

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

DISH NETWORK L.L.C.,
DISH TECHNOLOGIES L.L.C., and
SLING TV L.L.C.,

v.

SOUND VIEW INNOVATIONS, LLC,
Patent Owner.

IPR2020-01041
Patent 6,725,456 B1

Before DEBRA K. STEPHENS, DANIEL J. GALLIGAN, and
JOHN A. HUDALLA, *Administrative Patent Judges*.

STEPHENS, *Administrative Patent Judge*.

JUDGMENT
Final Written Decision
Determining No Challenged Claims Unpatentable
35 U.S.C. § 318(a)

I. INTRODUCTION

A. *Background and Procedural History*

DISH Network L.L.C., DISH Technologies L.L.C., and Sling TV L.L.C. (collectively, “Petitioner”) filed a petition for *inter partes* review (Paper 2 (“Pet.” or “Petition”)) challenging claim 13 of U.S. Patent

6,725,456 B1 (Ex. 1001 (“’456 Patent”)). Sound View Innovations, LLC (“Patent Owner”) timely filed a Preliminary Response (Paper 8 (“Prelim. Resp.”)). On the record before us, in our Decision to Institute, we determined Petitioner had established a reasonable likelihood that the relied upon references taught claim 13, the only challenged claim (Paper 13, “Dec.”). More specifically, based on our review of the record, we concluded that Petitioner was reasonably likely to prevail in demonstrating claim 13 is not patentable (*id.*). Thus, we instituted *inter partes* review of claim 13.

Patent Owner then filed a response (Paper 22 (“PO Resp.”); Petitioner filed a reply to Patent Owner’s response (Paper 28 (“Pet. Reply”); and Patent Owner filed a sur-reply (Paper 35 (“PO Sur-reply”)).

An Oral Hearing was held October 19, 2021, a transcript of which has been entered (Paper 40 (“Tr.”)).

B. Real Parties in Interest

Patent Owner states that Sound View Innovations, LLC and Sound View Innovation Holdings, LLC are the real-parties-in-interest (Paper 3, 1).

Petitioner states DISH Network L.L.C., DISH Technologies L.L.C., Sling TV L.L.C., Sling TV Holding L.L.C., DISH Network Corporation, and Cloudera, Inc. are the real parties-in-interest (Pet. xiii; Paper 6, 1).

C. Related Matters

As required by 37 C.F.R. § 42.8(b)(2), both parties identify various matters related to the ’456 Patent:

Ex parte reexamination Control No. 90/014,560 requested by Unified Patents, LLC (Notice of Intent to Issue *Ex Parte* Reexamination Certificate confirming patentability of claim 13 and terminating the reexamination, October 1, 2021) (Pet. xiii; Paper 3, 1–2; Paper 24, 1; Paper 36, 1);

Sound View Innovations, LLC v. DISH Network LLC et al., No. 1-19-cv-03707 (D. Colo.); and

Sound View Innovations, LLC v. Sling TV LLC, No. 1-19-cv-03709 (D. Colo).

Petitioner and Patent Owner set forth the following former proceedings involve the '456 Patent:

Sound View Innovations, LLC v. Walmart Inc. et al., No. 1-19-cv-00660 (D. Del.) (terminated Aug. 21, 2020);

Walmart Inc. et al. v. Sound View Innovations, LLC, IPR2020-00818 (PTAB) (terminated Aug. 25, 2020, Paper 9);

Sound View Innovations, LLC v. Delta Air Lines, Inc., No. 1-19-cv00659 (D. Del.) (terminated Nov. 18, 2020, Dkt. No. 166);

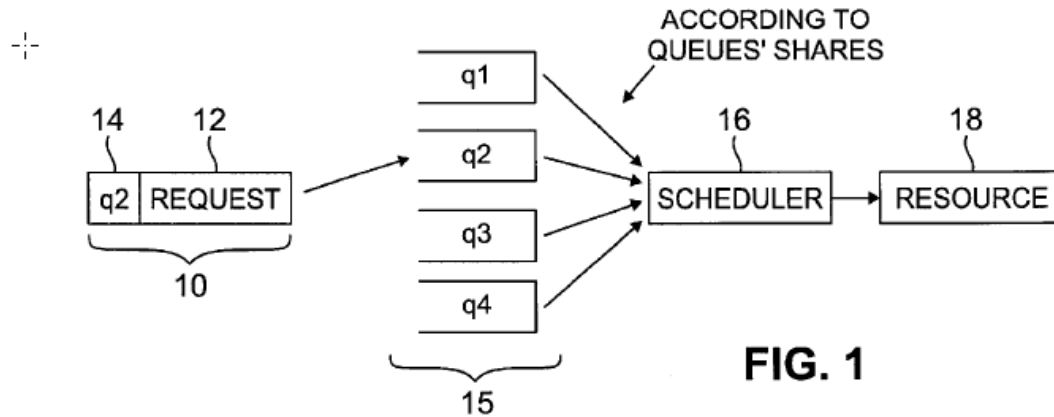
Cigna Corp. et al. v. Sound View Innovations, IPR2020-00924 (PTAB) (Paper 11) (terminated Nov. 24, 2020); and

Sound View Innovations, LLC v. Cigna Corp. et al., No. 1-19-cv-00964 (D. Del.) (terminated Nov. 18, 2020, Dkt. No. 166) (Pet. xiii; Paper 3, 1–2; Paper 24, 1–2).

D. The '456 Patent (Ex. 1001)

The '456 Patent, titled “Methods and Apparatus for Ensuring Quality of Service in an Operating System,” issued April 20, 2004 (Ex. 1001, codes (45), (54)). The '456 Patent describes “techniques for ensuring a desired quality of service (QoS) for an application running on an operating system” (Ex. 1001, 3:14–16, 4:25–26). In particular, the '456 Patent describes techniques to allocate and reserve computing resources, e.g., central processing unit (CPU), memory, and disk or network bandwidth between competing requests for those resources, in order to guarantee access to those resources (*see id.* at 1:14–20, 4:61–62).

The '456 Patent further discloses that Figure 1 “illustrates the manner in which requests are tagged with a queue identifier” (*id.* at 4:3–4):



As shown in Figure 1, “every request arriving at a given one of the . . . schedulers must specify a queue, and the given scheduler apportions resources to each queue based on the queue’s share of that resource” (Ex. 1001, 5:1–4). “The particular request 10 includes the request information 12 along with an identifier 14 of the particular queue to which the request will be directed” (*id.* at 5:4–7, Fig. 1). Figure 1 illustrates four different queues: q1, q2, q3, and q4 (*id.* at 5:7–8, Fig. 1). “A scheduler 16 submits the requests from the queues 15 to a resource 18 according to the queues’ shares of that resource” (*id.* at 5:8–10, Fig. 1).

E. Challenged Claim

Challenged claim 13 is independent and reproduced below.

13. A method of ensuring a particular quality of service for an application in a computer system, the method comprising the steps of:

utilizing an application programming interface of an operating system to establish one or more quality of service guarantees that correspond to a reference to an object; and

providing a particular quality of service to a request in accordance with the one or more quality of service guarantees that correspond to one or more object references used in the request;

wherein the quality of service guarantees comprise resource reservations, each specifying a portion of a resource set aside for exclusive use by one or more processes;

wherein the resource reservations are organized hierarchically such that each resource reservation *r* may have at most one parent and one or more siblings and children, and associated with *r* is a weight that specifies how *r* shares the resources of *r*'s parent with *r*'s siblings; and

wherein associated with each resource reservation *r* is a minimum amount of resources that *r* receives from its parent *p*, such that the minimum amount of resources associated with *p* is at least equal to the sum of the minimum amount of resources associated with each of *p*'s children

(Ex. 1001, 14:61–15:20).

F. References Relied Upon

Reference	Exhibit
Durand, US 6,338,072 B1, issued Jan. 8, 2002 (hereinafter, “Durand”).	1006
Pawan Goyal et al., <i>A Hierarchical CPU Scheduler for Multimedia Operating Systems</i> , USENIX 2nd Symposium on Operating Systems Design and Implementation (1996) (hereinafter, “Goyal”).	1007
Jon C. R. Bennett and Hui Zhang, <i>Hierarchical Packet Fair Queueing Algorithms</i> , IEEE/ACM Transactions on Networking (Vol. 5, No. 5 1997) (hereinafter, “Bennett”).	1008

(Pet. 1).

Petitioner also relies on the Declaration and the Reply Declaration of Dr. Kevin Negus (Exs. 1002, 1090).

Patent Owner relies on the Declaration of Dr. Mark T. Jones
(Ex. 2004).

G. Prior Art and Asserted Grounds

Petitioner asserts that claim 13 is unpatentable on the following
grounds:

Claim(s) Challenged	35 U.S.C. §	Reference(s)/Basis
13	103	Durand, Bennett
13	103	Goyal, Bennett

(Pet. 1).

II. ANALYSIS

A. Legal Standards

A patent claim is unpatentable under 35 U.S.C. § 103(a) if the differences between the claimed subject matter and the prior art are such that the subject matter, as a whole, would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains (*KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007)). The question of obviousness is resolved on the basis of underlying factual determinations including (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of ordinary skill in the art; and (4) any secondary considerations, if in evidence (*Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966)).

B. Level of Ordinary Skill in the Art

The level of skill in the art is a factual determination that provides a primary guarantee of objectivity in an obviousness analysis (*Al-Site Corp. v. VSI Int’l Inc.*, 174 F.3d 1308, 1324 (Fed. Cir. 1999) (citing *Graham v. John*

Deere Co., 383 U.S. 1, 17–18 (1966); *Ryko Mfg. Co. v. Nu-Star, Inc.*, 950 F.2d 714, 718 (Fed. Cir. 1991))).

Petitioner asserts that a person of ordinary skill in the art (“POSITA”) pertaining to the ’456 Patent at the relevant time would have been someone with at least a Bachelor of Science in at least one of Electrical Engineering, Computer Engineering, Computer Science, or a related field, as well as three to four years of experience in implementing resource reservations for at least UNIX-based computing systems, or a Master’s degree in Electrical Engineering, or an equivalent field, as well as two years of experience in implementing resource reservations for at least UNIX-based computing systems.

(Pet. 7–8 (citing Ex. 1002 ¶¶ 27–31)).

Patent Owner does not contest Petitioner’s assertion (*see generally* PO Resp.).

The prior art itself demonstrates the level of ordinary skill in the art at the time of the invention (*see Okajima v. Bourdeau*, 261 F.3d 1350, 1355 (Fed. Cir. 2001) (explaining that specific findings regarding ordinary skill level are not required “where the prior art itself reflects an appropriate level and a need for testimony is not shown” (quoting *Litton Indus. Prods., Inc. v. Solid State Sys. Corp.*, 755 F.2d 158, 163 (Fed. Cir. 1985))).

We regard Petitioner’s formulation of the level of skill as consistent with the prior art before us (*see Okajima* 261 F.3d 1355). Based on the present record, including the disclosure in the ’456 Patent, we apply Petitioner’s definition of the level of ordinary skill in the art. We determine Petitioner’s asserted level of skill comports with the qualifications a person would have needed to understand and implement the teachings of the ’456 Patent and the prior art of record (*cf. Okajima*, 261 F.3d at 1355 (the prior art itself may reflect an appropriate level of skill in the art)). As we

determine the four instances of “at least” in Petitioner’s formulation introduce ambiguity, we remove these terms from our definition. Accordingly, we determine an ordinarily skilled artisan would have a Bachelor of Science in one of Electrical Engineering, Computer Engineering, Computer Science, or a related field, as well as three to four years of experience in implementing resource reservations for UNIX-based computing systems, or a Master’s degree in Electrical Engineering, or an equivalent field, as well as two years of experience in implementing resource reservations for UNIX-based computing systems.

C. Claim Construction

We construe claim terms according to the standard set forth in *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312–17 (Fed. Cir. 2005) (en banc) (37 C.F.R. § 42.100(b) (2019)). Under *Phillips*, claim terms are accorded “their ordinary and customary meaning” (*Phillips*, 415 F.3d at 1312). “[T]he ordinary and customary meaning of a claim term is the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention” (*id.* at 1313). “Importantly, the person of ordinary skill in the art is deemed to read the claim term not only in the context of the particular claim in which the disputed term appears, but in the context of the entire patent, including the specification” (*id.*). An inventor may rebut that presumption by providing a definition of the term in the specification with reasonable clarity, deliberateness, and precision (*In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994)). In the absence of such a definition, limitations are not to be read from the specification into the claims (*In re Van Geuns*, 988 F.2d 1181, 1184 (Fed. Cir. 1993)).

We determine that it is not necessary to provide an express interpretation of any terms of claim 13 (*Vivid Techs., Inc. v. Am. Sci. & Eng'g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999) (only terms that are in controversy need to be construed, and only to the extent necessary to resolve the controversy); *Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017) (applying *Vivid Techs.* in the context of an *inter partes* review)).

D. Asserted Obviousness over Durand and Bennett

Petitioner contends the combination of Durand and Bennett renders claim 13 obvious (Pet. 33–57).

1. References

a) Durand (Ex. 1006)

Durand is a patent titled “Device and Process for Dynamically Controlling the Allocation of Resources in a Data Processing System” (Ex. 1006, code (54)). Durand describes a manner of “controlling the allocation of resources in a data processing system” by “modify[ing] the priority associated with the jobs in progress,” i.e., executed computer processes or programs (*id.* at 1:8–10, 1:65–2:1, 3:4–6). For example, “[m]ore or fewer resources should be allocated to [a process] as a function of [its] importance and as a function of the load on the system” (*id.* at 4:64–66).

Durand uses “dimensions” to allocate resources and modify the priority of processes (*see id.* at 4:64–4:67). “A dimension is a set of currently executed processes which have the same importance” (*id.* at 4:67–5:3). For example, “processes [that] are started by the normal users directly in the background . . . belong to the dimension ‘MISC’” and “processes [that] are started by means of a script . . . belong to [another] dimension

‘BATCH’” (*id.* at 5:3–12). Further, “[w]hen [a] process has an ancestor whose command name and user are associated with a dimension, the process belongs to this dimension” (*id.* at 5:35–40).

Dimensions have respective “relative weight[s]” which are “used to manage the priority of the processes comprising the dimension” (*id.* at 5:16–19). As such, the “priority of all the processes belonging to a dimension will vary accordingly” and “processes controlled by the dimensions will always have resources allocated as a function of their relative weight” (*id.* at 5:19–26, 5:40–42). In an exemplary system with dimensions MYDIM and BATCH, MYDIM can be assigned a relative weight of 2 and BATCH can be assigned a relative weight of 1 (*id.* at 6:21–27). Based on those respective relative weights, “[w]hen the system is heavily loaded,” and “the processes of the dimensions ‘MYDIM’ and ‘BATCH’ are using 90% of the central processing unit (CPU), the priority of the processes of the dimension ‘MYDIM’ will be modified in one direction or another so that the percentage of CPU used by the dimension ‘MYDIM’ will be closer to 60%” (*id.* at 6:25–33).

Additionally, Durand “allow[s] the pre-allocation and dynamic control of resources” via “an application program interface 34 [that] communicates with [local resource manager] demon 3” (*id.* at 3:65–4:2). The local resource manager “application program interface 34 is associated with a graphical user interface ‘LRM gui’ 32” (*id.* at 6:34–36). The LRM gui associated with the application program interface allows users “to define the name of [a] dimension” and “to define the relative weight” of the dimension (*id.* at 7:32–37, Fig. 2E).

b) Bennett (Ex. 1008)

Bennett is a paper titled “*Hierarchical Packet Fair Queueing Algorithms*” and authored by Jon C. R. Bennett and Hui Zhang (Ex. 1008, 675). Bennett describes that “[f]uture integrated services networks will support multiple service classes that include real-time service, best-effort service, and others” (*id.*). For example, “where there are 11 agencies sharing the output link,” i.e., a computer network resource, an “administrative policy dictates that Agency A1 gets at least 50% of the link bandwidth whenever it has traffic” (*id.* at 675 (col. 1)). Addressing that goal, Bennett describes an “idealized Hierarchical Generalized Processor Sharing (H-GPS) model to simultaneously support guaranteed real-time, rate-adaptive best-effort, and controlled link-sharing services” (*id.*).

2. Analysis of Independent Claim 13

Our disposition of this ground turns on certain inconsistencies in Petitioner’s mapping of claim 13 to the prior art. We focus our analysis on the limitations of claim 13 for which we find these inconsistencies.

Petitioner relies on Durand to teach “utilizing an application programming interface of an operating system to establish one or more quality of service guarantees that correspond to a reference to an object” (Pet. 36–41 (citing Ex. 1002 ¶¶ 230–241; Ex. 1006, 2:54–64, 3:41–4:2, 5:12–21, 6:34–48, 7:9–22, 7:25–37, 7:50–52, 7:61–64, Figs. 1, 2D–2E, 2G)).

Petitioner additionally relies on Durand to teach “providing a particular quality of service to a request in accordance with the one or more quality of service guarantees that correspond to one or more object references used in the request” (*id.* at 41–44 (citing Ex. 1002 ¶¶ 244–256;

Ex. 1006, 5:57–5:62, 6:14–6:33, 7:65–8:5, Fig. 2H; Ex. 1009, 63, 93)).

Petitioner further asserts, “Bennett discloses providing a particular QoS to traffic streams associated with a service share (a ‘weight’) where the traffic streams receive service ‘in proportion to their relative service shares’” (Pet. 44 (citing Ex. 1002 ¶¶ 258–259; Ex. 1008, 675)).

Patent Owner points out the claim recites “[QoS] guarantees that correspond to a reference to an object’ are ‘establish[ed]’ [and] . . . a particular quality of service is ‘provide[ed]’ ‘to a request’ ‘in accordance’ with the QoS guarantee(s)” (PO Resp. 25). Moreover, Patent Owner argues, the claim “requires that the ‘request’ be, not just any request, but, rather, one in which ‘one or more object references [is] **used in the request**” (*id.* at 27–28). Thus, Patent Owner contends, the claim “requires that a ‘particular quality of service’ be ‘provid[ed]’ ‘**to a request**’ . . . however, Petitioner fails to identify providing a ‘particular quality of service **to a request**’” (*id.*). Patent Owner focuses on the “plain meaning” of the claim limitations: “the plain meaning [of the limitation,] in accordance with the specification, is that a particular quality of service be ‘provid[ed]’ ‘**to a request,**’ not ‘**in response** to a request” (*id.* at 28 (citing Ex. 2004 ¶¶ 116–121)).

According to Patent Owner, Petitioner is “essentially” reading the “providing” limitation “out of the claim” (*id.* at 28, 30). More specifically, Patent Owner argues, Petitioner is collapsing the “utilizing” and “providing” limitations “into a single limitation[,] rel[ying] on the exact same proof for both”: Durand’s relative weights by dimension (*id.* at 30 (citing Ex. 2004 ¶¶ 122–124; Pet. 44)). “Petitioner, however, never attempts to identify what the ‘request’ is or demonstrate that it is being ‘provide[d]’ a ‘particular quality of service’” (*id.* at 31).

Patent Owner also argues that Petitioner fails to show that Durand’s relative weights (i.e., the alleged “quality of service guarantee”) correspond to either Durand’s file descriptors or process identifier (PID) value (i.e., the alleged “reference to an object”) in accordance with the recitation in claim 13 (PO Resp. 34–42; PO Sur-reply 7–8). Patent Owner disputes that a “file descriptor” is even disclosed by Durand and that Petitioner’s contentions relative to any such file descriptor are based on conclusory expert testimony (PO Resp. 36–38). Patent Owner also argues that PIDs are associated with a particular command (as shown in Durand’s Figure 2G) and not relative weights (as shown in Durand’s Figures 2E and 2H) (PO Resp. 38–42 (reproducing Ex. 1006, Figs. 2E, 2G, 2H) (citing Ex. 2004 ¶¶ 134–139)). Instead, Patent Owner argues that “Durand’s weights correspond to the dimension to which they apply” (PO Sur-reply 8 (reproducing Ex. 1006, Fig. 2H)).

For the reasons explained below, we agree with Patent Owner.

Petitioner contends,

Durand discloses an LRM API in a UNIX system (“utilizing an application programming interface of an operating system”) that adjusts execution of priorities of jobs within dimensions based upon relative weights of the dimensions (thereby “establish[ing] one or more quality of service guarantees”) where the guarantees correspond to a relative weight for each dimension in a file identified by using a file descriptor and/or a process indirectly identified by a UNIX process identifier or PID field value (“correspond to a reference to an object”)

(Pet. 40–41 (citing Ex. 1002 ¶¶ 237–241; Pet. 8–9)). Thus, Petitioner indicates Durand’s relative weight for each dimension in a file teaches the recited “object” and “a file descriptor and/or a process indirectly identified

by a UNIX process identifier or PID field value” teaches the recited “reference.”

Petitioner also relies on Durand to teach “providing a particular quality of service to a request in accordance with the one or more quality of service guarantees that correspond to one or more object references used in the request,” as recited in claim 13 (Pet. 41–44). In particular, Petitioner asserts Durand’s two values: “*Pdc*” (“the current percentage of resources used by all the processes belonging to the dimension”) and “*Pde*” (“the percentage of resources which should be used by the dimension”) are “used to provide quality of service to requests” (Pet. 41 (citing Ex. 1002 ¶ 246; Ex. 1006, 5:57–62)). Petitioner further asserts, “Durand modifies the NICE parameter [– a parameter associated with processes that is ‘inversely proportional to the priority’ of each process –] by comparing the difference between the *Pdc* and *Pde* values with a predetermined threshold” (*id.* at 42 (Ex. 1002 ¶ 250; Ex. 1006, 6:14–21)).

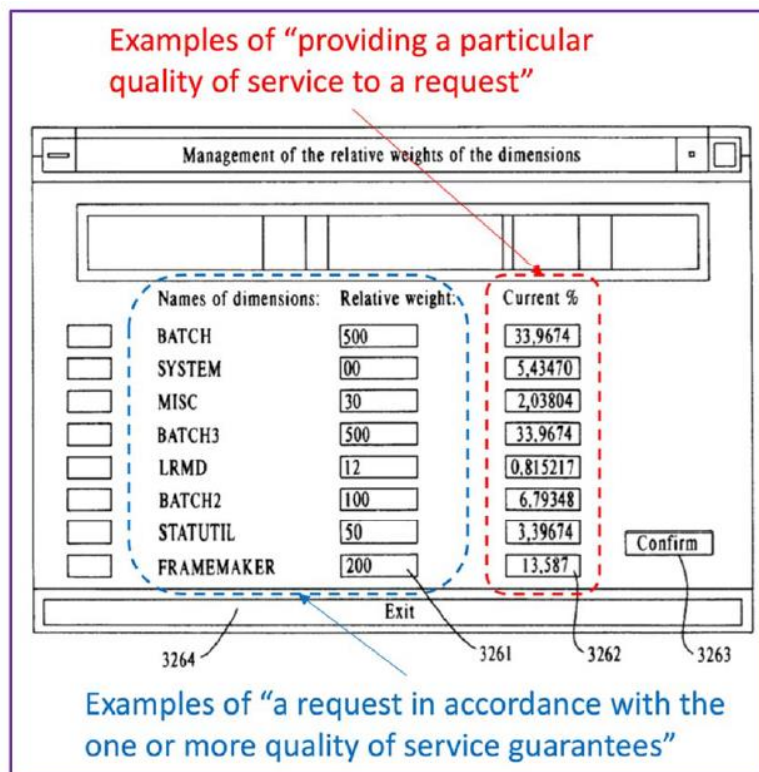
Petitioner points to Durand’s example as support for modifying of the NICE parameter:

When the value “NICE” increases, the priority decreases. For example, in a system having a dimension “MYDIM,” the latter can be defined in the system with a relative weight of 2, a relative weight of 1 having been declared for the dimension “BATCH.” When the system is heavily loaded, the priorities of the processes belonging to the dimension “MYDIM” will be modified in the following way:

If the processes of the dimensions “MYDIM” and “BATCH are using 90% of the central processing unit (CPU), the priority of the processes of the dimension “MYDIM will be modified in one direction or another so that the percentage of CPU used by the dimension “MYDIM” will be closer to 60%

(Ex. 1006, 6:20–33).

Next Petitioner asserts Durand “discloses an exemplary display illustrating ‘the management of the relative weights for each dimension’ that allows ‘the user to enter, in each box associated with each dimension, a new relative weight value and to display, in the box showing the current percentage, the modification of the current percentage” (*id.* at 42 (citing Ex. 1002 ¶ 253; Ex. 1006, 7:65–8:5, Fig. 2H)). Petitioner provides an annotated version of Durand’s Figure 2H, which is reproduced below.



(Ex. 1006, Fig. 2H (annotated by Petitioner); Pet. 42; Pet. Reply 12). Figure 2H of Durand depicts a display window for managing relative weights by dimension (Ex. 1006, 7:65–8:2). In the display window of Figure 2H, “the user [may] enter, in each box associated with each dimension, a new relative weight value and to display, in the box showing the current percentage, the modification of the current percentage” (Ex. 1006, 8:2–5, Fig. 2H). Petitioner then contends that “Durand discloses

a user can request a quality of service by modifying the relative weights for each dimension, using the *Pdc* and *Pde* values to adjust the NICE parameter for processes within each dimension,” and further, that “Durand’s UNIX-based system guarantees access by allocating a particular percentage of resources, for example, the ‘percentage of CPU,’ for use by the dimensions” (Pet. 43 (citing Ex. 1002 ¶ 254; Ex. 1006, 7:65–8:5, Fig. 2H)). Without providing any supporting citations to Durand, Dr. Negus states that

Durand discloses that a “**user**” can “**enter**” (or “**request**”) “**values**” for “**relative weights for each dimension**” such that by computing the parameters “**Pdc**” and “**Pde**” and adjusting the “**NICE**” parameter for “**processes**” within each “**dimension**”, then Durand’s UNIX-based system causes a particular “**current percentage**” of “**resources**” such as “**percentage of CPU**” to be “**used by the dimension**”, or hence discloses “**providing a particular quality of service**” (the particular “**current percentage**” of “**resources**” such as “**percentage of CPU**” to be “**used by the dimensions**”) to a “**request in accordance with the one or more quality of service guarantees**” (the overall set of “**values**” for “**relative weights for each dimension**”)

(Ex. 1002 ¶ 254).

Thus, Petitioner contends the request is the modification of the relative weights for each dimension.

Petitioner further asserts, “Durand discloses the values of the relative weights for each dimension are stored in a file, which necessarily also discloses the use of ‘a file descriptor as an argument to specify the file’” (Pet. 43–44 (citing Ex. 1002 ¶ 255; Pet. 36–41)). Thus, this storing of the relative weight occurs after a user modifies the relative weight for the dimension.

Based on this disclosure, Petitioner asserts “Durand discloses adjusting the NICE parameter for processes within dimensions based upon

the user-defined relative weights of the dimensions” which teaches “providing a particular quality of service to a request in accordance with the one or more quality of service guarantees” (Pet. 44). Petitioner further asserts Durand discloses these “guarantees correspond to relative weights stored in files identified using respective file descriptors and/or processes indirectly identified via a UNIX process identifier or PID field value” and, thus, Durand teaches the quality of service guarantees “correspond to one or more object references used in the request” (*id.*).

The request set forth by Petitioner, therefore, is Durand’s “user-defined relative weights of the dimensions” according to Petitioner’s assertion (Pet. 44). However, input to this “user-defined weight of the dimension” does not use what Petitioner identifies as the “one or more object references” — “the file descriptor and/or a process indirectly identified by a UNIX process identifier or PID field value” (*id.*). Rather, the user chooses what Petitioner contends is the object by its reference, e.g., “BATCH,” and then enters the desired relative weight (*see* Ex. 1006, Fig. 2H).

Thus, Durand’s object reference (as identified by Petitioner) is not used *in* the request; rather, according to Petitioner’s mapping, the user selects the object and then inputs the request (user-defined relative weight) for that object.

Further, the claim recites that a particular quality of service is provided to a request *in accordance with the one or more quality of service guarantees* (Ex. 1001, 15:1–4). The “one or more quality of service guarantees” is established “utilizing a resource reservation application programming interface” (*id.* at 14:64–67). We agree with Patent Owner’s contention that Petitioner essentially collapses these limitations into a single

limitation and relies on the same proof for both (PO Resp. 30). According to Petitioner's mapping, entering the user-defined relative weight is setting a quality of service guarantee. But, as noted by Patent Owner, Petitioner maps Durand's weights for each dimension to the recited "quality of service guarantee" (PO Sur-reply 8). Dr. Negus states, "Durand discloses . . . '**one or more quality of service guarantees**' in the form of '**adjusting the execution priorities of the jobs of each dimension as a function of the relative weights of the dimensions**' as an example of 'conventional proportional share scheduling' that the '456 Patent describes as providing '**QoS guarantees**'" (Ex. 1002 ¶ 230). Entering a user-defined relative weight does not teach "providing a quality of service to a request in accordance with the [] quality of service guarantees" established utilizing an application programming interface.

As such, Petitioner has not shown Durand teaches "utilizing an application programming interface of an operating system to establish one or more quality of service guarantees that correspond to a reference to an object" and "providing a particular quality of service to a request in accordance with the one or more quality of service guarantees that correspond to one or more object references used in the request," as recited in claim 13 (Ex. 1001, 14:64–15:4).

Petitioner additionally asserts, "Bennett discloses a 'hierarchical packet scheduling' quality of service solution shared across service classes, where service is distributed to processes in proportion to relative service shares and based on a guaranteed service rate for each node in the service hierarchy" (Pet. 41 (citing Ex. 1002 ¶¶ 242–243; Ex. 1008, 675, 684, Fig. 3)). Petitioner further asserts, "Bennett discloses providing a particular QoS to traffic streams associated with a service share (a 'weight') where the

traffic streams receive service ‘in proportion to their relative service shares’” (*id.* at 44 (citing Ex. 1002 ¶¶ 258–259; Ex. 1008, 675)).

However, Petitioner has not set forth with specificity where Bennett teaches the request uses the one or more object references. Indeed, Petitioner has not explained how these two teachings of Bennett disclose the recited limitations with sufficient specificity or how one of ordinary skill in the art would modify Durand with these teachings of Bennett to achieve the recited limitations.

Accordingly, Petitioner has not established that the combination of Durand and Bennet teaches the subject matter of claim 13. Therefore, based on the entire trial record, we determine Petitioner has not established by a preponderance of the evidence that the subject matter of claim 13 would have been obvious over Durand and Bennet under 35 U.S.C. § 103.

E. Asserted Obviousness over Goyal and Bennett

Petitioner contends the combination of Goyal and Bennett renders independent claim 13 obvious (Pet. 1).

1. Reference - Goyal (Ex. 1007)

Goyal is a paper titled “A Hierarchical CPU Scheduler for Multimedia Operating Systems” and authored by Pawan Goyal, Xingang Guo, and Harrick M. Vin (Ex. 1007). Goyal describes the need for “a multimedia computing environment” having “an operating system framework that: (1) enables different schedulers to be employed for different application classes, and (2) provides protection between the various classes of applications” (*id.* at Abstract). Goyal further describes that “[t]he requirements for supporting different scheduling algorithms for different applications as well as

protecting application classes from one another leads naturally to the need for *hierarchical partitioning* of CPU bandwidth” hierarchical partitioning specified through a tree structure in which “[e]ach thread in the system belongs to exactly one leaf node [and e]ach leaf node represents an aggregation of threads” (Ex. 1007, 3, § 2).

2. *Analysis of Independent Claim 13*

Our disposition of this ground turns on an apparent gap in Petitioner’s unpatentability contentions for claim 13. As above, we focus our analysis on the limitations of claim 13 related to this gap in Petitioner’s contentions.

Petitioner relies on Goyal and Bennett to teach “utilizing an application programming interface of an operating system to establish one or more quality of service guarantees that correspond to a reference to an object” (Pet. 60–63 (citing Ex. 1002 ¶¶ 126–127, 344, 346, 348–360; Ex. 1007, Abstract, §§ 1–4, 6, Fig. 4; Ex. 1008, 675, 684, Fig. 3)). Specifically, Petitioner asserts Goyal’s Quality of Service (“QoS”) manager “uses an API to hierarchically allocate resources to [computing] applications” (*id.* at 60). According to Petitioner, Goyal implemented this “hierarchical CPU allocation framework” that uses “a hierarchical structure (a tree), where each node in the tree is associated with a parent node, a weight, and ‘has a unique identify and a name similar to a UNIX filename’” (*id.* (citing Ex. 1002 ¶¶ 348–349 (citing Ex. 1007 § 4))). Petitioner further asserts, “Goyal’s scheduling structure is created using a system call, in conjunction with the QoS manager, that ‘creates a node with the given name as a child of node parent in the scheduling structure and returns the identifier of the new node’” (*id.* at 61 (citing Ex. 1002 ¶¶ 350, 354 (citing Ex. 1007 § 4))).

Petitioner relies on this teaching to contend “Goyal discloses using system calls and/or a QoS manager to connect user applications to a hierarchical scheduler interface within a UNIX system to disclose “utilizing an application programming interface of an operating system” (*id.* at 63 (citing Ex. 1002 ¶¶ 357–60 (citing Ex. 1002 ¶¶ 188–90, 249, 340–44))).

Petitioner asserts, “[l]eaf nodes in Goyal’s scheduling structure are also provided a pointer to a ‘scheduling function’”; “Goyal’s scheduling structure . . . is created using a system call”; and, further, “system calls are used to remove, move, and administer nodes in the scheduling structure” (*id.* at 61 (citing Ex. 1002 ¶ 351 (citing Ex. 1007 § 4))). Petitioner contends, “Goyal teaches the scheduling structure also provides ‘bounds on minimum throughput and maximum delay observed by nodes’ via ‘admission control only if the applications desire a certain guaranteed minimum CPU bandwidth’” (*id.* at 62–63 (citing Ex. 1002 ¶ 356 (citing Ex. 1007 §§ 2–3, 6))). Petitioner thus contends that Goyal’s disclosure of “invoking a scheduling function based upon weights to provide guaranteed minimum CP[U] bandwidth” teaches the recited “establish one or more quality of service guarantees” and the disclosure that “each node in a scheduling structure is named similar to a UNIX filename to indirectly reference the associated process and contains a pointer to a function that selects the next thread to be executed” teaches the recited “correspond to a reference to an object” (Pet. 63 (citing Ex. 1002 ¶¶ 357–60 (citing Ex. 1002 ¶¶ 188–190, 249, 240–344))).

Petitioner further relies on the combination of Goyal and Bennett to teach “providing a particular quality of service to a request in accordance with the one or more quality of service guarantees that correspond to one or more object references used in the request” (Pet. 63–67 (citing Ex. 1002

¶¶ 363–379 (citing Ex. 1007 §§ 2–3, 5.3, 5.4, 6, Figs. 2, 6; Ex. 1008, 675, 684, Fig. 3; Ex. 1009, 37–38, Fig. 2.2))).

More specifically, Petitioner contends, “Goyal discloses an operating system that hierarchically partitions CPU bandwidth for applications” thereby disclosing the recited “one of more quality of service guarantees” (Pet. 63–64 (citing Ex. 1002 ¶ 363; Pet. 60–63)). Petitioner next relies on Goyal’s teaching of “organizing the ‘one or more quality of service guarantees’ in a scheduling structure comprising nodes, where each node is created using a system call that provides an identifier of the node and nodes have a UNIX-like filename identifying the node as a child of a parent node” to teach the recited “one or more object references” (*id.* at 64 (citing Ex. 1002 ¶ 364 (citing Ex. 1009, 37–38, Fig. 2.2))).

Petitioner further contends Goyal’s QoS manager “‘allocates resources as per the requirements of [the] application’ and uses an ‘admission control algorithm . . . to determine if the request can be satisfied’ . . . [and] provides a resource allocation algorithm for distributing processor bandwidth from a parent node to child nodes based on weights associated with the child nodes” (*id.* at 64 (citing Ex. 1002 ¶¶ 365–366 (citing Ex. 1007 § 1), ¶ 367 (citing Ex. 1007 § 2))).

According to Petitioner, the scheduling of threads in Goyal occurs hierarchically, “where parent nodes schedule[] child node[s], until a node without children schedules a thread for execution” (*id.* at 64 (citing Ex. 1002 ¶ 368 (citing Ex. 1007 § 2))). Petitioner further contends, “Goyal teaches the resources allocation algorithm provides ‘bounds on minimum throughput and maximum delay observed by nodes’ in the scheduling structure via ‘admission control only if the applications desire a certain guaranteed

minimum CPU bandwidth”” (*id.* at 66 (citing Ex. 1002 ¶ 372 (citing Ex. 1007 §§ 2–3, 6))). Petitioner thus asserts,

Goyal discloses providing quality of service guarantees to applications via hierarchical CPU allocation based on the requested weights or desired minimum throughput bounds associated with nodes of a scheduling structure (“providing a particular quality of service to a request in accordance with the one or more quality of service guarantees”) where each of the nodes in the scheduling structure are named similar to UNIX filenames and have weights and unique identifiers (“correspond to one or more object references used in the request”)

(*id.* at 66–67 (citing Ex. 1002 ¶¶ 374–377)).

Patent Owner argues that Petitioner fails to show how Goyal teaches that a particular quality of service is provided to a request or how the quality of service guarantees correspond to one or more object references used in the request (PO Resp. 31–33). We agree with Patent Owner that Petitioner has not identified where Goyal, Bennett, or the combination thereof, teaches the recited “one or more object references used in the request,” as recited in claim 13 (Ex. 1001, 15:1–4).

To summarize the above discussion, for the “utilizing” limitation of claim 13, Petitioner contends:

Goyal discloses using system calls and/or a QoS manager to connect user applications to a hierarchical scheduler interface within a UNIX system (“utilizing an application programming interface of an operating system”) and invoking a scheduling function based upon weights to provide guaranteed minimum CPI bandwidth (to “establish one or more quality of service guarantees”), where each node in a scheduling structure is named similar to a UNIX filename to indirectly reference the associated process and contains a pointer to a function that selects the next thread to be executed (“correspond to a reference to an object”)

(Pet. 63 (citing Ex. 1002 ¶¶ 357–360 (citing Ex. 1002 ¶¶ 188–190, 249, 340–244))). But then for the “providing” limitation of claim 13, Petitioner asserts,

Goyal discloses providing quality of service guarantees to applications via hierarchical CPU allocation based on the requested weights or desired minimum throughput bounds associated with nodes of a scheduling structure (“providing a particular quality of service to a request in accordance with the one or more quality of service guarantees”) where each of the nodes in the scheduling structure are named similar to UNIX filenames and have weights and unique identifiers (“correspond to one or more object references used in the request”)

(*id.* at 66–67 (citing Ex. 1002 ¶¶ 374–377)).

As these passages demonstrate, the Petition does not clearly allege that any application request uses any references to nodes (which Petitioner asserts teaches the recited “object”) (*see* Ex. 1002 ¶ 358).

Further, Petitioner asserts, “Goyal . . . discloses a ‘QoS manager that allocates resources as per the requirements of application’ and uses an ‘admission control algorithm . . . to determine if the request can be satisfied’” (Pet. 64 (citing Ex. 1002 ¶¶ 365–366 (citing Ex. 1007 § 1))). Dr. Negus further explains, “a QoS manager can use a deterministic (statistical admission control algorithm which utilizes the capacity allocated to hard (soft) real-time classes to determine if the **request** can be satisfied, and if so, assign it to the appropriate partition (Ex. 1002 ¶ 366) (bold emphasis added)).

Petitioner does not set forth with sufficient specificity where Goyal teaches requests from applications to provide the quality of service guarantees use “one or more object references.”

Dr. Negus, in his initial Declaration, identifies the “int hsfq_mknod” system call as disclosing “quality of service guarantees that correspond to a reference to an object”:

Goyal discloses to a POSITA that each “**node**” within the “**scheduling structure**” comprises “**a unique identity and a name similar to a UNIX filename**”, wherein such “**name**” derives “**as a child of node parent** in the **scheduling structure**” (for example, “**/best-effort/user1**”, which a POSITA would understand is “**similar to a UNIX filename**” per, for example, ¶¶ 188-190 above), as well as a “**weight**” associated to such “**node**”, which a POSITA would understand to be an example of an “**object**” per the terminology of the ‘456 Patent . . . Goyal further discloses to a POSITA, per my citations noted above, that for the “**system call**” “int hsfq_mknod()” such “**unique identity**” is given by the “**identifier of the new node**” which the “**system call**” “int hsfq_mknod()” “**returns**” when it “**creates a node**”, thereby such “**identifier of the new node**” discloses to a POSITA an example of a “**reference**” to such “**node**” (or “**object**”) per the terminology of the ‘456 Patent

(Ex. 1002 ¶ 358 (citing Ex. 1002 ¶¶ 340–344); PO Resp. 42–43; PO Sur-reply 10; see also Ex. 1002 ¶ 364). Thus, Dr. Negus is correct that the int hsfq_mknod system call will return an identifier of the new node (Ex. 1002 ¶ 358), but Petitioner must show, then, that this identifier is “used in [a] request” based on Petitioner’s mapping of the identifier to the object reference. Petitioner did not do so in the Petition. We credit Dr. Jones’s testimony that “Goyal is scheduling the threads and CPU applications Invoking this system call simply creates a node” (PO Resp. 43 (citing Ex. 2004 ¶ 142)) because Goyal discloses the int hsfq_mknod system call “creates a node with the given name as a child of node parent in the scheduling structure and returns the identifier of the new node” (Ex. 1007 § 4).

In its Reply, as noted by Patent Owner (PO Sur-reply 11), Petitioner offers a new theory, pointing to a different system call of Goyal (Pet. Reply 19–20 (citing Ex. 1090 ¶ 144 (citing Ex. 1002 ¶ 352–354))). Petitioner contends this new argument is to address Patent Owner’s alleged contention that “‘quality of service’ must already be ‘provided’ at the same time when the correspondence to the ‘reference to an object’ is created (Pet. Reply 19 (citing Ex. 1090 ¶ 141)). Thus, in Dr. Negus’ Reply Declaration, Dr. Negus asserts Goyal teaches the recited “‘quality of service guarantees that correspond to a reference to an object” relying on the “int hsfq_move (int from, int to, ...)” which “is used to *move a thread from one leaf node to another*”¹ (Ex. 1090 ¶ 144).

Petitioner’s alleged reliance on the hsfq_move (int from, int to, ...) system call instead of the int hsfq_mknod (char *name, int parent, int weight, int flag, scheduler_id side) system call represents a change in Petitioner’s unpatentability contentions that is presented for the first time in Petitioner’s Reply (see Pet. Reply 19–20). We acknowledge that the Petition mentions a “move” function: “Additionally, system calls are used to remove, move, and administer nodes in the scheduling structure. Goyal also discloses several additional functions to manage the scheduling structure, including removing, moving, and administering nodes.” (Pet. 61 (citing Ex. 1002 ¶¶ 350–352 (citing Ex. 1007 § 4))). But the cited portion of Dr. Negus’ Declaration simply states that there are other system calls, including “int hsfq_parse(char *name, int hint),” “int hsfq_rmnod(int id, int

¹ We note Dr. Negus refers to the system call as “int hsfq_move” (see e.g., Ex. 1002 ¶¶ 144, 352) although Goyal discloses the system call as “hsfq_move” (Ex. 1007 § 4). We consider Dr. Negus’ naming to refer to the same system call as disclosed in Goyal.

mode),” and “int hsfq_admin(int node, int cmd, void *args)” and the system call ““int hsfq_move(int from, int to, ...)’ which ‘is used to move a thread from one leaf node to another’” (Ex. 1002 ¶ 352; *see also id.* ¶ 144 (repeating the same information as paragraph 352)). Additionally, these teachings of Goyal are discussed in conjunction with the limitation “utilizing an application programming interface . . . to establish one or more quality of service guarantees” (*id.* ¶¶ 356, 359). Thus, Dr. Negus’ initial Declaration does not elaborate on the move function, and the Petition does not rely on the “hsfq_move” system call to teach the limitations of claim 13.

Petitioner asserts that this argument in reply is to address “Patent Owner’s new claim construction position” (Pet. Reply 19). We disagree. As outlined above, the Petition does not identify where an object reference is “used in the request,” as specifically recited in claim 13. Petitioner’s new reply argument, therefore, attempts to fill in a gap in Petitioner’s unpatentability contentions from the Petition that was pointed out by Patent Owner. Under the circumstances here, we do not consider this a permissible reply argument (*see* Patent Trial and Appeal Board Consolidated Trial Practice Guide 73 (Nov. 2019), available at <https://www.uspto.gov/sites/default/files/documents/tpgnov.pdf> (“Petitioner may not submit new evidence or argument in reply that it could have presented earlier, e.g. to make out a prima facie case of unpatentability.”))). Thus, we do not consider Petitioner’s newly presented contentions.

Petitioner additionally points to Bennett as disclosing these limitations (Pet. 63, 67). Petitioner, however, does not set forth with specificity how Bennett teaches the limitation, how a combination of Goyal and Bennett would teach the limitation, or why an ordinarily skilled artisan would have combined those teachings.

Accordingly, Petitioner has not established that the combination of Goyal and Bennet teaches the subject matter of claim 13. Therefore, based on the entire trial record, we determine Petitioner has not established by a preponderance of the evidence that the subject matter of claim 13 would have been obvious over Goyal and Bennet under 35 U.S.C. § 103.

III. CONCLUSION

Petitioner has not demonstrated by a preponderance of the evidence that claim 13 of the '456 Patent would have been obvious over Durand and Bennett or over Goyal and Bennett.

IV. ORDER

In consideration of the foregoing, it is hereby:

ORDERED that, on the record before us, Petitioner has failed to show by a preponderance of the evidence that claim 13 of the '456 Patent is unpatentable.

In summary:

Claim	35 U.S.C. §	Reference(s)/Basis	Claims Shown Unpatentable	Claims Not shown Unpatentable
13	103	Durand, Bennett		13
13	103	Goyal, Bennett		13
Overall Outcome				13

IPR2020-01041
Patent 6,725,456 B1

FOR PETITIONER:

Matthew Hindman
PILLSBURY WINTHROP SHAW PITTMAN LLP
matthew.hindman@pillsburylaw.com

Eliot Williams
Ali Dhanani
BAKER BOTTS L.L.P.
eliot.williams@bakerbotts.com
ali.dhanani@bakerbotts.com

FOR PATENT OWNER:

Kenneth Weatherwax
Bridget Smith
Edward Hsieh
Parham Hendifar
Patrick Maloney
Jason Linger
Nathan Lowenstein
LOWENSTEIN & WEATHERWAX LLP
weatherwax@lowensteinweatherwax.com
smith@lowensteinweatherwax.com
hseih@lowensteinweatherwax.com
hendifar@lowensteinweatherwax.com
maloney@lowensteinweatherwax.com
linger@lowensteinweatherwax.com
lowenstein@lowensteinweatherwax.com