

# Opteron: Pushing x86 to the Limit

By Johan De Gelas – April 2003

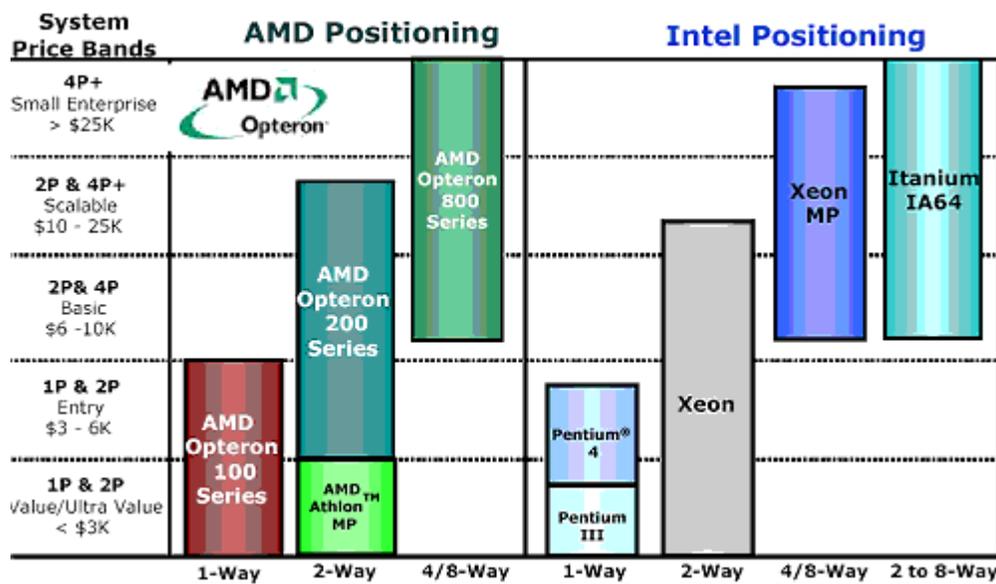
With Contributions from Chris Rijk and Brian Neal

The AMD Opteron has hit the shelves and the expectations are running high, in fact, very high. Not only is this AMD's first CPU that really can take on the Xeon in the lucrative and prestigious server market, it is also the first 64 bit x86 CPU. Being 64 bit, it can address a flat memory space of no less than 1024 GBytes and a virtual memory space of 256 TBytes.

As workstations and server applications push the 4 GB limit of the current 32 bit CPUs, it's clear we need a better solution than slick gimmicks and hacks like the PSE and PAE. What's worse is the 4 GB virtual memory limitation. Today, you can't even store a 4 GB file in virtual memory. Because the kernel and user processes share this paltry 4 GB of virtual memory, applications are probably limited to 3 GB or less. A whopping 256 TBytes of virtual memory will certainly make it much easier to manipulate huge amounts of data. The key selling point of the Opteron is that it also runs all 32-bit x86 applications at high speed, in contrast with Intel's Itanium that only runs applications specifically compiled for it. Being a vastly improved Athlon on steroids with 64 bit extensions, AMD's newest CPU can run 32 bit x86 applications on a 32 bit OS just like the Athlon did. If the hardware vendors provide you with 64 bit drivers for your system, you can install a 64 bit OS and run 32 bit and 64 bit applications on the same OS.

Considering that most chipset and graphic chip vendors have promised to support the AMD64 platform, it shouldn't be a problem to find 64 bit drivers. Furthermore, on a 64-bit enabled OS, existing 32-bit applications would also be able to use 4GB each.

So it's not surprising that AMD is aiming high with the extremely flexible Opteron chip. Opteron was designed to be a good fit to the volume server market (1-8 way), which comprises 99% of the server market. There are also system design companies who intended to push it well beyond 8-way. For a CPU company like AMD, the focus is high-volume sales that target competing Intel's Xeon processors that represent 90% of all server CPUs sold.



The 1U Newisys 2100 server is a dual Opteron, however, a 4-way 3U server will be available next month. The 4-way is definitely not the pot of gold at the end of the Opteron's road, because Newisys plans to introduce a 32-way(!) Opteron at the beginning of the 4th quarter this year. You'll find pricing for Opteron's 2-way systems below. The 1-way models and those for 4-way and higher will have different model numbers and pricing.

Processor	Clock Rate	Price per unit (in lots of 1000)
Opteron Model 240	1.4GHz	\$283
Opteron Model 242	1.6GHz	\$690
Opteron Model 244	1.8GHz	\$794

AMD's pricing is rather ambitious. The 1.6 GHz Opteron costs more than a Xeon 2.8 GHz and the 1.8 GHz model costs slightly more than the Xeon 3.06 GHz.

## The Penguin Drives Another Ferrari (\*)

We tested the Opteron as a HPC (scientific) crunching machine, a web server, a database server, an OLTP server and a rendering farm, because these are the markets the Opteron is slotted for. We also tested the different components, such as the memory controller, cache controller, and the FPU unit with Sciencemark and other more synthetic tests. If you prefer Linux, you will be happy to know we also tested with SuSE Linux Enterprise Server "8" 32 bit, SuSE Linux Enterprise Server "8" 64 bit, Debian Linux Kernel 2.4.20 Pentium 4 optimized and Debian Linux Kernel 2.4.20 k7-SMP optimized. Of course, we also did quite a few tests on a Windows 2000 Server.

Reviewing servers is pretty complex. If you are not careful, you'll end up testing the bandwidth or latency of your network or the transfer rates of your hard disk system. So we carefully tuned our benchmarks so they would only show the power of CPU and its memory subsystem and not the network or hard disks. We tested with three different operating systems, namely SuSE Linux Enterprise Server "8", Debian Linux Kernel 2.4.20 and MS Windows 2000 Server. Naturally, each operating system can be tweaked differently, and few people on earth have yet to master them all. So please, if you are knowledgeable in one of these benchmarked software areas, let us know on the message board, because reader feedback is vital to Ace's Hardware.

One more thing, we only had a week and the half with a new dual Opteron server. Needless to say, our work isn't finished yet, nevertheless, we hope that this review will give you a good idea of what the dual Opteron can do. Keep your eyes peeled in the coming weeks for more to come.

## The Opteron--SledgeHammer Architecture

While an in depth discussion of the Hammer architecture is beyond the scope of this article, let's take a look under the hood at what makes the Opteron achieve higher peak performance than an equally clocked Athlon.

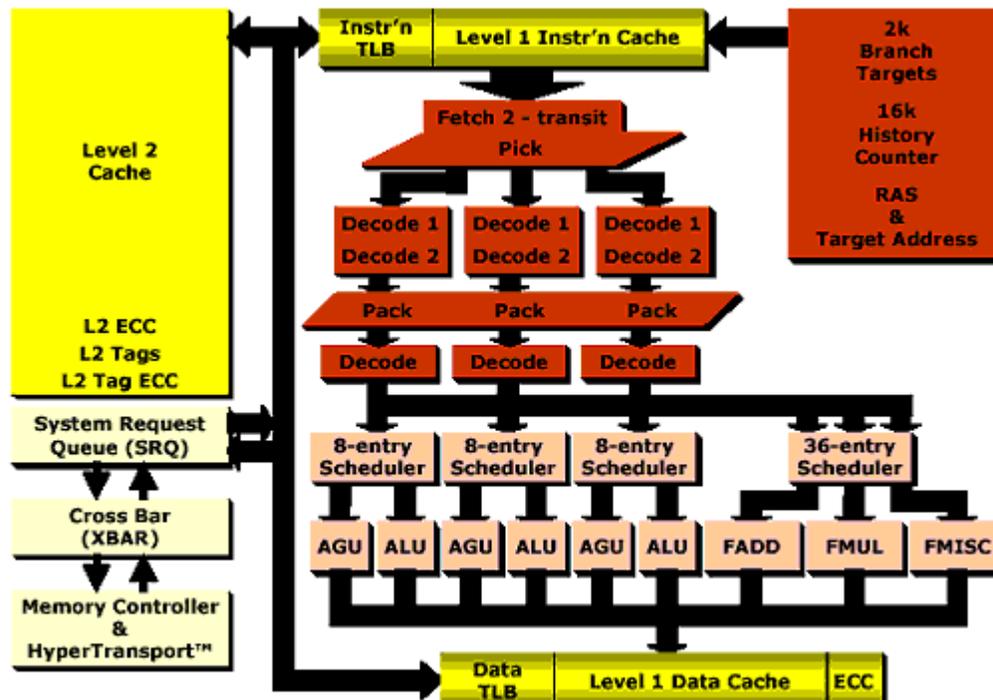
- Integrated memory controller, dual channel DDR333 on Opteron
- Separate HyperTransport links for both inter-CPU communication and communication with the AGP Tunnel and Southbridge
- SSE-2 instruction support with 16 registers in 64 bit long mode
- 12-stage integer pipeline (Athlon = 10), 17-stage FP pipeline (Athlon: 15) for slightly higher frequency headroom
- An extra pipeline stage also analyzes instruction interdependencies, just after decoding
- Slightly deeper integer buffers (3x8 instead of 3x6)
- L1-instruction cache TLB increased from 24 to 40 entries
- L2-cache TLB twice as big (512 entries instead of 256)
- Flush filter allowing multiple processes to share the TLB
- Better branch prediction and branch predictor with 16 K instead of 4K entries in the global history counter

It is important to note that while 5.4 GB/s might not be incredibly impressive in days of Canterwood's theoretical 6.4 GB/s, the Opteron has a separate HyperTransport link to the AGP tunnel--or in the case of the system we tested--to the AMD8131 chip that links to the PCI-X subsystem. This means that fast writes, which aren't passed from the memory, are sent from the CPU directly to the AGP card. Both the Athlon 64 and the Opteron could actually employ extra bandwidth (3.2 GB/s full duplex) on top of the "normal" memory bandwidth (5.3 GB/s in the case of the Opteron). But

in this server review we did not have an AGP tunnel, so extra juice probably won't help much, as PCI-X devices need their data from the memory (via DMA transfers). This means the extra Hypertransport bandwidth via the crossbar is of little importance in this situation. However, we expect it will be quite important when we test the Opteron as a workstation because gigantic amounts of geometry transfer via Fast Writes.

## Opteron Architecture-Reloaded

As you can see, the back end hasn't changed much.



Conversely, the front end has been significantly optimized. After the instructions are decoded, they are analyzed based on type and dependencies. This allows the processor to optimize what instructions should be sent together into the three pipelines, which in turn makes the scheduling a lot more efficient.

You might ask yourself why AMD says Opteron is an eighth-generation processor when it actually looks like a seventh-generation Athlon pumped up on steroids. The pertinent question is "What makes it an athlete instead of just a poser?"

The answer boils down to two serious showstoppers regarding CPU performance--memory latency and conditional branches. As explained in our article, [The Future of x86 Performance](#), there are two serious showstoppers when it comes to CPU performance: memory latency and conditional branches:

"But it is safe to assume that this report confirms other reports which claims that 50% of the x86 instructions found in your average applications are Loads or Stores. Loads seem to happen twice as much as Stores, which is not so surprising. Most arithmetic and logical operations load two data variables, compare/add/multiply those two variables and store one result back."

[...]

"The study also makes it clear (Table 7) that 14 to 16 percent of a typical x86 program consists of branches and indicates that 92% of those branches are conditional (Table 8). In other words, aside from loads and stores, branches are probably the most important x86 instructions"

With the bulk of instructions being Load/Store and Branches, more units and pipelines won't work. A six pipe architecture might sound fancy, but the integrated memory controller, better branch prediction, more efficient Out of Order scheduling and bigger TLBs ([Read here about TLBs](#)) are much better ways to improve IPC (Instructions Per Second).

## Opteron: The Server Processor

A server CPU is much more than just good architecture. Many servers must deliver their services around the clock, and people want to cram as much servers in a rack as possible. Even more importantly, all data must be reliable and redundantly stored, because you don't want to lose weeks of work when gremlins rear their ugly head.

Let's see what Linux CPUInfo utility reports.

```
processor      : 0
vendor_id     : AuthenticAMD
cpu family    : 15
model         : 5
model name    : AMD Opteron(tm)
stepping      : 0
cpu MHz       : 1792.254
cache size    : 1024 KB
physical id   : 0
threads       : 1
fdiv_bug     : no
hlt_bug      : no
f00f_bug     : no
coma_bug     : no
fpu          : yes
fpu_exception : yes
cpuid level   : 1
wp           : yes
flags        : fpu vme de tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat
pse36 clflush mmx fxsr sse sse2 syscall1 mmxext lm 3dnowext 3dnow
bogomips     : 3578.26

processor      : 1
vendor_id     : AuthenticAMD
cpu family    : 15
model         : 5
model name    : AMD Opteron(tm)
stepping      : 0
cpu MHz       : 1792.254
```

It is interesting that the Opteron also supports PSE 36 bit, the 36 bit Physical Address Extensions that the all Intel CPUs have supported since the Pentium Pro. This allows the Opteron to make use of PAE just like the Xeon. Even in today's 32 bit operating systems, the Opteron can address up to 64 GB of RAM, albeit it does so much slower compared to 64-bit mode.

One of the weakest points of the Athlon was the fragile die. If you mount your heat sink poorly, the corners of the die can break. If your heat sink doesn't have enough contact with the Athlon die, the only communication your processor will be doing is with smoke signals. While you can avoid these disasters by being careful, the Athlon MP just doesn't feel as safe as the Xeon and Pentium 4 do. The heatspreader on those CPUs prevent deadly temperature increases and gives other circuitry enough time to throttle back or shutdown the CPU.

According to AMD the Opteron does not feature throttling, but "Thermtrip" is present. AMD wouldn't give us much details, but Thermtrip is fast on-die hardware circuitry that enforces an "over temperature failure prevention mechanism," fast enough to prevent damage and can you believe this--even if you start up the Opteron without a heat sink!

<b>Data Protection</b>	<b>Opteron</b>	<b>Athlon MP</b>	<b>Xeon</b>
L1-data Cache	ECC	Parity	ECC
L2-Cache	ECC	ECC	ECC
Memory Controller	ECC + Chipkill	n/a	ECC + Chipkill
Thermal Protection	Opteron	Athlon MP	Xeon
Heatspreader	yes	no	yes
Shutdown on overheat (reaction time)	On-die thermal diode (slow)	On-die thermal diode (slow)	Clock Throttling (fast)
Temperature damage prevention	Thermtrip (fast)	no	Clock Throttling (fast)

What a huge leap forward. Another step in the right direction, the Opteron has an on-die thermal diode just like the Athlon, but this time motherboards can support it properly. Furthermore, the L1-cache is now also protected by ECC (Error Correcting Code) instead of the Athlon's simple parity protection (only 1 bit detection, no correction). The memory controller is also protected by ECC and can kill off bad memory chips.

Hardware scrubbing is implemented on all ECC protected arrays, including DRAM. Hardware scrubbing is a technique that comes from the mainframe world. It means that memory is read during idle periods to search for and correct errors. However, when x86 server companies talk about hardware scrubbing, they are referring to the checking, detecting and correcting of single bit errors via ECC while the processor requests data but before the data reaches the CPU.

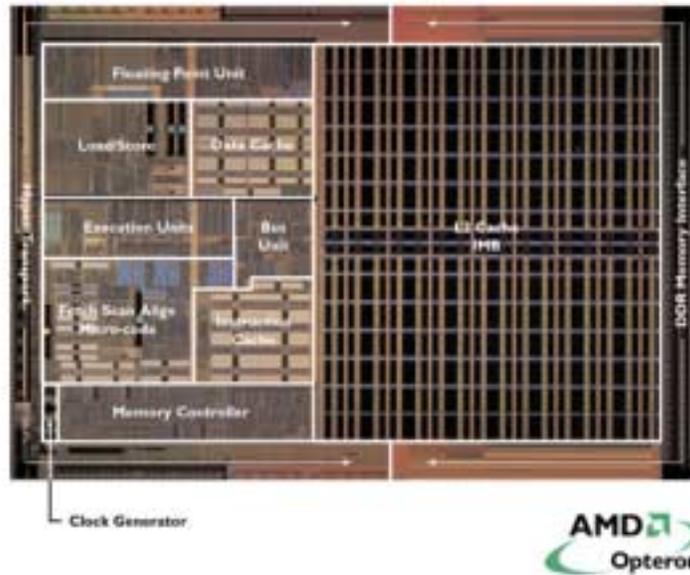
A good overview table is better than a thousand words, so here it is...and the thousand words, naturally.

Features	Opteron	Xeon	Xeon MP	Itanium II	Athlon MP
clockspeed	1.4- 1.8 GHz	2 - 3.06 GHz	1.4 - 2 GHz	1 GHz	1.6 - 2.213 GHz
process technology (µm)	0.13 SOI Cu	0.13 Cu	0.13 Cu	0.18	0.13 Cu
Transistors (million)	105.9	55	?	221	37.5
voltage	1.55V	1.5-1.55V	1.5V	?	1.65V
die size (mm <sup>2</sup> )	193	131	?	464	80
<b>MP and Address Space</b>	<b>Opteron</b>	<b>Xeon</b>	<b>Xeon MP</b>	<b>Itanium II</b>	<b>Athlon MP</b>
Typical Multi processor system	2- 8 (up to 32)	2	2-8 (up to 32)	2-8 (up to 64)	2
Max. Physical Address Space	1024 GB flat (40 bit)	64 GB PSE (36 bit)	64 GB PSE (36 bit)	1024 TB (50 bit)	4 GB
Max. Virtual Space	256 TB (48 bit)	4 GB	4 GB	1024000 TB (60 bit)	4 GB
<b>Processing power</b>	<b>Opteron</b>	<b>Xeon</b>	<b>Xeon MP</b>	<b>Itanium II</b>	<b>Athlon MP</b>
FPU Units	2 FMUL/FADD 1 FSTORE	1 FMUL/FADD 1 FSTORE	1 FMUL/FADD 1 FSTORE	2 FMAC	2 FMUL/FADD 1 FSTORE
Integer Units / Load / Store	3 Int / 3 AGU	2 DP* + 1 Slow /1 Load /1 Store	2 DP* + 1 Slow /1 Load /1 Store	6 Int / 2 Load / 2 Store	3 Int / 3 AGU
SIMD	1 x SSE2/3DNow!/SSE	1 x SSE2/SSE	1 x SSE2/SSE	1 x SSE	1 x 3DNow!/SSE
<b>Cache configuration</b>	<b>Opteron</b>	<b>Xeon</b>	<b>Xeon MP</b>	<b>Itanium II</b>	<b>Athlon MP</b>
L1-cache (Data/Instr)	64/64 KB	8 KB/ +20 KB**	8 KB/ +20 KB**	16 KB/ 16KB	64/64 KB
L1-cache latency (load to use)	3	2	2	2	3
L2-cache	1 MB	512 KB	512 KB	256 KB	256 KB
L2-cache Width	128 bit? (x)	256 bit	256 bit	256 bit	64 bit
L2-cache Latency load to use (+L1-latency)	16(*)	9-20	9-20	5	11-20 (*)
L3-cache	-	-	1 - 2 MB	3 MB	-
<b>Memory</b>	<b>Opteron</b>	<b>Xeon</b>	<b>Xeon MP</b>	<b>Itanium II</b>	<b>Athlon MP</b>
Memory configuration	2 x DDR333	2 x DDR266	2xDDR200/266	4xDDR266	DDR266
Max. Memory Bandwidth to CPU	5.4 GB/s	4.2 GB/s	3.2 GB/s	6.4 GB/s	2.1 GB/s

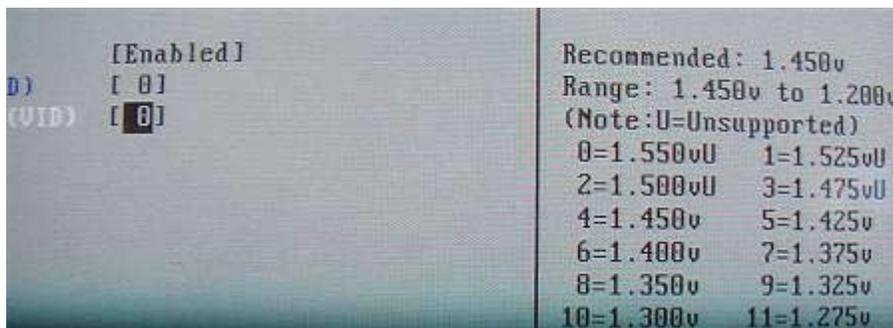
\* Pentium 4 architecture (Xeon/Xeon MP) contains 2 Double Pumped ALU's - ALUs running at twice the speed of the core.  
(\*) See further.

\*\* 12000 Micro ops, which is probably comparable to about 20 KB x86 instructions cache  
(x) not confirmed by AMD, but by sciencemark

The ECC protection, the big L2-cache and the memory controller have made the Opteron a big CPU, with 105.9M transistors.

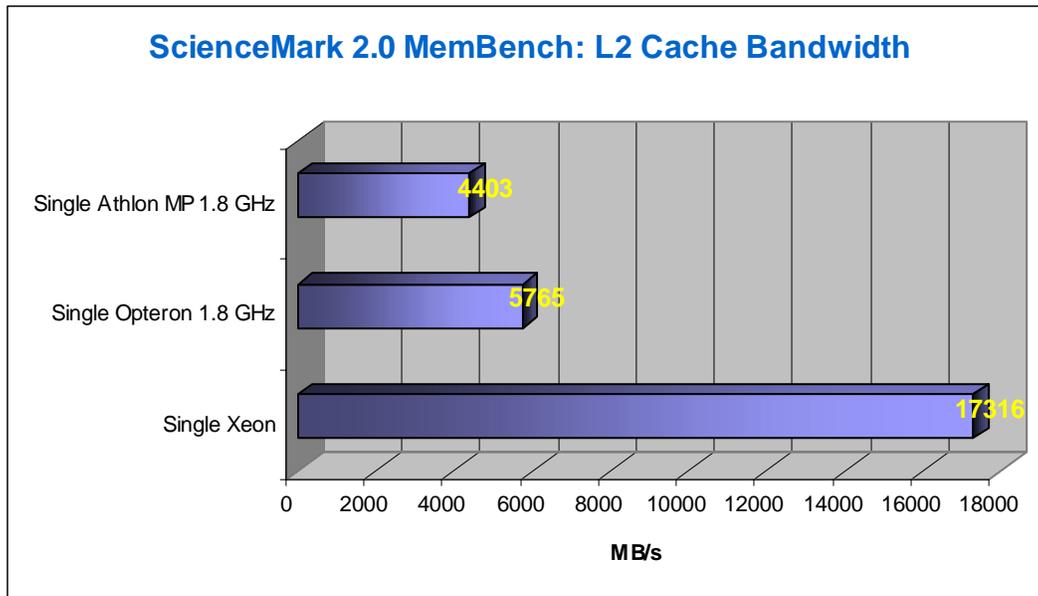


The 1.55V voltage indication is only for the Opteron 1.8 GHz. The BIOS of our test system suggested that an Opteron typically should use 1.45V.

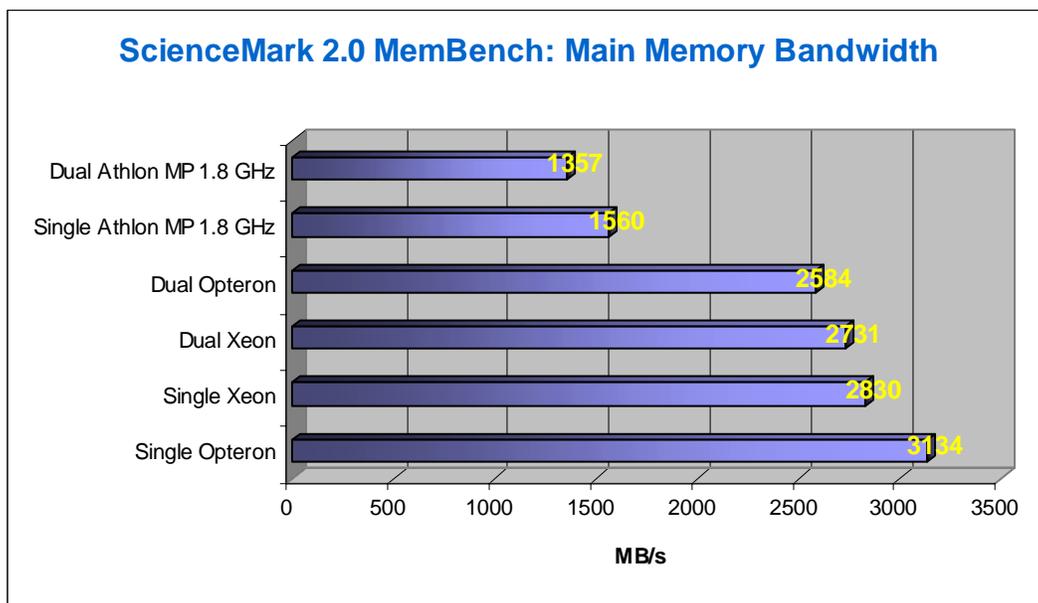


## Opteron Memory Bandwidth Analysis

PR specs are nice, but the truth lies in real measured data. We let Sciencemark 2.0 Membench running on a Windows 2000 Server - SP3 reveal the lowdown on the new cache and memory subsystem of the Opteron.

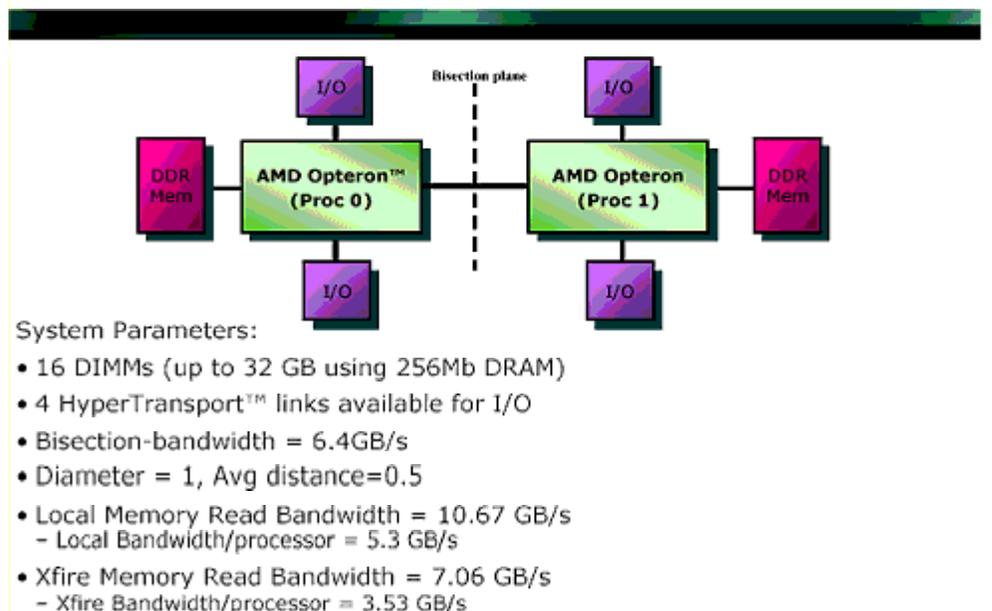


The L2-cache is clock for clock definitely better than the Athlon's by about 30%. This could indicate a 128 bit wide cache or a much better optimized 64-bit interface but AMD wouldn't tell us which. In all cases, it seems that another bottleneck of the Athlon architecture has been made a little wider.



Notice that the Xeon only performs 4% faster than the Dual Xeon system, but that the Opteron performs no less than 21% better when running in single CPU configuration. The reason is most likely the fact that the Windows 2000 Server does not support the (cc) NUMA - Non Uniform Memory Access ? architecture. (Find more information on how this works on the Opteron [here](#) and [here](#)). There are two possible memory accesses: local memory references which are fast (low latency, 5.3 GB/s bandwidth) and remote ones which are a lot slower (high latency, 3.2 GB/s bandwidth).

## Dual Processor System Topology



Windows 2003 does support the NUMA feature. This means that the OS is aware that remote references are slower and can optimize memory accesses by prefetching remote data from the remote memory subsystem, before it's needed. A simple BIOS update and an upgrade to Windows 2003 should enable NUMA. Bill Robbins at AMD told Ace's Hardware:

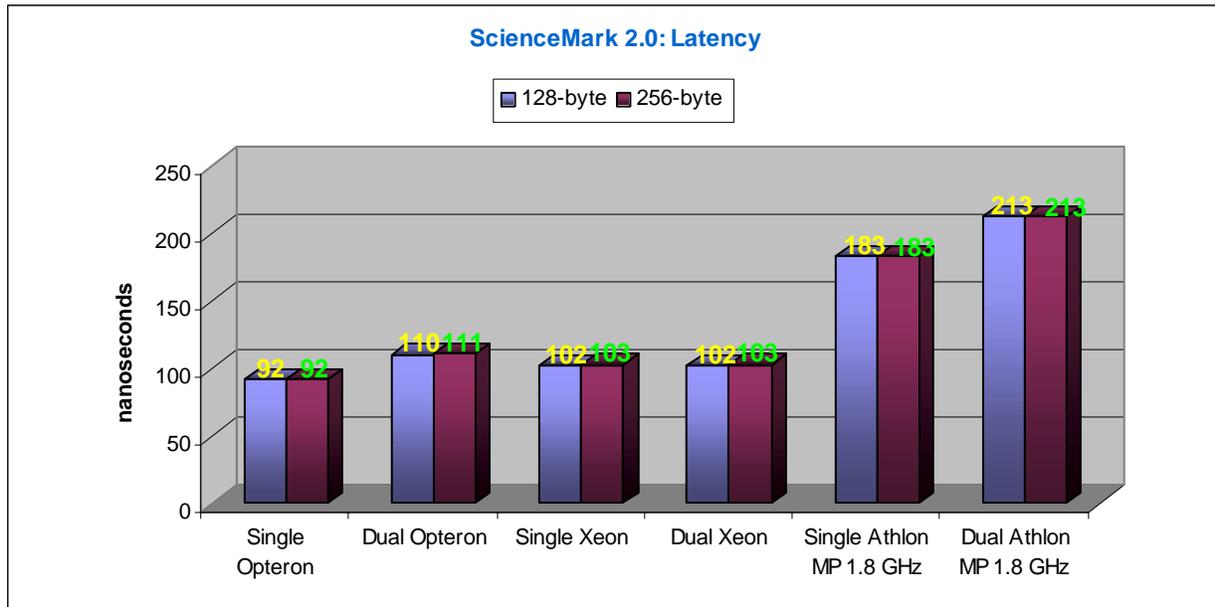
"An updated developmental BETA bios from Newisys takes advantage of the NUMA features provided under Microsoft Windows Server 2003 beta. We have found this BIOS significantly improves memory sub-system performance. And we fully expect as Newisys advances their BIOS development, the benefits of Newisys's NUMA-enabled BIOS will increase as well."

"NUMA is a new memory management feature from Microsoft that benefits AMD64 platforms by more efficiently managing system memory usage. The performance improvement is best seen where multiple memory controllers and multiple banks of memory are present, as with the Newisys / Opteron server you are testing. This Microsoft feature is a significant development that currently benefits AMD64 more than any other platform."

We may conclude that the performance of the Opteron should improve quite a bit once we run our benchmarks on (32 bit) Windows 2003 instead of Windows 2000 server. We will be testing this out and reporting back in a later review, but let's look at the latency of the Opteron's memory subsystem.

## Opteron Memory Latency Analysis

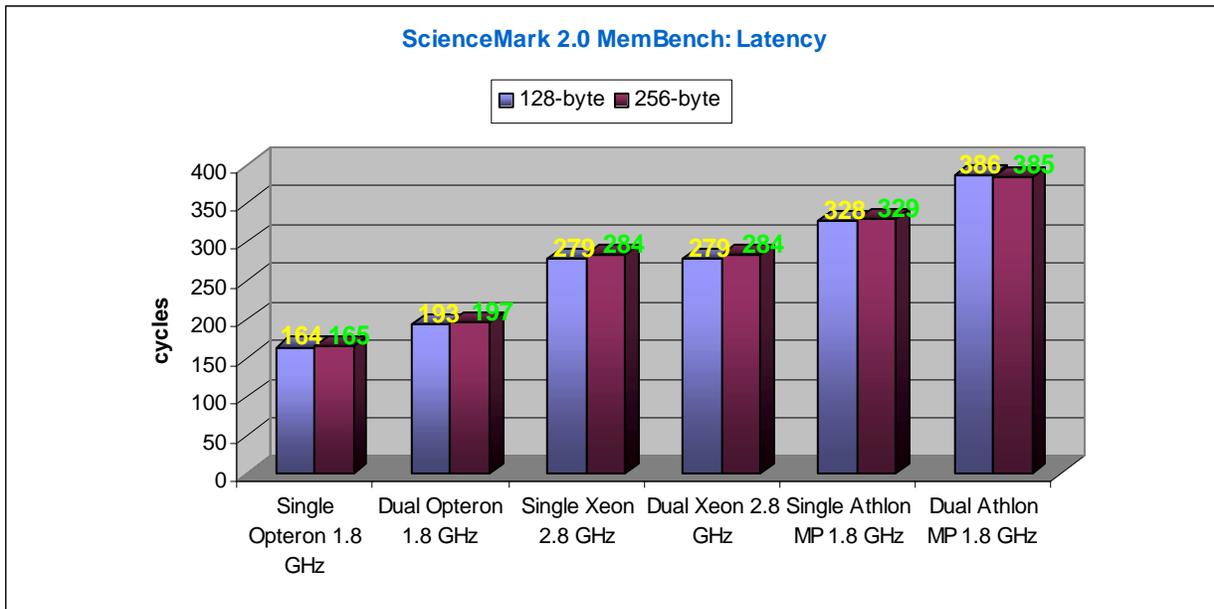
You might have noticed that we jotted a lower maximum L2-cache latency down for the Opteron (16 cycles) than for the Athlon in the overview table. Sciencemark Membench did indeed report that the maximum total Hammer L2-cache is indeed 16 cycles, an improvement of 4 cycles compared to the Athlon MP. Next, we investigated the influence of the integrated memory controller with Sciencemark. Note that we have expressed the latency in absolute time. On advice from Newisys, we did not remove the 2nd CPU to do the 1-way tests, but rather forced the system to prefer the first CPU, which gives slightly poorer results than the same system with just one CPU. We did the same thing for the Xeon.



The Opteron has about half the latency of the Athlon. Admittedly, the Athlon runs with DDR266 and the Opteron with DDR333. However, our previous measurements show that using DDR333 instead of DDR266 typically reduces latency by about 5-7%.

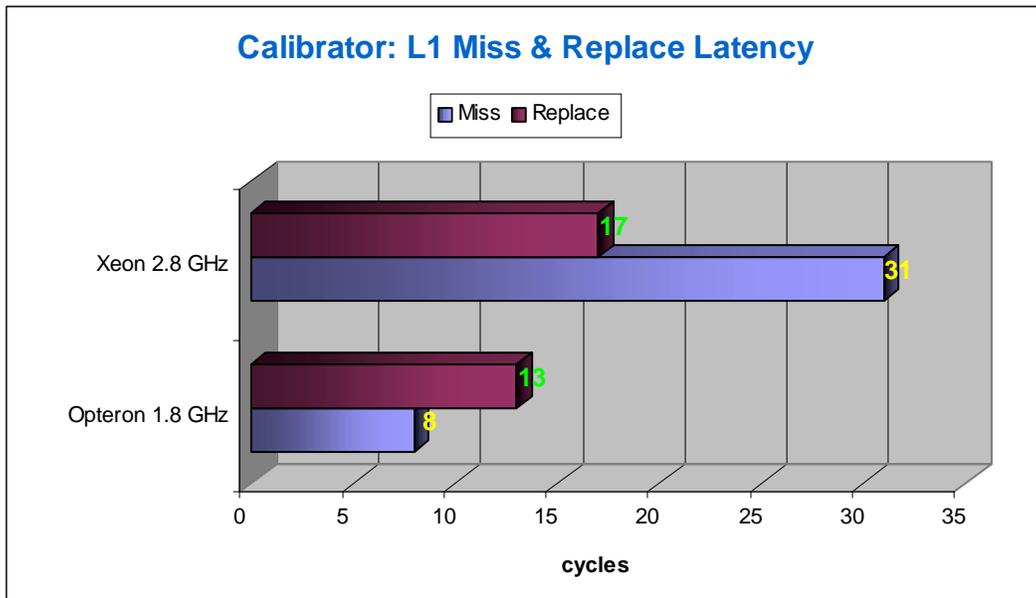
The integrated memory controller is impressive, but again, Windows 2000 cannot take full advantage in dual CPU mode. This along with a remarkable performing Intel E7501 chipset make sure that the latency advantage of the Opteron over the Xeon is relatively small (about 10%).

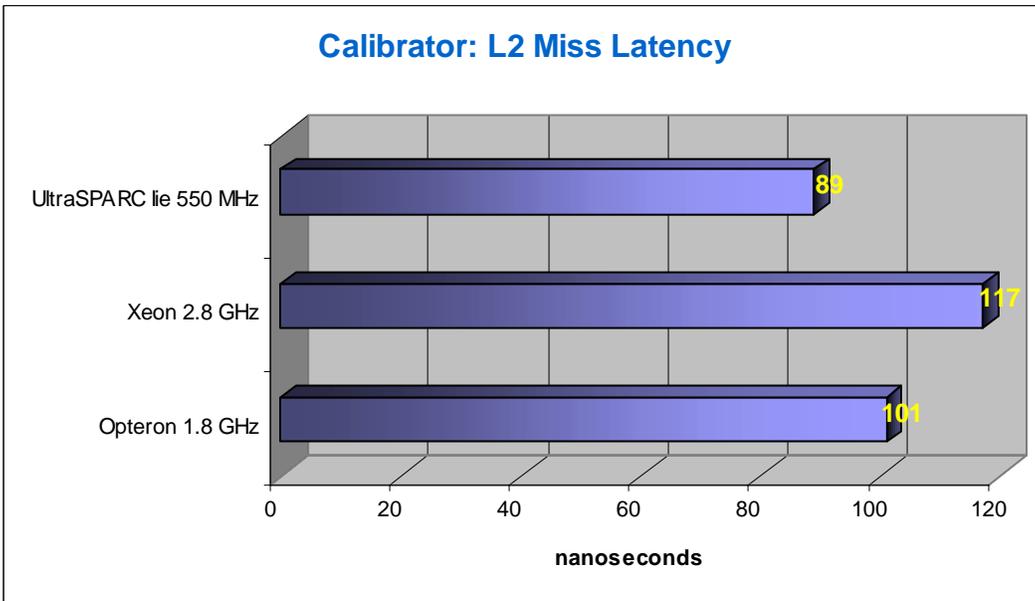
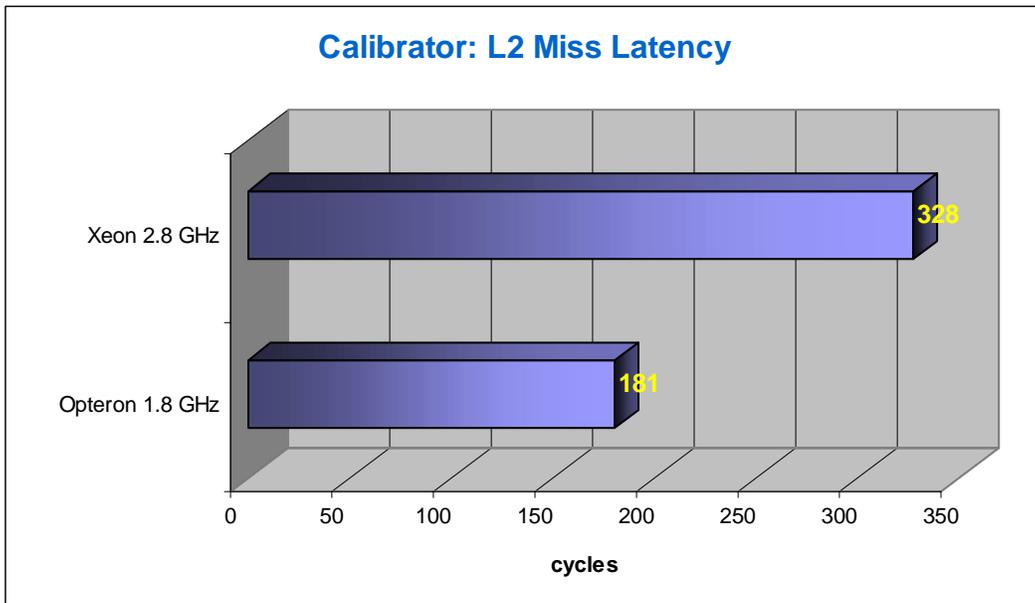
It is completely different matter, of course, when you look at latency from the point of view of the CPU. Just how many cycles does the CPU wait to get the data that it needs ASAP?



The Opteron wastes a lot fewer cycles when a memory access stalls the CPU.

We also tried out Calibrator v0.9e, written by by **Stefan Manegold**. Note: Please see the Calibrator graphs in the Appendix for more detailed results.





The Calibrator results seem to indicate a 16% better memory latency for the Opteron. The integrated memory controller makes a difference, but it is not earth shattering. Additionally, we have another result in this test, from a 550 MHz UltraSPARC IIe. The USIIe is designed for single-processor systems, but is interesting for comparison purposes here because, like the Opteron, it too features an integrated memory controller. In fact, we see that the L2 miss latency of the USIIe is slightly lower than that of the Opteron (12%), though it is expected the latency will be lower to some extent when comparing a single-processor system to a dual-processor one.

## Newisys: AMD's Beachhead into the Server Market

Entering the server market demands more than a CPU, it requires building up a whole platform. AMD was smart enough to see that they wouldn't be able to pull it off alone. AMD's Hammer presentation excited quite a few people at the server departments of IBM and DELL. Newisys was founded in August 2000 with the goal of creating then necessary platform for AMD's SledgeHammer.

The objective of Newisys is not to sell rack-mountable servers directly to customers, but instead to sell their reference designs to Tier 1 and 2 OEMs which will then market those servers under their own name. In other words, Newisys designs the motherboards, the custom chips, the system management software, and puts it altogether to create a complete Opteron platform. OEMs are free to adopt some of the components, or buy the whole server design and put a different label on it. Newisys is a testimony to the fact that many people in the server industry believe that the AMD Opteron could make a serious difference in server market. When you take a look at their management page, you can see that many of the Newisys people are server technology veterans, who held high positions at IBM and DELL. The management of Newisys took a risk leaving those safe and well-paid positions for a startup built around a completely new platform.

The big question is whether or not a Tier 1 OEM signed up. The answer for the moment is "No," but these things take time. The enthusiasm of Tier 2 vendors is wild though: Avnet Applied Computing, Appro, RackSaver and Microway will all sell Newisys-based Opteron servers and are pretty excited about it.

### The Newisys 2100 Server

The first Newisys product, the system we have tested, is called the Khepri. When I heard the name, I knew I heard it this name before. Khepri was the Egyptian god that pushed the sun across the sky, the god that made the sun rise. In a poetic sense, you could say that this khepri server will push AMD's sun in the server sky. But then again, Khepri was symbolized by a dung beetle (scarab)... they certainly have a sense of humor at Newisys.



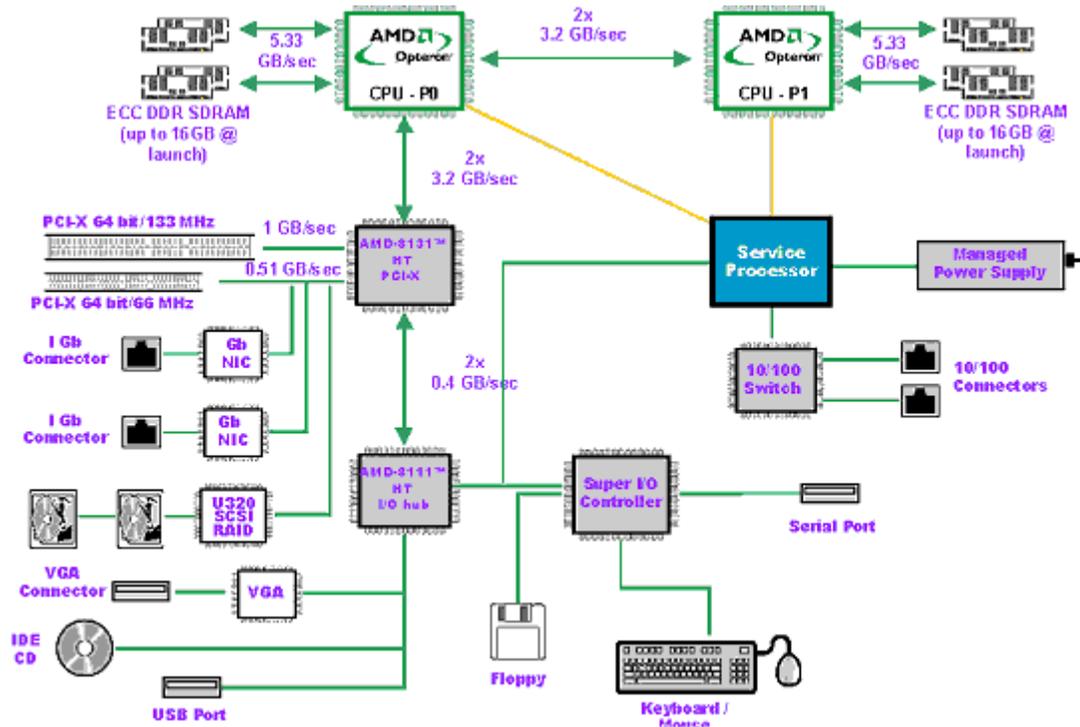
All joking aside, the Khepri server or Newisys 2100 is a 1U dual Opteron system which can support the following:

- A maximum of 16GB of DDR-333 ECC (4 Slots on each CPU)
- One full length 133MHz PCI-X slot
- One half length 66MHz 64-bit PCI-X slots
- Two Gigabit Broadcom 5703 ethernet connections
- Two internal ATA or hot plug U320 SCSI drives
- An embedded service processor for advanced system management.
- Two 10/100 Mbit connections to the service processor,

The system management functionality is probably one of the most advanced features to be found in any 1U server on the market. You give the service processor a static IP or an automatic DHCP IP and you are then able to control the server remotely from a web browser anywhere on your network. No need to install software, and completely OS independent.

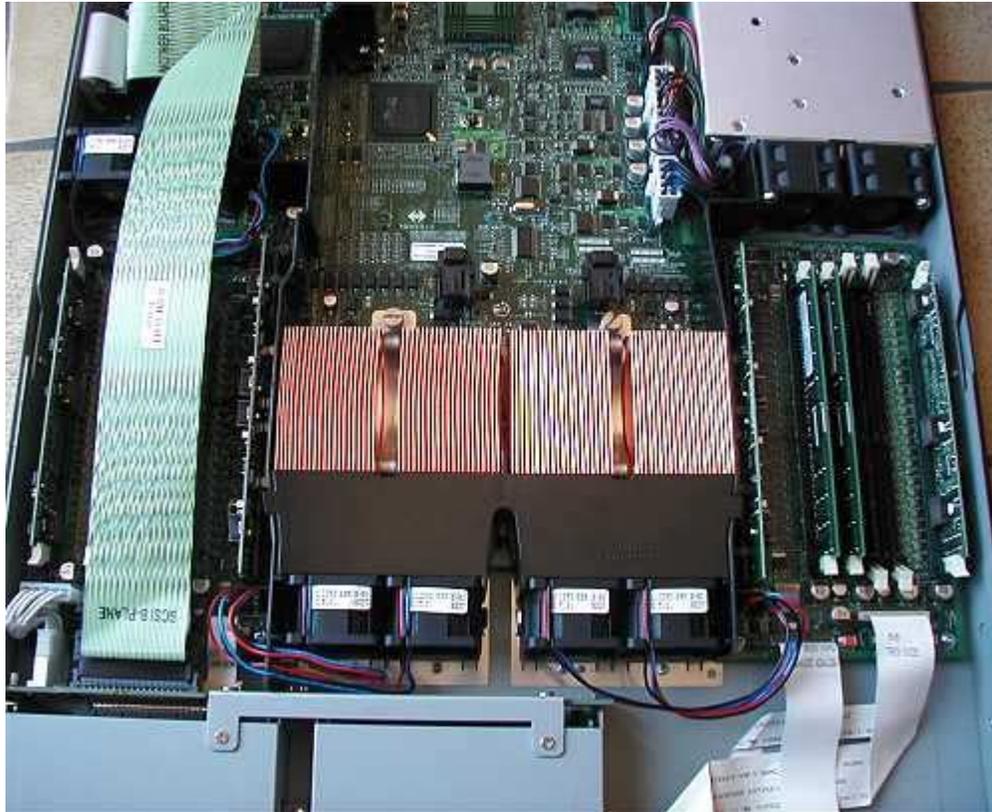
The service processor runs an embedded Linux OS, and the system management software was written with standard Linux tools. This means that anybody who licenses the management software can easily add their own code. Now most rack servers do come with management software, but you often have to buy separate proprietary modules. And few of these management systems can act as an SNMP agent, manage active directory services and NIS. We have not been able to test this software in-depth as we haven't had enough time as of yet. Of course the service processor also handles OS reboot, system health, fans and thermal diagnostics, power and BIOS setup and updates.

Below you can find a schematic overview of the 1U dual Opteron server.

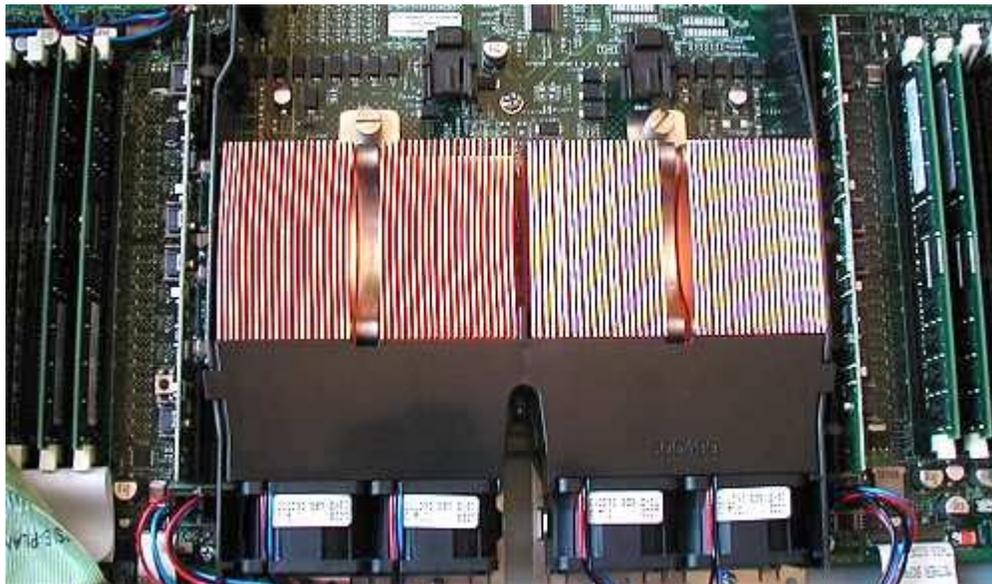


It is important to note that Newsys offers 4 DIMM slots alongside each processor. Quite a few motherboard companies **designed their motherboard with 4 slots for the first processor** and two for the second. But as the Dual Opteron works best (with a NUMA aware OS) with an equal amount of memory connected to each processor, the Newsys board is a lot more flexible.

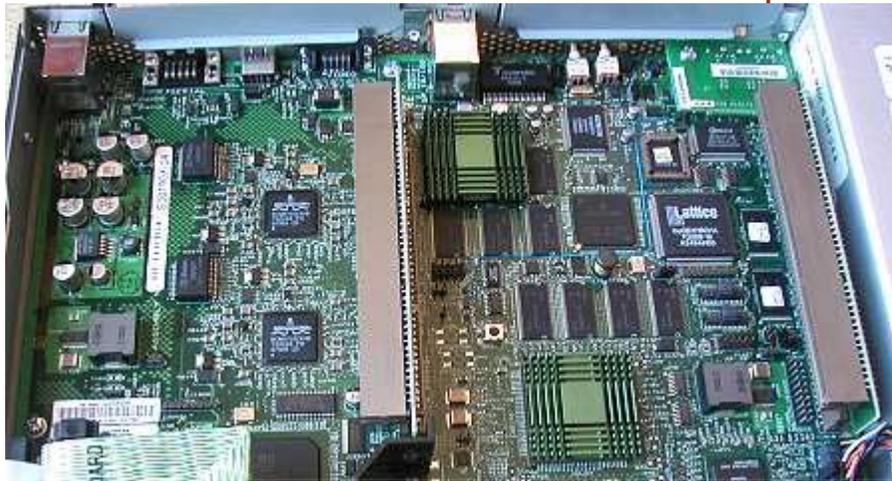
## What Does it Look Like?



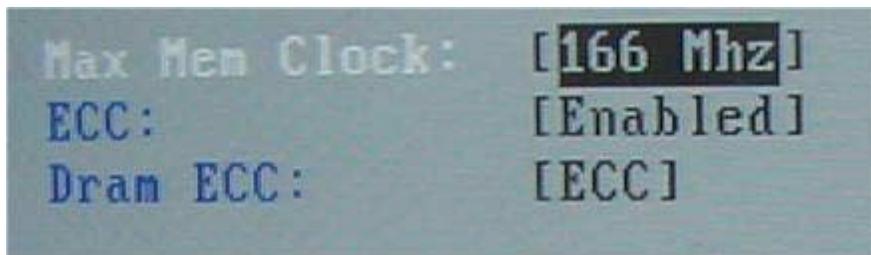
When we push away the Ultra320 SCSI cable, you can clearly see that each processor has access to two DIMM slots of 512 MB ECC PC2700 DDR SDRAM.



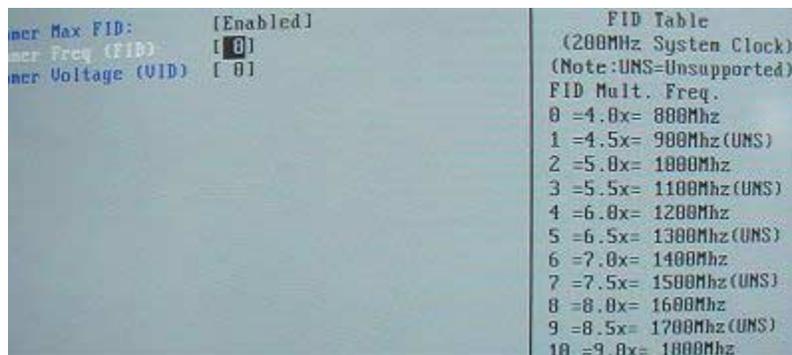
At the end of the server we find two PCI-X slots, one at 133 MHz, one at 66 MHz.



In the BIOS, it was possible to set the speed of the memory...



...and of CPU.



As we didn't want to risk any stability issues and because an overclocked server is a rare thing to begin with, we did not try any overclocking at this point in time. But it seems to be possible: disable Max FID, and then enter the appropriate code in Hammer Freq (FID). While the table stops at 10 (1.8 GHz), it was possible to enter a number from 11 up to 31.

## The Competition

We needed a good competitor for our Dual Opteron system, and luckily Nicole Chia of Gigabyte was able to provide us with a worthy adversary. The **Gigabyte GA-8IPXDR-E** was the stable E7501 board we were looking for.



The board supports a maximum of 12GB of DDR266 ECC memory, comes with a dual Adaptec 7902W SCSI Controller and also features two ATA-100 channels. Four PCI-X slots are available, each of which can be set to 133/100/66 MHz. Even more importantly for our server tests was the fact that a dual 1 Gbit Intel RC82546EB LAN chip was available, linked to the PCI-X bus. The board also supports Console Redirection, we'll discuss it and several other server boards in more detail in an upcoming review.

## Benchmarked Configurations: Hardware

We will discuss the software settings for each of the tests as we used different configurations for HPC, webserver and database server tests. The desktop was set in Windows at a resolution of 1024x768x32bpp with at a 75 Hz refresh rate. In Linux OS, we always used the command line.

A very special thanks to Shelley Baldiga (**Crucial**), who made this review possible by sending us 2x 512 MB of the best quality **Crucial PC2100R-ECC Buffered RAM**.

**Server 1: Intel Pentium 4 2.4 GHz, 2.8 GHz, 3.06 GHz (Hyperthreading enabled)**

- Gigabyte GA-8INXP (E7205/ "Granite bay Chipset) - Dual DDR266
- 1.5 GB: 2x512 MB and 2x 256 MB Corsair PC3200 XMS (DDR-SDRAM) running at 266 MHz CAS 2 (2-2-2-6)
- NIC: 1 Gb Intel RC82540EM - Intel E1000 driver.

**Server 2: Dual AMD Athlon MP 2200+**

- Tyan Tiger MPX

- 2 GB: 4x512 MB Crucial PC2100R - 250033R (2.5-3-3-6)
- NIC: 1 Gb Edimax EN-9230TX-32 bit PCI - Standard Debian/Suse Driver

**Server 3: Dual AMD Opteron 244 (1.8 GHz)**

- Newisys Khepri
- 2 GB: 4x512 MB Infineon PC2700R - 250033R (2.5-3-3-6)
- NIC: Broadcom 5703, bcm5700 driver

**Server 4: Dual Xeon DP 2.8 GHz - 533 MHz FSB**

- Gigabyte GA-8IPXDR-E
- 2 GB 4x512 MB Crucial PC2100R - 250033R (2.5-3-3-6)
- NIC: 1 Gb Intel RC82540EM - Intel E1000 driver.

**Client Configuration 1: 1x Dual Xeon 2.8 GHz - 533 MHz FSB**

- MSI GNB MAX FISR (E7205)
- 2x256 MB Crucial PC2100R - 250033R (2-2-2-6)
- NIC: 1 Gb Intel RC82546EB - Intel E1000 driver.

**Clients Configuration nr 2: 4 x Duron 1300 pcs.**

- ECS K7S5A
- 384 MB PC133 SDRAM.
- NIC: 100 Mbit RTL 8139, standard driver

**Shared Components**

- Seagate 36 GB - 320 MB/s SCSI
- Maxtor 80 GB DiamondMax 740X (7200 rpm, ATA-100/133)

**Software**

- Intel chipset inf update 5.09.1012

We'd like to thank the following helpful people for their support and important contributions to this review:

- Damon Muzny, Bill Robins (**AMD**)
- Kevin Baker, Bill Zipoy (**Newisys**)
- Andreas Jaeger (**Suse**)
- Nicole Chia (**Gigabyte**)
- Kristof Semhke (**Intel**)
- Angelique Berden (**MSI**)
- Shelley Baldiga (**Crucial**)
- Robert Pearce (**Corsair**)

Let's see some benchmarks!

## The Opteron as a Webserver

Testing webservers is a complex undertaking. Luckily my real job is teaching PC technology to the bachelor and **Multimedia Communications Technology** (MCT) students at the **PIH**, a bachelor degree university in Kortrijk, Belgium. Two students, Tom Gilis and Ben Boogaerts, were currently assisting me in my investigation of (web)server benchmarks. I like to thank Tom and Ben for their assistance, and Lode De Geyter, manager of the PIH, for letting us use the infrastructure of the PIH to test webservers and database servers.



*Tom and Ben ready to test the Opteron*

Our first test was to actually mimic the test that Brian performed a while ago. The difference was that Brian's test machine was a 300 MHz Ultra 30 (2 MB L2 cache) running Solaris 9, with 386 MB of memory, which could be maxed out with a relatively low amount of network traffic. Our dual processor monsters were quickly running into performance walls.

The webservers were tested in two configurations: with 4 to 5 clients (client 1 configuration: Duron 1300 PC) connected to one of the eight 100Mbit ports of our gigabit switch. This means that we should be able to push about 40 to 50 MB/s of client traffic to our server which is hooked up to the gigabit uplink.



*4 clients connected to the Longshine gigabit uplink switch*

Our second configuration consisted of one client (client 2: P4 2.8 GHz PC) connected on the integrated gigabit Intel RC82540EM via a UTP5 cross over cable to our servers and their gigabit connectors. In this configuration it was theoretically possible to push about 100 MB/s of client traffic to the server. This way we could see whether or not extra clients or higher network bandwidth could push our servers higher.

The 2.8 GHz Pentium 4 client was loaded with a Debian Linux kernel version 2.4.20-686-smp, which is a Pentium 4 Hyperthreading-optimized version, despite its 686 name. The Duron clients were running Debian Linux kernel version 2.4.20-k7-smp, which is an Athlon optimized version of the stable 2.4 Linux kernel.

For benchmarking, **httperf** was used in conjunction with **autobench**, a Perl script written by Julian T. J. Midgley, designed to run httperf against a server several times, with the number of requests per second increasing with each iteration. The output from the program enables us to see exactly how well the system being tested performs as the workload is gradually increased until it becomes saturated.

```

Opened connection to 192.168.0.5:4600
Command: --server 192.168.0.1 --uri "/read.php?article_id=1" --num-conns 3000 --
-num-call 10 --timeout 5 --rate 490 --port 80
Time : 1838724650
Issued start command to: 192.168.0.3
Issued start command to: 192.168.0.4
Issued start command to: 192.168.0.5
192.168.0.3 accepted command
192.168.0.4 accepted command
192.168.0.5 accepted command
Testing server 192.168.0.1 at rate: 1500
Opened connection to 192.168.0.3:4600
Opened connection to 192.168.0.4:4600
Opened connection to 192.168.0.5:4600
Command: --server 192.168.0.1 --uri "/read.php?article_id=1" --num-conns 3000 --
-num-call 10 --timeout 5 --rate 500 --port 80
Time : 1838724660
Issued start command to: 192.168.0.3
Issued start command to: 192.168.0.4
Issued start command to: 192.168.0.5
192.168.0.3 accepted command
192.168.0.4 accepted command
192.168.0.5 accepted command
Benchmark complete
debian2:/XeonP4# lynx 192.168.0.1/Spades/read.php?article_id=71
    
```

*Autobench and httperf in action*

In each case, the server was benchmarked with 5 requests per connection. The benchmark was configured to timeout any request that takes longer than 5 seconds to complete, 10 seconds in case of the "compressed message board index."

However we had to recompile httperf, as it was running on the clients with only 1024 file descriptors, an old unix open file per user limitation. Our own httperf does not have that limitation.

The clients were thus set to:  
 ulimit -n 10000, set number of open files to 10000 (default 1024)

The Dual Athlon server used the Debian Linux kernel version 2.4.20-k7-smp, the Dual Xeon Debian Linux kernel version 2.4.20-686-smp and the Opteron the SUSE SLES 8 kernel 2.4.19. As these kernels are the most stable ones which are optimized for each of our server CPUs, we believe this is a fair comparison. The same kernel on different distribution will perform very similar if you tune the right parameters such as we have done.

```

opteron:~ # cat /proc/version
Linux version 2.4.19-64GB-SMP (root@SMP_X86.suse.de) (gcc version 3.2) #1 SMP
    
```

*The 32 bit SUSE on the Opteron supports up to 64 GB via PSE*

The servers were tweaked with the following parameters:

- `ulimit -n 64000`, set number of open files to 64000 (default 1024)
- `shmmx = 512288000` or 512 MB, shared memory maximum (default 33 MB)
- `net.core.optmem_max=100000`, maximum amount of option memory buffers, default 10240
- `net.core.rmem_default=131071`, default receive socket buffer size, default 65535
- `net.core.rmem_max=131071`, maximum receive socket buffer size, default 131071
- `net.core.wmem_default=1000000`, default send socket buffer size, default 65535
- `net.core.wmem_max=1000000`, maximum send socket buffer size, default 131071

When comparing our benchmark methods with others, you will see that you'll see that these 3 following parameters are turned off.

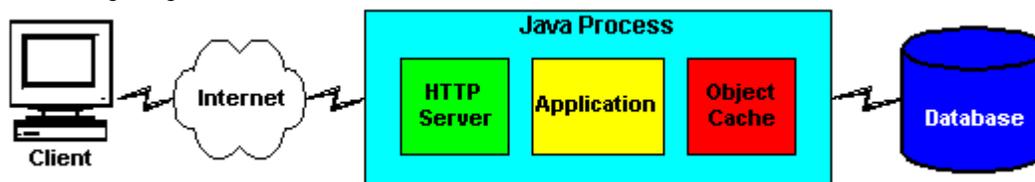
- `net.ipv4.tcp_timestamps=1`, turns TCP timestamp support on, (default)
- `net.ipv4.tcp_sack=0`, turn SACK support on, (default)
- `net.ipv4.tcp_window_scaling=0`, turn TCP window scaling support off, (default on)

We decided to leave them on as as these features allow a webserver to perform well on a high latency WAN. Yes, we were testing on low latency LAN, but webserver are mostly deployed on the Internet, one of the highest latency WANs or as Intranet server on very big LANs.

The benchmarked software includes:

- **Caucho Technology's Resin 2.1.6**
- **Java Virtual Machine 1.4.1\_02 SDK**
- **Sybase ASE 11.9.2 for Linux**

These benchmarks test the production Ace's Hardware application with the exact same dataset. The full platform is described in the following diagram:



To understand this better, I recommend your read **Brian's article "Scaling Server Performance"**. Basically, the clients run `httperf`, which generates a lot of "HTTP GET" requests. Looking at the diagram below, the "Internet" is in our case a crossover cable, or the gigabit switch. Resin or another webserver (green) accepts the requests and transmits the response back to the client. The Java application (yellow) fetches data from the object cache (red) and generates the response to be sent back to the client via the HTTP server. The cache consists of data contained in the database (blue).

## Cached Java Webserver Benchmark: Message Board Index

OS: Linux 2.4.20 / 2.4.19

Multi - threaded: Yes

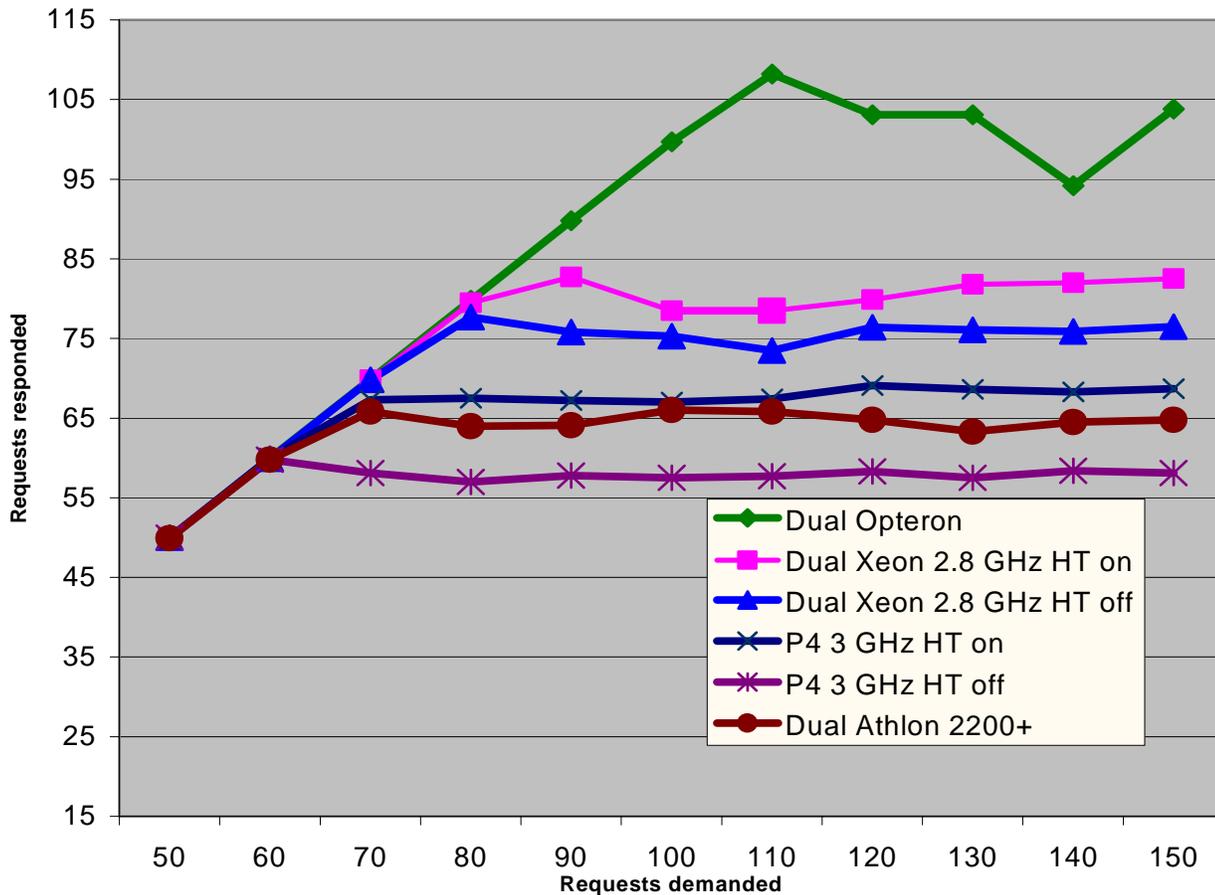
Memory Usage: 800 - 850 MB

Typical Error Margin Between Different Runs: 3-4%

Maximum Network Traffic: up to 15 MB/s

Our first test is very CPU intensive one, as it requests the index of our message board. As the content is not compressed, it requires a lot of network traffic.

Java webserver response rate: Messageboard index



The Opteron sweeps the floor with the competition. Running Java applications and building threaded message indexes at the same time seems to be one of its favorite tasks. The Opteron is up to 38% faster than the Dual Xeon, the far behind second in this test. A Dual 3 GHz Xeon would perform probably about 7% better (performance scales well with clockspeed), but would not touch the Opteron's performance.

As the message index can be several hundred kilobytes in size and is generated from an array of thousands of Java objects, this can be a very cache and memory intensive application. When we consider the advantage the Opteron has thanks to its 128-bit per-CPU memory interface and large 1 MB L2 cache, it's performance in this test becomes clear.

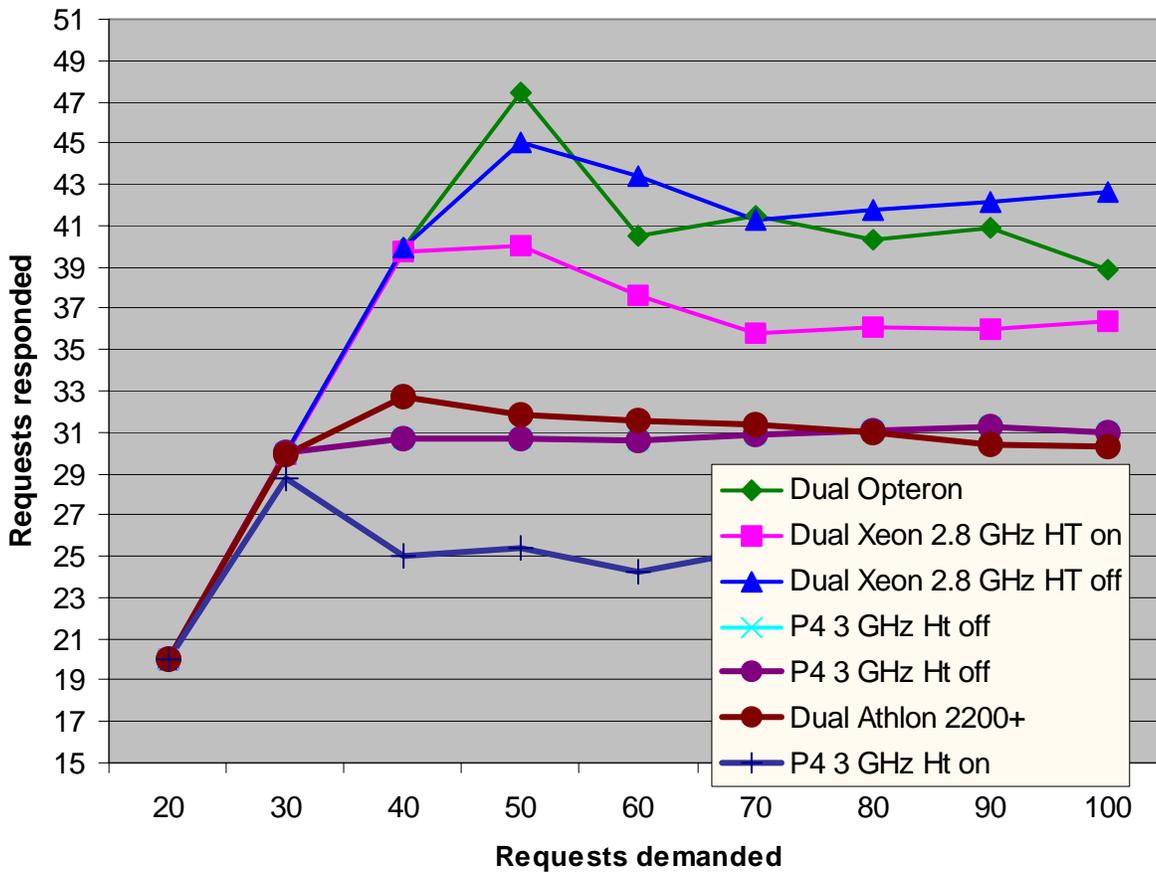
Hyperthreading helps, of course, as this test is highly multi-threaded but the generating the message index is the most important bottleneck.

## Cached and Compressed Java Webserver Benchmark: Message Board Index

OS: Linux 2.4.20 / 2.4.19  
 Multi - threaded: Yes  
 Memory Usage: 850- 900 MB  
 Typical Error Margin Between Different Runs: 2-3%  
 Maximum Network Traffic: up to 1 MB/s

Fast java performance is great, but one of the main concerns of a webmaster is bandwidth. Costs scale sometimes higher than linear with bandwidth consumption, so excessive bandwidth usage can be an expensive proposition. So, why not compress content on the server with gzip before sending it to the client? Almost every browser out there supports **gzipped** content.

Java Gzip compressed response rate: Messageboard index



In this test, we are compressing the pages before sending them to the client. The added computational demand of the compression reduces performance by half, but it also reduces the network I/O to 1/15<sup>th</sup> of its compressed size. Performance may be lower, but the bandwidth savings make up for it ten times over, and we can always make up the difference in performance by buying faster hardware with all the money saved on bandwidth charges.

Now you might recall that the Pentium 4 architecture likes most compression schemes, and gzip is no exception. Whereas the previous uncompressed benchmark was dominated by size of the data, compressing the output serves to shift this to a more processing-bound test. The result is that the 2.8 GHz P4 Xeon is able to close the gap with the Opteron in this test.

Interestingly, performance decreases when Hyperthreading is enabled. Both the P4 and Xeon perform worse with Hyperthreading. A possible explanation is that gzip eats up quite a bit of the L2-cache, taking away space from the Java threads. Compression also keeps more units busy which lowers the amount of idle execution power that Hyperthreading can utilize.

## Cached and Compressed Java Webserver Benchmark: Dual Mac Article

OS: Linux 2.4.20 / 2.4.19

Multi - threaded: Yes

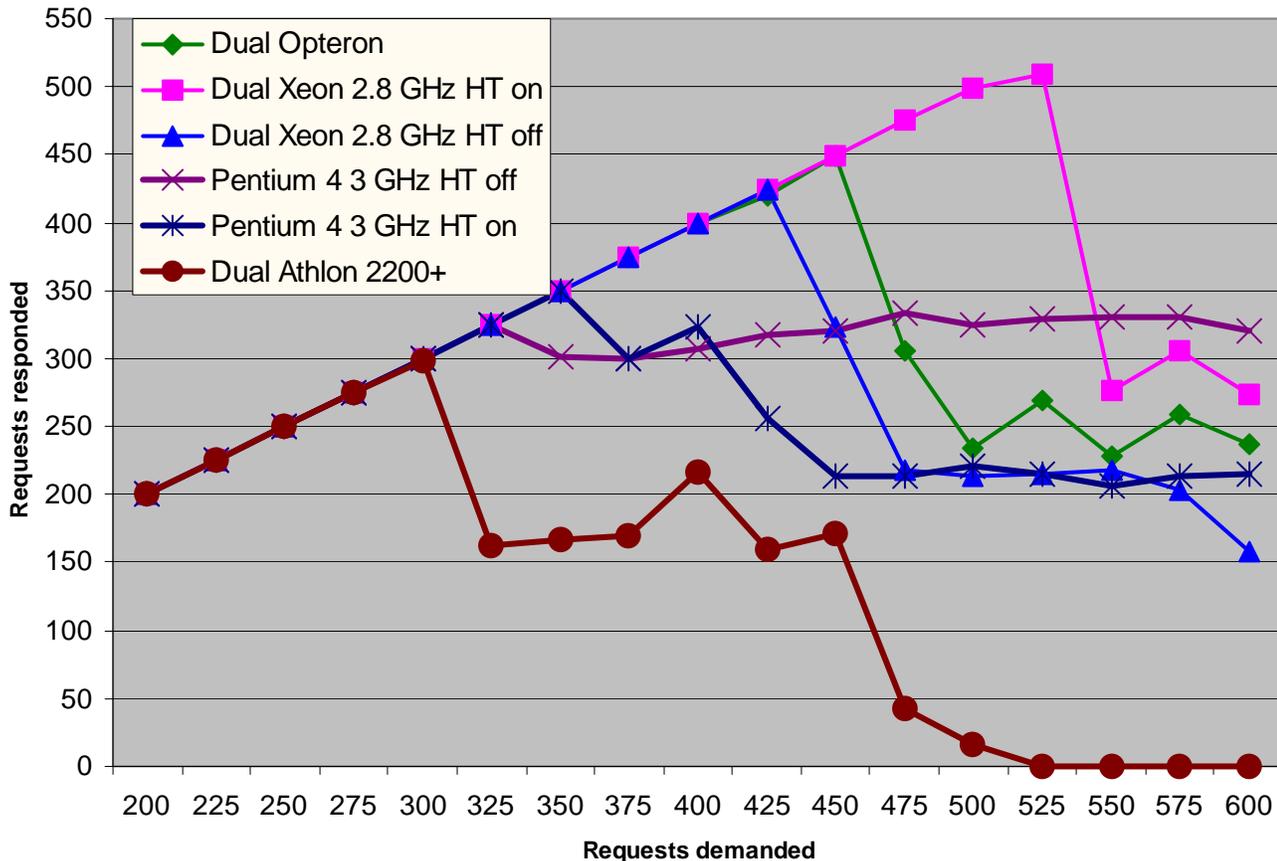
Memory Usage: 750 - 800 MB

Typical Error Margin Between Different Runs: 5-7%

Maximum Network Traffic: up to 2.5 MB/s

In the next benchmark we do not have the index/tree structure to generate, we simply request a 18 KB article, namely **A Quick Look at the Fastest Apple PowerMac**.

Java Gzip compressed response rate: article



Now that we removed the burden of building the message board index, the number of threads that can be processed increases enormously. Which means that the main performance factor shifts to how fast a CPU can tackle a lot of java threads. The dual P4 Xeon with its Hyperthreading can tackle 4 threads at once and outperforms the rest of the pack. The dual Opteron turns in the second best performance, pulling ahead of the dual Xeon with Hyperthreading disabled.

## Opteron Performance in SPECjbb2000

**OS:** Linux 2.4.20 / 2.4.19

**Multi - threaded:** Yes

**Memory Usage:** 1000 - 1150 MB

**Typical Error Margin Between Different Runs:** 1-3%

**Maximum Network Traffic:** N/A, directly on server

The SPECjbb2000 is a software benchmark product developed by the Standard Performance Evaluation Corporation (SPEC), and it simulates a transaction processing server. Let SPEC tell you more:

SPECjbb2000 is implemented as a Java program emulating a 3-tier system with emphasis on the middle tier. All three tiers are implemented within the same JVM. These tiers mimic a typical business application, where users in Tier 1 generate inputs that result in the execution of business logic in the middle tier (Tier 2), which calls to a database on the third tier. In SPECjbb2000, the user tier is implemented as random input selection. SPECjbb2000 fully implements the middle tier business logic. The 3rd tier is represented by binary trees rather than a separate database.

SPECjbb2000 is totally self contained and self driving (generates its own data, generates its own multi-threaded operations, and does not depend on any package beyond the JRE).

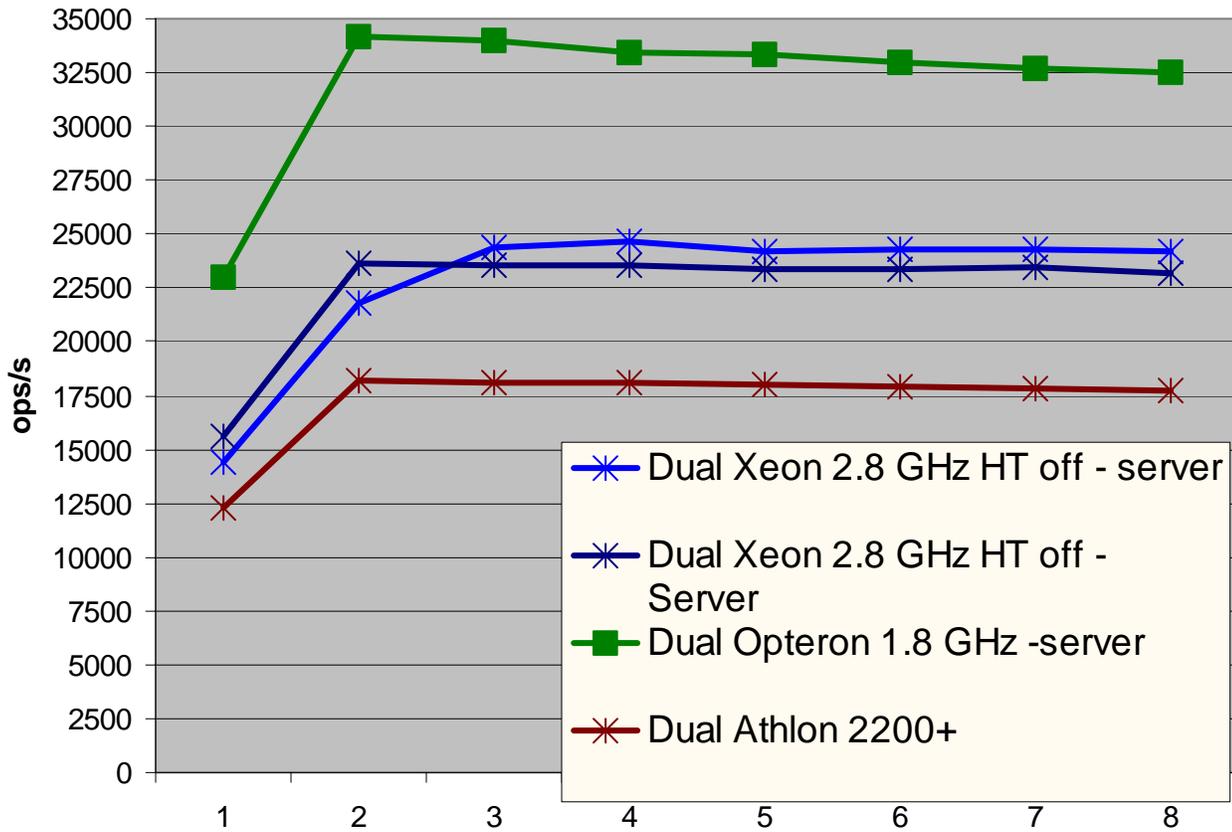
SPECjbb2000 is inspired by the TPC-C benchmark and loosely follows the TPC-C specification for its schema, input generation, and operation profile. SPECjbb2000 replaces database tables with Java classes and replaces data records with Java objects. The objects are held in memory by either BTrees (also Java objects) or other data objects. Therefore SPECjbb2000 does no disk IO. Since there is no database, it does not support object persistence with ACID properties corresponding to a RDB implementation. SPECjbb2000 uses only Java synchronization to synchronize multi-threaded access to shared objects in the population. Since users do not reside on external client systems, there is no network IO in SPECjbb2000.

"While SPECjbb2000 is inspired by TPC-C, it is in no way comparable. SPECjbb2000 is memory resident, uses totally different data set sizes, mix of workloads, performs no I/O to disks, and has no think times. It has a different set of run and reporting rules, a different measure of throughput, and a different metric.

In SPECjbb2000, there is only one terminal (or customer) active per warehouse. A warehouse is a unit of stored data. It contains roughly 25 MB of data stored in Btrees. Terminals map directly to Java threads. Each thread executes operations in sequence, with each operation selected from the operation mix using a probability distribution. As the number of warehouses increases during the full benchmark run, so does the number of threads.

SPECjbb2000 uses at least a heap size of 198 to 300 MB, so we gave it 1024 MB by adapting the -Xms/-Xmx JVM options. We also made sure the JVM was actually the server JVM and not the client JVM which is also included in the 1.4.1\_02 HotSpot Java SDK.

### SPECjbb 2000 1.02 - server



*SPEC is a registered trademark of the Standard Performance Evaluation Corporation ([spec.org](http://spec.org))*

The Opteron has a massive 41% lead on its competitor and proves to be a very good transaction processing CPU.

## RDBMS Performance

So far we have tried to minimize database access as much as possible for better web performance. However, we can imagine that many people would like to know how the Opteron and Xeon handle Relational Database Servers, like MS SQL Server and the free open source database MySQL.

## Microsoft SQL Server 2000 SP3

**OS:** Windows 2000 Server SP3

**Multi - threaded:** Yes

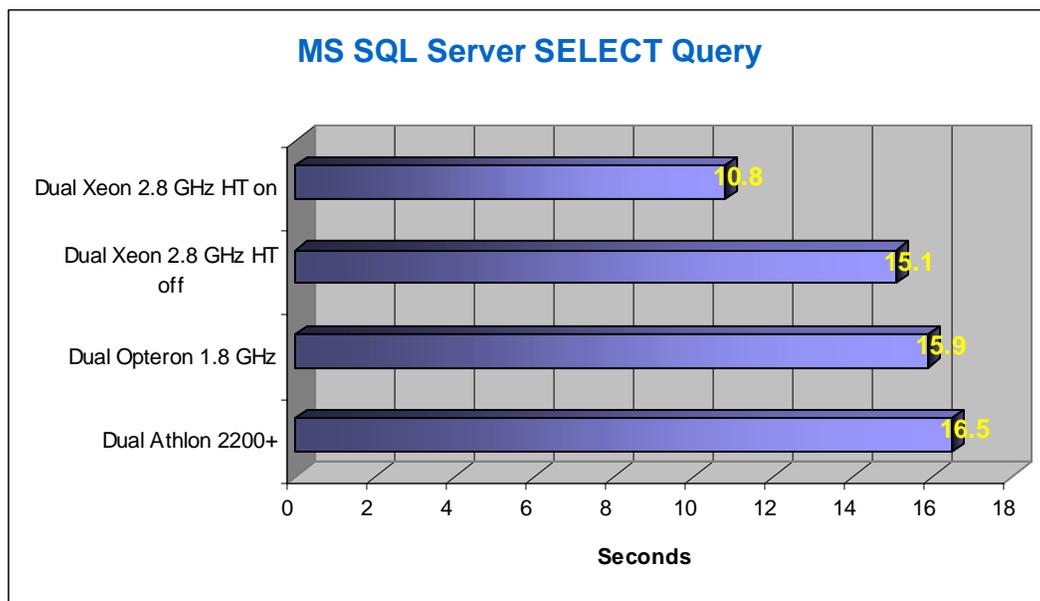
**Memory Usage:** up to 980 MB

**Typical Error Margin Between Different Runs:** 1-2%

**Maximum Network Traffic:** N/A, directly on server

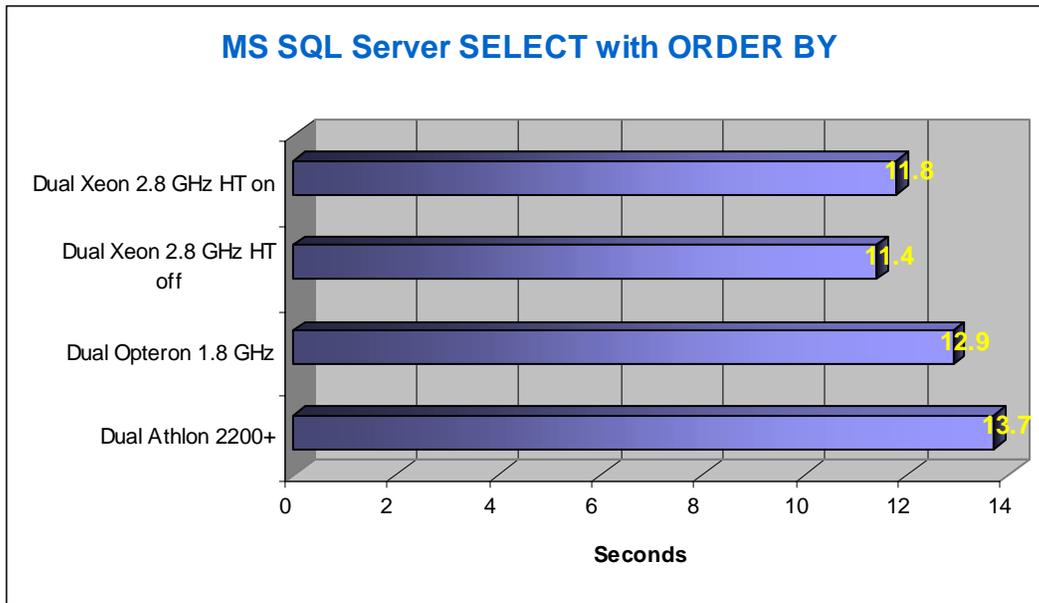
For our MS SQL test, we imported the 5 GB of database of medium sized company. The database consist of a "person table" which contains about 460,000 unique rows and which has 1 to n relation with the "document" table, which is good for no less than 800,000 rows. Furthermore, there is another table which essentially consists of orders, and is a few 100,000 rows in size. Finally, there is also a "value" table that contains data referenced by foreign keys in the other tables. A simple example of this is a table that contains the strings "Male" and "Female." The rows in this table are referenced by a key in others (like the "person" table). To run a query on this data, we JOIN the columns we want using keys, and specify various constraints in a WHERE clause. The database then generates a temporary table with the requested data in the requested order and sends us the results. For the first test, we query rows created within a specific date range where the name column begins with "V." We can not show the actual SQL query as we agreed that we would not disclose the actual contents of this massive database. The query has three JOINS, a partial string search, and a date search constraint.

A SELECT is performed 4 times, and we discard the first result as it is bottlenecked by the hard disk. The next three runs are very repeatable and the times required to complete each one are averaged together. Each time we run the query we perform a "dbcc freeproccache" command to make sure that the results can not be delivered directly from the OS memory cache. While the databases stay in the memory of the OS (almost no disk access), the actual resulting recordset must be generated again. Each CPU reaches a utilization that fluctuates between 60 and 90% (most of the time between 80-85%). The query returns 8800 rows.

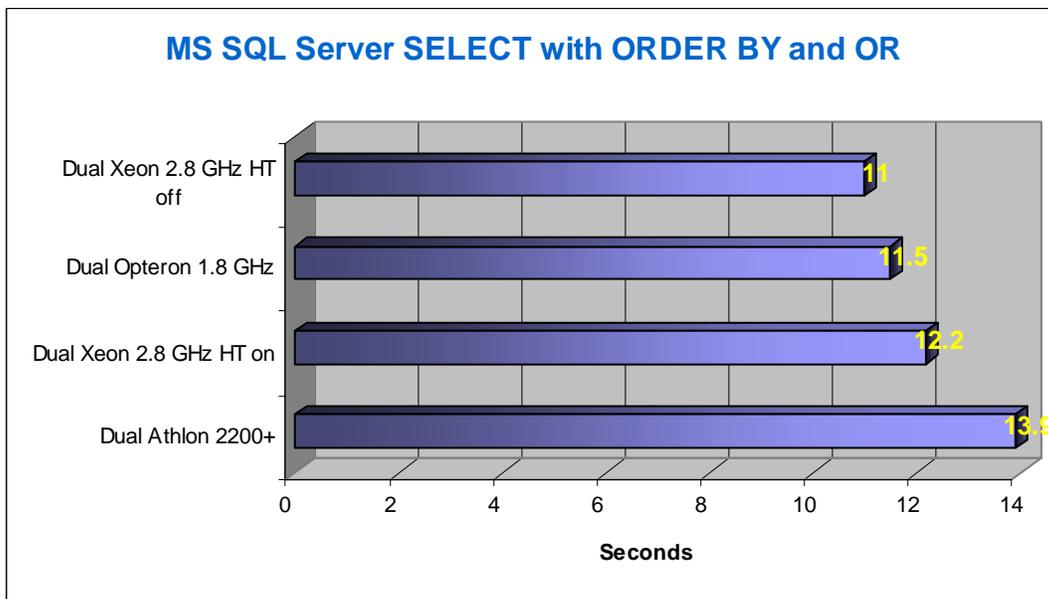


for the first time we see the dual Xeon outperform the Opteron by a relatively big margin. As this benchmark requires almost 1 GB of memory, it is possible that the Opteron suffers from the fact that Windows 2000 does not manage memory very efficiently. Still, it is a big win for the Hyperthreaded Xeon, which seems to fill in the IPC gaps that the numerous conditional branches create.

Our next benchmark consists of a similar query, but the result is sorted with an ORDER BY clause and there are fewer constraints in the WHERE clause.



The gap between the Xeon and Opteron has narrowed with this test, but the Xeon remains the indisputable winner here. Next we perform the query on only 4000 rows, but apply very complex conditions with a few ORs mixed in.



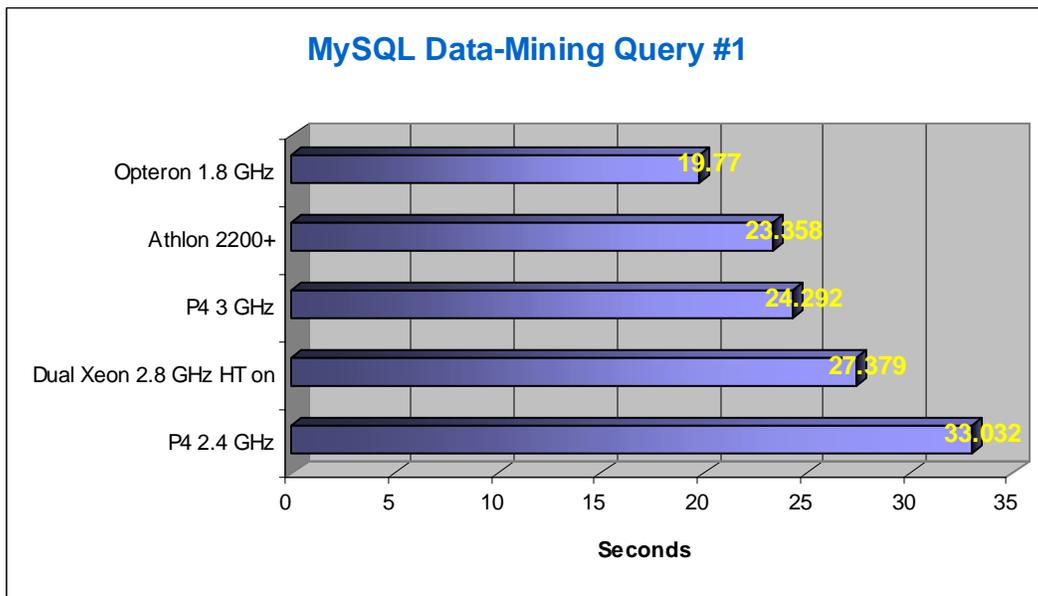
Adding OR logic really pleased the Opteron, which now performs better than the dual Xeon with Hyperthreading.

## MySQL 3.23.49 - Open Source Database Performance

Multi - threaded: No  
OS: Linux 2.4.20 / 2.4.19  
Memory Usage: 240 MB  
Typical Error Margin Between Different Runs: 3-5%  
Maximum Network Traffic: N/A, directly on server

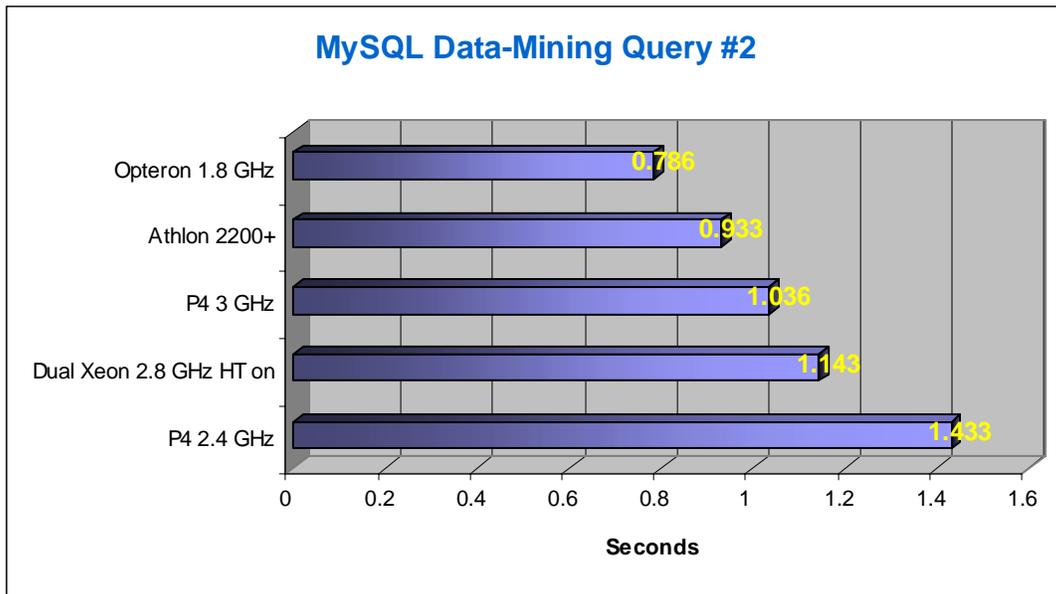
For this series of benchmarks, we imported a 400 MB HTTP log from webserver into a MySQL database. In this test we perform complex "datamining" queries and time them to determine how long they take to run. As this is our own database, we can show you the actual queries.

```
SELECT COUNT(*) AS hits,SUM(data_size),f.type FROM files_map f,log l WHERE f.id=file_id  
GROUP BY f.type ORDER BY hits DESC
```



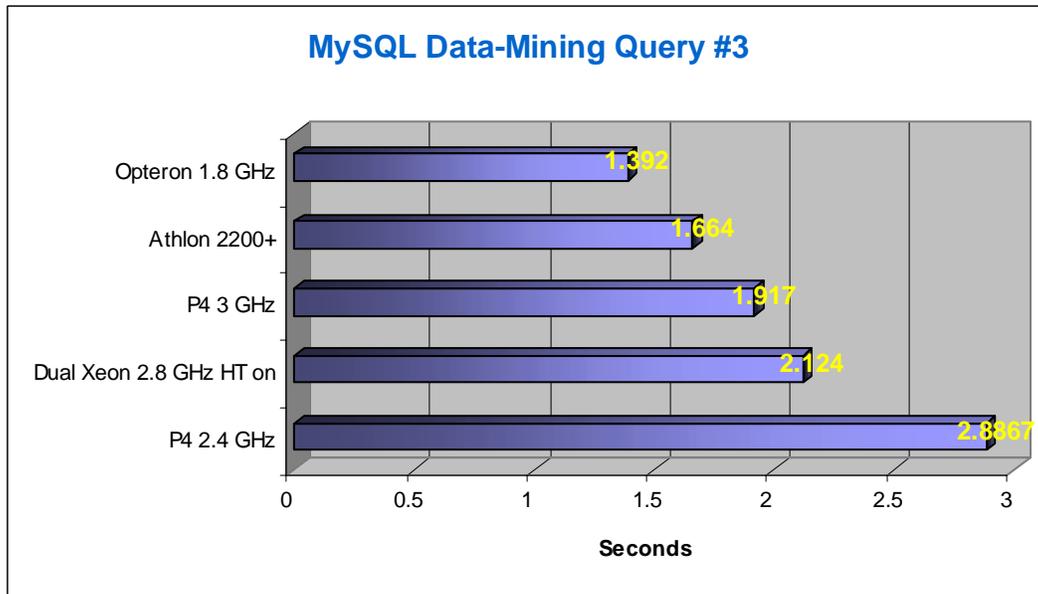
Mysql seems to like the Athlon architecture, the Opteron is up to 23% faster, but even the Athlon 2200+ leaves the P4s behind.

```
SELECT COUNT(*) AS hits,-data_size FROM log WHERE data_size < 0 GROUP BY data_size ORDER BY hits DESC
```



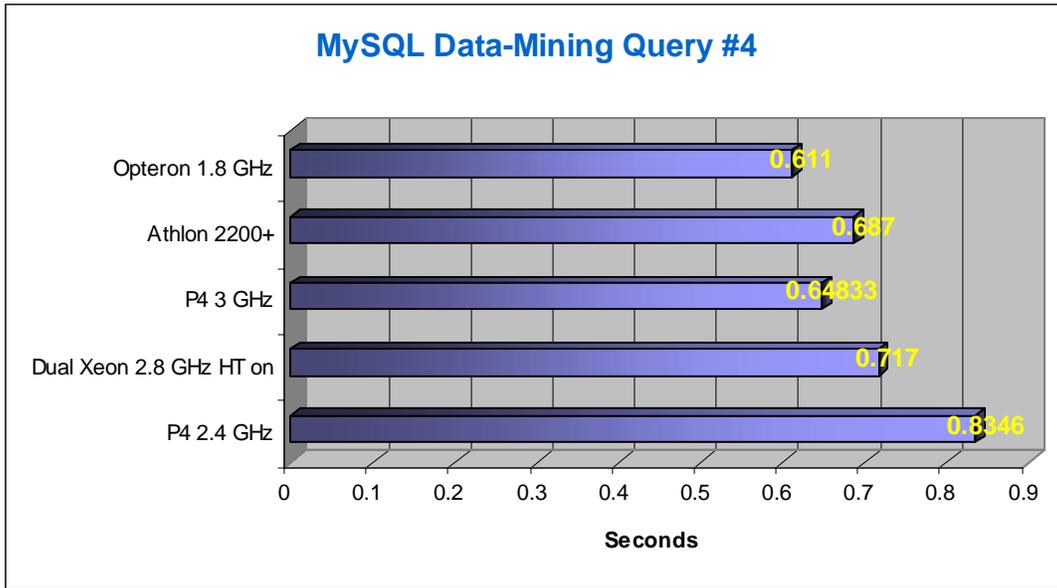
Again, a landslide victory for AMD.

```
SELECT COUNT(*) AS hits,FLOOR((data_size)/1024) AS kb FROM log WHERE data_size >= 0 GROUP BY kb ORDER BY kb ASC
```

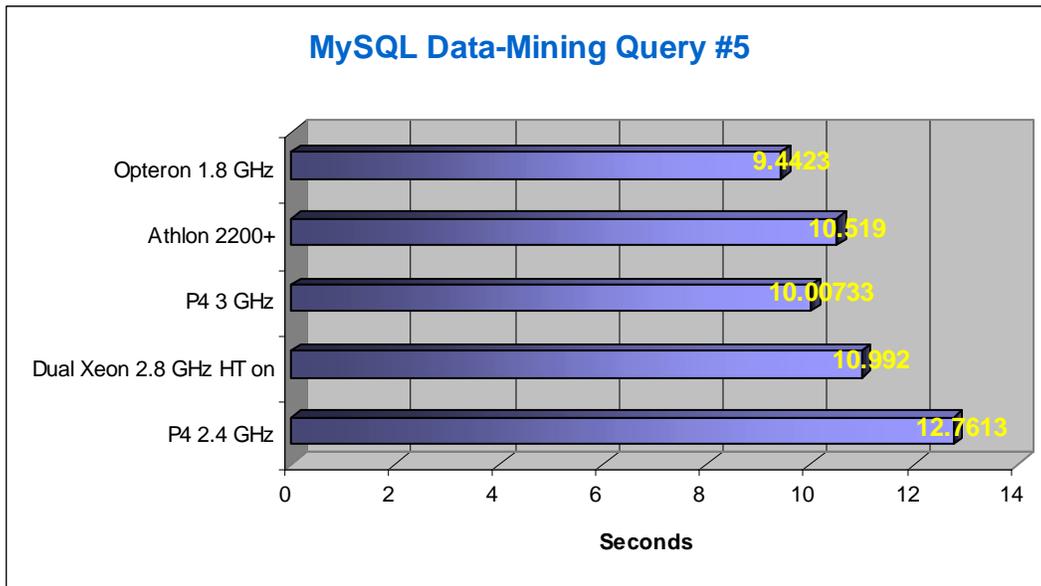


No less than 38% faster, the Opteron sweeps the floor with the competition.

```
SELECT COUNT(*) AS hits,f.file FROM files_map f,log l WHERE f.type='' AND f.file LIKE
'%/' AND f.id=l.file_id GROUP BY f.file ORDER BY hits DESC
```



```
SELECT COUNT(*) AS hits,f.file FROM files_map f,log l WHERE f.id=l.file_id GROUP BY
f.file ORDER BY hits DESC LIMIT 50
```

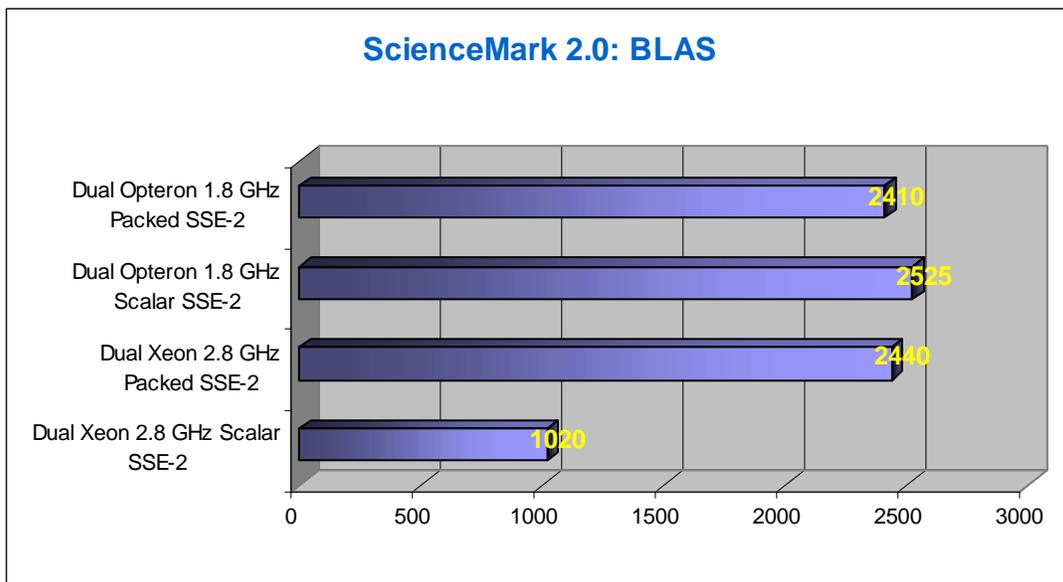


Two smaller victories, the Opteron proves to be 6% faster.

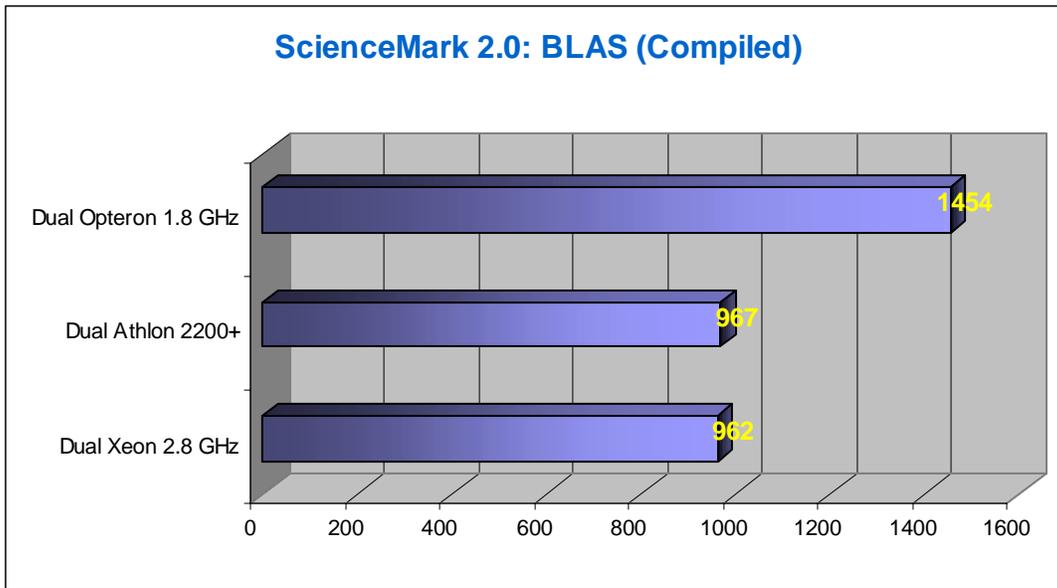
## HPC: ScienceMark 2.0

Aside from data-mining and transaction processing, the Opteron also aims at the scientific HPTC market. We ran the "BLAS" benchmark, a matrix multiplication floating point test similar to Linpack that is a component of ScienceMark 2.0. However, contrary to our C Linpack binary, the BLAS bench is extremely optimized to ensure it makes the most out of the CPU's caches. So the BLAS bench gives us a very realistic view of how fast large matrix multiplications will perform on a certain CPU.

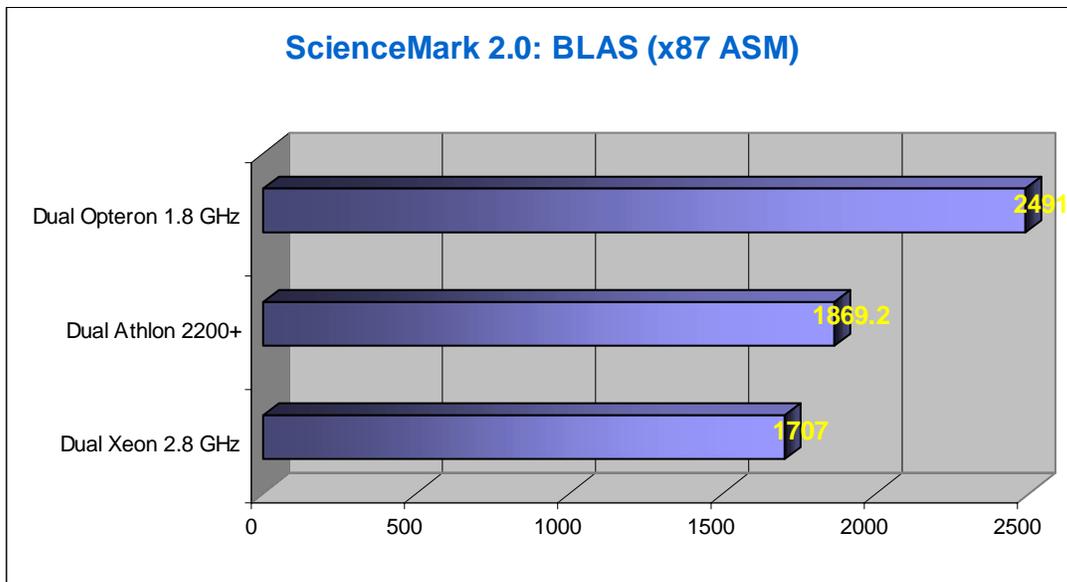
Even more interesting is the fact that it can measure SSE-2 and x87 performance, as well as compiled performance from a high-level language (no ASM). It is, however, a single-threaded benchmark, so even though it is run on dual-processor systems, only one processor is utilized.



The Opteron is the first AMD CPU to implement support for SSE-2, and here we see the performance of that implementation. When software is vectorized, a 1.8 GHz Opteron seems to reach the level of the P4 Xeon at 2.8 GHz. In scalar mode, it manages to surpass its competitor by a huge margin.

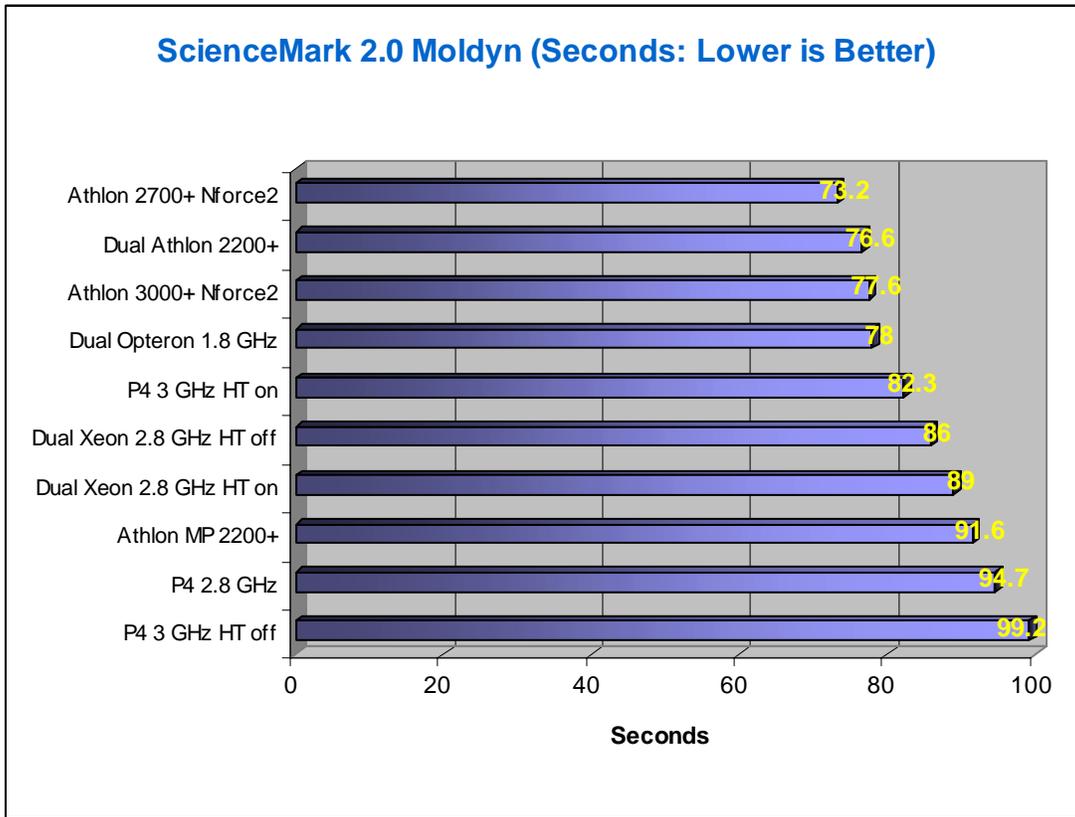


A Lower latency memory subsystem together with a better cache unleashes the power of the triple-pipeline FPU. This power, which was hidden in the Athlon, is now released by the Opteron.



Same conclusion as with the compiled benchmark, with the only difference that x87 assembly is probably dying out.

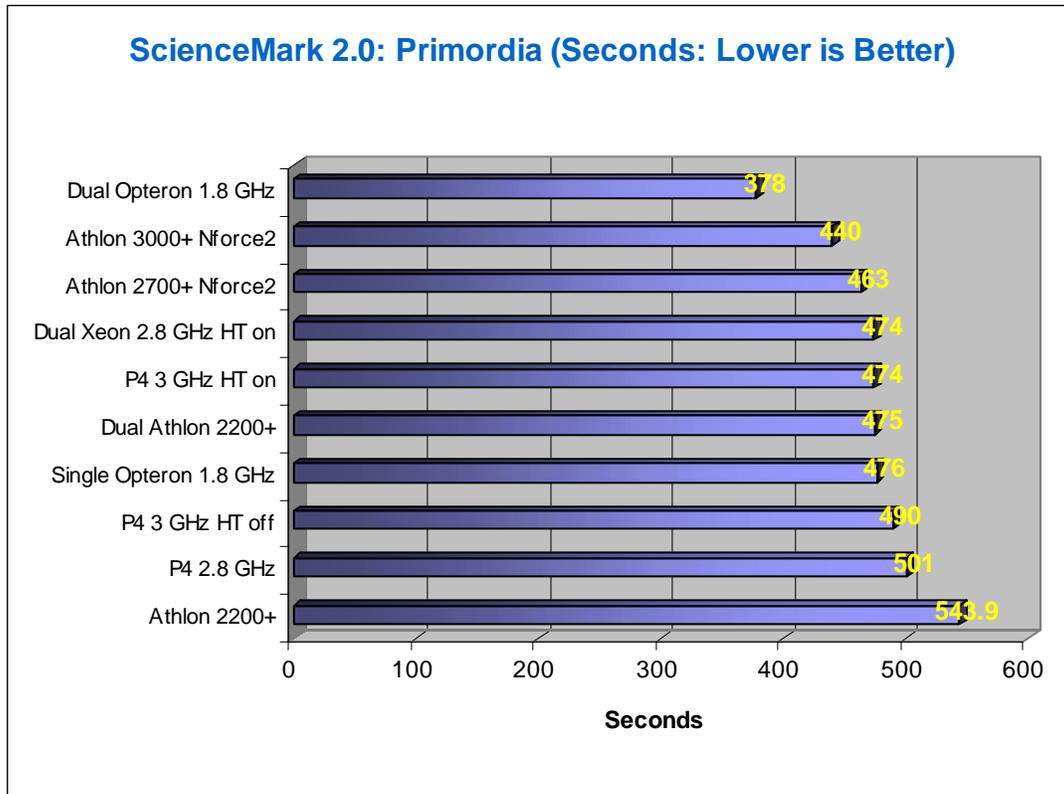
Moldyn



**Primordia**

From [Sciencemark.org](http://Sciencemark.org):

"This code calculates the Quantum Mechanical Hartree-Fock Orbitals for each electron in any element of the periodic table. The problem involved in solving for the orbitals is discussed in great detail here. A self-consistent loop is performed. At each step in the loop the hartree, exchange, and the correlation potentials for each orbital are evaluated. The user has a choice of a variety of algorithms with which to evaluate these potentials."



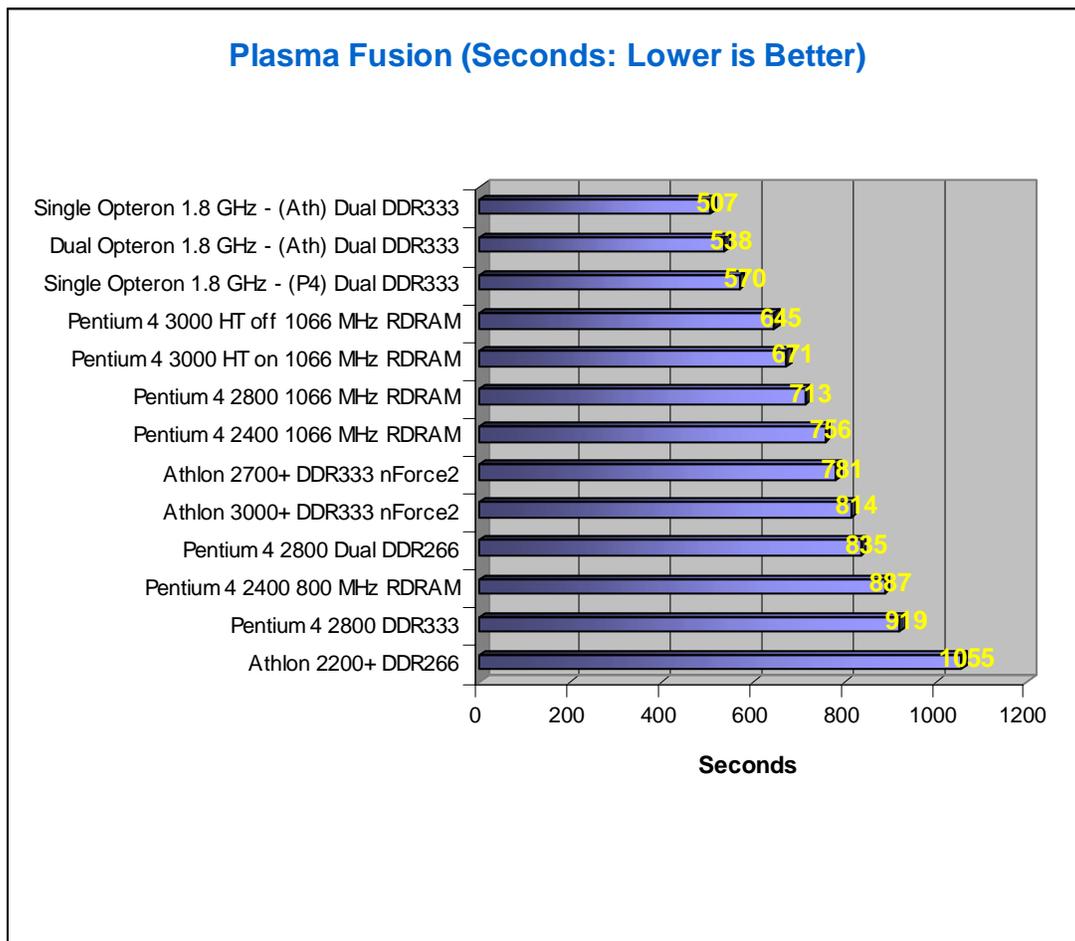
The most interesting thing about Primordia is that the Opteron seems to scale the best with more than one CPU. A second Opteron gives you 25% better performance, while a second Athlon gives you only 14%. In the case of the Xeon, it is even worse.

## Plasma Fusion

The Plasma benchmark is one of our more recent benchmarks, and you can [read an article about this benchmark here](#). Dr. Simon Bland gave us some new information:

"The MHD code is speed limited by the matrix inversion. The matrix consists of 2.1 million rows by 2.1 million columns, all values to double precision. It is, however, very sparsely populated... there are 29 non-zero diagonals. The current matrix solving method is an iterative solving method (bi-conjugate gradient solutions method). It uses 100 iterations to solve the matrix, each iteration consisting of ~5 matrix multipliers. As mentioned we are actively looking for better solving methods both for single and parallel. "

Note that this is a single-threaded benchmark.



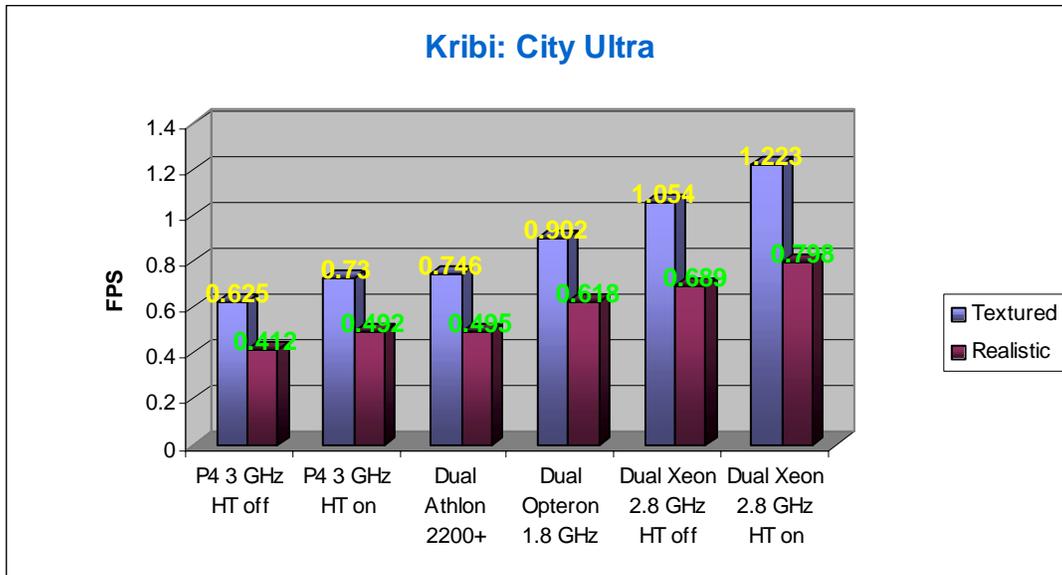
The Plasma calculations with the gigantic but sparsely populated matrixes couldn't care less about the 1 MB L2-cache of the Opteron. Nevertheless, the Opteron is almost twice as fast as its older brother, however. The Opteron delivers simply earth shattering performance thanks to the integrated memory controller, which awakens the FPU power hidden deep in the Athlon/Opteron backend.

## The Opteron as Part of the Rendering Farm

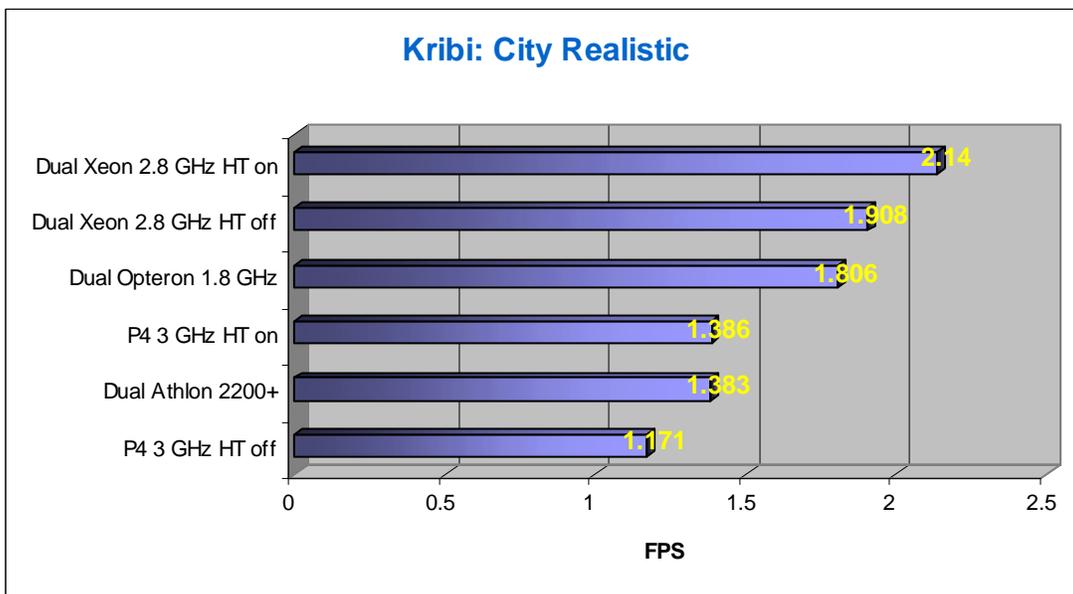
Few applications are so demanding as photo-realistic rendering. AMD believes that the Content Creation industry is one of the prime targets for the Opteron, so let's see how it fares with these kinds of applications.

## Kribi

Putting the OpenGL card manufacturers out of business is the vision behind the Kribi renderer. Perhaps we are exaggerating, but nevertheless, Kribi, a product of **Adept Development**, is an ultra powerful software rendering 3D engine. Originally developed by Eric Bron, a regular Ace's Hardware reader, it is designed to handle up to 10 billion polygons (!!) thanks to hidden surface removal and enables real-time photorealistic rendering at a few frames per second. The Kribi engine uses 100% software rendering (a pure CPU benchmark) and cannot work without SSE instructions. Thus it is a sort of SSE and FPU benchmark. This time we used Kribi version 1.1 which is much faster, and carefully optimized for Hyperthreading. We tested with several models to evaluate whether or not the used model has a significant influence on performance. The first scene - City Ultra - is the most spectacular: no less than 16.7 billion polygons in total. All results are expressed in frames per second (FPS).



The Opteron comes close to the Dual Xeon without HT, but the Hyperthreading boost thanks to extremely good optimization makes the Xeon with HT the best software renderer. The city scene contains about 107 million polygons.

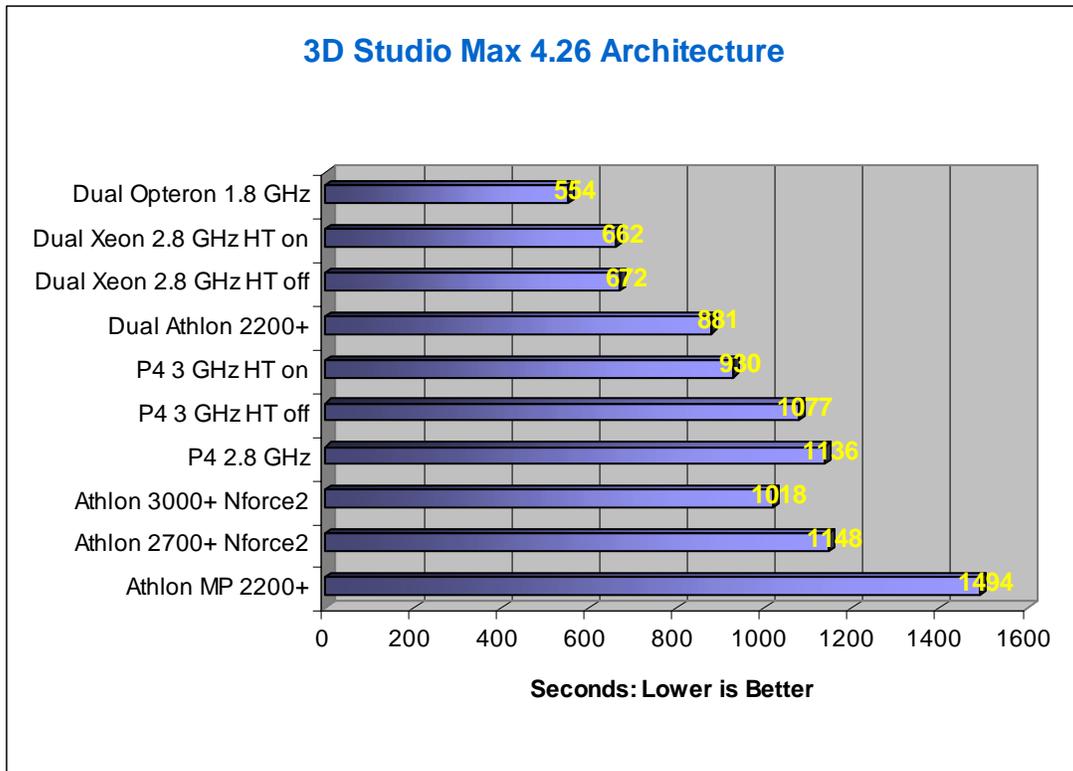


The next kribi benchmark confirms our previous assessment.

**3DS Max: 4.26**

Yes, we know, 3DS Max 5 is out and has been for awhile. But we like to continue one of our favorite benchmarks, as it is a pure (SMP) CPU, and to a lesser degree, memory subsystem benchmark. It is completely independent from the hard disk and graphics card, so you can compare these results with render benchmarks found in older articles like **this one**.

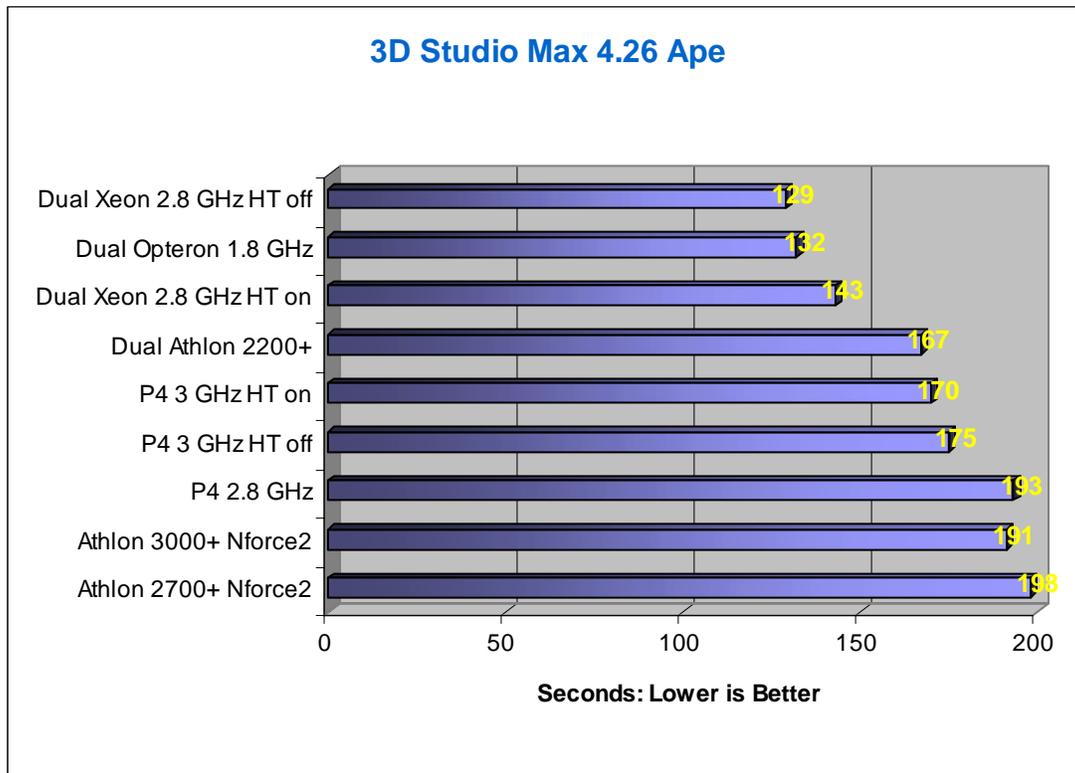
We tested the architecture scene from the SPECapc 3DS MAX R4.2 benchmark. This test has a moving camera that shows a complicated building, a virtual tour of a scale model. This complex scene has no less than 600,000 polygons and 7 lights. It runs with raytracing and fog enabled. Frames 20 to 22 were rendered at a resolution of 500x300 to the virtual frame buffer (memory).



Hollywood, game artists, here comes the Opteron: no less than 20% faster than a 1 GHz faster Hyperthreaded Pentium 4 Xeon. Impressive!



The ape animation, a typical game character rendering makes heavy use of lighting with no less than 44 different light sources. The scene also features complicated inverse kinematics: bone manipulation to control the facial animation and parameter wiring to move the fingers. Maxscript (macro language) is used to control certain movements. The polygon count is relatively low, only 26,000 polygons. Motion blur, which is one of the functions optimized for SSE-2, is achieved by rendering the scene in six passes. We rendered frames 20 to 25 at 320x240 to the virtual frame buffer.



The Dual Opteron equals the Dual Xeon 2.8 GHz here. The Ape benchmark is not very demanding though.

## POV-Ray 3.5

**POV-Ray** is a freely available raytracer for Windows, Linux, MacOS, and more. POV-Ray can be used to generate impressive **raytraced scenes**. From the **POV-Ray Documentation**:

Ray-tracing is a rendering technique that calculates an image of a scene by simulating the way rays of light travel in the real world. However it does its job backwards. In the real world, rays of light are emitted from a light source and illuminate objects. The light reflects off of the objects or passes through transparent objects. This reflected light hits our eyes or perhaps a camera lens. Because the vast majority of rays never hit an observer, it would take forever to trace a scene.

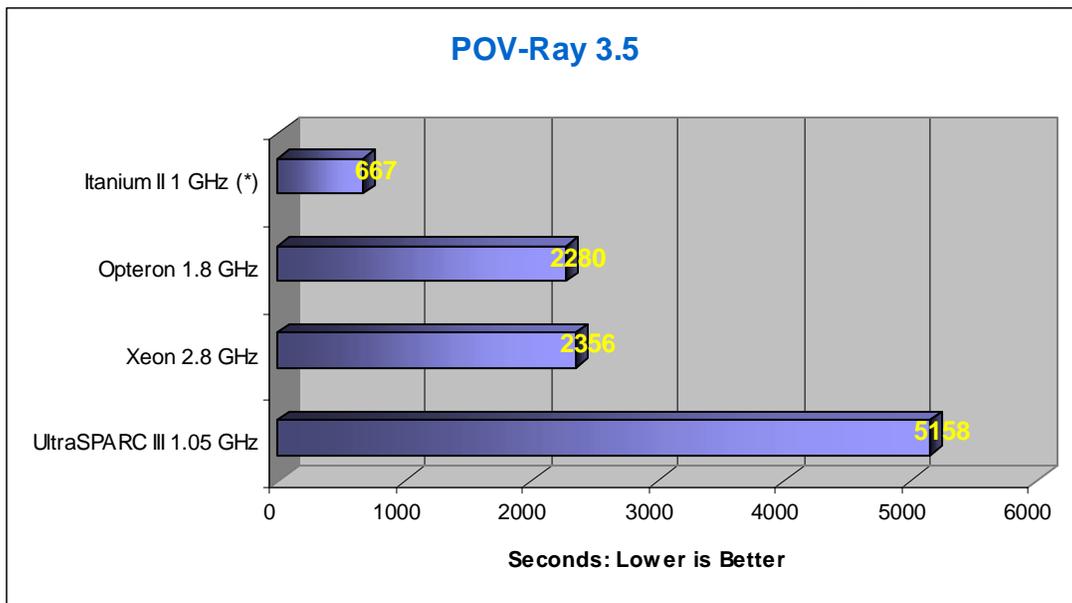
Ray-tracing programs like POV-Ray start with their simulated camera and trace rays backwards out into the scene. The user specifies the location of the camera, light sources, and objects as well as the surface texture properties of objects, their interiors (if transparent) and any atmospheric media such as fog, haze, or fire.

For every pixel in the final image one or more viewing rays are shot from the camera, into the scene to see if it intersects with any of the objects in the scene. These "viewing rays" originate from the viewer, represented by the camera, and pass through the viewing window (representing the final image).



*The Benchmark Scene*

To benchmark POV-Ray, we used the standard benchmark scene (**benchmark.pov**) pictured above with the standard benchmark settings (**benchmark.ini**).



Despite a 1 GHz clockrate deficit, the Opteron manages a slight lead over the 2.8 GHz P4 Xeon. This is a single-threaded benchmark, so only one CPU is being utilized. The standard binary distribution from the POV-Ray site was used (running Linux). The 1.05 GHz UltraSPARC III is the slowest, requiring more than twice the amount of time to render the same scene as the Opteron. The UltraSPARC III may be at something of a disadvantage given the benchmark is running on a single CPU in a 12 processor SunFire 6800 and depending upon how well-optimized its binary is relative to the others.

The fastest CPU, however, is in fact running 50 MHz slower. At 1 GHz, the Itanium II is outperforming the 1.8 GHz Opteron by more than three times. Please note that as this result was not generated by us, we cannot verify its accuracy. According to the [benchmark details](#), the Itanium II binary was compiled with Intel C++ 7.0.

## Core Improvements

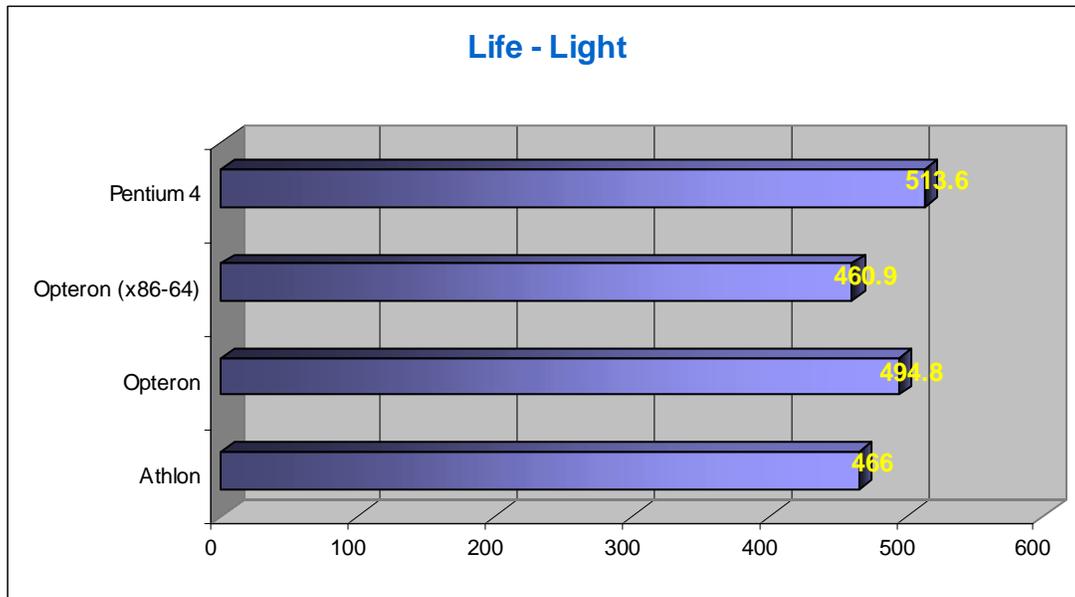
To analyse how the core of a CPU performs by itself requires benchmarks that are entirely or mostly dependant on the core - that is, performance scales linearly with clock rate. All this requires is a benchmark that fits perfectly into the on-die caches, since cache speed scales linearly with clock rate.

To make the comparison we have benchmarked some relatively simple C programs, compiled with GCC 3.2 under Linux, and most have perfect L2 cache hit rates. To consider the impact x86-64 code can make (with the extra registers as well as 64-bit registers), we also have x86-64 compiled code with GCC using the same optimisation flags as the 32-bit versions. For the x86-32 versions, the same binary was used, mostly for simplicity, though the GCC binaries were actually faster over all even compared to Intel's Linux C compiler with Pentium 4 optimisations.

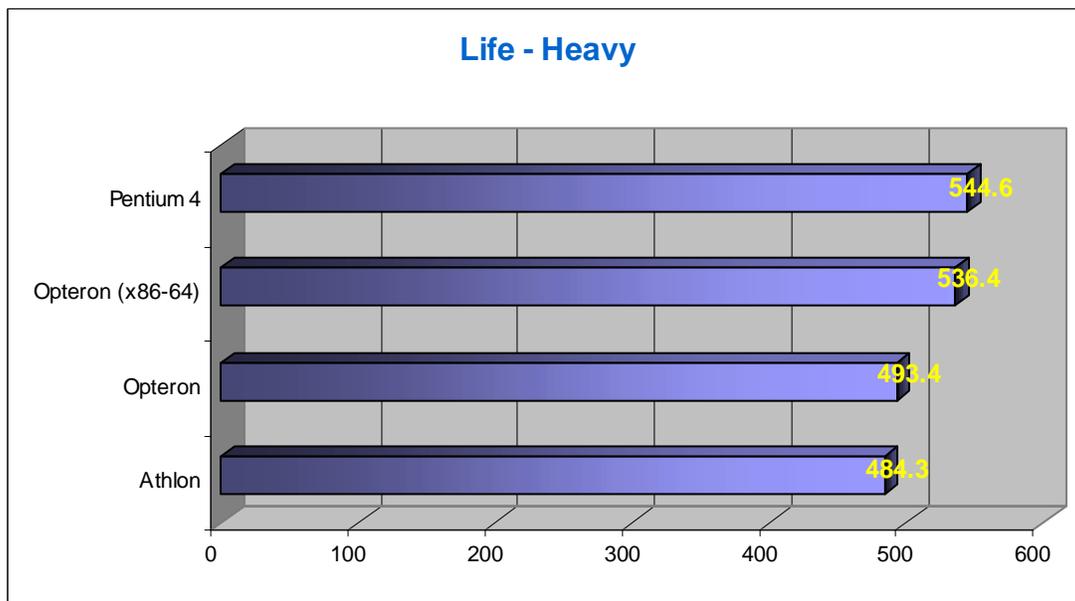
The "Life" benchmark is an implementation of [Conway's Game of Life](#), an example of "Cellular Automata". This is a nice pure-integer benchmark for teaching algorithm and code optimisation since it can be written in so many different ways. The main "Life" benchmark has a finite size board, with two variations, one with "light" population (sparsely populated) and the other with a "heavy" population (densely populated). A second algorithm uses a hash function to simulate an "infinite" sized board, which is generally slower overall though performance scales with the population fairly linearly.

The FFT benchmark is taken from the Java/C tests in the Java Grande [benchmark comparison suite](#) which "performs a one-dimensional forward transform of N complex numbers. This kernel exercises complex arithmetic, shuffling, non-constant memory references and trigonometric functions."

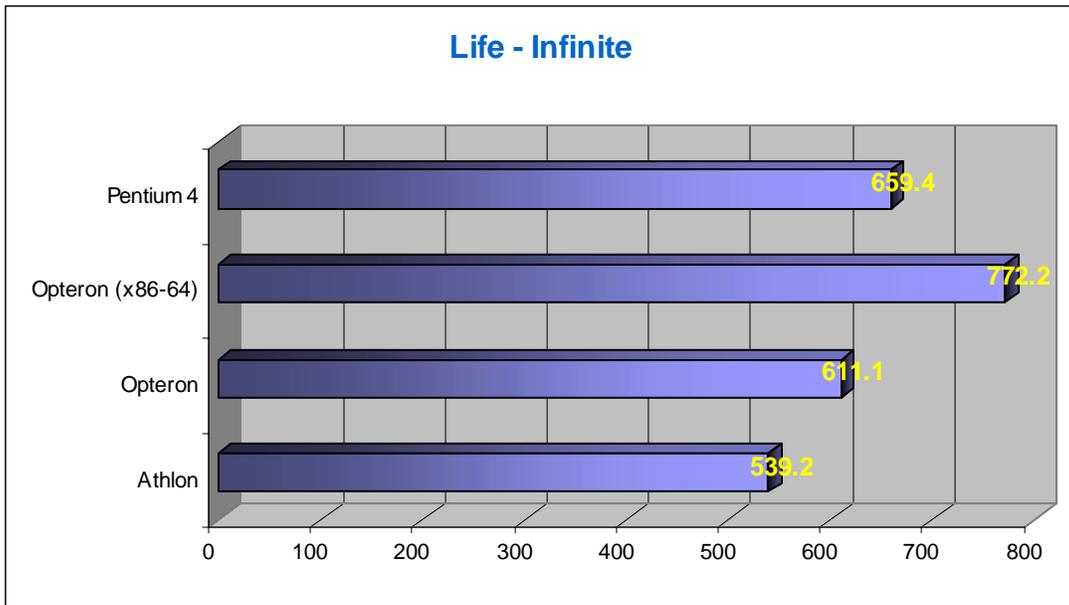
For both these benchmarks, a number of array/complexity sizes are compared, with the active data taking up to about 200KBytes, and often relative performance can flip between the "small" sizes and the "large" ones. The sizes were deliberately limited to have high cache hit rates so that small differences in core CPU design and software optimisation would show up better. These benchmarks are exactly the same as from our **binaries vs byte-codes** benchmarks from 3 years ago. They were all compiled to give maximum performance.



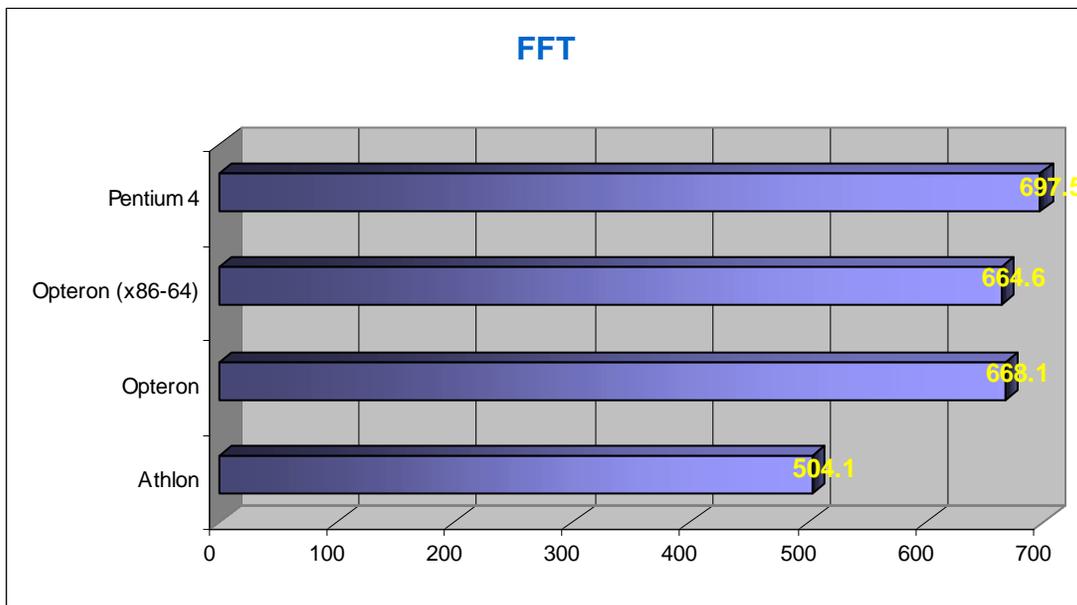
The Opteron score is quite close to the Pentium 4 score, despite the huge clock rate difference, though surprisingly, the x86-64 score is lower. Why this is the case is unclear, though perhaps the optimisations allowed by the extra registers turned out to be counter-productive - maybe the x86-64 version had greater loop-unrolling, which doesn't help for short-running loops.



This benchmark uses identical code to the "light" score, but different data. The differences between the Athlon, 32-bit Opteron and the Pentium 4 are about the same as before, except this time x86-64 optimizations improve the score, and by a useful amount as well. With more data, the loops would run for longer, suggesting that loop-unrolling is the main factor.



This time, there is quite a large difference between the Athlon and Opteron scores, and the benefit with x86-64 is 26%, putting it well above the Pentium 4.



On this benchmark, the Opteron is 33% faster than the Athlon, though can't quite match the Pentium 4. Clearly the x87 floating-point unit on the Opteron is a step above the Athlon's. Again, the x86-64 binary is actually slightly slower than the x86-32 one, but this time loop-unrolling seems an unlikely culprit because the average array size is much larger than in the "life" benchmarks.

## Benchmarks with 64-bit Data Processing

The Opteron has 64-bit general purpose registers, so any code that uses 64-bit integer operations should benefit a lot - using 2 32-bit registers, to do a 64-bit integer multiply takes 4 32-bit multiplies, and a 64-bit add requires 2 32-bit adds. To show this we have a custom micro-benchmark just on the Opteron which performs a "dot product" operation on two arrays of integers, which requires a lot of multiplies, with 32-bit and 64-bit integer versions and x86-32 and x86-64 code, using GCC.

Integer data size	x86-32 speed	x86-64 speed	Percentage increase
32-bits	437	581	33%
64-bits	96.2	437	354%

The x86-64 version is 33% faster for 32-bit integers, probably due to greater loop-unrolling. With 64-bit operations, the x86-64 version is a massive 354% (4.54x) faster, with a combination benefit of more registers and 64-bit processing. Notice also that the 64-bit x86-64 result is the same as the 32-bit x86-32 result - with double the complexity of the data, performance is the same.

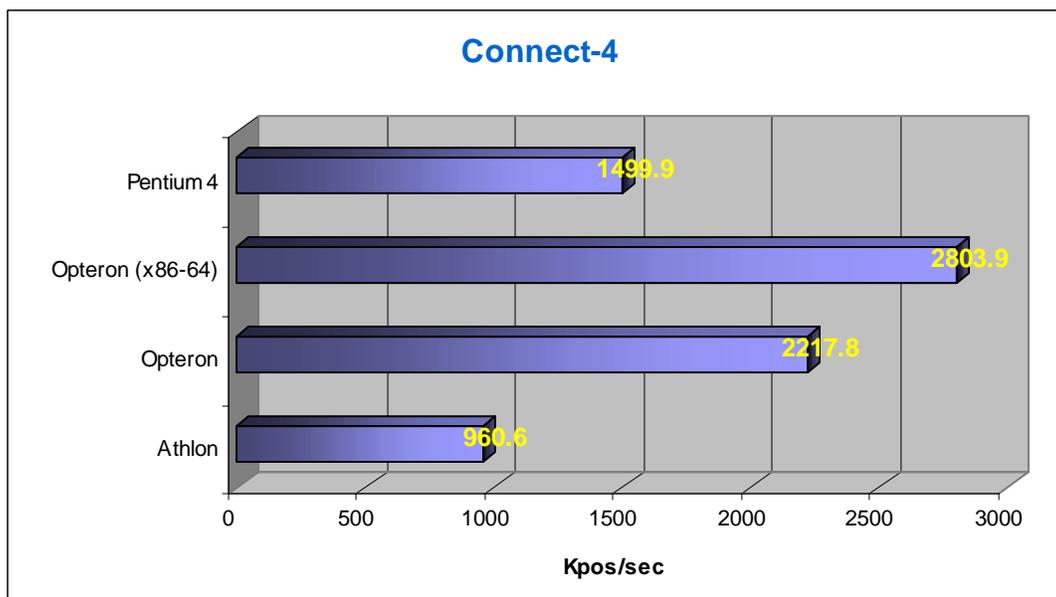
This micro-benchmark simply confirms the obvious - doing 64-bit operations with 64-bit code is much faster than with 32-bit code. Unfortunately, we do not have results with trying to do the same using SSE2, which has integer operations as well, and in theory could be used instead of 64-bit general-purpose registers.

In real life situations though, very few benchmarks have 100% L1 data cache hit rates, or only use 64-bit operations. To give a more realistic indication of the benefit of 64-bit integers, we looked for some benchmarks (with C source-code available) that made use of such operations. Financial applications (which need to use precise arithmetic instead of approximate floating-point) and encryption algorithms were too obvious candidates. Though we found several candidate benchmarks in general, only one proved compliable with x86-64 and that was only after tweaking the source code - not all C code out there is portable across multiple architectures and compilers.

This benchmark is the **Hourstones 2.0 connect-4** by John Tromp, which uses 64-bit integers for a hash data structure:

Implementation of the well-known game played on a vertical board of 7 columns by 6 rows, where 2 players take turns in dropping counters in a column. The first player to get four of his counters in a horizontal, vertical or diagonal row, wins the game. If neither player has won after 42 moves, then the game is drawn.

Two binaries were used, both created with GCC, both with the same optimizations except that one used x86-64 code.



Unfortunately, this benchmark seems to benefit quite a bit from larger caches, making the results difficult to compare from a CPU core perspective - the process used up 5.4MB of memory while running. Still, the x86-64 binary gets a 26% boost so it's clearly not main memory limited on the Opteron.

## Floating Point Performance with Flops

We have **used Flops before** to look at floating-point performance at the CPU core level. The benchmark has 9 modules, each with a different mix of floating-point operations, and we have a 6 different results: A 1.8GHz Athlon with GCC (Intel's CC gave about the same score), a 1.8GHz Opteron with a x86-32 GCC binary, a x86-64 GCC binary and a x86-32 binary using Intel's C compiler with SSE2 optimizations, a 2.8GHz Pentium 4 (no hyperthreading) with Intel's C compiler, and a 1.05GHz UltraSPARC III Cu using Sun's C compiler.

Module	1.8GHz Athlon (x86-32 GCC)	1.8GHz Opteron (x86-32 SSE2 GCC)	1.8GHz Opteron (x86-64 GCC)	1.8GHz Opteron (x86-32 SSE2 ICC)	2.8GHz Pentium 4 (x86-32 SSE2 ICC)	1.05GHz UltraSPARC III Cu (SPARC v9.2 SCC)
1	759	863	1079	1068	1138	865
2	492	907	716	701	461	525
3	1189	1278	1508	1696	2058	1956
4	908	1167	1815	1769	2299	1901
5	1001	1205	1311	1489	2202	1668
6	1216	1355	1385	1872	2330	1846
7	325	326	397	388	324	247
8	1145	1117	1360	1786	2265	1801

The Opteron easily beats the Athlon at the same clock rate, and by a significant margin with x86-64 code, but still can't match the higher clocked SSE2 optimized Pentium 4 result. It will be interesting to see what the fastest Athlon 64 can manage - a 2.4GHz one seems likely to beat a 3.2GHz Pentium 4. How much difference a heavily x86-64 optimized compiler can make remains to be seen. The Pentium 4's higher clock rate wins the day here, but you may be surprised to see the 0.18um 1.05GHz UltraSPARC III Cu be on par with the 0.13um 1.8GHz Opteron (wins 4, loses 4). At 0.13um the **UltraSPARC IIIi** is expected to reach up to about 1.6GHz and current systems are in a similar price range to the Opteron. The UltraSPARC IIIi has the same core as the UltraSPARC III Cu, so Flops performance would scale linearly with clock rate in comparison. How long prospective buyers will have to wait for these higher speed parts is unclear and Sun's compiler suite costs \$995, but since floating-point performance is 3-4x higher than GCC binaries, it is pretty much required.

## Opteron: Our Conclusions So Far Is the Opteron a Real Sledgehammer?

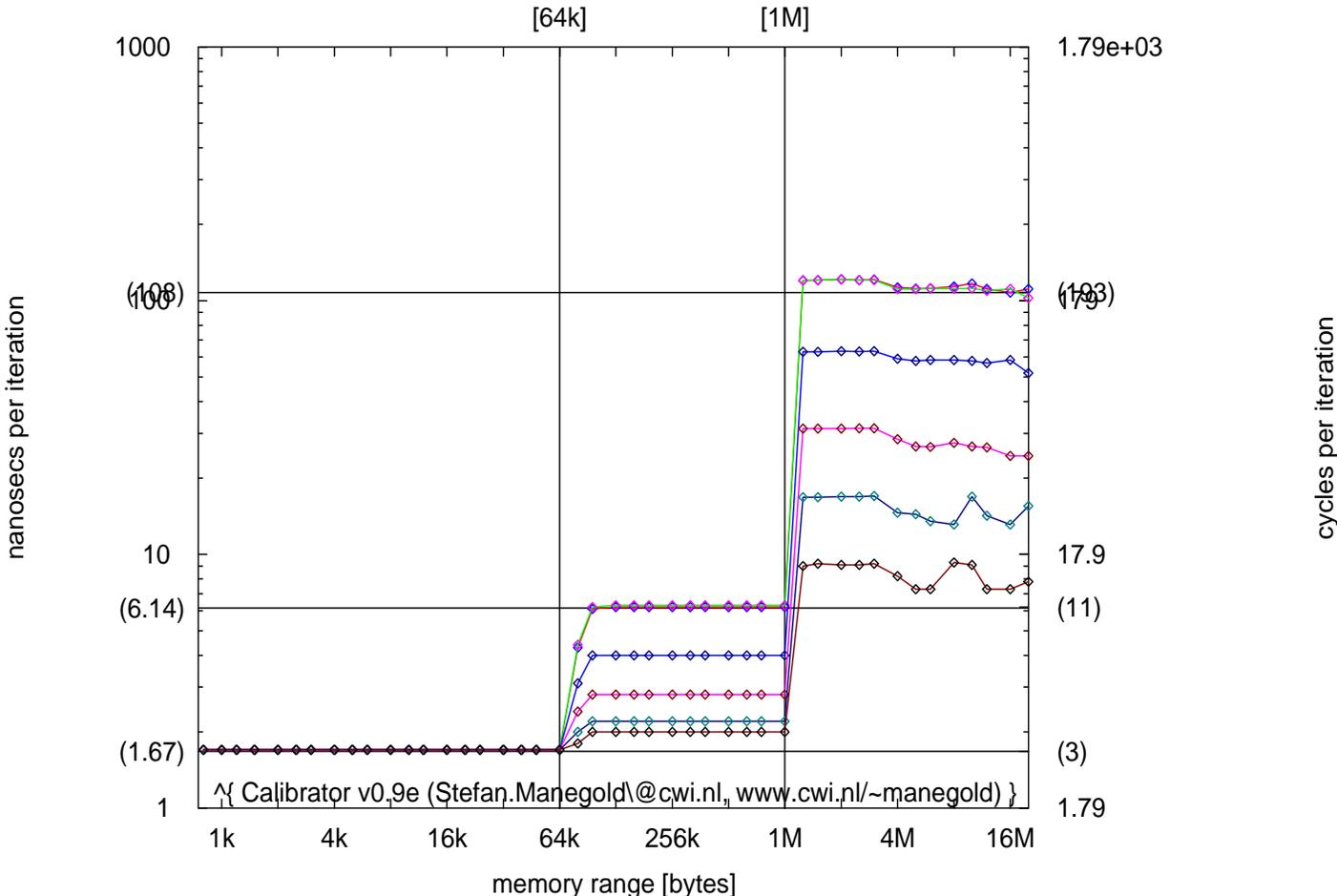
We think it is. Twenty percent better 3DSMax performance, by far the fastest in Matrix multiplication, up to 38% better MySQL performance, up to 40% better Java application performance (SPECjbb) the Opteron offers superior performance to a lot of people in the HPC, offline rendering, and server markets. If you're developing software for your own use, you can achieve a significant boost if you compile with an x86-64 compiler as we clearly illustrated. If you are using 64-bit integer data, the performance boost can range from significant to spectacular.

Yes, it is not a price performance winner as it costs as more than a Pentium 4 Xeon 3.06 GHz, and is slower than a 2.8 GHz Xeon in a few cases. At this point of time, as the Opteron is still limited to lower clockspeeds, you must first analyze whether or not your application is one of the "favorites" of the Opteron. But there are few applications that are not: the Opteron could not really convince in MS SQL Server 2000, for example, but that was about it.

But this is only the beginning. The introduction of a NUMA aware, 64 bit OS, better compilers, and 64 bit drivers should all boost performance. And there is no reason to believe that AMD will be limited to 1.8 GHz this year. Once the circuit engineers get the knack of SOI circuitry design, and there are reports showing that they already have, they should be able to get a lot more clockspeed out of the Opteron than they have out of the Athlon. The slightly longer but more balanced pipeline and SOI should take care of that and push the Opteron past 2.2 GHz and more.

Appendix A  
Calibrator Graphs

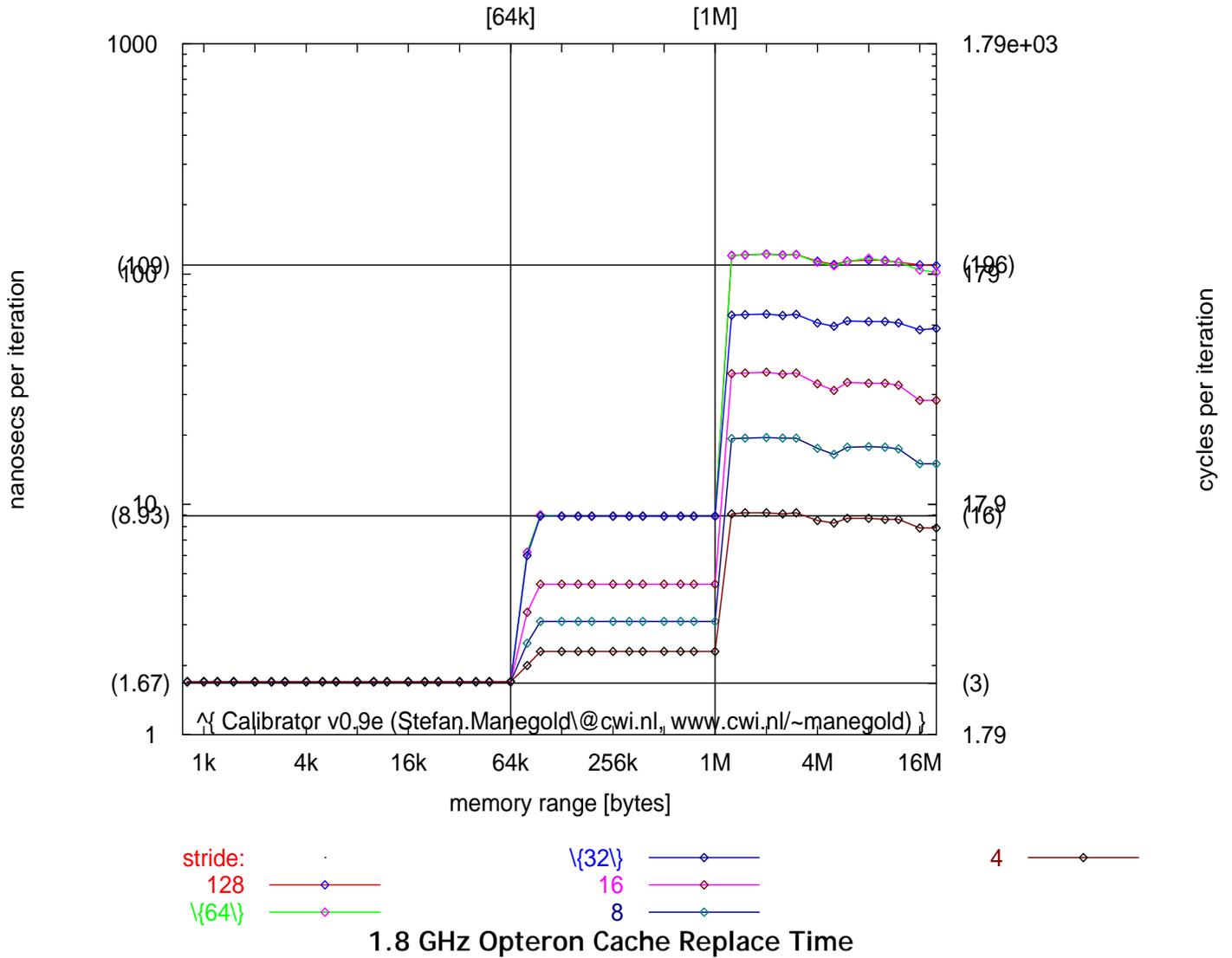
Opteron\_16M.cache-miss-latency



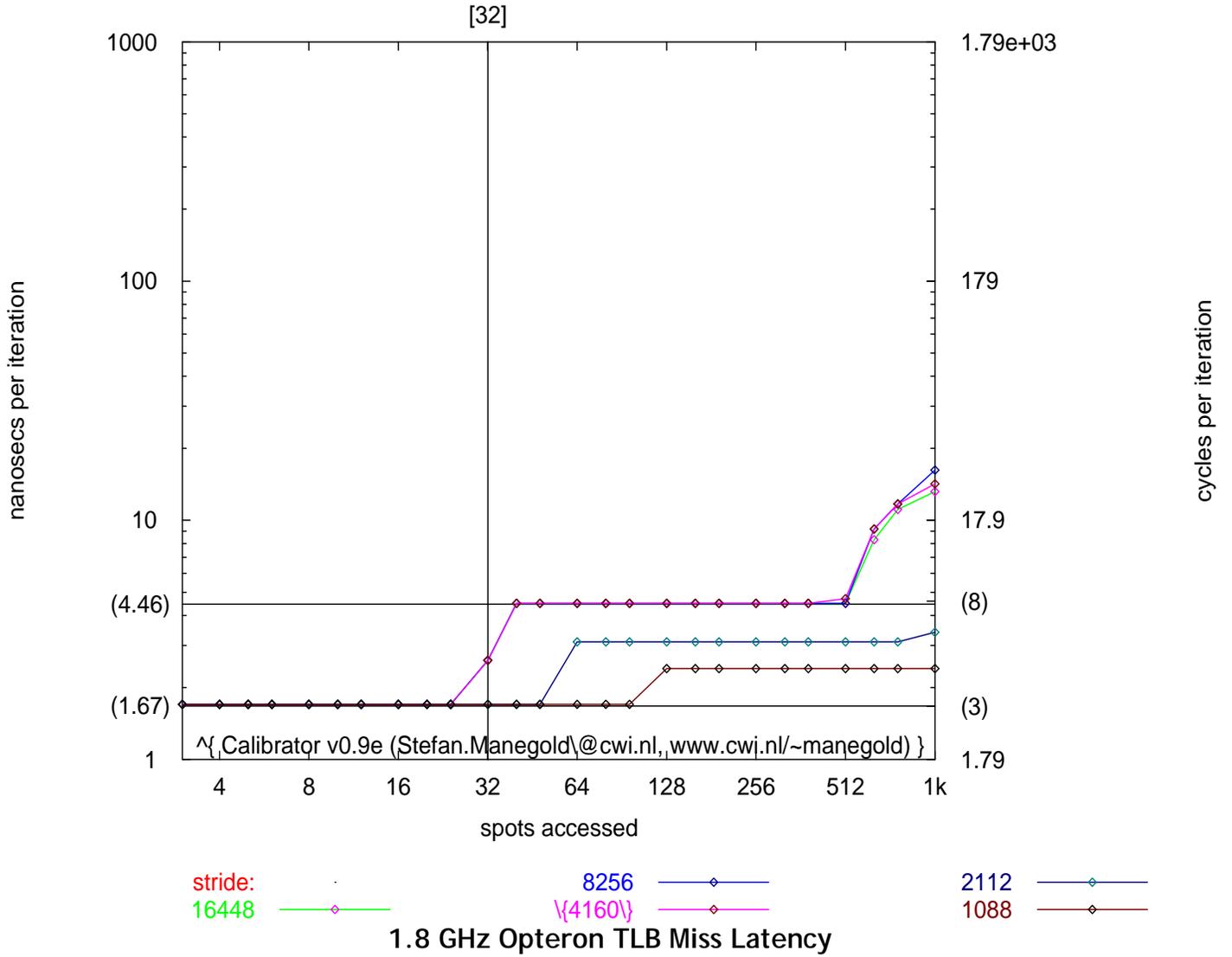
stride: 128 (red line with diamond), 64 (green line with diamond), 32 (blue line with diamond), 16 (magenta line with diamond), 8 (cyan line with diamond), 4 (black line with diamond)

1.8 GHz Opteron Cache Miss Latency

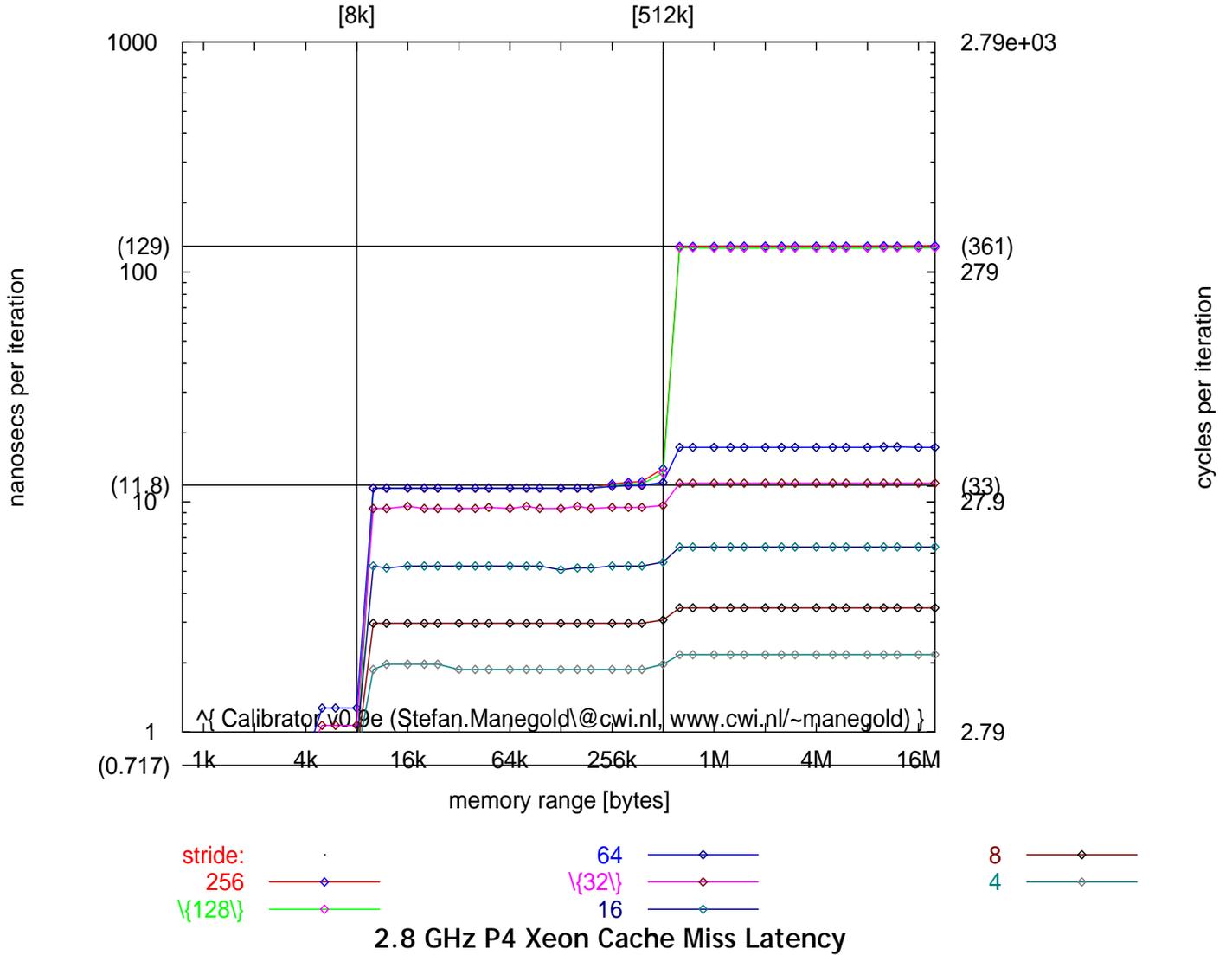
Opteron\_16M.cache-replace-time



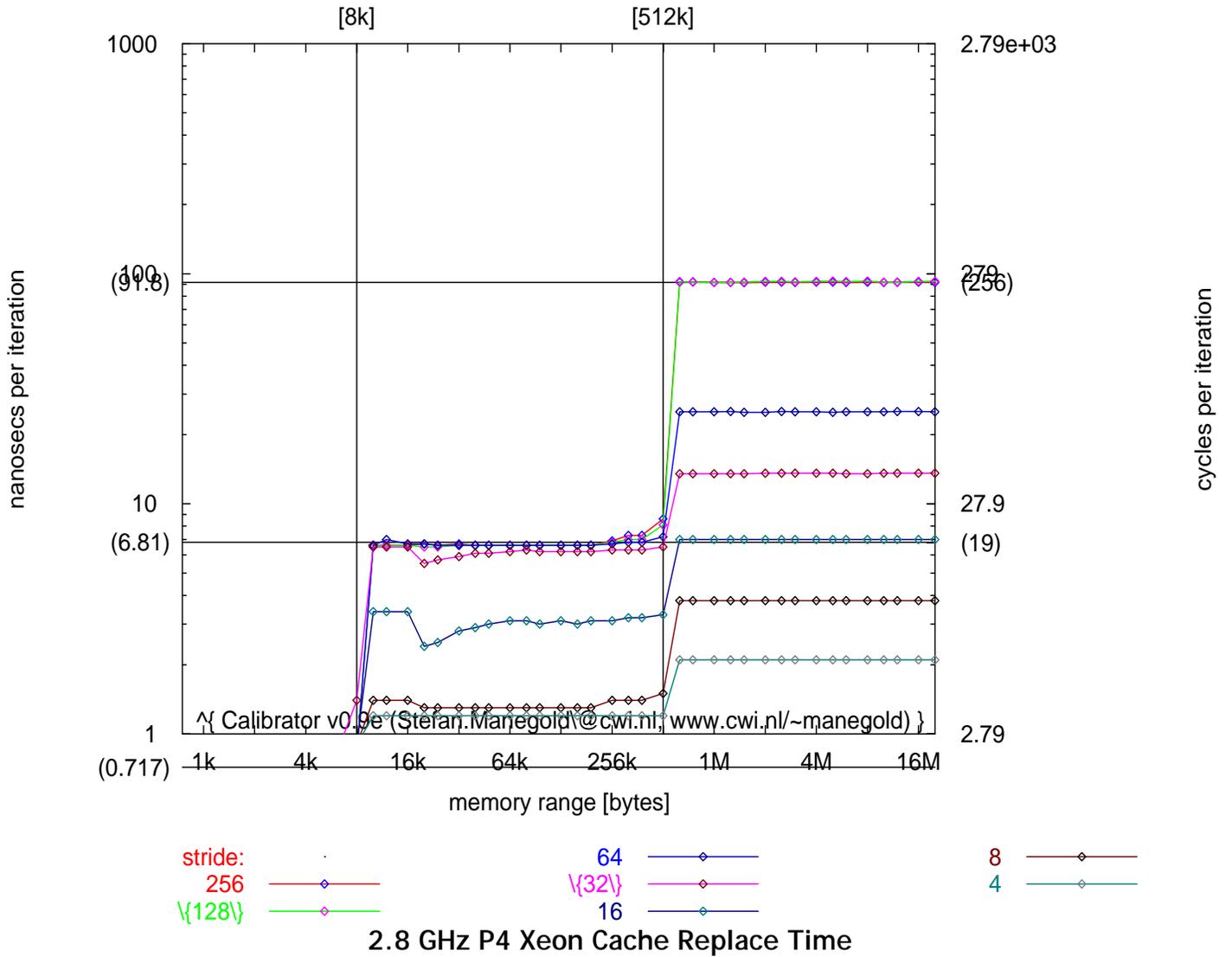
Opteron\_16M.TLB-miss-latency



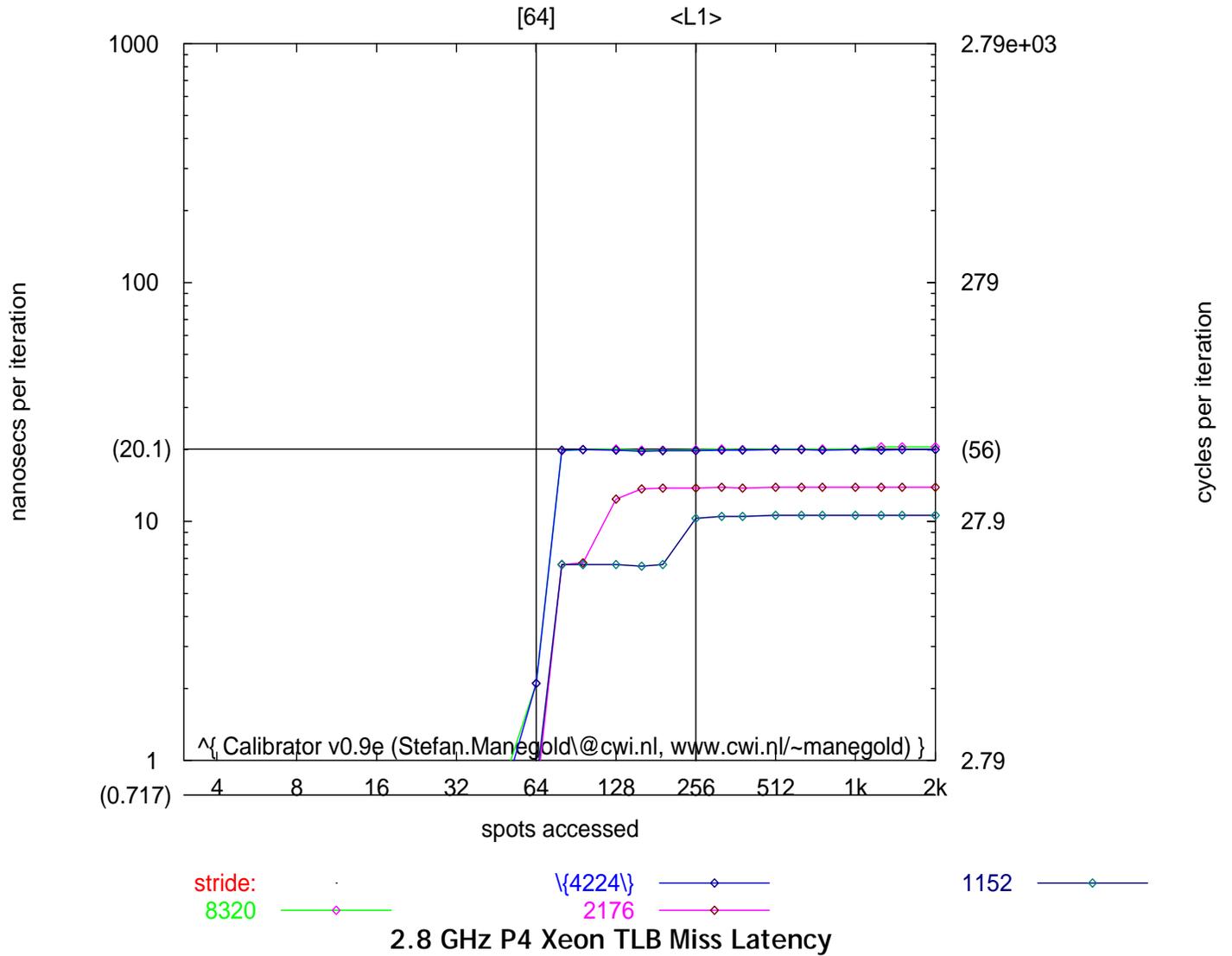
2\_8Xeon\_16M.cache-miss-latency



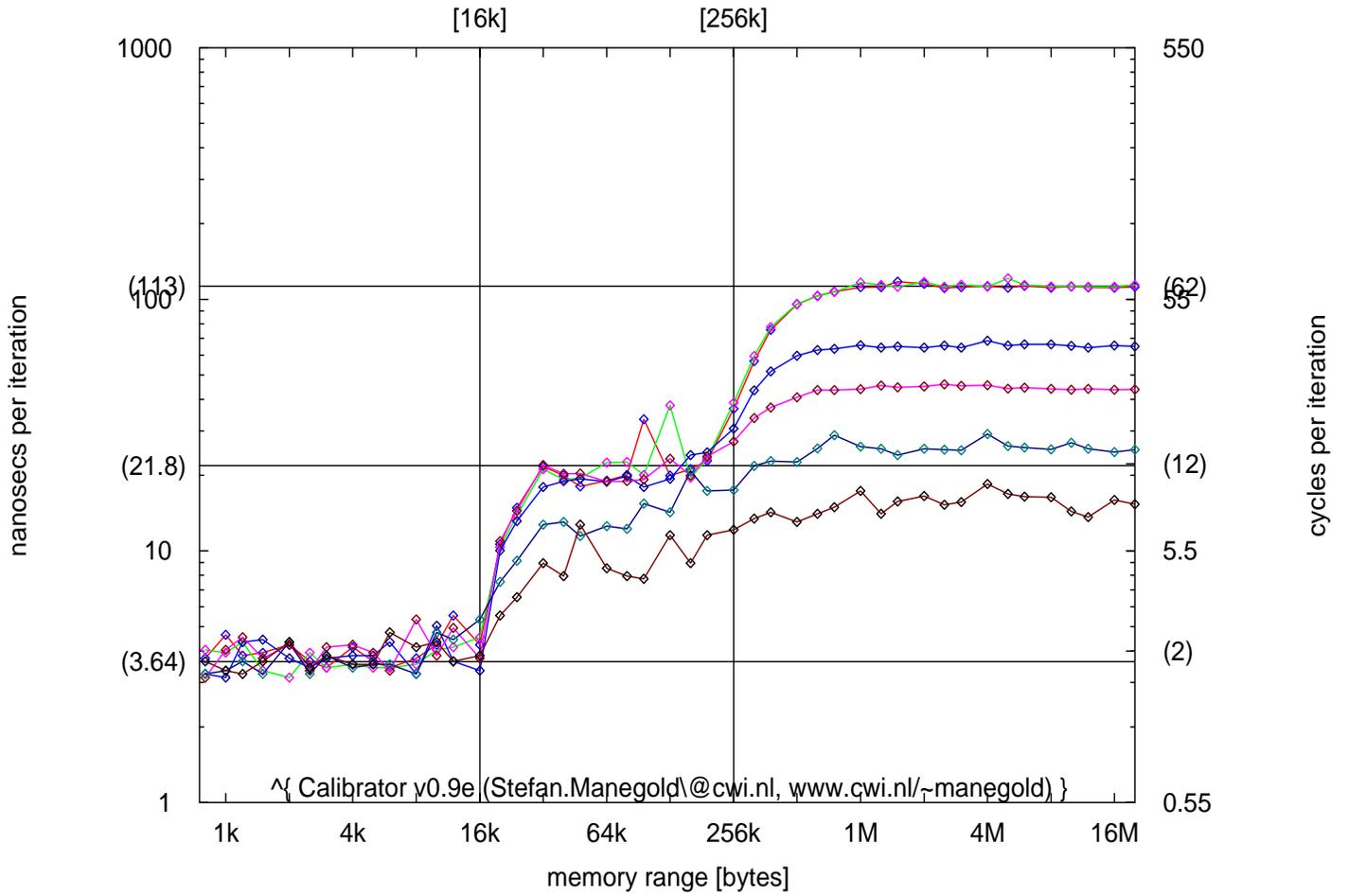
2\_8Xeon\_16M.cache-replace-time



2\_8Xeon\_16M.TLB-miss-latency

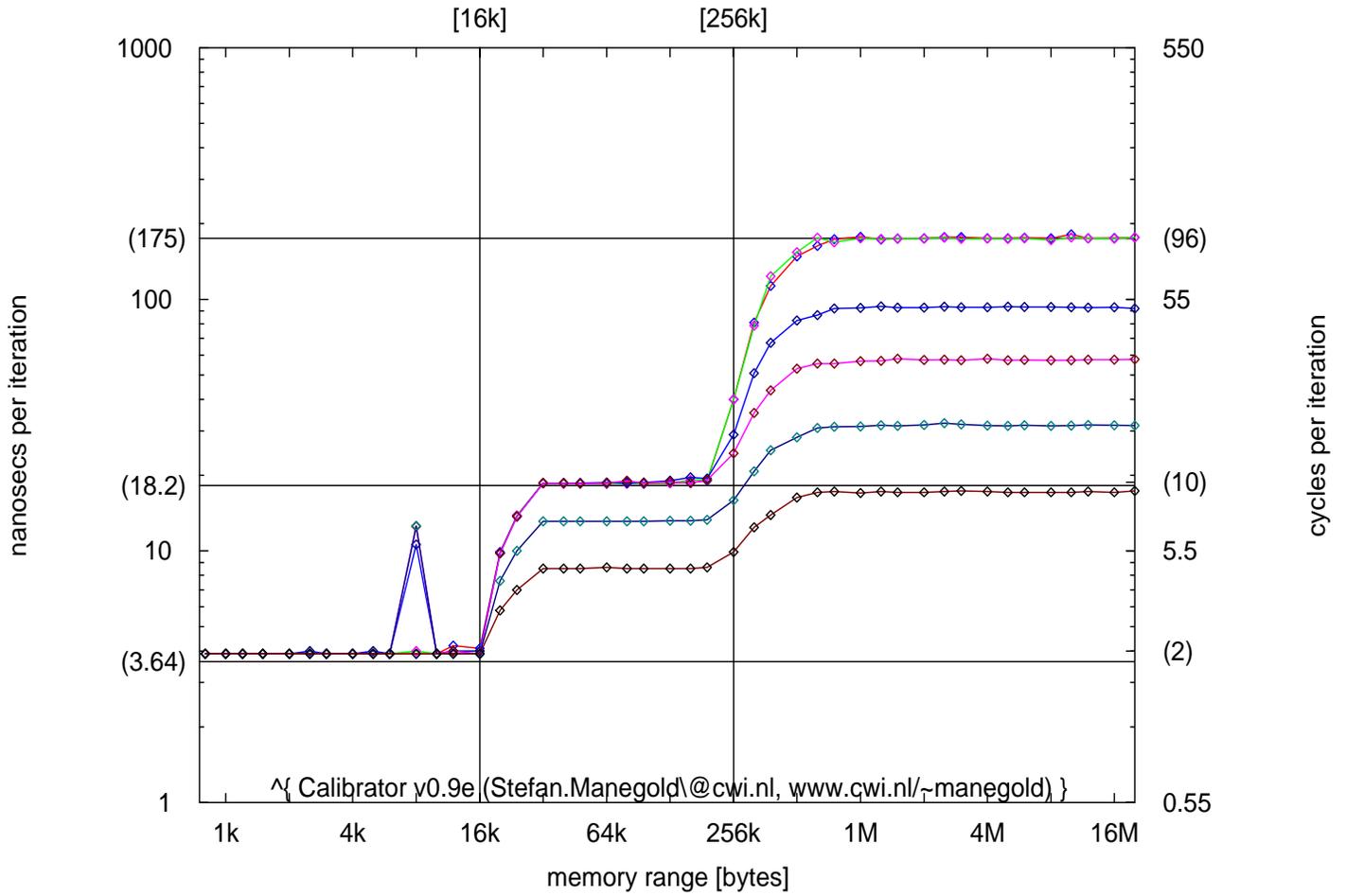


usiie550\_16M.cache-miss-latency



550 MHz UltraSPARC Iie Cache Miss Latency

usiie550\_16M.cache-replace-time



550 MHz UltraSPARC IIe Replace Time

usiie550\_16M.TLB-miss-latency

