System Overview of the SGI Origin 200/2000 Product Line

James Laudon and Daniel Lenoski
Silicon Graphics, Inc.
2011 North Shoreline Boulevard
Mountain View, California 94043
laudon@sgi.com lenoski@sgi.com

Abstract

The SGI Origin 200/2000 is a cache-coherent non-uniform memory access (ccNUMA) multiprocessor designed and manufactured by Silicon Graphics, Inc. The Origin system was designed from the ground up as a multiprocessor capable of scaling to both small and large processor counts without any cost, bandwidth, or latency cliffs. The Origin system consists of up to 512 nodes interconnected by a highly scalable Craylink network. Each node consists of one or two R10000 processors and up to 4 GB of coherent memory. Each node also connects to the scalable XIO IO subsystem. This paper discusses the motivation for building the Origin 200/2000 and describes its architecture and implementation.

1 Background

Silicon Graphics has offered multiple generations of symmetric multiprocessor (SMP) systems based on MIPS microprocessors. From the 8 processor Power Series to the 36 processor Challenge and Power Challenge systems, the cache-coherent, globally addressable memory architecture of these SMP systems has provided a convenient programming environment for large parallel applications while at the same time providing an excellent system performance for both parallel and throughput-based workloads.

The follow-on system to the Power Challenge needed to meet three important goals. First, it needed to scale beyond the 36 processor limit of the Power Challenge and provide an infrastructure that supports higher performance per processor. Second, the new system had to retain the cache-coherent globally addressable memory model of the Power Challenge. This model is critical for achieving high performance on loop-level parallelized code and for supporting the existing Power Challenge customers. Finally, lower entry level and incremental system costs than a high-performance SMP were desired, with costs approaching that of a cluster of workstations.

Simply building a larger and faster snoopy bus-based SMP system could not meet all three of these goals. The shared-memory goal might be achievable, but it would surely compromise performance for larger processor counts and costs for smaller configurations. Therefore a very different architecture was chosen for use in the next generation Origin system. Origin employs distributed shared memory (DSM), with cache coherence maintained via a directory-based protocol.

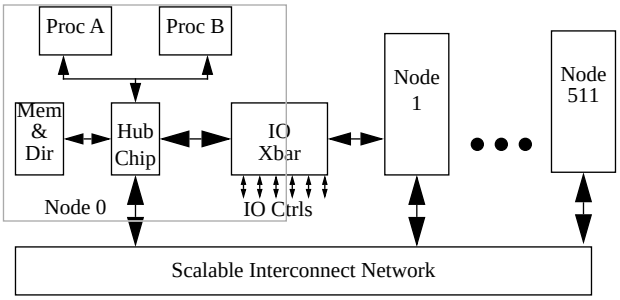

Figure 1 Origin block diagram

A DSM system has the potential for meeting all three design goals. The directory-based coherence removes the broadcast bottleneck that prevents scalability of the snoopy bus-based coherence. The globally addressable memory model is retained, although memory access times are no longer uniform. To address this non-uniformity, Origin was designed to minimize the latency difference between remote and local memory and to include hardware and software support to insure that most memory references are local. Finally, a low initial and incremental cost can be provided if the natural modularity of a DSM system is exploited at a relatively fine grain by the product design.

In Section 2 of this paper, the scalable shared-memory multiprocessing (S^2MP) architecture of the Origin is presented. Section 3 details the implementation of the Origin 200/2000. Finally, Section 4 concludes the paper.

2 The Origin S^2MP architecture

A block diagram of the Origin architecture is shown in Figure 1. The basic building block of the Origin system is the dual-processor node. In addition to the processors, a

node contains up to 4 GB of main memory and its corresponding directory memory, and has a connection to a portion of the IO subsystem.

The architecture supports up to 512 nodes, for a maximum configuration of 1024 processors and 1 TB of main memory. The nodes can be connected together via any scalable interconnection network. The cache coherence protocol employed by the Origin system does not require in-order delivery of point-to-point messages to allow maximum flexibility in implementing the interconnect.

The S^2MP architecture provides global addressability of all memory, and in addition, the IO subsystem is also globally addressable. Physical IO operations (PIOs) can be directed from any processor to any IO device. IO devices can DMA to all memory in the system, not just their local memory.

One of the major goals for the Origin system was to keep both absolute memory latency and the ratio of remote to local latency as low as possible to provide an easy migration path for existing SMP software. To keep the local memory latency low, the intra-node interconnect consists of single Hub chip that implements a full four-way crossbar between processors, local memory and the I/O and network interfaces. To keep the remote memory latency low, the global interconnect is based on a six-ported router chip configured in a multi-level fat-hypercube topology. Together, these chips minimize both intra- and inter-node delays to both local and remote memory.

In addition, while the two processors share the same bus connected to the Hub, they do not function as a snoopy cluster. Instead they operate as two separate processors multiplexed over the single physical bus (implemented as such to save Hub pins). This is different from most other ccNUMA systems, where the node is a SMP cluster, and further reduces memory latency. Local memory latency is reduced because the bus can be run at a much higher frequency with only two processors on it. Remote memory latency benefits from both the higher bus frequency and by not having to wait for the result of the bus snoop before the request can be forwarded to the remote node.

Latency focus also shows up in the coherence protocol which supports a clean-exclusive state to minimize latency on read-modify-write operations. Furthermore, it allows cache dropping of clean-exclusive or shared data without notifying the directory in order to minimize the memory bandwidth impact of directory coherence. The architecture also supports request forwarding to minimize the latency of interprocessor communication.

Origin includes architectural features to address the NUMA aspects of the machine. First, a combination of hardware and software features are provided for extremely effective page migration and replication. Page migration and replication reduces effective memory latency by allowing a greater percentage of accesses to be satisfied locally. To support page migration Origin provides per-page hardware memory reference counters and a block copy engine that is able to copy data at near peak memory speeds.

For effective synchronization in large systems, the Origin system provides fetch-and-op primitives on memory in addition to the standard MIPS load-linked/store-conditional (LL/SC) instructions. These operations greatly reduce the serialization for highly contended locks and barrier operations.

Origin includes many features to enhance reliability and availability. All external cache SRAM and main memory and directory DRAM are protected by a SEC/DED ECC code. Furthermore, the high-speed router and IO links are protected by a full CRC code and a hardware link-level protocol that detects and automatically retries corrupted packets. Origin's modular design provides the overall basis for a highly available hardware architecture. The flexible routing network supports multiple paths between nodes, partial population of the interconnect, and the hot plugging of cabled-links that permits the bypass, service, and reintegration of faulty hardware.

To address software availability in large systems, Origin provides access protection rights on both memory and IO devices. These access protection rights prevent unauthorized nodes from being able to modify memory or IO and allows an operating system to be structured into cells or partitions with containment of most failures to within a given partition[5][6].

3 The Origin implementation

While existence proofs for DSM systems have been available in the academic community for some time[1][4], the key to commercial success of this architecture is an aggressive implementation that provides for a truly scalable system with low memory latency and no unexpected bandwidth bottlenecks. In this section we explore how the Origin 200/2000 implementation meets this goal.

3.1 Network topology

The interconnect employed in the Origin 200/2000 system is based on the SGI SPIDER router chip[2]. The main features of the SPIDER chip are:

- six pairs of unidirectional links per router
- software programmable routing tables
- low latency (41 ns pin-to-pin) wormhole routing
- DAMQ buffer structures[2] with global arbitration to maximize utilization under load.
- four virtual channels per physical channel
- congestion control allowing messages to adaptively switch between two virtual channels

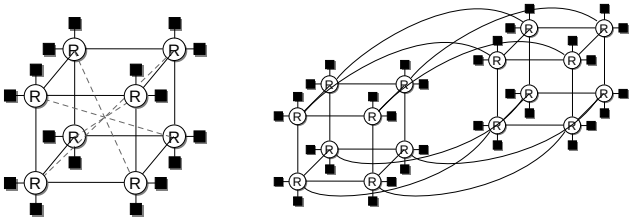

Figure 2 32P and 64P Bristled Hypercubes

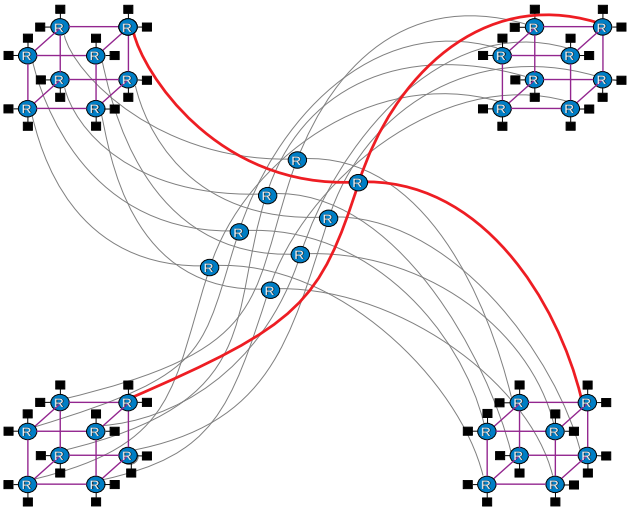

Figure 3 128P Hierarchical Fat Bristled Hypercube

- support for 256 levels of message priority with increased priority with packet aging
- CRC checking on each packet with retransmission on error via a go-back-n sliding window protocol

The Origin 2000 employs SPIDER routers to create a bristled fat hypercube interconnect topology. The network topology is bristled in that two nodes are connected to a single router instead of one. The fat hypercube comes into play for systems beyond 32 nodes (64 processors). For up to 32 nodes, the routers connect nodes in a bristled hypercube as shown in Figure 2. The SPIDER routers are labeled using R, the nodes are the block boxes connecting to the routers. In the 32 processor configuration, the otherwise unused SPIDER ports are shown as dotted lines being used for Express Links which connect the corners of the cube to reduce latency and increase bandwidth.

Beyond 64 processors, a hierarchical fat hypercube is employed. Figure 3 shows the topology of a 128 processors Origin system. The vertices of four 32 processor hypercubes are connected to eight *meta*-routers. To scale up to 1024 processors, the eight single meta-routers in the 128 processor system are replaced with eight 5-D hypercubes that connect thirty-two 32 processor hypercubes.

3.2 Cache coherence protocol

The cache coherence protocol of Origin is similar to the Stanford DASH protocol[4], but has several significant performance improvements. Like the DASH protocol, the Origin cache coherence protocol is non-blocking. Memory can satisfy any incoming request immediately; it never buffers requests while waiting for other messages to arrive. The Origin protocol also employs the reply forwarding of the DASH protocol for three party transactions. Reply forwarding reduces the latency of requests which target a memory that is cached by other processors.

The Origin coherence protocol has several enhancements over the DASH protocol. First, the Clean-exclusive (CEX) processor cache state (also known as the exclusive state in MESI) is fully supported by the Origin protocol. This state allows for efficient execution of read-modify-write accesses since there is only a single fetch of the cache line from memory. The protocol also permits the processor to replace a CEX cache line without notifying the directory. The Origin protocol is able to detect a re-request by a processor that has replaced a CEX cache line and immediately satisfy that request from memory. Support of CEX state in this manner is very important for single process performance since most of the gains from the CEX state would be lost if directory bandwidth was needed each time a processor replaced a CEX line. By adding protocol complexity to allow for the “silent” CEX replacement, all of the advantages of the CEX state are realized.

The second enhancement of the Origin protocol over DASH is full support of upgrade requests which move a line from a shared to exclusive state without the bandwidth and latency overhead of rereading the data from memory.

For handling incoming I/O DMA data, Origin supports a write-invalidate transaction that only requires a single memory write as opposed to the processor’s normal write-allocate plus writeback. This transaction is fully cache coherent (i.e. any implied cache invalidates are sent), and increases I/O DMA bandwidth by as much as a factor of two.

Origin’s protocol is completely insensitive to network ordering. Messages are allowed to bypass each other in the network and the protocol detects and resolves all out-of-order message deliveries. This allows Origin to employ adaptive routing in its network to deal with network congestion.

The Origin protocol uses a more sophisticated network deadlock avoidance scheme than DASH. As in DASH, two separate networks are provided for requests and replies (implemented in Origin via different virtual channels). The Origin protocol does have requests which generate additional requests (the additional requests being either *interventions* or *invalidations*). This request-to-request dependency could lead to deadlock in the request network.

In DASH, this deadlock was broken by detecting a potential deadlock situation and sending negative-acknowledgments (NAKs) to all requests which required generating additional requests. In Origin, rather than sending NAKs in such a situation, a *backoff* intervention or invalidate is sent to the requestor on the reply network. This backoff message simply tells the requestor that the memory was unable to generate the intervention or invalidation directly and the requestor must generate that message instead. The backoff intervention or invalidate changes the request-intervention-reply chain to two request-reply chains (one chain being the request-backoff pair, one being the intervention-reply pair). The ability to generate backoff interventions and invalidations allows for better forward progress in the face of very heavily loaded systems (the deadlock detection in both DASH and Origin is done conservatively at the memory based on local information).

Since the Origin system is able to maintain coherence over 1024 processors, it obviously employs a more scalable directory scheme than in DASH. For tracking sharers, Origin supports a bit-vector directory format with either 16 or 64 bits. Each bit represents a node, so with one bit per node the directory can track up to a maximum of 128 processors. For systems with greater than 64 nodes, Origin dynamically selects between a full bit vector and coarse bit vector[6] depending on where the sharers are located. This dynamic selection is based on the machine being divided into up to eight 64 node *octants*. If all the processors sharing the cache line are from the same octant, the full bit vector is used (in conjunction with a 3-bit octant identifier). If the processors sharing the cache line are from different octants, a coarse bit vector where each bit represents eight nodes is employed.

Finally, the coherence protocol includes an important feature for effective page migration known as *directory poisoning*, which will be discussed in Section 3.6.

3.3 Node design

The design of an Origin node fits on a single 16” x 11” printed circuit board. A block diagram of the Origin node board is shown in Figure 4. At the bottom of the board are two R10000 processors with their secondary caches. Each processor and its secondary cache is mounted on a horizontal in-line memory module (HIMM) daughter card. The HIMM is parallel to the main node card and utilizes very low-inductance fuzz-button processor and HIMM interposers to connect to the node card. The system interface buses of the R10000s are connected to the Hub chip. The Hub chip also has connections to the memory and directory on the node board, and has two ports that exit the node board via the 300-pin CPOP (compression pad-on-pad) connector. These two ports are the Craylink connection to router and the XIO connection to the IO subsystem.

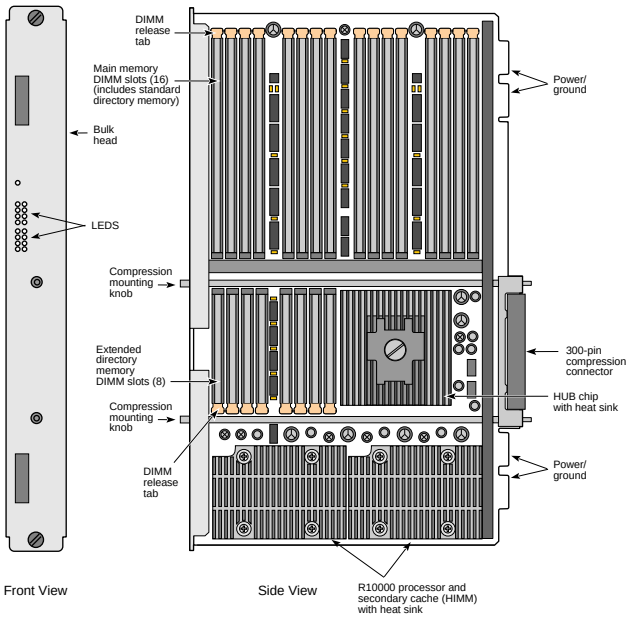

Figure 4 An Origin node board

As was mentioned in Section 3.2, a 16 bit-vector directory format and a 64 bit-vector format are supported by the Origin system. The directory that implements the 16-bit vector format is located on the same DIMMs as main memory. For systems larger than 32 processors, additional expansion directory is needed. These expansion directory slots, shown to the left of the Hub chip in Figure 4, operate by expanding the width of the standard directory included on the main memory boards. The Hub chip operates on standard directory entries by converting them to expanded entries upon their entry into the Hub chip. All directory operations within the Hub chip are done on the expanded directory entries, and the results are then converted back to standard entries before being written back to the directory memory. Expanded directory entries obviously bypass the conversion stages.

Figure 5 shows a block diagram of the Hub chip. The peak raw data bandwidth is listed for the four hub external ports. The hub chip is divided into five major sections: the crossbar (XB), the IO interface (II), the network interface (NI), the processor interface (PI), and the memory and directory interface (MD). All the interfaces communicate with each other via FIFOs that connect to the crossbar.

The IO interface contains the translation logic for interfacing to the XIO subsystem. The XIO subsystem is based on the same low-level signalling protocol as the Craylink network (and uses the same XIO interface block as in the SPIDER router of [2]), but utilizes a separate higher level message protocol The IO section also contains the logic for two block transfer engines (BTEs) which are able to perform memory to memory copies at near the peak of a

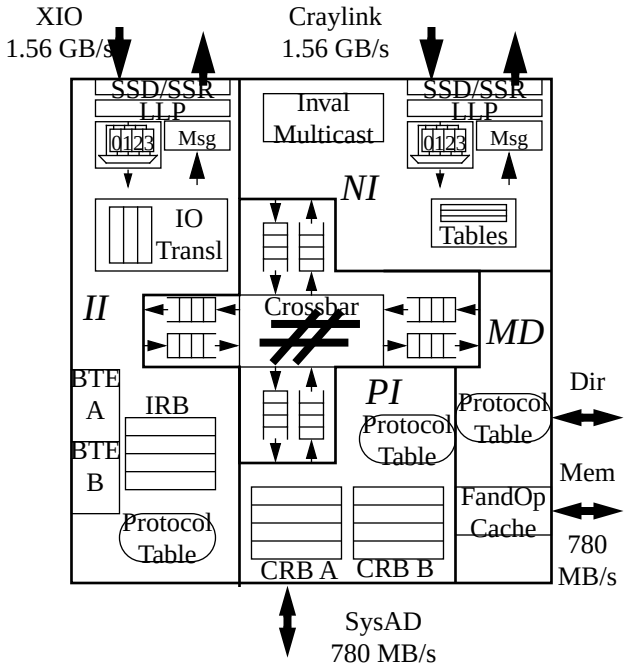

Figure 5 Hub ASIC block diagram

node's memory bandwidth. It also implements the IO request tracking portion of the cache coherence protocol via the IO request buffers (IRB) and the IO protocol table. The IRB tracks both full and partial cache line DMA requests by IO devices as well as full cache line requests by the BTEs.

The network interface takes messages from the II, PI, and MD and sends them out on the Craylink network. It also receives incoming messages for the MD, PI, II, and local Hub registers from the Craylink network. Software programmable routing tables are provided in the NI to support the pipelined routing used by the SPIDER chip[2]. The NI also is responsible for taking a compact bit-vector version of invalidation messages resulting from a coherence operation and generating the multiple unicast invalidate messages implied by that message.

The processor interface contains logic for implementing the request tracking for both processors. Read and write requests are tracked via a coherent request buffer (CRB), with one CRB per processor. The PI also includes the protocol table for its portion of the cache coherence protocol. The PI is responsible for controlling the flow of requests to and from the R10000 processors and generating interrupts to the processors.

Finally, the memory/directory section contains logic to interface to external data and directory memory implemented with synchronous DRAMs (SDRAMs). The MD performs the directory portion of the cache coherence protocol via its protocol table and generates the appropriate requests and/or replies for all incoming messages. It also contains a small fetch-and-op cache which sits in front of

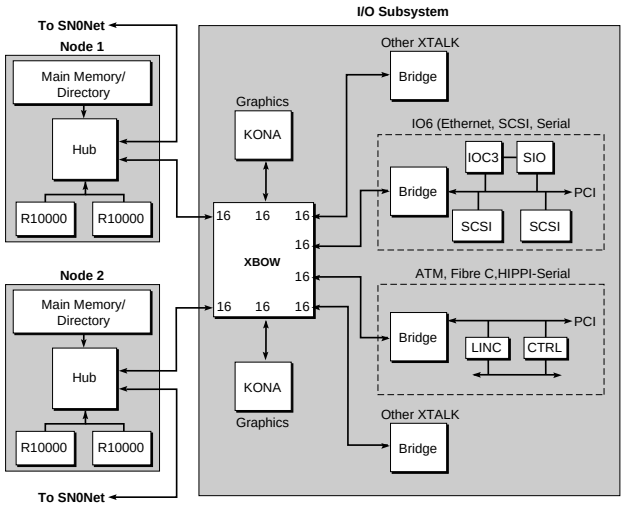

Figure 6 Example IO subsystem block diagram

the memory. This fetch-and-op cache allows fetch-and-op variables that hit in the cache to be updated at the minimum network serialization rate of 41 ns instead of at the much slower SDRAM read-modify-write timing.

3.4 IO subsystem

Not too surprisingly, the Origin system utilizes crossbars in its IO subsystem. Figure 6 shows one possible configuration of IO cards connected to two nodes. Using the same link technology as in the Craylink interconnect, each Hub link provides a peak of to 1.56 GB/sec of bandwidth to the six XIO cards connected to it (actually limited to half this amount if only local memory bandwidth is considered). As shown in Figure 6, at the heart of the IO subsystem is the Crossbow (Xbow) ASIC. The crossbow has many similarities with the SPIDER router. The primary difference in the Xbow being a simplification of buffering and arbitration protocols given the chip's more limited configurations. These simplifications reduce the cost and permit eight ports to be integrated on a single chip. Some of the main features of the Xbow are:

- eight XIO ports connecting two nodes and six XIO cards.
- two virtual channels per physical channel
- low latency wormhole routing
- support for allocated bandwidth among particular XIO devices
- CRC checking on each packet with retransmission on error via a go-back-n sliding window protocol

The Crossbow has support in its arbiter for allocating a portion of the bandwidth to a given IO device. This feature is important for certain system applications such as video on demand.

Board	Number of Ports
Base IO	2 Ultra SCSI, 1 Fast Enet, 2 serial
Ultra SCSI	4 differential
10/100 Enet	4 + 6 serial
HiPPI	1 serial
Fibre Channel	2 Cu loops
ATM OC3	4
Infinite Reality Gfx	1
Standard PCI Cage	3
VME Adapter	1

Table 1 Origin IO boards

A large number of XIO cards are available to connect to the Crossbow. Table 1 contains a listing of the common XIO cards. The highest performance XIO cards, such as Infinite Reality Graphics, connect directly to the XIO, but most of the cards bridge XIO to an embedded PCI bus with multiple external interfaces. The IO bandwidth together with integration provide IO performance which is effectively added as a PCI-bus at a time versus individual PCI cards.

3.5 Product design

The Origin 2000 is a highly modular design. The basic building block for deskside and rack use is shown Figure 7 and has slots for 4 node, 2 router, 12 XIO boards, and up to 6 Ultra SCSI devices. This module can be used as a stand-alone deskside system, or two modules can be mounted in a rack to form a 16 processor system. In addition to the embedded disk support, each rack also includes a disk bay for up to six 3.5” disks and two 5.25” removable media devices. One of the modules can be replaced with an Infinite reality graphics module or with 4 additional disk bays. An Origin Vault which contains 9 disk bays in a single rack is also available. For systems beyond 16 processors, multiple racks are connected together via Craylink cables.

The Origin 200 is a cost-reduced version of the Origin 2000. The basic building block for the Origin 200 is a two processor tower. This tower contains one or two processors, up to 2 GB of memory, 3 standard PCI slots, and up to 6 Ultra SCSI devices. In addition, a Craylink port is provided which allows two towers to be connected together to form a single shared-memory system containing up to four processors, as shown in Figure 8.

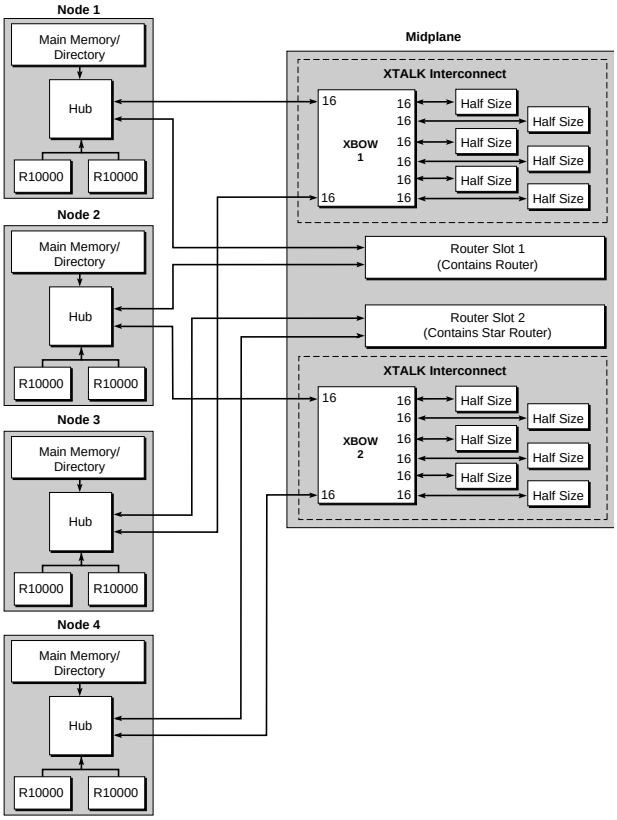

Figure 7 Deskside module block diagram

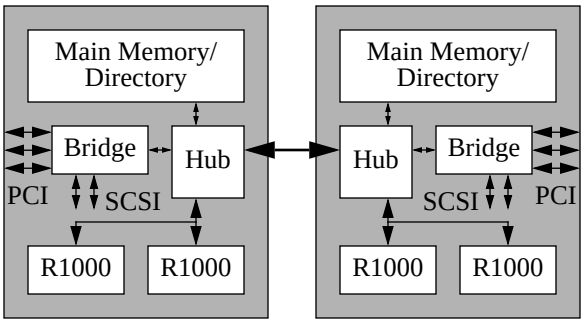

Figure 8 Origin 200 block diagram

3.6 Performance features

The Origin system has two features very important for achieving good performance in a highly scalable system. First, fetch-and-op primitives are provided as uncached operations that occur at the memory. Fetch-and-op variables are used for highly contended locks, barriers, and other synchronization mechanisms. The typical serialization rate (the rate at which a stream of requests can be serviced) for fetch-and-op variables is 41 ns.

Second, Origin provides hardware and software support for page migration. Page migration is important for NUMA systems as it changes many of the cache misses

which would have gone to remote memory to local misses. To help the OS in determining when and which page to migrate the Origin system provides an array of per-page memory reference counters, which are stored in the directory memory. This array is indexed by the nodes in a system (up to 64 nodes, beyond which 8 nodes share a single counter). When a request comes in, the reference counter associated with the requesting node is read out during the directory lookup and incremented. In addition, the reference counter of the home node is read out during the same directory lookup. The requestor's count and home count are compared and if the difference exceeds a software programmable threshold register (and the migration control bits stored with the requestor's reference counter indicates that this page is a candidate for migration), an interrupt is generated to the home node. This interrupt signals a potential migration candidate to the operating system.

If the operating system determines it does indeed want to migrate the page, two operations need to be performed. First, the OS must copy the page from its current location to a free memory page on the requestor's node. Second, all translations to the old page cached in processor's TLBs must be invalidated and the translation for the migrated page updated to point to the new page.

The block transfer engines in each Hub allow us to copy a 16 KB page from one node's memory to another in under 30 microseconds. Unfortunately, in a very large Origin system, the cost to invalidate all the TLBs and update the translation can be 100 microseconds or more, removing much of the benefit of providing a fast memory to memory copy. To solve the TLB update problem, the directory supports a block transfer copy mode known as directory poisoning, which overlaps the copy of the data to the new page with the update of the TLB, thereby allowing the operating system to be fairly aggressive in determining when to migrate a page.

4 Conclusions and status

The Origin 200/2000 is a highly scalable server designed to meet the needs of both the technical and commercial marketplaces. Origin achieves this by providing a highly modular system with a low-entry point and incremental growth to very large system sizes. A bristled fat hypercube network is used to provide a high bandwidth, low-latency interconnect. Low latency to local memory and a low remote to local memory latency ratio allow users to easily migrate their applications from the uniform access of the existing SMP systems to the NUMA Origin systems. Origin also includes features improve the performance including hardware and software support for page migration and fast synchronization.

5 Acknowledgments

The Origin system design resulted from the very hard work of a top-notch team of chip, board, and system engineers. Major contributors on the core logic design/verification team besides the authors included: Michael Anderson, John Andrews, Rick Bahr, John Burger, John Carlson, Hansel Collins, Pat Conway, Ken Choy, Asgeir Eiriksson, Paul Everhardt, Mike Galles, Sameer Gupta, Gary Hagensen, Dick Hessel, Roger Hu, Lee Jones, George Kaldani, John Keen, Ron Kolb, Yuval Koren, Waidy Lee, Viranjit Madan, John Manton, Greg Marlan, Dawn Maxon, David McCracken, Ali Moyedian, Bob Newhall, Kianoosh Naghshineh, Chuck Narad, Ron Nikel, Steve Padnos, Dave Parry, Ed Priest, Azmeer Salleh, Ken Sarocky, Chris Satterlee, Alex Silbey, Doug Solomon, Jim Smith, Tuan Tran, Swami Venkataraman, Rich Weber, Eric Williams, Mike Woodacre, and Steve Yurash.

References

[1] Anant Agarwal, Ricardo Bianchini, et. al. The MIT Alewife machine: Architecture and Performance. In *Proceedings of the 22nd Annual International Symposium on Computer Architecture*, pages 2-13, June 1995.

[2] Mike Galles. Scalable Pipelined Interconnect for Distributed Endpoint Routing: The SGI SPIDER chip. In *Hot Interconnects '96.*

[3] Daniel Lenoski, James Laudon, Kourosh Gharachorloo, Anoop Gupta, and John Hennessy. The directory-based cache coherence protocol for the DASH multiprocessor. In *Proceedings of the 17th Annual International Symposium on Computer Architecture*, pages 148-159, May 1990.

[4] Daniel Lenoski, James Laudon, Truman Joe, David Nakahira, Luis Stevens, Anoop Gupta, and John Hennessy. The DASH prototype: Logic overhead and performance. *IEEE Transactions on Parallel and Distributed Systems*, 4(1):41-61, January 1993.

[5] Mendel Rosenblum, John Chapin, Dan Teodosiu, Scott Devine, Tirthankar Lahiri, and Anoop Gupta, Implementing efficient fault containment for multiprocessors. *Communications of the ACM*, 39(3):52-61, September, 1996.

[6] Wolf-Dietrich Weber, *Scalable Directories for Cache-Coherent Shared-Memory Multiprocessors*. Ph.D.thesis, Stanford University, Stanford, California, January 1993.

[7] Steve Whitney, John McCalpin, Nawaf Bitar, John L. Richardson, and Luis Stevens, The SGI Origin Software Environment and Application Performance. To appear in COMPCON '97.