System simulation methodology of optical interconnects for high-performance computing systems

Avinash Karanth Kodi[1,*] and Ahmed Louri[2,3]

[1]*Department of Electrical Engineering and Computer Science, Ohio University, Athens, Ohio 45701, USA*
[2]*Department of Electrical and Computer Engineering, University of Arizona, Tucson, Arizona 85721, USA*
[3]*E-mail: louri@ece.arizona.edu*
[*]*Corresponding author: avinashk@eecs.ohio.edu*

Received September 29, 2007; accepted October 5, 2007;
published November 6, 2007 (Doc. ID 83117)

The relentless quest for processing speed in the range of teraflops and beyond has accelerated the need for scalable, parallel, high-performance computing (HPC) systems. To meet this high bandwidth and low power requirements, optical interconnect-based system architectures are being implemented by the HPC industry. While computer-aided design tools have significantly assisted electronic system simulation, the field of system level optoelectronics modeling has lagged behind owing to lack of simulation methodologies and tools. This paper explores the design space of developing OPTISIM, a system level modeling and simulation methodology of optical interconnects for HPC systems. OPTISIM can provide computer architects, designers, and researchers a highly optimized, efficient, and accurate discrete-event environment to test various research hypotheses on HPC systems with power-performance implications. For any given optical interconnect architecture with optical transceivers, wavelength assignment, and traffic patterns, OPTISIM provides end users with network throughput, average latency, power loss, power consumption, and signal strength at the output. The proposed OPTISIM simulation methodology is explained with a case study on the performance of an optical HPC architecture called RAPID. © 2007 Optical Society of America

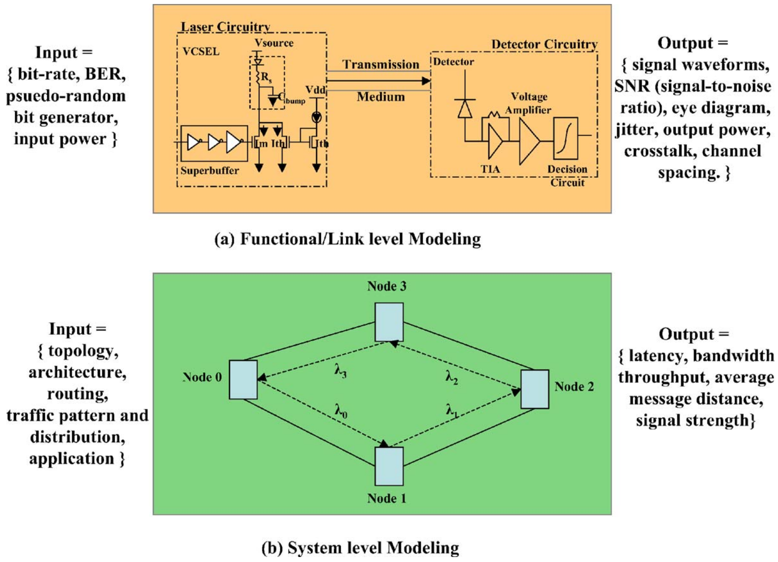
OCIS codes: 200.0200, 200.4650.

1. Introduction

During the past few years, the computer and communication industries have recognized that short-range optical interconnects could potentially provide a cost-effective solution to the increasing bandwidth demands of high-performance computing (HPC) systems [1,2]. Optical interconnects offer several well-known advantages such as higher spatial and temporal bandwidths, lower cross talk independent of data rates, higher interconnect densities, better signal integrity at high frequencies, lower signal attenuation, and lower power requirements at higher bit rates [3,4]; all of which could potentially enhance the scalability and performance of HPC systems.

Modeling and simulation play a very pivotal role in the design of any HPC system [5]. Computer-aided design (CAD) tools are essential to optimize design and system parameters to reduce the fabrication cycle time and end-product cost. While electronic system simulation has made significant progress, the field of optoelectronics modeling has lagged behind due to lack of simulation methodology and tools. Although optoelectronic tools exist that can be used for simulating and designing optical interconnects, they are either suitable for optical link level and not for system level simulation, or they are intended for electrical interconnects and are used for the lack of tools more suitable for optical interconnects' unique needs.

The end-to-end system design and simulation of optical interconnects for HPC systems for intraboard, board-to-board, and backplane applications can be addressed at different levels of abstraction, namely the *functional–link* level and the *system* level as shown in Fig. 1. Prior research work in the field of optoelectronics simulation has focused primarily at the link or at the functional level. The results [or outputs as shown in Fig. 1(a)] that are desirable from the simulation of an optical link include signal waveforms, eye diagrams, deterministic and random jitter, signal-to-noise



(a) Functional/Link level Modeling

(b) System level Modeling

Fig. 1. Example of an optical interconnect simulation, (a) functional–link simulation methodology and (b) system simulation methodology.

ratios (SNRs), cross talk, and other output parameters. To obtain these results, waveform simulation is performed at a given bit error rate (BER), bit rate (frequency), and input signal power. There are several simulation tools proposed for the link level modeling including the *iFrost* [6], a Matlab-based CAD tool developed by the Optoelectronic Technology Consortium (OETC) and subsequently evolved as the "IBM optical link simulator" [7]; a mixed signal multidomain simulation of optoelectronic interconnect using the *Chatoyant* tool [8]; multilevel simulation using VHDL-AMS in the *SHAMAN* project [9]; and commercial products including OptiBPM from Optiwave Corporation (http://www.optiwave.com) and BeamProp from RSoft Design (http://www.rsoft.com). The above-mentioned tools provide the flexibility to model the complex optoelectronic link from the laser to the photodetector taking into effect mechanical, electrical, and thermal interactions.

However, from the perspective of HPC systems, these tools do not provide quantitative metrics regarding the system level optoelectronics simulation parameters such as latency, bandwidth, throughput, average message distance, power consumption, and signal strength, as shown in Fig. 1(b). For example, the architecture, topology, routing and wavelength allocation (RWA), and traffic distribution can have significant effects on the system parameters such as the average network latency, the offered throughput, the power loss, and the power consumed in the system. Moreover, the above link level simulation methodology is not compatible with computer architecture simulation. Additionally, optical system design tools like OPNET, and OptiSystem are primarily geared towards telecommunication applications and cannot simply be used for HPC applications. Even though the optical link simulators are useful for functional–link modeling of the optical interconnects, they have limited capabilities in simulating system level models for both optical devices and optical architectures–topologies. In the literature, there are several optical system designs [10–12] that have evaluated the performance of optoelectronic architectures for HPC systems. While these groups have focused on the optoelectronic architecture, and performance trade-offs, there has been no documented simulation methodology available for system designers. To the best of our knowledge, such a detailed simulation methodology that could capture both electrical and optical, component, technology, and architecture into an integrated system simulator has not been presented.

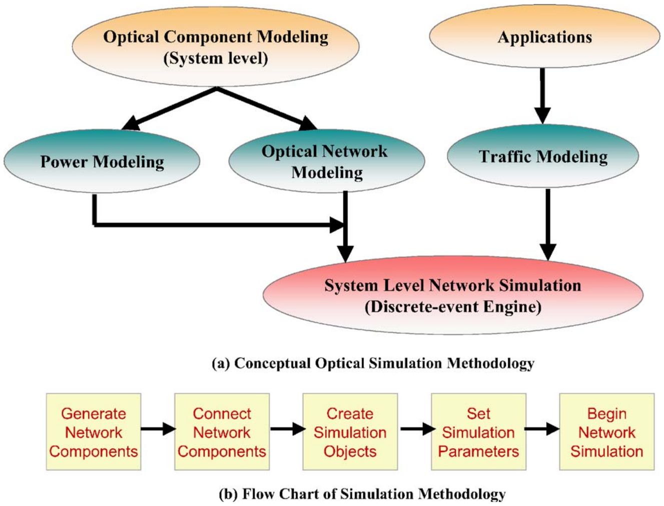
In this paper, we propose a discrete-event, system level modeling and simulation methodology of optical interconnects for HPC systems, called OPTISIM. In OPTISIM, we augment an existing electrical discrete-event simulator by extending its *network component library*, develop an *optical packet simulation methodology*, and *validate the proposed simulation methodology*. In OPTISIM, the optical components–networks are modeled at a level of abstraction more suitable for system level than link level simulation. OPTISIM is also responsive to the traffic patterns, routing, and network architecture. Given that power consumption in interconnection networks is increasing, OPTISIM models different transmitter and receiver designs, thereby providing power models that can be incorporated at the system level. The significant advantages of OPTISIM include (1) *efficient component modeling*: each optical component or device is modeled independently at a level of abstraction that minimizes the computational requirements, while attaining the required system level simulation accuracy and precision; (2) *accurate latency modeling*: transmission, propagation, and receiver delays are accumulated to provide accurate optical packet latency; (3) *optoelectronic modeling*: as future HPC systems will consist of optical components (transmitter, receiver, medium) and electronic components (buffers, switches, queues), our proposed methodology incorporates both technologies in the network design to understand cost-performance trade-offs; (4) *optoelectronic power modeling*: power modeling of optical interconnects evaluates the power consumed in the links the different transmitters and receiver designs and at varying bit rates; (5) *expandability*: active–passive optical components can be easily added to the simulator based on number of inputs, outputs, and expected functionality; and (6) *Extensibility*: the designed optical interconnect simulation framework can be easily integrated with other complex computer architecture system simulators for distributed and parallel computers.

For any given optical interconnect architecture with optical transceivers, wavelength assignment, and traffic patterns, OPTISIM provides end users with network throughput, average latency, power loss, power consumption, and signal strength as the output. In what follows, the system simulation methodology of optical interconnects is explained in detail with a case study.

2. Simulation Methodology of Optical Interconnects

The proposed, conceptual modeling, and simulation framework for optical interconnects is shown in Fig. 2(a).

Parameterized optical passive–active components–devices are modeled as a black box with a set of input and output functions. These modular optical components are recalled from the network library to design the user specified network topology. To this network model, we add optical power consumption models of the link. The traffic model, either from a HPC benchmark or from a synthetic traffic distribution is extracted. Both the modeled network topology and the traffic pattern are embedded into a system simulation engine. This discrete-event simulation engine could be run independently or could be a part of a complete computer architecture simulator. The discrete-event simulator chosen was the YACSIM–NETSIM simulator developed by Rice University [13]. YACSIM provides several simulation objects such as processes, events, semaphores, queues, and barriers—basic utilities required for any discrete-event simulator. NETSIM is an electrical network component and simulation library. YACSIM and NETSIM can be combined to construct a wide range of direct–indirect


Optical Component Modeling (System level)
Applications
Power Modeling
Optical Network Modeling
Traffic Modeling
System Level Network Simulation (Discrete-event Engine)
Generate Network Components
Connect Network Components
Create Simulation Objects
Set Simulation Parameters
Begin Network Simulation

(a) Conceptual Optical Simulation Methodology

(b) Flow Chart of Simulation Methodology

Fig. 2. Simulation methodology. (a) Conceptual optical simulation methodology. (b) Flow chart of optical simulation.

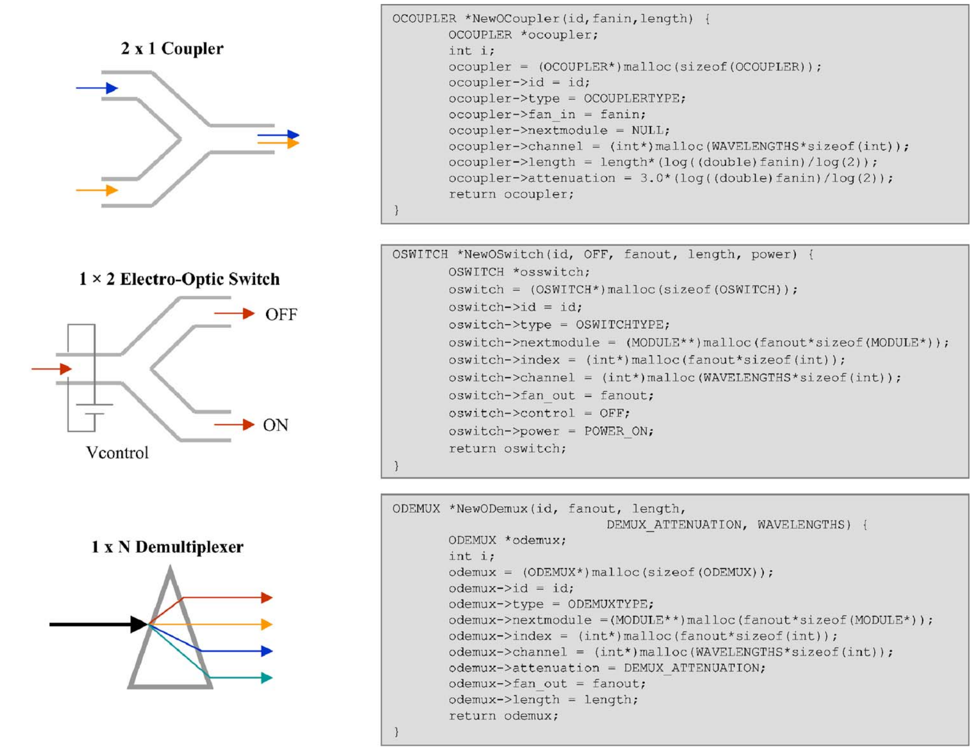
electrical interconnects. Using YACSIM as the simulator engine, we augment the NETSIM library with optical components and optical simulation. We first explain the design of optical components and architecture, and then we explain the power models in our simulator.

2.A. Design of Optical Components and Architecture

From Fig. 2(b), the first step in designing a system level optical interconnect-based simulator is to generate network components. NETSIM includes a library of several electrical components including ports (packets transmitting–receiving units), buffers (packet storage units), electronic routing units, and electronic switching units. The NETSIM library is augmented with several active–passive optical components such as lasers, couplers, splitters, switches, wavelength converters, waveguides, fibers, multiplexers, demultiplexers, and photodetectors. From the link–functional modeling of each of these components, four relevant parameters are extracted for the system level modeling: (1) length—to determine the propagation latency through the component, (2) attenuation—to determine the signal loss due to component, (3) wavelength—to determine the routing within a component, and (4) power—to determine the power consumed by the component. Each optical component is designed with a set of input parameters, $Optical_{component}\langle fanin, fanout, length, attenuation, wavelengths, power\rangle$, where *fanin* provides the number of inputs to the component, *fanout* provides the number of outputs from the component, *length* parameter specifies the length in meters, *attenuation* refers to the signal loss in decibels due to the component, *wavelengths* specifies the number of channels the component can transmit, and *power* calculates the power consumed due to the component. In certain optical components such as wavelength converters, output wavelength will be a function of the received input wavelength. The power consumed is calculated based on the type of optical component specified. This value is added only for active optical components such as transmitters, receivers, and other electro-optic devices.

In OPTISIM, optical components are abstracted by capturing key attributes needed for system level modeling. For example, a transmitter is a single output device, emitting at a given wavelength λ_k with a certain coupling loss between the laser and the coupling device. Therefore a laser can be designed as $Optical_{transmitter}(0,1,0.0\,\mathrm{M}, 1.0\,\mathrm{dB}, k, p)$, where p is the power consumed by the laser and driver circuitry. For example, Fig. 3(a) shows a coupler, an electro-optic switch and a demultiplexer while Fig. 3(b) shows the sample code snippets. Consider a $N\times 1$ coupler, which has the functionality of coupling multiple wavelengths from different inputs N to the single output. A coupler is a passive device, and therefore can transmit most of the wavelengths originating from its inputs. It has a largely fixed attenuation and is approximately dependent on the number of inputs $[3\times\log(N)\mathrm{dB}]$. The length of a coupler is of the order of $2-5\,\mathrm{mm}$. Therefore, the coupler can now be characterized as $Optical_{coupler}[n,\ 1,\ 0.002\,\mathrm{m},\ 3\times\log(N)\mathrm{dB},\ k,\ 0]$. Similarly, a $1\times N$ splitter has the opposite effect, where the same signal is split into n outputs and can be characterized as $Optical_{splitter}[1,\ n,\ 0.002\,\mathrm{m},\ 3\log(n)\mathrm{dB},\ k,\ 0]$. Moreover these splitters can be extended to design optical switches using some additional device parameter (voltage, temperature, current) that can be controlled as shown. From Fig. 3, a 1×2 electro-optic switch is shown in which the switching is performed based on the applied voltage, $V_{control}$. These simple 1×2 can be extended to form large and more complex switch designs. A demultiplexer acts as a $1\times N$ switching device that switches based on the transmitted wavelength. Similarly, we have designed a waveguide, fiber, $N\times N$ arrayed waveguide gratings (AWG), and wavelength converters. Two important features of any component are *nextmodule* and *channel*, both of which will be explained below.

From Fig. 2(b), the next step is to connect the various network components. Modularly designed network components are now connected to each other using the *OpticalNetworkConnect* (src, $dest$, src_{index}, $dest_{index}$). Here, the *src* is the originating component that is connected to the *dest* component. From Fig. 3(b), the *nextmodule* function embedded within the design of the component is used to form this connection. If multiple components have to be connected, then depending on whether the concerned component is the *src* or the *dest*, the src_{index} and $dest_{index}$ is used. For example, consider a demultiplexer, which routes the packet based on the wavelength of the optical signal. Every output is associated with a particular wavelength. The src_{index} is



```
OCOUPLER *NewOCoupler(id,fanin,length) {
        OCOUPLER *ocoupler;
        int i;
        ocoupler = (OCOUPLER*)malloc(sizeof(OCOUPLER));
        ocoupler->id = id;
        ocoupler->type = OCOUPLERTYPE;
        ocoupler->fan_in = fanin;
        ocoupler->nextmodule = NULL;
        ocoupler->channel = (int*)malloc(WAVELENGTHS*sizeof(int));
        ocoupler->length = length*(log((double)fanin)/log(2));
        ocoupler->attenuation = 3.0*(log((double)fanin)/log(2));
        return ocoupler;
}
```

```
OSWITCH *NewOSwitch(id, OFF, fanout, length, power) {
        OSWITCH *osswitch;
        oswitch = (OSWITCH*)malloc(sizeof(OSWITCH));
        oswitch->id = id;
        oswitch->type = OSWITCHTYPE;
        oswitch->nextmodule = (MODULE**)malloc(fanout*sizeof(MODULE*));
        oswitch->index = (int*)malloc(fanout*sizeof(int));
        oswitch->channel = (int*)malloc(WAVELENGTHS*sizeof(int));
        oswitch->fan_out = fanout;
        oswitch->control = OFF;
        oswitch->power = POWER_ON;
        return oswitch;
}
```

```
ODEMUX *NewODemux(id, fanout, length,
                              DEMUX_ATTENUATION, WAVELENGTHS) {
        ODEMUX *odemux;
        int i;
        odemux = (ODEMUX*)malloc(sizeof(ODEMUX));
        odemux->id = id;
        odemux->type = ODEMUXTYPE;
        odemux->nextmodule =(MODULE**)malloc(fanout*sizeof(MODULE*));
        odemux->index = (int*)malloc(fanout*sizeof(int));
        odemux->channel = (int*)malloc(WAVELENGTHS*sizeof(int));
        odemux->attenuation = DEMUX_ATTENUATION;
        odemux->fan_out = fanout;
        odemux->length = length;
        return odemux;
}
```

(a) Design of Optical Components **(b) Example code for various components**

Fig. 3. Modular approach of designing optical components. (a) Example optical components shown include a coupler, a demultiplexer, and a fiber. (b) Example code for generating these components.

used to indicate the correct next module the demultiplexer's output should be connected to. The third step from Fig. 2(b), is to create simulation objects, and the fourth step is to set the simulation parameters, both of which are accomplished by using the YACSIM engine.

2.B. Optical Packet Simulation

Each packet is generated with a unique sequence number in the system. In OPTISIM, an optical packet is simulated by using two events (procedures), one is called the *head* event and the other is called the *tail* event. Every time an optical packet is ready to be injected into the network, the two events are automatically generated. The optical packet is injected into the network with some attributes, such as signal strength of the laser and the wavelength associated with the transmitter port. The head event immediately sets the path from the source to the destination. Consider Fig. 4, which describes the optical packet simulation methodology.

This consists of four tunable transmitters and four fixed receivers. Each transmitter is associated with multiple wavelengths (an array of lasers) so that it can reach any of the receivers. Consider the packet transmission from transmitter (Tx 0) to receiver (Rx 1) on wavelength λ_0 as shown in Fig. 4(a). The head identified as H uses the *nextmodule* function embedded in each component to trace to the next component. The head from Tx 0 traces the route through coupler 1, coupler 2, demultiplexer, waveguide, and receiver. At each component, the head event accrues several attributes of the component, such as length of the component, attenuation due to the component, and routing due to the component. In addition, the head of the packet embeds the packet sequence number into the *channel* specified by the component. When the head of the packet from Tx 0 reaches the coupler 2, the head event first checks, and then embeds the sequence number (1200) of the packet into channel λ_0 associated with the component. Additional features embedded into the functionality of the component are executed when the head reaches the particular component. For example, consider a splitter that splits the input signal into all its outputs. Here, the head event needs to re-create multiple instances of the packets with similar attributes

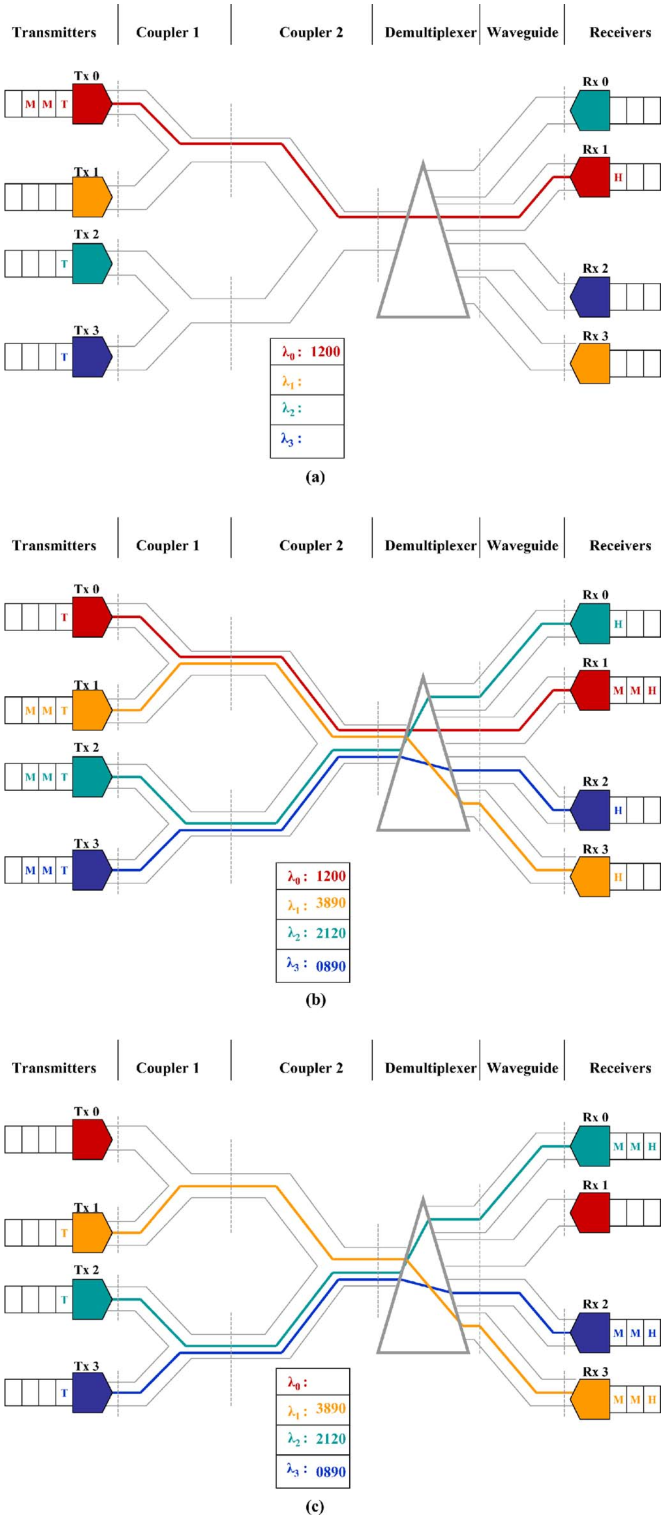


Fig. 4. Simulation example. (a) Tx 0 transmits the packet, the head reaches the Rx 1 embedding the sequence numbers (1200) within each component of the network. (b) Other transmitters Tx 1, Tx 2, and Tx 3 transmit the packets. The mid flits from Tx 0 have now reached the receiver. (c) The tail flit from Tx 0 removes the embedding, and the mid flits from Tx 1, Tx 2, and Tx 3 have reached the receiver.

and restart the simulation for each of the newly generated packets. Once, the head event reaches the receiver port, it is terminated.

After the tail event is created and identified as T, it is immediately delayed for the transmission latency and held in the transmitter port. The transmission latency is obtained by dividing the packet size (in bits) with the bit rate of the transmitter. Figure 4(b) shows the mid flits, identified as *M* (flit is the smallest unit of packet transmission, generally consisting of a several bits) of the packet transmitted by Tx 0 having reached the receiver. In addition, other head events from transmitters Tx 1, Tx 2, and Tx 3 have reached their respective receivers. The tail event then retraces the same path as the head of the optical packet, and further delays for the propagation latency. The tail event checks each component that it traces whether the packet's sequence number exists. If the sequence number exists at the correct wavelength, then the tail erases the sequence number, thereby tears down the path as shown in Fig. 3(c) for Tx 0. This embedding of the sequence number enhances the validity of the proposed model. Moreover, once it reaches the receiver port, it delays for receiver latency in detecting the packet.

2.C. Power Modeling of Optical Interconnects

Power consumption of an optical link is becoming as critical as its speed [11] in HPC system design. In this subsection, we provide an analytical framework to capture power consumption that can be incorporated into the system modeling design through power configuration files. An optical link consists of the transmitter, the receiver, and the channel. Considering a passive channel, the total power consumption of an optical link depends on the transmitter and the receiver power. Transmitter power is consumed at the laser, and laser driver–modulator, whereas the receiver power is consumed at the photodetector, transimpedance amplifier (TIA), and clock and data recovery (CDR) circuitry [11,14]. Multiple-quantum wells (MQWs) [14] with external modulators and vertical-cavity surface-emitting lasers (VCSELs) [14] are suitable candidates for laser sources. MQW needs an external laser source to generate light, where as for VCSEL the light is generated on-chip itself. For the receiver, two designs are incorporated, low-impedance resistive receiver and TIA-based receiver design. Below, we evaluate the power dissipated in an optoelectronic link based on different transmitters and receiver designs. The total power consumed by an entire optoelectronic link is given by

$$\begin{aligned} P_T = P_{\mathrm{TX}} + P_{\mathrm{RX}} = (P_{\mathrm{driver}} + P_{\mathrm{laser}})_{\mathrm{TX}} \\ + (P_{\mathrm{photodiode}} + P_{\mathrm{TIA}} + P_{\mathrm{CDR}})_{\mathrm{RX}}. \end{aligned} \tag{1}$$

The superbuffer in the laser driver is a set of cascaded inverters, and the size of each inverter is larger than the previous one by a constant factor δ. This superbuffer stage will be used for both the MQW- and VCSEL-based designs [14]. The total power dissipated in the driver stages is calculated as

$$P_{\mathrm{driver}} = \gamma C_L V_{dd}^2 B_R, \tag{2}$$

where γ is the switching factor, C_L is the total load capacitance of the superbuffers (of n inverters), V_{dd} is the supply voltage, and B_R is the bit rate. The total capacitance is the sum of input and output capacitance of all the inverters, and is given as [14]

$$C_L = C_{\mathrm{load}} - C_{\mathrm{in}} + \sum_{k=0}^{n-1} (C_{\mathrm{in}} + C_{\mathrm{out}})\delta^k, \tag{3}$$

where C_{load} is the load capacitance of the inverter chain, and C_{in} and C_{out} are the input and output capacitances of the minimum sized inverters.

In MQW-based modulators, light is received from the external mode-locked laser. The modulator performance is characterized by its contrast ratio (CR), insertion loss (IL) at its optimal bias voltage (V_{bias}), and the voltage swing required ΔV_0. The power dissipated in the modulator is given as [15]

$$P_{\mathrm{MQW}} = \frac{P_l}{\eta_{\mathrm{link}}} \frac{q}{h\nu} \left(V_{\mathrm{bias}} \left(1 + \mathrm{IL} - \frac{1 - \mathrm{IL}}{\mathrm{CR}} \right) - \Delta V_0 \mathrm{IL} \right), \tag{4}$$

where P_l is the average optical power required at the receiver input, and η_{link} is the optical system efficiency. For a VCSEL-based system, we adopt a complementary metal-oxide semiconductor (CMOS) driver design from [14], where the driver circuitry consists of two *n*-type metal-oxide semiconductor (NMOS) transistors providing the threshold and modulation currents and a superbuffer driving the gate that delivers the modulation current. The VCSEL power consumed is given as [14]

$$P_{\mathrm{VCSEL}} = I_{\mathrm{total}} V_{\mathrm{source}} = (I_{th} + I_m \gamma)(V_{th} + I_m R_s + V_{dd} - V_{tn}). \tag{5}$$

The total current is the sum of threshold (I_{th}) and modulation currents times the switching factor. The total voltage is the sum of the VCSEL threshold voltage (V_{th}), the voltage drop across the series resistance (R_s), and the minimum source-drain voltage ($V_{dd} - V_{tn}$) to ensure the gate that delivers the modulation current is in saturation.

For the TIA-based receiver design, we determine the power consumed by the photodetector and the TIA. This is modeled similar to [16], which consists of the photodetector as a current source ($I_d + \alpha\beta I_m$) and a common source amplifier connected by a feedback resistance, R_f. I_d is the dark current, α is the VCSEL efficiency in A/W, and β is the detector efficiency in W/A. The input capacitance to the amplifier $C_{\mathrm{in}} = C_D + C_g$, where C_D is the diode capacitance and $C_g = C_{ox} WL$ is the gate capacitance. The VCSEL needs to generate enough light that depends on I_m such that the receiver will produce an output signal of amplitude ΔV_0, which can then be amplified by further receiver stages. This can be approximated as [16]

$$\Delta V_0 = \frac{\gamma I_m}{\beta \alpha R_f}. \tag{6}$$

Therefore, the power consumption of the VCSEL is defined by the needs of the receiver for a given B_R and V_{dd}. The total power dissipated in the TIA-based receiver circuit is then given as

$$P_{\mathrm{TIA}} = I_b V_{dd} + I_d^2 V_{dd} + \gamma (\alpha \beta I_m)^2 R_f, \tag{7}$$

where I_b is the bias current of the internal amplifier and is given by $I_b = \omega_{3\ \mathrm{db}\ int} V_e C_0$ where $\omega_{3\ \mathrm{db}\ int}$ is the 3 db bandwidth of the internal amplifier, V_e is the early voltage, and C_0 is the output capacitance. The gain-bandwidth (GBW) product of the internal amplifier is $\mathrm{GBW} = A(\omega)\omega_{3\ \mathrm{db}\ int} = g_m / C_0$, where $w = 2\pi B_R$, and g_m is the transconductance. The relationship between the internal amplifier bandwidth and the maximum bit rate is given as $\omega = 0.35\omega_{3\ \mathrm{db}\ int}$. The bandwidth of TIA is assumed to be half the bandwidth of the internal amplifier, therefore, the 3 dB bandwidth of TIA is approximated as

$$\omega_{3\ \mathrm{db}\ TIA} = \frac{A(\omega)}{R_f C_{in}} = \frac{w}{0.7}. \tag{8}$$

Then the total power dissipated at the receiver can be obtained as

$$P_{\mathrm{TIA}} = \frac{0.7 A(\omega) I_d^2}{2\pi C_{\mathrm{in}} B_R} + \left(\frac{2\pi V_e C_0 V_{dd}}{0.35} + \frac{2\pi \gamma \Delta V_0^2 C_{\mathrm{in}}}{0.7 A(\omega)} \right) B_R. \tag{9}$$

Then the desired I_m at the transmitter can be obtained by solving Eqs. (5), (7), and (8).

For the low-impedance resistive receiver link design, the total receiver power consumption is given as [16]

$$P_{RC} = V_{dd} I_d + \frac{\gamma V_{dd} \Delta V_0 \pi C_L B_R}{\alpha \beta 0.7}, \tag{10}$$

where C_L is the load capacitance on the RC receiver composed of the photodetector capacitance and the capacitance of the next stage. The power dissipated at the CDR unit is given as [11]

$$P_{\mathrm{CDR}} = \gamma C_{\mathrm{CDR}} V_{dd}^2 B_R, \tag{11}$$

where C_{CDR} is the capacitance of the CDR unit. We have modeled two transmitter designs, VCSEL and MQW, and two receiver designs, resistive and TIA receivers. These transmitters and receivers can be incorporated into the link design to capture the power consumption of the designed optoelectronic link.

3. OPTISIM: Parameter Extraction and System Simulation

To explain the working of the simulator, it is necessary to validate the simulation methodology by comparing our approach to a real machine employing optical interconnects. However, given the difficulties in testing a real machine [17] and the limited scope of this research, we have adopted a different approach of extracting parameters from well-known simulation tools (OptiBPM and Optisystem) from Optiwave Corporation and plugging these into our proposed system simulator.

We designed various optical components–devices using different materials to obtain the desired refractive index contrast, output signal amplitude, and wave propagation using the device level simulator (OptiBPM). These discrete components were then plugged into the link level simulator (OptiSystem) to ensure that the eye opening, BER, receiver power and signal amplitude were sufficient at the specified bit rates and frequency. In addition, we also modeled the power dissipated by the devices and calculated the power consumed by an electro-optic link. This permitted us to test and to some extent validate the proposed simulation methodology. Then the values (power, attenuation, length, and other parameters) obtained from OptiSystem were plugged into our proposed system level simulation (OPTISIM). Then to explain the results (throughput, latency) that can be obtained using the simulator, we show with a case study of an optical interconnect-based system architecture.

The following subsections are explained as follows. In Subsection 3.A the device–component study performed using OptiBPM is explained. Then, in Subsection 3.B the link level simulation for a four-channel (wavelengths) using the parameters extracted from OptiBPM is explained using the OptiSystem tool. In Subsection 3.C, at the link level, we also explain the power consumed by a single channel and how to model MQW and VCSEL lasers. Finally, in Subsection 3.D we show with a case study, how to model an optical interconnect and the results obtained.

3.A. OptiBPM Modeling

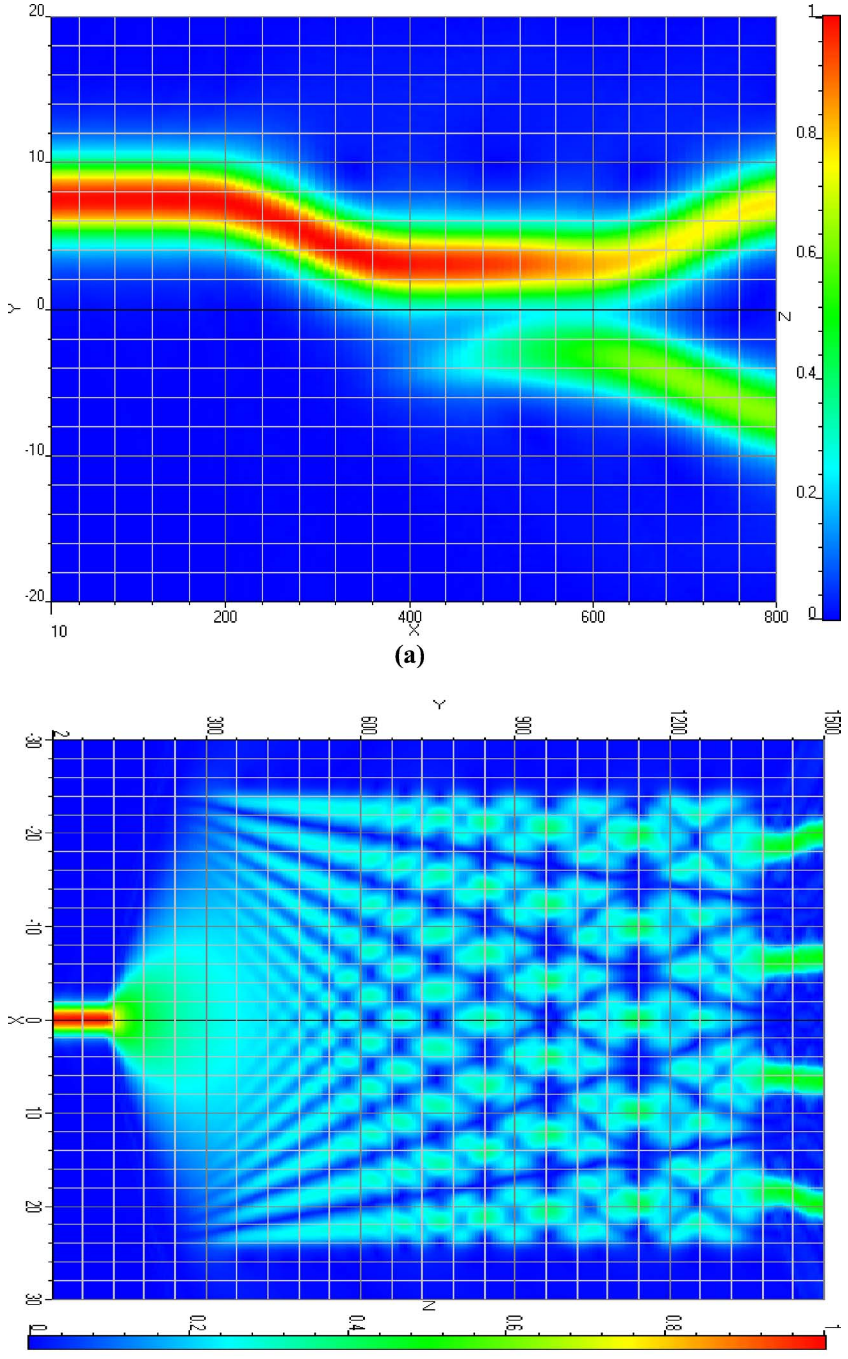
We modeled a 3 dB coupler that was designed using a substrate with cladding refractive index, n_c of 1.442 and core refractive index, n_r of 1.5. The output of this wafer simulation is shown in Fig. 5(a). The device length was 0.8 mm and the output signal had an intensity of 0.45 ($\sim$3.3 dB attenuation). Similarly a 1×4 splitter was also modeled as shown in Fig. 5(b). The length of the splitter was 1.4 mm and the received intensity at the output was measured to be 0.24 implying a 6 dB attenuation. Using the WDM phasar, we evaluated the length of the demultiplexer to be 1.9 mm and an attenuation of 2.1 dB. These parameters were included in the definition files for the OPTISIM simulation library. Currently, we have couplers, splitters, electro-optic switches, wavelength converters, demultiplexers, multiplexers, waveguides, and fibers in our simulator.

3.B. OptiSystem Modeling

The OptiBPM parameters obtained were then simulated using the OptiSystem simulation tool for a four-channel optical interconnect-based architecture. The directly modulated laser for four channels was solved using rate equations at data rate of 2.5 Gbits/s. The lazing channels were 1116.2, 1116.9, 1117.6, and 1118.3 nm; the wavelength spacing being 0.7 nm. The input power to the laser was 2 mW or 0.3 dBm. The losses seen by the signal include propagation loss of 0.2 dB/km, multiplexer loss of 3 dB per stage, and demultiplexer loss of 2.1 dB. Figure 6(a) shows the multiplexed spectrum, the received signal and eye diagram at 2.5 Gbits/s. The eye diagram shows the eye height of 1.39×10^{-5}, threshold of 9×10^{-7}, and a low BER. Figure 6(b) shows the optical interconnect performance for a four-channel system with CW laser and an external Mach–Zehnder modulator at 10 Gbits/s. The eye diagram shows the eye height of 2.21×10^{-5}, threshold of 2.89×10^{-6} and a low BER. This clearly shows that the four-channel system designed using OptiSystem performs within accepted BER and power budget.

3.C. Power Estimations

The parameters for VCSEL and MQW modulators were extracted from [14–16]. Table 1 shows various parameters of the laser driver module, VCSEL, MQWs, and the receiver design parameters.

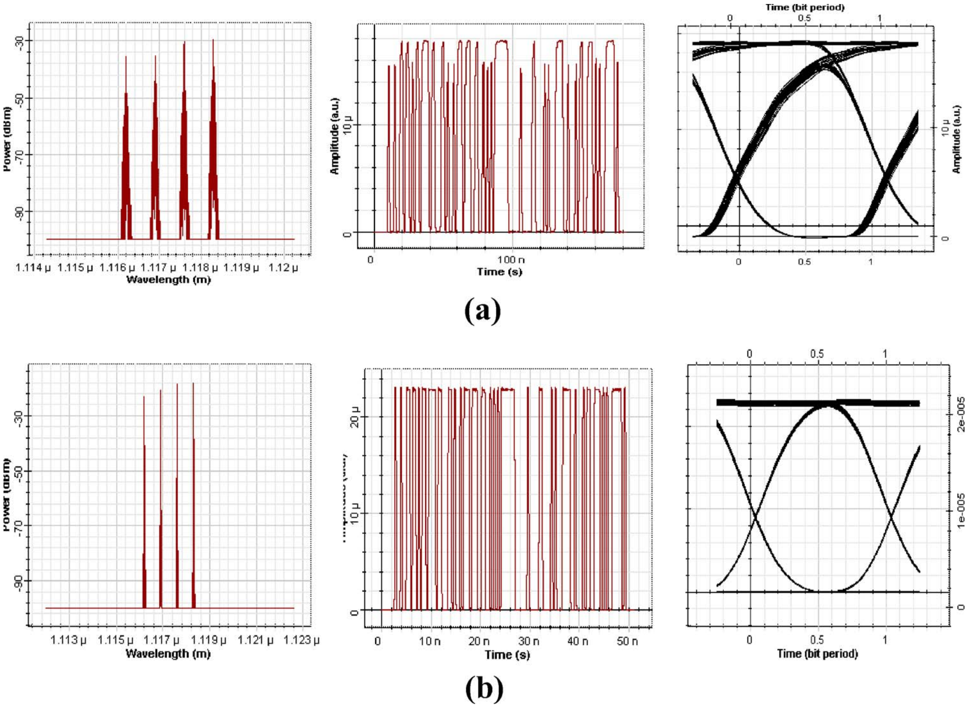


(a)

(b)

Fig. 5. (a) Output from a 3 dB coupler and (b) 1 × 4 splitter.

From the parameters shown in Table 1 and solving equations from Subsection 2.C, we estimated the power dissipated at the transmitter and the receiver at varying bit rates. The link power is dominated by the receiver power consisting of the TIA and CDR whereas the laser and driver dissipate minimal power. Figure 7(a) shows the link power for VCSEL-based configuration with fixed V_{dd}, where the supply voltage is not varied, scaling V_{dd}, where the supply voltage is scaled with the bit rates, transmitter power with scaled V_{dd} and receiver power with scaled V_{dd}. With scaling of bit rates and supply voltages, the power dissipated in a VCSEL is dominated by the receiver consisting of TIA and CDR. The total power dissipated at 10 Gbits/s is approximately 535 mW. With the bit rate scaling from 10 to 5 Gbits/s and the supply voltage scaling from 1.8 to 0.9 V, the power dissipation for a 5 Gbits/s link reduces to almost 108 mW. Figure 7(b) shows the power dissipated at varying bit rates for TIA-



(a)

(b)

Fig. 6. Signal spectrum for a four-channel system at the multiplexer, received data, and the eye diagram for (a) 2.5 Gbits/s using a directly modulated laser source and (b) for 10 Gbits/s using an externally modulated laser source.

based receiver for VCSEL and the MQW modulator. MQWs consume marginally less power in the above graph, as we considered a constant power required at the receiver in the base design. However, further designs will be improved to incorporate distance into the calculations. The receiver power is reduced by considering low-impedance resistive circuits instead of the TIA [16].

3.D. Case Study: RAPID System Architecture

As a case study, we consider reconfigurable all-photonic interconnect for distributed (RAPID) and parallel systems [18,19] system architecture as shown in Fig. 8. A RAPID network is defined as a three-tuple:(C,B,D) where C is the total number of clusters, B is the total number of boards per cluster, and D is the total number of nodes per board. The total number of nodes in RAPID is the multiplicative factor, $N = C\times D\times B$. In Fig. 8(a), 0 up to D-1 nodes are connected together to form a board. Boards, 0 up to B-1, are connected to form a single cluster. All nodes are connected to two subnetworks, a scalable intraboard optical interconnection (IBI) and a scalable remote superhighway (SRS) via passive couplers. We have separated intraboard and interboard (remote) communications from one another to provide a more efficient implementation for both communications. RAPID is designed such that every node has two sets of fixed-array transmitters and fixed receivers for intraboard and interboard communication. Figure 8(b) shows the conceptual diagram of a RAPID network. All interconnections on the board are implemented using optical waveguides and the interconnections from the board to SRS are implemented using optical fiber using multiplexers and demultiplexers. Although the architecture is shown as a ring system, this is only done for the clarity of the illustration. RAPID is actually implemented as a point-to-point topology as explained next in the discussion of routing and wavelength assignment.

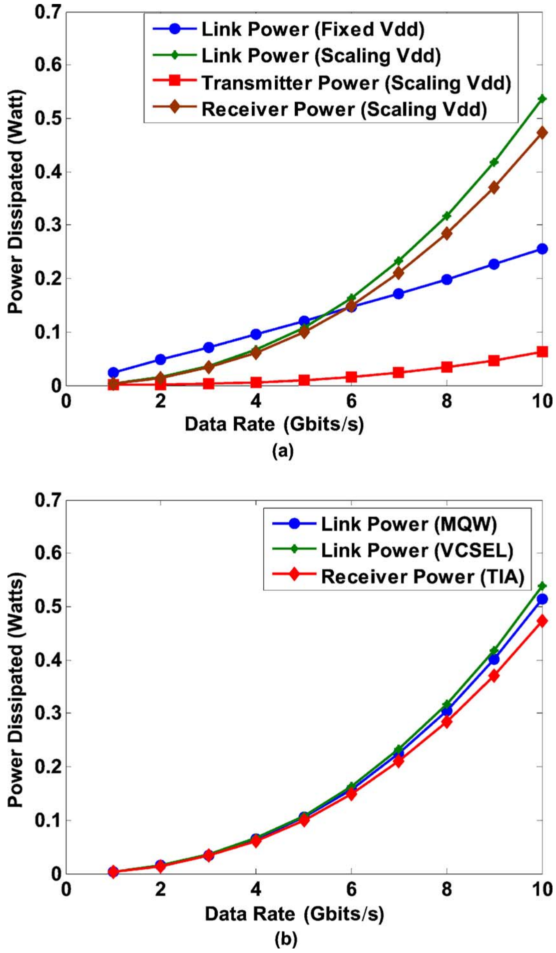
Figure 9(a) shows the RAPID architecture with intercluster connectivity. Consider the intercluster interconnect ($C1$). The original cluster [cluster 0 from Fig. 8(b)] will be replicated to obtain a new cluster. The original SRS will be replicated and named scalable intercluster interconnect (SICI). These clusters will be connected to the SICI using bidirectional AWG as shown in Fig. 9(b). This scaling is achieved by replacing a system board from Fig. 8(b) and using bidirectional AWG to connect the cluster to the intercluster interconnect as shown in Fig. 9(b). This implies that there are no new wavelengths required for designing the intercluster interconnect. The same number of

Table 1. Optical Simulated System Parameters

Laser driver Parameters		
Activity switching parameter	γ	0.5
Driver load capacitance	C_{load}	50 pF
Input capacitance	C_{in}	2 pF
Output capacitance	C_{out}	2 pF
Inverter sizing factor	δ	3
VCSEL Parameters		
Threshold current	I_{th}	0.1 mA
Series resistance	R_s	250 ohm
Threshold voltage	V_{th}	2 V
Optical efficiency	β	0.3 W/A
Threshold voltage	V_{tn}	0.38 V
Multiple-quantum well modulator parameters		
Insertion loss	IL	0.475
Contrast ratio	CR	4.6
Bias voltage	V_{bias}	4.7 V
Link efficiency	η_{link}	0.7
Laser power	P_l	50 μW
Wavelength	λ	850 nm
Receiver design parameters		
Load capacitance for RC design	C_L	0.1 pF
Detector optical efficiency	α	0.4 A/W
Output voltage swing	ΔV_0	100 mV
Dark current	I_d	100 nA
Amplifier gain	A	10
Early voltage of load transistor	V_e	20 V
Output capacitance	C_0	0.05 pF
Photodiode capacitance	C_D	0.05 pF
Device length	L	0.25 μM
Mobility	μ_n	1300 cm^2/V s
CDR capacitance	C_{CDR}	9.26 pF

wavelengths needed to design cluster 0 or cluster 1 is sufficient to ensure complete intercluster communication. With the assumption that there are 16 wavelengths, 16 clusters, 16 boards, and 16 nodes, we can scale the system to $N=16\times 16\times 16=16^3$ $=4096$ nodes [19].

Figure 10 shows the remote wavelength assignment scheme in a $R(1,4,4)$ system, i.e., $C=1$, $D=4$, $B=4$. For remote communication, different wavelengths from various boards are selectively merged to separate channels to provide high connectivity. Remote wavelengths are indicated by $\lambda_i^{(s,c)}$, where i is the wavelength, s is the source board number, and c is the cluster number from which the wavelength originates. To clarify, c is dropped since only single cluster working is explained. The wavelength assigned for a given source board s and destination board d is given by $\lambda_{B-(d-s)}^{(s)}$ if $d > s$ and $\lambda_{(s-d)}^{(s)}$ if $s>d$, where B is the total number of boards in the system, the superscript indicates the source board (in parentheses), and the subscript indicates the wavelength to be transmitted on. For example, if any node on board1 needs to communicate with any node in board 2, the wavelength to be used is $\lambda_3^{(1)}$ and for reverse communication, the wavelength required is $\lambda_1^{(2)}$. To illustrate with an example, consider board 0 transmitter set. All nodes on board 0 have an array of transmitters such that they can transmit on any wavelength $\lambda_i^{(0)}$, $i=0,1,2,3$. Any node in board 0 can communicate with itself on $\lambda_0^{(0)}$, with board 1 on $\lambda_3^{(0)}$, with board 2 on $\lambda_2^{(0)}$ and with board board 3 on $\lambda_3^{(0)}$. The physical fiber channel on which λ_0 is transmitted is called the *home channel* for that particular board (shown as a dotted line for board 0). All signals originating from a particular board are demultiplexed and then selectively multiplexed with different home board channels. For board 0, the multiplexed signal on home channel, $(\lambda_0^{(0)}+\lambda_1^{(1)}+\lambda_2^{(2)}+\lambda_3^{(3)})$ is then demultiplexed at the board 0 receiver. As the receivers are fixed, λ_i, $i=1,2,3$ are received by node i-1.



(a)

(b)

Fig. 7. (a) Power consumption for a VCSEL-based configuration with scaling and fixed V_{dd}. (b) Power consumption for VCSEL- and MQW-based laser sources.

3.D.1. Simulation Methodology

The RAPID network was designed using OPTISIM simulation methodology. Multiple transmitters, fibers, demultiplexers, and receivers along with media access protocol were designed for network simulation. Packets were injected according to the Bernoulli process based on the network load for a given simulation run. The network load is varied from 0.1–0.9 of the network capacity. The network capacity was determined from the expression N_c (packets–node–cycle), which is defined as the maximum sustainable throughput when a network is loaded with uniform random traffic [20]. The simulator was warmed up under load without taking measurements until steady state was reached. Then a sample of injected packets were labeled during a measurement interval. The simulation was allowed to run until all the labeled packets reached their destinations. Cycle accurate simulations were performed to evaluate the performance of various topologies for 16 to 1024 nodes [19]. In addition, two cost-effective alternatives of RAPID were designed, a modified version called M-RAPID and an extended version called E-RAPID that minimized the cost of the interconnect based on the number of transmitters required [19]. The electrical networks chosen for comparison were 2D torus, hypercube, and fat-tree topologies as these topologies are the most common clustering interconnects.

Network workloads that accurately reflect the high temporal and spatial traffic variance of many parallel numerical algorithms usually employed by scientific applications are most useful for evaluating the performance of HPC systems [21–24]. The performance of E-RAPID was compared to other electrical networks for several communication patterns including uniform, butterfly ($a_{n-1},a_{n-2},\dots,a_1,a_0$ communicates with $a_0,a_{n-2},\dots,a_1,a_{n-1}$), complement ($a_{n-1},a_{n-2},\dots,a_1,a_0$ communicates with node $\overline{a_{n-1},a_{n-2},\dots,a_1,a_0}$), and perfect shuffle ($a_{n-1},a_{n-2},\dots,a_1,a_0$ communicates with node

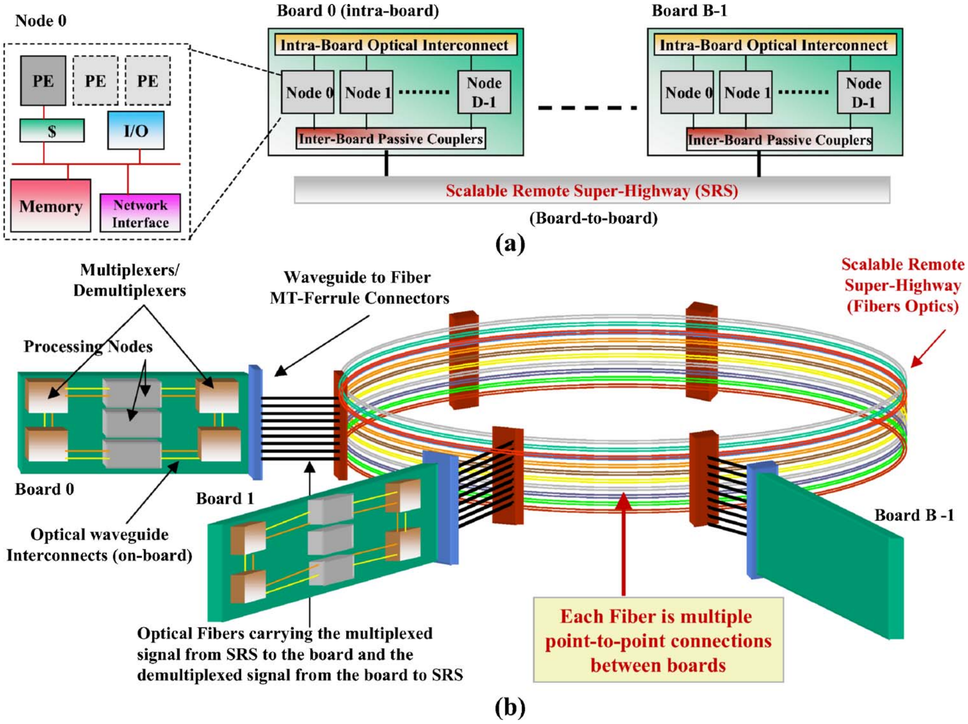

Node 0
PE
PE
PE
$
I/O
Memory
Network Interface
Board 0 (intra-board)
Intra-Board Optical Interconnect
Node 0
Node 1
Node D-1
Inter-Board Passive Couplers
Board B-1
Intra-Board Optical Interconnect
Node 0
Node 1
Node D-1
Inter-Board Passive Couplers
Scalable Remote Super-Highway (SRS)
(Board-to-board)
(a)
Multiplexers/ Demultiplexers
Waveguide to Fiber MT-Ferrule Connectors
Scalable Remote Super-Highway (Fibers Optics)
Processing Nodes
Board 0
Board 1
Board B -1
Optical waveguide Interconnects (on-board)
Optical Fibers carrying the multiplexed signal from SRS to the board and the demultiplexed signal from the board to SRS
Each Fiber is multiple point-to-point connections between boards
(b)

Fig. 8. Architectural overview of RAPID. Every node is connected to two scalable interconnects: an optical intraboard interconnect and a SRS.

$a_{n-2}, a_{n-3}, \ldots, a_0, a_{n-1}$) for a network size of 64 nodes. While traditional HPC applications will employ these traffic patterns in various phases for communication, by separately testing these traffic patterns, it will be possible to identify the best- and worst-case traffic patterns for a given network topology [20,23].

3.D.2. Simulation Results

Figures 11 and 12 show the throughput and latency for a subset of traffic patterns; namely, uniform, matrix transpose, and complement. Uniform is the most common

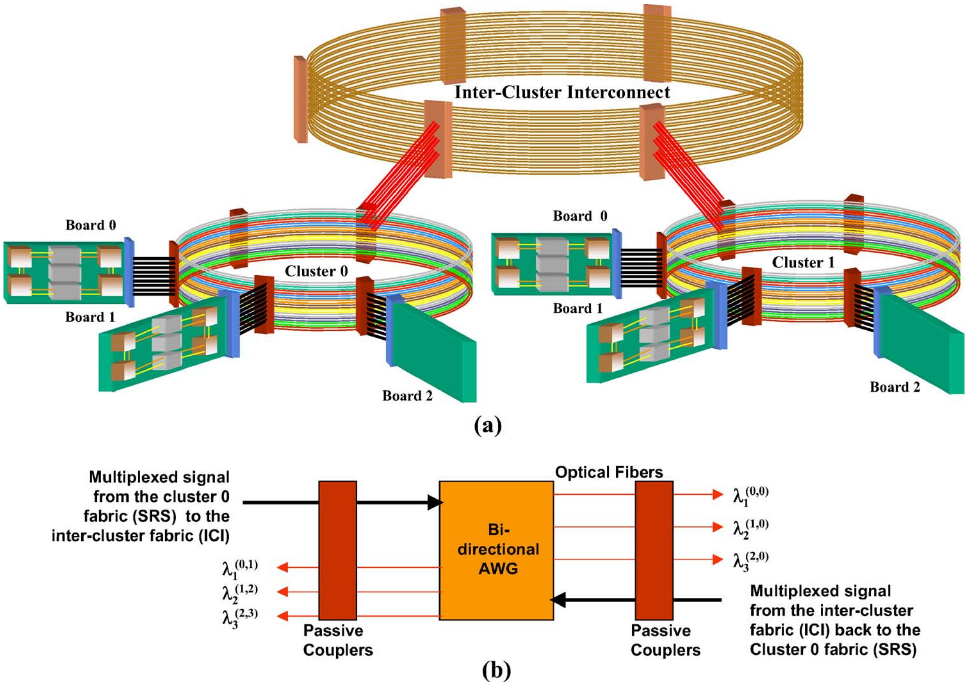


Fig. 9. Scalability of RAPID architecture. (a) Multiple clusters are connected to scalable intercluster interconnect (SICI). (b) Intercluster interconnect implementation using bidirectional AWG.

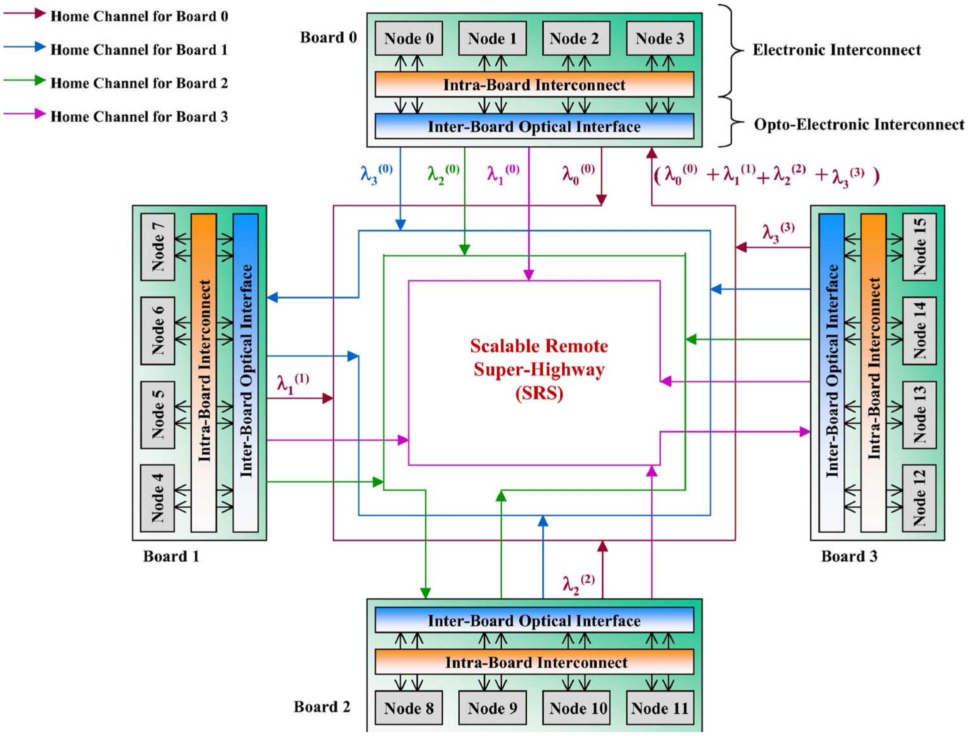


Fig. 10. Routing and wavelength assignment for four nodes/board, four boards, and four wavelength system.

traffic pattern tested on most interconnection networks. In addition, we chose matrix transpose and complement to contrast the best and worst traffic for the RAPID system architecture in this paper.

For 64 nodes uniform traffic, RAPID configurations outperform all electrical networks, with all RAPID configurations showing almost 30% improvement in throughput due to the ample bandwidth provided by optics. From the latency plot for 64 nodes uniform traffic, it can be seen that the latency for RAPID and M-RAPID saturates at 40% of the network load, while E-RAPID shows better performance and saturates at almost 60% of the network load. RAPID configurations show much better performance for matrix transpose traffic patterns with throughput improvement of almost 100%. For complement traffic pattern, electronic networks outperform RAPID configurations. This is due to the design, routing, and wavelength allocation of RAPID architecture where all nodes within a board communicate with a particular destination board on a single wavelength. For example, nodes $0,1,\ldots,7$ on board 0 communicate with nodes $63,62,\ldots,56$ on board 7 using wavelength $\lambda_1^{(0)}$. This results in highly contented access for the same wavelength by all the nodes within the board, leading to low throughput and high average latency. As seen with complement traffic, system level modeling and simulation of optical interconnects is crucial to understanding various performance trade-offs. Routing, wavelength allocation, bit rate, signal power, and topology all play a critical role in performance evaluation of system level optical interconnects.

3.D.3. Simulation Shortcomings

A discrete-event simulation has two major components, *processes* and *events*, which coordinate to provide timing guarantees. Each node has send–receive processes (electrical and optical) and these do not change during the course of the simulation. Packets are events that are created dynamically during the course of the simulation, transmitted, received and destroyed. Components are also created dynamically as required; for example, the coupler used in Fig. 3 would take 96 bytes memory storage for 16 wavelengths. All of the above (processes, events, components) are dynamically created and destroyed when no longer required and are all loaded onto the heap segment. These were run on a Sun-Fire-V440, Solaris 10 OS with 8 Gbytes of RAM. This has also been tested on a Sun-Fire-480R with 4 Gbytes of RAM. Most of the testing has

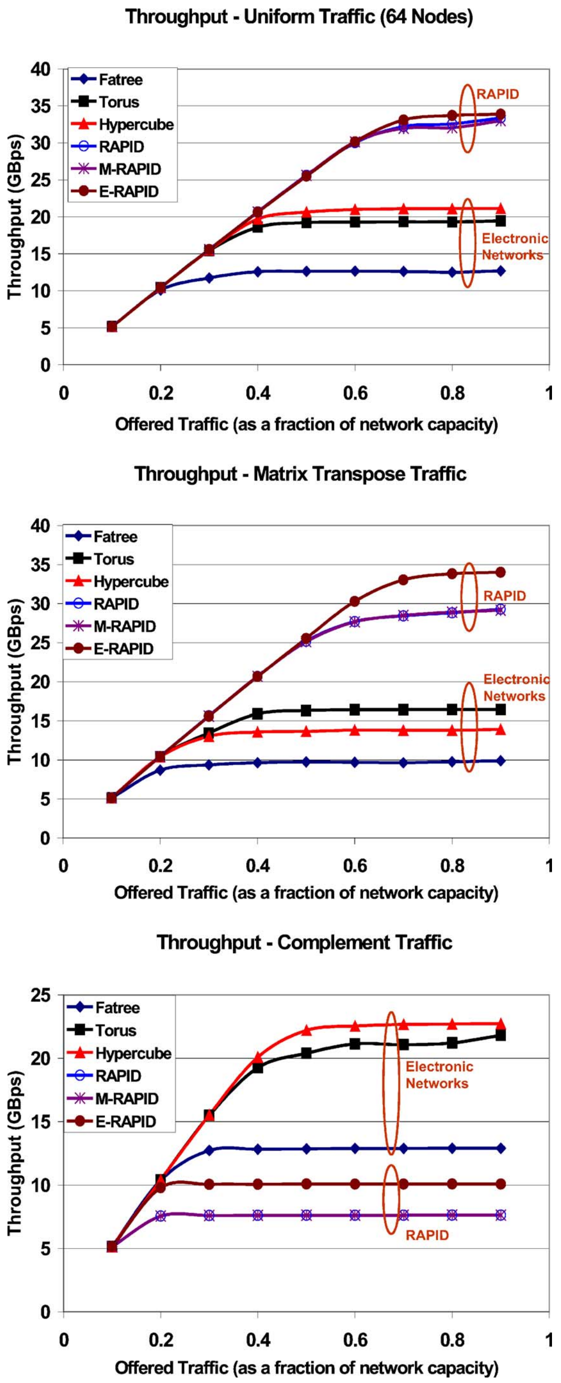


Fig. 11. Throughput estimations for uniform and permutation traffic traces for 64 nodes.

been carried out on a 1024 node RAPID architecture, and there have been no major problems. However, it does takes a significantly long time to complete one run as it depends on the maximum size of the network as well as the maximum load offered. For a load of 0.9 and for 64 node network, it would take approximately >5 min. For a load of 0.9 and for 1024 node network, it would take approximately 30+h. To improve this latency, the simulation could alternatively be run on i386 with Linux OS, which is currently being tested. Scaling beyond 1024 nodes has not been tested, though memory constraints and duration of simulation could also be a limiting factor. Preseeding for running the simulation is not required, as the user can use the default configuration. However, the simulation can be reseeded at runtime, if necessary. The user can issue command line options to rerun the simulation differently under differ-

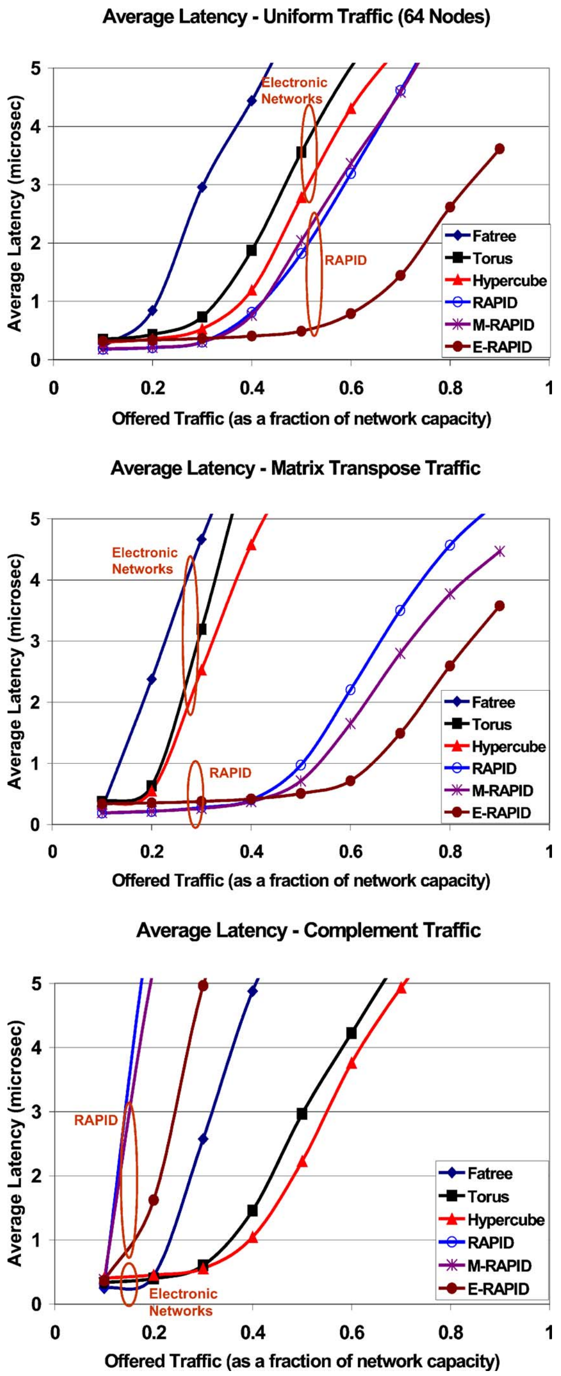


Fig. 12. Average latency estimations for uniform and permutation traffic traces for 64 nodes.

ent load, number of simulation cycles, packet size, and traffic pattern. This can be added to a configuration file that can be read.

4. Conclusion

In this paper, we proposed OPTISIM, a system level optoelectronic modeling and simulation framework. Understanding the design trade-offs (in terms of bandwidth, bit rate, wavelengths, routing, cost, power, and traffic patterns) at the system level is extremely important in the design of optoelectronic HPC systems. In addition, power consumption in optoelectronic networks is becoming as critical as its speed. OPTISIM provides performance modeling along with accurate power models to be used for different transmitters and receivers. Although the framework of NETSIM–YACSIM is used, it has been modified extensively by enhancing the component design space, extending the network design space, and modifying the simulation design space. A discrete event simulation environment combined with component–device modeling provides an attractive avenue for analyzing the power-performance trade-offs in HPC systems. Additionally, the proposed modeling and simulation methodology can easily be integrated with other complete computer architecture tool sets such as the Rice Simulator for ILP Multiprocessors (RSIM) [25] to study architectural design trade-offs.

We are currently in the process of creating a reference manual to simplify the understanding of the simulator. In addition, we are currently testing the RSIM simulator integrated with our OPTISIM to test HPC applications such as fast Fourier transforms, LU, Ocean and other Splash-2 suites. Lastly, we want the simulator to be working on even Linux systems in the near future. Once we have the OS compatibility, architectural platform compatibility (RSIM), and the manual ready, we will disseminate this simulator through the web.

Acknowledgments

This research is supported by National Science Foundation (NSF) grants CCR-0538945 and ECCS-0725765.

References

1. E. Mohammed, A. Alduino, T. Thomas, H. Braunisch, D. Lu, J. Heck, A. Liu, I. Young, B. Barnett, G. Vandenton, and R. Mooney, "Optical interconnect system integration for ultra-short reach applications," Intel Technol. J. **8**, 115–128 (2004).
2. A. F. Benner, M. Ignatowski, J. A. Kash, D. M. Kuchta, and M. B. Ritter, "Exploitation of optical interconnects in future server architectures," IBM J. Res. Dev. **49**, 755–775 (2005).
3. D. A. B. Miller, "Rationale and challenges for optical interconnects to electronic chips," Proc. IEEE **88**, 728–749 (2000).
4. J. H. Collet, D. Litaize, J. V. Campenhut, C. Jesshope, M. Desmulliez, H. Thienpont, J. Goodman, and A. Louri, "Architectural approaches to the role of optics in mono and multiprocessor machines," Appl. Opt. **39**, 671–682 (2000).
5. J. J. Yi and D. J. Lilja, "Simulation of computer architectures: simulators, benchmarks, methodologies, and recommendations," IEEE Trans. Comput. **55**, 268–280 (2006).
6. B. K. Whitlock, J. J. Morikuni, E. Conforti, and S.-M. Kang, "Simulating optical interconnects," IEEE Circuits Syst. Mag. **11**, 12–18 (1995).
7. P. K. Pepelijugoski and D. M. Kuchta, "Design of optical communications data links," IBM J. Res. Dev. **47**, 223–237 (2003).
8. M. Kahrs, S. P. Levitan, D. M. Chiarulli, T. P. Kurzweg, J. A. Martnez, J. Boles, A. J. Davare, E. Jackson, C. Windish, F. Kiamilev, A. Bhaduri, M. Taufik, X. Wang, A. S. Morris, J. Kruchowski, and B. K. Gilbert, "System-level modeling and simulation of 10 g optoelectronic interconnect," J. Lightwave Technol. **21**, 3244–3256 (2003).
9. Z. T. M. Pez, P. Desgreys, Y. Herv, C. Le Brun, J.-C. Mollier, G. Barbary, J.-J. Charlot, S. Constant, A. Destrez, M. Karray, M. Marec, A. Rissons, and S. Snaidero, "Multilevel behavioral simulation of vcsel-based optoelectronic modules," IEEE J. Sel. Top. Quantum Electron. **9**, 949–960 (2003).
10. J.-H. Ha and T. M. Pinkston, "The speed cache coherence for an optical multi-access interconnect architecture," in *Proceedings of the Second International Conference on Massively Parallel Processing Using Optical Interconnections* (IEEE, 1995), pp. 98–107.
11. X. Chen, L.-S. Peh, G.-Y. Wei, Y.-K. Huang, and P. Prucnal, "Exploring the design space of power-aware opto-electronic networked systems," in *11th International Symposium on High-Performance Computer Architecture (HPCA-11)*, (IEEE, 2005), pp. 120–131.
12. O. Liboiron-Ladouceur, B. A. Small, and K. Bergman, "Physical layer scalability of wdm optical packet interconnection networks," J. Lightwave Technol. **24**, 262–270 (2006).
13. J. R. Jump, *YACSIM Reference Manual* (Rice Univ., 1993).
14. O. Kibar, A. Van Blerkom, C. Fan, and S. C. Esener, "Power minimization and technology comparisons for digital free-space optoelectronic interconnections," J. Lightwave Technol. **17**, 546–555 (1999).
15. H. Cho, P. Kapur, and K. C. Saraswat, "Power comparison between high-speed electrical and optical interconnects for interchip communication," J. Lightwave Technol. **22**, 2021–2033 (2004).
16. A. Apsel and A. G. Andreou, "Analysis of short distance optoelectronic link architectures," in *Proceedings of the 2003 International Symposium on Circuits and Systems* (IEEE, 2003), pp. 840–843.
17. A. R. Alameldeen, M. M. K. Martin, C. J. Mauer, K. E. Moore, M. Xu, M. D. Hill, D. A. Wood, and D. J. Sorin, "Simulating a $2m commercial server on a $2k PC," IEEE Computer **36**, 50–57 (2003).
18. A. K. Kodi and A. Louri, "Rapid: reconfigurable and scalable all-photonic interconnect for distributed shared memory multiprocessors," J. Lightwave Technol. **22**, 2101–2110 (2004).

19. A. K. Kodi and A. Louri, “Design of a high-speed optical interconnect for scalable shared memory multiprocessors,” IEEE Micro **25**, 41–49 (2005).
20. W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks* (Morgan Kaufmann, 2004).
21. F. Petrini, E. Frachtenberg, A. Hoisie, and S. Coll, “Performance evaluation of the quadrics interconnection network,” J. Cluster Comput. **6**, 125–142 (2003).
22. A. Singh, W. J. Dally, and B. Towles, “Goal: a load balanced adaptive routing logarithm for torus networks,” in *Proceedings of the 30th Annual International Symposium on Computer Architecture* (IEEE, 2003), pp. 194–205.
23. B. Towles and W. J. Dally, “Worst-case traffic for oblivious routing functions,” in *ACM Symposium on Parallel Algorithms and Architectures (SPAA)* (ACM, 2002), pp. 1–8.
24. Y. Qian, A. Afsahi, N. R. Fredrickson, and R. Zamani, “Performance evaluation of the sun fire link smp clusters,” in *Proceedings of the 18th International Symposium on High Performance Computing Systems and Applications* (IEEE, 2004), pp. 145–156.
25. V. Pai, P. Ranganathan, and S. V. Adve, *RSIM Reference Manual Version 1.0* (Rice Univ., 1997).