Traffic-Adaptive Power Reconfiguration for Energy-Efficient and Energy-Proportional Optical Interconnects

Yuyang Wang, and Kwang-Ting Cheng

2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)

In this study, we proposed POLESTAR, i.e., POwer LEvel Scaling with Traffic-Adaptive Reconfiguration, for microring-based optical interconnects. Featuring a collection of runtime reconfiguration strategies that target the power states of the lasers and the microring tuning circuitry, POLESTAR demonstrates remarkable effectiveness for improving the energy efficiency and energy proportionality of underutilized datacenter/HPC interconnects. Through traffic-adaptive adjustment of the reconfiguration mechanism, POLESTAR achieves a reasonable balance between energy saving and application execution time. Good scalability across topologies, network loads, and potential advances in optical device design is also observed. POLESTAR is extensible by incorporating more reconfiguration strategies and improving existing ones. With future work targeting better traffic prediction techniques and the possible inclusion of runtime traffic scheduling, POLESTAR paves a promising way to the energy-efficient and energy-proportional optical interconnects for future datacenter/HPC applications.

This work makes a significant contribution by proposing the concept of effective energy efficiency. This concept is particularly noteworthy for its focus on the runtime management of power-consuming components in optical interconnects, which, if not cleverly executed, could inadvertently offset the benefit from device- and link-level optimization efforts. This leads to the realization of the utmost importance of runtime reconfiguration capabilities for optical interconnects in data center/high-performance computing settings to achieve high energy efficiency and proportionality.

Traffic-Adaptive Power Reconfiguration for Energy-Efficient and Energy-Proportional Optical Interconnects

Yuyang Wang
Department of Electrical and Computer Engineering
University of California, Santa Barbara
Santa Barbara, California, USA
yuyang_wang@ucsb.edu

Kwang-Ting Cheng
School of Engineering
Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong SAR, China
timcheng@ust.hk

2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD) | 978-1-6654-4507-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCAD51958.2021.9643475

***Abstract*—Silicon microring–based optical interconnects offer great potential for high-bandwidth data communication in future datacenters and high-performance computing systems. However, a lack of effective runtime power management strategies for optical links, especially during idle or low-utilization periods, is devastating to the energy efficiency and the energy proportionality of the network. In this study, we propose POLESTAR, i.e., POwer LEvel Scaling with Traffic-Adaptive Reconfiguration, for microring-based optical interconnects. POLESTAR offers a collection of runtime reconfiguration strategies that target the power states of the lasers and the microring tuning circuitry. The reconfiguration mechanism of the power states is traffic-adaptive for exploiting the trade-off between energy saving and application execution time. The evaluation of POLESTAR with production datacenter traces demonstrates up to 87 % reduction in pJ/b consumption and significant improvements in energy proportionality metrics, notably outperforming existing strategies.**

I. INTRODUCTION

The recent explosive growth of data-driven artificial intelligence (AI) applications has triggered the convergence between datacenters and high-performance computing (HPC) systems in terms of performance requirements [1]. As the computational capability continuously improves through hardware parallelism and specialization, the bottleneck of system performance is gradually shifting from computation to communication [2]. According to the latest technology projections, the bandwidth capacity provisioned for intra-datacenter/HPC interconnects has exceeded hundreds of Gb/s, a data rate at which traditional electrical interconnects become uneconomical [3]. As a result, optical interconnects are expected to replace electrical ones in both datacenters and HPC systems with a growing trend toward shorter reach [4].

Silicon photonics is considered a scalable and cost-effective technology for implementing optical interconnects with a CMOS-compatible fabrication process [5]. In particular, optical links based on quantum dot (QD) comb lasers and silicon microring resonators (MRRs) have drawn increasing attention for achieving dense wavelength-division multiplexing (DWDM) within compact footprints [6]. Innovations at device, link, and system levels have been reported to advocate microring-based interconnect solutions for future datacenters and HPC systems [7].

Unlike the bandwidth capacity, for which technologies beyond 1 Tb/s are already under active investigation [8], the energy issues of microring-based optical interconnects have long remained challenging [9]. Despite recent advances in device design [10]–[12] and link-level power mitigation techniques [13]–[19] that are pushing the best-case energy efficiency of an individual link toward ~1 pJ/b, the effective energy efficiency of an interconnected network is often far from this optimum due to traffic dynamics [3]. Moreover, failure to properly manage the link power during idle or low-utilization periods is devastating to the energy proportionality of the network [20]. Given these issues, the following major contributors to the energy consumption of microring-based optical interconnects must be addressed by system-level reconfiguration strategies at application runtime:

Static power consumed by the continuous-wave (CW) lasers and the microring tuning circuitry can take up over 80 % of the link power consumption, according to a recent analysis of microring-based DWDM links [21]. The static power is inevitable as long as the link remains on, even if it is not transmitting data. Due to the burstiness of traffic in datacenters and HPC systems, the interconnects can often stay idle for relatively long periods [22]. As a result, both the lasers and the microring tuning circuitry need runtime power reconfiguration to avoid excessive waste of energy during idle periods.

Bandwidth overprovisioning is a common practice in datacenters and HPC systems [23]. This naïve strategy aims to avoid data-starved computation nodes by deploying optical links that can accommodate the peak bandwidth requirement at the cost of higher communication power [24]. However, as the laser power drastically increases at higher data rates, the optimal energy-per-bit consumption of an optical link often occurs at a data rate slower than the peak [25]. Meanwhile, the skewed (spatially non-uniform) traffic patterns in datacenters and HPC systems [26] often result in underutilized links in certain parts of the network. Therefore, it is unwise to always use the maximum data rate for all network activities.

Inspired by the dynamic voltage and frequency scaling (DVFS) techniques that are commonly adopted by the computational hardware (e.g., CPUs and GPUs) [27], runtime reconfiguration of power states can also be applied to the lasers and the microring tuning circuitry to save energy [28]. However, special considerations must be taken regarding the reconfiguration delay of the optical components, such as the turn-on delay of the lasers [29] and the stabilization time of microring tuning [30]. The reconfiguration delay harms the application execution time and incurs extra energy consumption that could offset the energy saving. Therefore, the reconfiguration strategies must be adaptive to the runtime traffic patterns to avoid unnecessary changes of the power states to the maximum extent.

Given such design considerations, we propose POLESTAR, i.e., POwer LEvel Scaling with Traffic-Adaptive Reconfiguration, for microring-based optical interconnects in datacenters and HPC systems. To be elaborated in Section III, POLESTAR offers a collection of runtime power reconfiguration strategies designed around the following objectives:

1) reducing the energy consumption of idle links by switching the lasers and the microring tuning circuitry to *off* or some *low-power* states;

978-1-6654-4507-8/21/$31.00 ©2021 IEEE

2) improving the energy efficiency of active links by using an *intermediate* (as opposed to the maximum) data rate for select network activities; and
3) minimizing the overhead to the application execution time by making the reconfiguration mechanism of the power states *traffic-adaptive*.

We evaluate the effectiveness of POLESTAR for representative network topologies with an event-driven simulator modified from SimGrid [31] and WREHCN [32]. The network in simulation is driven by production data center traces from Alibaba, Inc., containing task execution details of ~4000 machines over eight days [33]. The simulation results demonstrate a 72–87 % reduction in pJ/b consumption and significant improvements in energy proportionality metrics, notably outperforming existing strategies.

II. Background and Related Work

A. Overview of Microring-Based Optical Interconnects

The communication links in datacenters and HPC systems connect the computation nodes to their entry-point routers and the routers among themselves. As of today, optical links have dominated the interconnects above the rack-to-rack level and started to penetrate the intra-rack regime [3]. silicon microring–based optical interconnects are made up of active links enabled by optical transceivers. As illustrated in Fig. 1, a microring-based optical transceiver achieves DWDM communication by deploying cascaded microrings along a shared waveguide. At the transmitter (Tx) side, each microring modulator modulates a specific wavelength at its resonance. At the receiver (Rx) side, a corresponding microring filter couples the signal out for detection.

B. Energy Issues of Microring-Based Optical Interconnects

1) Energy efficiency: The energy efficiency of optical interconnects is usually measured in pJ/b. However, in most literature, this metric is computed as mW/(Gb/s), reflecting the power required to attain a target data rate, which has a unit equivalent to pJ/b as Watt = J/s. In this study, we refer to the above power-oriented metric as the *nominal energy efficiency* to distinguish it from the *effective energy efficiency*, the latter measuring the actual energy consumption associated with data movement.

The *nominal energy efficiency* of a microring-based optical link heavily relies on the power consumption of three components, namely the laser, the microring tuning circuitry, and the electrical driver circuitry:

$$E_{\text{nom}} = \frac{P_{\text{laser}} + P_{\text{tuning}} + P_{\text{driver}}}{m \cdot \text{DR}}. \tag{1}$$

Here, m is the number of DWDM channels, and DR is the target data rate per channel. Due to the process variations that deviate the resonance wavelengths of the microrings from their design values [34], P_{tuning} is required to thermally tune the microrings and align the Tx/Rx channels to a mutual set of carrier wavelengths. As analyses show that the tuning power can take over half of the link power consumption [21], many link-level optimization techniques were proposed to improve the nominal energy efficiency of optical links by reducing the tuning power [13]–[19].

The *effective energy efficiency*, on the other hand, measures the actual energy consumed by the optical interconnects to transfer a total number of bits during the entire timespan of application execution:

$$E_{\text{eff}} = \frac{\text{Energy consumption}}{\text{\# of bits transferred}}. \tag{2}$$

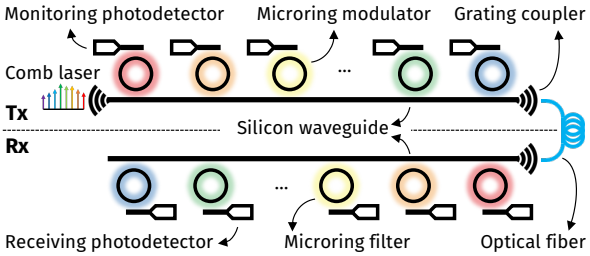


Fig. 1. Illustration of a silicon microring–based optical transceiver.

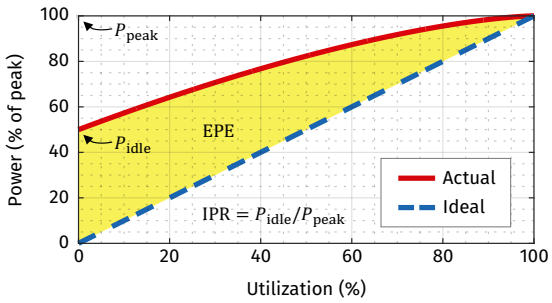


Fig. 2. Illustration of energy proportionality metrics.

In the presence of traffic dynamics, an optical link without proper power management may still consume energy when it is idle. As a result, the effective energy efficiency of an optical network can be orders of magnitude worse than the nominal energy efficiency of individual links [3]. Several techniques were proposed to reconfigure the laser power at application runtime [35]–[38]. However, these techniques target the optical network-on-chip (ONoC) [39], where the traffic patterns are significantly different from off-chip scenarios. For ONoCs, the inter-arrival time between two data transmission requests usually ranges from nanoseconds to hundreds of nanoseconds [40], much smaller than the thermal time constants of microring tuning (~1 µs to ~1 ms) [41]–[43]. Thus, power reconfiguration for the microring tuning circuitry was deemed unnecessary for ONoCs. However, for datacenters and HPC systems where the links can often stay idle for milliseconds to seconds [22], such reconfiguration capability becomes imperative for the effective energy efficiency of the optical interconnects.

2) Energy proportionality: The energy proportionality of datacenter/HPC interconnects is also becoming more critical as the computation nodes become more energy-proportional over the years. As illustrated in Fig. 2, the energy proportionality can be measured by various metrics, such as the *idle-to-peak ratio* (IPR) [44]:

$$\text{IPR} = P_{\text{idle}} / P_{\text{peak}}, \tag{3}$$

and the *energy proportionality error* (EPE) [45]:

$$\text{EPE} = \int_0^1 |P(u) - u| \, \mathrm{d}u, \tag{4}$$

where $u \in [0, 1]$ is the utilization of the network, and $P(u) \in [0, 1]$ is the normalized network power as a function of the utilization level. The energy proportionality of the optical interconnects also relies on proper management of the link power during idle or low-utilization periods.

C. Scope of This Study

Our POLESTAR strategies aim to improve the *effective* energy efficiency and the energy proportionality of the optical interconnects for datacenter/HPC applications by reconfiguring the power states of the lasers and the microring tuning circuitry on the fly. Note that another line of work on energy-efficient optical interconnects focuses on connectivity reconfiguration, which lets busy links borrow bandwidth from idle ones to reduce the need for bandwidth overprovisioning [46], [47]. POLESTAR is orthogonal to and can be applied on top of these techniques because power reconfiguration of optical devices is always applicable as long as traffic dynamics exist.

III. STRATEGY DESIGN AND MOTIVATION ANALYSIS

POLESTAR features a collection of power reconfiguration strategies for optical interconnects designed with the following questions in mind:

1) What are the power states that a link can switch to when it becomes idle?
2) What data rate should be assigned when a data transmission request is received?
3) How can the reconfiguration mechanism of the power states adapt to the traffic patterns that are spatially non-uniform and constantly changing?

We thus elaborate on the design of our POLESTAR strategies in three steps.

A. Power Reconfiguration for Idle Links

Due to the turn-on delay of the lasers and the microring tuning circuitry, it is unwise to immediately turn off all the components of an optical link as soon as it becomes idle, in case there is an upcoming transmission request shortly after. POLESTAR extends the laser power scaling concept proposed in [37] into a fine-grained power reconfiguration strategy that includes both the lasers and the microring tuning circuitry. As summarized in Table I and illustrated in Fig. 3, besides `ON` and `OFF`, two additional states are introduced to the optical link, namely `READY` and `STANDBY`. The switching between the power states depends on how long the link has remained idle, where two threshold values, t_1 and t_2, come into play:

- When the link becomes idle, it is first switched to the `READY` state from the `ON` state by reducing the laser bias current to its threshold. At this state, the laser consumes significantly less power with only spontaneous emission and maintains the capability of a fast turn-on. The reconfiguration delay from `READY` to `ON` is roughly proportional to the differential carrier lifetime of the laser and in the order of several nanoseconds [29].
- When the link has remained idle for longer than t_1, it is further switched to the `STANDBY` state by turning the laser bias off. The reconfiguration delay from `STANDBY` to `ON` is roughly proportional to the total carrier lifetime of the laser and can be up to ~100 ns [48].
- When the link has remained idle for longer than t_2 ($t_2 \geqslant t_1$), it is finally switched to the `OFF` state by suspending the microring tuning circuitry. The reconfiguration delay from `OFF` to `ON` is dominated by the thermal time constants of microring tuning (up to ~1 ms) [41]–[43].

It is further assumed that during the reconfiguration delay, the link consumes the same power as that of the final state but cannot transmit data. Note that in this study, the time required for clock recovery and frame synchronization is not counted toward the reconfiguration delay. In contrast to traditional electrical interconnects that maintain synchronization between two connected ports by filling idle periods with repetitive patterns, optical interconnects for datacenters and HPC systems usually employ burst-mode receivers to perform synchronization of clock and data amid bursty traffic [49]. Such receiver technologies prepend some carefully designed preamble bits to the data packets and can achieve fast synchronization of clock and data within several nanoseconds [50]. The synchronization time is considered as a small overhead to the packet transfer time independent of the initial power state of the optical link.

TABLE I
AVAILABLE POWER STATES FOR IDLE LINKS.

Idle time	Power state	Laser bias	MRR tuning	Turn-on delay
0	`ON`	$> I_{\text{th}}$	On	-
$(0, t_1]$	`READY`	$\sim I_{\text{th}}$	On	Small
$(t_1, t_2]$	`STANDBY`	0	On	Medium
$(t_2, +\infty)$	`OFF`	0	Off	Large

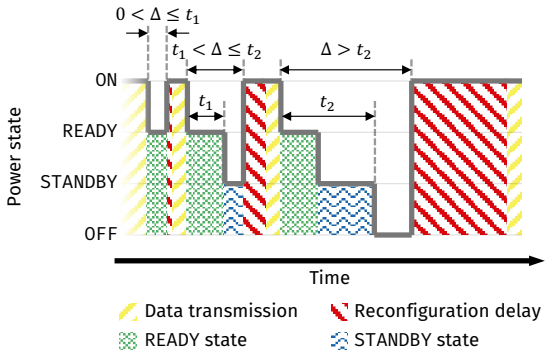
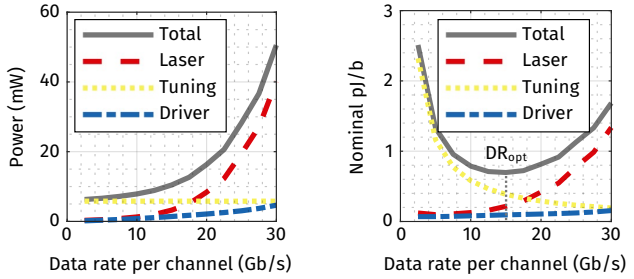


Fig. 3. Power state reconfiguration for idle links (not drawn to scale for illustration purpose). Δ denotes the idle time since the last transmission.

As shown in Fig. 3, the state transition profile of a link during the idle period is determined by the relationship between Δ (i.e., the idle time since the last transmission) and the values of t_1 and t_2. If $t_1 = t_2 = 0$, the strategy reduces to simple `ON-OFF` reconfiguration that switches the link off as soon as it becomes idle. While positive t_1 and t_2 could benefit some transmissions with reduced turn-on delay, both thresholds cannot be infinitely large as the `READY` and `STANDBY` states themselves also consume energy. Therefore, t_1 and t_2 should be made adjustable at application runtime for the constantly changing Δ, in other words, traffic-adaptive. Section III-C will further elaborate on this design objective.

B. Power Reconfiguration for Active Links

Besides providing multiple power states for idle links, POLESTAR also features two strategies that take effect when a link receives a data transmission request and becomes active. The first strategy seeks to improve the energy efficiency by using an *intermediate data rate* (as opposed to the maximum supported) for select network activities. As illustrated in Fig. 4, due to the highly nonlinear growth of laser power consumption at higher data rates, the optimal pJ/b of an optical link could occur at a data rate slower than the peak, which we denote by DR_{opt}. (The power models for the optical link will be revisited in Section IV-C2.) In this study, we propose to use DR_{opt} for transmitting control messages (whose volume is usually orders of magnitude smaller than the actual data) and data that do not serve as the input to any pending tasks. According to our observation from



(a) (b)

Fig. 4. (a) Power consumption and (b) nominal energy efficiency of an optical link as functions of the data rate per channel.

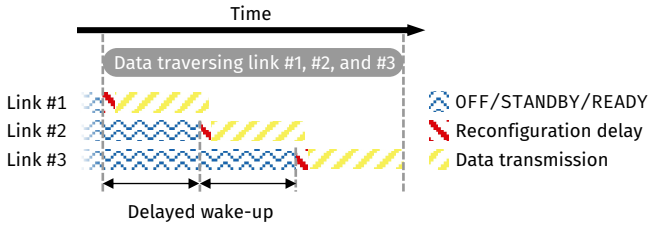


Fig. 5. Illustration of delayed wake-up of downstream links for a network activity traversing multiple links.

the Alibaba cluster traces, these messages can take up over 25 % of all network activities, offering a reasonable opportunity for energy optimization.

Another strategy accounts for the scenario where a network activity traverses multiple links. This is common for datacenter/HPC interconnects as the network topology is usually hierarchical rather than point-to-point. As illustrated in Fig. 5, when a data transmission request is received, instead of immediately waking up all the links en route, POLESTAR wakes up a downstream link with a delay d computed as

$$d_i = \max\left(0, \sum_{j=1}^{i-1} s/\mathrm{DR}_j - \delta_i\right), \tag{5}$$

where i denotes the sequential position of the target link en route, s denotes the data size, DR_j denotes the data rate assigned to the data by link #j, and δ_i denotes the reconfiguration delay corresponding to the current power state of link #i. Eq. (5) ensures that only the reconfiguration delay of the first link will affect the overall communication time, and those of the downstream links can be already in the ON state by the time the data arrives.

C. Making the Reconfiguration Mechanism Traffic-Adaptive

As the traffic patterns vary from link to link and changes with time, POLESTAR features a mechanism that adjusts the values of the idle thresholds, t_1 and t_2, at application runtime. Referring to Fig. 3, an upper bound for t_1 can be calculated from

$$P_{\mathrm{READY}} \cdot t_1 + P_{\mathrm{ON}} \cdot \delta_{\mathrm{READY}} \leqslant P_{\mathrm{STANDBY}} \cdot t_1 + P_{\mathrm{ON}} \cdot \delta_{\mathrm{STANDBY}}, \tag{6}$$

which gives

$$t_1 \leqslant t_{1,\max} = \frac{P_{\mathrm{ON}}\,(\delta_{\mathrm{STANDBY}} - \delta_{\mathrm{READY}})}{P_{\mathrm{READY}} - P_{\mathrm{STANDBY}}}, \tag{7}$$

where P_* is the link power consumption of state $*$, and δ_* is the turn-on delay of the link from state $*$. In other words, for an idle time Δ greater than $t_{1,\max}$, it is less energy-consuming to skip the READY state and use the STANDBY state for the entire idle period despite a larger turn-on delay. Similarly, an upper bound for t_2 can be calculated from

$$P_{\mathrm{READY}} \cdot t_1 + P_{\mathrm{STANDBY}}\,(t_2 - t_1) + P_{\mathrm{ON}} \cdot \delta_{\mathrm{STANDBY}} \leqslant P_{\mathrm{ON}} \cdot \delta_{\mathrm{OFF}}, \tag{8}$$

which gives

$$t_2 \leqslant t_{2,\max} = \frac{P_{\mathrm{ON}}\,(\delta_{\mathrm{OFF}} - \delta_{\mathrm{STANDBY}}) - (P_{\mathrm{READY}} - P_{\mathrm{STANDBY}})\,t_1}{P_{\mathrm{STANDBY}}}. \tag{9}$$

For an idle time Δ greater than $t_{2,\max}$, it becomes less energy-consuming if the link remains off for the entire idle period despite the even larger OFF-ON delay.

An ideal mechanism is expected to predict the next Δ and adjust the idle thresholds to ensure that

$$\begin{cases} \Delta \leqslant t_1 \leqslant t_{1,\max}, & \text{if } \Delta \in (0, t_{1,\max}], \quad (10a) \\ t_1 = 0, & \text{if } \Delta \in (t_{1,\max}, +\infty); \quad (10b) \end{cases}$$

and

$$\begin{cases} \Delta \leqslant t_2 \leqslant t_{2,\max}, & \text{if } \Delta \in (0, t_{2,\max}], \quad (11a) \\ t_2 = 0, & \text{if } \Delta \in (t_{2,\max}, +\infty). \quad (11b) \end{cases}$$

However, it is impossible to predict the exact length of an upcoming idle period. Moreover, as the runtime adjustment of t_1 and t_2 is local to each link, it is desirable that the implementation could be done at the router level with simple hardware logic and require no centralized management or sophisticated software support. To this end, we propose a simplified mechanism for the runtime adjustment of the idle thresholds. Taking the adjustment of t_1 as an example (the adjustment of t_2 follows the same principle), we define that an idle period Δ is

$$\begin{cases} \textit{in-range}, & \text{if } \Delta \in (0, t_{1,\max}]; \quad (12a) \\ \textit{out-of-range}, & \text{if } \Delta \in (t_{1,\max}, +\infty). \quad (12b) \end{cases}$$

Then, by recording this piece of information for historic idle periods, the simplified mechanism predicts whether the next Δ will be in-range or not, instead of predicting its exact length. The mapping of Δ into binary states enables us to explore simple digital logic for adjusting the idle thresholds, inspired by the extensively-studied branch prediction strategies in the computer architecture domain [51]. Specifically, in this chapter, we employ a simple one-level mechanism for adjusting t_1 and t_2 based on the prediction of Δ range.

Inspired by the existing one-level branch prediction techniques [51], the one-level adjustment mechanism for idle thresholds maintains an n-bit saturating up-down counter. If the last recorded idle period (denoted by Δ_{last}) is in-range (Eq. (12a)), the counter increases by one (and saturates at $(2^n - 1)$); otherwise, the counter decreases by one (and saturates at 0). Then, the idle thresholds, t_1 and t_2, are updated based on the following criteria:

1) If the counter is greater than $(2^n - 1)/2$, an operation called *match* is performed:

$$\begin{cases} t_1 \leftarrow \min\left[\max\left(t_1, \Delta_{\mathrm{last}}\right), t_{1,\max}\right], & (13a) \\ t_2 \leftarrow \min\left[\max\left(t_2, \Delta_{\mathrm{last}}\right), t_{2,\max}\right]; & (13b) \end{cases}$$

2) otherwise, if the counter is smaller than $(2^n - 1)/2$, an operation called *reset* is performed:

$$t_1, t_2 \leftarrow 0. \tag{14}$$

The one-level idle threshold adjustment mechanism based on the n-bit counter can be represented by finite-state machines. Fig. 6 shows

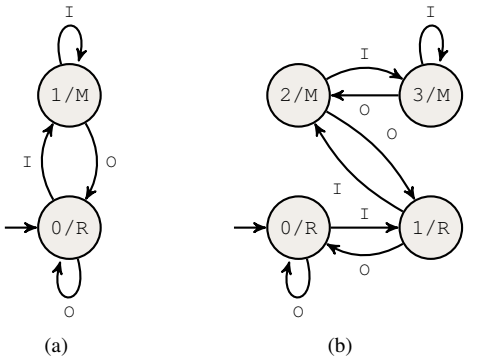


Fig. 6. State diagrams of the one-level idle threshold adjustment mechanism using (a) a 1-bit saturating counter and (b) a 2-bit saturating counter, where I: Δ_{last} in-range (Eq. (12a)); O: Δ_{last} out-of-range (Eq. (12b)); M: match operation (Eqs. (13a) and (13b)); and R: reset operation (Eq. (14)).

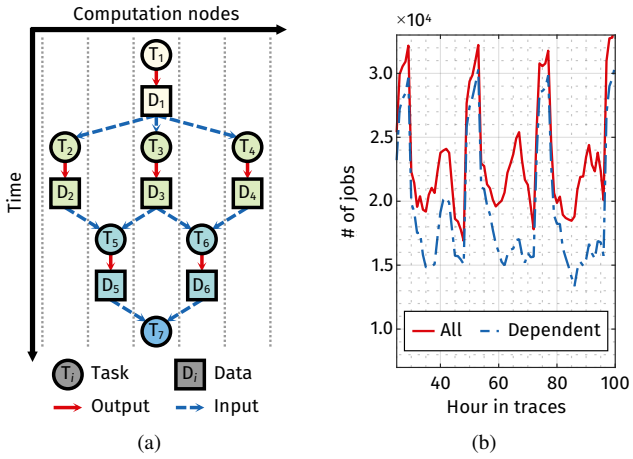


Fig. 7. (a) DAG representation of a job containing multiple tasks and data dependencies and (b) temporal distribution of jobs in the Alibaba traces.

the cases for $n = 1$ and $n = 2$, which are analogous to the 1-bit and 2-bit branch predictors described in [51].

Note that another source of hardware overhead is for keeping track of the length of each idle period Δ. Hardware-assisted techniques, such as the 1-bit busy/idle register proposed in [52], is able to extract this information with several logic gates per link.

IV. Simulation Setup

A. Overview of the Simulation Environment

1) Dataset: Among various public datasets of datacenter/HPC workloads [22], we opted for the traces recorded on a production cluster of Alibaba, Inc. [33] to evaluate our POLESTAR strategies. The Alibaba traces, published in 2018, contain execution details of ~1.3 billion tasks on ~4000 machines over eight days. Besides its recentness and large size, another reason for choosing the Alibaba traces is the inclusion of task dependency information. As illustrated in Fig. 7a, a group of dependent tasks (often referred to as a job or a workflow) can be characterized by a directed acyclic graph (DAG) describing the inter-task data dependencies. A task is assumed to generate a single piece of output data, which may serve as the input data to multiple child tasks. A task can also depend on the output of multiple parent tasks. As the tasks are distributed to different computation nodes, the data dependencies among the tasks result in communication patterns across the network. As shown in Fig. 7b, an average of 80 % of the jobs in the Alibaba traces are dependent ones. The task dependency information enables us to investigate the impact of POLESTAR on application execution time because we can identify the tasks that are subsequently affected by changing the communication.

2) Simulator: For replaying the Alibaba traces and simulating our POLESTAR strategies in operation, we employed two open-source tools, namely WREHCN [32], a library for workflow management, and SimGrid [31], a matured simulation framework for distributed computing platforms. WREHCN features a DAG processing engine which we tweaked to parse the Alibaba traces and generate simulation entities recognizable by SimGrid. Then, SimGrid, modified and implemented with our POLESTAR strategies, simulates the task execution and the network communication.

B. Trace Preprocessing

The Alibaba traces do not include statistics on data size or communication time. Instead, only the execution time of each task

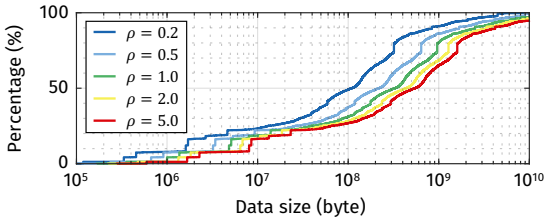


Fig. 8. Cumulative distribution of data size w.r.t. different values of ρ.

is recorded. To this end, we denote the execution time of task T_i (Fig. 7a) by $t_{\text{exec},i}$ and assume that it consists of two parts: 1) the time spent waiting for data communication from its parent tasks, $t_{\text{comm},i}$; and 2) the time spent on actual computation, $t_{\text{comp},i}$. We then define a parameter ρ, which we refer to as the *communication-to-computation ratio*, and thus

$$t_{\text{comm},i} = \frac{\rho}{1+\rho} t_{\text{exec},i}, \tag{15}$$

$$t_{\text{comp},i} = \frac{1}{1+\rho} t_{\text{exec},i}. \tag{16}$$

The SimGrid simulator can strictly enforce $t_{\text{comp},i}$ for each task in the simulation, but $t_{\text{comm},i}$ has to be simulated based on the data size information. According to the observation of a linear relationship between the input data size and the computation time for various datacenter applications [53], we denote the size of data D_i (Fig. 7a) by s_i and further assume that

$$s_i = a \cdot \sum_{j \in \mathbb{C}_i} t_{\text{exec},j} / |\mathbb{P}_j| + b, \tag{17}$$

where $\mathbb{C}_i$ is the set of child tasks of task T_i, and $\mathbb{P}_j$ is the set of parent tasks of task T_j. Finally, we find proper values for a and b by solving

$$\min_{a,b} \left\| \hat{\boldsymbol{t}}_{\text{comm}} - \boldsymbol{t}_{\text{comm}} \right\|^2, \tag{18}$$

where $\hat{\boldsymbol{t}}_{\text{comm}}$ is the simulated values of the communication time for all tasks by replaying the traces in SimGrid, and $\boldsymbol{t}_{\text{comm}}$ is the expected values computed from Eq. (15).

In this study, we assume $\rho = 0.5$ by default, as it was observed in [54] that the time spent on data communication is roughly half

of that spent on computation. However, we also vary the value of ρ between 0.2 and 5 to account for broader scenarios. Fig. 8 shows the cumulative distributions of data sizes generated for the Alibaba traces w.r.t. different values of ρ. The overall range and shape of the distributions are comparable to other communication traces used in system-level studies of optical interconnects [47].

C. Network Configuration

1) Topology: The network topologies for datacenters and HPC systems can be categorized as direct or indirect. In *direct* topologies, every router has computation nodes directly connected to it, while in *indirect* topologies, some routers are not exposed to the computation nodes and only connect to other routers [3]. In this study, we evaluate our POLESTAR strategies for one representative topology from each category. For indirect topologies, we choose *Fat-Tree*, which has been widely adopted in many real-world clusters. For direct topologies, we choose *Dragonfly*, which is promising for future high-throughput datacenter/HPC networks [55]. While both topologies have been implemented in the SimGrid simulator, as described in [56], we added a property to the link implementation specifying whether it is electrical or optical. We further configure the links above the first-level routers as optical, which are the reconfiguration targets of POLESTAR.

2) Power models: We assume the pairing of a 24-channel comb laser (reported in [7]) and a 24-channel microring-based transceiver (reported in [34]) to form an optical link with a maximum data rate of 30 Gb/s per channel. As Eq. (1) in Section II-B1 has mentioned, the computation of link energy relies on models for the power of the laser (P_{laser}), the microring tuning circuitry (P_{tuning}), and the electrical driver circuitry (P_{driver}). The power models employed in this study are summarized in Table II and further explained as follows.

a) Laser: We assume a Gaussian-shaped comb spectrum with a spectrum efficiency $\eta = P_{\text{usable}}/P_{\text{total}} \approx -3.2\,\text{dB}$ [48]. The optical power at the laser output must be high enough so that the following power budget equation holds for any channel $k \in \{1, 2, \cdots, m\}$:

$$P_{\text{comb},k} \cdot \alpha_k \geqslant P_{\text{sensitivity}}. \tag{19}$$

Here, m is the number of DWDM channels; $P_{\text{comb},k}$ is the optical power of the kth comb line; $\alpha_k \in (0, 1)$ is the accumulated loss of optical power as the light travels along channel k [19], [25]; and $P_{\text{sensitivity}}$ is the sensitivity requirement of the receiver, which is a function of the data rate [25]. The laser subjects to a wall-plug efficiency (WPE) when converting electrical power into optical power, and then the spectrum efficiency (η) that accounts for the usable portion of the comb lines:

$$P_{\text{laser}} \cdot \text{WPE} \cdot \eta = \sum_{k=1}^{m} P_{\text{comb},k}. \tag{20}$$

Based on Eqs. (19) and (20), the laser power consumption can be computed for various data rates, as seen in Fig. 4a.

b) Microring tuning: The microring tuning power is estimated from the variation distribution of the resonance wavelengths measured from a wafer fabricated with 66 24-channel transceivers [34]. The transceivers have a channel spacing of ~0.35 nm (~61 GHz in the O-band) and are designed to support up to 30 Gb/s per channel. For lower data rates, we assume the same channel spacing (0.35 nm) despite that denser channels may be used. As a results, the microring tuning power is considered independent of the data rate in this study. Fig. 4a shows the modeled tuning power assuming a thermal tuning efficiency of 0.15 nm/mW.

TABLE II
POWER MODELS FOR OPTICAL LINKS.

Laser efficiency			
Wall-plug efficiency	5 % [7]	Spectrum efficiency	-3.2 dB [48]
Data rate dependency			
$P_{\text{sensitivity}}$	[25]	P_{driver}	[57]
Microring			
Passing loss	0.2 dB [19]	Drop-port loss	1 dB [19]
Insertion loss	0.5 dB [19]	Tuning efficiency	0.15 nm/mW [58]
Waveguide			
Coupling loss	1 dB [25]	Propagation loss	1 dB/cm [25]

TABLE III
CORNER CASES FOR THE RECONFIGURATION DELAY.

	Corner	FF	FS	SF	SS
	READY-ON	Assumed 1/10 of STANDBY-ON delay			
Delay	STANDBY-ON	10 ns	10 ns	100 ns	100 ns
	OFF-ON	1 µs	1 ms	1 µs	1 ms

c) Electrical driver: The power models for the driver circuitry, including the modulator drivers at the Tx side and the transimpedance amplifiers (TIAs) at the Rx side, are taken from [57], both depending on the data rate. Note that in this study, the serializer/deserializer (SerDes) circuitry is considered part of the computation nodes rather than the link drivers. Therefore, its power consumption is not included in the link power. Similar assumptions are found in other literature, such as [21].

3) Reconfiguration delay: The reconfiguration delay, described in Section III-A, is a key parameter affecting the trade-off between energy saving and application execution time. In this study, we consider four corner cases corresponding to the fast/slow stabilization of the laser/microring tuning [29], [41]–[43], [48], as summarized in Table III.

V. EVALUATION

A. Case Study for Strategy Effectiveness

We first conduct a case study of our POLESTAR strategies for a simulated Fat-Tree cluster with 64 nodes. The SS corner in Table III is assumed for conservativeness. A one-hour segment of the Alibaba traces is used for stressing the network with the job arrival rate down-sampled to match the 64 nodes. The idle threshold adjustment mechanism is configured with $n = 2$ (Fig. 6b), which achieves a prediction accuracy of ~86 % for the match/reset operations. Increasing the value of n beyond 2 does not bring further improvement to the prediction accuracy.

1) Improvement of effective energy efficiency: The effective energy efficiency can be calculated for the overall network or for each individual link, using Eq. (2). In Fig. 9a, we first show the improvement of effective pJ/b of the network achieved by different strategies compared to a baseline scenario where the links are always kept on. Among existing strategies [35]–[38] that only consider the laser power as a tuning knob, we include the dynamic laser power scaling (DLPS) strategy [37] for comparison. Note that DLPS also proposes to use an intermediate data rate for transmissions that can finish within a clock cycle. However, such transmissions are not observed in the Alibaba traces where the data size is significantly larger than the on-chip scenario discussed in [37]. As observed in Fig. 9a, POLESTAR is able to reduce the effective pJ/b of the network by ~85 % when all of its featured strategies are enabled (the rightmost

bar), notably outperforming DLPS (the leftmost bar). This indicates the necessity of extending the power reconfiguration mechanism to the microring tuning circuitry in datacenter/HPC interconnects. Among the POLESTAR strategies, the power reconfiguration for idle links contributes the most energy saving (~57 %), followed by the delayed wake-up (~20 %), and finally the intermediate data rate (~8 %). The second bar corresponds to the case where $t_1 = t_2 = 0$, a.k.a., the ON-OFF strategy. The fact that each data transmission must start with an OFF-ON delay results in less energy saving. The third bar corresponds to the case where $t_1 = t_{1,\max}$ and $t_2 = t_{2,\max}$. The lack of traffic adaptability for the idle thresholds also results in slightly less energy saving compared to POLESTAR at full play.

In Fig. 9b, we plot the effective pJ/b for each individual link in ascending order for different reconfiguration strategies. Even though the links in the network are identical and share the same nominal energy efficiency, their effective pJ/b could be vastly different due to the unbalanced traffic patterns across the network. For the baseline and the DLPS strategy where the links are never turned off, it is especially devastating to the effective pJ/b of the low-utilization links due to the small number of bits transmitted. The flattest curve in Fig. 9b belongs to POLESTAR with all features enabled, indicating that the POLESTAR strategies are particularly effective for managing the energy of idle and low-utilization links.

2) Overhead to application execution time: Fig. 9c demonstrate the trade-off between energy saving and application execution time observed in this case study. The horizontal axis is again the saving of effective pJ/b of the network for each strategy compared. The vertical axis plots the overhead to the execution time of all tasks, where the cross signs indicate the mean values. Intuitively, the DLPS strategy incurs the smallest overhead as it only involves laser reconfiguration. However, the energy saving achieved by DLPS is far from ideal. Our POLESTAR strategies, on the other hand, can significantly improve the energy saving while still keeping the overhead to the application execution time manageable. Notably, POLESTAR with traffic adaptability enabled (④ in Fig. 9c) can limit the average overhead of execution time within 0.18 %, outperforming the two strategies with static t_1 and t_2 (② and ③) in both energy saving and application execution time.

3) Improvement of energy proportionality: Similar to the effective energy efficiency, the energy proportionality can also be computed for either the overall network or each individual link. For the network, its utilization rate at a specific time is calculated as the sum of bandwidth capacity requested by the active links over the total bandwidth capacity supported by the network. The power consumption of the network comes from the active links, as well as the idle links that are in READY/STANDBY states or in state transition. Fig. 10 plots all of the utilization-power pairs observed during the simulation of the 64-node Fat-Tree network with our POLESTAR strategies. An averaging curve is also drawn as the energy proportionality curve for the overall network, which is significantly closer to an ideal energy-proportional curve compared to the baseline scenario that always keeps the links on.

As for the energy proportionality of an individual link, we calculate its average power consumption for a specific utilization rate as the accumulated energy divided by the total time spent at that utilization rate. For example, the average idle power of a link is computed as the energy consumed during its idle periods, including the energy consumption of the READY and STANDBY states, as well as those consumed during state transition. We compute the idle-to-peak power ratio, as defined in Section II-B2, for all links in the simulated network, which demonstrates an average of ~82 % improvement

Fig. 9. Case study of POLESTAR for a 64-node Fat-Tree topology: (a) improvement of effective energy efficiency for the network; (b) effective energy efficiency for individual links; and (c) trade-off between energy saving and application execution time.

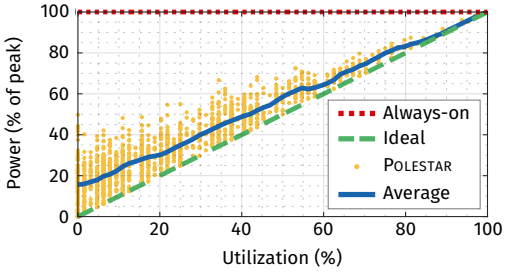


Fig. 10. Energy proportionality curve for the overall network with POLESTAR, averaged from the simulated utilization-power pairs.

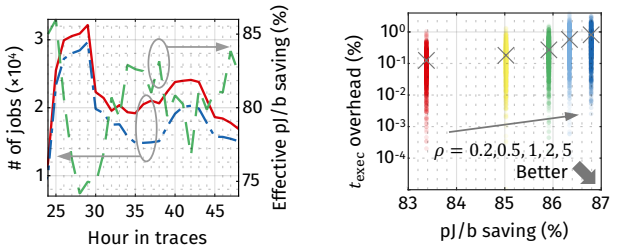
compared to the baseline scenario that always keeps the links on.

B. Scalability Analyses

We then evaluate the scalability of the POLESTAR strategies for a broader range of configurations.

1) Network loads:

a) Different hours in the traces: As shown in Fig. 7b, the workloads in datacenters can drastically fluctuate with time. Therefore, we evaluate the POLESTAR strategies for 24 consecutive trace hours with other assumptions unchanged. As shown in Fig. 11a, the improvement of network energy efficiency achieved by POLESTAR also fluctuates with the workloads. The reduction in energy saving at higher workloads could be explained by the increased link utilization. As the load balancing mechanism of the computation infrastructure tends to schedule tasks uniformly across the nodes, the traffic patterns resulted from task execution also becomes more spatially uniform under higher workloads. In other words, there are less idle links in the network, which means less opportunities for idle power reconfiguration. Moreover, the increased link utilization also reduces the opportunities for using the intermediate data rate. Nevertheless, POLESTAR can still achieve a ~72 % reduction of the network



(a) (b)

Fig. 11. Evaluation of POLESTAR strategies for (a) different hours in the traces and (b) different values for ρ.

pJ/b for the worst case, demonstrating a considerable scalability for network loads.

b) Communication-to-computation ratio (ρ): Datacenters and HPC systems running different applications could have different traffic characteristics. As some applications are computational-intensive while others are communication-bounded, we also vary parameter ρ, the communication-to-computation ratio defined in Section IV-B, to study its impact on the POLESTAR strategies. Fig. 11b plots the trade-off between the attainable energy saving and the execution time overhead w.r.t. different values for ρ. The horizontal axis is the saving of effective pJ/b of the network. The vertical axis plots the overhead to the execution time of all jobs, where the cross signs indicate the mean values. As can be observed, applying POLESTAR for applications with a larger ρ could lead to greater energy saving at the cost of larger overhead to application execution time. Overall, our POLESTAR strategies scale well across a wide range of ρ by achieving at least 83 % of energy saving with less than 0.8 % overhead to the application execution time.

2) Corner cases for the reconfiguration delay: To account for potential advances in device design in the near future, we also evaluate POLESTAR for various technology corners mentioned in Table III. As summarized and observed in Table IV, POLESTAR can achieve even greater energy saving compared to the SS corner evaluated in the previous sections by using microrings with faster thermal time constants. Further reducing the laser turn-on delay, on the other hand, has limited impact on the effectiveness of POLESTAR strategies, as the laser turn-on delay is already small enough compared to most communication transactions in the traces. This motivates future effort on device design for datacenter/HPC interconnects to focus on reducing the stabilization time required for microring tuning.

3) Topology: Finally, we evaluate the scalability of POLESTAR for both the Fat-Tree and the Dragonfly topologies in a simulated network with up to 256 nodes. Further increasing the number of nodes requires excessive memory space during simulation, which is beyond the capability of our server. Table V summarizes the simulated results for energy saving. It is first observed that the energy saving achieved by POLESTAR for the Dragonfly topology is slightly smaller than that for the Fat-Tree topology. A possible reason is that the Dragonfly topology strongly relies on a grouped structure where intra-group links are assumed to be electrical. Therefore, for the same network size, the Dragonfly topology contains less optical links, thus offering less opportunities for reconfiguration. Another observation is the increase of achievable energy saving with the network size, possibly due to the lower utilization of links in larger networks. Overall, the effectiveness of POLESTAR strategies on various network topologies and sizes remains significant.

TABLE IV
ENERGY IMPROVEMENT OF POLESTAR AT FOUR TECHNOLOGY CORNERS.

Corner	FF	FS	SF	SS
Effective pJ/b improvement	87.45 %	84.78 %	87.86 %	85.02 %

TABLE V
ENERGY IMPROVEMENT OF POLESTAR FOR DIFFERENT TOPOLOGIES AND NETWORK SIZES.

# of nodes	64	128	256
Fat-Tree	85.02 %	86.18 %	87.24 %
Dragonfly	82.77 %	83.01 %	83.89 %

VI. CONCLUSION AND FUTURE WORK

In this study, we proposed POLESTAR, i.e., POwer LEvel Scaling with Traffic-Adaptive Reconfiguration, for microring-based optical interconnects. Featuring a collection of runtime reconfiguration strategies that target the power states of the lasers and the microring tuning circuitry, POLESTAR demonstrates remarkable effectiveness for improving the energy efficiency and energy proportionality of underutilized datacenter/HPC interconnects. Through traffic-adaptive adjustment of the reconfiguration mechanism, POLESTAR achieves a reasonable balance between energy saving and application execution time. Good scalability across topologies, network loads, and potential advances in optical device design is also observed. POLESTAR is extensible by incorporating more reconfiguration strategies and improving existing ones. With future work targeting better traffic prediction techniques and the possible inclusion of runtime traffic scheduling, POLESTAR paves a promising way to the energy-efficient and energy-proportional optical interconnects for future datacenter/HPC applications.

ACKNOWLEDGMENT

The author from HKUST would like to acknowledge the sponsorship of the Research Grants Council (RGC) of Hong Kong SAR, China. This work was partially supported by Hong Kong General Research Fund (GRF) 16203918.

REFERENCES

[1] D. A. Reed and J. Dongarra, "Exascale computing and big data," *Commun. ACM*, vol. 58, no. 7, pp. 56–68, Jun. 2015.
[2] R. Lucas *et al.*, "DOE advanced scientific computing advisory subcommittee (ASCAC) report: Top ten exascale research challenges," USDOE Office of Science, SC, United States, Tech. Rep., Feb. 2014.
[3] S. Rumley *et al.*, "Evolving requirements and trends of HPC," in *Springer Handbook of Optical Networks*, B. Mukherjee *et al.*, Eds. Cham: Springer International Publishing, 2020, pp. 725–755.
[4] R. G. Beausoleil, M. McLaren, and N. P. Jouppi, "Photonic architectures for high-performance data centers," *IEEE J. Sel. Top. Quantum Electron.*, vol. 19, no. 2, p. 3700109, Mar. 2013.
[5] R. G. Beausoleil, "Large-scale integrated photonics for high-performance interconnects," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 7, no. 2, pp. 1–54, Jun. 2011.
[6] M. Ashkan Seyedi *et al.*, "Overview of silicon photonics components for commercial DWDM applications," in *Adv. Photonics Congr. (IPR, Networks, NOMA, PVLED, SPPCom)*. OSA, 2019, p. ITh1A.3.
[7] D. Liang *et al.*, "Integrated green DWDM photonics for next-gen high-performance computing," in *Opt. Fiber Commun. Conf. (OFC)*. OSA, 2020, p. Th1E.2.
[8] Y. London *et al.*, "Performance requirements for terabit-class silicon photonic links based on cascaded microring resonators," *J. Light. Technol.*, vol. 38, no. 13, pp. 3469–3477, Jul. 2020.

[9] A. H. Ahmed *et al.*, "Silicon-photonics microring links for datacenters—challenges and opportunities," *IEEE J. Sel. Top. Quantum Electron.*, vol. 22, no. 6, pp. 194–203, Nov. 2016.

[10] C. A. Thraskias *et al.*, "Survey of photonic and plasmonic interconnect technologies for intra-datacenter and high-performance computing communications," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 4, pp. 2758–2783, 2018.

[11] Q. Cheng *et al.*, "Recent advances in optical technologies for data centers: a review," *Optica*, vol. 5, no. 11, p. 1354, Nov. 2018.

[12] J. C. Norman *et al.*, "A review of high-performance quantum dot lasers on silicon," *IEEE J. Quantum Electron.*, vol. 55, no. 2, pp. 1–11, Apr. 2019.

[13] A. V. Krishnamoorthy *et al.*, "Exploiting cmos manufacturing to reduce tuning requirements for resonant optical devices," *IEEE Photonics J.*, vol. 3, no. 3, pp. 567–579, Jun. 2011.

[14] M. Georgas *et al.*, "Addressing link-level design tradeoffs for integrated photonic interconnects," in *Cust. Integr. Circuits Conf. (CICC)*. IEEE, Sep. 2011, pp. 1–8.

[15] H. Li *et al.*, "Towards maximum energy efficiency in nanophotonic interconnects with thermal-aware on-chip laser tuning," *IEEE Trans. Emerg. Top. Comput.*, vol. 6, no. 3, pp. 343–356, Jul. 2018.

[16] R. Wu *et al.*, "Pairing of microring-based silicon photonic transceivers for tuning power optimization," in *23rd Asia South Pacific Des. Autom. Conf. (ASP-DAC)*. IEEE, Jan. 2018, pp. 135–140.

[17] Y. Wang *et al.*, "Energy-efficient channel alignment of dwdm silicon photonic transceivers," in *Des. Autom. Test Eur. Conf. Exhib. (DATE)*. IEEE, Mar. 2018, pp. 601–604.

[18] Y. Wang *et al.*, "Bidirectional tuning of microring-based silicon photonic transceivers for optimal energy efficiency," in *24th Asia South Pacific Des. Autom. Conf. (ASP-DAC)*. ACM, Jan. 2019, pp. 370–375.

[19] Y. Wang *et al.*, "Energy efficiency and yield optimization for optical interconnects via transceiver grouping," *J. Light. Technol.*, vol. 39, no. 6, pp. 1567–1578, Mar. 2021.

[20] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *IEEE Computer*, vol. 40, no. 12, pp. 33–37, Dec. 2007.

[21] Y. London *et al.*, "Energy efficiency analysis of comb source carrier-injection ring-based silicon photonic link," *IEEE J. Sel. Top. Quantum Electron.*, vol. 26, no. 2, pp. 1–13, Mar. 2020.

[22] L. Versluis *et al.*, "The workflow trace archive: Open-access data from public and private computing infrastructures," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 9, pp. 2170–2184, Sep. 2020.

[23] A. K. Kodi, B. Neel, and W. C. Brantley, "Photonic interconnects for exascale and datacenter architectures," *IEEE Micro*, vol. 34, no. 5, pp. 18–30, Sep. 2014.

[24] Y. Shen *et al.*, "Silicon photonics for extreme scale systems," *J. Light. Technol.*, vol. 37, no. 2, pp. 245–259, Jan. 2019.

[25] M. Bahadori *et al.*, "Energy-performance optimized design of silicon photonic interconnection networks for high-performance computing," in *Des. Autom. Test Eur. Conf. Exhib. (DATE)*. IEEE, Mar. 2017, pp. 326–331.

[26] C. Avin *et al.*, "On the complexity of traffic traces and implications," in *Int. Conf. Meas. Model. Comput. Syst.*, vol. 48, no. 1. ACM, Jun. 2020, pp. 47–48.

[27] Q. Wu *et al.*, "Formal control techniques for power-performance management," *IEEE Micro*, vol. 25, no. 5, pp. 52–62, Sep. 2005.

[28] A. Hammadi and L. Mhamdi, "A survey on architectures and energy efficiency in data center networks," *Comput. Commun.*, vol. 40, pp. 1–21, Mar. 2014.

[29] L. A. Coldren, S. W. Corzine, and M. L. Mašanović, "Dynamic effects," in *Diode Lasers and Photonic Integrated Circuits*. Hoboken, NJ, USA: John Wiley & Sons, Ltd, Mar. 2012, ch. 5, pp. 247–333.

[30] M. Borghi *et al.*, "On the modeling of thermal and free carrier nonlinearities in silicon-on-insulator microring resonators," *Opt. Express*, vol. 29, no. 3, p. 4363, Feb. 2021.

[31] H. Casanova *et al.*, "Versatile, scalable, and accurate simulation of distributed applications and platforms," *J. Parallel Distrib. Comput.*, vol. 74, no. 10, pp. 2899–2917, Oct. 2014.

[32] H. Casanova *et al.*, "Developing accurate and scalable simulators of production workflow management systems with WRENCH," *Futur. Gener. Comput. Syst.*, vol. 112, pp. 162–175, Nov. 2020.

[34] Y. Wang *et al.*, "Characterization and applications of spatial variation models for silicon microring-based optical transceivers," in *57th Des. Autom. Conf. (DAC)*. IEEE, Jul. 2020, pp. 1–6.

[33] Alibaba Group, "Alibaba cluster trace program—cluster-trace-v2018," https://bit.ly/2K8DWCa, 2019.

[35] C. Chen and A. Joshi, "Runtime management of laser power in silicon-photonic multibus NoC architecture," *IEEE J. Sel. Top. Quantum Electron.*, vol. 19, no. 2, p. 3700713, Mar. 2013.

[36] Y. Demir and N. Hardavellas, "EcoLaser: An adaptive laser control for energy-efficient on-chip photonic interconnects," in *Int. Symp. Low power Electron. Des. (ISLPED)*. ACM, Aug. 2014, pp. 3–8.

[37] F. Lan *et al.*, "DLPS: Dynamic laser power scaling for optical network-on-chip," in *22nd Asia South Pacific Des. Autom. Conf. (ASP-DAC)*. IEEE, Jan. 2017, pp. 726–731.

[38] Y. Wang and K.-T. Cheng, "Task mapping-assisted laser power scaling for optical network-on-chips," in *IEEE/ACM Int. Conf. Comput.-Aided Des. (ICCAD)*, Nov. 2019, pp. 1–6.

[39] S. Pasricha and M. Nikdast, "A survey of silicon photonics for energy-efficient manycore computing," *IEEE Des. Test*, vol. 37, no. 4, pp. 60–81, Aug. 2020.

[40] Z. Wang *et al.*, "A case study on the communication and computation behaviors of real applications in NoC-based MPSoCs," in *IEEE Comput. Soc. Annu. Symp. VLSI*. IEEE, Jul. 2014, pp. 480–485.

[41] P. Dong *et al.*, "Thermally tunable silicon racetrack resonators with ultralow tuning power," *Opt. Express*, vol. 18, no. 19, p. 20298, Sep. 2010.

[42] W. Bogaerts *et al.*, "Silicon microring resonators," *Laser Photon. Rev.*, vol. 6, no. 1, pp. 47–73, Jan. 2012.

[43] K. Padmaraju *et al.*, "Integrated thermal stabilization of a microring modulator," *Opt. Express*, vol. 21, no. 12, p. 14342, Jun. 2013.

[44] G. Varsamopoulos and S. K. S. Gup, "Energy proportionality and the future: Metrics and directions," in *39th Int. Conf. Parallel Process. Work.* IEEE, Sep. 2010, pp. 461–467.

[45] P. Ruiu *et al.*, "On the energy-proportionality of data center networks," *IEEE Trans. Sustain. Comput.*, vol. 2, no. 2, pp. 197–210, Apr. 2017.

[46] M. Kennedy and A. K. Kodi, "Laser pooling: Static and dynamic laser power allocation for on-chip optical interconnects," *J. Light. Technol.*, vol. 35, no. 15, pp. 3159–3167, Aug. 2017.

[47] M. Y. Teh, Z. Wu, and K. Bergman, "Flexspander: augmenting expander networks in high-performance systems with optical bandwidth steering," *J. Opt. Commun. Netw.*, vol. 12, no. 4, p. B44, Apr. 2020.

[48] M. J. R. Heck and J. E. Bowers, "Energy efficient and energy proportional optical interconnects for multi-core processors: Driving the need for on-chip sources," *IEEE J. Sel. Top. Quantum Electron.*, vol. 20, no. 4, pp. 332–343, Jul. 2014.

[49] N. Parsons and N. Calabretta, "Optical switching for data center networks," in *Springer Handbook of Optical Networks*, B. Mukherjee *et al.*, Eds. Cham: Springer International Publishing, 2020, pp. 795–825.

[50] X.-Z. Qiu, "[OFC 2013 Tutorial OW3G.4] burst-mode receiver technology for short synchronization," in *Optical Fiber Communication Conference (OFC)/National Fiber Optic Engineers Conference 2013*. OSA, 2013, p. OW3G.4.

[51] J. E. Smith, "A study of branch prediction strategies," in *25 years of the international symposia on Computer architecture (selected papers) - ISCA '98*. ACM Press, 1998, pp. 202–215.

[52] C. Xu *et al.*, "Automated OS-level device runtime power management," in *Int. Conf. Archit. Support Program. Lang. Oper. Syst. (ASPLOS)*, vol. 50, no. 4. ACM, Mar. 2015, pp. 239–252.

[53] N. Ahmed *et al.*, "A comprehensive performance analysis of Apache Hadoop and Apache Spark for large scale data sets using HiBench," *J. Big Data*, vol. 7, no. 1, p. 110, Dec. 2020.

[54] M. Chowdhury *et al.*, "Managing data transfers in computer clusters with orchestra," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 98–109, Oct. 2011.

[55] J. Kim *et al.*, "Cost-efficient Dragonfly topology for large-scale systems," *IEEE Micro*, vol. 29, no. 1, pp. 33–40, Jan. 2009.

[56] "Platform Examples — SimGrid documentation," https://simgrid.org/doc/latest/Platform_examples.html.

[57] R. Polster *et al.*, "Efficiency optimization of silicon photonic links in 65-nm cmos and 28-nm fdsoi technology nodes," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 24, no. 12, pp. 3450–3459, Dec. 2016.

[58] K. Yu *et al.*, "A 25 Gb/s hybrid-integrated silicon photonic source-synchronous receiver with microring wavelength stabilization," *IEEE J. Solid-State Circuits*, vol. 51, no. 9, pp. 2129–2141, Sep. 2016.