Volume Multi-Processor Systems: Part 2

By Chris Rijk – May 2002

Introduction

Welcome to Volume Multi-Processor Systems: Part 2. In the first part, we covered uniprocessor performance, workstation and server sizing, and CPU and system design considerations. For this part, we'll be looking at the specific architectural implementations of several volume multi-processor systems, including those based around the Pentium III, Pentium 4, Athlon, PowerPC, Itanium, and UltraSPARC architectures. Additionally, we'll also investigate into many developments in store for 2003, including integrated northbridges/memory controllers and improvements in thread-level parallelism (TLP) through on-chip multiprocessing (CMP) and fine-grained multithreading.

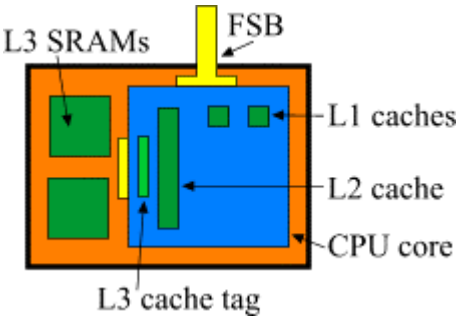
The primary goal of this article is to look into the volume workstation and server markets to determine what the implications are for the hardware design and usage, for software and the marketplace in general, concentrating on multi-processor systems with 2-8 CPUs. Though various aspects of system design, performance and cost are looked into, this article is not intended to be a buyers' guide. Some basic working knowledge of CPU and system design is assumed, with the target audience being computer systems professionals and computer enthusiasts.

Mostly to help reduce the scope of the article, the main focus is on CPUs and systems designed to be at least reasonably cost effective and for decent volume. This certainly includes AMD, Apple (with PowerPCs) and Intel, who all have reasonably low-cost high volume desktop CPUs that are also used in multi-processor workstations and servers. Intel also has some CPU designs for multi-processor systems only, and AMD is planning to follow. UltraSPARC systems from Sun Microsystems have the highest volume of 64-bit systems and their recent 8-way systems compete well against similar Intel-based systems, and have some interesting lower-end designs in the pipeline.

In the future, we'll take a look at some of the specific architectural elements of upcoming multi-processor systems, including those designed around AMD's Opteron architecture (codenamed "Hammer"), as well as Intel's Itanium II. For the time being, however, we'll be looking at existing systems.

System Architectures Introduction

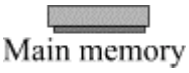
Key to Interpreting the Diagrams



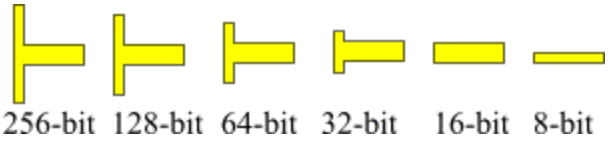
The diagram of a CPU to the left is an example from one of the following pages, and shows a CPU core (the blue box) with some of the features of the CPU core such as the on-die caches. Also shown are some SRAM chips for the external cache that comes in the CPU module (the brown box) for this particular processor, and a connection from the CPU to the rest of the system (the yellow upside-down "T"), often called the FSB - Front Side Bus.

The size of the features on the CPU die are only roughly accurate, though the general size of the CPU core is to a consistent scale - CPUs that in real life have a large core will have a larger size in the following diagrams. The exception is one or two of the system diagrams for the larger systems, which are done at half the scale.

Parts of the chipset are also shown using blue blocks (though with no brown surrounding box), and can be called ASICs - Application Specific Integrated Circuit.



Where the memory goes - in the actual diagrams the type of memory will be indicated. Generally the more memory icons shown, the more memory individual modules can be added, though it's not always clear what the practical maximum is for any given chipset.



Connections between CPUs, ASICs, SRAM and main memory are shown using thick yellow lines between the chips. Where the bars touch the chips, the connection is flattened out, roughly in proportion to the width of the connection. Normal connections have both signaling lines for the address and data as part of the same group, though often the address lines are separate to the data lines. Some connections have control and addressing completely separate to the data paths - connections for address (or control) signals only are shown colored red in the diagrams.

The most common types of connections are bi-directional buses and point-to-point connections. A bus connects between two or more chips, and extra pins are used so that the chips can decide who "owns" the bus - can send data over it. A point-to-point connection is basically like a bus that only ever connects between two chips, though the two chips using the connection need a similar mechanism to buses to decide who has access to it. With a bi-directional bus, which is what most buses and point-to-point connections use, data can be sent in either direction over the same lines, so while data is being transferred in one direction, none can go the other direction.

An alternative to a bi-directional connection is to have two uni-directional connections, one for each direction. This means the total bandwidth is the sum of both channels, but the maximum read bandwidth and maximum write bandwidth is just the value of one channel.

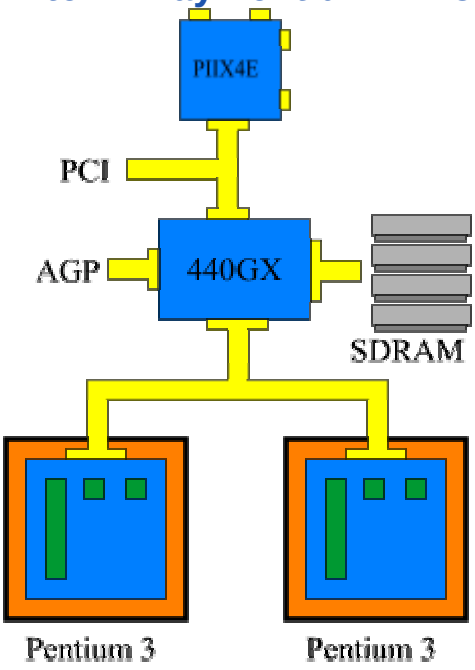
SDR and DDR signaling

A few years ago most connections used SDR (Single Data Rate) signaling, while some more recent ones use DDR (Double Data Rate) or QDR (Quad Data Rate). From the electrical point of view, it's mostly about using a smaller part of the signal to indicate a piece of data. From a bandwidth point of view, the peak transfer rate doubles, but basic latency stays the same, in terms of nanoseconds. An alternative way of viewing a 133MHz DDR connection (e.g. for PC2100 memory) is a 266MHz connection but with the same raw latency as PC133 memory. An alternative terminology that is sometimes used is "millions of transfers per second" - so 133MHz SDR would be 133MT/s and 133MHz DDR would be 266MT/s.

Other notes

1. For most of the following designs, the use of ECC (error correcting codes to correct single-bit errors or detect multi-bit errors) is not included, nor are parity checks (being able to detect single-bit errors). This is generally because most of the systems have this and the differences in design for reliability are fairly small. Though reliability is very important, going by a feature count isn't very useful for real world situations, a proper feature count isn't available anyway, and it's pretty much impossible to get hardware reliability statistics, so this aspect of the system design has been left out, for the most part.
2. The most common chipset design for PCs today is to have a "northbridge" and "southbridge" as the main ASICs in the design. The northbridge connects to the CPUs via the FSB, and also has the memory controller, AGP connection and another connection to the southbridge, which handles the rest of system I/O.
3. When talking about latency figures in the following pages, this is generally "load to use latency", though CPU and system documentation doesn't always define which type of latency they mean. Load-use latency is the time taken (generally expressed in nanoseconds) from when the CPU core starts the load to the time when it can use the data (load finished), so includes all cache effects.
4. Bandwidth figures are given for peak only, and for in one direction only - just reads or just writes. Sustained bandwidth figures are harder to come across, and can be up to about 50% lower for main memory bandwidth figures.
5. "Mission critical" applications often run from a machine room, which have expensive cooling systems, raised floors and so on. More recently, similar rooms became a common place to host web server systems, like the server for Ace's Hardware. Low power consumption helps reduce electricity costs and also cooling costs because of less heat. The costs of running such rooms are high, and the less space is taken up by a solution the better. Most machine rooms are filled with racks (about 2m high each) with space for about 42 1U systems - 1U is 1.75 inches.

Intel 2-way Pentium III Chipsets



Chipset architecture: The main 2-way Pentium III chipset is the 440GX which uses normal SDRAM, though there is also the 840 chipset which uses two Rambus channels (like the 850 Pentium 4 chipset), both of which support up to 4GByte of main memory. The architecture is standard northbridge and southbridge, with the two Pentium IIIs sharing a single bus.

Bandwidth and latency: The 840 chipset does in fact support 3.2GBytes/s of memory bandwidth (with the same dual Rambus connection as the 850 chipset) though there are very few systems on sale that use this - most Pentium III systems sold today use a chipset from ServerWorks which uses PC133 SDRAM, like the 440GX. Even with 3.2GBytes/s of memory bandwidth the CPUs would still be limited by the 1GByte/s FSB. I have not found any official latency figures for Pentium III systems, but some tests using software suggest a latency of around 130ns, which is about what would be expected for a regular northbridge design.

Scalability: With a bus-based architecture, the cache coherency protocol is quite simple, has good latency and should scale fine to 2 CPUs for many low-level server programs. The problem for scalability is that that FSB and memory bandwidth is too low except for rather bandwidth light programs. This is because either CPU can use all the available memory bandwidth, so for programs entirely dependent of memory bandwidth an extra CPU won't help.

I/O: The 440GX chipset only supports AGP and 33MHz PCI, while the 840 chipset comes with a 64-bit 66MHz PCI bus as well as AGP and 33MHz PCI.

CPU caches: The 0.13um Pentium IIIs have up to 512KB of Level 2 cache, and the Level 1 instruction and data caches are 16KB each. The 0.18um Pentium IIIs have 256KB of Level 2 cache.

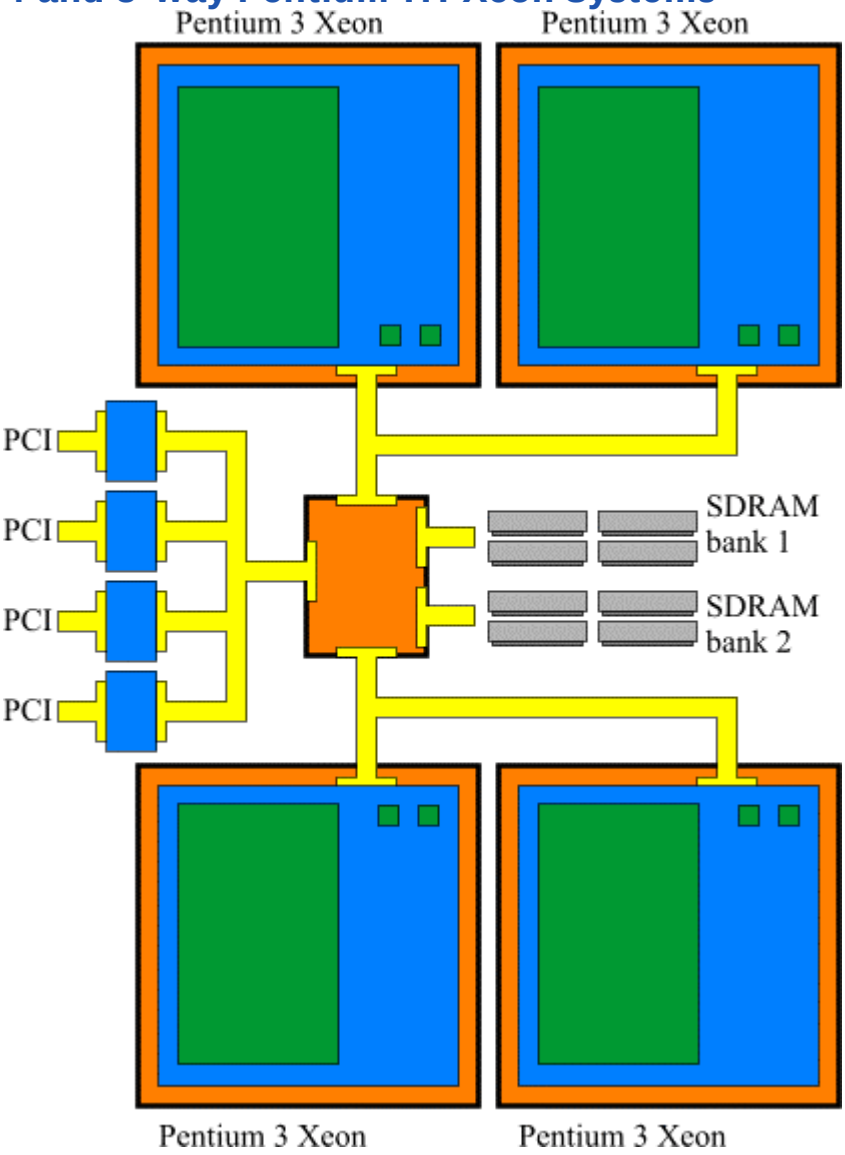
CPU core: The Pentium III reached 1GHz at 0.18um and has so far got to 1.4GHz at 0.13um. The Pentium III is a 3-issue superscalar, with out of order issue, like the Pentium 4, but has a much shorter pipeline, and poorer branch prediction. At 0.18um it's also about half the size of the 0.18um Pentium 4, and maximum power consumption is 31W at 1.4GHz.

Summary: Though it's quite old, the low-end Pentium III based designs are still very popular as low-end servers, particularly in high density systems (1U/2U rack mount servers), where the low power consumption is very useful - helps increase the amount of performance for a given space while keeping running costs low. Being relatively old and low-end, the Pentium III systems are quite cheap, while offering enough performance for most small business tasks. According to market research companies the 2-way x86 server market was worth over $10Bn in 2001 and the significant majority were Pentium III systems.

References:

- List of Intel chipsets including 440GX and 840.

4 and 8-way Pentium III Xeon Systems



Chipset architecture: The 440GX chipset mentioned previously supposedly can support 4 Pentium IIIs off the single FSB, but it seems none of the major vendors are using this chipset for 4-way systems, and are using chipsets from ServerWorks instead. Possibly the most common 8-way chipset used is the Intel ProFusion chipset, though it was actually designed by another company originally. The ProFusion chipset has 3 normal Pentium III FSBs, each of which handle up to 4 devices, for a total of 8 CPUs and 4 I/O ASICs. It's not quite clear if the main part of the ProFusion chipset is a single ASIC or a small group, but all 3 FSBs are connected to it, and it copies requests and data between the 3 FSB domains.

Bandwidth and latency: The Profusion FSB is about the same as normal Pentium III systems - 64-bits wide running at 100MHz for 800MByte/s of bandwidth each. The memory controller supports two separate memory channels each supporting up to 16GB of memory and 800MByte/s of bandwidth. Memory latency is about 140ns when the system is idle, according to the documentation.

Scalability: For 8-way and above, a single shared FSB would be impractical to design and also have poor scalability (too much contention), so for bus-based FSBs, multiple bus domains are used. The 8-way Pentium III Xeon systems have 2 buses for the CPUs, and main memory and I/O requests would only go over one bus, while cache snoops would go over both. This helps, though as the scalability figures earlier in the first part of the article show, the Pentium III Xeons don't scale too well from 4-way to 8-way - too much bandwidth demand for the system. The CPUs do have 2MBytes of on-die cache, which would help with cache hit-rates, but not enough.

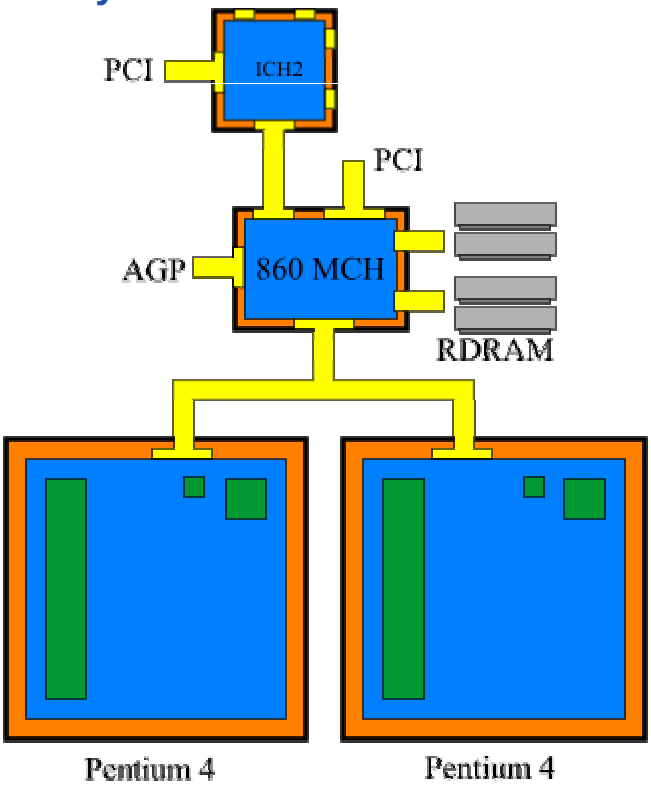
CPU: The high-end Pentium III Xeons are essentially the same as the lower end ones, except they support 2MBytes of on-die Level 2 cache, and a 36-bit virtual memory system, allowing them to address up to 64GBytes of memory. On the 0.18um process the die size is 385mm^2 (nearly 4x bigger than a normal Pentium III), they run at up to 900MHz, and typical/maximum power is 27W/41W.

Summary: Though it adds quite a bit to the price, the higher end Xeons probably do need their 2MB cache quite badly as the available memory bandwidth is only 1.6GByte/s maximum. It does seem however, that performance in server applications does improve enough to be useful over 4-way x86 systems - particularly for customers tied to the x86 platform. During their lifetime, the 4 and 8-way Pentium III Xeon systems have been quite popular with customers, being much cheaper than any of the RISC alternatives for much of the time. However, the higher-end line has more or less come to an end, apart from a 900MHz part being released last year. During 2002, the whole Pentium server line will transition to Pentium 4 based designs.

References:

- ProFusion chipset

2-way Pentium 4 Xeon Workstations



Chipset architecture: Intel's chipset architecture for the 1 and 2-way capable Pentium 4s centers around the "Memory Controller Hub", which has a direct connection to all the system's main memory, an AGP bus, direct connection to the I/O Controller Hub which does other I/O, and the FSB which connects to up to 2 Pentium 4 CPUs. In theory Intel could use the same chipset and CPUs for 1 and 2-way workstations motherboards, because of the connection between the Pentium 4 and the MCH is a bus architecture, but Intel's "PC" Pentium 4 chipsets (850 and 845) are single processor only, and they have a specific chipset for the dual processor workstations and servers - the 860. They also have Xeon branded Pentium 4s specifically for this chipset, though in design they're identical to regular Pentium 4s. The chipset supports up to 8GBytes of main memory.

Bandwidth and latency: The 860 chipset has two separate Rambus RIMM channels on board, each using PC800 specification RIMMs (16-bit 400MHz DDR, or 1.6GByte/s), for a total of 3.2GByte/s of main memory bandwidth. The 860's bus with the Pentium 4s is 64-bit 100MHz QDR for 3.2GByte/s of bandwidth, though more recent boards and CPUs support a 133MHz QDR bus. Though the bandwidth and especially the sustained bandwidth is unusually high for it's class, the latency is perhaps around that of PC1600 DDR SDRAM.

Scalability: Like with the Pentium III, a single Pentium 4 can use up all the available memory bandwidth, so while applications entirely dependent on bandwidth get a big boost with the 3.2GByte/s of bandwidth, an extra CPU won't help. However, most workstation applications aren't so dependent. Like the Pentium III, the bus-based architecture allows for quick broadcasting of snoop requests, though when memory usage is very high, there could be some delays.

I/O: One 64-bit 66MHz capable PCI bus, supporting up to 2 64-bit 66MHz PCI slots or up to 6 32-bit 33MHz PCI slots.

CPU caches: Each Pentium 4 has 256KBytes of Level 2 cache (512KBytes for 0.13um version) with low latency and high bandwidth, which is good because the Level 1 data cache is just 8KB, though with a 2-cycle latency. The Pentium 4 has an interesting instruction cache, which stores instructions in a partially decoded state, instead of the raw data - it can store 12,000 such instructions, perhaps equivalent to a 48KB instruction cache.

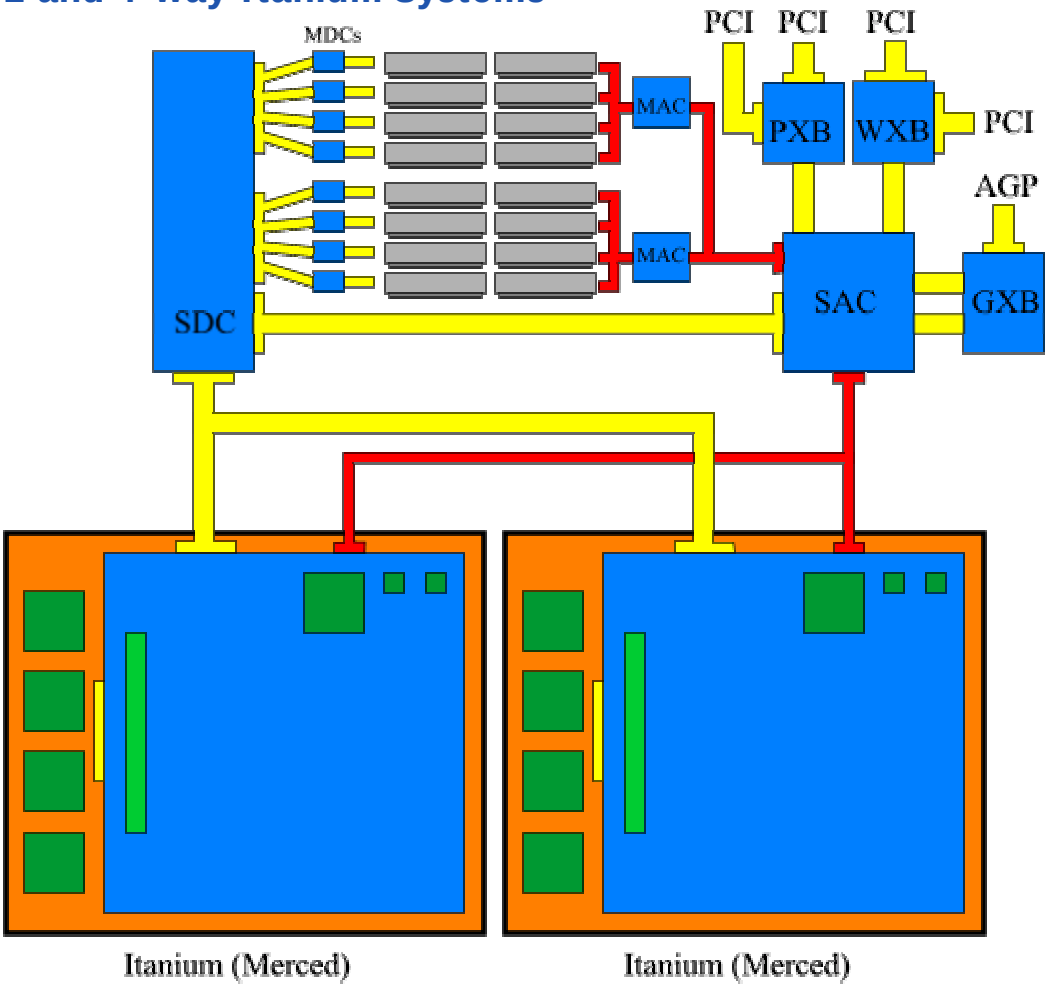
CPU core: The Pentium 4 runs at up to 2GHz on a 0.18um process, but it has a downside in that it has a branch mis-prediction penalty of about 20 cycles due to the very long pipeline. Though it does have very accurate and advanced branch prediction unit, tests indicate it does not perform well on branch-heavy code. The CPU can issue up to 3 instructions per cycle out of order, and has a 128-bit SIMD unit (SSE2), which is useful for many workstation tasks. Typical/maximum power consumption is up to about 75/100W for the 0.18um 2GHz versions.

Summary: The dual processor Pentium 4 systems looks well suited to workstation applications on paper, and in practice too based on benchmarking. But it's less well suited to server applications, and the 0.13um Pentium III based (Tualatin) dual processor servers are quite popular in comparison, probably because it's cheaper, much lower power and has decent server performance compared to standard Pentium 4s. Indeed, Intel are pushing the 860 chipset as a workstation solution only and none of the major OEMs have used the 860 chipset for servers at all - it's not until the "Prestonia" versions that the Pentium 4 has started to be pushed as a server system.

References:

- Intel 860 chipset

2 and 4-way Itanium Systems



Chipset architecture: Although like Intel's other CPUs the design uses a bus-based architecture for the Itanium, there are two main ASICs that handle the FSB - one for address and control, and the other purely for data. Splitting the complexity like this (some higher-end RISC chipsets do something similar) enables a higher-end solution more easily than with a single large main ASIC. The SAC (System Address Controller) ASIC directly links to 3 I/O ASICs, which handle PCI, AGP, and other system I/O. Data from I/O goes back to the SAC and is routed to the SDC (System Data Controller), and vice versa. Memory address/control signals are sent by the SAC to the memory banks and data goes to (or comes from) the SDC. The SAC and SDC ASICs have a private connection for passing data to and from the I/O system to the CPUs and main memory system. The chipset supports up to 64GBytes of main memory.

Bandwidth and latency: The memory used in the chipsets 66MHz SDRAM (though PC100 specification, as it's hard to run many SDRAM channels in parallel at high speed), and with 8 controllers the maximum memory bandwidth is 4.2GByte/s. Itanium motherboards do not have DIMM sockets, but have pluggable memory cards which have up to 16 SDRAM DIMM sockets each. The SAC has control/address lines connected to the two memory cards, but instead of connecting directly to main memory they go to an ASIC ("MAC" - Memory Address Controller) which fans out the addresses to 2 sets of 4 parallel banks of memory. This means that the data connection from each card is 2 x 256-bits wide, running at 66MHz, and that is sent into a group of ASICs ("MDC") which pump it out at 266MHz and 64-bit wide - 2.1GByte/s. So the SDC has two 2.1GByte/s connections to memory, for 4.2GByte/s in total, though the shared FSB bus speed on the Itanium is 2.1GByte/s, which leaves quite a bit for I/O DMA. However, because the memory is running at 66MHz, and because of the extra ASICs the address and data has to hop through (each adds more latency), the latency would be quite low by current standards - Itanium chipsets seem much more optimized for bandwidth than latency.

Scalability: The bus style of connecting between CPUs and the chipset is more or less the same situation as with current Pentium systems, except compared to the Pentium III based ones, it's running at just over twice the speed. It has the same advantages and disadvantages as well - relatively simple, but doesn't scale quite so well. However, since each Itanium comes with up to 4MB of cache, memory usage would be a bit lower on comparable workloads due to the greater cache hit rates.

I/O: With up to 4 separate PCI channels, each supporting 64-bit 66MHz PCI, the Itanium systems certainly have a huge amount of I/O available, probably enough for just about anything.

CPU caches: Each Itanium has 2 or 4MB of high-speed SRAM for a Level 3 cache, but external to the CPU, rather than internal as on the high-end Pentium III Xeons. The Level 3 cache is in a MCM (Multi Chip Module) which makes running signals at higher speed between the SRAM and the CPU easier, and Intel are using custom designed high-speed SRAMs running at the same clock rate as the CPU. The CPU has a 96KByte on-die Level 2 cache and the instruction and data caches are 16KByte each with 2-cycle latency.

CPU core: The Itanium core, with the EPIC ISA is a VLIW design, where the code the CPU executes specifies how individual instructions will be executed in parallel, rather than the CPU deciding this itself, which makes performance especially tied to how good the compiler is. The idea is to make the CPU design simpler (for a particular performance level), allowing for either a cheaper design or a higher performance design for the same price. With the EPIC architecture, instructions are grouped into bundles of 3, and up to 2 bundles can be executed per cycle, equivalent to 6 instructions. It also has about double the floating-point hardware as most CPUs, being able to issue 2 MACs (multiply-add, a staple of many floating-point heavy programs) per cycle. At 800MHz with 4MBytes of Level 3 cache, the maximum power consumption for the CPU module is 130W.

Most high-end DSP chips, some of which are 8-way issue, which have processing tasks similar to workstations, but somewhat simpler, while the rest of the embedded market is dominated by RISC CPUs. The downside with a VLIW architecture, is that the less predictable the execution of the code is (and the system as a whole, including caching) the harder it is for the more rigid design to maintain good performance, while a more intelligent (complex) design has more flexibility. The typical programs that DSPs execute tend to be more predictable and being embedded systems they can be re-compiled and re-optimized for each installation making backwards compatibility (in operation and performance) less of an issue. Size, cost and power consumption is also particularly critical for DSPs, but even still it's taken VLIW a while to take off there. This is partly because the compiler work is so hard - many DSPs just run one simple algorithm for the particular product but even then it can take several major versions of the compiler to get maximum performance.

For Itanium, the problem is that server code is about the least predictable and hardest to get efficient. Due to the long time taken to develop good reliable compilers in general (seems Intel started from scratch for Itanium) and the complexity of compiling for Itanium, it's hard to get a good picture of the Itanium's real potential, though results to date suggest Intel hasn't been able to mask the downsides of a VLIW architecture.

Summary: The initial Itanium launch was about 2 years late and performance was lower than expected, and in the end the initial Itanium release was demoted to an extended trial run, instead of the major push for enterprise servers that was originally expected. With it's comparatively fast main memory bandwidth, and significant floating point hardware the Itanium does well on some floating-point heavy benchmarks (which are also somewhat more suited to VLIW architectures, like DSPs) but performance in integer specific benchmarks (like SPECint) have been quite low, especially given the high-performance cache architecture. Also, the Itanium consumes too much power for most high-density (rack-mount) sites.

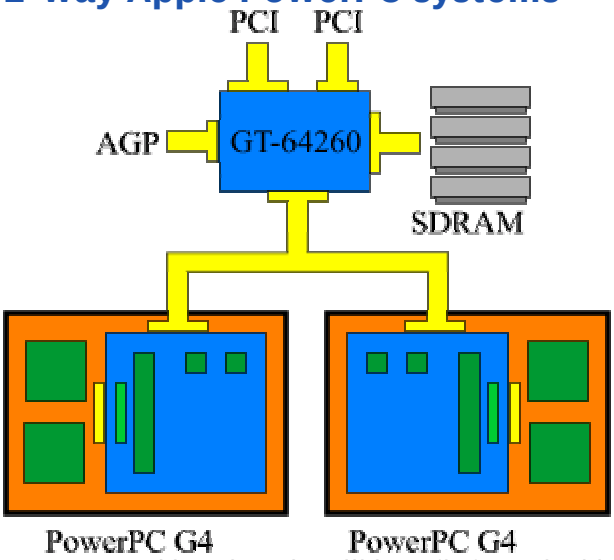
For most HPC applications, the end-users compile and run the programs themselves, so if they have a good OS and a good compiler (and development tools in general) - they can "roll their own" effectively. For the rest of the market, most customers are really waiting for applications to become available before even starting to evaluate systems, which is probably why sales have been quite tiny so far - there's almost no software available to date. Though a huge number of x86 programs can be run directly on the Itanium, the performance is far too low for any kind of development or production system.

However, even when this becomes less of a problem, Merced and successors could have some tricky problems with reliability, because of the compilers. Itanium pretty much needs an entirely new set of compilers, but getting reliable compilers takes a long time, and the very complex EPIC ISA makes this far worse - it's very hard for humans to understand and debug. Some Merced customers have encountered several situations when the cause of bugs couldn't be resolved - couldn't tell if it was a compiler fault or a CPU fault. These kinds of problems will certainly make enterprise customers wary, and these are the customers that Intel most wants to attack with the Itanium in the long term.

References:

- 460GX chipset documentation

2-way Apple PowerPC systems



Chipset architecture: The basic architecture is similar to Intel's dual processor systems - two CPUs sharing a bus with an ASIC ("GT-64260") that has a memory controller. However, it seems the chipset Apple uses has the southbridge integrated as well, which should save costs, power, size and increase (I/O related) performance a little bit. Up to 1.5GByte/s of SDRAM is supported, though a newer chipset supports up to 2GByte/s of DDR SDRAM.

Bandwidth and latency: The FSB runs at 133MHz at 64-bits wide, and uses 133MHz SDRAM, so is basically the same as the Pentium III systems. The new chipset supports 133MHz DDR SDRAM - double the main memory bandwidth, though the FSB is the same.

Scalability: The shared FSB has roughly the same characteristics in terms of performance and latency as Pentium III systems. However, with up to 2MBytes of cache for each PowerPC, many programs and benchmarks will benefit from the higher bandwidth and lower latency of the cache compared to main memory, and also get better scalability to two CPUs as there would be less contention for the FSB and main memory.

I/O: The GT-64260 ASIC has two separate PCI channels, one supporting 32-bit 33MHz PCI, and the other supporting 64-bit 66MHz PCI. The newer chipset supports 2 64-bit 66MHz PCI slots, and 1 32-bit 33/66MHz slot.

CPU caches: The G4+ PowerPCs instruction and data caches are 32KBytes each with 3 cycle latency, have a 256KByte Level 2 cache with 9 cycle latency, and a 1-2MByte off-die Level 3 cache with a 64-bit 250MHz connection and 40 cycle latency (quite slow), though more recent versions use up to 2MBytes of DDR SRAM effectively running at one-half the CPU clockrate (250MHz DDR = 500 MHz effective at 1 GHz).

CPU core: The PowerPC is a 32-bit RISC architecture (which is a cut down version of IBM's POWER CPU design), can issue 3 instructions per cycle out of order, with a 7 stage pipeline, and an advanced 128-bit SIMD unit called AltiVec. The PowerPC G4+ die size is 106mm^2 at 0.18um, runs at up to 1000MHz on Motorola's 0.18um SOI process. It dissipates a maximum of 30W at 1 GHz, and typically 21.3W at the same clockrate.

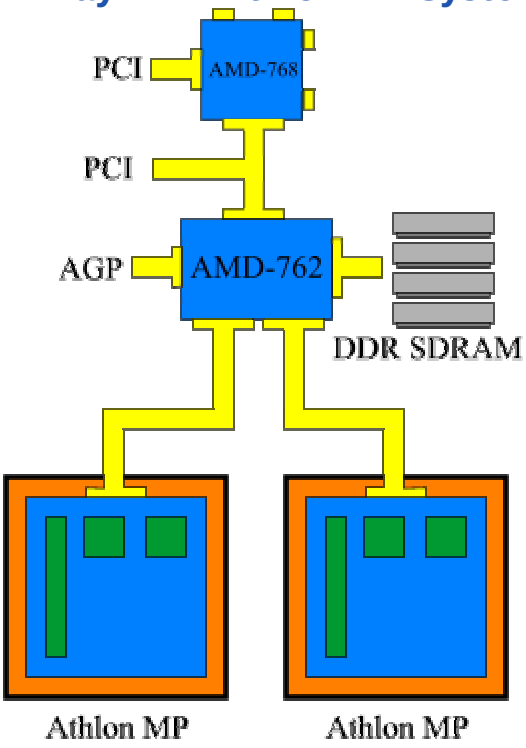
Against a comparably performing CISC design a RISC CPU would normally be somewhat simpler to design and smaller as well, as RISC ISAs are mostly more streamlined and less wasteful. Like with the Itanium VLIW design, RISC CPUs do need a better compiler (mostly to make use of the extra registers), but RISC ISAs do not specify how instructions should be executed in parallel, which makes a big difference, so are closer to CISC CPUs in this regard. The more efficient design is why almost the entire embedded market is now based on RISC or VLIW designs - this market is much more sensitive to performance, costs and power consumption.

Summary: Without the 2MByte Level 3 cache and AltiVec unit, the current PowerPCs would almost certainly be beaten in any (reasonable) benchmark by the other systems in this article - the clock rate, core design, cache memory and bandwidth and memory latency and bandwidth are just too low. This is pretty much because current PowerPC CPUs are mostly sold to the embedded market (which requires low cost and low power, and mostly requires DSP like algorithms to which AltiVec is well suited) and though Motorola do some higher-end versions with a more aggressive core design, the embedded design requirements are what's responsible for the slow SRAM speeds (slower SRAM is smaller and consumes less power), the relatively slow clock speed (simpler, shorter pipeline) and relatively unaggressive use of higher voltage (1.6v core voltage). The newer chipsets which use DDR SDRAM help, but the FSB needs to be faster too, and ironically the first DDR SDRAM systems from Apple are for servers - workstations would make much better use of the bandwidth. Even with DDR memory, the CPU is rather too slow for many workstation tasks - the CPU core and cache speed is rather too low. However, with it's relatively low power consumption and decent cache size and good I/O, it makes a nice low-end MP server, particularly for high density (rack-mount) markets.

References:

- MPC7450 RISC Microprocessor Family User's Manual
- MPC7450 Product Summary

2-way AMD Athlon MP Systems



Chipset architecture: Each Athlon MP has a point-to-point connection to the main ASIC in the 760MPX chipset ("AMD-762"), and each CPU can only talk to each other via the same chip. The northbridge has a DDR SDRAM memory interface, AGP port, and link to the "southbridge" ("AMD-768"). Like the normal Pentium 4s and the Xeon versions, non Athlon MP CPUs can be used in the MP systems, but only the Athlon MP is guaranteed for work correctly in such systems. The northbridge comes in a 949-pin package (quite large), and supports up to 4GBytes of DDR SDRAM.

Bandwidth and latency: The FSB is 133MHz DDR, as is the main memory speed, which at 64-bits wide gives 2.1GByte/s of bandwidth. With two independent FSB connections, the aggregate FSB bandwidth is 4.2GByte/s, but given that most of this will be main memory requests (limited to 2.1GByte/s), utilization will rarely be high, as I/O and cache snooping bandwidth won't add much. Latency for an idle system would be about the same as with a bus-based FSB using the same speed memory, etc.

Scalability: With heavy load, latency should scale better and actual maximum throughput should be higher than for a 3 device shared bus of the same speed - with two connections, both CPUs can make requests or receive data concurrently. With light load the latency and bandwidth would be very similar to an equivalent shared bus. However, because requests and data between the two CPUs have to go via the central switch, the cache snoop latency is double that of a bus running at the same speed as it takes 2 bus transfers for a request (or response) instead of 1. With two channels to the northbridge, one CPU could be reading data while the other could be writing it.

I/O: The connection between the northbridge and southbridge is actually a 64-bit 66MHz PCI connection, which can support two 64-bit 66MHz PCI cards as well. In addition, the southbridge has a 32-bit 33MHz PCI connection, supporting 8 devices, and other system components such as IDE and USB.

CPU caches: The Athlon's instruction and data caches are 64KBytes each with 3 cycle, has a 256KByte Level 2 cache. With it's much larger Level 1 caches compared to Pentium 4, the Athlon's Level 2 performance is somewhat less critical to performance, and has poorer latency and bandwidth, though this is possibly partly due to the Athlon's Level 2 cache originally being an external design. The Athlon also has an "exclusive" cache design, in that there is never any data duplicated between the Level 1 and Level 2 caches. This does make the caching algorithm a bit more complicated, but does increase the total amount of cacheable code and data from 256KBytes to 384KBytes. Most cache designs are "inclusive," where all data in the Level 1 caches is also present in the Level 2 cache.

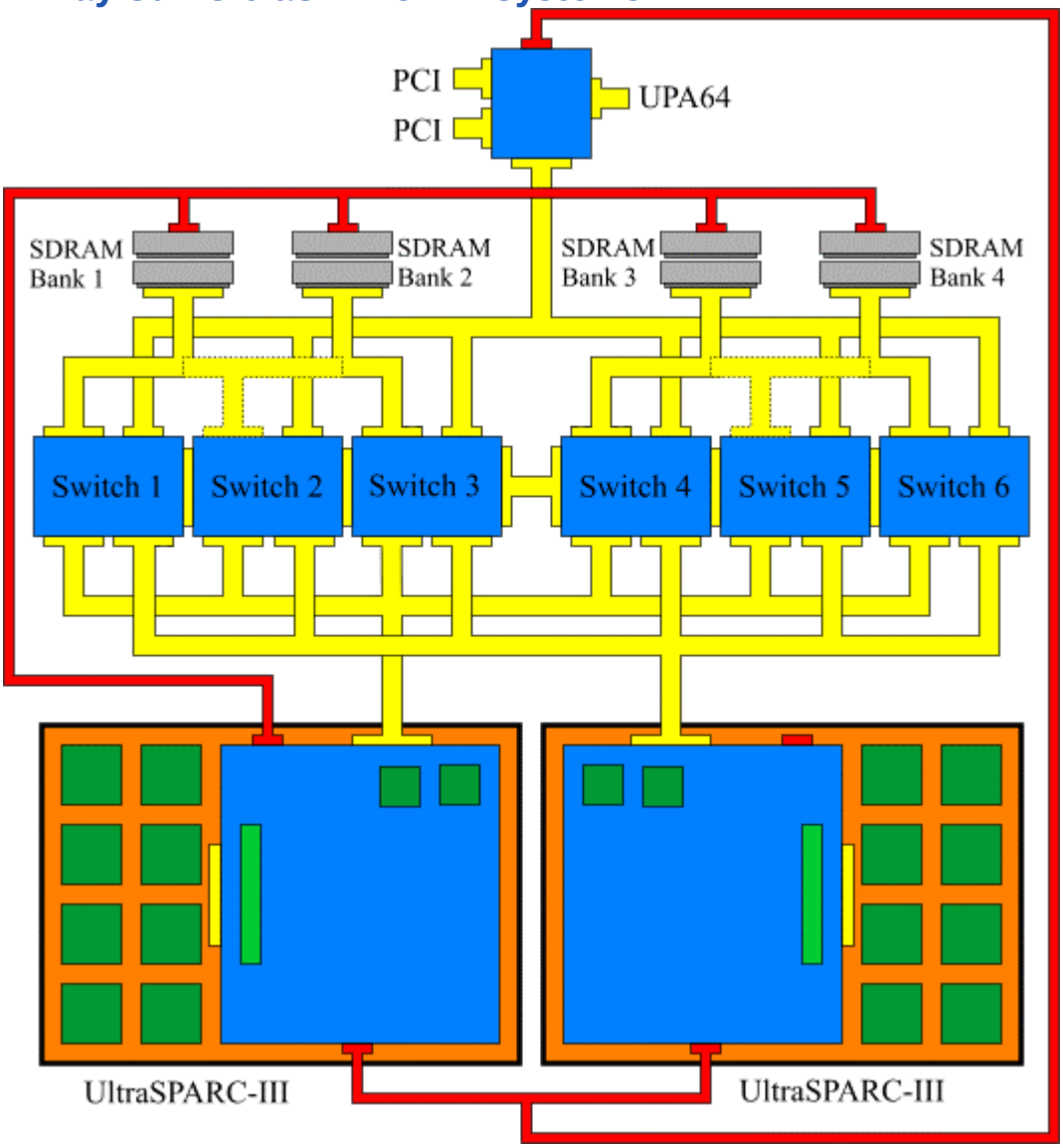
CPU core: Like the Pentium III design, the Athlon seems to be quite a good all-round, balanced design, though in implementation it seems better in every respect, and has in particular a much higher clock rate despite having a similarly long pipeline. Performance for general floating-point code is much better than other x86 CPUs too, though loses out against Pentium 4 SSE2 optimizations when applicable. Average/Maximum power consumption for Athlon MP at 1666MHz is 62/70W on AMD's 0.18um process, and will probably be 54/60W when released on the 0.13um process.

Summary: Considering this is AMD's first dual processor system, they have been well received among enthusiasts and some companies, due to its high performance and low pricing. One area where it loses out though is in power consumption, as the fastest parts consume about twice the power as the fastest 0.13um Pentium IIIs, and for business applications, the performance difference is not that significant. While AMD hasn't submitted much in the way of server benchmarks itself, some testing has shown it can make a good server, but its relative performance is not entirely clear currently. AMD has won a number of design wins from small OEMs (for both workstations and servers), and is mostly concentrating on the "white box" OEMs, and none of the large OEMs are using the chipset or have announced plans to.

References:

- 760MPX documentation

2-way Sun UltraSPARC-III Systems



Chipset architecture: To look at, the chipset is quite complex (and certainly hard to draw) and it doesn't help that Sun's documentation isn't quite complete on the layout and operation, so the above is somewhat of a guess. Each UltraSPARC-III has an on-die memory controller and can directly address up to 4 banks of 2 SDRAM DIMMs. Sun call these DIMMs "NG-DIMMs" (Next Generation) because they're 128-bits wide each, though use normal SDRAM chips. The 4 banks each have a connection to the 6 identical ASICs that form the "Combined Processor Memory Switch". The chipset has some level of redundancy (only 4 of the 6 CPMS chips are needed), a guess of how this could be implemented is shown in the diagram - the memory connection to "Switch 2" and "Switch 5" uses a dash outline pattern to indicate that it's optional, and perhaps if "Switch 1" or "Switch 3" fails, "Switch 2" can take over, and so on.

Each of the switch ASICs has 2 separate bus-like connections to the 2 CPUs, and another to main I/O chip. Any single one of the ASICs can read or write data to one or both of the CPUs concurrently and a separate bus connection is used to transfer data between the switch ASICs. For data transfers from memory to the CPUs, either each ASIC could take turns in transferring the data, or while one ASIC sends data to the CPU, the others could forward their data to it at the same time - the documentation isn't clear on this. Data transfers between CPUs also go via these ASICs, though the address requests are sent by a bus the CPUs and main I/O ASIC share, allowing lower latency than sending the request through a switch ASIC - address requests require much less bandwidth than data, which makes this much more practical. In the dual-processor UltraSPARC-III Sun systems, only one of the CPUs is used for main memory requests (to reduce costs perhaps), and the shared address bus is used to initiate requests.

Bandwidth and latency: Each UltraSPARC-III has a dedicated 128-bit 150MHz (2.4GByte/s) connection to the chipset, which supports up to 4.8GByte/s of memory bandwidth and 1.2GByte/s of I/O bandwidth. With an on-die memory controller, and a short path for the data, latency should be lower than any other current solution with the same 75MHz SDRAM, perhaps around the level of a simple northbridge design with 100MHz SDRAM.

Scalability: With a shared address bus, but a dedicated data bus, the design is a hybrid approach compared the shared bus style Pentium systems and the dedicated point-to-point Athlon systems. The address bus is shared giving 1 cycle latency, but with a 2-way system even bandwidth-dominated applications would not be able to saturate the address bus with cache snoop and memory requests. As each CPU has a separate data bus, cache snoop data, memory data and I/O data can all occur at the same time, which helps bandwidth scalability.

I/O: A 64-bit 66MHz PCI bus, a 64-bit 33MHz bus and two UPA64S graphics cards (Sun equivalent of AGP effectively) are supported.

CPU caches: The UltraSPARC-III comes with 8MBytes of external cache, using SRAM running at up to 350MHz, using a 256-bit interface, for 11GByte/s of bandwidth. The newer "UltraSPARC-III Cu" variant replaces the slower "UltraSPARC-III" which has only 150MHz SRAM. The CPU core has a 90KByte tag-RAM for the external cache (record of the addresses for the data being cached), a 64KByte 2 cycle latency Level 1 data cache, a 32KByte 2 cycle latency instruction cache, a 2KByte pre-fetch cache (for floating-point data only) and a 2KByte write cache.

CPU core: The core has a 9 stage pipeline for integer instructions, though memory requests, branches and floating-point operations use up to 14 stages, and a special alternative branch instruction buffer reduces the branch miss-prediction penalty to just 4 cycles. The original SPARC ISA from 1988 was heavily based on the original RISC-1 design, and was later extended to 64-bits with the UltraSPARC-I in 1995. The UltraSPARC-III is in-order, 4-way issue - like the UltraSPARC-II, though it has significant improvements in the cache and branch prediction systems. The CPU core runs at 1.6V, consuming up to 75W of power at 1050MHz.

Summary: Sun has been the leading supplier for RISC/Unix workstations for over 10 years, mostly in engineering for the high-end and unlike SGI they have relatively little presence in the 3D animation industry. Although at time of writing the 1050MHz Sun Blade 2000 has the second highest 2-way SPECfp_rate peak result, the price/performance compared to x86 based systems is poor. Furthermore, Sun's in-house graphics card development has been very slow of late, and even though their new XVR-1000 card is 3x faster than the previous generation, it still lags in performance and is expensive, though it has many advanced visualization features.

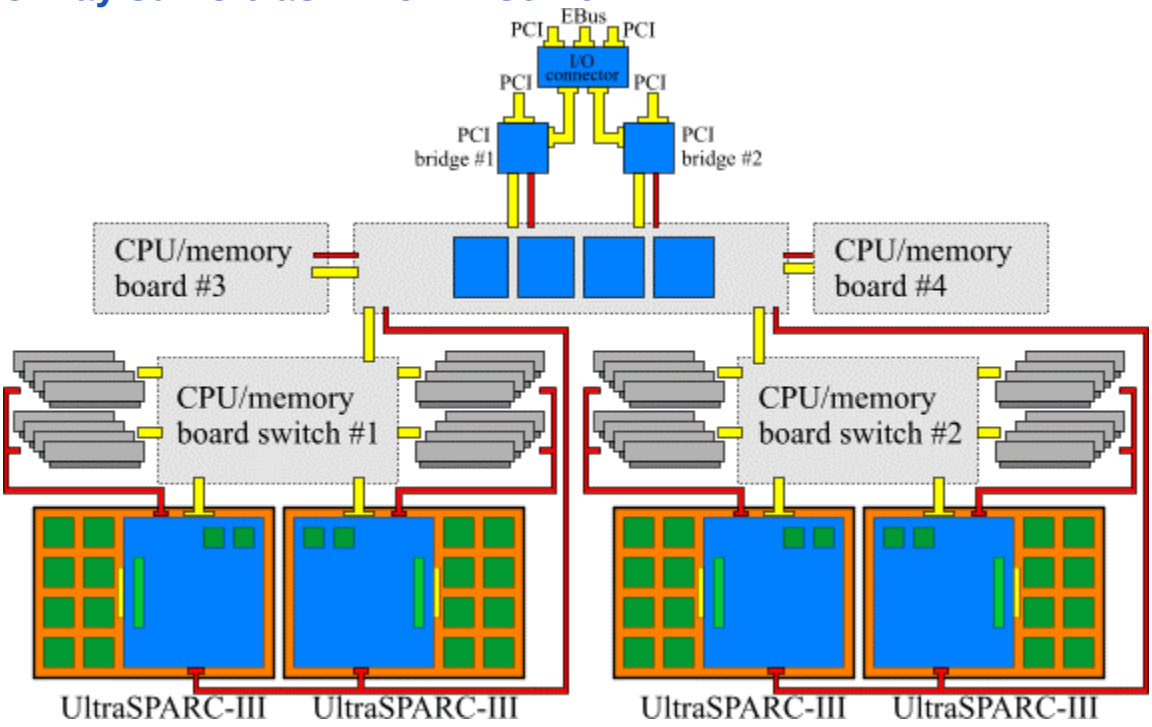
The proportion of the workstation market that requires 64-bit addressing and lots of memory is growing every year and many high-end add-ons and packages in the workstation market are only available on UltraSPARC and HP's PA-RISC, the two leading 64-bit workstation architectures. In addition, for high-end workstation tasks, the software costs can be huge, and the cost of any software problems can be significant, making even "expensive" workstations look cheap. Those factors are probably the main reasons why the high-end RISC workstation market is still profitable. With HP moving to Itanium, and AMD working on 64-bit extensions to x86, competition in the high-end workstation market will likely increase significantly, driving prices down. For some workstation tasks, like EDA (which is CPU and memory intensive but light on graphics), Sun has been moving to a server model - low-end workstations or thin clients sharing the resources of a compute farm.

The server versions of Sun's dual processor UltraSPARC-III systems (Sun Fire 280R and Netra 20) are both rack-mount systems, though 4U high. Though benchmarks for things like web-serving suggest the these systems have performance comparable to high-end 2-way x86 based systems, the Sun systems are much too expensive for most tasks on this level and most web-servers don't need more than 4GBytes of memory or high levels of memory bandwidth. Given that many x86 based 2-way systems are 1U, the 4U Sun system doesn't have good performance density either (the 4-way Sun Fire V480 is 5U), though apparently a more slim-line model is being worked on. However, software such as heavy-duty application servers and in particular lower-end but intensive database servers would benefit well from the extra memory and large high-speed cache. But the price of a 2-way UltraSPARC-III system more than double that of any 2-way Pentium system, which isn't very competitive.

References:

- Sun Blade 1000 and 2000 technical white paper
- Sun Fire 280R technical white paper
- UltraSPARC-III users manual

8-way Sun UltraSPARC-III Server



Chipset architecture: Sun's 8-way UltraSPARC-III workgroup servers (Sun Fire V880) use the same chipset on a CPU level as the 2-way systems, though introduces some new features on a system level. Sun's entire UltraSPARC-III line is built upon the same basic chipset too. The system is split across 4 CPU/memory cards, each with 2 CPUs and up to 16GBytes of memory (currently), with each CPU controlling 8 DIMMs. The 4 CPU/memory cards are connected by a backplane ("Fireplane Interconnect"), which uses a 6-ported crossbar switch, between the 4 CPU memory cards and 2 I/O system connections. Unfortunately, the documentation isn't entirely clear on many of the details - while the documentation states that 4 ASICs are used for the Fireplane Interconnect, the configuration is not detailed, though they are based on 256-bit wide 150MHz connections in a point-to-point architecture. While the 2-way systems have 6 identical ASICs between the two CPUs and the local memory, the Sun Fire V880 CPU/memory cards have 8, though the configuration isn't clear.

Bandwidth and latency: In the 2-way systems, one CPU controls 8 DIMMs, for a maximum memory bandwidth of 4.8GBytes/s. In the 8-way (and probably 4-way) systems and higher, every CPU controls 8 DIMMs, but only provides 2.4GBytes/s each, so an 8-way system has an aggregate maximum memory bandwidth of 19.2GBytes/s. Local memory (on the same CPU/memory board) latency should be fairly similar to the 2-way systems. A non-local request (to a different CPU/memory board) would add several cycles (at 150MHz) to latency. Each of the CPU/memory boards has a 4.8GBytes/s (sustained) data connection to the central Fireplane Interconnect. The Fireplane interconnect can sustain 8.6GBytes/s of data in total.

Scalability: One complication from a scalability point of view is that for applications with an essentially random main memory access pattern (loads evenly distributed across the memory controls) and with 8 CPUs, only 1 in 8 of memory requests will be to the CPU's own memory controller, another 1 in 8 will be to the paired CPU, and the rest will be across the backplane. If all memory requests were local, in theory the maximum bandwidth with 8 CPUs active would be 19.2GBytes/s, but with distributed memory requests, bandwidth would be limited by the 9.6GByte/s backplane.

With 8MBytes of cache per CPU however, probably only some technical computing applications will have a high enough sustained memory bandwidth demands for the backplane to become a limitation. Using software optimizations to increase the chance of a memory request being local would benefit all applications as it reduces latency and increases maximum bandwidth. Given the system design however (local latency not much different from remote latency and bandwidth via the central switch not much of a limitation), the performance increase from such optimizations would likely be small.

I/O: The edge of I/O system connects to the backplane at similar speeds to the CPU/memory boards, and supports 4 PCI channels, all 64-bit with two being 66MHz and two 33MHz channels, and a total of 9 PCI cards. The PCI cards can also be added and used on the fly, without taking the system down.

CPU and cache: The same CPU and cache SRAM parts are used across the entire UltraSPARC-III range, so there's no difference to the 2-way systems. Unlike the 2-way workstations however, 1050MHz parts aren't yet available, though are expected by Q3 2002.

Summary: It's hard to estimate how the memory latency compares with current Intel based 8-way systems, however with more cache, faster CPU cores, far more memory bandwidth and it's switch based architecture, the Sun system has clear performance advantages. Though only a relatively small number of benchmarks have been published so far for the V880, performance has been above that of 8-way Pentium systems to date. Unfortunately, there are not many well-recognized cross platform server benchmarks.

When the Sun Fire V880 was originally launched in October 2001, it cost about the same as similarly configured 8-way capable 900MHz Pentium III Xeon Intel systems, and has been very popular. In fact its popularity was apparently the cause for a delay in the release of a 4-way version - the Sun Fire V480 which was announced just before this article was published. The V480 system design is pretty much identical to the V880 except only having 2 of the CPU/memory cards, and lower I/O capabilities. However, one area where either system will have difficulty getting sales is to customers who entirely use Windows based systems. Such customers would be unlikely to already have staff to administrate the Unix systems, so to them, buying just one could have significant extra costs - need to get at least one trained member of staff who can support it.

References:

- Sun Fire V880 architecture.
- A white paper from IDC on the Sun Fire V480 - little technical documentation on the 4-way system yet, but this article has some useful figures on the server market in general.

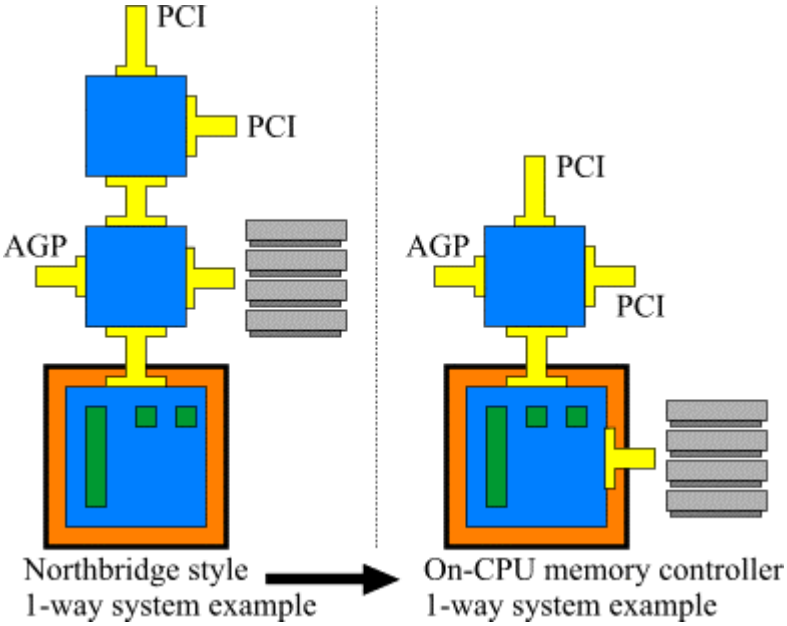
Next Generation Multi-Processor System Features

On-CPU Memory Controller

It's much easier to optimize memory systems for sequential memory access ("streaming data") because it's simpler and much more predictable - programs with large simple arrays of data use the available bandwidth efficiently. However, many programs have much more complicated access patterns (nearly random, particularly given that cache hit are not predictable), meaning bandwidth isn't used nearly as efficiently. Such programs, typical of business applications, become limited by memory latency, even with CPU features like out-of-order execution. In the end, if the CPU can't execute any instructions until a load returns the data needed for processing (either from the CPU's caches or to the main memory system), then it stalls (does nothing) until then. With memory latency on most lower-end systems being around 120-150ns, and CPU speeds being as high as 2.5GHz currently, a single memory request can waste 200-300 CPU cycles. Even with a cache hit-rate of 99% a CPU can still spend over 50% of the time stalled on main memory requests to finish - i.e. memory latency.

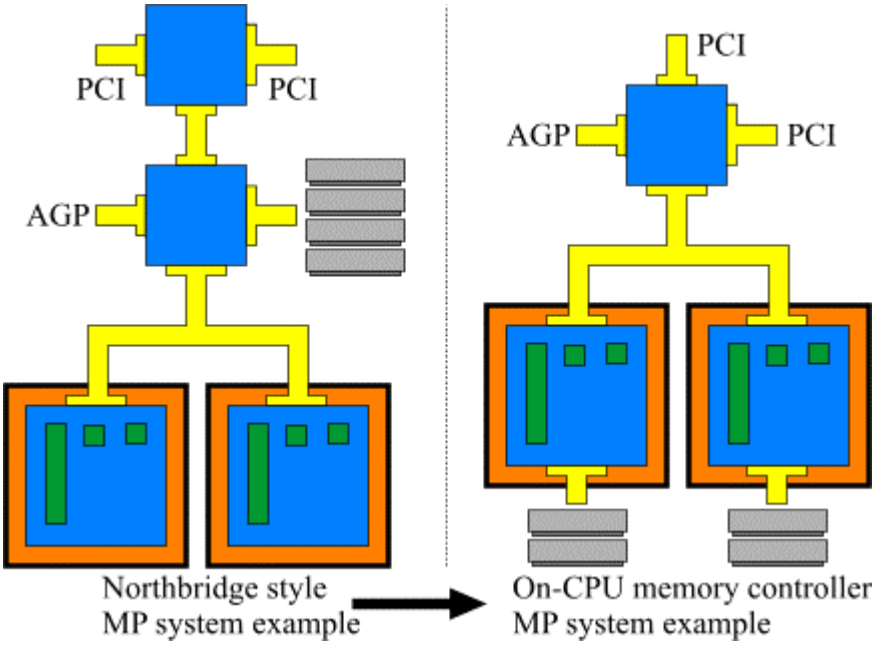
The UltraSPARC-III is an interesting half-way bridge to a CPU design feature that is most likely going to become common in the next few years, namely having a memory controller on each CPU, rather than in a shared ASIC. The advantage for single processor systems over a typical northbridge design is particularly clear as the signal for a memory request goes directly from the CPU and back, instead of via the northbridge, which saves time and so reduces latency. This would save 2-4 or more cycles of the front side bus and memory system (depending on implementation), which would equal around 25-50ns for current designs.

With a northbridge style design, this is about a 15-30% saving in latency, and lower latency also often results in improved bandwidth utilization. For CPUs with relatively small caches or for applications that are dependent on latency, this is very useful. It also pretty much eliminates the need for a "northbridge" and would often enables a cheaper and simpler motherboard. The CPU would become more expensive, though given the extra performance, it can easily be worth it, and moving some of the total system cost from the motherboard onto the CPU is also useful for those companies focused on selling CPUs rather than systems - they get a bigger percentage of the total system revenue.



Transmeta's low-power Crusoe CPUs have an embedded memory controller, as does the forthcoming Alpha EV7. Sun's UltraSPARC-IIe and the new UltraSPARC-IIi design also have a SDRAM controller on-die, and even a PCI controller (ie integrated southbridge). These two CPUs, which are for single processor systems only, have 370 pins in total, and the motherboards for them can be very small. The SDRAM memory controller in the UltraSPARC-III adds about 4mm^2 to the die size, and adds about 35 pins to the CPU packaging - since the data actually goes through the chipset, the pin count is reduced, hence being a half-way bridge. As the UltraSPARC-III's memory bus is 512-bits wide (plus ECC), this is quite a saving

For multi-processor systems with a full on-die memory controller, the aggregate memory system bandwidth increases linearly with each extra CPU, which makes designing a system with high bandwidth easier, and also more scalable - memory resources increase with memory demands. This also helps for server systems which need lots of memory as it's very hard to control multiple memory DIMMs from a single controller at high speed - having more controllers and fewer DIMMs per controller makes a higher speed design easier.



Multi-processor systems for CPUs with on-die memory controllers still need high-speed cache coherency connections, and also a way to access "non local" memory - this is a bit like having a system with multiple northbridges. Such a coherency/non-local connection could be rather like a "front side bus" in current multi-processor systems, which would mean that non-local memory requests would have similar latency and bandwidth characteristics as current systems.

Application Performance and Scalability

For applications where nearly all memory requests are to the local memory (the memory the CPU directly controls), performance could scale nearly perfectly linearly with extra CPUs, depending on the cache coherency algorithm. This is far easier said than done however - most operating systems will allocate memory evenly across the memory controllers, which would mean that memory accesses would tend to be evenly spread as well. One possibility is for the OS to work to try to allocate memory so as to be more local - for example, when a process on a particular CPU allocates some memory, the OS could try to allocate the memory local to that CPU.

This isn't a general solution however - with any application that has a long term large shared in-memory cache (like a database), the memory accesses to that cache would be quite evenly spread as each CPU is just as likely to need any particular bit of that cache. As an alternative, an OS could provide a mechanism (i.e. an API) for a program to allocate some memory local to a particular CPU, and then that program can try to make memory accesses local itself, but can still be limited by how the application works and design difficulty.

Still, the first thing would be to have general purpose OS optimizations for local memory, and I expect this will become common in a few years as all the major OSs either have it already will get it soon, even if they're initially targeted at the high-end. For systems with on-die memory controller, such optimizations could improve the performance of memory-dominated applications by around a factor of 2, so it's certainly useful.

In the meantime, 1-way systems with on-die memory controller will always get the maximum benefit of the local memory, and dual processor systems get a decent improvement. For larger systems the benefit compared to "normal" systems start to fade, particularly those without local memory optimizations. However, even with these caveats, systems with on-CPU memory controller do have another permanent advantage (to those mentioned at the top) - it can be simpler (and cheaper) to design the systems, particular a range of systems, because the chipset partly vanishes. Some flexibility is lost, but most CPUs use just one memory type for the whole generation.

Chip-level Thread Level Parallelism

Improving performance of a single application by using a normal multi-processor system is an example of TLP (Thread Level Parallelism) - an application that has multiple threads (or can be run multiple times in parallel) can have them executed on multiple CPUs in parallel. One way a single chip can exploit TLP is by having two separate CPU cores onto one chip - there is little difference between a computer with 2 separate Pentium IIIs, for example, and the same system but with a single chip containing those two separate Pentium IIIs on the same die - the "front side bus" is already shared, so there's no change there, and each CPU core can run entirely independently as before.

Compared to a similar speed single core, the new CPU die would be about twice the size and consume about twice the power. Compared to two separate CPUs it would actually have a slight performance advantage, as communication between the CPUs would be 10-100x faster, making cache snooping far faster. In general, this technique of putting two or more CPU cores on the same die is called CMP - Chip Multi-Processing.

At 0.13um two Pentium III cores would take up about 160mm^2, only slightly more than a 0.13um Pentium 4. At time of writing the fastest 0.13um Pentium IIIs run at 1.4GHz and the fastest 0.13um Pentium 4s run at 2.53GHz, but for most server applications two such Pentium IIIs would often be faster than a single Pentium 4. For reference, a single 2.53GHz Pentium 4 gets a SPECint_rate of 10.7 while 2 1.4GHz Pentium IIIs get a SPECint_rate of 13.1, or 22% higher performance. In similar production volume, a system with a single dual-core Pentium III would cost about much as a single Pentium 4 system, which is quite an advantage. In addition, the Pentium III core often seems to do better on server benchmarks than SPECint suggests compared to the Pentium 4. However, a dual-core Pentium III would pretty much only be useful for server systems, so in practice, it wouldn't benefit from PC volumes, while a 2-way Pentium 4 system would. For server specific CPUs (large cache Pentium IIIs and Pentium 4s), using dual-core Pentium IIIs may well be faster than Pentium 4s, while costing about the same.

Fine Grained Multi-Threading

Another way of gaining TLP with a single chip is to design a single CPU core such that two or more threads of execution (or processes) can be running through it at the same time. That is, the threads would share the CPU pipeline, execution units, cache system and system interface, though each thread would need it's own set of registers and state values. There's several flavors of this, and the definitions tend to blur a bit - a flavor popular with in-order execution CPU designs is CMT (Coarse-grained Multi-Threading) where the CPU core will only execute instructions for one thread at a time, and switch between threads (in a few clock cycles) on events that take a while to resolve, such as cache misses. Several embedded CPU designs use or plan to use this, and the IBM "Northstar" (POWER RS64 range) series also use this.

Another variant, sometimes called a "barrel processor" is for each stage in the CPU pipeline to work on a different thread, changing to the next every cycle, and to have a very large number of threads running at once through a single CPU. This means that it takes such a long time for a single instruction for a single thread to execute that by the time the next one starts, all processing from the previous one (including main memory requests) will likely have completed. The Tera processor, from the company who now own Cray, has 128 threads on core, and no cache.

Last, but not least, is SMT (Simultaneous Multi-Threading), which is likely to become common for CPUs for more typical multi-processor systems. To some extent this is a natural evolution from CPUs capable of out-of-order execution, which in general have the capability to poll all the instructions waiting to be issued, and execute any instructions where the data they depend upon has already been calculated, regardless of ordering. Instructions from different threads never need to wait on each other, so apart from some special cases (branching, branch miss-prediction, trap events for example), the CPU core isn't that much different. The general idea is that modern CPU cores on typical real world programs often use only 20-40% of their available resources on average. So by running more than one thread through the same core, utilization can be improved, leading to better performance overall.

Performance Improvements with SMT and CMP

For SPECjbb2000 on IBM's POWER4, a system with 16 processor chips gets a score of 202081 with 1 CPU core active per chip, and a score of 339484 with both CPU cores active per chip, an improvement of 68%. Apart from this, the systems (hardware and software) were identical, and on the software side IBM's JVM doesn't seem to have scalability problems, so this appears to be a reasonable example of scalability when using two CPUs cores per processor chip instead of 1.

However, this is simply a single benchmark - other benchmarks will get both higher and lower improvements, depending on the software characteristics. Each POWER4 has 1.44MBytes of shared Level 2 cache and 128MBytes of external Level 3 cache, shared by 4 POWER4s. Given that scalability is much better with 1 CPU active per POWER4 than 2, it's possible that the cache hit rate on the shared Level 3 cache gets significantly worse when 8 CPUs are sharing it instead of 4. From what I know of the benchmark, each thread would use about 30-40MBytes of data, so it's quite possible this is a cache effect that's uncommon in practice. However, without more benchmarks (IBM hasn't submitted 16 and 32-way SPECrate results for the POWER4), it's hard to guess how typical the SPECjbb2000 result is.

For SMT, the recently released Pentium 4 Xeons with "Hyperthreading" are currently the only available example. Intel's documentation suggestions that (for multi-threaded applications) a 20-30% performance improvement might be typical, while almost all programs fit in a range of 10% slower to 80% faster, though as OS optimizations and program optimizations improve and become available, the performance gain from SMT should improve. In one set of tests, the performance increase with SMT enabled varied between 19.8% and 43.3% for tests in the first benchmark, and between 34.2% and 60.1% in the second benchmark. Quite a large variation just for two benchmarks.

Single CPU Design Aspect	Pentium 4 with Hyperthreading	POWER4
Has shared system interface	Yes	Yes
Has shared Level 2/3 caches	Yes	Yes
Has shared Level 1 data cache	Yes	No
Has shared Level 1 instruction cache	Yes	No
Has shared pipeline	Yes	No
Has shared execution units	Yes	No
Has shared branch prediction unit	Yes	No
Number of register files	Two	Two

Consider the table above as a rough comparison between CMP and SMT. From this point of view, it's clear that for a given CPU design, CMP will have higher performance, because fewer CPU resources are shared. This is also why CMP is more expensive, for a given design - less sharing means a larger die size. A CMP design with no shared cache would be the fastest and performance increases would depend on the inherent scalability of the benchmark, and on the shared system interface (memory, I/O and cache coherency bandwidth). Having shared Level 2 and Level 3 caches (instead of having copies) would reduce performance on average because the cache hit-rate would fall.

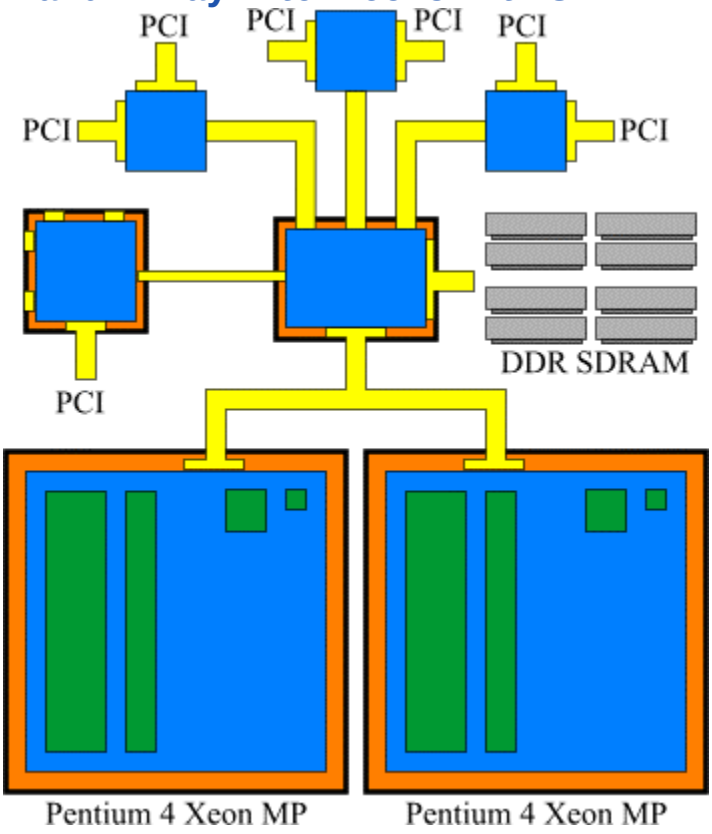
The same would happen with shared Level 1 data caches or instruction caches - performance falls due to conflicts over shared resources. However, for Level 1 caches, unless the CPU is stalled on a cache miss or something else, many programs will do a memory load nearly every cycle, meaning the Level 1 caches will be used every cycle. But cache misses may happen only every 10-20 cycles, so the Level 2/3 caches are shared on a much coarser level - used much less often. This also partially explains why performance can actually get worse with SMT in some cases - with just one thread active, cache hit rates may be high but doubling the amount of active data with a second could cause cache hit rates to drop significantly.

Since with SMT, the pipeline, execution units and branch prediction units (which are caches as well) are all also shared to some extent, so this further decreases average performance compared to a shared-nothing design. However, like with the Level 1 caches, the pipeline, execution units and branch prediction units are often being used every cycle, which makes performance analysis more complex. With just shared caches, the performance will only depend on the pattern of data access, while with shared cores, the performance will be affected even by individual instructions. With so many different aspects of a CPU design, it's more likely that one will become a significant bottleneck, which is probably why there can be such a wide range in performance improvements with activating Hyperthreading on a Pentium 4, even on the same benchmark.

Other Considerations

A processor with two CPU cores should always be faster but more expensive than one of those same CPU cores running two programs simultaneously with SMT. However, a relatively advanced SMT design could well beat a CMP design with two simple cores, cost less to produce as well, and also be particularly fast for situations with only one active program/thread. The downside is that the more complex SMT design is less flexible and also costs more to design - the simpler CMP cores could also be used in cheap non-CMP variations, or high-end non-CMP variations with a lot of cache.

2 and 4-way Intel Xeons with SMT



Chipset architecture: The "Intel E7500 chipset" is actually the ServerWorks GC-HE chipset, but the basic architecture is rather like that of the 860 chipset, except using a 128-bit DDR SDRAM channel (two 64-bit DDR SDRAM channels which are controlled together) instead of 2 Rambus channels, with up to 16GBytes of DDR SDRAM supported. The I/O side is significantly different however, with the northbridge having 4 I/O channels, an 8-bit wide one (hub interface A) leads to an ASIC handling basic system I/O and a simple PCI bus, while the other 3 are 16-bits wide (hub interfaces B-D) each use the PC64H2 ASIC which handles heavy duty I/O.

Bandwidth and latency: With the same FSB and the same memory bandwidth (3.2GBytes/s) as i860-based designs, the memory side is pretty much identical except that DDR SDRAM is used and much more memory is supported.

Scalability: Pretty much the same as the i860 chipset, except that with the Xeon MP, with 4 or more CPUs sharing the same bus, there would be about double the contention for memory bandwidth.

I/O: Each of the PC64H2 ASICs supports two 33/66MHz PCI or two 64-bit 66/100/133MHz PCI-X buses, and the connection to each runs at 566MHz 16-bits wide for 1GByte/s I/O each. It's likely that some of the lower specification dual-processor systems based on this chipset will only use 1 or 2 of the 3 high-speed I/O channels though.

CPU and cache: The lower-end Pentium 4 for servers, the Xeon (codename "Prestonia"), is for 1 to 2-way servers. Prestonia is a Northwood (0.13um Pentium 4 with 0.5MB of Level 2 cache) with Hyperthreading and the OS sees the single Prestonia as 2 CPUs. Prestonia is expected to ramp in speed similar to Northwood, though trailing by a few months, starting at 2.2GHz and moving up by about 200MHz per quarter. The Xeon MP (codename "Foster MP") is based on the 0.18um Pentium 4 design, but with up to 1MBytes of Level 3 cache on-die, and is designed for 4-way and larger servers, and replaces the Pentium III Xeons with 2MBytes of on-die cache. Currently, the 1.4 and 1.5GHz models have only 0.5MBytes of Level 3 cache and only the 1.6GHz model has 1MByte of Level 3 cache. A 1.7GHz version is expected by summer 2002, with further improvements when the design is moved to 0.13um by the end of 2002. The maximum power consumption for the Xeon MP at 1.6GHz is 87W or about 7W more than an equivalent speed Pentium 4.

Summary: In terms of memory bandwidth, total memory capacity and I/O, the new Xeon systems are significantly better, and also rather more expensive. In performance terms, with SMT the Xeons definitely are faster than Pentium III Xeons, but it's not nearly as decisive as for Pentium 4 workstations - the lower-end Xeon MPs are slower than the top-end Pentium III Xeons. Compared to the Pentium III, it seems the Pentium 4's long pipeline (which gives it the higher clock rate) is not actually an advantage for server programs. In fact it seems some OEMs are being slow to adopt the Xeon MPs and some will apparently wait until the 0.13um versions become available late 2002.

Until the Hyperthreading became available for the Pentium 4, it's understandable that it hasn't been pushed as a server solution - it seems the Pentium 4 quite badly needs the extra 20-30% (or more) to regularly beat the Pentium III. For TPC-H 100GByte results, a 4-way 1.6GHz Xeon system only beat a 4-way 900MHz Pentium III Xeon system by 30%. Further software optimization is expected to increase the benefit Hyperthreading brings, though how much remains to be seen. Meanwhile, IBM and others have done their own chipsets for 4-way and larger Xeon MP systems, which come with a Level 4 cache in the chipset with 32MBytes of cache per 4 CPUs.

References:

- e7500 chipset

System Price Guide

Prices here were taken on 10th June 2002. Apple's system prices come from store.apple.com, Intel based system prices come from Dell's website (except the Itanium workstation which is from HP's store), Sun system prices come from store.sun.com. It's hard to find Athlon MP systems OEMs who are reasonably well known, though Fujitsu-Siemens have started shipping Athlon MP servers for HPC clusters, but prices don't seem to be available yet. The Athlon MP server prices come from Rack Saver and the workstation prices come from Alienware.

Prices for the x86 systems do not include OS - add $100-200 to the price for workstations, $200 for Linux, and $800-3500 for Windows server licenses. Prices do not include once-off specials and the like. The workstations chosen were with low-end graphics cards and without a monitor. The cheapest support options were also chosen for all systems. For CPUs, the fastest available were chosen. For the configurations, the specifications were chosen to be close to the Sun systems - unlike most companies, Sun does not offer full configuration of systems on-line, but instead 2-4 "typical" configurations, in terms of CPUs, memory, storage and graphics options. Prices reflect list prices only. The Sun and Apple systems can be bought cheaper (5-15%) via resellers.

Vendor/System	Low-End Configuration	Low-End Price	High-End Configuration	High-End Price
2-way Workstations				
Alienware, MJ-12	1 1.67GHz Athlon MP, 1GB DDR SDRAM, 36GB SCSI	$2300	2x 1.67GHz Athlon MP, 2GB DDR SDRAM, 1 36GB SCSI	$3300
Apple, PowerMac	1 933MHz PowerPC G4, 1GB SDRAM, 36GB SCSI	$3000	2x 1GHz PowerPC G4s, 1.5GB SDRAM, 36GB SCSI	$3800
Dell, Precision Workstation 530	1 2.4GHz Pentium 4, 1GB RDRAM, 36GB SCSI	$3500	2x 2.4GHz Pentium 4s, 2GB RDRAM, 36GB SCSI	$6230
HP, i2000	1 733MHz Itanium, 1GB SDRAM, 18GB SCSI	$8000	2x 800MHz Itaniums, 2GB SDRAM, 18GB SCSI	$15000
Sun, Sun Blade 1000	1 750MHz UltraSPARC-III, 1GB SDRAM, 36GB FC-AL	$6000	2x 750MHz UltraSPARC-IIIs, 2GB SDRAM, 36GB FC-AL	$10000
Sun, Sun Blade 2000	1 900MHz UltraSPARC-III, 1GB SDRAM, 73GB FC-AL	$11000	2x 900MHz UltraSPARC-IIIs, 2GB SDRAM, 72GB FC-AL	$16000
2-way Servers				
Apple, Xserve	1 1GHz PowerPC G4, 1GB DDR SDRAM, 60GB IDE	$3500	2x 1GHz PowerPC G4, 2GB DDR SDRAM, 2x 60GB IDE	$5500
Dell, PowerEdge 2550	1 1.4GHz Pentium III, 1GB SDRAM, 36GB SCSI	$3400	2x 1.4GHz Pentium III, 2GB SDRAM, 2x 36GB SCSI	$5500
Dell, PowerEdge 2650	1 2.4GHz Xeon, 1GB DDR SDRAM, 36GB SCSI	$4700	2x 2.4GHz Xeon, 2GB DDR SDRAM, 2x 36GB SCSI	$7200
RackSaver, UltraThin RS-1129	2x 1.53GHz Athlon MP, 1GB DDR SDRAM, 36GB SCSI	$2400	2x 1.67GHz Athlon MP, 2GB DDR SDRAM, 2x 36GB SCSI	$3700
Sun, Sun Fire 280R	1 900MHz UltraSPARC-III, 1GB SDRAM, 36GB FC-AL	$10500	2x 900MHz UltraSPARC-IIIs, 2GB SDRAM, 2x 36GB FC-AL	$18000
4-way Servers				
Dell, PowerEdge 6400	2x 900MHz Pentium III Xeon, 4GB SDRAM, 2x 36GB SCSI	$18000	4x 900MHz Pentium III Xeons, 8GB SDRAM, 2x 36GB SCSI	$33000
Dell, PowerEdge 6600	2x 1.6GHz Xeon MP, 4GB DDR SDRAM, 2x 36GB SCSI	$18000	4x 1.6GHz Xeon MPs, 16GB DDR SDRAM, 2x 36GB SCSI	$36000
Dell, PowerEdge 7150	2x 800MHz Itanium, 4GB SDRAM, 2x 36GB SCSI	$23500	4x 800MHz Itaniums, 16GB SDRAM, 2x 36GB SCSI	$55000
Sun, Sun Fire V480	2x 900MHz UltraSPARC-IIIs, 4GB SDRAM, 2x 36GB FC-AL	$23000	4x 900MHz UltraSPARC-IIIs, 16GB SDRAM, 2x 36GB FC-AL	$47000

8-way Servers				
Dell, PowerEdge 8450	2x 900MHz Pentium III Xeons, 4GB SDRAM, 2x 72GB SCSI	$26800	8x 900MHz Pentium III Xeons, 16GB SDRAM, 2x 72GB SCSI	$80000
Sun, Sun Fire V880	2x 900MHz UltraSPARC-IIIs, 4GB SDRAM, 6x 72GB FC-AL	$35000	8x 900MHz UltraSPARC-IIIs, 16GB SDRAM, 6x 72GB FC-AL	$100000

Notes (where options don't quite compare)

There doesn't seem to be a way to select a 1GHz 1-way PowerMac, and the most memory the workstations can take is currently 1.5GBytes. Currently the only 1050MHz model Sun Blade 2000 comes with 8GBytes of memory and expensive options (and there is no "low end" version) so this couldn't be used for a comparison - will have to wait until the 1050MHz parts are in volume. For the "low-end" Itanium workstation from HP, a higher-end CPU couldn't be chosen, and a low-end graphics card couldn't be chosen either. The "low-end" Rack Saver configuration has two low-end CPUs because a single CPU couldn't be chosen. For the 4-way Pentium III Xeon systems, 16GBytes of memory isn't available - maximum is 8GBytes for that system. For the 8-way Pentium III Xeon, the server can only take 2 discs internally, so 6 couldn't be selected.

Comments

For most of the workstations, IDE was the default disc configuration, and selecting SCSI added about $300 to the price ($600 for the Apple system). Given that IDE was the default on many systems, this suggests that most customers think IDE is good enough for professional systems. Overall for the workstation market, the top-performing Pentium 4 systems come at quite a margin over other x86 systems ($1100 for the 2nd CPU), mostly because of the high cost of the fastest Pentium 4s, and the memory - which is about twice the price per gigabyte as Dell's DDR systems. There is also a premium for Dell systems over smaller OEMs - the Pentium 4 systems are Alienware are cheaper, though this is normal. It would make for an interesting price comparison if a top-tier OEM has both Intel and AMD workstations and servers.

The situation for the 2-way servers is similar to the workstations, but things change significantly at 4-way and above. A 2 CPU 4-way capable Sun server (with more memory) is a bit more expensive than a 2 CPU 2-way Sun server, but the 2 CPU 4-way Pentium servers are 2-3 times the price of a 2 CPU 2-way Pentium server. Not only that, but the CPU clock speeds are quite a bit lower, though cache sizes are bigger. Overall, the Sun systems are much more competitive here, more so when you count in the cost of a Windows server license, perhaps because volume and margins are similar.