Dawning 5000 High-Performance Interconnect Network Design Postprint

Authors: Cao Zheng, Wang Dawei, Liu Xingkui, Liu Xinchun, Shen Hua

Date: 2016-11-02T00:00:00+00:00

Abstract

In petaflop-scale systems, interconnect network design faces new challenges. Continuously improving node performance and expanding system scale represent the primary technical trends in constructing petaflop-scale systems. The ever-increasing node computing capability demands higher performance from interconnect networks, while the continuously growing scale imposes more stringent requirements on interconnect network scalability. Moreover, as system scale increases, the execution time of collective communications continues to grow, constraining application scalability, necessitating further optimization of collective communication performance. To address these issues, this paper investigates interconnect network design methodologies from several perspectives, including interconnect network architecture, network interface controller design, switch architecture design, and collective communication performance optimization, and proposes the design philosophy of the Dawn 5000 high-performance interconnect network. Test and simulation results demonstrate that the Dawn 5000 interconnect network can achieve high performance in unicast, multicast, and barrier synchronization (Barrier) communications.

Full Text

Preamble

Information Technology Letters, Vol. 7 No. 4
Dawning 5000 High-Performance Interconnect Network Design
Zheng Cao, Dawei Wang, Xingkui Liu, Xinchun Liu, Hua Shen

Abstract

In petaflops-scale systems, interconnect network design faces new challenges. The primary technical trends in building petaflops systems are continuously improving node performance and expanding system scale. Increasing node

computing capability demands higher performance from interconnect networks, while growing scale imposes stricter requirements on network scalability. Moreover, as system scale increases, the execution time of collective communications continues to grow, limiting application scalability and necessitating further optimization of collective communication performance. Addressing these issues, this paper investigates interconnect network design methodologies from the perspectives of network architecture, network interface controller design, switch architecture design, and collective communication performance optimization, proposing the design philosophy for the Dawning 5000 high-performance interconnect network. Test and simulation results demonstrate that the Dawning 5000 interconnect network achieves high performance in unicast, multicast, and barrier synchronization communications.

Keywords: Petaflops computing; large-scale interconnect networks; crossbar; switch chip; multi-rail network; collective communication; barrier synchronization communication; multicast communication

1 Introduction

The performance of high-performance parallel computers is increasing at a rate of 1000-fold per decade to meet the growing demands of applications in national defense, automotive manufacturing, aerospace, life sciences, and other fields. Currently, constructing petaflops (Petaflops) computers has become a major challenge. The Dawning 5000 high-efficiency computer system, developed by the National Intelligent Computer Research Center at the Institute of Computing Technology, Chinese Academy of Sciences, aims to address key issues in petaflops computing. The Dawning 5000 high-performance interconnect network is used to achieve high-speed interconnection among Dawning 5000 system nodes, with the goal of systematically studying key technologies for large-scale interconnect networks oriented toward petaflops computer systems and designing an interconnect network with independent intellectual property rights.

In petaflops-scale systems, interconnect network design faces new challenges. Due to limitations in process technology and power consumption, achieving petaflops systems solely by increasing processor frequency is no longer feasible. High-density, high-performance nodes and large-scale networks have become the mainstream choice for petaflops computing. For example, the recently petaflops-capable IBM Roadrunner comprises 3,456 nodes, with each node providing 449.6 GFlops of computing power. This trend presents higher challenges for interconnect networks in four aspects: latency/bandwidth, scalability, reliability and manageability, and collective communication performance.

First is the challenge in latency and bandwidth. The use of high-density, high-performance nodes exacerbates the imbalance between computing capability and network I/O capacity. As shown in [Figure 1: see original paper], over the two-year period from 2006 to 2008, single-node performance increased nearly 40-fold, while network I/O capability only doubled. This enormous disparity

imposes higher demands on network latency and bandwidth performance.

Second is the scalability challenge. Scalability requirements manifest at two levels: inter-node scalability and intra-node scalability. Inter-node scalability must accommodate increasing numbers of nodes, enabling interconnect network designs to be used for larger-scale system interconnection with only minor modifications and without significant performance degradation. Intra-node scalability must accommodate increasing numbers of processors, allowing network interface controller designs to provide network interfaces for more processors with only minor modifications while ensuring each processor obtains sufficient network performance.

Third is the challenge of reliability and manageability. Taking a fat-tree network with over ten thousand nodes as an example, there are nearly 60,000 network ports and approximately 3,500 cables. Equipment failures and link failures occur more frequently, requiring network designs that can tolerate transient link errors and promptly obtain network status information to identify fault points.

Finally, there is the collective communication performance challenge. Research has found that in large-scale scientific computing and applications, collective communication overhead often accounts for 80% of total message overhead. Particularly for global communications in collective operations, their execution time grows rapidly with system scale. We are especially concerned with the communication performance of barrier synchronization and multicast.

As shown in Figure 2: see original paper, the barrier synchronization operation is a global synchronization operation where all processes must reach this synchronization point before continuing with subsequent actions. The system remains blocked until the last process arrives, so barrier synchronization performance directly impacts system execution performance. Barrier synchronization operations have even become performance bottlenecks for some applications. Analysis of a proprietary meteorological application on the Cray T3E revealed that at 128-node scale, a barrier synchronization operation occurs on average every 200 s. If each barrier synchronization operation increases by 15 s, total system execution time increases by 7%. If implemented based on unicast, barrier synchronization latency grows approximately linearly with the number of processes, necessitating further optimization of barrier synchronization performance in large-scale networks.

Multicast communication is the process where a process replicates data multiple times and distributes it to other processes, as shown in Figure 2: see original paper. It is widely used in parallel graph algorithms, parallel search, and basic operations in high-performance computing such as matrix multiplication and LU decomposition. More critically, multicast is a subprocess of other collective operations (such as Gather and Allreduce operations), so optimizing multicast performance indirectly optimizes the performance of these collective communications. If multicast is implemented based on unicast, multicast latency increases rapidly with the number of destination nodes and multicast packet length, re-

quiring further optimization of multicast performance in large-scale systems.

Faced with these four major challenges, exploring Dawning 5000 interconnect network design methods and studying key technologies for building large-scale interconnect networks are the primary objectives of this paper. This paper first introduces several representative high-performance interconnect networks and collective communication performance optimization technologies both domestically and internationally. Based on analysis of their key technologies, it presents the detailed design of the Dawning 5000 interconnect network.

2 Background and Related Research

Among cluster-dedicated interconnect networks, IBM' s IBM SP2 network [1], Myricom's Myrinet network [2], Quadrics'QsNet network [3], and Infiniband networks are most representative. This section introduces their design philosophies and implementation technologies. However, collective communication is not universally supported in these interconnect networks. Therefore, this section also provides detailed introduction to collective communication performance optimization methods in interconnect networks, exploring current optimal collective communication optimization approaches.

2.1 IBM SP2 Network

IBM RS/6000SP is a Deep Blue series large-scale server launched by IBM, featuring parallel computing, single-point console control, centralized management, and high-speed interconnection among nodes, making it widely applicable in various critical business application systems requiring server clusters. The key component of the IBM SP system is the switching network that interconnects all nodes. Its main design objectives are high bandwidth, low latency, scalability, modularity, and convenient connection to processing nodes [1]. The first-generation IBM SP network (Vulcan) has the following characteristics: it employs a Bi-directional Multistage Interconnection Network (BMIN) topology with good scalability (equivalent to fat-tree topology); uses buffered wormhole switching and source routing; implements relative credit-based flow control, where the buffer notifies the counterpart via flow control signal lines whenever space for one flow control unit is generated; operates as a synchronous network driven by a global synchronous clock; connects to node processors via I/O buses, with network adapters containing i860 communication processors and bidirectional DMA engines to support efficient communication protocols; and features a crossbar-based 8-port switch chip with central buffered architecture, where each queue corresponds to one output port (a type of output-queued structure), employing a Least Recently Served packet scheduling strategy within the switch chip to improve data transmission fairness.

Although synchronous networks can simplify synchronization processing between transmitting and receiving ports for data transmission on links, clock signal distribution issues limit the operating frequency of interconnect networks,

thereby affecting data transmission bandwidth. Clock distribution also limits network scalability. Therefore, the third-generation network Switch3 [4] improved upon this by adopting asynchronous network mode. Switch3 also added adaptive routing support to achieve lower latency and hardware-accelerated multicast implementing tree-based multicast algorithms.

The switch chip structure used in Switch3 is similar to Vulcan, but internally provides two queues with different priorities for each output port, supporting priority transmission to some extent. However, its output-queued structure inherently results in poor switch chip scalability and inability to operate at high frequencies.

The SP system' s interconnect network possesses certain scalability and demonstrates good performance in both high bandwidth and low latency, becoming mainstream for high-performance interconnect networks starting in 1998 and accounting for nearly half of TOP500 systems around 2000. However, since the emergence of higher-performance Myrinet networks, the number of SP systems has gradually declined, with only 10 systems using it in the November 2008 TOP500.

2.2 Myrinet

Myrinet [2] originated from two research projects in the United States: Mosaic (a low-granularity multicomputer experimental system) at the California Institute of Technology (CALTEC) and Atomic LAN at the University of Southern California Information Sciences Institute (USC/ISI). The network built using Mosaic components led researchers from these projects to establish Myricom and develop the Myrinet network.

The design goal of Myrinet networks was to achieve system area network performance in local area network environments. The first release version adopted full-duplex bidirectional 1.28 Gbps high-bandwidth links. Simultaneously, Myrinet considered the application environment and performance requirements within parallel processing systems, making the following technical choices: it employs a non-blocking Clos network topology to reduce packet conflicts in the network, improve network throughput, and further reduce network latency; uses wormhole switching and source routing; implements Xon/Xoff flow control on each transmission path to ensure reliable data delivery; simplifies link control protocols considering the application environment of internal interconnect networks in parallel systems with short transmission distances and low error rates, reducing protocol overhead during data transmission and thereby lowering transmission latency; features a crossbar-based switch chip microarchitecture; and includes a RISC processor in the network interface controller to execute low-level communication protocols, implementing user-level communication protocols to further reduce software overhead. The network interface controller contains two types of DMA controllers: one called the DMA engine responsible for data exchange between the host and network interface controller memory, and the other

called packet DMA responsible for data exchange between the network and network interface controller. The two DMAs can operate simultaneously, enabling pipelined packet transmission and reception, thereby improving communication system bandwidth and eliminating unnecessary waiting delays.

Compared to SP networks, Myrinet achieves higher network performance by simplifying software and link layer protocols. The latest Myrinet release adopts 10 Gigabit links with a minimum MPI latency of 2.2 s. Myrinet networks are widely used in large-scale cluster systems due to their high bandwidth, low latency, and good scalability, becoming mainstream for dedicated interconnect networks from 2001 to 2005, with Dawning 4000A also adopting Myrinet networks.

2.3 Quadrics

The Quadrics network [3] originated from the high-performance interconnect network of the Meiko CS-2 [5] MPP system. Quadrics network design primarily targeted the internal application environment of parallel processing systems, aiming from the outset to provide efficient communication among nodes in high-end high-performance computer systems. The main features of QsNet (Quadrics Network) can be summarized as follows: it employs a fat-tree topology to provide high bisection bandwidth and good scalability, as shown in [Figure 3: see original paper] for an FT(4,3) network where 4 is the switch port count and 3 is the number of switch layers; uses wormhole switching with primarily source routing while also providing adaptive routing support based on fat-tree topology; implements end-to-end reliability protocols at the link layer with automatic data retransmission to ensure reliable data link transmission; provides hardware support for multicast and barrier synchronization operations; features an input-queued buffered structure in the switch chip microarchitecture that reduces head-of-line (HOL) blocking through increased virtual channels; and the Elan network interface controller provides a programmable processor that integrates local virtual memory into global shared virtual memory, with network routing calculated based on global virtual addresses. It implements a memory management unit that maintains synchronization updates with the host' s memory management unit, achieving true user-level communication (processes use virtual addresses to initiate direct memory access) and significantly reducing software overhead.

QsNet design exhibits high complexity, and its high bandwidth, low latency, and support for shared-memory communication make it an ideal choice for constructing large-scale distributed shared memory (DSM) systems. After 1998, Quadrics collaborated with Compaq to develop the world' s fastest parallel computer systems. Their first product was AlphaServer SC, and the 2002 ASCI Q system used Quadrics networks to connect 1024 4-CPU AlphaServer ES45 SMP servers, achieving a peak computational speed of 10.24 trillion operations per second. The latest third-generation QsNetIII [6] adopts 10 Gigabit links with a minimum MPI latency of 2 s.

The number of QsNet systems in TOP500 has always been small. After Infiniband gradually matured, QsNet' s share in TOP500 decreased further, with only 8 systems using QsNet in the November 2008 TOP500. In fact, QsNet can be considered a precursor to Infiniband, with most features identical to those defined in the Infiniband specification.

2.4 Infiniband

Infiniband [7] is a powerful architecture designed to support Internet infrastructure I/O interconnects, supported by top companies in the industry whose executive committee members include Compaq, Dell, HP, IBM, Intel, Microsoft, and Sun, with over 220 members in the Infiniband Trade Association. Infiniband is the only standard that provides both intra-chassis backplane interconnect solutions and inter-chassis high-bandwidth interconnects, unifying I/O and system area networks.

Infiniband networks are designed to support cluster applications, storage area networks, and inter-processor communication, achieving high bandwidth while also providing quality of service (QoS) and reliability, availability, and serviceability (RAS) performance. Infiniband' s complete communication protocols and underlying network implementation technologies draw on research experience from Ethernet LANs, Fibre Channel storage networks, and wide area networks, giving it strong versatility. The Infiniband architecture defines multiple devices for system communication: Channel Adapters (CA), switches, routers, and Subnet Managers (SM).

Among various Infiniband specification-based products, Mellanox' s products are most representative. Their main features related to high-performance computing environments include: fat-tree as the typical topology with high bisection bandwidth and good scalability; virtual cut-through switching with source routing and table-based routing, plus adaptive routing support; absolute credit-based flow control; an in-band management network for RAS design; link layer protocols where each link can support multiple transport services and multiple priority virtual channels, thereby providing quality of service guarantees and solving reliability issues for critical links through multiple redundant paths between nodes. The latest InfiniScaleIV supports Error Checking and Correction (ECC) during transmission, further ensuring transmission reliability.

Infiniband fully leveraged Virtual Interface Architecture (VIA) [8] during development, using work queues to offload communication volume control functions from upper-layer applications. These work queues are initialized by upper-layer applications and then handed over to Infiniband management. For each inter-device communication channel, a Work Queue Pair (WQP) is allocated at both ends for sending and receiving. Upper-layer applications place a transaction record in the work queue, which the channel adapter then sends to remote devices. When remote devices respond, the channel adapter returns status to upper-layer applications via completion queues or events.

Upper-layer applications can set up multiple work queue pairs, with channel adapter hardware processing each communication request and generating a Completion Queue Entry (CQE), providing status for each work queue pair in an appropriate priority order, allowing upper-layer applications to continue other processing activities while transactions are being handled. Simultaneously, upper-layer applications use virtual addresses to initiate communication requests, with virtual-to-physical address translation performed by channel adapter hardware, truly achieving user-level communication. Infiniband' s hardware-software interaction method significantly reduces software overhead, providing guarantees for its low-latency communication. The latest InfiniScaleIV uses 40 Gbit/s links with a minimum MPI latency of less than 1 s.

Due to its excellent applicability and high performance, Infiniband has gained increasingly widespread use, especially in high-end high-performance computing where its share continues to grow. In the November 2008 TOP10, nearly half adopted Infiniband, including IBM Roadrunner, which first achieved petaflops performance.

2.5 Collective Communication Optimization

2.5.1 Barrier Synchronization Barrier synchronization operations not only directly affect application execution time but also impact application scalability. Performance optimization work for barrier synchronization operations is divided into two categories: unicast-based barrier synchronization performance optimization (also called software-based) and hardware-based barrier synchronization performance optimization. Absolute low latency and good scalability are key metrics for evaluating barrier synchronization performance, while software solutions often involve higher latency. Therefore, many high-performance computer systems employ hardware implementations. NEC' s Earth Simulator uses hardware to implement a Global Barrier Counter, completing synchronization operations for 64 nodes within 2-3 s. Cray T3D [9] implements a tree-based barrier synchronization dedicated network with node degree 4, completing global barrier synchronization operations in only 2 s. IBM BlueGene [10] implements a dedicated barrier synchronization network using a binomial tree structure, completing barrier synchronization operations for 65,536 nodes in just 1.4 s.

Dedicated networks can achieve high barrier synchronization operation performance but suffer from high cost, poor scalability, and inflexibility. In practical systems, barrier synchronization operations belonging to different applications exist simultaneously, and the processor sets for these operations may overlap. Dedicated networks, being mostly hardwired circuits, can only implement concurrent execution of multiple barrier synchronization operations by dividing physical boundaries. This limits the scale of barrier synchronization operations to determined physical boundaries and prevents free configuration. Supporting only one barrier synchronization operation per physical partition causes multiple barrier synchronization operations on the same processor to execute serially, reducing multi-task execution efficiency in the system.

Cray T3E [11] and Quadrics [12] provide barrier synchronization implementation solutions based on data networks. Cray T3E implements a tree-based barrier synchronization algorithm supporting 32 concurrent barrier synchronization operations. Cray T3E' s interconnect network is a direct network using a 3D Torus topology. The barrier synchronization tree structure is established by configuring barrier synchronization configuration registers in routers. These registers record leaf ports and parent ports of the tree structure at each router. During operation, when all leaf ports have received barrier synchronization arm packets, the router forwards the barrier synchronization arm packet to the parent port. If the router is configured as root, it broadcasts barrier synchronization completion packets to all leaf ports. To further reduce barrier synchronization latency, barrier synchronization data uses independent virtual channels for transmission in routers with the highest output priority. Ultimately, Cray T3E' s embedded barrier synchronization network can complete synchronization for 56 nodes within 2 s, representing a 7x performance improvement over pure software implementation, though its implementation method only suits 3D torus topology.

Quadrics' typical topology is fat-tree, implementing tree-based barrier synchronization algorithms. In Quadrics' barrier synchronization tree, leaves and roots are network cards. Its barrier synchronization execution process, shown in [Figure 4: see original paper], consists of three steps: (1) the root node initiates barrier synchronization start notification; (2) participating nodes reply with arrival synchronization responses to the root node upon receiving the start notification; (3) after receiving all nodes' arrival synchronization responses, the root node sends synchronization completion notifications to other nodes. Using this mechanism, synchronization for 64 nodes can be completed in 6 s [12].

2.5.2 Multicast Communication To reduce multicast communication latency and alleviate network congestion, multicast communication requires optimization. Similar to barrier synchronization, multicast optimization can be divided into software-based (i.e., unicast-based) multicast optimization and hardware-supported multicast optimization. Unicast-based multicast offers good portability but performs worse than hardware-supported multicast.

Hardware-based multicast algorithms can be divided into two categories [13]: path-based hardware multicast and tree-based hardware multicast. Path-based multicast is implemented by carrying multiple destination node addresses in the message header. As the multicast message passes each destination node, it discards one address from the header and continues routing with the next address. Tree-based hardware multicast transmits messages as far as possible along a shared path, then replicates and distributes messages to multiple links at tree branch points. This method requires switches/routers to support message replication and forwarding.

Path-based multicast only forwards messages to one output port per level. Therefore, under severe network contention, path-based multicast performance

will be superior to tree-based multicast. However, in large-scale network environments, message header overhead becomes excessive. Additionally, in multistage interconnection networks where nodes are located at leaf positions, the average distance of path-based multicast is longer than tree-based hardware multicast. For this reason, path-based multicast is more suitable for direct interconnect networks, with related research focusing on direct interconnect networks as the background [14-16]. Consequently, most existing cluster networks supporting hardware multicast adopt tree-based hardware multicast algorithms.

Hardware-based multicast has two implementation approaches: dedicated network acceleration and embedded network acceleration. Multicast operations involve large amounts of data transmission, so the implementation cost of dedicated multicast networks is no less than that of data networks. For this reason, aside from two IBM systems, few systems adopt dedicated multicast networks. The IBM CM5 [17] system uses a dedicated control network to implement tree-based hardware multicast, but it only supports one multicast operation at a time and is primarily used for control message transmission with very short message lengths (maximum 16 bytes), providing limited acceleration for multicast operations. IBM Blue Gene/L also uses a dedicated network to implement tree-based multicast.

Embedded network implementation can achieve high multicast bandwidth while also providing excellent scalability. IBM SP2 [1] implements hardware multicast via embedded networks. Since its Switch3 employs output-queued structure, its multicast uses fanout-splitting approach, achieving over 90% multicast throughput. Quadrics also uses embedded multicast networks, with hardware multicast providing 10x bandwidth improvement over software [18]. However, Quadrics only supports multicast where all destination node addresses are contiguous, with no "holes" allowed between multiple destination addresses, causing a single multicast operation to potentially involve multiple multicast trees [18]. The limitation of only supporting contiguous destination nodes also exists in NEC Cenju-3' s multicast network [19].

The latest Infiniband networks also provide reliable hardware tree-based multicast support based on HCA (Host Channel Adapter).

2.6 Summary

Through the introduction of these representative networks, the following conclusions can be drawn regarding the challenges facing network design:

- **Bandwidth:** 10 Gigabit links remain the choice for most high-performance interconnect networks. To achieve higher effective bandwidth, network topologies with bisection bandwidth (such as fat-trees) are widely used, and transceiver DMA engines are embedded in network interfaces to enable efficient data transmission. However, bandwidth growth is limited by high-speed serial transceiver circuit development,

and existing network structures cannot break through this limitation.

- **Latency:** Low network diameter topologies, cut-through switching, and source routing all help reduce hardware latency, while software latency reduction is achieved through user-level communication interfaces and hardware offloading of some protocol processing overhead.
- **Scalability:** Inter-node interconnect scalability is achieved through network topologies with good scalability, while intra-node interconnect scalability has received little attention to date.
- **Manageability:** Manageability has received universal attention, with both in-band and out-of-band management approaches being used.
- **Collective Communication:** Hardware-based performance optimization can achieve high performance, with hardware approaches including dedicated networks and embedded networks. The dedicated network approach sets up a separate dedicated collective communication network outside the unicast data network, used only for transmitting collective communication data. The embedded network approach adds collective communication support to existing unicast networks without requiring dedicated devices or independent wiring.

The advantages and disadvantages of these two implementation methods are shown in . Dedicated networks can achieve optimal acceleration performance, but the use of dedicated devices and separate wiring reduces system reliability and increases complexity and cost, making them unsuitable for large-scale systems, especially petaflops systems. The embedded network approach can also achieve good performance and represents a more cost-effective solution.

3 Dawning 5000 High-Performance Interconnect Network Design

This section presents the design of the Dawning 5000 high-performance interconnect network. As shown in [Figure 5: see original paper], it addresses shortcomings in existing network designs and draws on experience from various networks, featuring the following characteristics: multi-rail network structure with fat-tree topology for single-layer networks; virtual cut-through switching, which according to related research can achieve lower latency in cluster environments [20]; source-based deterministic routing, which has ordering preservation and implementation simplicity characteristics, and according to research in [21] can achieve performance similar to adaptive routing under fat-tree topology; absolute credit-based flow control, which has timely flow control and can tolerate packet loss/errors; embedded network-based collective communication optimization, which from a cost-performance perspective is the most reasonable choice for accelerating collective communications; and out-of-band management network using Ethernet links with TCP/IP-based data transmission to ensure reliable management data transfer.

The multi-rail network structure is an important feature of the Dawning 5000 interconnect network and will be introduced in detail below. Additionally, the network interface controller and switch chip are core components of the interconnect network, and this section will also provide detailed introduction to their microarchitecture.

3.1 Multi-Rail Network

A multi-rail network refers to interconnecting system nodes with two or more layers of networks that have identical functions and performance and are independent of each other. Multi-rail technology is analogous to multi-core technology in processors: by establishing multiple parallel network layers, it breaks through network bandwidth limitations imposed by process technology, enabling aggregated network bandwidth to increase with the number of output ports on network interface controllers. Additionally, multiple network layers provide mutual redundancy, greatly enhancing network fault tolerance.

Two strategies exist for building multi-rail networks: a high-bandwidth strategy using fewer network layers but with higher per-layer bandwidth, and a multi-port strategy using more network layers but with lower per-layer bandwidth. In practice, the multi-port strategy can be viewed as further subdivision of per-layer bandwidth in the high-bandwidth strategy. To determine the optimal construction strategy, this section establishes performance models for both strategies. The message passing latency model under the high-bandwidth strategy is:

$$T_{high} = \text{Max}\left(\frac{L}{BW_{sl}}, T_{switch}\right) \times \text{Hop\_cnt} + \frac{L}{BW_{sl}} \tag{3.1}$$

The message passing latency model under the multi-port strategy is:

$$T_{multi} = \text{Hop\_cnt} \times \left(\text{Max}\left(\frac{L}{BW_{ml}}, T_{switch}\right) + \frac{L}{k \times BW_{ml}}\right) + \frac{m \times L}{BW_{ml}} \tag{3.2}$$

where MTU (maximum packet length) in virtual cut-through networks is L, with n consecutive network packets of length L being transmitted. In equations (3.1) and (3.2): BW_{ml} is the per-layer network link bandwidth in the multi-port strategy, BW_{sl} is the network link bandwidth in the high-bandwidth strategy; $k \times BW_{ml}$ is the network interface controller' s input bandwidth; T_{switch} is the single-stage switch latency, T_{line} is the single-stage transmission delay; Hop_cnt is the number of network hops, and m is the number of parallel network layers.

The performance improvement metric is defined as the ratio of high-bandwidth latency to multi-port latency:

$$G = \frac{T_{high}}{T_{multi}} \tag{3.3}$$

If G is greater than 1, the multi-port strategy' s performance is higher than the high-bandwidth strategy' s performance. Setting the single-layer network as a three-level fat-tree with Hop_cnt = 5, $k = 4$, $T_{line} = T_{switch} = 50$ns, $BW_{ml} = 2.5$Gbit/s (consecutively sending 10,000 packets), the performance improvement curve obtained from equation (3.3) is shown in [Figure 6: see original paper]. It can be seen that when the number of network layers m equals the k value, the multi-port strategy can achieve performance no lower than the high-bandwidth strategy, and within a certain packet length range (shorter packets), it outperforms the high-bandwidth strategy. It is worth emphasizing that the multi-port strategy does not reduce latency for individual messages but reduces sustained message transmission latency, i.e., it improves message rate (number of messages sent per unit time). The continuously increasing number of processor cores will be accompanied by large numbers of messages, making message rate an important communication performance metric. Therefore, multi-rail networks will become mainstream in the future, with the multi-port strategy being the optimal construction method.

3.2 Dawning 5000 Network Interface Controller

The structure of the Dawning 5000 network interface controller is shown in [Figure 7: see original paper], with the following design features: system interface directly connected to the crossbar, with input bandwidth increasing as the number of processors increases, solving intra-node scalability issues at the structural level; implementation of a globally unified physical address space to help further reduce communication latency; multi-rail network interface cooperating with the previous multi-rail network design to achieve data distribution and aggregation across multiple network layers; remote Load/Store implementation for low-latency fine-grained access; multi-channel support for implementing user-level communication interfaces and offloading some communication protocol overhead; multiple DMA engines for timely request processing; and barrier synchronization module for hardware completion of intra-node process synchronization.

The key to reducing software latency in the Dawning 5000 network interface controller lies in its multi-channel design. For MPI' s Eager and Rendezvous message passing modes, the multi-channel design includes NAP [7] and MSG [8] channels.

The NAP channel is used for short message transmission, supporting Eager mode. To enable data reception before the receiving process initiates reception, the NAP channel pre-allocates a receive buffer. This buffer is divided into a series of equal-length spaces forming a circular queue managed by the NAP channel. The NAP channel continuously fills received packets at the queue tail,

while the communication library reads queue head elements directly through polling.

The MSG channel is used for long message transmission, supporting Rendezvous mode. The handshake process in Rendezvous is completed through the NAP channel, with the MSG channel responsible for data transmission. MSG messages carry target receive buffer addresses (obtained during handshake), implementing Remote Direct Memory Access (RDMA [9]) functionality by writing data directly to this address. Since data is written directly into the process receive buffer, the MSG channel does not require pre-allocated receive space, though it still maintains a receive event circular queue that the communication library polls to obtain data reception completion notifications.

NAP and MSG channels are paired in the Dawning 5000 network interface controller and bound to processes, providing user-level communication interface support. Multiple DMA engines set up in the controller can timely process data transmission requests for all NAP and MSG channels.

3.3 Dawning 5000 Switch Chip

The Dawning 5000 switch chip employs Virtual Channel Input Queuing structure, which can greatly alleviate performance degradation caused by head-of-line blocking, and is easy for flow control, physical implementation, and scalability. However, different from traditional structures, the switch chip uses parallel switching, i.e., multiple crossbars implement internal data exchange, and employs function-separated virtual channels where multicast and barrier synchronization data are buffered through independent virtual channels, as shown in [Figure 8: see original paper].

Proper selection of virtual channel numbers and buffer sizes is key to improving switch throughput. Using uniform random communication patterns, switch performance curves under different virtual channel numbers are shown in [Figure 9: see original paper], where Figure 9: see original paper shows test curves for a 16-port switch and Figure 9: see original paper shows throughput improvement with increasing virtual channel numbers. The figure reveals the following characteristics of virtual channel impact on switch performance:

1. For all virtual channel configurations, switch throughput has a critical saturation value. When throughput is below the critical saturation value, average packet delay grows slowly; when throughput approaches or exceeds the critical saturation value, average packet delay increases rapidly; ultimately, switch throughput converges and does not grow indefinitely.
2. For all virtual channel configurations, average packet delay is essentially the same with minimal differences under low throughput conditions.
3. As the number of virtual channels increases, maximum throughput also increases, but the increase is not linear. Two virtual channels provide a 27% performance improvement over one virtual channel, three virtual channels

provide 7.9% improvement over two, and four virtual channels provide only 2.9% improvement over three, yet four virtual channels increase resources by one-third compared to three. Therefore, virtual channel number selection must consider cost-performance factors to achieve balance between hardware resources and performance. For a 16-port switch, three virtual channels is a reasonable choice.

With virtual channel number determined, buffer size selection becomes critical. In a 16-port switch using 3 virtual channels, performance curves obtained with different buffer sizes are shown in [Figure 10: see original paper]. It can be seen that throughput increases with buffer size but eventually converges. For packet lengths between 128-1024 bytes, throughput converges when buffer size is four times the packet length. Therefore, in a 16-port switch, to achieve maximum throughput under various packet lengths, buffers must be set to 4 times the maximum packet length.

In virtual channel input-queued structure, a $kN \times N$ crossbar must be implemented (where k is the number of virtual channels and N is the number of ports), with switch input speedup of k. Different from traditional single-plane switching, Dawning 5000 switch chips use parallel switching, replacing the $kN \times N$ crossbar with k $N \times N$ crossbars while maintaining input speedup of k.

Parallel switching facilitates higher switch chip operating frequencies. Implementing a $kN \times N$ crossbar requires a kN arbiter with arbitration path length proportional to kN, limiting arbitration frequency improvement. Parallel switching only requires a $1 : N$ arbiter with arbitration length of only $1/k$ that of single-plane switching. Additionally, parallel switching facilitates modular design, with timing constraints targeting $N \times N$ crossbars for easier timing convergence and higher chip operating frequencies.

4 Dawning 5000 Embedded Barrier Synchronization Network Design

4.1 Design Goals

Achieving low-latency barrier synchronization communication is the primary goal of barrier synchronization network design. As shown in equation 4.1, barrier synchronization communication latency consists of fixed latency $T_{barrier\_constant}$ and network contention latency $T_{contention}$, where network contention latency varies dynamically with network load. Fixed latency is shown in equation 4.2, where $T_{software}$ is the network interface card' s per-message send/receive overhead, T_{switch} is the single-stage switch barrier synchronization processing delay, T_{line} is the inter-stage transmission delay, n is the number of times passing through the network interface card, and Hop_cnt is the number of times passing through switches. $T_{software}$, T_{switch}, and T_{line} are hardware implementation-related, while n and Hop_cnt are barrier synchronization communication path-related, and $T_{contention}$ is barrier

synchronization communication mechanism-related. Therefore, the key to reducing barrier synchronization communication latency lies in:

1. Reducing $T_{software}$ and T_{switch}: Simplifying barrier synchronization processing flows in network interface cards/switches;
2. Reducing Hop_cnt: Shortening barrier synchronization communication path length;
3. Reducing $T_{contention}$: Reducing network competition between barrier synchronization and unicast communications, minimizing unicast communication impact on barrier synchronization communication.

In addition to achieving low latency, barrier synchronization networks must ensure reliability of barrier synchronization communication. In a network with 1024 nodes, 320 switches are needed for interconnection, with 3072 communication links and 6144 serializer/deserializer (Serdes) converters. With a bit error rate of 10^{-13}, one global barrier synchronization error occurs daily. A single barrier synchronization packet error can either affect barrier synchronization operation performance or cause program crashes. Therefore, barrier synchronization networks need to provide reliability support at the link layer protocol to ensure timely recovery from errors.

Simultaneously, for issues that cannot be recovered through communication protocols such as equipment failures and link disconnections, barrier synchronization networks require manageability design. Manageability means the barrier synchronization network can monitor barrier synchronization communication status in real-time, detect and report faults promptly, and respond to subsequent processing by management software. Manageability will be implemented through Dawning 5000' s out-of-band management network. This section focuses on discussing barrier synchronization network design from latency and high reliability perspectives.

4.2 Latency

Fat-tree topology has a natural tree structure, so Dawning 5000 barrier synchronization networks choose to implement tree-based barrier synchronization algorithms. In tree-based algorithms, barrier synchronization operations consist of two processes: barrier synchronization arrival notification and barrier synchronization completion notification. The arrival notification is the process where processors notify the system that they have reached the barrier synchronization point. This process has a bottom-up reduction characteristic and is also called the reduction process. The completion notification is the process where the system notifies processors that the system has reached the barrier synchronization point. This process has a top-down distribution characteristic and is also called the distribution process. The reduction and distribution paths constitute the communication path for a barrier synchronization operation. To reduce barrier synchronization communication time, the first step is to reduce the length of the barrier synchronization communication path.

Under fat-tree topology, processor nodes are located at the bottom level of the tree structure. Let the communication length between nodes be $L_{leaf \to leaf}$, and the communication length between nodes and top-level switches be $L_{leaf \to root}$. If a processor node is selected as the root of the barrier synchronization tree, the barrier synchronization operation communication path length is:

$$L_{leaf \to leaf} + 2 \times L_{leaf \to root}$$

If the top-level switch is selected as the root of the barrier synchronization tree, the barrier synchronization communication length is:

$$2 \times L_{leaf \to root}$$

Therefore, selecting the top-level switch as the root of the barrier synchronization tree can halve the communication path, reducing barrier synchronization transmission latency. We call this tree structure with switches as barrier synchronization root nodes the "Dawning 5000 barrier synchronization tree structure." In Dawning 5000' s barrier synchronization tree structure, switches are responsible for reduction and distribution of barrier synchronization packets. The barrier synchronization operation process is shown in [Figure 11: see original paper], with the communication path being only 1/3 that of Quadrics network' s barrier synchronization communication path.

Using Dawning 5000' s barrier synchronization tree structure can reduce fixed latency of barrier synchronization communication. Barrier synchronization network design also needs to reduce barrier synchronization network contention latency. To reduce the impact of unicast communication on barrier synchronization communication, Dawning 5000 barrier synchronization networks implement two optimizations: First, dedicated virtual channels are set up for barrier synchronization in Dawning 5000 switch chips. By adding dedicated virtual channels for barrier synchronization, barrier synchronization performance can be improved by nearly 7x, as shown in [Figure 12: see original paper]. Additionally, the highest output priority is set for barrier synchronization channels in Dawning 5000 switch chips, ensuring barrier synchronization packets are prioritized when output contention occurs with unicast traffic.

4.3 Reliability

In Dawning 5000' s barrier synchronization tree structure, switches are responsible for barrier synchronization data processing, providing the prerequisite for implementing point-to-point reliability protocols. Compared to Quadrics'end-to-end protocol, point-to-point reliability protocols can achieve more timely error recovery.

To solve packet loss/corruption, existing point-to-point reliability protocols mostly use request-response mechanisms, where the sender actively retransmits

data if no acknowledgment packet is received within a certain time. This section also considers another timeout-urge mechanism, where the receiver sends an urge packet requesting retransmission if no data packet is received within a certain time.

Applied to barrier synchronization communication, the workflows of both mechanisms are analyzed. Let operation start time be t_0, timeout threshold be D, and communication delay between adjacent switch chips be T. The operation process is shown in [Figure 13: see original paper]. For the reduction process, the request-response mechanism can recover from errors more promptly. For the distribution process, both mechanisms complete error recovery in the same time.

Therefore, the reduction process is suitable for adopting a request-response-based reliability mechanism. For distribution process reliability mechanism selection, communication volume and implementation complexity must also be considered. Using request-response requires the switch chip to collect distribution ACK packets from all leaf ports for each distribution operation, similar to the barrier synchronization reduction process requiring independent processing state setup. Using timeout-urge only requires processing distribution urge packets from individual leaf ports for each distribution operation without requiring state setup. Therefore, the timeout-urge mechanism can reduce the number of barrier synchronization packets used for reliability and simplify barrier synchronization state machine design, making it suitable for the distribution process from a reliability mechanism perspective.

Based on the above analysis, the reliability mechanism for barrier synchronization communication is defined as follows: After sending reduction packets, the switch chip initiates reduction retransmission and distribution urge timeout counting. If retransmission timeout occurs without receiving reduction ACK packets, reduction packet retransmission is initiated; otherwise, retransmission counting stops. If urge retransmission timeout occurs without receiving distribution packets, distribution urge packet transmission is initiated.

5 Dawning 5000 Embedded Multicast Network Design

Tree-based multicast algorithms easily enable shortest-path communication between source nodes and all destination nodes, making them more suitable for fat-tree topology. Implementing tree-based multicast operations makes tree structure establishment strategy (i.e., multicast path selection strategy) critical to multicast performance. Implementing tree-based multicast via embedded networks must consider two issues:

1. **Deadlock resolution:** Implementing tree-based hardware multicast carries deadlock risks. [Figure 14: see original paper] shows two deadlock risk scenarios: In Figure 14: see original paper, multicast A occupies port P3 of switch 0 requesting port P1 of switch 1, while multicast B occupies port P1 of switch 1 requesting port P3 of switch 0, forming inter-switch

cyclic dependency and causing deadlock, called inter-switch deadlock. The other scenario, shown in Figure 14: see original paper, has multicast A occupying port P1 requesting port P0, while multicast B occupies port P0 requesting port P1, forming intra-switch output port cyclic dependency and causing deadlock, called intra-switch deadlock. Both deadlock problems can be avoided through multicast path selection strategies or resolved through specific hardware support.

2. **Network contention reduction:** Multicast paths involve multiple output ports at each level, greatly increasing the probability of network contention and reducing multicast performance. Network contention occurs both among multicast communications and between multicast and unicast communications. Therefore, multicast path selection strategies involve both static and dynamic aspects: measures to reduce network contention should be incorporated during initial multicast path establishment, and multicast paths should be reasonably adjusted based on network congestion status.

This section discusses design issues related to deadlock resolution and multicast path selection to achieve optimal multicast network performance, using fat-tree topology as the background.

5.1 Deadlock Avoidance Mechanism Based on Global Resource Bulletin

Dawning 5000 interconnect networks adopt FIFO (First-In-First-Out) packet buffer management, where packets can only be read once, thus only supporting no-splitting multicast and carrying deadlock risks. Dawning 5000 interconnect networks use virtual cut-through switching, so intra-switch deadlock must also be resolved.

This section proposes a "Global Resource Bulletin" deadlock avoidance mechanism with low implementation complexity and high multicast throughput. The core idea of the Global Resource Bulletin mechanism is to revoke invalid requests. Invalid requests refer to multicast requests that cannot be fully satisfied by output ports. The Global Resource Bulletin is a global resource register using bitmap format to record output ports in the switch without multicast packet transmission.

Using the Global Resource Bulletin mechanism, the multicast processing flow in switches is shown in [Figure 15: see original paper], consisting of five steps:

1. When all output ports requested by a multicast packet are valid in the Global Resource Bulletin register and have sufficient flow control credits, the request is initiated.
2. After arbitration succeeds, the requested ports are removed from the Global Resource Bulletin.
3. The multicast packet is read from the input queue.

4. The packet is transmitted to all requested output ports.
5. When transmission completes, the ports removed in step 4 are reset to valid in the Global Resource Bulletin.

As seen in step 2, using the Global Resource Bulletin completely avoids situations where a multicast operation occupies some output ports while requesting others, preventing cyclic dependency and achieving atomicity of multicast packet output. Due to the use of fair arbitration strategies, it better achieves load balancing compared to methods in [22]. After the arbiter provides arbitration grant signals, the next packet' s arbitration can proceed, allowing arbitration and data transmission to execute in pipeline fashion for high multicast transmission efficiency.

5.2 Multicast Path Selection Mechanism Based on HLSE

Compared to deterministic path selection algorithms, load balancing-based strategies can reduce network contention to some extent and improve multicast throughput. Existing methods only focus on load at routing computation time and use precise path selection, causing multiple multicast communications at the same time to select similar paths and exacerbating network congestion. Therefore, they cannot truly achieve load balancing, as seen in LLP [23] and LLP_EC [24] algorithms.

Addressing this issue, this section proposes a multicast path selection mechanism based on Heavy-Loaded Switch Eliminating (HLSE).

Analysis of multicast communication behavior in fat-tree FT(8,3) reveals that 50% of multicast latency is consumed in root switches, and root switch load can reflect the load of entire multicast paths. Additionally, fat-tree topology has this characteristic: if a switch is selected as root, the multicast path connecting all nodes in the multicast tree is determined and unique. Therefore, multicast path selection is essentially root switch selection.

Different from LLP [11] and LLP_EC' s precise selection methods, this section adopts a random-based HLSE multicast path selection algorithm. HLSE algorithm pseudocode is shown in [Figure 16: see original paper]. The main idea of HLSE algorithm is to eliminate heavily loaded switches, then randomly select lightly loaded switches. Using HLSE algorithm can distribute multicast communications across multiple lightly loaded parallel communication paths at transmission time, effectively avoiding multicast path conflicts caused by precise selection methods and mitigating effects of lagging load messages.

6 Dawning 5000 Interconnect Network Prototype System

The Dawning 5000 interconnect network prototype system targets interconnection of 1024 processing nodes. The network interface controller connects to two processors via two HT buses [12], with ingress aggregated bandwidth of 12.8 Gbps (single HT interface is 16-bit data, 200MHz DDR). Network links use TI'

s TLK2711 as serializer/deserializer, supporting maximum bandwidth of 3.125 Gbps. Therefore, the ratio of network interface controller input bandwidth to single-layer network link bandwidth is 4. According to analysis in Section 3.1, the multi-rail network layer count is set to 4 to achieve maximum performance improvement.

The core component for building the Dawning 5000 interconnect network prototype system is the 16-port Dawning 5000 switch chip, which uses 2 unicast virtual channels, one unicast/multicast mixed virtual channel, and one dedicated barrier synchronization virtual channel. Unicast and multicast virtual channel buffer sizes are 4 Kbytes, while barrier synchronization is 128 bytes.

The Dawning 5000 switch chip was first implemented using a Xilinx XC4Vlx160-10ff1513 FPGA chip for logic function verification. Based on correct FPGA functional verification, a dedicated ASIC implementation was performed. Xilinx XC4Vlx160 belongs to the Virtex4 series with 16 million gates, containing 67,584 logic slices (each with two D flip-flops and two 4-input LUTs), 288 Block RAMs (2 Kbytes each). The Dawning 5000 switch chip operates at 100MHz, uses 17 clock domains, 77% of logic slice resources, 38% of Block RAM resources, and 690 I/O pins. The FPGA-implemented Dawning 5000 switch contains four layers of parallel networks, with out-of-band management using ARM processors for data processing.

The Dawning 5000 switch chip ASIC implementation uses UMC HJ' s generic 0.18 m/6M CMOS process standard cells, integrating 20 million transistors with 1053 pins (690 I/O pins), using flipchip packaging. The entire die area is 12.4 $\times$ 12.4 mm^2, with standard cell area of 13.44 mm^2, memory area of 24.49 mm^2, and total area (excluding I/O pads) of 38.2 mm^2. The chip operates at 156.25 MHz with worst-case power consumption of 3W.

7 Performance Evaluation

This section evaluates Dawning 5000 interconnect network performance in unicast, multicast, and barrier synchronization. Each evaluation consists of two parts: performance evaluation of the FPGA prototype system (interconnected through single-level switches) and large-scale network performance analysis using HPPNetSim [25] based on prototype system performance parameters.

7.1 Unicast Performance

Under completely unloaded network conditions, the minimum application-level latency for a single data transfer in the Dawning 5000 interconnect network prototype system is 1.73 s, slightly higher than Infiniband' s 1.31 s but lower than Quadrics and Myrinet networks. In the minimum latency, software and network interface controller send/receive latency is 1.4 s, data transmission line latency (including serializer/deserializer) is 0.2 s, and switch chip latency is 0.13 s.

Substituting prototype system performance parameters into HPPNetSim yields unicast communication performance curves for 1024-node scale Dawning 5000 interconnect networks, as shown in [Figure 17: see original paper]. It can be seen that under hot-region [13] communication patterns, due to poor communication locality and hotspot access, maximum throughput can only reach 28%. Under other communication patterns, system throughput can reach over 70%, with the highest locality Sphere pattern [14] achieving up to 80% throughput (beyond the range shown in Figure 17: see original paper).

7.2 Barrier Synchronization Performance

Figure 18: see original paper shows single-level barrier synchronization operation latency in the prototype system, where latency for 8 processes performing one barrier synchronization operation is only 1.78 s. The other curve in Figure 18: see original paper shows measured performance of Infiniband third-generation cards (20 Gbps unidirectional link bandwidth). Since it does not provide hardware-level barrier synchronization operation support, barrier synchronization operations can only be implemented based on unicast, with latency increasing as process count increases. It can be seen that Dawning 5000' s barrier synchronization operation performance shows significant improvement over Infiniband.

Figure 18: see original paper shows barrier synchronization performance under mixed communication patterns obtained by substituting prototype system parameters into HPPNetSim. It can be seen that barrier synchronization network contention latency is directly related to unicast packet length, growing linearly with unicast packet length. Second, barrier synchronization network contention latency is related to unicast communication patterns. When unicast communication follows random uniform distribution, barrier synchronization latency is lowest; under communication patterns with higher locality, barrier synchronization latency is higher. This indicates that local communication hotspots reduce barrier synchronization communication performance. However, even under worst conditions (unicast packet length of 1K bytes, hot-region communication pattern), maximum barrier synchronization latency is only 8.23 s, demonstrating high performance.

7.3 Multicast Performance

The Dawning 5000 switch chip completes one multicast communication with 150ns latency. Since the network interface controller prototype does not include corresponding multicast interfaces, single-level multicast latency in the prototype system cannot be obtained. Using unicast communication parameters (software latency and network interface controller latency) in HPPNetSim yields broadcast latency under mixed communication patterns as shown in Figure 19: see original paper. When both unicast and multicast coexist in the network, the HLSE multicast routing algorithm proposed in this paper achieves the lowest latency, with nearly 3x performance improvement.

Under 1024-node scale, the all-to-all multicast performance curve is shown in Figure 19: see original paper. It can be seen that system multicast throughput can ultimately reach nearly 93%, proving that Dawning 5000 multicast networks achieve high performance. When multicast throughput reaches maximum (i.e., network saturation), single multicast latency grows rapidly. However, the increasing latency comes entirely from network queuing time, which is amortized across large numbers of multicast communications during pipelined output. Testing and analysis show that after network saturation, the system completes one multicast communication on average every 5.2 s.

8 Conclusion

This paper discusses design methodologies for the Dawning 5000 high-performance interconnect network from perspectives of interconnect network architecture, network interface controller design, switch architecture design, and collective communication performance optimization.

This paper first proposes the Dawning 5000 interconnect network structure based on multi-rail networks to break through process technology limitations on network bandwidth. The paper further proposes using multiple low-bandwidth networks to build multi-rail networks to improve short message message rates. Additionally, Dawning 5000 interconnect network' s single-layer network design fully considers performance, scalability, and manageability needs, adopting fat-tree topology, virtual cut-through switching, source-based deterministic routing, absolute credit-based flow control, embedded collective communication networks, and out-of-band management networks. This paper conducts in-depth research on hybrid network design, fully considering performance balance between embedded collective communication networks and unicast networks. Test results show that Dawning 5000 interconnect network achieves high performance in unicast, multicast, and barrier synchronization communications.

References

[1] C.B. Stunkel, D. G. Shea, B. Aball, et al., "The SP2 High-Performance Switch," IBM Systems J, vol. 34, 1995: 185-204
[2] N. Boden, D. Cohen, R. Felderman, et al., "Myrinet: A Gigabit per Second Local Area Network.," in IEEE Micro Magazine, February 1995
[3] Fabrizio Petrini, Wu-chun Feng, Adolfy Hoisie, et al., "The Quadrics Network (QsNet): High-Performance Clustering Technology," in Proceedings of the 9th IEEE Hot Interconnects, Palo Alto, California, August 2001: 125-130
[4] Jay Herring, Craig B. Stunkel, Bulent Abali, Rajeev Sivaram, "A new switch chip for IBM RS/6000 SP systems," in Proceedings of the ACM/IEEE conference on Supercomputing Portland, Oregon, United States, 1999: 16-es
[5] Jon Beecroft, Mark Homewood, and Moray McLaren, "Meiko CS-2 interconnect Elan-Elite design," Parallel Computing vol. 20, 1994: 1627 - 1638
[6] Duncan Roweth and Trevor Jones, "QsNetIII an Adaptively Routed Network

for High Performance Computing," in 16th Annual IEEE Symposium on High Performance Interconnects Stanford, CA, USA, August 2008: 157-164
[7] Infiniband Trade Association, "The InfiniBand™ Architecture," 2001
[8] Intel and Microsoft Corporations Compaq, "Virtual Interface Architecture Specification. Version 1.0," Dec. 1997
[9] Adams D., "Cray T3D System Architecture Overview," in Technical report HR-040433, Cray Research Inc., 1994
[10] The BlueGene/L Team, "An Overview of the BlueGene/L Supercomputer," in International Conference for High Performance Networking and Computing, Maryland, 2002: 1-22
[11] Steven L. Scott, "Synchronization and communication in the T3E multiprocessor," in Proc. 7th International Conference on Architectural Support for Programming, Cambridge, MA, 1996: 26-36
[12] Petrini F., et al., "Hardware- and software-based collective communication on the Quadrics network," in IEEE International Symposium on Network Computing and Applications, Cambridge, MA, 2001: 24-36
[13] Jose Duato, Sudhakar Yalamanchili, and Lionel Ni, Interconnection Networks: An Engineering Approach Morgan Kaufmann Publishers, 2003
[14] Chien-Min Wang, Yomin Hou, and Lih-Hsing Hsu, "Adaptive Path-Based Multicast on Wormhole-Routed Hypercubes," in Proceedings of the 8th International Euro-Par Conference on Parallel Processing 2002: 757-766
[15] A. Yassin Al-Dubai, M. Ould-Khaoua, and L. M. Mackenzie, "An Efficient Path-Based Multicast Algorithm for Mesh Networks," in Proceedings of the International Parallel and Distributed Processing Symposium, 2003
[16] H. Harutyunyan and S. Wang, "Path-based multicasting in multicomputers," in Proceedings of the 25th IASTED International Multi-Conference, Innsbruck, Austria, 2007: 220-226
[17] Charles E. Leiserson, Zahi S. Abuhamdeh, David C. Douglas, et al., "The network architecture of the Connection Machine CM-5," in Proceedings of the fourth annual ACM symposium on Parallel algorithms and architectures, San Diego, California, Jun. 1992: 272-285
[18] Salvador Coll, José Duato, Fabrizio Petrini, et al., "Scalable Hardware-Based Multicast Trees," in Proceedings of the 2003 ACM/IEEE conference on Supercomputing 2003: 54-54
[19] N. Koike, "NEC Cenju-3:A microprocessor-based parallel computer,"in Proc. of the 8th International Parallel Processing Symposium, April 1994: 396-401
[20] J. Duato, A. Robles, F. Silla, et al., "A Comparison of Router Architectures for Virtual Cut-Through and Wormhole Switching in a NOW Environment," in Proceedings of the 13th International Symposium on Parallel Processing and the 10th Symposium on Parallel and Distributed Processing, 1999: 240 -247
[21] C. Gómez, F. Gilabert, M.E. Gómez, et al., "Deterministic versus Adaptive Routing in Fat-Trees," in CAC, California, USA, 2007
[22] Jaehyung Park, Lillykutty Jacob, and Hyunsoo Yoon, "Performance Analysis of a Multicast Switch based on Multistage Interconnection Networks," 1997
[23] Sameer Kumar, "Optimizing Communication for Massively Parallel Processing," in department of computer science. vol. Ph.d: University of Illinois at

Urbana Champaign, May 2005
[24] Quanbao Sun, Minxuan Zhang, and Liquan Xiao, "Hardware-Based Multicast with Global Load Balance on k-ary n-trees," in International Conference on Parallel Processing (ICPP 2007), 2007
[25] Zheng Cao, Jianwei Xu, Mingyu Chen, et al., "HPPNetSim: A Parallel Simulation of Large-scale Interconnection Network," in 42nd Annual Simulation Symposium, 2009
[26] Culler D., "LogP: towards a realistic model of parallel computation," in Principles Practice of Parallel Programming, San Diego, 1993: 1-12

Author Biographies

Zheng Cao: Ph.D., Institute of Computing Technology, Chinese Academy of Sciences
Dawei Wang: Ph.D., Institute of Computing Technology, Chinese Academy of Sciences
Xingkui Liu: Graduate Student, Institute of Computing Technology, Chinese Academy of Sciences
Xinchun Liu: Associate Researcher, Institute of Computing Technology, Chinese Academy of Sciences
Hua Shen: Associate Researcher, Institute of Computing Technology, Chinese Academy of Sciences

Note: Figure translations are in progress. See original paper for figures.