



OPEN
Compute Project



FAST-SWITCHING OCS: BENEFITS, TECHNOLOGIES, SYSTEM-LEVEL CONSIDERATIONS, AND EMERGING APPLICATIONS FOR DYNAMIC AI FABRICS

Author (s):

Iwan Kartawira, iPronics

Michael DeMerchant, Lumentum

Swapna Nannapaneni, Lumentum

Spencer Giacalone, Ciena

Shawn Chen, Carnegie Mellon University and AMD

James Burch, Lumotive

Seema Mehta, Intel

Zibo Hu, Triple-Stone Technology

Date: October 19, 2026

Executive Summary

As AI infrastructure continues to scale, traditional packet-switched networks face growing challenges in bandwidth efficiency, congestion management, and workload adaptability. Fast-Switching Optical Circuit Switching (OCS) is emerging as a promising technology that can dynamically reconfigure network connectivity to better match application demands while reducing power consumption and network overhead.

This paper examines the evolution of OCS technologies from millisecond-scale systems deployed today to emerging microsecond- and nanosecond-scale solutions. While advances in switching speed are significant, the paper argues that the true measure of performance is end-to-end reconfiguration latency, which includes control-plane, transceiver, synchronization, and software overheads in addition to the optical switch itself.

The paper reviews key fast-switching technologies, discusses their trade-offs, and explores challenges and opportunities for emerging use cases in AI and HPC environments, including traffic engineering, workload-aware topology reconfiguration, failure recovery, and Mixture-of-Experts (MoE) workloads. By enabling more adaptable and workload-aware network fabrics, fast-switching OCS has the potential to improve infrastructure utilization and unlock new AI system architectures.

The paper concludes that the value of fast-switching OCS depends on system-level optimization beyond the optical switch itself. It also identifies the need for technology-neutral benchmarking of end-to-end reconfiguration performance, creating an opportunity for the OCP-OCS community to drive system-level measurement frameworks, co-design standards, and deployment best practices.

Table of Contents

1. Introduction	5
2. Compliance with OCP Tenets	6
2.1 Openness	6
2.2 Efficiency	6
2.3 Impact	7
2.4 Scale	7
2.5 Sustainability	8
3. What is Fast-Switching OCS?	8
3.1 The Fast Switch Use Case Paradox	10
3.2 Benefits and Challenges of Fast-Switch OCS	12
4. Enabling Technologies for Fast-Switching OCS	13
5. End-to-End Reconfiguration Latency Breakdown	20
5.1 Latency Topology Breakdown and Improvement	22
5.1.1 Scheduler and Calculation Latency ($\text{Time}_{\text{Scheduler}}$)	22
5.1.2 Control Plane Software and Protocol Latency ($\text{Time}_{\text{Control}}$)	24
5.1.3 Physical Switch Settle Time ($\text{Time}_{\text{Switch}}$)	25
5.1.4 Transceiver Locking and Clock/Data Recovery ($\text{Time}_{\text{Transceiver}}$)	26
5.1.5 Host and Network Stack Overhead ($\text{Time}_{\text{Host}}$)	27
5.2 System-Level Takeaway	28
6. Theoretical Use Cases For Fast-Switching OCS	29
6.1 Microsecond-Scale OCS: Data Center Traffic Engineering	29
6.2 AI Workloads Enabled by Fast-Switching OCS	32
6.2.1 AI-Cluster Utilization Opportunity	33
6.3 Nanosecond-Scale Optical Switching Technology	35
6.4 Future Architectural Opportunities Enabled by OCS	36
7. Conclusion	38
8. Table of Glossary	39
9. References	41
10. License	45
10.1 Creative Commons	45

Date: October 19,2026


 This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

11. About Open Compute Foundation

46

Date: October 19,2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

1. Introduction

The increasing demands of AI/ML and GPU-driven workloads are driving the need for dynamically reconfigurable, high-performance network fabrics. Optical Circuit Switching (OCS) presents a compelling foundation; however, the value of fast switching is determined not only by device switching speed alone but by rapid end-to-end reconfiguration.

This paper provides a technology-neutral overview of fast-switching OCS, its technologies, challenges, and possible use cases. Importantly, it highlights that overall system performance is constrained not only by switching elements but also by key bottlenecks such as transceiver lock times and control-plane coordination.

The paper also explores a range of emerging use cases, including dynamic GPU interconnects and workload-aware topology reconfiguration, demonstrating the transformative potential of fast switching. Finally, it outlines the broader market opportunity, in which overcoming these challenges could unlock new applications across AI and data center network infrastructure.

Evaluating trade-offs between radix, insertion loss, and switching speed is out of scope for this paper.

This white paper adheres to the Open Compute Project (OCP) core tenets of openness, efficiency, impact, scale, and sustainability, as defined by the OCP Foundation. While this document is a technical white paper rather than a hardware or software specification, it is intentionally structured to align with OCP's philosophy of open collaboration, transparent knowledge sharing, and at-scale innovation.

2. Compliance with OCP Tenets

The following subsections describe how the content and intent of this white paper support each OCP tenet.

2.1. Openness

This white paper supports the principle of openness by providing a vendor-neutral and technology-neutral analysis of fast-switching Optical Circuit Switching (OCS) technologies and architectures. Rather than advocating a specific implementation, the paper examines a broad range of switching approaches including integrated photonics, metasurfaces, semiconductor optical amplifiers, photonic MEMS, wavelength-routed architectures, and other emerging technologies. The discussion focuses on technical trade-offs, system-level challenges, and deployment considerations, drawing on information from publicly available research and industry publications.

The paper also promotes open collaboration by identifying common challenges that extend beyond individual products, including control-plane design, transceiver synchronization, end-to-end reconfiguration latency, and system-level benchmarking. By encouraging a shared understanding of these issues, the paper seeks to provide a foundation for future collaboration within the OCP-OCS community among network operators, hyperscalers, system vendors, component suppliers, and researchers.

2.2. Efficiency

Efficiency is a central theme of fast-switching OCS. The paper examines how dynamic optical connectivity can improve network resource utilization by adapting the physical network topology to changing communication workload patterns. Fast reconfiguration has the potential to reduce stranded bandwidth, improve circuit utilization, lower buffering requirements, and enable more efficient allocation of network resources compared to static topologies. The magnitude of these efficiency gains depends on the performance of the complete system, including scheduling, control-plane coordination, transceiver recovery, and host software.

Date: October 19, 2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

The paper also discusses the opportunity to improve AI infrastructure efficiency by reducing communication-induced bottlenecks. Emerging use cases such as collective-aware topology adaptation, accelerator topology reconfiguration, and Mixture-of-Experts (MoE) communication optimization demonstrate how fast-switching OCS may contribute to better utilization of expensive AI computing resources when combined with appropriate software, scheduling, and control-plane mechanisms.

2.3. Impact

The increasing scale of AI and high-performance computing infrastructures has created growing interest in network architectures capable of adapting to highly dynamic workload demands. This white paper contributes to the industry by consolidating current research, identifying common system-level bottlenecks, and establishing a framework for understanding fast-switching OCS beyond switch actuation speed alone.

An important contribution of this paper is its emphasis on end-to-end reconfiguration latency as the relevant performance metric for future deployments. The paper identifies opportunities for the OCP-OCS community to develop technology-neutral benchmarking methodologies, measurement frameworks, and system-level best practices. These activities can help accelerate industry adoption by providing a common framework for evaluating and comparing future fast-switching OCS implementations

2.4. Scale

Scalability considerations are discussed throughout the paper. The analysis spans technologies and architectures intended for deployment across AI clusters, cloud-scale environments, and large distributed computing systems. The paper reviews both current and future approaches that aim to enable dynamic optical connectivity across an increasing number of endpoints while supporting high-bandwidth workloads.

In addition to evaluating fast-switching technologies themselves, the paper highlights the importance of scheduler design, distributed control, synchronization, transceiver operation, and orchestration as essential scaling factors. The discussion considers how large groups of optical switches may ultimately operate as a single distributed fabric and identifies architectural challenges that must be addressed to achieve deployment at

hyperscale. While several studies suggest paths toward distributed OCS fabrics, significant research and operational challenges remain.

2.5. Sustainability

Fast-switching OCS has the potential to support sustainability objectives by improving utilization of existing infrastructure and reducing dependence on power-intensive electronic packet-switching resources. Maintaining traffic in the optical domain for a larger portion of the communication path can reduce repeated optical-electrical-optical processing while enabling more efficient use of deployed network capacity.

The paper also highlights the broader opportunity to improve effective utilization of AI infrastructure. By enabling network connectivity to adapt more closely to workload demands, fast-switching OCS may help reduce inefficiencies associated with communication bottlenecks and underutilized computing resources. While the benefits depend on complete system-level optimization and workload characteristics, improved utilization of networking and compute infrastructure aligns with the OCP goal of more sustainable and efficient data center operations.

3. What is Fast-Switching OCS?

The term "Fast Switching" for Optical Circuit Switches (OCS) can mean different things to different readers. This white paper aims to clarify the subtle distinctions in the definition of fast switching, which is determined solely by the time required to establish an optical connection between two optical ports. Specifically, OCS switching time, or reconfiguration time, is measured from the moment the OCS receives a command to create, delete, or reconfigure an optical connection until the connection has been successfully established.

Some may consider an OCS to be "fast switching" if the optical connection can be established within the millisecond (ms) range. In contrast, this white paper defines fast switching as sub-millisecond reconfiguration, typically in the 10–1000 microsecond (μ s) range.

Date: October 19, 2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

This distinction is significant. Compared to millisecond-class switching, sub-millisecond switching provides an order-of-magnitude faster reconfiguration speed. This speed difference ultimately enables a fundamentally different class of applications.

Millisecond-class fast switching is commonly used in three general application scenarios:

1. Patch-panel-like scenarios, such as shimming between switch layers to simplify physical connectivity. Reconfiguration occurs only when needed and is relatively infrequent.
2. Spine augmentation or replacement through slow reconfiguration. These scenarios optimize network capacity over time and, in some cases, reduce overall capacity requirements. Reconfiguration typically occurs on a weekly or monthly basis.
3. Training scenarios in which the topology remains fixed for the duration of a job. Reconfiguration may occur anywhere from minutes to weeks apart.

OCS technologies with millisecond-class switching times, such as MEMS, piezoelectric, and liquid crystal technologies, are commonly used in Scenarios 1–3.

OCS solutions based on robotic or mechanical-arm technologies can also be used in Scenarios 1–3 when switching requirements can be relaxed to seconds or minutes.

Microsecond-class fast-switching OCSs, however, open up to any AI workload where communication patterns turn over faster than a few msec. These include reconfiguration within a training iteration, collectives based reconfiguration, node to node mesh networks, fast failover, and more.

This white paper also highlights the challenges of fast-switching OCS networks. Although OCS switching time is a key metric, the overall end-to-end network system reconfiguration latency required to establish a new end-to-end optical connection includes software control-plane processing, scheduler decision latency, OCS switching/reconfiguration time, and transceiver lock/recovery time. Therefore, end-to-end network system reconfiguration performance is

Date: October 19,2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

determined by the combined contribution of all these elements, rather than the OCS switching time alone. Fast-Switching OCS use cases will be defined in more detail in a subsequent section. The implication is that currently deployed OCS technologies with switching times exceeding 1000 us (1 ms) may be unable to meet the requirements of emerging AI workloads whose communication patterns evolve on sub-millisecond time scales, creating a significant opportunity for microsecond-class OCS solutions.

3.1 The Fast Switch Use Case Paradox

Today's OCS deployments are largely shaped by the capabilities of current optical switching technology. Most commercially available systems are designed for relatively infrequent reconfiguration, and several supporting technologies—such as fast-lock transceivers, advanced control planes, and orchestration software—are still evolving.

This creates a Fast Switch Use Case Paradox:

Many of the most compelling applications for fast-switching OCS depend on capabilities that are not yet widely available. Because those capabilities do not yet exist at scale, most deployments focus on use cases that fit within today's limitations. As a result, it can be difficult to envision or evaluate the architectures that future fast-switching optical fabrics might enable.

Several factors contribute to this paradox:

1. Operational focus. Most operators are naturally focused on solving today's challenges in scaling, reliability, and cost.
2. Supporting technologies are still maturing. Fast-switching OCS depends on advances in areas such as transceivers, control planes, orchestration, and scheduling.
3. Scale remains a challenge. Many fast-switching OCS technologies are currently available at relatively small radix sizes, limiting the range of applications that can be explored today.
4. OCS is often viewed as a point solution. It is frequently evaluated as a way to solve a specific connectivity problem, rather than as part of a larger networking and computing ecosystem.

The goal of this document is to explore a future in which large-scale, fast-switching OCS and its supporting ecosystem have matured. In that future, optical connectivity becomes a dynamic resource that can be adapted to application needs, failure recovery, resource allocation, and changing traffic patterns.

While significant challenges remain, OCS technology continues to evolve. As switching speed, scale, and supporting technologies improve, the set of practical applications is expected to expand as well. This document explores both the benefits and the challenges of that evolution.

Date: October 19,2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

3.2 Benefits and Challenges of Fast-Switch OCS

Benefits	Challenges
Higher network utilization – bandwidth can be rapidly reassigned to changing traffic demands, reducing stranded capacity and improving overall fabric efficiency	Increased operational complexity – Dynamic topologies require more sophisticated control software, scheduling, routing, telemetry, monitoring, and troubleshooting.
Dynamic topology adaptation – network connectivity can be optimized in real time to match changing application and traffic demands.	Advanced transceiver requirements – Fast switching requires transceivers to rapidly acquire clock and data, restore equalization, and stabilize links after every reconfiguration.
Reduced network overbuild and oversubscription – faster reconfiguration can lower the amount of excess capacity required to accommodate traffic variability, reducing infrastructure costs	Higher switch endurance requirements – If fast switching enables more frequent topology changes, switching hardware must withstand significantly more switching cycles over its operational lifetime.
Enables AI-native fabrics – supports dynamic GPU interconnects, disaggregated inference, MoE communication, and workload-aware network optimization	
Reduced reliance on electrical packet switching – more traffic can remain in the optical domain, lowering packet-processing bottlenecks, latency, and power consumption	

Table 1 Benefits and Challenges of Fast Switching OCS

4. Enabling Technologies for Fast-Switching OCS

Below are the various enabling technologies for Fast-Switching OCS that cover switching times ranging from microseconds to nanoseconds. The intent of this paper is only to inform readers of these technologies. Robotics, MEMS (free space), Piezo, and Digital Liquid Crystal are excluded from the list due to their reconfiguration that is greater than 1 millisecond. The values given in Table 2 below reflect the approximate state-of-the-art not a first principles physical analysis. That said, these values are highly indicative of future performance. The paper does not inform readers whether such technologies are available in the market or still in the research stage.

Integrated photonics describes in-plane switching within a photonic integrated circuit, encompassing a multitude of material platforms and switching mechanisms with wildly different performance characteristics. Accordingly, we have broken integrated photonics into five categories based on mechanism. Readers may be more familiar with the term “silicon photonics”, which is not used here as it narrows to a single material platform (e.g. excludes silicon nitride, indium phosphide, thin film lithium niobate, etc). Readers looking for silicon photonics in Table 2 below should refer to “Integrated photonics — thermo-optic”, although the term is often also used for “Integrated photonics — electro-optic”. For most technologies listed below, reconfiguration time is independent of which ports are being addressed, and batch reconfiguration proceeds at the same speed as individual-port reconfiguration. The exception is Rotor (free-space), this is described in detail right after Table 2.

Date: October 19,2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

Technology	Reconfiguration Time	Limiting Factor for Reconfiguration Time	Refs
Rotor (free-space) (Figure 1)	10 μ s - 1 ms	Beam is incident on a spinning disc whose segments deflect it to different output ports. Guard band is set by spot size divided by the disc's linear velocity at the beam. Segments are physically ordered, so access is sequential on a fixed schedule. Typically, many switches run in parallel, each carrying a subset of matchings to shorten the cycle proportionally. Best case, the segment you need is next; worst case, you wait a full cycle ($\sim$ 1 ms at datacenter scale).	[10, 30]
Metasurface (free-space) (Figure 2)	$\sim$ 50 μ s	LC relaxation (on-to-off) time; scales with pixel size. Sub-wavelength pixel volumes $\sim$ 1000 $\times$ smaller than traditional digital liquid crystal $\rightarrow$ correspondingly faster switching.	[31, 32]
Integrated photonics — thermo-optic	10–100 μ s	Heat propagation from an on-chip heater into the waveguide, shifting refractive index via the thermo-optic effect. (Figure 3)	[33, 34]
Integrated photonics — MEMS	1–20 μ s	Cantilever mass, drive voltage, and mechanical settling time. (Figure 4)	[22, 35, 36]
Integrated photonics — electro-optic (Figure 5)	ns– μ s (tens of GHz)	Carrier injection/depletion, Si PN junction: limited by RC charging time of the diode.	[37]
	ns (30–56 GHz)	Electro-absorption (Franz-Keldysh), Ge/GeSi: effect itself is near-instantaneous; limited by RC charging time of the device	[38]
	sub-ns (100+ GHz – THz)	Pockels effect, thin-film lithium niobate: limited by RF electrode design, not carrier transport — fastest of the three E/O mechanisms	[39, 40]
Integrated photonics — phase change material	ns (amorphization) – ms (crystallization)	Asymmetric: amorphization is limited by thermal quench rate (fast); crystallization is limited by nucleation/growth kinetics (much slower). (Figure 6)	[41, 42]
Integrated photonics — SOA switch (Figure 7)	ns (30–56 GHz)	Carrier injection into the SOA; in practical devices, switching is primarily limited by RC charging of the SOA and its electrical interconnects, with carrier lifetime providing the intrinsic upper bound.	[43, 44]

Table 2: Fast switching OCS technologies, reconfiguration time and its limiting factor

A rotor circuit cannot be established on demand and held; every connection is torn down and remade on the fixed schedule forced by the spinning disks, so endpoints must buffer traffic until their slot comes round. Critically, this is a property of the fabric rather than of the switch alone. Endpoints must buffer traffic until their slot arrives, ToRs must run a load-balancing protocol to route around the schedule, and NICs must be designed to tolerate continuous link teardown and re-establishment [30]. A rotor is therefore not a component that can be substituted into an existing network — it is a co-designed architecture that dictates the buffering, routing, and transceiver behavior around it. Consequently, rotors are excluded from the subsequent analysis.

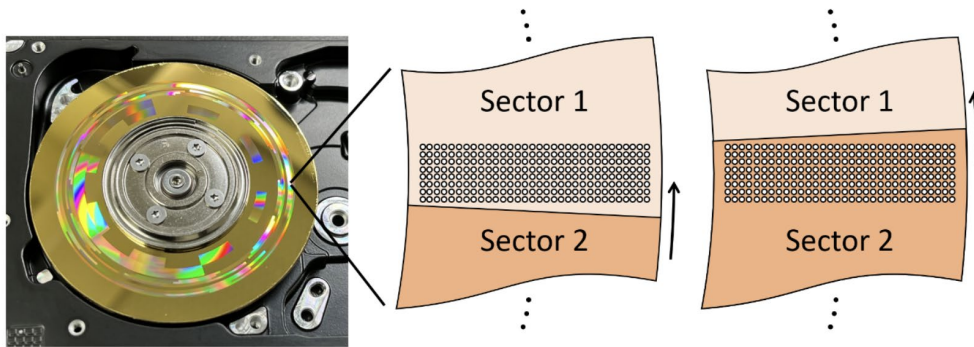


Figure 1: Rotor (free-space) — as used in the RotorNet OCS, showing diffractive sectors. A spinning disc carries segments that deflect an incident beam to different output ports, so each rotation cycles through a fixed schedule of input/output matchings [10].



Figure 2: Metasurface (free-space) -- a sub-wavelength liquid-crystal pixel array imposes a programmable phase profile on the wavefront, steering the beam to a chosen output port [27].

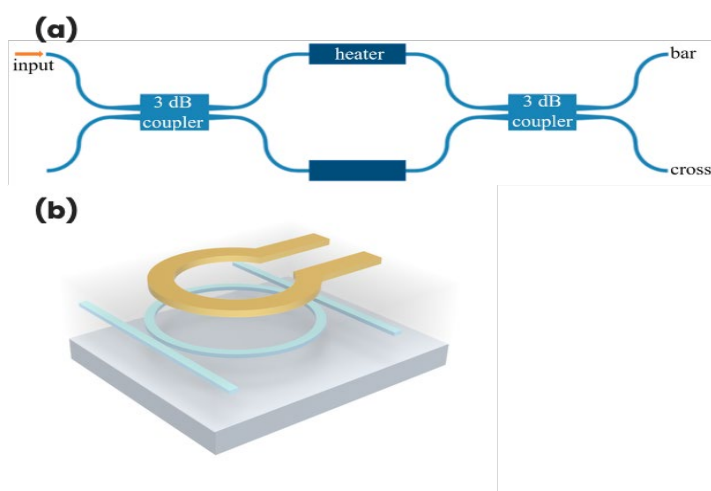


Figure 3: Integrated photonics — thermo-optic. An on-chip resistive heater warms the waveguide, shifting its refractive index via the thermo-optic effect. (a) Linear heater: the heater runs alongside a straight waveguide, changing the accumulated phase in one arm of an interferometer to steer light between output ports. (b) Ring resonator: the heater tunes the ring's resonance wavelength on or off the signal, switching light between the through and drop ports [11, 28].

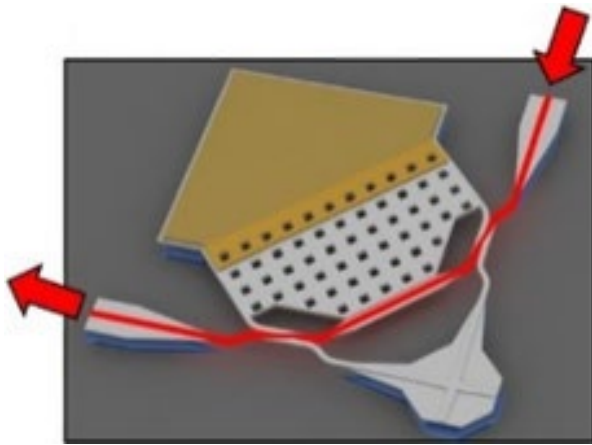


Figure 4: Integrated photonics — MEMS. An electrostatically actuated cantilever physically moves a waveguide or mirror in and out of the optical path to make or break a coupling [40].

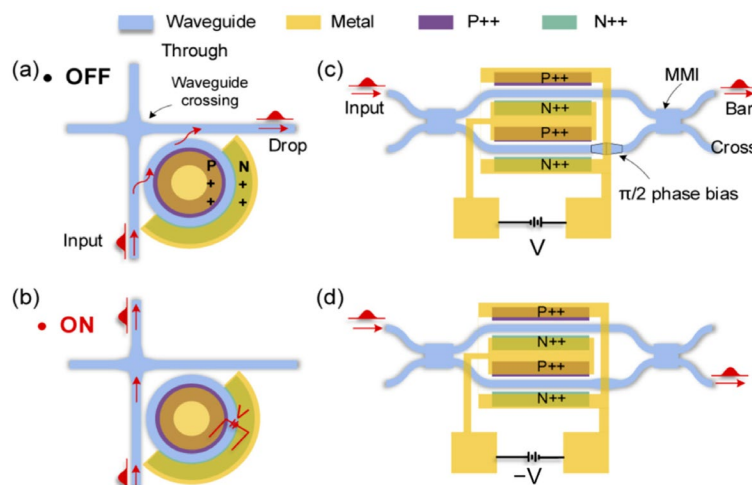


Figure 5: Integrated photonics — electro-optic. An applied voltage changes the waveguide's refractive index or absorption (carrier injection/depletion, Franz–Keldysh, or Pockels), switching the routing of light. (a, b) Ring resonator: biasing the ring shifts its resonance on or off the signal wavelength, sending light to the drop port (a) or letting it pass to the through port (b). (c, d) Linear switch: the voltage-induced phase shift in one arm sets the interferometer to the bar state (c) or the cross state (d) [11].

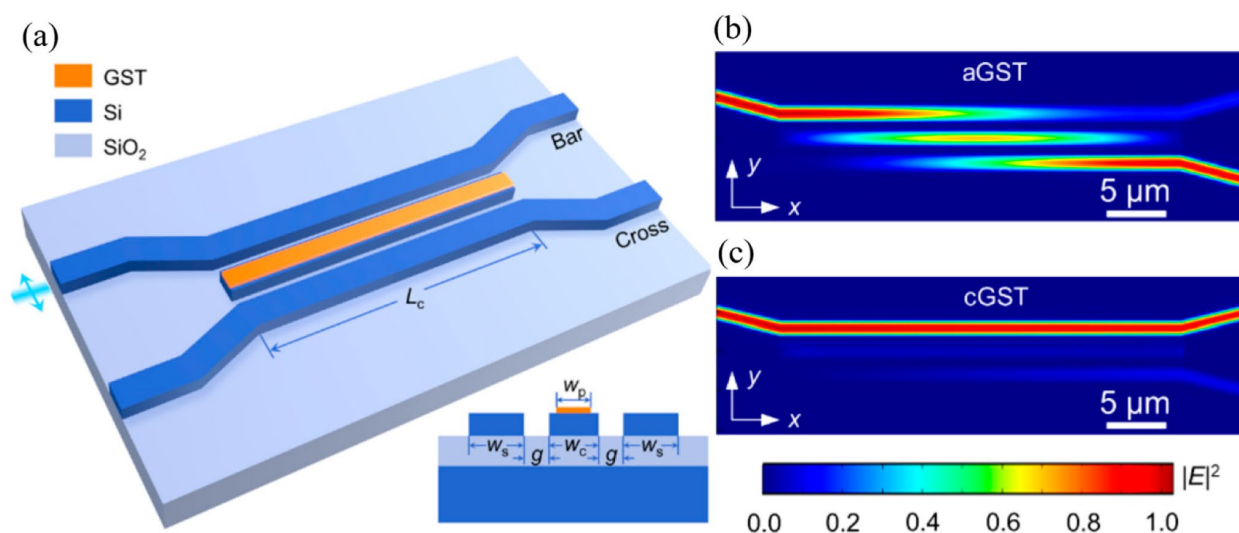
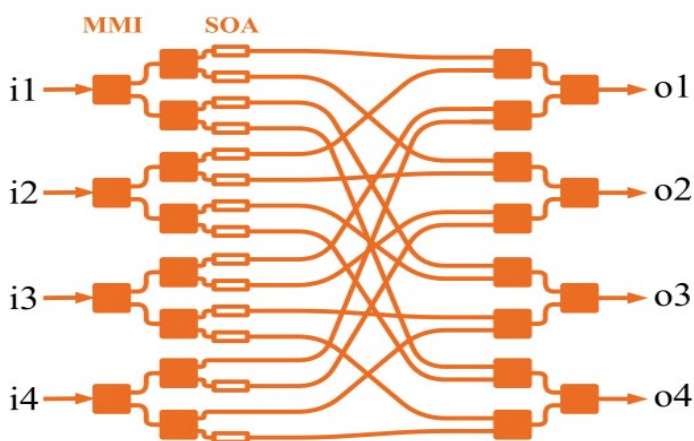


Figure 6: Integrated photonics — phase change material. A patch of PCM on the waveguide is switched between amorphous and crystalline states by heat pulses, changing the local refractive index and absorption to redirect light non-volitely. (a) Device structure, showing the PCM patch on the coupling region. (b) Cross state: light launched into one waveguide is transferred to the opposite output. (c) Bar state: the PCM state suppresses coupling, so light stays in the input waveguide and exits on the same side [11].



Date: October 19, 2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

Figure 7: Integrated photonics — SOA switch. Current injected into a semiconductor optical amplifier turns gain on or off in each arm, passing or blocking light on that path [29].

The physical limit for reconfiguration time of each technology comes from the amount of physical material that must be reconfigured. Slow- to medium-switching OCS, including Robotics, MEMS, piezo, and digital liquid crystal, all switch relatively large amounts of physical material. For example, MEMs move ~600 μm mirrors, piezo actuates collimators with ~1 mm diameter, digital liquid crystal uses large pixels ~5 μm . The fast technologies of integrated photonics and metasurfaces are entirely solid-state with no moving parts (except integrated photonics MEMs, which moves micro-cantilevers). There is scope to improve reconfiguration time of the slow to medium switching technologies; however, we expect these to remain 3-6 orders of magnitude slower than integrated photonics and metasurfaces.

Another key point is cycle endurance. A switch that reconfigures 1000 $\times$ faster may need to endure 1000 $\times$ more cycles over the same lifetime, so what matters is whether degradation scales with operating hours or with switching cycles. At kHz rates and above, cycle-limited mechanisms may be exhausted in minutes.

Integrated photonics is mostly time-driven and therefore largely indifferent to switching rate: heater electromigration and oxidation in thermo-optic phase shifters [45], DC bias drift in thin-film lithium niobate [46], and active-region defect growth in SOAs [47] all scale with operating hours. Phase change materials are the exception and the clearest cycle-limited case, with melt-quench cycling driving elemental segregation, dewetting, delamination, and contact metal diffusion [41, 48]. Many PCM devices report 10-10,000 cycles [48], corresponding to ~10 seconds of use at kHz reconfiguration rates. Although significant work is needed to improve PCM cycle endurance, the technology is likely far from any fundamental limit. Integrated MEMS, despite moving parts, has demonstrated 10^9 – 10^{10} cycles, limited by stiction, dielectric charging, and contact wear [49].

Date: October 19,2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

Metasurfaces are also time-driven, limited by ionic accumulation and photodegradation in the liquid crystal cell [50] — functions of exposure time and optical intensity rather than cycle count.

Radix and insertion loss are out of scope for this whitepaper; however, the reader should remember they are essential for commercial OCS deployment. It is harder to scale radix for some technologies in Table 12 above compared to others. Insertion loss in integrated photonics scales strongly with radix. Loss in free-space approaches is broadly independent of radix. For future fast switches, care will be needed to ensure loss and radix are acceptable in addition to improving reconfiguration speed.

5. End-to-End Reconfiguration Latency Breakdown

The required end-to-end reconfiguration latency of an OCS network is intrinsically determined by the specific application and its underlying communication patterns. For workloads characterized by sporadic data transmissions separated by prolonged idle periods (see Figure 8), the reconfiguration time window is highly permissive; commercially available OCS platforms can readily accommodate these demands, provided the topology update completes prior to the onset of data transmission. Conversely, applications with periodic, highly structured communication patterns—such as machine learning (ML) training—constrain the reconfiguration budget to the idle intervals between consecutive communication phases, which typically range from microseconds to milliseconds. Ultimately, supporting general-purpose traffic with unpredictable communication profiles imposes the most stringent constraints, where the allowable reconfiguration latency is dictated entirely by the minimum temporal gap between successive data transmissions.

Date: October 19, 2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

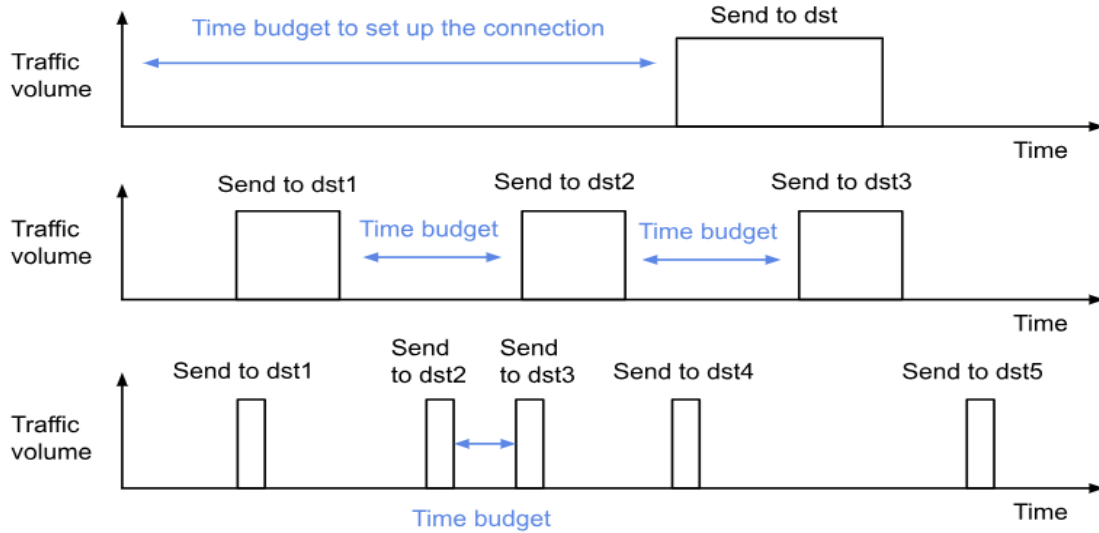


Figure 8: visualization of different communication patterns.

Specifically, this end-to-end reconfiguration latency is the time from the moment a fabric controller decides to change connectivity until the new optical paths are usable by endpoints. The OCS device switching time is only one term in that budget. In deployed OCS systems, the critical path can also include demand collection, schedule computation, config distribution, device settling, receiver clock/data recovery, time synchronization, and host or NIC readiness. The practical question is therefore not only how fast the optical element can move, but which parts of the sequence must be serialized and which can be overlapped [1, 2].

The end-to-end latency can be modeled as:

$$\text{Latency}_{\text{E2E}} \leq \text{Time}_{\text{Scheduler}} + \text{Time}_{\text{Control}} + \text{Time}_{\text{Switch}} + \text{Time}_{\text{Transceiver}} + \text{Time}_{\text{Host}}$$

The terms are not necessarily serialized in a well-designed system. Solstice-style schedule computation [3], Opera-style rolling reconfiguration [4], Sirius-style nanosecond switching [4], PULSE-style hardware scheduling [5], and workload-aware ML fabrics all show that useful

designs move work off the critical path through pipelining, precomputation, deterministic epochs, and local hardware control [6].

Chronological Reconfiguration Timeline

For a single non-overlapped reconfiguration, the sequence is usually:

- Trigger or prediction: a controller, workload runtime, or transport layer identifies that the current topology will soon be suboptimal and decides to trigger reconfiguration.
- Schedule selection: the system chooses the next matching, topology, or wavelength assignment. This can be a fresh optimization, a lookup into a precomputed schedule, or a deterministic rotor/expander epoch.
- Control dispatch: the selected configuration is delivered to the optical switch, transceiver, and endpoint agents.
- Physical switch and optical stabilization: the optical element changes state and the lightpath reaches usable power, phase, and alignment margins.
- Receiver reacquisition and synchronization: the end points recover clock, data, phase, and amplitude.
- Host/NIC readiness: endpoints drain or release buffered traffic, update queue mappings, and resume packet transmission.

5.1. Latency Topology Breakdown and Improvement

5.1.1 Scheduler and Calculation Latency ($\text{Time}_{\text{Scheduler}}$)

$\text{Time}_{\text{Scheduler}}$ covers the time the control plane needs to decide what the new topology should be. It has to collect a picture of who is sending how much traffic to whom, work out which ports should be connected to which, and coordinate the update so that no two switches disagree about the result. Two costs dominated early designs. Demand had to

Date: October 19, 2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

be pulled in over a separate electrical control network, and computing an optimal set of connections for a large cluster costs on the order of N^3 operations for N nodes [3]. Together these put the loop in the tens of milliseconds to seconds range, which is why the first hybrid deployments used optical circuits only for large, long-lived background transfers and left bursty traffic on electrical packet switches [6].

Modern architectures hide this cost rather than shrink it. The fabric runs in fixed intervals called epochs, and Solstice computes the topology for the next epoch while the current one is still carrying data, so the solver never sits on the critical path [3]. Opera and other demand-oblivious designs skip traffic collection entirely and cycle through a fixed, precomputed sequence of topologies [4]. D3 pushes the decision outward, letting top-of-rack switches adapt locally at nanosecond scale [2]. AI training clusters offer a further lever: collective communication is predictable enough that the workload can tell the scheduler what it is about to do, and the topology can be set up before the first packet is sent [7].

None of these results settle what happens once several OCS are wired together, and that gap is where much of the remaining work sits. Paths then have to be set up and torn down across and through multiple chassis, in arrangements such as an OCS Clos [2], with the topology and the size and number of switches all treated as arbitrary. A reasonable target is to establish an end-to-end path through that arbitrary topology in roughly the time a single OCS takes to configure. Put another way, a network of optical switches should behave like one distributed OCS of arbitrary scale.

Whether the computation behind that behavior is centralized or decentralized is left open here, since production fabrics have been demonstrated both ways [2, 8], as is how the computing entity talks to the switches it programs. The behavioral target is the requirement. Several path-level problems sit alongside it. Crankback is needed when a chosen path fails partway through setup and the request has to be handed back for

another attempt. Established circuits need protection so that a failure does not simply drop them. Admission has to work on strictly non-blocking fabrics, which can always accept a new request without touching existing circuits, and on reconfigurably non-blocking ones, which may have to move circuits that are already up in order to accept it. The second case is harder, because granting a request disturbs traffic that is already flowing. Computing a schedule quickly is only half of the problem, since the result still has to reach the hardware.

5.1.2 Control Plane Software and Protocol Latency ($\text{Time}_{\text{Control}}$)

A schedule that exists only inside the controller changes nothing. $\text{Time}_{\text{Control}}$ measures the gap between decision and actuation: packing the topology command into a message, moving it across the control network, and parsing it at the optical device. Conventional software-defined networking sends that command down the same separate control network the demand data arrived on, using OpenFlow or gRPC over an ordinary operating system TCP/IP stack [8]. Context switching, buffer management, and acknowledgment handshakes add microseconds to milliseconds [10]. Legacy 3D-MEMS mirrors settled slowly enough to hide all of it [6, 12]. For solid-state devices that switch in microseconds or nanoseconds, including the silicon photonic Mach-Zehnder interferometers and thin-film lithium niobate devices surveyed in Section 2, that same overhead becomes the structural bottleneck [11].

The fix is to separate the slow optimization loop from the fast actuation path. Instead of sending a software request per event, the controller installs schedules covering many epochs on the switches ahead of time, or distributes short identifiers that select among them [5, 6]. Local FPGA or ASIC agents then advance network state on synchronized hardware timers, kept in step by a clock distribution protocol [5]. The software cost is paid once per schedule instead of once per reconfiguration.

Pushing further in this direction leads to in-band control, meaning control information that travels on the data path itself rather than on a separate network. It shows up in two distinct forms. In-band actuation replaces the electrical control network with optical labels and distributed

Date: October 19, 2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

hardware timing, collapsing software overhead to nanosecond scales [5, 6]. In-band signalling answers a different question, which is how the control plane learns what to build and when. One option modifies a GPU driver to announce upcoming communication phases over a software path. The other builds a switch that reads packet headers in real time without converting the signal from optical to electrical and back, with control messages encoded into the header by GPU transceivers. The second simplifies host integration at the price of hardware dependencies, since reading a header without an optical-electrical-optical conversion tends to cost signal loss or electrical power.

Where optical and packet switching coexist, the two control planes also have to be reconciled. Reconfiguration cannot be allowed to destabilize packet routing, drive protocols into oscillation, or make routing state explode [8]. For an OCS spine layer between L3 switches, a workable pattern keeps a minimum level of connectivity between all switches at all times and treats reconfiguration as a way to add capacity above that floor [6, 9]. Adjacent switches carry interface bundles and virtual interfaces that can be resized but never taken down, which limits how far the effects of any capacity change can spread.

On the implementation side, fast-switching designs will likely need hardware-assisted control functions integrated alongside that framework to reach microsecond and nanosecond targets. With scheduling pipelined and dispatch moved into hardware, the switch itself becomes the next term.

5.1.3 Physical Switch Settle Time ($\text{Time}_{\text{Switch}}$)

$\text{Time}_{\text{Switch}}$ is the interval in which the switching medium physically changes state and stabilizes into the new topology. Production deployments such as Google's Jupiter fabric use 3D MEMS mirrors, which offer high port counts but keep reconfiguration in the millisecond to tens-of-milliseconds regime, because a physical mirror has to move and then stop moving [6, 12]. Thermo-optic integrated photonics, metasurface switches, and integrated photonic MEMS lower that barrier to microseconds [4]. Electro-optic silicon photonics and semiconductor optical

amplifier arrays, which steer light without moving anything, reach nanosecond and sub-nanosecond transitions [5, 11].

A speedup of that magnitude removes the physical medium as the dominant term, provided the switch is co-designed with the rest of the network. The open questions here are about port count, how much light is lost passing through the switch, and how many times it can be cycled before it wears out, rather than about speed itself, and they fall outside the scope of this paper. The dependency that matters for the latency budget is the receiver, which cannot use a new path until it has locked onto it.

5.1.4 Transceiver Locking and Clock/Data Recovery (TimeTransceiver)

A stable lightpath is not yet a usable link. When a switch connects a receiver to a new source, that source arrives with a different clock phase, a different signal strength, and its own channel characteristics, and the receiver has to adapt before it can read bits. Conventional continuous-mode transceivers do this with phase-locked loops and automatic gain control that converge by trial and error, taking hundreds of microseconds to milliseconds to settle the amplitude and recover the clock and data [13]. Relocking on that timescale erases whatever the preceding three terms saved [5].

Caching and burst-mode reception collapse that boundary. Burst-mode receivers, designed from the start to lock onto a signal that appears abruptly, reach lock times of a few nanoseconds at 30 Gbps [13]. Clock phase caching, used in Microsoft's Sirius architecture, stores pre-measured phase offsets for each source across a synchronized network so the receiver can look up the right setting instead of searching for it, bringing clock and data recovery below a nanosecond [5, 14]. Amplitude caching does the same for signal strength, so the receiver's amplifier gain is correct on arrival rather than converging afterward [15]. Both rest on an assumption worth stating plainly: the set of source and destination combinations has to be small enough and stable enough to measure once at commissioning, or to refresh through background maintenance. The arbitrary, multi-chassis topologies discussed in 5.1.1 weaken that assumption, because the number of combinations grows and calibration data has to be kept for paths that may be used rarely.

Date: October 19,2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

Fast relocking also raises a question the optics cannot answer alone: when is the endpoint allowed to start transmitting? Whatever asked for the connection needs an answer on a timescale comparable to the reconfiguration itself. Hosts connected directly through the fabric are the clearest case. Where control travels on a separate network, the control plane has to report whether the circuit actually came up, so the host can hold traffic or send it elsewhere while the answer is pending [9]. A switch asking for additional capacity has the same requirement, since sending on an unconfirmed circuit produces packet loss.

The partial and negative outcomes introduced in 5.1.1 need defined signalling behavior as well. A crankback event has to be reported in a form the requester can act on, and a reconfigurably non-blocking fabric has to say whether it rearranged existing circuits to admit the new one or refused the request outright. How much of that detail the requester should see, and how much traffic it has to buffer while waiting, is unresolved. Once the answer arrives and bits are flowing, the remaining delay sits above the optics.

5.1.5 Host and Network Stack Overhead ($\text{Time}_{\text{Host}}$)

Fast optical switching alone does not make applications fast. Once a path is established and the receiver is locked, data still has to move through the networking software on the servers at each end, and $\text{Time}_{\text{Host}}$ covers everything that happens there. In a conventional Ethernet and TCP/IP stack, the operating system handles interrupts, manages ring buffers, updates congestion control, and serializes packets. Together these steps add tens of microseconds [16], far more than a modern OCS needs to reconfigure. Behind a conventional stack, a sub-microsecond fabric looks to the application like an ordinary network.

The fix is to get out of the operating system. RDMA and other host-bypass mechanisms let an application hand data to the network interface card directly, which brings endpoint latency below a microsecond [16]. Reconfigurable transport protocols go further and align the endpoint with the fabric: time-division TCP, for example, lets a host transmit only during the window in which

Date: October 19, 2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

its path exists [5, 17]. Host bypass is not a substitute for fast switching. It is what makes microsecond-scale reconfiguration visible to the application.

Cutting host processing time is only half the problem, because fast reconfiguration also changes what the endpoint has to do. In a conventional network the device at the far end of a port stays the same for months, so the protocol work that runs at link-up runs once and is then done. MAC learning, in which a device discovers which addresses are reachable through which port, is the clearest example. When the far end can change every few microseconds, that work starts running continuously, and if it runs after each reconfiguration rather than alongside it, it lands on the critical path and the application pays for it. Learning therefore has to be shortened, eliminated, or overlapped with OCS configuration. Candidate mechanisms include pre-populating forwarding state, pushing MAC tables out ahead of a reconfiguration, anycast addressing, and combinations of these.

Addressing needs similar attention. Any host on an OCS fabric can be connected directly to any other, so traffic meant for a peer has to be deliverable without going through a default gateway that may not be reachable at that moment. Placing every host in a single subnet is one option; anycast addressing, in which several hosts share one address and traffic goes to whichever is currently reachable, is another. Traffic leaving the fabric still has to work the usual way, which means a server sending to the outside world must find its gateway as it would in a conventional network.

The same point applies across all five terms. A fabric whose connections change on a microsecond cadence cannot be treated as a drop-in optical component. It has to be designed as a system, from the scheduler down to the endpoint software, with every layer operating on comparable timescales.

5.2 System-Level Takeaway

Today, multiple bottlenecks for OCS reconfiguration time exist. Current OCS deployments are typically based on 3D MEMs or piezo actuation. The settling time of these technologies is

Date: October 19,2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

comparable to the transceiver lock-on, synchronization, and control plane times. Once these bottlenecks are removed, workload-driven fabrics may be scheduler-limited unless topology decisions are predicted or precomputed. The feasible improvement path is therefore staged: first hide scheduler and switch time with pipelining, next move switching technologies with faster reconfiguration times, then reduce receiver lock with calibration and burst-mode circuitry, and finally move actuation into synchronized hardware so the end-to-end critical path approaches the physical switching capability.

6. Theoretical Use Cases For Fast-Switching OCS

In hyperscale AI data centers, an Optical Circuit Switch (OCS) serves as a software-controlled, programmable Layer-1 physical networking fabric. An OCS can bypass repeated optical-electrical-optical (O-E-O) processing within the network fabric by maintaining traffic in the optical domain across the switched path, reducing latency and power consumption relative to multi-hop electrical switching. However, end-to-end reconfiguration also includes control-plane decisions, scheduling, synchronization, transceiver lock or recovery, and clock/data recovery. This distinction is important because transceiver lock time and control-plane latency have been identified as major practical bottlenecks even when the optical switch element itself is fast. A decade of published research shows OCS capability moving from millisecond-scale operation toward sub-microsecond and nanosecond switching demonstrations, with each speed tier enabling a different class of application.

6.1 Microsecond-Scale OCS: Data Center Traffic Engineering

The first practical demonstration of a multiport microsecond optical circuit switch for data center networking was presented by Farrington et al., who developed Mordia, the Microsecond Optical Research Datacenter Interconnect Architecture. Mordia demonstrated an end-to-end reconfiguration time of approximately 11.5 μ s [25] and showed that optical circuit switching could move closer to servers and racks at the Top-of-Rack level rather than being limited to

Date: October 19,2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

highly aggregated portions of the network. The work also demonstrated the feasibility of combining microsecond optical circuit switching with conventional packet switching in a hybrid architecture, while highlighting synchronization, duty-cycle, and scalability challenges associated with operating optical circuits at microsecond timescales.

Date: October 19,2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

Two key deployment implications follow from this work:

Top-of-Rack optical switching: Mordia demonstrated that microsecond optical switching is fast enough to move OCS closer to servers and racks, rather than limiting deployment to highly aggregated core portions of the network.

Hybrid packet + optical networking: The design does not replace the packet network. Instead, the OCS operates alongside packet switching, with the optical layer carrying traffic that benefits from scheduled circuits while the packet layer continues to handle traffic less suited to circuit scheduling. This hybrid model makes incremental deployment practical without requiring a wholesale network redesign.

Building on the Mordia prototype, Porter et al. addressed the control-plane challenge associated with microsecond optical switching. Their Traffic Matrix Scheduling (TMS) framework [26] demonstrated how short-lived optical circuits could be efficiently scheduled, allowing optical bandwidth to be dynamically allocated according to traffic demand while maintaining practical buffering requirements and high circuit utilization. By coordinating circuit schedules with traffic demand and leveraging microsecond switching, TMS transformed optical circuits from a statically provisioned resource into a dynamically scheduled network resource, leading to following use cases:

Large-scale cloud data centers: Traffic between racks in modern cloud data centers is bursty and skewed. A small number of rack pairs can carry a disproportionate share of traffic at a given time, and which pairs are “hot” can change over minutes.

Dynamic bandwidth allocation between racks: TMS lets the operator allocate optical capacity toward rack pairs currently exchanging the most data, then reassign capacity as traffic shifts. This improves circuit duty cycle and makes dynamic optical scheduling practical at data center scale.

Date: October 19,2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

Reduced buffering and improved circuit utilization: Microsecond switching enables shorter scheduling intervals and reduces reconfiguration overhead, improving circuit duty cycle and lowering buffering requirements associated with TMS.

While Mordia and TMS demonstrated the practical use of microsecond optical circuit switching, parallel research explored switch technologies capable of scaling optical fabrics to larger port counts.

MEMS-based optical switching has long been investigated as a path toward scalable, high-radix optical fabrics. Early work from UC Berkeley demonstrated silicon photonic MEMS switch architectures capable of integrating large numbers of optical ports while maintaining low insertion loss and power consumption. These studies established many of the architectural foundations for large-scale optical circuit switching and highlighted the potential for photonic switching fabrics to complement or replace portions of conventional electronic switching hierarchies in future data center networks [24].

While these studies established important architectural foundations for scalable optical switching, significant challenges remain in achieving high radix, low insertion loss, fast switching speed, manufacturability, and deployment-scale economics simultaneously.

6.2 AI Workloads Enabled by Fast-Switching OCS

As optical switching speeds progress from microseconds toward nanoseconds, OCS evolves from a traffic-engineering mechanism into a dynamic infrastructure resource for AI systems. Across both research platforms and early production deployments, four recurring workload-driven themes are emerging: resiliency through dynamic reconfiguration, topology-aware collective communication, accelerator topology flexibility, and specialized optimization for Mixture-of-Experts training.

- **Automated failure recovery:** In large-scale ML supercomputers, OCS-backed SDN can dynamically configure routes around machine, chip, and link failures. Google's TPUv4

Date: October 19,2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

resiliency work reports that this approach helped TPUv4 supercomputers achieve 99.98% system availability while handling hardware outages experienced by about 1% of training jobs [18].

- **Adaptive topology for collective communications:** Collective Communication for Distributed ML proposes reconfiguring the network topology to match collective communication patterns, creating direct, contention-free circuits between communicating GPUs. The paper reports up to 3× speedup over state-of-the-art collective algorithms on 128 GPUs and a 1.3× speedup [19] in end-to-end training throughput.
- **Accelerator topology reconfiguration and resource flexibility:** OCS has already been used in TPUv4-style ML supercomputers to dynamically reconfigure interconnect topology, improve availability, utilization, modularity, deployment, security, power, and performance. Mission Apollo supports the broader claim that OCS has been deployed at datacenter scale, while TPU v4 and Resiliency at Scale are the stronger references for ML-supercomputer topology reconfiguration and failure resilience [18, 20, 21].
- **Optical convergence for MoE training:** MixNet, A Runtime Reconfigurable Optical-Electrical Fabric for Distributed Mixture-of-Experts Training augments existing electrical GPU interconnects with a regionally reconfigurable OCS domain for MoE all-to-all traffic. The paper reports a 32 A100 GPU prototype and large-scale simulations showing cost-efficiency improvements versus fat-tree and rail-optimized fabrics [7].

6.2.1 AI-Cluster Utilization Opportunity

Several of these use cases, particularly collective-aware topology adaptation and MoE all-to-all optimization, highlight a broader opportunity to improve the productive use of AI-cluster resources by reducing network-induced communication stalls. Other use cases, such as failure recovery and accelerator topology reconfiguration, primarily contribute through improved availability, resilience, and resource flexibility.

Model FLOPs Utilization (MFU) measures the percentage of a GPU cluster's theoretical peak computing capacity that is used for productive model computation during AI training.

For example, an MFU of 40% means that the model performs productive training calculations at approximately 40% of the cluster's theoretical peak compute rate.

The remaining capacity does not necessarily represent completely idle GPUs. It represents time and resources that do not contribute directly to the model calculations counted by MFU. Contributing factors can include communication, synchronization, memory access, load imbalance, scheduling, and software overhead.

A presentation at the 2026 OCP APAC Summit cited MFU levels of approximately 38–43% for AI training clusters. Applying the corresponding utilization gap to a hypothetical \$500M compute investment yields an illustrative equivalent of approximately \$285–\$310M in installed compute capacity not contributing directly to model computation at a given time. This calculation is intended only as an economic framing example and should not be interpreted as a directly recoverable loss or as being caused solely by networking [23].

Nevertheless, even recovering a fraction of the network-related portion could represent significant infrastructure value. Fast-switching OCS could help reduce network-induced communication stalls and dynamically align optical connectivity with changing workload communication patterns. However, the achievable MFU improvement depends on workload characteristics and the complete system design, including scheduling, control-plane latency, switch reconfiguration, transceiver recovery, synchronization, collective communication software, and host or NIC readiness.

Taken together, these use cases indicate that dynamic optical connectivity could support failure recovery, collective-aware communication, accelerator topology adaptation, and MoE all-to-all optimization. The connection to MFU is most direct for collective-aware topology adaptation and MoE communication, where network-induced communication stalls can affect productive training throughput. Fast-switching OCS could contribute to higher AI-fabric efficiency by allowing the physical network topology to respond more closely to changing workload demands. Translating this capability into higher MFU requires coordinated optimization of the complete communication path and validation through workload-level measurements.

Date: October 19, 2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

6.3 Nanosecond-Scale Optical Switching Technology

Sirius (Ballani et al., Microsoft Research, SIGCOMM 2020) explored a nanosecond-scale optical switching architecture based on custom tunable transmitters and passive wavelength-routing components. Rather than using conventional standalone OCS, Sirius combines rapidly tunable lasers with passive arrayed-waveguide gratings (AWGs) to dynamically steer traffic through wavelength selection. Although not an OCS, as Sirius is a co-optimization of an entire datacenter not a drop-in switch replacement, Sirius is relevant here as a north star for ns switching. The work demonstrated laser tuning below 912 ps and 3.84 ns end-to-end reconfiguration on 50 Gbps channels [4]. Through large-scale simulations, Sirius approximated the performance of an ideal electrically switched non-blocking network while showing up to 74-77% lower power.

- **AI clusters:** The flat, high-radix architecture is relevant to AI training fabrics because Sirius explicitly identifies distributed DNN training and resource disaggregation as emerging hardware-driven workloads requiring higher bandwidth and ultra-low latency.
- **High-bandwidth, low-latency cloud fabrics:** Sirius was an important research prototype showing that nanosecond-scale end-to-end optical reconfiguration and sub-nanosecond laser tuning are feasible in a tightly co-designed optical switching architecture [4].

While Sirius demonstrates that nanosecond-scale optical reconfiguration is possible, it relies on custom wavelength-agile transceivers and a tightly co-designed network architecture rather than a conventional drop-in OCS platform. As a result, questions remain regarding deployment complexity, interoperability with existing transceiver ecosystems, operational manageability, and scaling such architectures in production environments. Sirius should therefore be viewed as an important proof of concept demonstrating the feasibility of nanosecond optical reconfiguration [4] rather than a complete solution to fast-switching optical networking.

A complementary system-level result comes from Eindhoven University (Nature Communications, 2022), which rather than optimizing the switch element alone integrated

Date: October 19,2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

switching and control into a single co-designed system, achieving 43.4 ns combined switch-and-control latency alongside 3.1 ns clock and data recovery [1]:

- **Optical packet-switched data centers:** This result matters specifically because it addresses the gap this white paper has emphasized throughout: a fast switch is not useful if the surrounding control plane and transceiver synchronization can't keep pace. Eindhoven's result shows the full system, not just the switch, operating at the nanosecond scale.
- **Latency-sensitive cloud applications:** Applications where tail latency directly affects user experience or SLA compliance benefit disproportionately from system-level (not just device-level) nanosecond switching.
- **Optical fabrics require contention resolution:** When multiple packets compete for the same optical path simultaneously, the control system must resolve that contention and reconfigure the fabric before the next payload arrives. This is the specific problem the co-designed control system addresses.

Although this work significantly reduces the gap between device-level switching speed and end-to-end system reconfiguration, it was demonstrated in a specialized research environment. Additional work remains to scale control systems, integrate production networking stacks, and validate operation in large-scale deployments.

6.4 Future Architectural Opportunities Enabled by OCS

While the previous sections focused on demonstrated technologies and emerging workload benefits, researchers are also exploring new network architectures built around fast optical reconfiguration. These proposals move beyond using OCS as an enhancement to existing networks and instead treat optical switching as a foundational building block for future AI and HPC fabrics.

PULSE (Benjamin et al., Journal of Lightwave Technology): Optical Circuit-Switched Data Center Architecture Operating at Nanosecond Timescales, proposes a sub-microsecond optical

Date: October 19, 2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

circuit-switched data center network architecture that uses distributed hardware schedulers. The design targets optical reconfiguration times on the order of hundreds of picoseconds, timeslots on the order of tens of nanoseconds, and a 120 ns scheduling epoch. The proposed architecture scales to 4,096 nodes and 25.6 Pbps of aggregate capacity. The picosecond-scale figure refers to optical reconfiguration, not application-level latency. The evaluated architecture reports average latency of 0.7–1.2 μ s, a best-case median of 0.4 μ s, and tail latency of up to 5 μ s under the evaluated conditions [5].

All the proposals point towards building a complete optical network architecture built around nanosecond-scale switching.

- **Very large AI clusters:** The proposed architecture targets up to 4,096 nodes, large enough to be architecturally comparable to current frontier training clusters, rather than a small research testbed.
- **HPC and distributed training:** At 25.6 Pbps proposed aggregate capacity, PULSE is explicitly sized against the bandwidth demands of large-scale distributed training and high-performance computing workloads, not general enterprise cloud traffic.
- **Optical scheduling fabrics:** The timeslot/epoch structure (10–20 ns slots within a ~120 ns epoch) implies a scheduling discipline closer to a TDM (time-division multiplexed) fabric than to a circuit that's held open for the duration of a flow, a meaningfully different operational model from the earlier microsecond-tier work.

Although PULSE demonstrates the architectural potential of nanosecond-scale optical scheduling fabrics, it remains a proposed and simulated architecture rather than a production deployment. Practical implementation challenges remain, including scheduler scalability, synchronization overhead, transceiver constraints, fault tolerance, and integration with existing software and networking stacks.

Date: October 19, 2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

Taken together, these architectural directions suggest that future AI and HPC networks may increasingly treat optical switching as a first-class fabric resource rather than an overlay on conventional packet networks. Architectures such as PULSE illustrate how fast optical reconfiguration could become a foundational element of future large-scale distributed computing systems, although significant engineering and deployment challenges remain.

7. Conclusion

Fast-switching OCS reframes the network fabric as a dynamic, workload-responsive resource, but its practical impact is determined at the system level, not only at the device level. Published work spans millisecond-class production OCS deployments, microsecond-scale research prototypes, sub-microsecond switch platforms, and nanosecond- to sub-nanosecond wavelength switching demonstrations. Each tier maps to different use cases: millisecond-class systems are suitable for job-level topology setup, failure recovery, and scheduled reconfiguration; microsecond-class systems begin to support finer-grained traffic engineering and collective-aware topology changes; and nanosecond-class systems point toward the possibility of packet- or burst-scale optical path assignment, provided that transceiver recovery, synchronization, and control-plane coordination can keep pace.

The central takeaway is that end-to-end reconfiguration latency is the metric that matters. This includes SDN or orchestration decisions, scheduling, switch actuation or settling, transceiver lock/recovery, CDR, synchronization, and host/software readiness. Across the examples discussed in this paper, system-level bottlenecks repeatedly emerge outside the optical switching element itself. The OCP fast-switching discussion explicitly identified transceiver lock time, control-plane latency, synchronization, software propagation overhead, and lack of standardized end-to-end measurement as key challenges. The OCP OCS team is therefore well-positioned to drive technology-neutral benchmarking and system-level co-design standards that translate fast-switching OCS from a component capability into a deployable infrastructure advantage.

Date: October 19, 2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

8. Table of Glossary

Term	Meaning
Capex	Capital expenditure
DC	Data center
DP	Data Parallelism
DWDM	Dense Wavelength Division Multiplexing
EPS	Electrical packet switching
EP	Expert Parallelism
ICI	Google's Inter-Chip Interconnect
LCoS	Liquid Crystal on Silicon
MEMS	Micro-Electro-Mechanical Systems
MoE	Mixture of Experts
MZI	Mach-Zehnder Interferometer
OCP	Open Compute Project
OCS	Optical circuit switching
O-E-O conversion	Optical-electrical-optical conversion
Opex	Operational expenditure
OPM	Optical Power Monitoring

Date: October 19,2026


 This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

PP	Pipeline Parallelism
SOA	Semiconductor Optical Amplifier
SiP	Silicon Photonics
Switch radix	Port count on a switch
TPU	Google's Tensor Processing Unit
VOA	Variable Optical Attenuator
WSS	Wavelength Selective Switches

Date: October 19,2026

This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

9. References

- [1] Xue et al., [Nanosecond optical switching and control system for data center networks](#), Nature Communications, 2022.
- [2] Zerwas et al., [D3: An Adaptive Reconfigurable Datacenter Network](#), arXiv, 2024.
- [3] Mellette et al., [Expanding across time to deliver bandwidth efficiency and low latency](#), NSDI, 2020.
- [4] Ballani et al., [Sirius: A Flat Datacenter Network with Nanosecond Optical Switching](#), SIGCOMM, 2020.
- [5] Benjamin et al., [PULSE: Optical Circuit Switched Data Center Architecture Operating at Nanosecond Timescales](#), Journal of Lightwave Technology, 2020.
- [6] Farrington et al., [Helios: a hybrid electrical/optical switch architecture for modular data centers](#), SIGCOMM, 2010.
- [7] Liao et al., [MixNet: A Runtime Reconfigurable Optical-Electrical Fabric for Distributed Mixture-of-Experts Training](#), SIGCOMM, 2025.
- [8] Ferguson et al., [Orion: Google's Software-Defined networking control plane](#), NSDI, 2021.
- [9] Raja et al., [Ultrafast optical circuit switching for data centers using integrated soliton microcombs](#), Nature Communications, 2021.
- [10] Mellette et al., [Realizing RotorNet: Toward Practical Microsecond Scale Optical Networking](#), SIGCOMM, 2024.
- [11] Wencheng Yue et al., [Review of 2 × 2 Silicon Photonic Switches](#), Photonics 2023, 10, 564.
- [12] Yamamoto et al., [A three-dimensional MEMS optical switching module having 100 input and 100 output ports](#), IEEE Photonics Technology Letters, 2006.
- [13] Wang et al., [A Co-Designed DC-Coupled 30-Gbps Burst-Mode Receiver and CDR with 3.2-ns Locking Time for Fast Optical Switching](#), ECOC, 2024.
- [14] Clark et al., [Synchronous subnanosecond clock and data recovery for optically switched data centres using clock phase caching](#), Nature Electronics, 2021.
- [15] Gerard et al., [Fast and Uniform Optically-Switched Data Centre Networks Enabled by Amplitude Caching](#), OFC, 2021.

Date: October 19,2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#).

- [16] Su et al., [EDM: An Ultra-Low Latency Ethernet Fabric for Memory Disaggregation](#), ASPLOS, 2024.
- [17] Chen et al., [Time-division TCP for reconfigurable data center networks](#), SIGCOMM, 2022.
- [18] Zu et al., [Resiliency at Scale: Managing Google's TPUv4 Machine Learning Supercomputer](#), USENIX NSDI, 2024.
- [19] Kumar et al., [PCCL: Photonic Circuit-Switched Collective Communication for Distributed ML](#), arXiv, 2025.
- [20] Urata et al., [Mission Apollo: Landing Optical Circuit Switching at Datacenter Scale](#), arXiv, 2022.
- [21] Jouppi et al., [TPU v4: An Optically Reconfigurable Supercomputer for Machine Learning with Hardware Support for Embeddings](#), ISCA, 2023.
- [22] Miller et al., [Integrated Photonics for Low-Power Optical Interconnects](#), Berkeley EECS, 2015.
- [23] Fiorini, C., [Nvidia ships co-packaged optics switch as ecosystem testing gap widens](#), TechTimes, 2026.
- [24] Han, S., [Large Scale Silicon Photonic MEMS Switch](#), UC Berkeley Tech Report, 2015.
- [25] Farrington et al., [A Multiport Microsecond Optical Circuit Switch for Data Center Networking](#), IEEE Photonics Technology Letters, 2013.
- [26] Porter et al., [Integrating Microsecond Circuit Switching into the Data Center](#), ACM SIGCOMM, 2013.
- [27] Sabbir Rangwala, [Lumotive Raises \\$45M for Chip-Scale Light Management Chip](#), Forbes, 2025.
- [28] Flexcompute, [Thermally Tuned Ring Resonator](#), 2025.
- [29] Rombouts et al., [Design and Demonstration of an O-Band InP Monolithically Integrated SOA-Based Broadcast and Select Optical Space Switch](#), Journal of Lightwave Technology (JLT), Vol 43, Issue 7. 2025.
- [30] Mellette et al., [RotorNet: A Scalable, Low-complexity, Optical Datacenter Network](#), Proc. ACM SIGCOMM, 2017.

Date: October 19,2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#).

- [31] Murray Slovick, [Solid-State LiDAR Leverages Liquid-Crystal Metasurfaces, CMOS Manufacturing](#), Electronic Design, 2020.
- [32] Jake Hertz, [Lumotive Releases First Commercial Optical Beam-Steering Chip for LiDAR](#), All About Circuits, 2023.
- [33] Pérez-López et al., [General-purpose programmable photonic processor for advanced radiofrequency applications](#), Nat. Commun., 2024.
- [34] Matsuura et al., [Accelerating Switching Speed of Thermo-optic MZI Silicon-Photonic Switches with 'Turbo Pulse' in PWM Control](#), OFC, 2017.
- [35] Han, [Large Scale Silicon Photonic MEMS Switch](#), UC Berkeley Tech Report, 2015.
- [36] Seok et al., [Highly Scalable Digital Silicon Photonic MEMS Switches](#), J. Lightwave Technol., 2016.
- [37] Thomson et al., [50-Gb/s silicon optical modulator](#), IEEE Photon. Technol. Lett., 2012.
- [38] Mastronardi et al., [High-speed Si/GeSi hetero-structure electro absorption modulator](#), Opt. Express, 2018.
- [39] Wang et al., [Integrated lithium niobate electro-optic modulators operating at CMOS-compatible voltages](#), Nature, 2018.
- [40] Mercante et al., [Thin film lithium niobate electro-optic modulator with terahertz operating bandwidth](#), Opt. Express, 2018.
- [41] Constantin-Popescu et al., [New phase change materials for active photonics](#), Proc. SPIE, 2022.
- [42] Zhang et al., [Broadband transparent optical phase change materials for high-performance nonvolatile photonics](#), Nat. Commun., 2019.
- [43] Rasoulzadeh Zali et al., [Monolithically Integrated Ultra-dense Polarization Insensitive 8×8 InP Switch for Optical Networks](#), OFC 2025.
- [44] Rombouts, M. et al., [Design and Demonstration of an O-Band InP Monolithically Integrated SOA-Based Broadcast and Select Optical Space Switch](#), J. Lightwave Technology, 2025.
- [45] Nikita Yu. Dmitriev et al., [High-performance robust ScNx/Nb/ScNx heaters for an integrated silicon nitride low-loss photonic platform](#), Phys. Rev. Applied, 2025.

Date: October 19, 2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#).

- [46] Hirotoshi Nagata et al., [Improved Long-Term DC Drift in OH-Reduced Lithium Niobate Optical Intensity Modulators](#), Applied Optics, 1996.
- [47] S. Huyghe et al., [Electroluminescence spectroscopy for reliability investigations of 1.55 \$\mu\text{m}\$ bulk semiconductor optical amplifier](#), Microelectronics Reliability, 2005.
- [48] Popescu et al., [Unravelling and circumventing failure mechanisms in chalcogenide optical phase change materials](#), arXiv, 2024.
- [49] Seok et al., [Reliability study of digital silicon photonic MEMS switches](#), IEEE 12th Int. Conf. Group IV Photonics, 2015.
- [50] Wouter Oepts et al., [Degradation of Liquid Crystal Alignment upon High Intensity Illumination in Microdisplays](#), Proceedings of 22th International Display Research Conference, 2002.

10. License

10.1. Creative Commons

OCP encourages participants to share their proposals, specifications and designs with the community. This is to promote openness and encourage continuous and open feedback. It is important to remember that by providing feedback for any such documents, whether in written or verbal form, that the contributor or the contributor's organization grants OCP and its members irrevocable right to use this feedback for any purpose without any further obligation.

It is acknowledged that any such documentation and any ancillary materials that are provided to OCP in connection with this document, including without limitation any white papers, articles, photographs, studies, diagrams, contact information (together, "Materials") are made available under the Creative Commons Attribution-ShareAlike 4.0 International License found here: <https://creativecommons.org/licenses/by-sa/4.0/>, or any later version, and without limiting the foregoing, OCP may make the Materials available under such terms.

As a contributor to this document, all members represent that they have the authority to grant the rights and licenses herein. They further represent and warrant that the Materials do not and will not violate the copyrights or misappropriate the trade secret rights of any third party, including without limitation rights in intellectual property. The contributor(s) also represent that, to the extent the Materials include materials protected by copyright or trade secret rights that are owned or created by any third-party, they have obtained permission for its use consistent with the foregoing. They will provide OCP evidence of such permission upon OCP's request. This document and any "Materials" are published on the respective project's wiki page and are open to the public in accordance with OCP's Bylaws and IP Policy. This can be found at

Date: October 19,2026



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

<http://www.opencompute.org/participate/legal-documents/>. If you have any questions, please contact OCP.

11. About Open Compute Foundation

The Open Compute Project (OCP) brings at-scale innovations and hyperscaler best practices to all, spanning technology domains from the data center to the edge, and the technology stack from silicon, to systems, to site facilities and services. The international OCP Community is made up of organizations and people from hyperscale and tier-2 cloud data center operators, communications providers, colocation providers, diverse enterprises, and technology vendors. The OCP Foundation fosters collaboration within the Community to tackle market challenges in an array of OCP Projects that create open contributions such as specifications, designs, and more. The OCP Solution Provider Program then recognizes the application of those contributions in compliant products, solutions, and data center facilities and services with designations like OCP Inspired, OCP Accepted and OCP Ready. With the tenets of openness, impact, efficiency, scale and sustainability, the OCP engages and educates thousands of engineers every year through many events and webinars. Across many initiatives the OCP Foundation and Community are meeting the market today and shaping the future.

Learn more at: www.opencompute.org.