

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICROSOFT CORPORATION,
Petitioner,

v.

QOMPLX LLC,
Patent Owner.

IPR2026-00298
Patent 12,143,424

**PETITIONER’S REPLY TO
PATENT OWNER’S PRELIMINARY RESPONSE (“POPR”)**

Petitioner submits this authorized Reply to the POPR (Paper 8). *See* EX3101.

The Barsness compute nodes store “respective first data” representing a “respective portion of a distributed computational graph.” Patent Owner (“PO”) fragments the Petition’s coherent mapping (POPR, 6-13). A holistic read shows that Barsness satisfies this element. The [1.2.a] and [1.2.b] mappings are expressly linked (Pet. 18-19) and show how each compute node stores data in memory, including the PEs, that represents the node’s portion of the operator graph. Pet. 20-24; *id.* 35 (“stored data that identifies the downstream destination”); EX1003, ¶62; EX1004, 8:66-9:9, 10:49-51, FIG. 2. PO’s “backwards” argument (POPR, 12-13) imports a requirement the claims do not recite. The claims require data that “represents a respective portion” of a graph, *not* data configured as a graph structure.¹ The compute nodes satisfy that requirement because each node’s stored PEs/operator data corresponds to its local portion of the operator graph. EX1004, 9:4-9; Pet. 20-24. Consistent with this, the ’424 patent describes pipelines as directed graphs in which transformations are nodes and output messages are edges. EX1001, 9:31-35; Pet. 7-8. Dr. Malek’s analogy (EX2001, ¶47) asks the wrong question because the claims do not require the stored data itself to be a separate graph/map artifact.

The Petition provides a complete theory showing element [9.3], mapping node 130₂’s PEs to the second transformation pipeline. Pet. 65 (citing Sections

¹ If the claims did require a graph, Ground 2 meets that requirement. Pet. 74-75.

VII.B.3.d-VII.B.4 (pp. 37-41)). Contrary to PO's argument (POPR, 15-16), the Petition specifies the relevant data stored in each compute node that identifies the second transformation pipeline, namely (1) PE data and (2) connection information specifying adjacent compute nodes, "as designated by the operator graph" (Pet. 64). For node 130₁, that data identifies the second transformation pipeline by specifying node 130₂ as the downstream destination (Pet. 64-65; EX1004, 9:4-9)—because "the group of PEs for a given compute node" is a transformation pipeline, data identifying node 130₂ identifies the second transformation pipeline executed by the node 130₂'s PEs (Pet. 25 (cited by Pet. 65), 37-41, 64-65). As for the relevant data stored on node 130₂, that data includes data for 130₂'s own PEs (thus "identif[ying]" the second transformation pipeline) and for adjacent-node connections (thus "describ[ing] ..." the related upstream pipeline, i.e., first transformation pipeline). Pet. 64-65. The Petition explains how this analysis shows how the first data on 130₁ and 130₂ satisfies the "describes" language ("Thus..."; Pet. 64-65), and the "identifies" language ("Similarly..."; Pet. 65) of [9.3]. In context, that paragraph is neither conclusory nor incomplete (POPR, 17-18); the full analysis shows element [9.3].

PO also reads the claim too narrowly. POPR, 18-20. Claim 9 does not require a linear "transformation pipeline."² See EX1001, 10:3-7, cl. 10 (reciting "non-linear" pipelines); Pet. 25 n.2. Moreover, the Petition explains how, under its constructions,

² PO raises no similar arguments as to the Ground 2 combination.

“split data elements” (e.g., the outputs from PE4 and PE5 incoming to PE6) “together still comprises a single output stream.” Pet. 22.

Barsness’s management system is the claimed “third computer system.”

PO’s “monitoring system” argument (POPR, 23-26) is misplaced. The Petition maps the claimed third computer system to the Barsness *management system*, which performs the claimed act at least because it “control[s] a stream application running on the compute nodes,” “change[s] the structure of the operator graph” (e.g., moving PEs between nodes), configures compute nodes, and handles “assigning or reassigning” PEs “on different compute nodes.” Pet. 24, 44-48; EX1004, 7:27-37, 10:12-64; *see also* Pet. 53; EX1003, ¶116. PO’s own cites show that the management system spawns new nodes and moves PEs. POPR, 25-26 (citing EX1004, 10:28-39).

PO’s “causal chain” argument (POPR, 26-28) also fails. The management system “execute[s] programming instructions stored in the memory 325,” including its node manager and stream manager, which “setup,” “configure,” “monitor,” and “instruct” PEs in the compute nodes so that the nodes execute the stream application. EX1004, 7:27-37, 9:60-63, 10:12-56; Pet. 24, 44-47. This includes reassigning PEs among nodes. EX1004, 10:56-64; Pet. 24, 53, 58-59. Thus, when the management system executes its instructions to provision, configure, and network a compute node with relevant PEs, it causes that node to execute the PE instructions that apply the corresponding pipeline. Pet. 44-48. PO’s argument that provisioning is insufficient

ignores the management system's control and instruction of node PEs.

A POSITA would have implemented Barsness to store route-map / graph portions on each compute node, as taught by Chakradhar. PO mischaracterizes the Petition by treating the motivation as if it required bodily incorporation of Chakradhar's controller into Barsness. POPR, 31-40. But the Petition's primary rationale is that Barsness requires PE-level routing—each PE emits output “to other PEs in the stream application as described by the operator graph” (EX1004, 9:4-9)—but “does not prescribe a specific internal data structure for locally storing and updating routing information.” Pet. 71; EX1003, ¶148. Chakradhar supplies that implementation detail: “a route-map that specifies the routes on which output messages should be sent after processing incoming messages.” EX1005, [0122]; *id.*, [0053-54] (route-maps “encode tuple-routing information and also the topology link structure”), [0069] (“Every operator chooses a destination ... based on route-maps.”); Pet. 69-76; *see also* EX1005, [0056] (“[A] topology is a graph.”) (quoted by Pet. 74). PO overlooks this motivation; the route-maps are a “natural and efficient” operator-level data structure for carrying out the Barsness routing functions. Pet. 71.

Since Barsness provides a management system that controls the operator graph and compute nodes, use of Chakradhar's route-maps did not require bodily incorporation of the full controller architecture. Pet. 71-72; EX1003, ¶¶148-149; *e.g.*, *MCM Portfolio LLC v. HP Co.*, 812 F.3d 1284, 1294 (Fed. Cir. 2015). PO's cited

cases are distinguishable. Unlike in *Medtronic* and *Illumina* (POPR, 32-33), PO argues neither inoperability nor inefficiency. Nor does PO identify any teaching away, incompatibility, or failure of the cited route-map teachings and benefits.

The combination thus predictably uses Chakradhar's route-maps—expressly “designed for distributed operator-based stream-processing systems” (Pet. 72)—in Barsness's similar distributed operator-based stream-processing system, under Barsness's existing management system control. Pet. 71-76. This meets the motivation demand: a primary reference that specifies a behavior (node-local routing) and a secondary reference that provides a specific data structure for that behavior (local route-maps) in a materially similar system. *Intel Corp. v. PACT XPP Schweiz AG*, 61 F.4th 1373, 1381 (Fed. Cir. 2023). By explaining how route-maps encode topology and direct operator output (POPR, 34-36), PO confirms a POSITA would have wanted to use them to facilitate the node-local routing of Barsness.

The Petition also shows a reasonable expectation of success. Barsness and Chakradhar teach similar distributed operator-based stream-processing systems, Chakradhar's route-maps were designed for the routing called for by Barsness (Pet. 72; EX1005, [0043], [0056], [0069], [0122]), and PO offers no incompatibility or technical obstacle. PO acknowledges that Chakradhar teaches how route-maps “are created, computed, updated, and injected” (POPR, 38), which confirms that POSITAs had a reasonable path to implementation, not the opposite.

Respectfully submitted,

July 8, 2026

By: /Andrew M. Mason/

Andrew M. Mason, Reg. No. 64,034

andrew.mason@klarquist.com

KLARQUIST SPARKMAN, LLP

One World Trade Center, Suite 1600

121 S.W. Salmon Street

Portland, Oregon 97204

Tel: 503-595-5300

Fax: 503-595-5301

Counsel for Petitioner

CERTIFICATE OF SERVICE
IN COMPLIANCE WITH 37 C.F.R. § 42.6 (e)(4)

The undersigned certifies that on July 8, 2026, the PETITIONER’S REPLY TO PATENT OWNER’S PRELIMINARY RESPONSE (“POPR”) was served via email to counsel for the Patent Owner as follows:

Kenneth Weatherwax
Parham Hendifar
Lowenstein & Weatherwax LLP
1016 Pico Boulevard
Santa Monica, CA 90405
Email: weatherwax@lowensteinweatherwax.com
hendifar@lowensteinweatherwax.com
qomplx_IPRs@lowensteinweatherwax.com

By: /Andrew M. Mason/
Andrew M. Mason, Reg. No. 64,034
andrew.mason@klarquist.com
KLARQUIST SPARKMAN, LLP
One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, Oregon 97204
Tel: 503-595-5300
Fax: 503-595-5301

Counsel for Petitioner