UCRL-JC-123883
PREPRINT

High-Performance Parallel Processors Based on Star-Coupled WDM Optical Interconnects

A.J. De Groot
R.J. Deri
R.E. Haigh
F.G. Patterson
S.P. DiJaili

This paper was prepared for submittal to the
Massively Parallel Processing Using Optical Interconnections Conference
Maui, HI
October 27-29, 1996

April 1996

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

High-performance Parallel Processors
based on Star-coupled WDM Optical Interconnects

A. J. De Groot, R. J. Deri, R. E. Haigh, F. G. Patterson, and S. P. DiJaili
Lawrence Livermore National Laboratory
7000 East Avenue, L-174, Livermore, CA 94550
{degroot1,deri1,haighre,fpatterson,dijaili}@llnl.gov

Abstract

As the performance of individual processing elements within parallel processing systems increases, increased capability to communicate information between individual processor and memory elements is required. Since the limited performance of today's electronic interconnects will likely prevent the system from achieving its ultimate performance, there is great interest in using fiber optics to improve interconnect communication. Many groups have considered approaches based on WDM, star-coupled fiber optics for moderate size multiprocessors. Here we propose a fiber optic transceiver approach to such systems that can provide low latency, high bandwidth channels using a robust multimode fiber technology. We use instruction-level simulation to quantify the bandwidth, latency, and concurrency requirements that enable a multiple bus-type optical interconnect based on such transceivers to scale to 256 nodes, each operating at GFLOPS performance. Our key conclusion is that scalable performance, to ≈100 GFLOPS, is achievable for scientific application kernels using a small number of wavelengths (8 to 32), one optical bus receiver per node, and achievable optoelectronic bandwidth and latency requirements.

1. Introduction and Motivation

The difficulty of providing sufficient communication resources between processor and memory elements in parallel, multiprocessor systems has led to many proposals to employ optical interconnects for improved bandwidth and latency [1-4]. These proposals are driven by communication requirements anticipated from significant increases in computing power per node (1 GFLOPS per CPU near term [5]) and system node count, and the recognition that traditional electronic interconnects will have increasing difficulty in meeting these requirements. Enhanced interconnects are required to provide sufficiently rapid access to remote, distributed memory so that available computing power is fully utilized for applications requiring tightly coupled multiprocessing. Cache-coherent, shared memory operation places additional stress on inter-element communications due to the short messages and rapid memory access associated with cache coherence traffic [6]. In addition, rapid remote access can significantly improve memory requirements, and thus system cost, for certain scientific codes (e.g., in which complex, underlying physics is represented by look-up tables), because large quantities of read-only data need not be replicated locally.

The use of wavelength-division-multiplexed (WDM) optical systems (figure 1), in which independent channels on different optical wavelengths are simultaneously broadcast to a large number O(10^2) of nodes over a star coupler [7], is an attractive proposal for multiprocessor interconnects, offering the potential for wide-bandwidth, single-hop communications among all nodes [1-4]. Each wavelength provides an independent, concurrent logical bus channel. With sufficient system wavelengths, it provides a non-blocking crossbar interconnect (output contention only) [7], and can lead to a knockout switch (no output contention) given sufficient receiver resources (e.g., LAMBDANET [7]). While scaling of such systems is ultimately limited by the optical power budget [7] and bandwidth limitations of the optical transceiver technology, use of bridged WDM star couplers as multi-ported routers or spanning busses [8] can enable scaling to higher node count. The large degree/fanout of such routers/busses is attractive for minimizing system diameter and global communication latency.

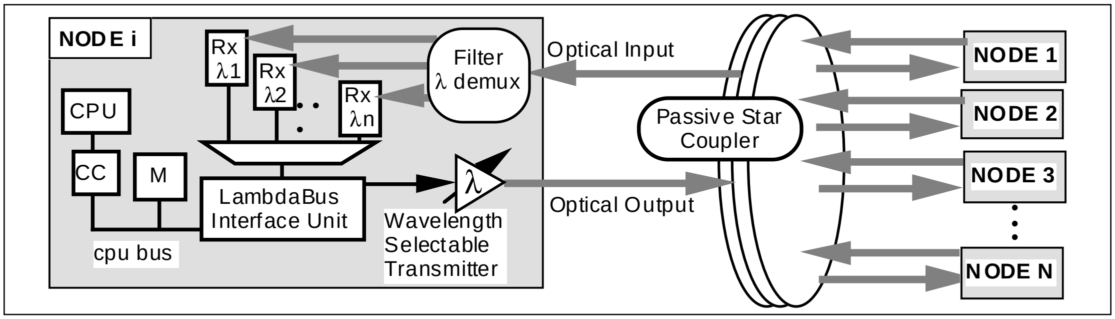

Figure 1 Lambdabus: a high-concurrency, ultrawide bandwidth multiprocessor bus.

Here we focus on the basic WDM star-coupled system, referred to as Lambdabus, rather than larger, massively parallel systems, because its scale conforms to our expectations for the future "sweet spot" of the multiprocessor market and needs for embedded systems on mobile platforms, while it also provides a building block for larger machines. Our concern is with the interconnect hardware requirements to provide robust, scalable performance at the level of 100 sustained GFLOPS and a few hundred nodes.

2. Optical Interconnect Hardware Approach

Optical transmission over single-mode optical fiber (SMF) offers serialized channel transmission rates of 10 GHz (Sonet OC-12) which are likely to increase to >40 GHz over the next several years, and the demonstrated potential for 100-channel WDM systems [9]. Unfortunately, such SMF technology is unsuitable for robust, cost-effective computer interconnects and embedded systems for several reasons.

• Tight SMF optical alignment tolerances (0.2 μm to 2 μm for efficient coupling) increase transceiver cost and shock, vibration, particulate, and temperature sensitivities. • More optical power is required for error-free transmission at higher serial rates, sacrificing connectivity/fanout and reliability by reducing the power budget [10].

• High-speed serialization adds complex and expensive clock recovery and multiplexing between interconnect and logic speeds. Serial data rates ≥2 GByte/s require ≥10:1 muxing to match the 1 GHz logic speeds expected for the next decade [5].

• High serial bitrate is incompatible with MMF dispersion, which limits 8 GByte/s streams to distances <6 m. In certain applications, this constraint restricts the technology's applicability, limiting commercial development and availability.

For these reasons, we pursue a technology based on parallel transmission over multimode fiber (MMF) optic ribbon cables. A similar philosophy governs current work on single-wavelength links by several organizations [11-14], which offer robust optical packaging (MMF tolerances ≈10X looser than SMF) and high channel capacity via the aggregation of multiple (32 demonstrated [11]) fiber bitlines--each running at up to 2.8 GHz [14] while avoiding the difficulties with optical power budget, complexity, and dispersion associated with high-speed serialized links. These links can provide a few GBytes/sec bandwidth with end-to-end latencies of a few nsec (excluding time-of-flight) [15]. The electrical power consumption of this optical transceiver technology is comparable to that of high-performance electronic transceivers [15], and while the cost of the technology is currently high due to its recent commercial introduction, we anticipate significant cost improvements as the technology gains acceptance. The two major issues associated with building upon this technology for a Lambdabus architecture are

1. providing WDM capability
2. the relatively high "costs" associated with the technology.

While these "costs" will likely prove acceptable for a small number of parallel transceiver arrays per each node, they will likely prove prohibitive if many arrays are required at each node -- for example, if a large number of receiver circuits are used, as shown for large n in the "λ_n Rx's" of figure 1. The cost of multiple arrays includes not only raw financial costs, but also those deriving from footprint constraints (about 1 in^2 per array module) and the associated packaging and n:1 multiplexing to access intranode interconnect media.

To avoid a large number of receiver modules per node (as suggested from the above cost rationale), we cannot allocate one receiver array for each system wavelength on every node. Therefore, wavelength selectable transmitter (Tx) and/or receiver (Rx) modules are required. Wavelength-selectable Rx's can be obtained by either (i) fixed 1:n optical wavelength demultiplexing to multiple receivers, followed by electronic selection of the associated WDM channel, or (ii) tunable optical demultiplexing to a single receiver module. The first approach requires many optoelectronic Rx modules and is precluded by the above cost rationale. The second approach is precluded by the slow (several 100's of nsec) tuning times of MMF WDM demultiplexers. We therefore desire a system in which a few fixed wavelengths are received at each node, using fixed demultiplexers and one Rx module per received wavelength. This approach requires rapid wavelength selection of Tx wavelengths to achieve low latency, a capability not available in current versions of MMF array interconnects.

Figure 2 shows our proposed Tx module design, which provides ≈1 nsec wavelength selection, broadcast capability, and large output power using a single module containing two optoelectronic chips. The first chip contains an array of λ laser diodes, each emitting at a different wavelength, with Λ equal to the total number of wavelengths in the system. The second chip contains two arrays of semiconductor optical amplifiers (SOAs) interconnected by a passive star coupler. The lasers emit continuously, and Tx wavelength is selected in the optical domain by using one SOA array (the leftmost in figure 2) to select Tx wavelength. The second SOA array provides modulators to impress word-wide electronic data onto the word-wide spatial channels realized via broadcast over the star coupler. A similar split-and-modulate approach for single-wavelength parallel Tx's has been proposed elsewhere [16]; our module differs in its WDM capability and use of SOAs to provide wavelength-insensitive modulation and high power output.

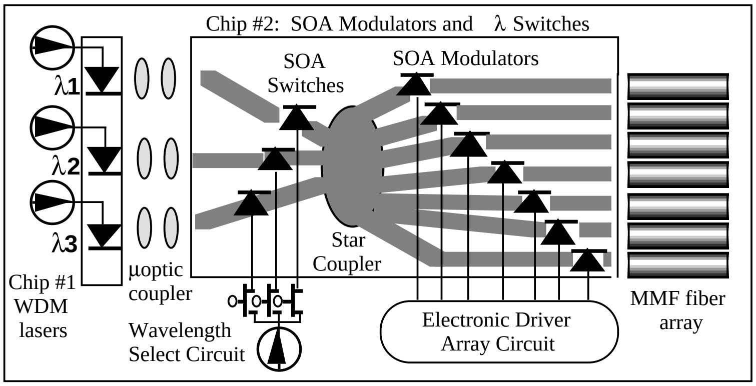
Chip #2: SOA Modulators and λ Switches
SOA Switches
SOA Modulators
λ1
λ2
λ3
Chip #1 WDM lasers
μoptic coupler
Wavelength Select Circuit
Star Coupler
Electronic Driver Array Circuit
MMF fiber array

Fig. 8: Word-wide WDM Tx module. For clarity, only 3 WDM channels and 7-fiber wide arrays are shown.

The integration technologies required to realize each of the two chips have already been demonstrated at several research labs [e.g., 17]. Particular advantages leading to the design of figure 2 are:

- Optical, rather than electronic, wavelength selection with ≈1 nsec SOA gating eliminates on-chip laser thermal transients which cause wavelength drift [18].
- The Tx can broadcast a message on several WDM channels simultaneously.
- Our two chip approach simplifies fabrication (only one active device type per chip), and permits the use of cleaved end facets for laser cavity feedback.
- SOAs improve optical power budget for large fanout and as a hedge against degradation or high-temperature operation.
- All spatial channels (MMFs) are driven with exactly the same wavelengths.

From a link-level perspective, the proposed Tx provides rapid wavelength selection with bandwidth, latency, footprint and power consumption comparable to those of the current, single-wavelength Tx modules described above [12]. The number of wavelength channels Λ is limited by the SOA gain-bandwidth (60-90 nm) and stability constraints on the interchannel spectral spacing. We anticipate that modules with Λ=16 to 64 wavelength should prove feasible. Preliminary, proof-of-principle link demonstrations at 1 Gbit/s per fiber show low bit-error-rates $<10^{-14}$, even in the presence of large mode selective loss [19].

3. Overview of Simulated Lambdabus System

The preceding discussion leads to a Lambdabus configuration in which each node contains a single, wavelength-tunable Tx and a few fixed-wavelength Rxs. The number of system wavelengths Λ is less than the number of nodes N, and each node does not receive all Λ channels. In particular, we assumed the "lowest Rx cost" configuration in which each node receives only one wavelength channel carrying memory access traffic. While increasing the number of memory traffic wavelengths received per node will undoubtedly improve system performance, for example by enabling snoopy or partial snoopy coherence protocols [3,20], this assumption was made to assess the performance of the minimal (low-cost) system using the simulations described below.

These assumptions imply that each distributed portion of main memory is remotely accessed by a unique system wavelength, and that some form of directory-based coherence protocol [5] must be used if cache-coherent, shared memory operation is desired. Since several nodes receive on the same WDM bus, our assumed implementation incorporates multicast ability, which aids cache grouping schemes which minimize the cost of coherence directory memory [6, 21-22]. It also adds contention, however, since messages destined to different nodes may require transmission on the same bus.

Since each of the Λ optical busses shares the fiber cable medium among all nodes, each requires a multi-access control (MAC) protocol to ensure transmission of only one message per bus at any given time. Previously proposed MACs for passive stars are random access (e.g., ALOHA) [1,2] and pre-allocation (time-slotting) MACs [3]. The former reduces capacity under heavy traffic load (37% for slotted ALOHA), while the latter increases latency for light load. We therefore propose to use arbitration, as in [23], and envision a "replicated arbitration" approach in which control information (medium access requests) is broadcast and received by all nodes, using a control bus implemented with a time-slotted MAC and either separate fiber cabling or a separate, out-of-band wavelength (e.g., a 1300 nm control bus wavelength in addition to the Λ memory traffic wavelengths in the 800 nm band). All nodes process control information using identical replicas of the same VLSI arbiter, similar to those in electronic busses [24]. This approach enables the fast MAC associated with "centralized" bus arbitration, while maintaining the fault-tolerance of "distributed" arbitration.

While arbitration adds latency to our interconnect, implementing the control channel with the same aggressive technology as the data channels minimizes delay. We estimated the achievable arbitration latency L_{arb} as

$$L_{arb} = N \bullet I_{cntrl} / B_{cntrl} + TOF + T_{arbiter} \quad , \qquad (1)$$

where N is the number of nodes, $I_{cntrl} = \log_2 \Lambda$ is the control information required from each node, is the control bus bandwidth, TOF is the time of flight across the interconnect (5 nsec/m in glass + 2 nsec optoelectronic delay), and $T_{arbiter}$ is the time to decode and arbitrate the control information, resulting in a control bandwidth latency $N \bullet I_{cntrl} / B_{cntrl}$= 5 nsec for N=256 and Λ=32. Since each bus is arbitrated separately and in parallel, the arbitration time is $T_{arbiter} = 3 \bullet \log_2 N$ gate delays for a logarithmic tree arbiter [24]. Thus, for a 256-node, 32-bus system of 2 meter spatial extent, we estimate a total arbitration latency of 21 nsec.

4. Details of Simulated Lambdabus System

The performance of the Lambdabus system was assessed using "Cerberus" [25], a discrete event simulator for shared memory multiprocessors, in which algorithm execution at the instruction level is simulated in time steps equal to one CPU clock. The SMP maintains cache coherency using a directory-based approach described elsewhere [21]. The remainder of this section details the characteristics of the simulated system elements.

Node processor: Each Cerberus node consists of a RISC processor CPU with an instruction set derived from the *Ridge 32*, a computer manufactured by the now defunct *Ridge Computers, Inc*., which is compatible with a fully pipelined processor timing model and supports a large number (256) of (simulated) outstanding requests. The Cerberus node clock speed was adjusted to 0.4 nsec in order to obtain near-GFLOPS performance for a single node (single Cerberus CPU) on two algorithms (Table I). While our adjusted clock speed is somewhat faster than that expected within a 5 year time frame [5], our approach is justified because we are investigating interconnect performance, and anticipate near-term

GFLOPS node performance due to other improvements (e.g., more superscalar units) or increasing CPU count per node.

Table I: Uniprocessor performance for two algorithms

Algorithm	Performance (GFLOPS)	
	with fast (5 ns) memory	with slow (70 ns) memory
Matrix-vector multiplication	1.05	0.51
9 point stencil	0.31	0.28

Memory hierarchy: Cerberus simulates the flow of data and cache coherence traffic and assumes that all instructions are already cached locally. Data availability is determined by the memory subsystem performance. Each Cerberus node includes a 4-way set-associative data cache [26], with typical size 64 cache lines and 1-cycle access time, which is intended to simulate the combined impact of both first and second level caches. Cache lines were chosen to be either 64B or 128B.

The main memory modules consist of memory, a memory access controller, and a directory for cache coherence information. If valid data is not locally cached, the processor transmits a request over Lambdabus to the appropriate memory module. Upon receipt, the memory controller queries the coherence directory for the status of the requested data. If the main memory contains unshared, valid data, main memory is accessed locally and the retrieved information is then transmitted back to the requesting node over the optical interconnect. If the data is invalid or shared, the memory controller initiates optical messages to other nodes in order to have the data sent to the requester, in a process described more fully below. Our simulations assume a memory controller response time of 20 nsec, including directory access, which is appropriate for directory implementation in a fast SRAM memory technology. Simulations were performed for two different memory access times of 70 and 5 nsec. The longer time is characteristic of today's DRAM access, and the shorter is characteristic of fast SRAM.

Cache coherence protocol: The simulation uses a write-invalidate, write back cache coherence protocol to ensure that all copies of any given datum are consistent with one another [21]. In brief, a full directory-based scheme is employed [5], using valid and dirty bits to track the state of each datum. Each datum has a "home" memory module, which contains its directory information. If a given memory address is shared or dirty, it is invalidated in all other cache locations before any node is permitted to modify the contents of that address.

Optical Interconnect: Based on the preceding discussion, the optical interconnect was simulated using the model outlined below.

- Bus architecture: the interconnect comprises Λ parallel, independent busses. Each bus is asynchronous; nodes transmit transmission requests at any time, unconstrained by time-slotted boundaries.
- Uncompelled, split-transaction bus protocol: After transmission of a request, the bus is not held by the transmitting node while waiting for a reply or acknowledgment, but rather is relinquished for use in other transactions.
- Arbitration latency: After a node requests access to a bus, there is a time delay L_{arb} due to arbitration latency of 20 or 53 nsec. The node can transmit on the bus following the delay only if there is no contention for that bus.
- Arbitration: Contention is resolved on a first-come, first-serve basis. Ungranted requests remain queued, and need not be re-transmitted. Arbiter pipelining is assumed, so that a previously queued message can be transmitted immediately after transmission after the preceding message. While this implies no guard bands, the effect of guard bands during transfer of bus ownership is accounted for in our

model for transmission latency. Simultaneous request arrivals are resolved with a random algorithm based on the simulator cycle number (integer number of CPU cycles).

- Transmission latency: After permission to transmit is granted, there is an additional transmission time delay given by

$$L_{trans} = M / (eff \cdot B_{\Lambda}) + TOF + Tg \quad , \qquad (2)$$

for message size M in bytes, channel bandwidth B_{Λ} in GByte/s, combined optical and optoelectronic time of flight TOF, guard band for bus ownership transfer Tg, and channel efficiency eff <1 to account for message headers and coding. Data reads are performed with message size M equal to one cache line, and coherence messages (read/write requests, invalidates) all assumed to be M=1B. The channel bandwidth B_{Λ} was treated as a variable parameter in the simulations. Since the efficiency and fixed latency TOF+Tg are implementation dependent, we assumed eff=1.00 and TOF+Tg=0 in the simulations. Our results for a given B_{Λ} will be comparable to other systems with no pipelining, different efficiency eff', and nonzero latency TOF' +Tg', if that system has a link bandwidth $B_{link}>0$ of

$$B_{link} = eff^{-1} \cdot \{1/B_{\Lambda} - (TOF' + Tg'')/M\}^{-1}. \qquad (3)$$

This model for B_{link} is conservative because it assumes no pipelining of bus transactions; i.e., multiple messages cannot be simultaneously in flight over the same bus.

- Interleaved Addressing: Each memory address is identified with a unique bus, so that only one bus provides access to a given main memory address or to a given node. Memory addresses are interleaved on the cache line size across memory controllers. Nodes are interleaved across busses.

Algorithms: Cerberus simulates the execution of C codes compiled for execution by the simulator. We simulated four common kernels for scientific applications: a 1024x1024 matrix vector multiplication (mvprod), a 256x256 2-dimensional iterative relaxation code using a 9-point stencil (relax), a 256x256 2-dimensional, complex fast Fourier transform (FFT), and a scatter/gather operation for a representative finite-element crash dynamics problem (an automobile representation in DYNA3D). Performance was evaluated from the reduction in total execution time as additional nodes were added to the interconnect. Improvements were quantified by the "speedup", defined as the ratio of single-node execution time (e.g., Table I) to N-node execution time. Execution rate for the simpler numerical algorithms (mvprod, relax) was also quantified in GFLOPS, with each FLOP corresponding to one single precision operation (add or multiply).

The mvprod algorithm calculates the matrix-vector product y = A•x, by assigning matrix portions to nodes on a row-by-row basis, so that the level of parallelism cannot exceed the dimensionality of the matrix. Most of the code requires only that data be read from memory (the A array), but never shared or over-written. Thus, it stresses only memory bandwidth and not the ability of the system to handle cache coherence traffic.

5. Selected Results

Figure 3 shows simulation results for Lambdabus performance. The figure shows performance in GFLOPS (mvprod, relax) or speedup (FFT, scatter-gather) as a function of the number of ≈1 GFLOPS nodes in the system, for an interconnect with Λ= 8 or 32 busses, each with B_{Λ}= 8 GByte/s bandwidth. Additional simulation parameters are 128 Byte cache line size, 8KB 4-way set-associative cache, and fast arbitration (20 ns) and memory access (5 ns) times.

These results show that an optical bus can support scalable computing to 256 nodes at the 100 to 150 GFLOPS level, using only one receiver per node for reduced optoelectronic interface cost as well as a small number of busses. A speedup of 50-200X was obtained for all algorithms, provided a sufficiently large problem was executed. For exceedingly small problem size, such as scatter gather on a small data set (lowest curve, figure 3), speedup was limited by the intrinsically small parallelism of the problem. Notably, system performance saturates at Λ=8 to 16 optical busses, which is significantly fewer than the number of nodes N.

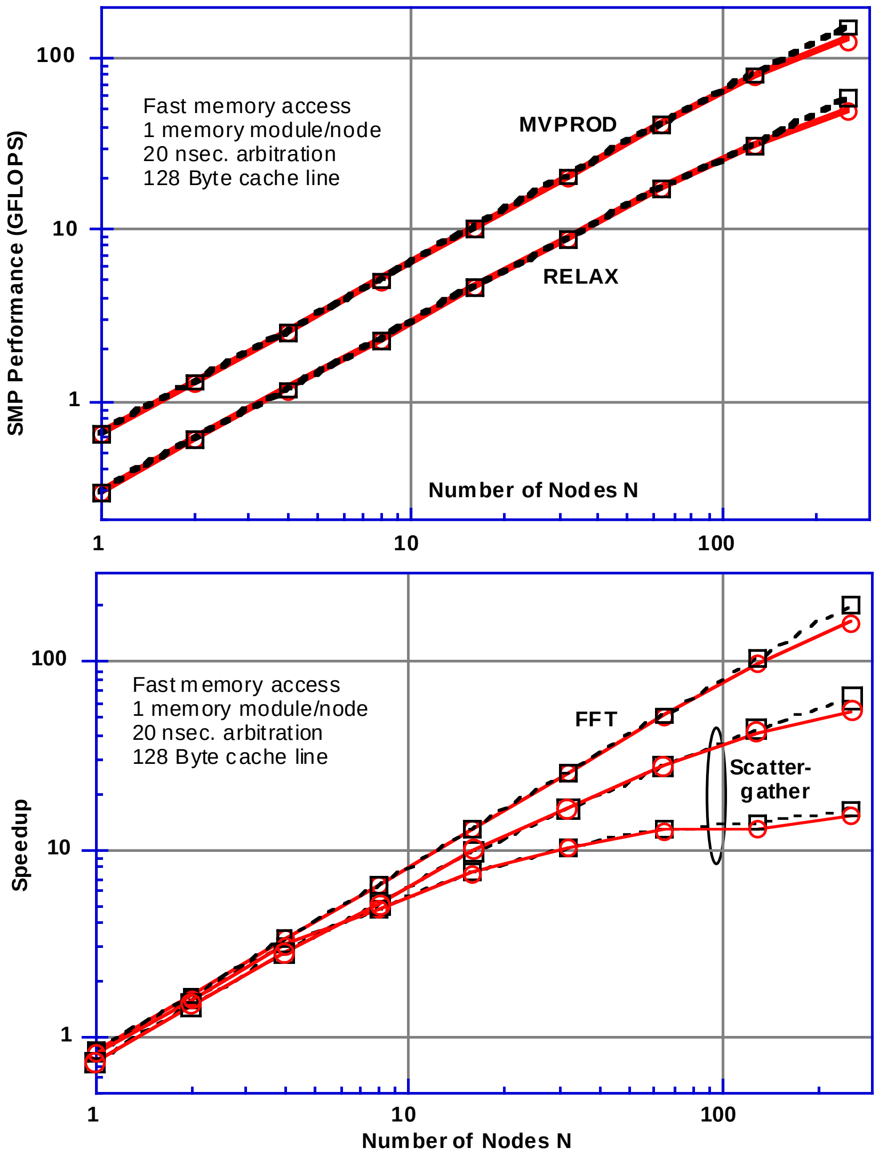

Figure 3: Simulated optical bus system performance for four algorithms using 8 or 32 optical busses (dashed or solid lines), each with B_Λ=8 GByte/s bandwidth.

Performance did not vary dramatically with the details of the memory system within each node. This is shown in figure 4 for the mvprod algorithm, for which ≈100 GFLOPS performance is achieved using either slow memory access (70 ns) or only a single outstanding request allowed per node. Similarly, relatively small (20 to 40% for mvprod) effects were observed due to changes in cache line size (64 verses 128 B) or arbitration latency (20 verses 53 ns).

Scalable performance depends on several factors in addition to the interconnect performance, such as the simulation problem size (discussed above in reference to the scatter-gather algorithm), system size (scalability limited to N), algorithm properties (e.g., computation to communication ratio), and traffic details. To assess interconnect performance under the varying traffic patterns of the different algorithms, we measured the average cache miss latency, or the time delay required to fetch a datum that is not in local cache. This time delay includes all latency associated with network access and communication to fetch the datum, as well as any required invalidations (if the datum is remotely cached) and memory accesses. Figure 5 shows that this latency for the mvprod algorithm is 100 to 300 CPU cycles under light traffic loading (small node count N), which is small enough for adequate performance. The interconnect enables scalable performance provided that the latency remains at this level as node count increases. Beyond a critical system size, the latency increases, marking the transition from an interconnect limited by fixed latencies to one limited by throughput capability. For large N, a simple shared medium model suggests that the latency should scale linearly with N. The simulations show a similar behavior N^X, scaling with an exponent x=1.04 to 1.22. Increasing the number of optical busses Λ improves throughput, and thus increases the system size that can be supported without latency penalty.

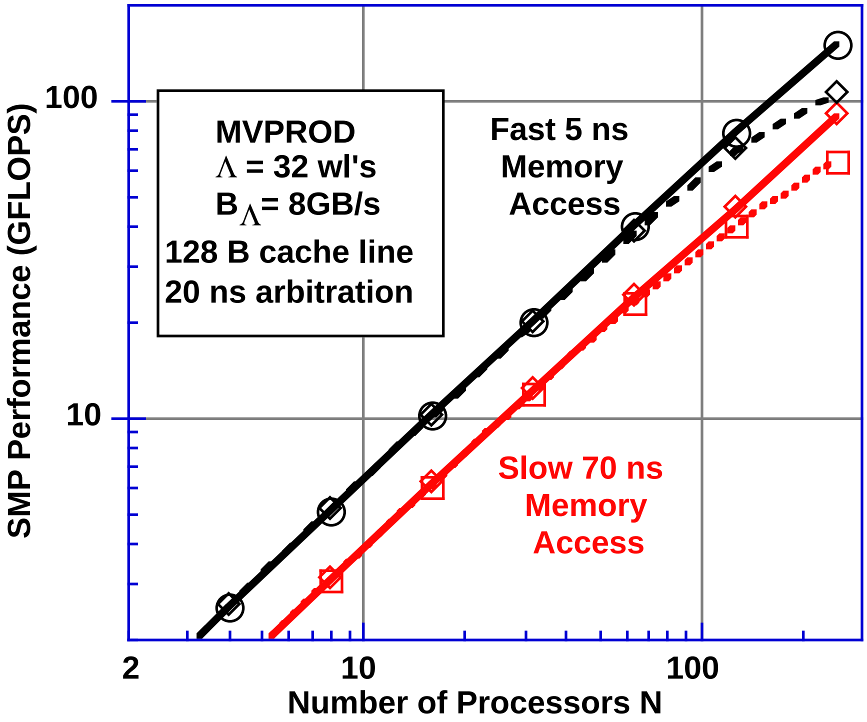

Figure 4: Dependence of MVPROD execution performance on memory system. Solid lines indicate multiple outstanding requests allowed, dashed lines indicate only one outstanding request allowed.

Matrix-vector multiplication performance depends primarily on the speed of remote memory accesses because the kernel involves no coherence traffic. For this reason, the performance for mvprod closely mirrors the behavior of the cache miss latency. Figure 5 shows that performance scales while the miss latency is dominated by fixed system latencies, and that performance saturates when the latency becomes throughput limited. The transition from latency-limited to throughput-limited performance clearly indicates the number of optical busses Λ required to support a given system size. Notably, performance can be improved by reducing memory access time only if the interconnect resources are sufficient to avoid the throughput-limited regime.

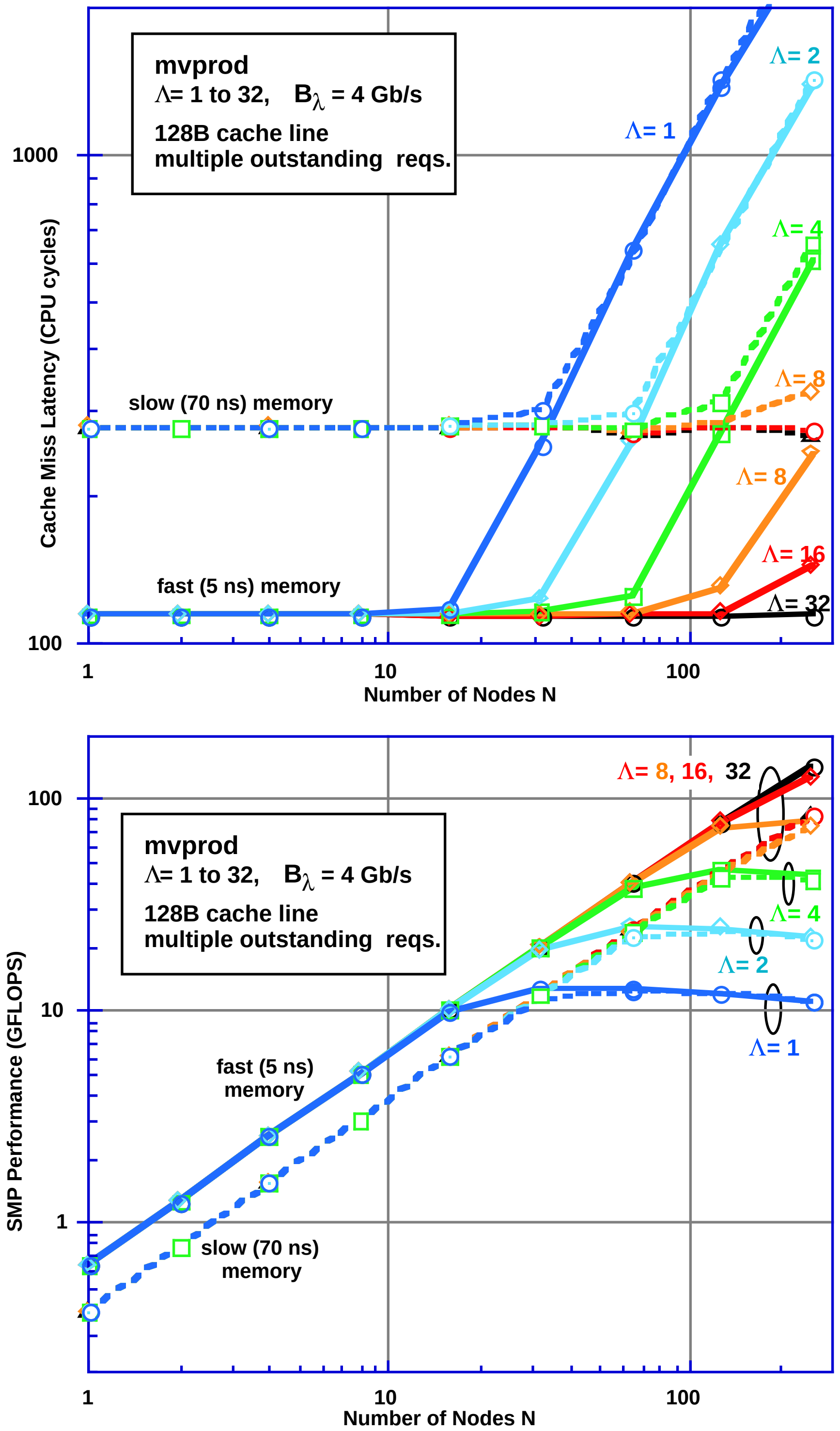

Figure 5: Cache miss latency and performance for mvprod, for the system parameters shown.

The cache miss latency for the relax and FFT algorithms behaves in a qualitatively similar manner to that observed for mvprod. Miss latency remains approximately constant below a critical system size, beyond which it increases in slightly superlinear fashion with node count N. The miss latency for relax and FFT is always larger than that for mvprod in the throughput-limited regime because mvprod communications involve only data reads, with no additional cache coherence traffic (write invalidates). The additional coherence traffic necessary to satisfy read requests increases the cache miss latency in the relax and FFT codes.

The relationship between cache miss latency and system performance is algorithm specific, due to performance dependencies on factors other than communication as indicated above. As an example, the FFT algorithm shows better scaling than mvprod despite longer cache miss latency. This occurs because of a lower cache miss rate for this code, which reduces the overall dependence of speedup on communication. As a result, the onset of throughput-limited communications results in weak saturation of FFT speedup (figure 6), as compared to that of mvprod (figure 5).

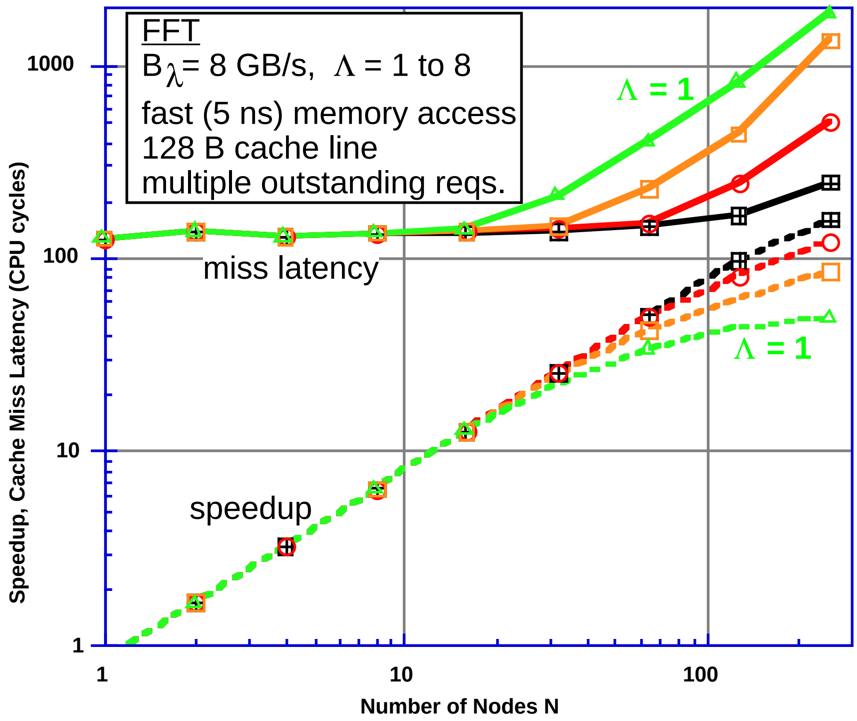

Figure 6: Speedup (dashed) and cache miss latency (solid curves) for FFT execution. Multiple curves show data for different numbers of optical busses Λ=1 (triangles), 2 (boxes), 4 (circles), and 8 (hatched boxes).

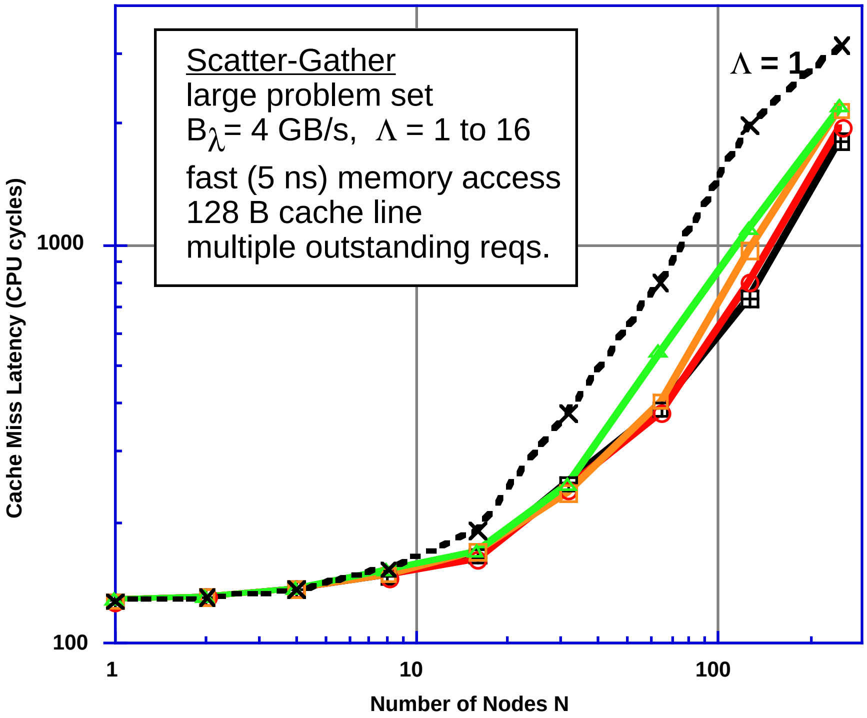

Figure 7: Cache miss latency verses Λ for the scatter/gather code.

The communication behavior for the scatter-gather code behaved rather differently from the other codes simulated (figure 7). Notably, the miss latency shows a markedly weaker dependence on the number of optical busses Λ. This difference arises from the communication pattern for the scatter-gather code, which is quite different from the other codes simulated. The other codes do not compete for write access to the same data because none of their output data are shared, whereas the scatter operation allows cache lines to be written by several processors. The resultant increase in cache coherence traffic increases miss latency (figure 7). Therefore, the miss latency shows a markedly weaker dependence on the number of optical busses Λ for this algorithm.

To optimize cost-performance tradeoffs for the multiple optical bus, it is necessary to quantify the requirements on link bandwidth. Notably, the channel bandwidth B_Λ used in the above discussion differs significantly from the bandwidth of the optoelectronic link B_{link} due to the effect of fixed latencies, as described in eq. (3). For our system, we assume a message size M= 128 B, an efficiency eff=1/1.125, and a fixed latency L= TOF + Tg = 10 ns. For these circumstances, we found a similar behavior for all simulated algorithms. For small node count N, performance is proportional to the aggregate link bandwidth for all optical busses ($\Lambda \cdot B_{link}$), and depends only weakly on the relative number of optical busses or link bandwidth provided their product is fixed. At larger node count, however, the performance saturates at lower levels for smaller channel number Λ. This occurs because the channel bandwidth saturates with increasing link bandwidth, to a value M•eff/L dominated by the fixed latency. Parallel optical busses are required to improve performance in the presence of fixed transmission latencies, whereas increasing link

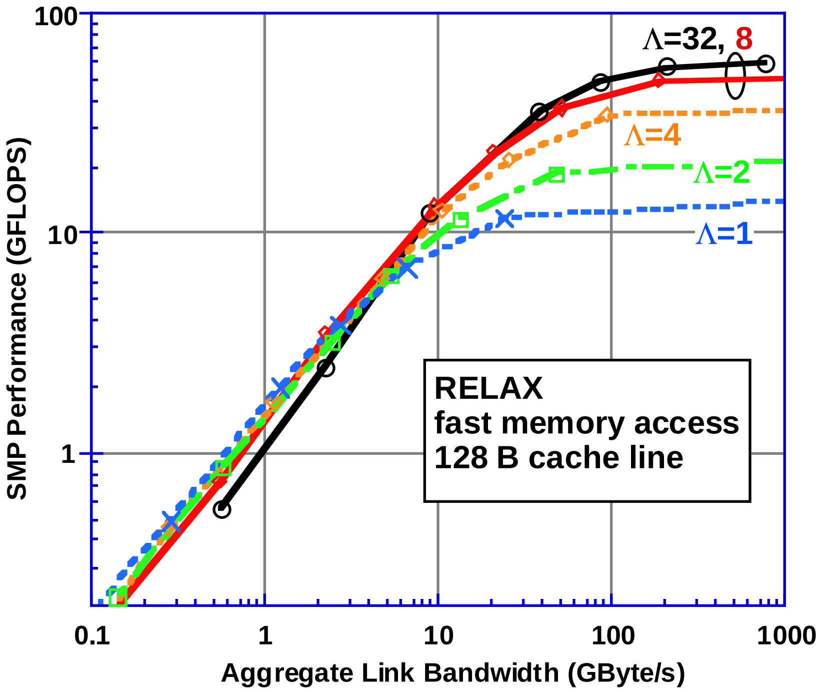

Figure 8: Performance vs. link bandwidth for different numbers of optical busses. The performance saturates for large Λ at a value corresponding to a speedup of 189X, which is limited by the fixed system size (N=256) of the simulations.

bandwidth results in limited improvement. Such behavior is portrayed in figure 8, which shows the maximum performance (optimized by selecting the best N≤256) as a function of aggregate link bandwidth for the relax code. Similar results were obtained for the other algorithms. The figure shows that a link bandwidth of 4 to 8 GByte/s per channel is sufficient to maintain high performance, and that greater bandwidth does not significantly improve performance. It should be stressed that these conclusions depend on the assumption of no transmission pipelining, as discussed in connection with equation 3. Pipelining transactions will improve the latency-limited channel bandwidth, to a value M•eff/Tg, by eliminating time-of-flight effects. Since the assumed latency (TOF +Tg =10 ns) in figure 8 is significantly larger than achievable guard bands Tg of 1-2 ns, we have conservatively evaluated the required link bandwidth.

6. Summary and Conclusions

We have proposed a robust, high-performance transceiver technology for star-coupled, optical interconnects based on WDM transport over multimode fiber ribbon cables, and shown that this approach enables multiprocessor scaling to 256 nodes and ≈100 GFLOPS sustained performance. Because the proposed transceiver's wavelength tuning latency is less than that required for bus arbitration, WDM tuning does not impact system performance. Our results quantify requirements on the optical bus in order to realize such systems. Only a moderate number Λ= 8 to 32 of wavelengths, each supporting a moderate link bandwidth of ≈ 4 to 8 GByte/s, are required. Furthermore, each node needs only a single optical bus receiver operating at a fixed wavelength. These parameters are well within the capabilities of the proposed technology.

Acknowledgments

We are grateful to E.D. Brooks III for useful discussions and for assistance with the Cerberus simulator, and to R.J. Sherwood for assistance in coding and porting sections of the bus simulator. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

References

[1] E. Arthurs et al., *Electron. Lett.* 24, 119 (1988).
[2] P.W. Dowd, *J. Lightwave Technol.* LT-9, 799 (1991).
[3] P.W. Dowd and J. Chu, "Photonic architectures for distributed shared memory multiprocessors", in *Proc. 1st Int. Workshop Mass. Parallel Proc. using Opt. Interconn.*, (IEEE, New York) p. 151 (1994).
[4] K. Ghose, "Performance potentials of an optical fiber bus using wavelength division multiplexing", *Proc. SPIE* 1849, 172-183 (1993).
[5] "National Technology Roadmap for Semiconductors", Semiconductor Industry Association (1994).
[6] D.E. Lenoski and W.-D. Weber, Scalable shared-memory Multiprocessing (Morgan Kaufmann, San Francisco, 1995).
[7] C.A. Brackett, IEEE J. Selec. Area Commun. SAC-8, 948 (1990).
[8] J.R. Goodman and P.J. Woest, "The Wisconsin Multicube", in *Proc. 15th Int. Symp. Comp. Arch.*, p. 422 (1988).
[9] K. Nosu et al., J. Lightwave Technol. 11, 764 (1993).
[10] Power must increase as bitrate-cubed for cost-effective pinFET optoelectronic Rxs.
[11] "Optobus Technical Information", Motorola Logic Integrated Circuits Division, (1994).
[12] Y.M. Wong et al., J. Lightwave Technol. LT-13, 995 (1995).
[13] K.H. Hahn, ARPA Program Review, Parallel/Backplane Sec. (Big Sky, Aug. 1995).
[14] J. Nishikido et al., J. Lightwave Technol. LT-13, 1104 (1995).
[15] R. K. Kostuk, T. J. Kim, et al., "Connection cube and interleaved optical backplane for a multiprocessor data bus", in *Proc. 2d Int. Workshop Mass. Parallel Proc. using Opt. Interconn.*, (IEEE, New York) p. 144 (1995).
[16] J. Bristow et al., *J. Lightwave Technol.* LT-13, 1041 (1995).
[17] T.M. Cockerill et al., *IEEE Photonics Technol. Lett* . 6, 786 (1994).
[18] H. Kobrinski et al., *IEEE J. Selec. Area Commun.* SAC-8, 1190 (1990).
[19] F.G. Patterson, S.P. DiJaili, J. Walker, and R.J. Deri (unpublished).
[20] J.-H. Ha and T.M. Pinkston, "The SPEED cache coherence protocol for an optical multi-access interconnect architecture", in *Proc. 2d Int. Workshop Mass. Parallel Proc. using Opt. Interconn.*, (IEEE, New York) p. 98 (1995).
[21] J.E. Hoag, "The cache group scheme for hardware controlled cache coherence", M.S. Thesis (University of California-Davis,; March 1991), UCRL-LR-106975.
[22] A. Gupta, W.-D. Weber, and T. Mowry, "Reducing memory and traffic requirements for scalable directory-based cache coherence schemes", in *Proc. 1990 Intl. Conf. on Parallel Processing* (vol. I, Penn State Univ. Press, 1990), p. 312.
[23] K.R. Desai and K. Ghose, "An evaluation of communication protocols for star-coupled multidimensional WDM networks for multiprocessors", in *Proc. 2d Int. Workshop Mass. Parallel Proc. using Opt. Interconn.*, (IEEE, New York) p. 42 (1995).
[24] T.N. Mudge, J.P. Hayes, and D.C. Windsor, *Computer Mag.* 40, 42 (June, 1987).
[25] E.D. Brooks III, et al., "The Cerberus Multiprocessor Simulator", in *Parallel Processing for Scientific Computing* (G. Rodrigue ed., SIAM), p. 384 (1989).

[26] J. Handy, The Cache Memory Book (Academic, New York, 1993).

Technical Information Department • Lawrence Livermore National Laboratory
University of California • Livermore, California 94551