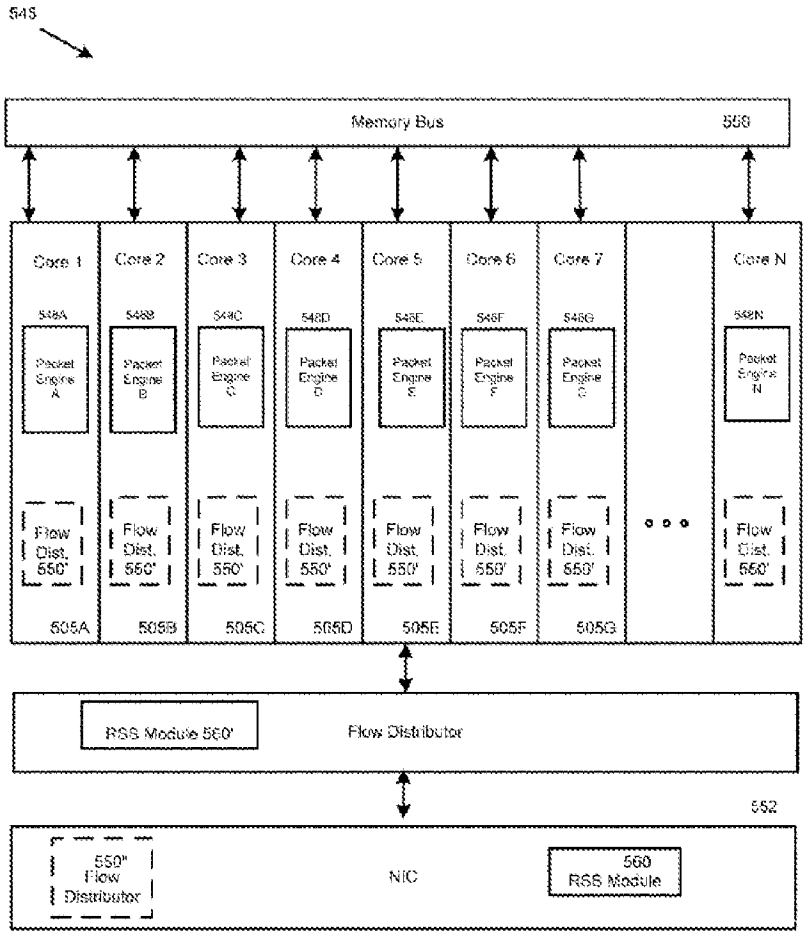

FIG. 5B

Figure 5B of Kumar is a block diagram illustrating an embodiment of Kumar's multi-core system. Kumar ¶ 26. Figure 5B of Kumar shows a plurality of processing cores (Cores 505A–N), wherein each processing core includes a packet engine or packet processing engine (548A–N) in communication with a memory bus. Kumar Fig. 5B, ¶ 187. We agree with the Examiner that Kumar's disclosure of a plurality of processing cores teaches "a plurality of clusters," as recited in claim 1. *See* Ans. 18 ("a core is equivalent to a cluster").

Relevant to "parsing network packets via a plurality of clusters," the Examiner found a packet processing engine in each of Kumar's processing cores can process received or transmitted network packets responsive to a time interval provided by a packet-processing timer. Final Act. 5 (citing Kumar ¶ 107); *see also* Ans. 16; Adv. Act. 2. Consistent with this finding, Kumar further discloses that each packet processing engine is responsible for managing the kernel-level processing of network packets received and transmitted by the multi-core system via a network. Kumar ¶ 106. Kumar additionally discloses that its packet processing engine can provide a variety of functions, including content switching, load balancing, packet acceleration, compression, encoding, decompression, decoding, network I/O processing, SSL offloading and processing, layer processing, and traffic management. Kumar ¶¶ 171, 188. These disclosures of Kumar teach or at least suggest that processing network packets includes checking or analyzing, in other words, "parsing," the contents of the packet to determine how the packet should be processed and which function the packet processing engine of the core should apply.

Accordingly, we agree with the Examiner that Kumar's disclosure of a packet processing engine of each processing core being configured to process network packets teaches or suggests "parsing network packets via a plurality of clusters," as recited in claim 1. The Examiner's findings are consistent with the Specification, which does not explicitly define "parsing," but instead describes the term broadly as "analyz[ing] the data in the packet to determine its characteristics[, which] may include its source, destination, or type." Spec. ¶ 3; *see also* Spec. ¶¶ 33–34 (disclosing non-limiting

examples in which parsing is performed on one or more of a packet's protocol layers and fields within those protocols).

As teaching the "determining" limitation of claim 1, the Examiner cites functions of Kumar's flow distributor (550), which communicates with processing cores (505A–N) as shown above in Figure 5B. Final Act. 4–5 (citing Fig. 5B, ¶ 187). The flow distributor distributes, forwards, routes, controls, and/or manages the distribution of data packets among the processing cores and/or packet processing engines running on the cores. Kumar ¶ 198. More specifically,

> [t]he flow distributor 550 can . . . tak[e] in data packets and distribut[e] them across the processors according to an operative load balancing or distribution scheme. . . . [T]he flow distributor 550 can comprise one or more operations, functions or logic to determine how to distribute packers [sic], work or load accordingly. . . . [T]he flow distributor 550 can comprise one or more sub operations, functions or logic that can identify a source address and a destination address associated with a data packet, and distribute packets accordingly.

Kumar ¶ 200.[4] Further, "[t]he flow distributor may use any type and form of statistical or probabilistic algorithms or decision making to balance the flows across the cores." Kumar ¶ 198; *see also* Kumar ¶ 200 ("The flow distributor 550 can distribute network traffic among the cores 505 according to a distribution, computing or load balancing scheme such as those described herein.").[5] Still further, the Examiner cites the following passage from Kumar:

---

[4] *See also* Kumar ¶ 199 ("In one instance, the one or more flow distributors 550 can determine how to balance load by communicating with each other.")
[5] *See also* Kumar ¶ 199 (disclosing various load criteria by which the flow distributor can determine whether to forward a data packet to a particular core or packet processing engine therein).

> [t]he flow distributor may operate responsive to any one or more rules or policies . . . [that] may identify a core or packet processing engine to receive a network packet . . . . The rules may identify any type and form of tuple information related to a network packet . . . . Based on a received packet matching the tuple specified by the rule, the flow distributor may forward the packet to a core or packet engine.

Kumar ¶ 215; *see* Final Act. 3.[6]

We agree with the Examiner that these disclosures of Kumar teach "determining a candidate cluster of the plurality of clusters for parsing the one or more packets," as recited in claim 1. In particular, the flow distributor performs these functions by determining which core ("candidate cluster") to receive a network packet and forwarding the packet to the determined core for processing by the core's packet processing engine.

Appellants argue the Examiner erred by failing to demonstrate that Kumar discloses "parsing network packets via a plurality of clusters," as recited in the preamble of claim 1. Appeal Br. 4–7; Reply Br. 2–8. In particular, Appellants assert paragraph 107 of Kumar references certain entities activated by the timer, but none of these entities allegedly performs packet parsing. Appeal Br. 6. Here, Appellants also assert the Examiner has not shown the packet processing engines of processing cores 505A–N operate these entities. Appeal Br. 6. Further, Appellants assert paragraph 215 of Kumar does not require parsing and instead suggests matching a mask to a received packet on a bit level. Appeal Br. 6. Appellants also assert that even if matching is considered parsing, Kumar's flow

[6] Kumar's tuple information also provides support in parsing IPv6 extension headers of a received network packet. Final Act. 3 (citing Kumar ¶ 210 ("2-tuple of source IPv6 address, and destination IPv6 address, including support for parsing IPv6 extension headers"), ¶ 215).

distributor—not its processing cores—would perform the parsing. Appeal Br. 6. Still further, Appellants assert paragraph 210 of Kumar does not disclose parsing but instead merely discloses a receive-side-scaling (RSS) module that generates different types of hash functions for load balancing across processors. Appeal Br. 6. In addition, Appellants assert the Examiner's finding that Panigrahy teaches packet parsing is irrelevant and introduces a new grounds of rejection because the Examiner rejected the preamble of claim 1 over Kumar. Appeal Br. 7; Reply Br. 2–3

We disagree with Appellants and agree with the Examiner that Kumar teaches or suggests "parsing network packets via a plurality of clusters" for the reasons discussed above. Additionally, the Examiner found Panigrahy discloses a network device including a parsing processor that parses incoming packets and sends processed outgoing packets to the network. Final Act. 5–6 (citing Panigrahy, Fig. 1, ¶ 10). Thus, we agree with the Examiner that Panigrahy's network device additionally teaches "parsing network packets." Indeed, Appellants acknowledge that Panigrahy's parsing processor is consistent with the Specification's disclosure of a parser comprising a parsing system configured to parse. Appeal Br. 5–6 (citing Spec. ¶¶ 33–34; Panigrahy ¶¶ 3, 9).

Next, Appellants argue the Examiner erred in finding Kumar teaches or suggests "determining a candidate cluster of the plurality of clusters for parsing the one or more packets," as further recited in claim 1. Appeal Br. 7–10. As an initial matter, Appellants interpret the term "cluster" as an entity configured to perform the parsing comprising a plurality of engines. Appeal Br. 8 (citing Spec. ¶¶ 36, 42). Accordingly, Appellants assert neither Figure 5B, paragraph 187, nor any other part of Kumar discloses the

organization of processor cores into any entity. Appeal Br. 8. Further, Appellants assert that although each of Kumar's processor cores may perform a different task, none of the tasks is identified as parsing. Appeal Br. 8. Still further, Appellants assert because Kumar only discloses a single cluster, Kumar cannot teach a determination of a candidate cluster from a plurality of clusters. Appeal Br. 8.

We disagree with Appellants. As an initial matter, for the reasons discussed above, we agree with the Examiner that Kumar's flow distributor performs functions that teach or suggest "determining a candidate cluster of the plurality of clusters for parsing the one or more packets." Further, contrary to Appellants' arguments, Kumar teaches "a plurality of clusters," not just a single cluster, with its disclosure of a *plurality* of processing cores, each core comprising one or more packet processing engines. *See* Ans. 4 ("Kumar[,] Fig. 5B and [paragraph 187] teach[] a plurality of core processors 505A–N (i.e.[,] clusters) comprising a packet processing engine" (emphasis omitted)); Ans. 18 ("Kumar Fig. 5B, showing a multi-core system (i.e., a core is equivalent to a cluster), including one or more packet engines or processors"); *accord* Final Act. 4–5 (citing Kumar Fig. 5B, items 505A–N, ¶ 187); *see also* Kumar ¶ 168 ("There may be multiple packet engines 240 each running on a respective core of the plurality of cores."). And as discussed above, Kumar's disclosure of a packet processing engine of each processing core being configured to process network packets teaches or suggests "parsing network packets via a plurality of clusters." Therefore, contrary to Appellants' arguments, we find Kumar's teachings to be consistent with the Specification, which similarly describes a cluster as

including one or more engines configured to parse received packets. *Compare* Kumar Fig. 5B, ¶¶ 168, 187, *with* Spec. ¶¶ 36, 42.

Appellants additionally argue the Examiner took inconsistent positions with respect to the claimed "plurality of clusters." Appeal Br. 9; Reply Br. 9–10. In particular, Appellants assert the "allegation [in the Examiner's Answer] that 'every core, is considered as a cluster' *contradicts* the examiner's allegation form [sic] the Office action (s) that a *plurality* of packet processing engines is considered to be *a cluster*" and appears to introduce a new ground of rejection. Reply Br. 9–10. We will not consider Appellants' argument that the Answer includes an undesignated new ground of rejection. This issue is addressable by petition to the Director, not appeal to the Board. *See* 37 C.F.R. §§ 41.40(a), 1.181; MPEP § 1207.03(b). Moreover, Appellants waived this argument by failing to petition the Director "within two months from the entry of the examiner's answer and before the filing of any reply brief." 37 C.F.R. § 41.40(a).

Appellants also assert "the term launching a cluster is to be interpreted as an activation of the cluster causing the candidate cluster to parse the subset of packets allocated to the cluster, i.e., by moving all of its allocated engines from an idle state to a processing state." Appeal Br. 11 (emphasis omitted). Thus, Appellants assert "at the minimum, the term launching is to be interpreted as starting engine(s) in the cluster to begin parsing, i.e., changing a state of the cluster from idle (non-parsing) to processing (parsing)." Appeal Br. 11 (emphasis omitted). Applying this interpretation, Appellants argue paragraph 107 of Kumar fails to disclose the state of the packet processing engine because the packet processing engine is able to accept more than one packet for processing. Appeal Br. 11.

We find Appellants' arguments unpersuasive because Appellants' interpretation of "launching the candidate cluster" is not commensurate with the scope of claim 1. The Specification does not provide a limiting definition of "launching the candidate cluster," but instead merely provides that "[*i*]*n some embodiments*, when the candidate cluster is launched, its state is changed to processing." Spec. ¶ 110 (emphasis added). Accordingly, contrary to Appellants' arguments, the term "launching the candidate cluster" does not require an activation of each engine in the candidate cluster that includes moving all of its allocated engines from an idle state to a processing state. Rather, we find a broad but reasonable interpretation of "launching the candidate cluster" includes running the candidate cluster to parse the transmitted one or more packets. Applying this interpretation, we agree with the Examiner that Kumar teaches or suggests "launching the candidate cluster." More specifically, the Examiner found Kumar discloses a packet processing engine that processes received network packets. Final Act. 5 (citing Kumar ¶ 107 ("packet engine 240 processes network packets")); *see also* Kumar ¶ 106 ("[P]acket engine 240, also generally referred to as a packet processing engine or packet engine, is responsible for managing the kernel-level processing of packets received and transmitted by appliance 200 via network ports 266."). The Examiner further found Kumar discloses one or more of these packet processing engines running on a respective processing core. *See* Final Act. 4–5 (citing Kumar Fig. 5B (showing Packet Engines 548A–N within Cores 505A–N), ¶ 187); *see also* Kumar ¶ 168 ("There may be multiple packet engines 240 each running on a respective core of the plurality of cores."), ¶ 188 ("packet engine(s) 548A-N can . . . comprise . . . packet engine 240"). These

disclosures of Kumar suggest running the processor core so that its one or more packet processing engines can process or parse the one or more received network packets. In other words, we agree with the Examiner that Kumar teaches or suggests "launching the candidate cluster," as recited in claim 1.

In view of the foregoing, Appellants have not persuaded us that the Examiner erred in rejecting claim 1 over the combination of Kumar and Panigrahy.

Independent Claim 13

As an initial matter, Appellants repeat arguments for independent claim 13 that we find unpersuasive for the reasons discussed above with respect to claim 1. *See* Appeal Br. 4–12. Appellants further argue (Appeal Br. 12) that in rejecting claim 13, the Examiner's interpretation of a "launcher" as including Kumar's packet processing timer is inconsistent with the Examiner's statement in the rejection of claim 1 that Kumar teaches "a flow distributor (i.e., a launcher)" (Final Act. 5). We find Appellants' argument unpersuasive. First, although claim 13 recites a "launcher," claim 1 does not. Nor have Appellants proffered an interpretation of claim 1 that requires a "launcher." Therefore, given this difference in scope between claims 1 and 13, we find Appellants' argument that the Examiner made inconsistent findings with respect to the term "launcher" unconvincing. Further, because "each claim must be considered as defining a separate invention," *Jones v. Hardy*, 727 F.2d 1524, 1528 (Fed. Cir. 1984), we find it entirely within the parameters of proper patent examination for the Examiner to tailor his application of the prior art to the specific language of each

independent claim. *See also Pall Corp. v. Micron Separations, Inc.*, 66 F.3d 1211, 1220 (Fed. Cir. 1995) ("each claim is a separate statement of the patented invention"); *Altoona Publix Theatres, Inc. v. Am. Tri–Ergon Corp.*, 294 U.S. 477, 487 (1935) ("each claim must stand or fall, as itself sufficiently defining invention, independently of the others").

Moreover, we agree with the Examiner that Kumar teaches or suggests "a launcher" as recited in claim 13. As teaching "a launcher" and its recited functionality, the Examiner cites various functions of Kumar's flow distributor (550) and packet processing timer. Final Act. 9–10 (citing Fig. 5B, ¶¶ 107, 187). As explained above regarding claim 1, the flow distributor distributes, forwards, routes, controls, and/or manages the distribution of data packets among the processing cores and/or packet processing engines running on the cores. *See* Kumar ¶¶ 198–200, 210, 215.

Notably, the Specification does not define "a launcher" and its recited functions as one or more structural elements, but instead describes them as functional software modules for execution by a computer processor. Spec. ¶ 41 (describing a system including, among other things, a launcher module and a loader module), ¶ 45 (describing a launcher module), ¶ 46 (describing a loader), ¶ 129 (disclosing that modules can be "implemented via one or more software programs for performing the functionality of the corresponding modules or via computer processors executing those software programs."). Accordingly, we agree with the Examiner that a broad but reasonable interpretation of "launcher" includes the disclosed functions of Kumar's flow distributor and packet processing engine. Final Act. 9–10 (citing Kumar Fig. 5B, ¶¶ 107, 187). In particular, as discussed above with respect to claim 1, Kumar's flow distributor performs the "determine" and

"transmit" functions of "a launcher" by determining which core ("candidate cluster") to receive a network packet and by forwarding the packet to the determined core for processing by the core's packet processing engine. Kumar ¶¶ 107, 198–200. Further, Kumar's packet processing engine performs the launch function of a "launcher" by providing one or more time intervals to trigger the processing of received network packets. Kumar ¶ 107.

Appellants further assert that because Kumar and Panigrahy are not analogous art, one of ordinary skill in the art would not have been motivated to combine these references to arrive at the claimed invention. Appeal Br. 13–14. But in support of this assertion, Appellants improperly compare the Kumar reference to the Panigrahy reference instead of to Appellants' claimed invention. *See In re Clay*, 966 F.2d 656, 658–59 (Fed. Cir. 1992). For at least this reason, Appellants' argument is fundamentally deficient.

Moreover, we disagree with Appellants that Kumar is not analogous art. The Federal Circuit has established two criteria for determining whether prior art is analogous to a claimed invention: (1) whether the art is from the same field of endeavor, regardless of the problem addressed, and (2) if the reference is not within the field of the inventor's endeavor, whether the reference still is reasonably pertinent to the particular problem with which the inventor is involved. *Id.* As discussed above, Kumar involves using packet processing engines to process or parse network packets, which shows that Kumar is in the same field of endeavor as Appellants' claimed invention, namely, "parsing network packets." *Compare* Kumar Fig. 5B, ¶¶ 106 ("a packet processing engine or packet engine, is responsible for managing the kernel-level processing of packets"), 107 ("integrated packet

engine 240 processes network packets"), 189 ("packet engine 548 will process and operate on packets"), *with* Spec. ¶ 2.

Additionally, Appellants argue the Examiner failed to provide an adequate rationale to combine Kumar and Panigrahy. Appeal Br. 14. More specifically, Appellants assert Kumar discloses a multi-core architecture (Appeal Br. 14 (citing ¶¶ 88, 168)) but the paragraph of Kumar cited to justify the combination of Kumar and Panigrahy merely discloses a single-core processing unit (Appeal Br. 14 (citing Kumar ¶ 78)). Further, Appellants argue the Examiner's rationale to combine failed to explain how Kumar's single instruction, multiple data (SIMD) processing would be improved by including Panigrahy's parsing processor. Appeal Br. 14. Still further, Appellants argue the Examiner failed to explain how the teaching of "efficiently store [a] large number of transitions on chips" is supposed to benefit Kumar's multi-processor structure. Appeal Br. 14.

We find Appellants' arguments unpersuasive. First, we disagree with Appellants that the Examiner relied on paragraph 78 of Kumar to justify the combination of Kumar and Panigrahy. Rather, in combining Kumar and Panigrahy, the Examiner articulates a rationale to combine—in order to benefit from a parsing processor that can efficiently store a large number of transitions on-chips—drawn directly from the Panigrahy reference *See* Final Act. 10; Panigrahy ¶¶ 76, 77 ("[T]he parsing processor 120 can efficiently store a large number of transitions on-chip."). Second, the Examiner was not required to explain how Kumar's SIMD processing would be improved by Panigrahy's processor or how the cited motivation would benefit Kumar's multi-processor structure. Indeed, "neither the particular motivation nor the avowed purpose of the patentee controls" an obviousness

analysis. *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 419 (2007). Instead, "[u]nder the correct analysis, any need or problem known in the field of endeavor at the time of invention and addressed by the patent can provide a reason for combining the elements in the manner claimed." *KSR*, 550 U.S. at 420 (2007). Accordingly, we find the Examiner's rationale to combine Kumar with Panigrahy is more than adequate to support the Examiner's proposed combination.

Additionally, we note that Appellants cite no evidence to show that applying Panigrahy's teachings to Kumar's multi-core system would have been uniquely challenging or anything more than a routine exercise of applying known techniques to achieve predictable results. *See KSR*, 550 U.S. at 416–17 (explaining as examples of combinations likely to be obvious "[t]he combination of familiar elements according to known methods . . . when it does no more than yield predictable results" and "the mere application of a known technique to a piece of prior art ready for the improvement"); *Leapfrog Enters., Inc. v. Fisher-Price, Inc.*, 485 F.3d 1157, 1162 (Fed. Cir. 2007) (citing *KSR*, 550 U.S. at 418).

For the reasons stated above, Appellants have not persuaded us the Examiner erred in rejecting independent claim 13 over the combination of Kumar and Panigrahy.

Independent Claims 20 and 21

Appellants do not separately address independent claims 20 and 21 beyond those arguments addressed above. *See* Appeal Br. 46. As such, we find no error in the Examiner's rejection of claims 20 and 21 for the same reasons.

Dependent Claims 2–12 and 14–18

Appellants assert that dependent claims 2–12 are allowable for the reasons argued for independent claim 1 (Appeal Br. 15) and that dependent claims 14–18 are allowable for reasons argued for independent claim 13 (Appeal Br. 28). For certain dependent claims, Appellants additionally advance arguments beyond those advanced for claims 1 and 13.

We have considered the Examiner's rejections of dependent claims 2–12 and 14–18 in light of each of Appellants' arguments and the evidence of record. As explained below, we are persuaded of error in the Examiner's rejection of dependent claims 5 and 6. We find no error, however, in the Examiner's rejection of dependent claims 2–4, 7–12, and 14–18. Where Appellants reiterate the arguments advanced for claims 1 and 13 for their respective dependent claims, we disagree with those arguments for the reasons addressed above. We highlight the following dependent claims and arguments primarily for emphasis.

Claims 2, 3, and 9

Claim 2 recites "wherein the launch condition is met after transmitting the one or more packets meets a fraction of a parsing capacity of the candidate cluster, wherein the fraction is a positive number less than or equal to one." Appeal Br. 36. In rejecting claim 2, the Examiner found Kumar discloses identifying the load on an associated core and forwarding packets for processing to the associated core based on any metric related to processing load, for example, if the load on the associated core is above or below a predetermined threshold, or if the load on the associated core is less

than the load on other cores. Final Act. 6 (citing Kumar ¶ 199). The Examiner explains, "it is understood broadly that even receiving a packet constitutes a launch condition (metric) since depending on the parsing capacity of the device, lets [sic] say 10000 packets/sec, then that one packet received creates a fraction greater than zero and less than 1." Final Act. 6 (emphasis omitted).

Appellants argue Kumar's disclosure of allocating packets based on threshold for the load on an associated core does not teach or suggest claim 2 because the term load does not correspond to "parsing capacity." Appeal Br. 16. Instead, Appellants argue load refers to an ability of the processing core to accept more than one packet for processing, while the Specification describes claimed "parsing capacity" as "an ability of a cluster to allocate packets to the cluster's engine." Appeal Br. 16 (citing Spec. ¶ 47). Appellants further argue that even if parsing capacity were equivalent to a maximum load, the cited disclosure of Kumar fails to disclose a maximum load. Appeal Br. 16.

Appellants' arguments have not persuaded us that the Examiner erred. Claim 2 broadly but reasonably encompasses that the launch condition is met any time after transmitting a packet to the candidate cluster up to and including the time when the candidate cluster's parsing capacity is met. As the Examiner found, "even receiving a packet constitutes a launch condition." Final Act. 6 (emphasis omitted). Applying this interpretation, the cited disclosure of Kumar teaches or suggests a packet has been received by the associated core and that the associated core's parsing capacity has not yet been met by identifying a current load on the associated core and successfully forwarding additional packet(s) to that core. *See* Kumar ¶ 199.

Additionally, as discussed above with respect to claim 1, Kumar teaches or suggests processing, or "parsing," of received network packets by the packet processing engine. Accordingly, one of ordinary skill in the art would understand that a condition precedent for processing or parsing network packets (i.e., "launch condition") is that the packet processing engine has received a network packet to process or parse. *See KSR*, 550 U.S. at 418 (explaining that an obviousness analysis can take account of the inferences and creative steps of a person of ordinary skill in the art); *DyStar Textilfarben GmbH & Co. Deutschland KG v. C.H. Patrick Co.*, 464 F.3d 1356, 1360 (Fed. Cir. 2006) (stating that "the prior art" includes basic principles unlikely to be restated in cited references). In other words, absent any evidence to the contrary, one of ordinary skill would understand that so long as Kumar's associated core has a load of at least one packet and no greater than the core's maximum load (i.e., "parsing capacity"), packet processing or "parsing" can occur (i.e., "the launch condition is met").

In view of the foregoing, Appellants' arguments do not persuasively rebut the findings and explanation of the Examiner, which we agree show that Kumar renders claim 2 obvious.

Still further, we find Appellants' comparison of Kumar's load and "parsing capacity" unavailing. Indeed, the paragraph Appellants cite to describe "parsing capacity" states "[a] fully allocated cluster has reached its parsing capacity and cannot accept any more packets to be allocated to its engines." Spec. ¶ 47. This disclosure implies that a cluster that has not yet reached its parsing capacity—for example, a cluster having met only a fraction of a parsing capacity—*could* accept one or more packets to be allocated to its engine(s). *See* Spec. ¶ 47. Accordingly, in contrast with

Appellants' argument, we find "a fraction of a parsing capacity" corresponds to Kumar's load, which Appellants acknowledge as an "ability of the processing core to accept *more than one packet* for processing." Appeal Br. 16. More specifically, the cited disclosure of Kumar supports this correspondence by forwarding additional packet(s) to an associated core that has already received a current load of packets, thereby suggesting that an associated core can accept one or more additional packets for processing. *See* Kumar ¶ 199. Additionally, in accordance with the claim language, even though Kumar does not explicitly disclose a maximum load, Kumar at least suggests a maximum load or "parsing capacity" that is either greater than or equal to the associate core's load after accepting the one or more additional packets for processing. *See* Kumar ¶ 199.

For the reasons stated above, Appellants have not persuaded us the Examiner erred in rejecting claim 2 over the combination of Kumar and Panigrahy.

Claim 3 further limits claim 2 by specifying, "the fraction is one such that the launch condition is met when transmitting the one or more packets meets a parsing capacity of the candidate cluster." Appeal Br. 36. Appellants present arguments for claim 3 that are substantially similar to those presented for claim 2. Appeal Br. 17–18. We find these arguments unpersuasive for the same reasons discussed above with respect to claim 2, which covers the scenario when transmitting the one or more packets meets a parsing capacity of the candidate cluster. With respect to claim 9, which recites limitations substantially similar to claim 3, Appellants present arguments substantially similar to those presented for claims 2 and 3.

Appeal Br. 24. We find these arguments unpersuasive for the same reasons discussed above with respect to claims 2 and 3.

Claim 8

Appellants contend the Examiner erred in finding Kumar teaches or suggests "wherein the launch condition is met when at least one cluster different from the candidate cluster is idle and can receive a packet for parsing," as recited in claim 8. Appeal Br. 22–23, 36. In rejecting claim 8, the Examiner found Kumar discloses wherein data packets can be allocated to cores with the least amount of load, which broadly includes cores that are "idle." Final Act. 7–8 (citing Kumar ¶ 183). In response, Appellants argue that even the core with the least amount of load in Kumar is still carrying out computational work, i.e., processing, and, thus, is not "idle." Appeal Br. 23. Appellants further argue "paragraph [183 of Kumar] is silent as to the ability of the processing core to be allocated a packet." Appeal Br. 23.

We find Appellants' arguments unpersuasive. As found by the Examiner, Kumar discloses a rule in which packets are allocated to the processing core having the least amount of load. Kumar ¶¶ 183–184; *see also* Kumar ¶¶ 187, 198–200 (describing the load balancing functions of the flow distributor). Similar to the limitation at issue, in the event one of the processing cores among a plurality of cores is packet-less and, thus, is "idle," applying Kumar's "least amount of load" rule, network packets would be sent to that packet-less core. *See* Kumar Figure 5A (top figure), item 505C (showing a functional parallelism distribution scheme in which Core 3 is packet-less and, thus, "idle"); *see also* Kumar ¶¶ 183–184. Further, absent any evidence to the contrary, we find Kumar's disclosure of

packets being allocated to a core having the least amount of load at least suggests that the allocated core can receive the packet for processing.

Additionally, given the cited disclosures of Kumar, one of ordinary skill in the art would understand that when Kumar's multi-core system is initialized and the flow distributor begins to function, each of the cores would be "idle" (and able to receive packets for processing or parsing) until the flow distributor distributed a network packet to one of the cores for processing. *See KSR*, 550 U.S. at 418; *DyStar*, 464 F.3d at 1368. In other words, consistent with the Specification, upon initialization of Kumar's multi-core system, each core would be idle because each core is "not processing [packets] and to which no packet has been allocated." Spec. ¶¶ 46, 47.

For the reasons stated above, Appellants have not persuaded us the Examiner erred in rejecting claim 8 over the combination of Kumar and Panigrahy.

Claim 11

Appellants argue the Examiner erred in finding Kumar teaches or suggests "checking whether a current candidate cluster can receive a packet," as recited in claim 11. Appeal Br. 25–26, 37. In particular, Appellants assert the cited disclosures of Kumar do not disclose checking whether the current core can receive a packet, but instead disclose checking whether the current core (1) has the least amount of load among the cores,

(2) has received a predetermined number of packets, or (3) has been receiving packets for a predetermined amount of time. Appeal Br. 26.

We find Appellants' arguments unpersuasive. As an initial matter, the Specification does not provide a limiting definition for the disputed limitation. *See* Spec. ¶¶ 7, 8, 11. Accordingly, contrary to Appellants' arguments, "checking whether a current candidate cluster can receive a packet" is not limited to checking the cluster's capabilities in a vacuum without the application of any constraints. In other words, a broad but reasonable interpretation of the disputed limitation includes checking whether the candidate cluster has met all conditions precedent for receiving a packet.

Applying this interpretation, we agree with the Examiner that Kumar teaches or at least suggests "checking whether a current candidate cluster can receive a packet." The Examiner found Kumar teaches or suggests this limitation with its disclosure that data packets can be allocated to the cores (i.e., clusters) for processing with the least amount of load (i.e., parsing capacity). Final Act. 8–9 (citing Kumar ¶¶ 183–184). The Examiner explained "[a]s such it is understood broadly that the system can determine, based on load, whether or not a system can receive a packet, and if not, accordingly, another (i.e., new) core can be selected with a lesser load. Final Act. 8–9 (emphasis omitted). The cited disclosures of Kumar suggest that if a first core (i.e., "candidate cluster") does not meet a certain condition precedent for receiving packets (e.g., least amount of load), packets cannot be allocated to the first core. *See* Kumar ¶¶ 183–184. Thus, the cited disclosures of Kumar also suggest that until the first core meets the condition precedent for receiving packets, the core cannot receive packets

for processing. Accordingly, by checking a first core to determine if a condition precedent is met, Kumar's system is effectively checking to see if the first core can receive packets for processing based on the condition precedent.

For the reasons stated above, Appellants have not persuaded us the Examiner erred in rejecting claim 11 over the combination of Kumar and Panigrahy.

Claim 12

Appellants argue the embodiment of Kumar cited by the Examiner to reject claim 12 discloses a plurality of candidate clusters, instead of a single candidate cluster as claimed. Appeal Br. 27. We find Appellants' argument unpersuasive because the argument is not commensurate with the scope of claim 12. Claim 12 recites in relevant part "wherein determining the candidate cluster further includes: checking whether the candidate cluster is parsing some packets." Appeal Br. 37. Because claim 12 (including the limitations of independent claim 1 and dependent claim 11) recites steps of a method using the open-ended transition terms "comprising" and "includes," the claim does not preclude steps in addition to those recited. *See Genentech, Inc. v. Chiron Corp.*, 112 F.3d 495, 501 (Fed. Cir. 1997) ("'Comprising' is a term of art used in claim language which means that the named elements are essential, but other elements may be added and still form a construct within the scope of the claim."); *see also Gillette Co. v. Energizer Holdings Inc.*, 405 F.3d 1367, 1371 (Fed. Cir. 2005); MPEP § 2111.03 ("The transitional term 'comprising,' which is synonymous with including,' . . . is inclusive or open-ended and does not exclude additional,

unrecited elements or method steps."). In other words, contrary to Appellants' argument, nothing in claim 12 precludes Kumar's disclosure of more than one candidate core or cluster.

Further, Appellants assert that in the embodiment of Kumar cited to reject claim 12, the flow distributor does not determine which packet engine is to receive the packet, but merely submits a vote to an arbitrator that makes the determination. Appeal Br. 27. Based on this assertion, Appellants argue the Examiner's findings for claim 12 are inconsistent with the findings for claim 1, which assert that the flow distributor is making the determination.

We find Appellants' argument unpersuasive. Appellants' argument identifies an alternative embodiment of Kumar in which a plurality of flow distributors can determine how to balance load by communicating with each other. Kumar ¶ 199. However, the existence of an alternative embodiment does not negate the fact that the cited disclosure also discloses an embodiment in which the flow distributor itself determines whether to forward a packet to an associated core. Kumar ¶ 199. Indeed, paragraph 199 of Kumar further discloses that "[i]n other embodiments, a first flow distributor 550' can identify the load on an associated core and determine whether to forward a first data packet to the associated core based on any of the following criteria." Kumar ¶ 199. Accordingly, because the Examiner's findings for both claim 1 and claim 12 show a flow distributor determining which core (and associated packet processing engine) is to receive a packet, we disagree with Appellants that the Examiner's findings for claims 1 and 12 are inconsistent with each other.

Claim 14

Appellants assert without evidentiary support or analysis that paragraph 107 of Kumar fails to disclose "a delay timer in communication with the launcher and configured to measure a time elapsed since a previous cluster was launched," as recited in claim 14. Appeal Br. 29. As Appellants do not substantively address the Examiner's findings, Appellants' arguments are unpersuasive. *See* 37 C.F.R. § 41.37(c)(1)(iv); *see also In re Lovin*, 652 F.3d 1349, 1357 (Fed. Cir. 2011) ("[W]e hold that the Board reasonably interpreted Rule 41.37 to require more substantive arguments in an appeal brief than a mere recitation of the claim elements and a naked assertion that the corresponding elements were not found in the prior art.").

Claim 15

Appellants argue Kumar, paragraph 199, is not applicable against claim 15 because it teaches a *plurality* of flow distributors, wherein *each* flow distributor checks an associated core for a load. Appeal Br. 31. In contrast, Appellants assert claim 15 recites a *single* launcher managing a plurality of clusters. Appeal Br. 31.

We find Appellants' argument unpersuasive. As explained above, we agree with the Examiner that Kumar teaches the claimed launcher. Appellants' argument identifies an alternative embodiment of Kumar in which each of a plurality of flow distributors can identify the load on (and determine whether to forward packets to) an associated core. Kumar ¶ 199. But the existence of this alternative embodiment does not negate the fact that the cited disclosure also discloses an embodiment in which a single flow distributor identifies the load on (and determines whether to forward packets

to) each of a plurality of associated cores. Kumar ¶ 199 ("[S]ystem 545 comprises *one* or more flow distributors 550." (emphasis added)); *see also* Kumar ¶ 198 ("In one embodiment, the flow distributor 550 executes on at least one of the cores 505A–N, while in other embodiments a separate flow distributor 550 assigned to each core 505A-N executes on an associated core 505A–N."). Accordingly, we disagree with Appellants that the cited disclosure of Kumar is not applicable against claim 15.

In addition, Appellants argue the cited disclosure of Kumar fails to teach or suggest an action taken by the flow distributor upon determination that the packet cannot be forwarded to the associated core. Appeal Br. 31–32. We disagree.

The Examiner found Kumar discloses that the flow distributor can determine whether to forward data packets to an associated core (i.e., cluster) for processing based on any metric, for example, if the core's load is less than a predetermined threshold. Final Act. 11–12 (citing Kumar ¶ 199). This disclosure suggests that if a first core (i.e., "current candidate cluster") does not meet a certain condition precedent for receiving packets (e.g., load less than a predetermined threshold), packets cannot be allocated to the first core. *See* Kumar ¶ 199. Thus, by checking a first core and determining that the condition precedent has not been met, Kumar's system is effectively determining that the first core cannot receive a packet for processing based on the condition precedent. *See* Kumar ¶ 199. Further, Kumar discloses that the flow distributor can distribute network traffic among the cores according to a computing or load-balancing scheme, which suggests that the flow distributor checks not just the first core but each core to determine if the respective core can receive a packet for parsing. *See* Kumar ¶ 200.

Therefore, given the above disclosures of Kumar, one of ordinary skill would reasonably infer that after determining that a first core cannot receive a packet for processing, the flow distributor would take further action by checking a second core to see if the second core can receive a packet for processing. *See also* Kumar ¶ 215 ("The flow distributor may operate responsive to any one or more rules or policies"). *See KSR*, 550 U.S. at 418 (explaining that an obviousness analysis can take account of the inferences and creative steps of a person of ordinary skill in the art). In view of the foregoing, we disagree with Appellants that Kumar fails to teach or suggest an action taken by the flow distributor upon determining that the packet cannot be forwarded to the associated core

For the reasons stated above, Appellants have not persuaded us the Examiner erred in rejecting claim 15 over the combination of Kumar and Panigrahy.

Claim 16

With respect to claim 16, Appellants argue "even the core with the least amount of load [in] the Kumar reference is still carrying out computational work, i.e., processing." Appeal Br. 32–33 (emphasis omitted). Thus, Appellants argue "such a condition of [Kumar's] core may under no interpretation be considered as corresponding to the condition of being *idle* as claimed." Appeal Br. 33.

We disagree with Appellants. As explained above, the Examiner found Kumar discloses a rule in which packets are allocated to the processing core having the least amount of load. Final Act. 12 (citing Kumar ¶ 183); *see also* Kumar ¶¶ 184, 187, 198–200 (describing the load

balancing functions of the flow distributor). In the event one of the processing cores among a plurality of cores is packet-less or "idle," applying Kumar's "least amount of load" rule would result in network packets being sent to that packet-less core. *See* Kumar Figure 5A (top figure), item 505C (showing a functional parallelism distribution scheme in which Core 3 is packet-less and, thus, "idle"); *see also* Kumar ¶¶ 183–184. In view of the foregoing, we disagree with Appellants that the core with the least amount of load in Kumar cannot correspond to the condition of being "idle" as claimed.

For the reasons stated above, Appellants have not persuaded us the Examiner erred in rejecting claim 16 over the combination of Kumar and Panigrahy.

Claim 18

Without substantively addressing the Examiner's findings (Final Act. 12 (citing Panigrahy ¶ 41)), Appellants argue Panigrahy fails to teach a ready counter and an executing flag as recited in claim 18. Appeal. Br. 34–35. Instead, Appellants' arguments simply quote the cited disclosure of Panigrahy and assert without evidentiary support or analysis that "[t]he broad and imprecise statement of rejection makes it virtually impossible to determine, which specific features of dependent claim 18 are allegedly anticipate [sic] by the reference to fails to [sic] Panigrahy." Appeal Br. 35. As Appellants do not substantively address the Examiner's findings, Appellants' arguments are unpersuasive. *See* 37 C.F.R. § 41.37(c)(1)(iv); *see also Lovin*, 652 F.3d at 1357.

Claims 5 and 6

Claim 5 recites "wherein the delay limit depends on an average time that each one cluster takes to parse packets transmitted to the one cluster." Appeal Br. 36. In rejecting claim 5, the Examiner found Kumar discloses, "packets can be distributed to a particular core for any period of time determined by a user, system or otherwise." Final Act. 7 (emphasis omitted) (citing Kumar ¶ 184). The Examiner explained

> that the average time to process is the same as the time that the packets are transmitted to a particular core, after the elapse of which, the packets are then routed to another core for processing. Therefore, the elapsed time is the same as the delay time upon which the processing of the packets can be started by another core unit.

Final Act. 7 (emphasis omitted). Appellants argue Kumar "fails to even mention *average* time." Appeal. Br. 21. Appellants further argue Kumar fails to disclose a time it takes to parse a packet and, therefore, Kumar's predetermined period of time cannot be dependent on the average time each cluster takes to parse packets. Appeal Br. 20. The Examiner did not respond to Appellants' arguments in the Examiner's Answer. *See generally* Ans. 16–20.

Kumar discloses packets can be distributed to a particular core for a predetermined period of time (for example, five milliseconds), after which, data packets are transmitted to a different core for the predetermined period of time. Kumar ¶ 184. However, the Examiner has not provided sufficient persuasive evidence or technical reasoning to support the finding that the predetermined period of time during which packets can be distributed or transmitted teaches or suggests an average time that each core takes to process the received packets. Further, the Examiner has not relied on, or

cited any disclosures from Panigrahy to cure this deficiency. *See* Final Act. 4–15. We decline to resort to impermissible speculation or unfounded assumptions or rationales to cure the deficiencies in the factual bases of the rejection before us. *In re Warner*, 379 F.2d 1011, 1017 (CCPA 1967).

Accordingly, we are constrained by the record before us to reverse the Examiner's rejection of claim 5. *See* Appeal Br. 19–20. We also reverse the Examiner's rejection of claim 6, which depends from claim 5 and includes the same deficiency. *See* Appeal Br. 20–21.

In conclusion, having considered the Examiner's rejections of claims 1–18, 20, and 21, we reverse the Examiner's rejection of claims 5 and 6 and sustain the Examiner's rejections of claims 1–4, 7–18, 20, and 21.

DECISION

We affirm the Examiner's decision rejecting claims 1–4, 7–18, 20, and 21.

We reverse the Examiner's decision rejecting claims 5 and 6.

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1). *See* 37 C.F.R. §§ 41.50(f), 41.52(b).

AFFIRMED-IN-PART