STARFIRE:
Extending the SMP Envelope

Alan Charlesworth

Sun Microsystems

Point-to-point routers and an active centerplane with four address routers are key components of today's largest uniform-memory-access symmetric multiprocessor system.

New and faster processors get all the glory, but the interconnect "glue" that cements the processors together should get the headlines. Without high-bandwidth and low-latency interconnects, today's high-powered servers wouldn't exist.

The current mainstream server architecture is the cache-coherent symmetric multiprocessor (SMP), illustrated in Figure 1. The technology was developed first on mainframes and then moved to minicomputers. Sun Microsystems and other server vendors brought it to microprocessor-based systems in the early 1990s.

In an SMP, one or more levels of cache for each processor retain recently accessed data that can be quickly reused, thereby avoiding contention for further memory accesses. When a processor can't find needed data in its cache, it sends out the address of the memory location it wants to read onto the system bus. All processors check the address to see if they have a more up-to-date copy in their caches. If another processor has modified the data, it tells the requester to get the data from it rather than from memory. For a processor to modify a memory location, it must gain ownership of it. Upon modification, any other processor that has a cached copy of the data being modified must invalidate its copy.

This checking of the address stream on a system bus is called snooping, and a protocol called cache coherency keeps track of where everything is. For more background on cache-coherent multiprocessing systems, see Hennessy and Patterson (chapter 8).[1]

A central system bus provides the quickest snooping response and thus the most efficient system operation, especially if many different processors frequently modify shared data. Since all memory locations are equally accessible by all processors, there is no concern about trying to have applications optimally place data in one memory module or another. This feature of a traditional SMP system is called uniform memory access.

Keeping pace

A snoopy bus is a good solution only so long as it is not overloaded by too many processors making too many memory requests. Processor speeds have been increasing by 55% per year.[1] With increased levels of component integration, we can squeeze more processors into a single cabinet. Sun has pushed the envelope of SMP systems through three generations of snoopy-bus-based uniform-memory-access interconnects: MBus[2] and XDBus,[2] and the

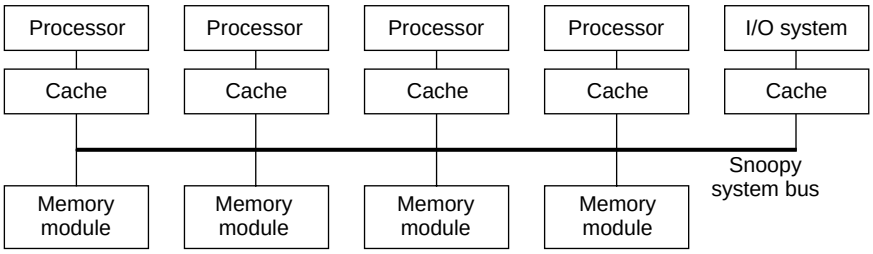

Figure 1. Symmetric multiprocessor.

0272-1732/98/$10.00 © 1998 IEEE

Ultra Port Architecture.[3] Table 1 shows the characteristics of these architectures.

Our system architects have made the following cumulative improvements in bus capacity:

- Increased bus clock rates from 40 MHz to 100 MHz by using faster bus-driving logic.[4]
- Changed from a circuit-switched protocol to a packet-switched protocol. In a circuit-switched organization, each processor's bus request must complete before the next can begin. Packet switching separates the requests from the replies, letting bus transactions from several processors overlap.
- Separated the addresses and data onto distinct wires so that addresses and data no longer have to compete with each other for transmission.
- Interleaved multiple snoop buses. Using four address buses allows four addresses to be snooped in parallel. The physical memory space is divided into quarters, and each address bus snoops a different quarter of the memory.
- Doubled the cache block size from 32 bytes to 64. Since each cache block requires a snoop, doubling the cache line width allows twice as much data bandwidth for a given snoop rate.
- Widened the data wires from 8 bytes to 16 bytes to move twice as much data per clock.

The combined effect of these improvements has been to increase bus snooping rates from 2.5 million snoops per second on our 600MP in 1991 to 167 million snoops per second on the Starfire in 1997—a 66-fold increase in six years. Combined with a two-times wider cache line, this has allowed data bandwidths to increase by 133 times.

Ultra Port Architecture interconnect

All current Sun workstations and servers use Sun's Ultra Port Architecture.[3] The UPA provides writeback MOESI (exclusive modified, shared modified, exclusive clean, shared clean, and invalid) coherency on 64-byte-wide cache blocks. The UPA uses packet-switched transactions with separate address and 18-byte-wide data lines, including two bytes of error-correcting code (ECC).

We have developed small, medium, and large implementations of the Ultra Port Architecture, as shown in Figure 2, to optimize for different parts of the price spectrum. The

Table 1. Sun interconnect generations.

Architecture	MBus 1990	XDBus 1993	Ultra Port Architecture 1996
Bus improvements			
Bus clock	40 MHz	40-55 MHz	83.3-100 MHz
Bus protocol	Circuit switched	Packet switched	Packet switched
Address and data information	Multiplexed on same wires	Multiplexed on same wires	Separate wires
Maximum number of interleaved buses	1	4	4
Cache block size	32 bytes	64 bytes	64 bytes
Data port width	8 bytes	8 bytes	16 bytes
Maximum interconnect performance			
Snoops/bus/clock	1/16	1/11	1/2
Maximum snooping rate	2.5 million/s	20 million/s	167 million/s
Corresponding maximum data bandwidth	80 MBps	1,280 MBps	10,667 MBps

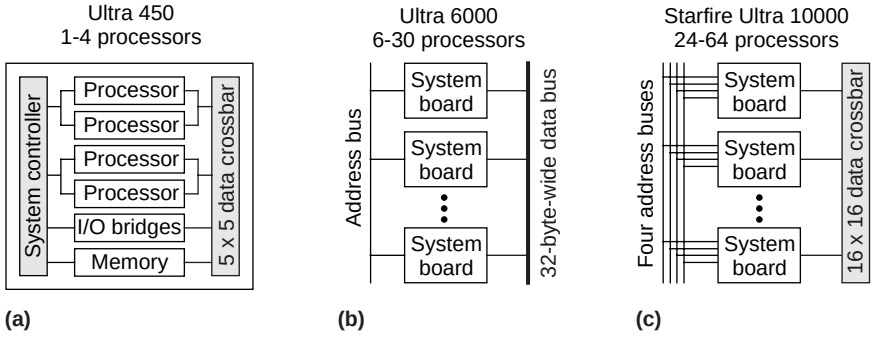

Figure 2. Three Ultra Port Architecture implementations: (a) a small system consisting of a single board with four processors, I/O interfaces, and memory; (b) a medium-sized system with one address bus and a wide data bus between boards; and (c) a large system with four address buses and a data crossbar between boards.

small system's centralized coherency controller and small data crossbar provide the lowest possible cost and memory latency within the limited expansion needs of a single-board system. The medium-sized system's Gigaplane bus[4] provides a broad range of expandability and the lowest possible memory latency. In the large system, Starfire's multiple-address broadcast routers and data crossbar extend the UPA family's bandwidth by four times and provide the unique ability to dynamically repartition and hot swap the system boards.

Starfire design choices

In the fall of 1993, we set out to implement the largest centerplane-connected, Ultra Port Architecture-based system that could be squeezed into one large cabinet. Our goals were to

- increase system address and data bandwidth by four times over the medium-sized Ultra 6000 system,
- provide a new dimension of Unix server flexibility with Dynamic System Domains, and
- improve system reliability, availability, and serviceability.

Our team had already implemented three previous generation enterprise servers. When we started designing the Starfire, our top-of-the-line product was the 64-processor CS6400, which used the SuperSparc processor and the XDBus interconnect. Since the scale of the CS6400 worked well, we decided to carry over many of its concepts to the new UltraSparc/UPA technology generation.

We made the following design choices:

- *Four-way interleaved address buses for the necessary snooping bandwidth.* This approach had worked well on our 64-processor XDBus-generation system. Each address bus covers 1/4 of the physical address space. The buses snoop on every other cycle and update the duplicate tags in alternate cycles. At an 83.3-MHz system clock, Starfire's coherency rate is 167 million snoops per second. Multiplied by the Ultra Port Architecture's 64-byte cache line width, this is enough for a 10,667-megabyte-per-second (MBps) data rate.
- *A 16 × 16 data crossbar.* To match the snooping rate, we chose a 16 × 16 interboard data crossbar having the same 18-byte width as the UPA data bus. Figure 3 shows how the snooping and data bandwidths relate as the system is expanded. Since the snooping rate is a constant two snoops per clock, while the data crossbar capacity expands as boards are added, there is only one point of exact balance, at about 13 boards. For 12 and fewer boards, the data crossbar governs the interconnect capacity; for 14 to 16 boards, the snoop rate sets the ceiling.
- *Point-to-point routing.* We wanted to keep failures on one system board from affecting other system boards, and we wanted the capability to dynamically partition the system. To electrically isolate the boards, we used point-to-point router ASICs (application-specific integrated circuits) for the entire interconnect—data, arbitration, and the four address buses. Also, across a large cabinet, point-to-point wires can be clocked faster than bused signals.
- *An active centerplane.* The natural place to mount the router ASICs was on the centerplane, which is physically and electrically in the middle of the system.
- *A system service processor (SSP).* On our previous system it was very useful to have a known-good system that was physically separate from the server. We connected the SSP via Ethernet to Starfire's control boards, where it has access to internal ASIC status information.

Starfire interconnect

Like most multiboard systems, Starfire has a two-level interconnect. The on-board interconnect conveys traffic from the processors, SBus cards, and memory to the off-board address and data ports. The centerplane interconnect transfers addresses and data between the boards.

Memory accesses always traverse the global interconnect, even if the requested memory location is physically on the same board. Addresses must be sent off board anyway to accomplish global snooping. Data transfers are highly pipelined, and local shortcuts to save a few cycles would have unduly complicated the design. As with the rest of the Ultra server family, Starfire's uniform-memory-access time is independent of the board where memory is located.

Address interconnect. Table 2 (next page) characterizes the address interconnect. Address transactions take two cycles. The two low-order cache-block address bits determine which address bus to use.

Data interconnect. Table 3 characterizes the data interconnect. Data packets take four cycles. In the case of a load-miss, the missed-upon 16 bytes are sent first. The Starfire

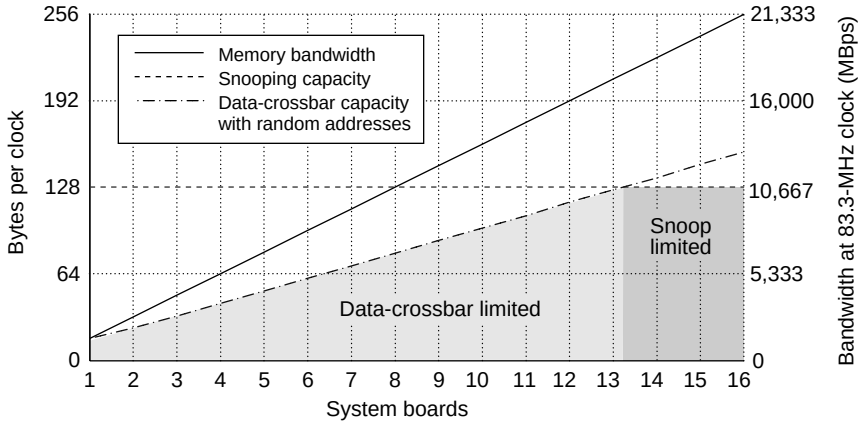

Figure 3. Snooping and data interconnect capacity.

Table 2. Address interconnect.

Unit	ASIC type	Purpose	ASICs per system board	ASICs on centerplane
Port controller	PC	Controls two UPA address-bus ports	3	0
Coherency interface controller	CIC	Maintains duplicate tags to snoop local caches	4	0
Memory controller	MC	Controls four DRAM banks	1	0
UPA to SBus	SYSIO	Bridges UPA to SBus	2	0
Local address arbiter	LAARB mode of XARB	Arbitrates local address requests	1	0
Global address arbiter	GAARB mode of XARB	Arbitrates global requests for an address bus	0	4
Global address bus	GAB mode of 4 XMUXes	Connects a CIC on every board	0	16

Table 3. Data interconnect.

Unit	ASIC type	Purpose	ASICs per system board	ASICs on centerplane
UltraSparc data buffer	UDB	Buffers data from the processor; generates and checks ECC	8	0
Pack/unpack	Pack mode of 2 XMUXes	Assembles and disassembles data into 72-byte memory blocks	4	0
Data buffer	DB	Buffers data from two UPA data-bus ports	4	0
Local data arbiter	LDARB mode of XARB	Arbitrates on-board data requests	1	0
Local data router	LDR mode of 4 XMUXes	Connects four Starfire data buffers to a crossbar port	4	0
Global data arbiter	GDARB	Arbitrates requests for the data crossbar	0	2
Global data router	GDR mode of 12 XMUXes	$16 \times 16 \times 18$-byte crossbar between the boards	0	12

data buffer ASICs provide temporary storage for packets that are waiting their turn to be moved across the centerplane. The local and global routers are not buffered, and transfers take a fixed eight clocks from the data buffer on the sending board to the data buffer on the receiving board.

Interconnect operation. An example of a load-miss to memory illustrates Starfire's interconnect operation. The interconnect diagram in Figure 4 shows the steps listed in Table 4 (on page 44).

Buses versus point-to-point routers. Starfire's pin-to-pin latency for a load-miss is 38 clocks (468 nanoseconds), counting from the cycle when the address request leaves the processor through the cycle when data arrives at the processor. The medium-sized Ultra 6000's bus takes only 18 clocks (216 ns).

Buses have lower latencies than routers: a bus takes only 1 clock to move information from one system component to another. A router, on the other hand, takes 3 clocks to move information: a cycle on the wires to the router, a cycle inside the routing chip, and another cycle on the wires to the receiving chip. Buses are the preferred interconnect topology for small and medium-sized systems.

Ultra 10000 designers used routers with point-to-point interconnects to emphasize bandwidth, partitioning, and reliability, availability, and serviceability. Ultra 6000 designers used a bus to optimize for low latency and economy over a broad product range.

Starfire packaging

Starfire's cabinet is 70 inches tall × 34 inches wide × 46 inches deep. Inside are two rows of eight system boards mounted on either side of a centerplane. Starfire is our fourth generation of centerplane-based systems.

Besides the card cage, power supply, and cooling system, the cabinet has room for three disk trays. The remaining peripherals are housed separately in standard Sun peripheral racks.

Starfire performs two levels of power conversion. Up to eight $N + 1$ redundant bulk supplies convert from 220 Vac to 48 Vdc, which is then distributed to each board. On-board supplies convert from 48 Vdc to 3.3 and 5 Vdc. Having local power supplies facilitates the hot swap of system boards.

Starfire uses 12 hot-pluggable fan trays, half above and half below the card cage. Fan speed is automatically controlled to reduce noise in normal environmental conditions.

Centerplane. The centerplane holds the 20 address ASICs and 14 data ASICs that route information between the 16 system-board sockets. It is 27 inches wide × 18 inches tall × 141 mils thick, with 14 signal layers and 14 power layers. The net density utilization is nearly 100%. We routed approximately 95% of the 14,000 nets by hand. There are approximately 2 miles of wire etch and 43,000 holes.

Board spacing is 3 inches to allow enough airflow to cool the four 45-watt processor modules on each system board. Signal lengths had to be minimized to run a 10-ns system clock across 16 boards. The maximum etched wire length is

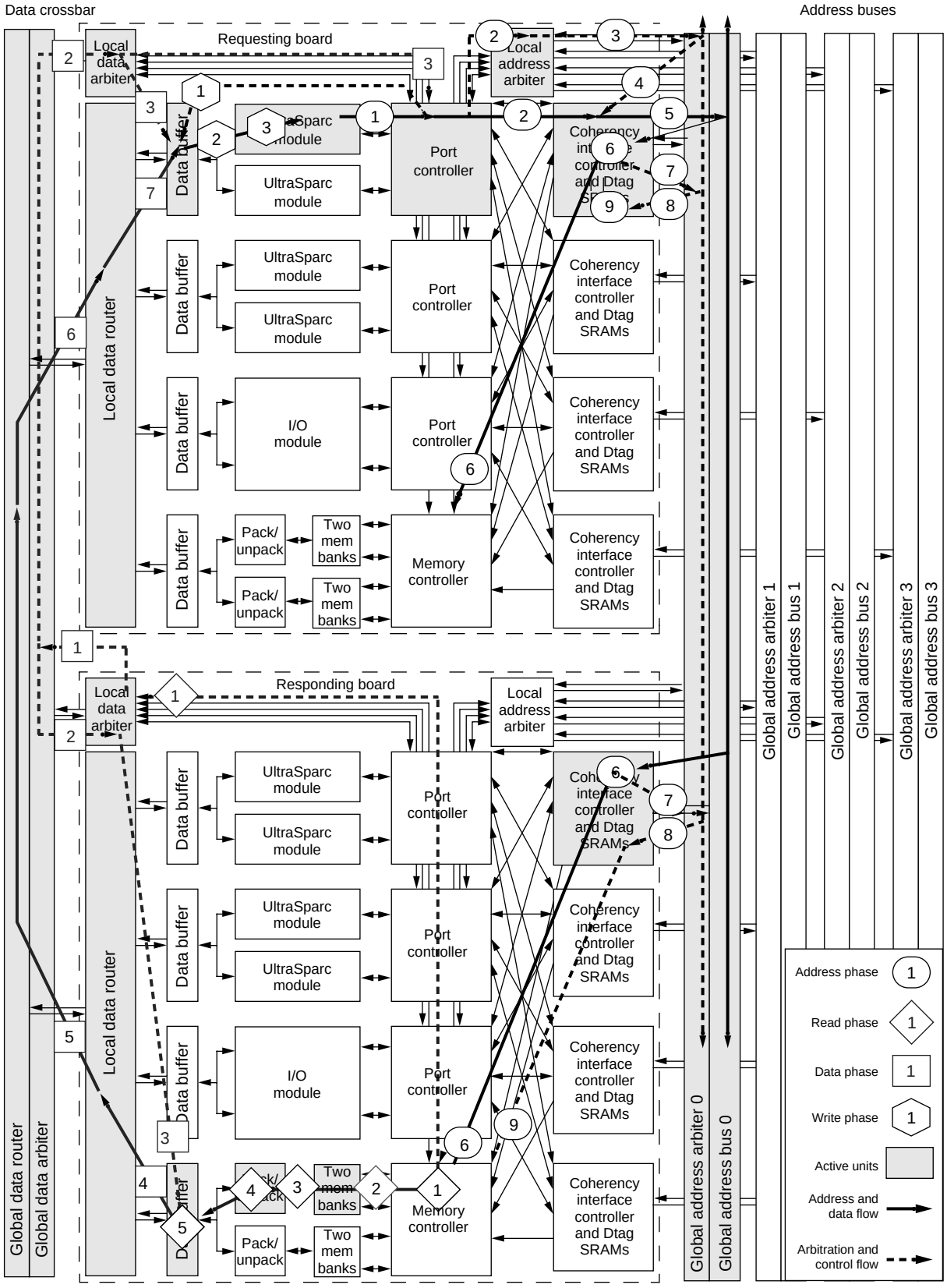

Figure 4. Interconnect steps for a load-miss from memory.

Table 4. Interconnect sequence for a load-miss to memory.

Phase	Steps	Clocks
Send address and establish coherency	1. Processor makes a request to its port controller. 2. Port controller sends the address to a coherency interface controller, and sends the request to the local address arbiter. 3. Local address arbiter requests a global address bus cycle. 4. Global address arbiter grants an address bus cycle. 5. Coherency interface controller sends the address through the global address bus to the rest of the coherency interface controllers on the other boards. 6. All coherency interface controllers relay the address to their memory controllers and snoop the address in their duplicate tags. 7. All coherency interface controllers send their snoop results to the global address arbiter. 8. Global address arbiter broadcasts the global snoop result. 9. Memory is not aborted by its coherency interface controller because the snoop did not hit.	13
Read from memory	1. Memory controller recognizes that this address is for one of its memory banks. 2. Memory controller orchestrates a DRAM cycle and requests a data transfer from its local data arbiter. 3. Memory sends 72 bytes of data to the unpack unit. 4. Unpack splits the data into four 18-byte pieces. 5. Unpack sends data to the data buffer to be buffered for transfer.	13
Transfer data	1. Local data arbiter requests a data transfer. 2. Global data arbiter grants a data transfer and notifies the receiving local data arbiter that data is coming. 3. Sending local data arbiter tells the data buffer to begin the transfer. 4. Sending data buffer sends data to the local data router. 5. Data moves through the local data router to the centerplane crossbar. 6. Data moves through the centerplane crossbar to the receiving board's local data router. 7. Data moves through the receiving local data router to the receiver's data buffer.	8
Write data	1. Port controller tells the data buffer to send the data packet. 2. Data buffer sends data to the UltraSparc data buffer on the processor module. 3. UltraSparc data buffer sends data to the processor.	4

approximately 20 inches. After extensive cross-talk analysis, we developed and implemented a novel method for absolute–cross-talk minimization on long, minimally coupled lines.

We distributed clock sources through the centerplane to each system board ASIC. Routing all traces on identical topologies minimizes skew between clock arrivals at the ASICs.

We used unidirectional, point-to-point, source-terminated CMOS to implement the 144-bit-wide 16 × 16 data crossbar and the four 48-bit-wide address-broadcast routers. We designed and tested the system to run at 100 MHz at worst-case temperatures and voltages. However, the UltraSparc-II processor constrains the system clock to be 1/3 or 1/4 of the processor clock. As of this writing, the processor clock is 250 MHz, so the system clock is 250/3 = 83.3 MHz.

System boards. The system boards, shown in Figure 5 (next page), each hold six mezzanine modules on the top side: the memory module, the I/O module, and four processor modules. The bottom side has nine address ASICs, nine data ASICs, and five 48-volt power converters. The boards are 16 × 20 inches with 24 layers.

Processor module. Starfire uses the same UltraSparc processor module as the rest of Sun's departmental and data center servers. As of this writing, the server modules have a 250-MHz processor with 4 Mbytes of external cache.

Memory module. The memory module contains four 576-bit-wide banks of memory composed of 32 standard 168-pin ECC DIMMs (dual in-line memory modules). It also has four ASICs, labeled Pk, which pack and unpack 576-bit-wide memory words into 144-bit-wide data-crossbar blocks. The memory module contains 4 Gbytes of memory using 64-Mbit DRAMs.

I/O module. The current I/O module interfaces between the UPA and two SBuses and provides four SBus card slots. Each SBus has an achievable bandwidth of 100 MBps.

ASIC types. We designed seven unique ASIC types. Six of them implement an entire functional unit on a single chip, while the seventh is a multiplexer part used to implement the local and global routers, as well as the pack/unpack function. The ASICs are fabricated in 3.3-V, 0.5-micron CMOS technology. The largest die is 9.95 × 10 mm, with five metal layers. The ASICs are all packaged in 32 × 32-mm ceramic

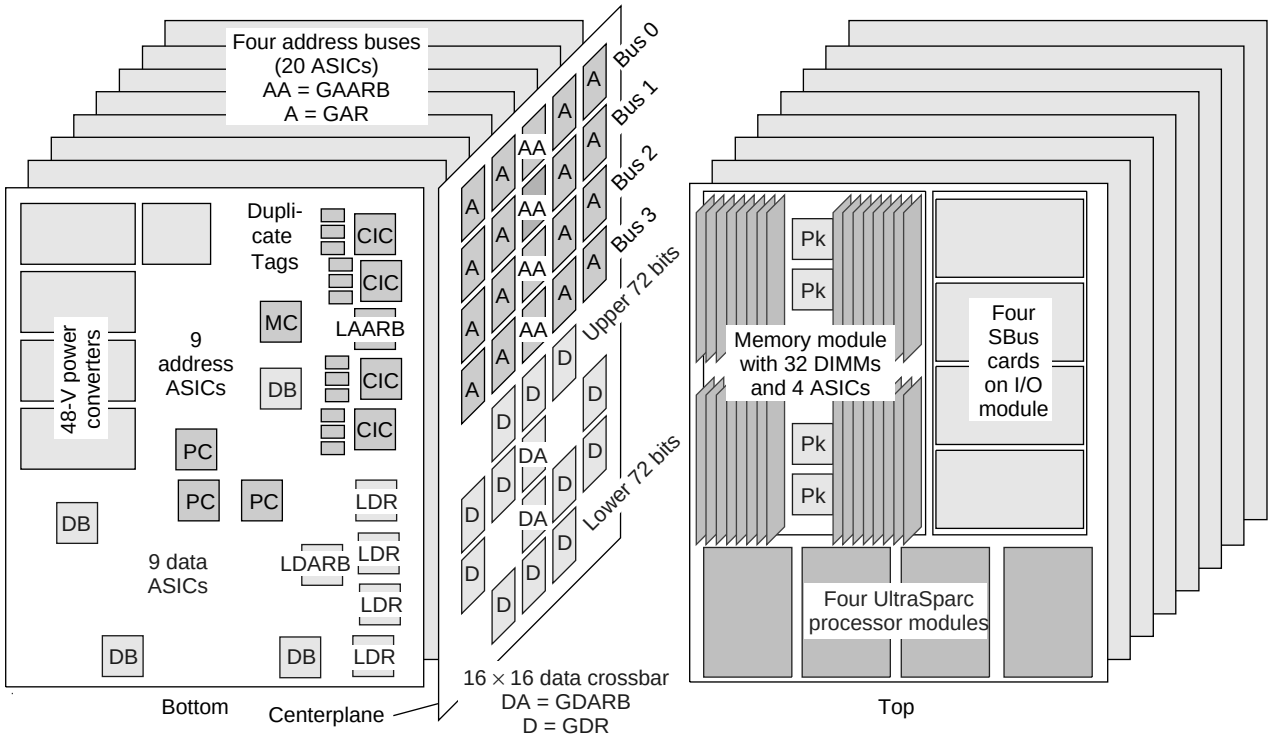

Figure 5. Bottom (left) and top (right) of the system boards, with the centerplane shown in the middle.

ball grid arrays with 624 pins.

For more details on Starfire's physical implementation, see Charlesworth et al.[5]

Interconnect reliability. In addition to the ECC for data that is generated and checked by the processor module, Starfire ASICs also generate and check ECC for address packets. To help isolate faults, the Starfire data-buffer chips check data-packet ECC along the way through the interconnect.

Failed components. If an UltraSparc module, DIMM, SBus board, memory module, I/O module, system board, control board, centerplane support board, power supply, or fan fails, the system tries to recover without service interruption. Later, the failed component can be hot swapped out of the system and replaced.

Redundant components. Customers can optionally configure a Starfire to have 100% hardware redundancy of configurable components: control boards, support boards, system boards, disk storage, bulk power subsystems, bulk power supplies, cooling fans, peripheral controllers, and system service processors. If the centerplane experiences a failure, it can operate in a degraded mode. If one of the four address buses fails, the remaining buses will allow access to all system resources. The data crossbar is divided into separate halves, so it can operate at half-bandwidth if an ASIC fails.

Crash recovery. A fully redundant system can always recover from a system crash, utilizing standby components or operating in degraded mode. Automatic system recovery enables the system to reboot immediately following a failure, automatically disabling the failed component. This approach prevents a faulty hardware component from causing the system to crash again or from keeping the entire system down.

Dynamic System Domains

Dynamic System Domains make Starfire unique among Unix servers. Starfire can be dynamically subdivided into multiple computers, each consisting of one or more system boards. System domains are similar to partitions on a mainframe. Each domain is a separate shared-memory SMP system that runs its own local copy of Solaris and has its own disk storage and network connections. Because individual system domains are logically isolated from other system domains, hardware and software errors are confined to their respective domain and do not affect the rest of the system. Thus, a system domain can be used to test device drivers, updates to Solaris, or new application software without impacting production usage.

Dynamic System Domains can serve many purposes, enabling the site to manage the Starfire resources effectively:

- *LAN consolidation*. A single Starfire can replace two or more smaller servers. It is easier to administer because it uses a single system service processor (SSP), and it is more robust because it has better reliability, availability, and serviceability features. Starfire offers the flexibility to shift resources quickly from one "server" to another. This is beneficial as applications grow, or when demand reaches peak levels and requires rapid reassignment of computing resources.
- *Development, production, and test environments.* In a production environment, most sites require separate

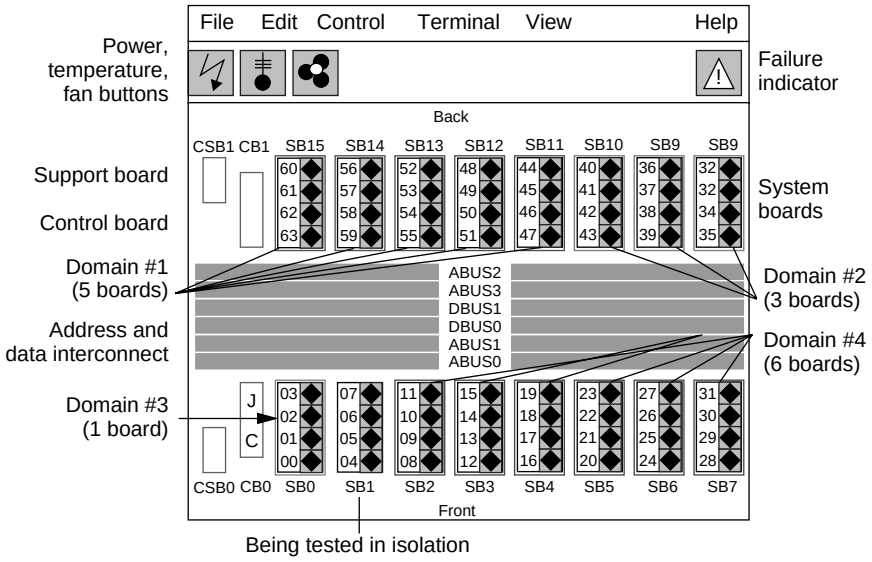

Figure 6. System service processor Hostview main screen.

development and test facilities. With Starfire, those functions can safely coexist in the same box. Having isolated facilities enables development work to continue on a regular schedule without impacting production.

- *Software migration*. Dynamic System Domains may be used as a way to migrate systems or application software to updated versions. This applies to the Solaris operating system, database applications, new administrative environments, and applications.
- *Special I/O or network functions.* A system domain can be established to deal with specific I/O devices or functions. For example, a high-end tape device could be attached to a dedicated system domain, which is alternately merged into other system domains that need to use the device for backup or other purposes.
- *Departmental systems.* Multiple projects or departments can share a single Starfire system, simplifying cost-justification and cost-accounting requirements.

Many domain schemes are possible with Starfire's 64 processors and 16 boards. For example, we could make domain 1 a 12-board (48-processor) production domain running the current release of Solaris. Domain 2 could be a two-board (8-processor) domain for checking out an early version of the next Solaris release. Domain 3 could be a two-board (8-processor) domain running a special application—for instance, proving that the application is fully stable before allowing it to run in the production domain. Each domain has its own boot disk and storage, as well as its own network connection.

Domain administration. System administrators can dynamically switch system boards between domains or remove them from active domains for upgrade or servicing. After service, boards can be reintroduced into one of the active domains, all without interrupting system operation. Each system domain is administered from the SSP, which services all the domains.

The system service processor is a Sparc workstation that runs standard Solaris plus a suite of diagnostics and management programs. It is connected via Ethernet to a Starfire control board. The control board has an embedded control processor that interprets the TCP/IP Ethernet traffic and converts it to JTAG control information. Figure 6 shows an example of Starfire's hardware and domain status in the Hostview main screen. In this instance there are four domains, plus an additional board being tested in isolation.

Domain implementation. Domain protection is implemented at two levels: in the centerplane arbiters and in the coherency interface controllers on each board.

Centerplane filtering. Global arbiters provide the top-level separation between unrelated domains. Each global arbiter contains a 16×16-bit set of domain control registers. For each system board there is a register that, when the bits are set to one, establishes the set of system boards in a particular board's domain group.

Board-level filtering. Board-level filtering lets a *group* of domains view a region of each other's memory to facilitate interdomain networking—a fast form of communication between a group of domains. As Figure 7 illustrates, all four coherency interface controllers on a system board have identical copies of the following registers:

- *Domain mask*. Sixteen bits identify which other system boards are in the board's domain.
- *Group memory mask*. Sixteen bits identify which other boards are in a board's domain group, to facilitate memory-based networking between domains.
- *Group memory base and limit registers.* These registers contain the lower and upper physical addresses of the board's memory that are visible to other domains in a group of domains. The granularity of these addresses is 64 Kbytes.

Dynamic reconfiguration. With dynamic reconfiguration, system administrators can logically move boards between running domains on-the-fly. The process has two phases: attach and detach.

Attach. This phase connects a system board to a domain and makes it possible to perform online upgrades, redistribute system resources for load balancing, or reintroduce a board after it has been repaired. Attach diagnoses and configures the candidate system board so that it can be introduced safely into the running Solaris operating system. There are two steps:

1. The board is added to the target domain's board list in the domain configuration files on the SSP. Power-on self-test

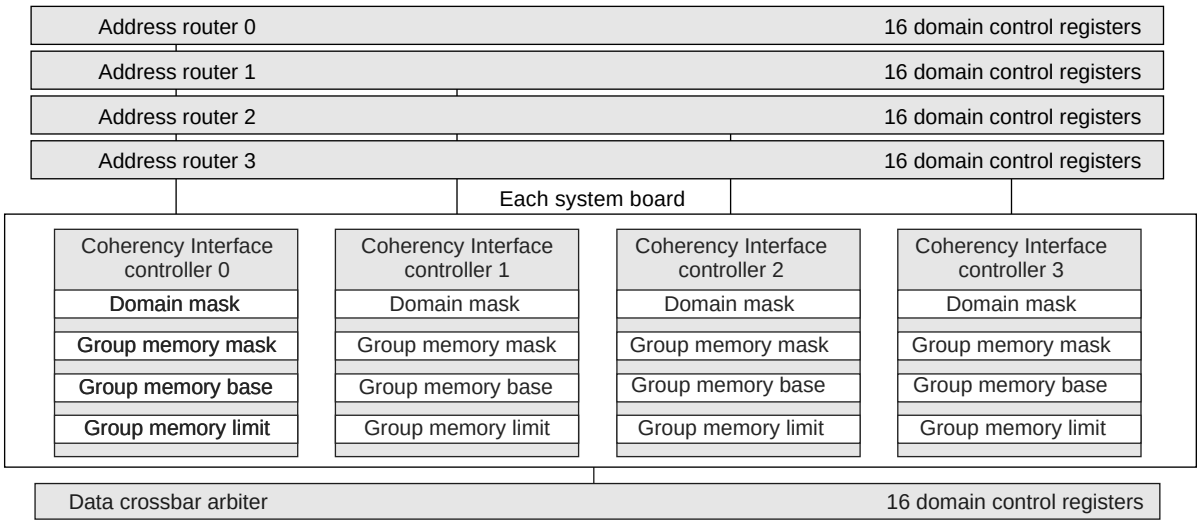

Figure 7. Domain registers in the Starfire interconnect.

(POST) executes, testing and configuring the board. POST also creates a single-board domain group that isolates the candidate board from the centerplane's other system boards. The processors shift from a reset state into a spin mode, preparing them for code execution. The centerplane and board-level domain registers are configured to include the candidate board in the target domain.

2. When these operations are complete, the Solaris kernel is presented with the board configuration. Though aware of the configuration, Solaris has not yet enabled it. At this juncture, the operator receives status information and can either complete the attach or abort it. After the operator authorizes completion, Solaris performs the final steps needed to start the processors, adds the memory to the available page pool, and connects any on-board I/O devices or network connections. The operator is notified when the candidate board is actively running the workload in the domain.

Detach. System boards are detached to reallocate them to another domain or to remove them for upgrade or repair. Process execution, network and I/O connections, and the contents of memory must be migrated to other system boards. Detach also has two steps:

1. The OS flushes all pageable memory to disk and remaps kernel memory to other boards. Free pages are locked to prevent further use. As detach proceeds, the OS switches network devices and file systems to alternate paths to other boards. Finally, processors are taken offline.
2. The operator can still abort the board's removal from the system so that the board can continue operation. Unless aborted, the detach is committed and the system board becomes available for attachment to another domain or for physical removal from the system.

Hot swap. Once detached, a board can be powered down and physically removed from a running system by a certified service provider. Conversely, a new, repaired, or upgraded board can be physically inserted into a running system and powered-on in preparation for doing an attach.

System price/performance

The Transaction Processing Council's (TPC) benchmarks are unique in that they measure both price and performance of computing systems. This lets us judge how well our uniform-memory-access SMP systems compare with alternative multiprocessor architectures.

TPC-D benchmark. The TPC Benchmark D (TPC-D) models a decision support environment in which complex ad hoc business-oriented queries access large portions of a database. The queries typically involve one or more of the following characteristics: multitable joins, extensive sorting, grouping and aggregation, and sequential scans.

Decision support applications typically consist of long and often complex read-only queries. Decision support database updates are relatively infrequent. The databases need not contain real-time or up-to-the-minute information, since decision support applications tend to process large amounts of data that usually would not be affected significantly by individual transactions.

TPC-D comes in sizes of 1 Gbyte, 30 Gbytes, 100 Gbytes, 300 Gbytes, and 1 Tbyte. We discuss the 300-Gbyte size because results are available for several large systems. This TPC-D size requires about 16 Gbytes of memory and around 500 disk drives totaling about 2 Tbytes of storage. TPC-D has three metrics:

- The power metric (QppD@Size) is based on a geometric mean of the 17 TPC-D queries, the insert test, and the delete test. It measures the system's ability to give a single user the best possible response time by harnessing all available resources.
- The throughput metric (QthD@Size) characterizes the

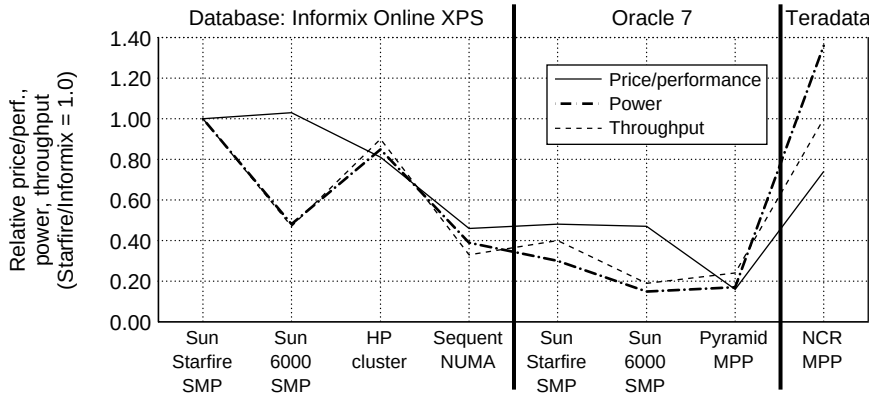

Figure 8. TPC-D (decision support) results for 300-Gbyte data size.

system's ability to support a multiuser workload in a balanced way. The test sponsor decides how many users to choose, and each executes the full TPC-D set of 17 queries in a different order. In the background, an update stream runs a series of insert/delete operations (one pair for each query user).

- The price/performance metric (Price-per-QphD@Size) is the ratio obtained by dividing the five-year system price by the composite query-per-hour rating, QphD@Size. This rating is equal to the geometric mean of the power and throughput metrics.

TPC-D 300-Gbyte results. Figure 8 and Table 5 show the TPC-D 300-Gbyte results as of this writing.[6] To examine the performance of different system architectures running the same database software, we sort the results by the two open-database vendors, since different databases offer different query optimizations.

The Starfire Ultra 10000 has the best Informix power and throughput and is second to the Ultra 6000 in price/performance. The Ultra 10000 leads in all three categories for Oracle. The proprietary Teradata database achieved the highest power rating of any solution, but fell short of the Starfire/Oracle results on throughput and price performance. Sun's results show that large SMP systems can match or exceed the best performance of other parallel architectures—nonuniform memory access, massively parallel processing, and clusters—and can do so for a lower system cost.

Interconnect price. We can use the TPC-D system price data to compare the hardware prices of these systems and determine whether the benefits of SMP interconnects come with a disproportionate price. Figure 9 shows a price breakdown for the systems in the TPC-D 300 benchmark test. The Starfire computing hardware's price is about 35% of its total five-year price, which is a bit below the average for all the systems.

We can examine more closely the price impact of Starfire's centerplane-crossbar interconnect by comparing it to the bus-based Ultra 6000. (The other TPC-D vendors did not break down their prices in sufficient detail to isolate processor and memory prices.) Figure 10 shows the price leverage for the interconnect "glue," which we obtain by separating the costs of the commodity processors and memory.

Starfire's glue fraction for a crossbar interconnect is 16% of its total price, compared with 8% for the Ultra 6000's low-cost bus interconnect. If it were possible to make a 64-processor bus-based system, and thereby cut the glue cost in half, it would reduce the total five-year system price by only 8%. For large systems like these, there is no point in saving money on the interconnect, since the total price is dominated by commodity processors, memory, disk, and database software.

Other benchmarks. The Starfire system has done well with workloads other than decision support. It has set online transaction processing records running SAP R/3 and BAAN benchmarks. A cluster of four Starfires has sustained over 100 Gflops (floating-point operations per second) on the Linpack parallel equation-solving benchmark, which is currently

Table 5. Processor, node, and price breakdown for systems compared in the TCD-D 300-Gbyte performance evaluation.

	Informix OnLine XPS				Oracle 7			Teradata
	Sun Starfire SMP	Sun 6000 SMP	HP cluster	Sequent NUMA	Sun Starfire SMP	Sun 6000 SMP	Pyramid MPP	NCR MPP
Processors	64	24	64	32	64	24	96	96
Nodes	1	1	16	8	1	1	16	20
Node size (processors)	64	24	4	4	64	24	6	4
System price (millions)								
Computer H/W	$2.6	$0.9	$2.6	$2.2	$2.2	$0.9	$3.9	$4.2
Disk storage	$0.9	$0.9	$1.0	$2.0	$0.8	$0.9	$1.0	$2.3
H/W maintenance	$0.9	$0.4	$1.2	$0.9	$0.9	$0.4	$2.0	$1.8
Software	$3.0	$1.2	$0.9	$2.9	$1.4	$0.5	$2.7	$3.3
Total 5-year price	$7.4	$3.4	$5.7	$8.0	$5.3	$2.7	$9.6	$11.7

the highest for any general-purpose system.[7] As of this writing, a cluster of two Starfires leads the SPECrate_int95 integer-application throughput benchmark.[8]

WE FOUND THAT PROVIDING an active centerplane-crossbar interconnect is money well spent, since it leverages the processors, memory, disks, and software that dominate a system's cost. Starfire's router-based implementation of the Ultra Port Architecture has extended Sun's Ultra server family bandwidth by a factor of four times. We did, however, pay a two-times penalty in pin-to-pin latency compared with the Ultra 6000; this is more than we would have liked. In the next generation, we expect to reduce the large-server latency penalty.

The error isolation of Starfire's point-to-point wires makes possible the flexibility of Dynamic System Domains and improves system availability and serviceability. Starfire uses the same processor modules and dual in-line memory modules as the midrange server, but its higher bandwidth and domain enhancements require a unique system board. We will extend router and system-domain technology down to the next generation of midrange servers, allowing us to use more common components across the family spectrum.

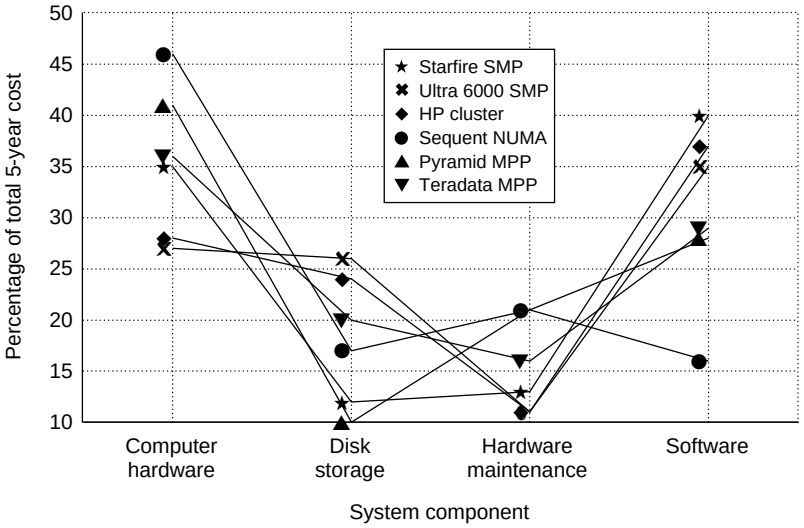

Figure 9. System price breakdown.

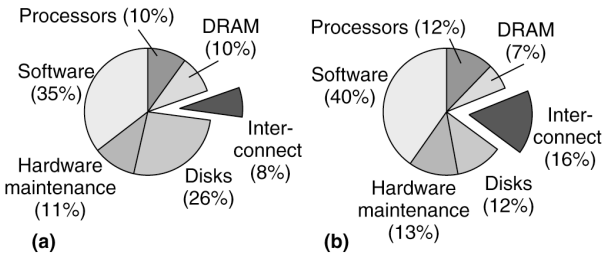

Figure 10. Interconnect cost percentages for (a) the Ultra 6000 with bus interconnect and (b) the Starfire Ultra 10000 with crossbar interconnect.

References

1. J. Hennessy and D. Patterson, *Computer Architecture: A Quantitative Approach*, 2nd ed., Morgan-Kaufman, San Mateo, Calif., 1996.
2. B. Catanzaro, *Multiprocessor System Architectures*, Prentice Hall, Englewood Cliffs, N.J., 1994.
3. K. Normoyle et al., "The UltraSPARC Port Architecture," *Proc. Hot Interconnects Symp. III*, 1995, available from author at kevin.normoyle@eng.sun.com.
4. A. Singhal et al., "Gigaplane: A High Performance Bus for Large SMPs," *Proc. Hot Interconnects Symp. IV*, 1996, available from author at ashok.singhal@eng.sun.com.
5. A. Charlesworth et al., "Gigaplane-XB: Extending the Ultra Enterprise Family," *Proc. Hot Interconnects Symp. V*, 1997, http://HTTP.CS.Berkeley.EDU/ culler/hoti97/E10000.ps.
6. Transaction Processing Performance Council, *TPC Benchmark Results*; see http://www.tpc.org/bench.results.html for current results.
7. J.J. Dongarra, "Performance of Various Computers Using Standard Linear Equations Software" (see http://performance.netlib.org/performance/html/PERFORM.ps for current report).
8. The Standard Performance Evaluation Corp., *SPEC CPU95 Results* (see http://www.specbench.org/osg/cpu95/results/ for current results).

Alan Charlesworth is a staff engineer with the Data Center and High Performance Computing Products Group of Sun Microsystems. He has worked in high-performance computing since developing software pipelining for the long-instruction-word AP-120B array processor from Floating Point Systems. He has helped develop high-bandwidth 64-processor Unix servers: the Cray Research CS6400 and Sun Starfire 10000. He attended Stanford University.

Address questions concerning this article to Alan Charlesworth, Sun Microsystems, 8300 SW Creekside Pl., Beaverton, OR 97008-7101; alanc@west.sun.com.