A Preview of Intel’s Bensley Platform

Intel’s Goes Dual Core

Intel’s first foray into CMP server designs is the Paxville DP, which is currently available from the major system builders. Paxville is a shared package design that uses the Lindenhurst chipset, and to some degree it is backwards compatible with existing systems. Unfortunately, this calls attention to the main weakness of Paxville; it relies on an older system architecture that was not intended for dual core processors. While Paxville is certainly a viable upgrade path for older systems, it is quite clear that another platform will be needed to fully take advantage of dual core MPUs. That platform will be the Bensley platform.

The Bensley platform will launch in late Q1 as a combination of the Blackford chipset, the Dempsey MPU and some associated software and drivers. In early Q3, Intel will release Woodcrest, which is based on a new microarchitecture and will be a pin compatible with Dempsey. The key features which truly mark Bensley as a high performance platform are the independent front side buses, the snoop filter and the FB-DIMM memory subsystem. At the same time, Intel will also integrate many of the harder to quantify factors into the Bensley platform such as RAS and manageability features.

Dual Core Servers, Done Right

Intel’s shared bus architecture has several advantages; it is simple, easy to implement, cheap, and it is quite effective for two processor systems. However, when you have four CPUs on a single shared bus, there is not a lot of bandwidth to go around and there is an increase in cache coherency traffic to boot. All together, this makes Intel’s shared bus a suboptimal solution for four processors or more. Fortunately, Intel has gone to great lengths to address these issues in Bensley. Figure 1 below shows a comparison of the Lindenhurst platform to the Bensley platform. The initial focus will be on the buses; the memory subsystem will be discussed later.

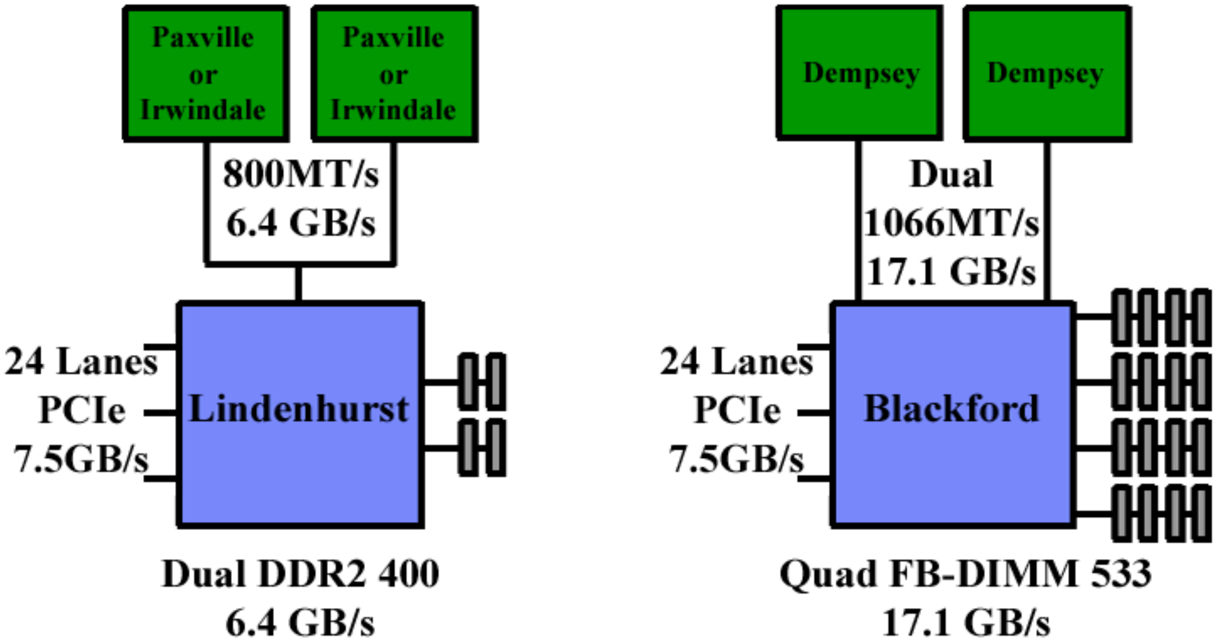


Figure 1 – Lindenhurst and Bensley platforms

Note that Lindenhurst has a 800MT/s bus, which provides 6.4GB/s of bandwidth. That's certainly not too bad for two processors, however, with Paxville DP, that's only 1.6GB/s of bandwidth for each CPU! On top of that, there is also the broadcast cache coherency traffic. In general, cache coherency traffic is proportional to the square of the number of processors in a system. All together, Lindenhurst is a fine chipset for single core MPUs, but definitely not great for dual core designs.

The Blackford chipset is far more reminiscent of IBM's X3 than Lindenhurst. Blackford uses two independent 1066MT/s buses (which will scale to 1333MT/s with the introduction of Woodcrest), and provides an aggregate of 17.1GB/s of bandwidth, 21.3GB/s with the upgraded bus. That is 4.3GB/s for each CPU and 5.3GB/s, when Woodcrest is introduced. In terms of processor bandwidth, Bensley is a huge step forward; but not only that, Bensley also packs an innovative way to deal with cache coherency, which is described in the next section.

Snoop Filter

One of the key issues in larger systems is effectively handling cache coherency traffic. Since coherency traffic is proportional to the square of the number of processors, this was not an issue for mainstream DP servers until the advent of dual core MPUs. However, now that CMP designs are the norm, and a 'mainstream DP server' may have 4 or more processors, it is necessary to address this problem. The Greencreek chipset (which is the workstation variant of Blackford) includes a snoop filter, which was designed to reduce cache coherency traffic.

The snoop filter separates each bus segment into a distinct cache coherency domain, with little traffic occurring between the two. While information on Greencreek's snoop filter is not publicly available, it is possible to infer how it works based on existing snoop filter implementations, principally the IBM X3 chipset. It is recommended that readers at least skim my discussion of IBM's snoop filter.

The easiest analogy is that the snoop filter essentially behaves like a switch between the two buses, rather than just a repeater. The snoop filter is most likely implemented as a large table that stores recent cache line requests, the state (MESI) of each cache line, and bits to indicate which segment the cache line is in (or both). When a cache miss occurs, the originating CPU broadcasts a snoop request on its bus. Both the snoop filter and the other CPU in the package will receive the request and take action appropriately. If the read request hits in the snoop filter, then it will check where the requested cache line is located. If the requested cache line is only available on the other bus segment, then the snoop request will be sent to the other segment. If the requested cache line is available on both buses or only on the originating CPU's bus or only in main memory, then the snoop filter does not pass along the request, thereby saving front side bus bandwidth. Read requests that miss in the snoop filter will probably go to main memory, but may also snoop the other bus segment for good measure. Since Intel's bus protocol is write-invalidate, write requests must always be propagated to any bus segment that has a copy of the cache line being written.

Without knowing the details of the snoop filter, it is difficult to assess how effective it will be. IBM's X3 chipset uses a 6MB eDRAM snoop filter that can cache the coherency information for 216MB of data. While eDRAM is very dense, it is rather difficult to manufacture in a logic process. Therefore, it is extremely unlikely that Intel is using eDRAM; given Intel's manufacturing and design strengths, the designers probably opted for a SRAM implementation. Since SRAM is less dense than eDRAM, it is tough to estimate the size of Greencreek's snoop filter. The X3's snoop filter is also used as a remote directory for larger systems, a design requirement not shared by Greencreek. Consequently, Greencreek does not need as large a cache as the X3 does, so 6MB is certainly the upper limit for the size of the snoop filter and in all likelihood the snoop filter is around 3MB. According to IBM estimates, their snoop filter provides a 10-15% performance boost for a 4P system, compared to using a simple repeater. Because there are so many unknowns surrounding Greencreek's snoop filter, it is hard to arrive at a precise estimate for the impact on performance, but it is reasonable to expect that the snoop filter will provide anywhere from a 5-12% performance boost. One known fact is that the snoop filter is ECC protected, like all good SRAM arrays.

*****NB: This page of the article has been updated.** I just received information from Intel indicating that Greencreek has a snoop filter, while Blackford does not. Blackford and Greencreek are actually separate ASICs, and the snoop filter is not included in the Blackford ASIC. According to Intel, at this point it is not certain whether Greencreek's snoop filter will be productized.***

Memory Subsystem and FB-DIMMs

Another significant improvement in the Bensley platform is the Fully Buffered-DIMM memory controller. FB-DIMMs are a new standard for memory, specifically targeted for servers and workstations. FB-DIMMs have been extensively described, and are fairly well known. To summarize, each FB-DIMM uses standard DRAMs, but with buffers on each DIMM. A 3.2GHz point-to-point uni-directional serial interconnect is routed between the memory controller and the buffers.

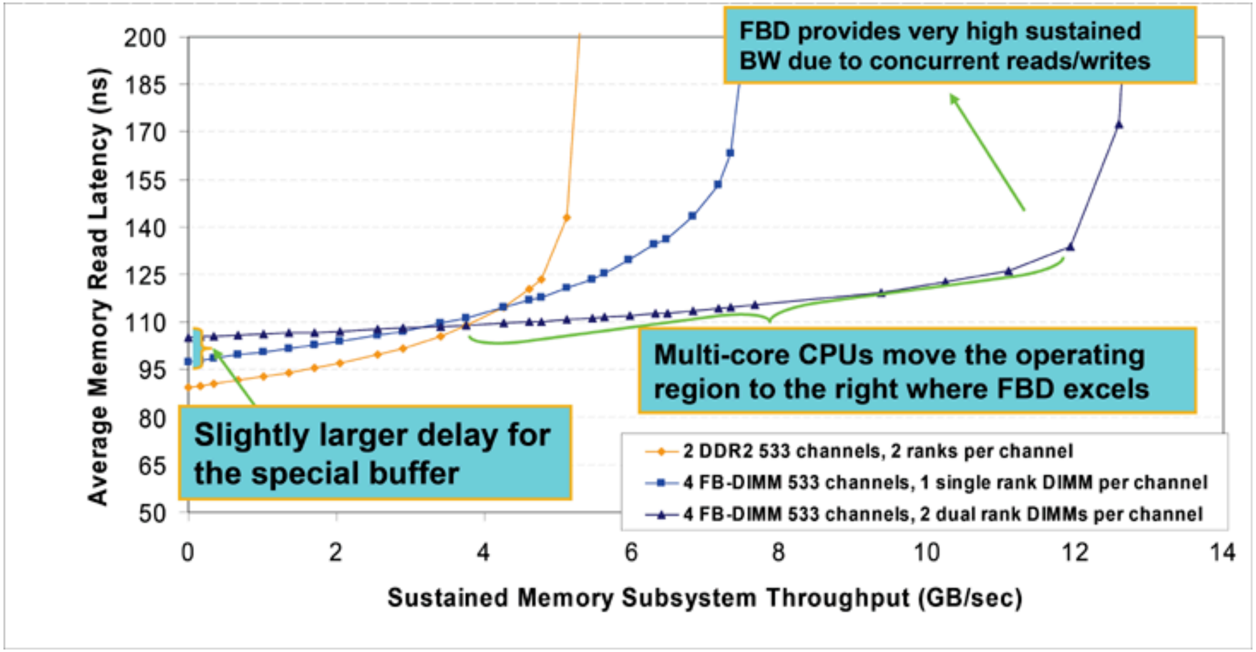


Figure 2 – Sustained bandwidth for FB-DIMMs (from an Intel presentation)

A single FB-DIMM channel requires 69 pins compared to 240 for DDR2; a little less than a third the pins, while providing the same bandwidth (4.264GB/s). Since FB-DIMMs are far higher bandwidth/pin than DDR2, and they support uneven trace lengths, the routing is much easier and more channels can be used in standard 4 or 6 layer motherboards. Blackford uses 4 channels, supplying 17.1GB/s of memory bandwidth, just enough to saturate the two front side buses. As Figure 2 indicates, FB-DIMMs can simultaneously write and read, but only from different DIMMs, so there is a very real bandwidth advantage to having multiple DIMMs on each channel. Each channel can work with up to 244 devices, four times as many as DDR2. Blackford supports DRAMs ranging in size from 256Mbits to 2Gbits, so using 2Gbit DRAMs, a system can have 64GB of memory. Blackford also improves upon the RAS capabilities of the prior generation chipset. CRC is used to detect errors in address or commands from the memory controller. The data is protected with SECDED ECC, and memory mirroring. Mirroring is a new feature to the DP market, and perhaps more importantly, with Blackford, there is only a small performance impact. Previous implementations of mirroring often made significant bandwidth and latency sacrifices for the increased availability. The memory controller can both detect and seamlessly correct failures of x4 and x8 DRAMs, the prior generation Lindenhurst chipset only supported x4 correction. Memory scrubbing and logging are also implemented, but that is hardly news; Lindenhurst also supported scrubbing to detect single bit errors before they can become uncorrectable double bit errors. The Blackford controller also allows a DIMM to be designated as a hot-spare that is used as a failover mechanism.

While FB-DIMMs are a vast improvement over DDR2, they also represent a trade-off. The unloaded latency for FB-DIMMs is worse than for DDR2, Figure 2 shows a disadvantage of around 10-20ns depending on the configuration. The latency to access any DIMM in a channel is determined by the latency of the last DIMM, so high capacity configurations will have slightly worse latency than those with only 1 or 2 DIMMs. However, this is nothing new. For all servers, more RAM means slower access; many current DP systems offer up to 16GB of memory, but only at the slowest speed grades. Moreover, unloaded latency is not that important; after all, people buy servers to run at 30-80% capacity, not unloaded. The real question is what does loaded latency look like? Now, the chart from Figure 2 is not quite gospel, but it does show that for applications which need more than 4GB/s of bandwidth, FB-DIMMs tends to have better latency.

The other drawback of FB-DIMMs is that they do add to the overall power consumption and thermal dissipation of the system. The buffer chips themselves are reported to dissipate around 3-7W, and with 4-16 DIMMs in a system and that could mean an extra 12-112W. However, that is for the first generation of buffers, which are mostly produced on older processes. As the buffers are moved to newer processes and as the manufacturers gain more experience, the power consumption and heat dissipation will decrease. This is a rather typical first generation teething problem and not a serious issue.

The Blackford chipset is the first to implement FB-DIMMs. The architects deliberately traded unloaded latency, power consumption and heat dissipation for better loaded latency, bandwidth and capacity. This is a very appealing trade-off for servers, where the main focus is high throughput at reasonable latencies. The additional bandwidth and capacity are essential for the next generation servers to scale up, since they are using multi-core MPUs. Ultimately, the choice of FB-DIMMs for the Blackford chipset will prove to be a wise one.

I/O and Platform Technology

While the CPU and the chipset are undoubtedly the heart of the platform, there is more to the Bensley platform than just Dempsey and Blackford. Bensley is also the first implementation of Intel’s I/O Acceleration Technology. Describing I/OAT is a bit beyond the scope of this article, but the general idea is to offload as much of the TCP/IP traffic from the CPU to the chipset and NIC as makes sense. According to Intel, an I/OAT Gigabit Ethernet controller delivers 30% more effective bandwidth than a regular controller. While this 30% improvement was probably measured on pathological code, it is quite clear that this will be an advantage for web servers, and other network intensive uses. Bensley optionally incorporates iAMT (Active Management Technology), which was described here.

Intel Performance Estimates

Bensley packs quite a few interesting features which should enhance performance: dual front side buses, a snoop filter and an enhanced FB-DIMM memory subsystem. But how does it perform? Intel was kind enough to provide the reviewers invited to their Portland workshop with quite a bit of performance estimates (in fact, my hand cramped from writing everything down). However, there are a couple of caveats. First of all, Intel's numbers for Paxville and Dempsey are estimates and therefore subject to change. Secondly, these are estimates for fully optimized and tuned systems, so the numbers may be hard to replicate. Lastly, the AMD Opteron performance numbers are for a 2.2GHz MPU; currently AMD is shipping 2.4GHz models and will be shipping 2.6GHz parts in the near future, while Bensley is 3-5 months away (and a lot can change in a few months). With that in mind, and a pinch of salt, it is time to look at Intel's projected numbers.

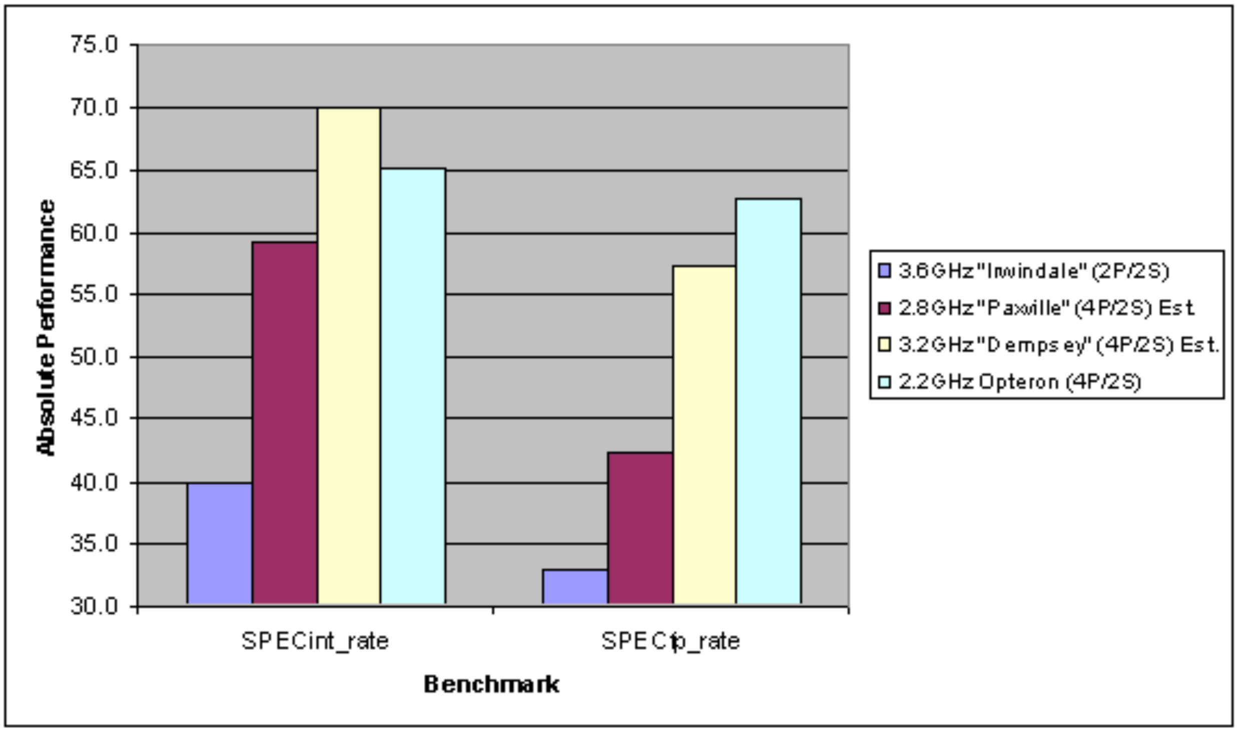


Figure 3 – SPEC_rate performance comparisons, from Intel

The SPEC_rate (base) benchmarks are CPU benchmarks that generally reflect workstation performance, rather than server performance. Of the two, SPECint_rate is more important, since integer workloads are far more common than their floating point counterparts. More info about SPEC CPU can be found at www.spec.org/cpu2000.

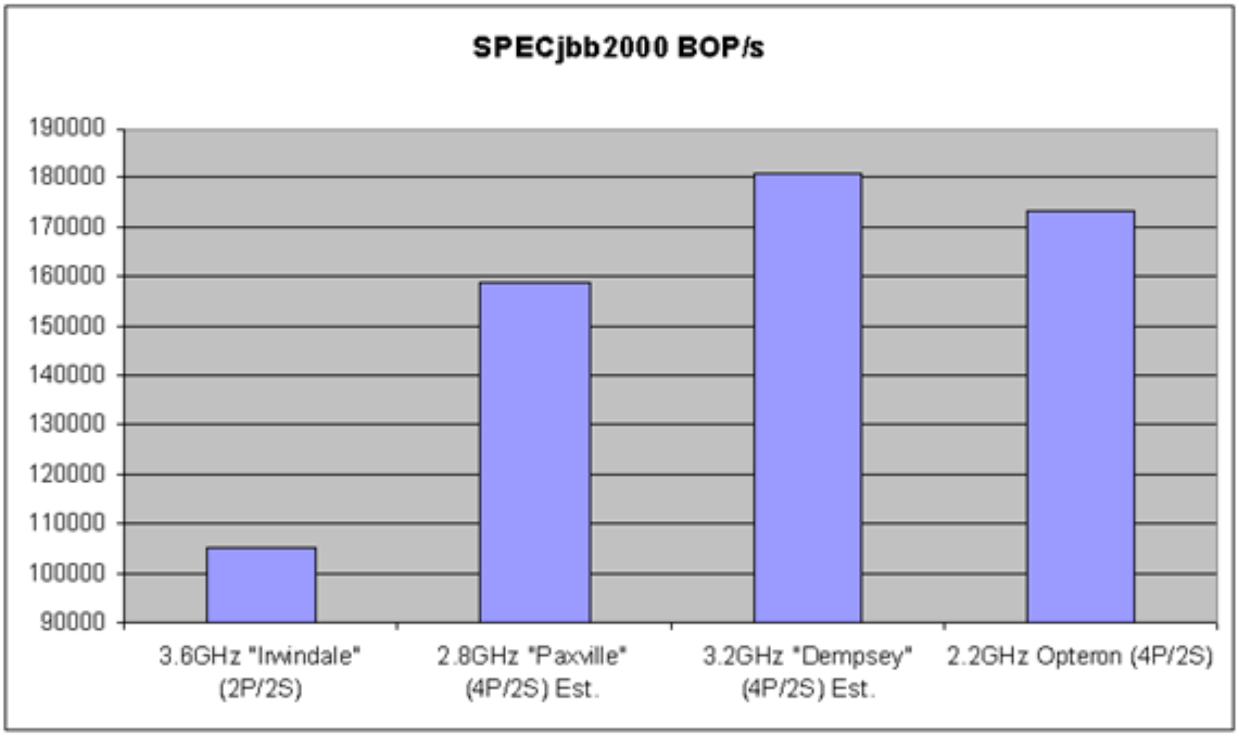


Figure 4 – SPECjbb2000 performance comparisons, from Intel

SPECjbb2000 is a Java server benchmark that is modeled loosely on TPC-C. It is a very good measure of server performance, but also heavily relies on the JVM used. In this case, the JVM was BEA System's JRockit 5.0 for the Intel systems and either JRockit 5.0 or Sun's JRE for AMD's system. SPECjbb2000 will be retired in early January of 2006, and in my performance measurements, I used SPECjbb2005, but it is still an excellent server benchmark. Information about SPECjbb2000 is at www.spec.org/jbb2000/.

Intel Performance Estimates Continued

Unfortunately, absolute performance numbers were not disclosed for these benchmarks. The OLTP benchmark is an open source workload based on MySQL that is nearly identical to TPC-C, the gold standard for server performance (along with SAP-2D). LS-Dyna is one of the most popular finite element analysis applications; the benchmark consists of simulating an automobile crash. LS-Dyna is available in both shared memory and distributed versions. Star CD is a computational fluid dynamics package, also available in both MPI and shared memory variants. Both LS-Dyna and Star CD are classic HPC and engineering workloads in commercial use throughout the world.

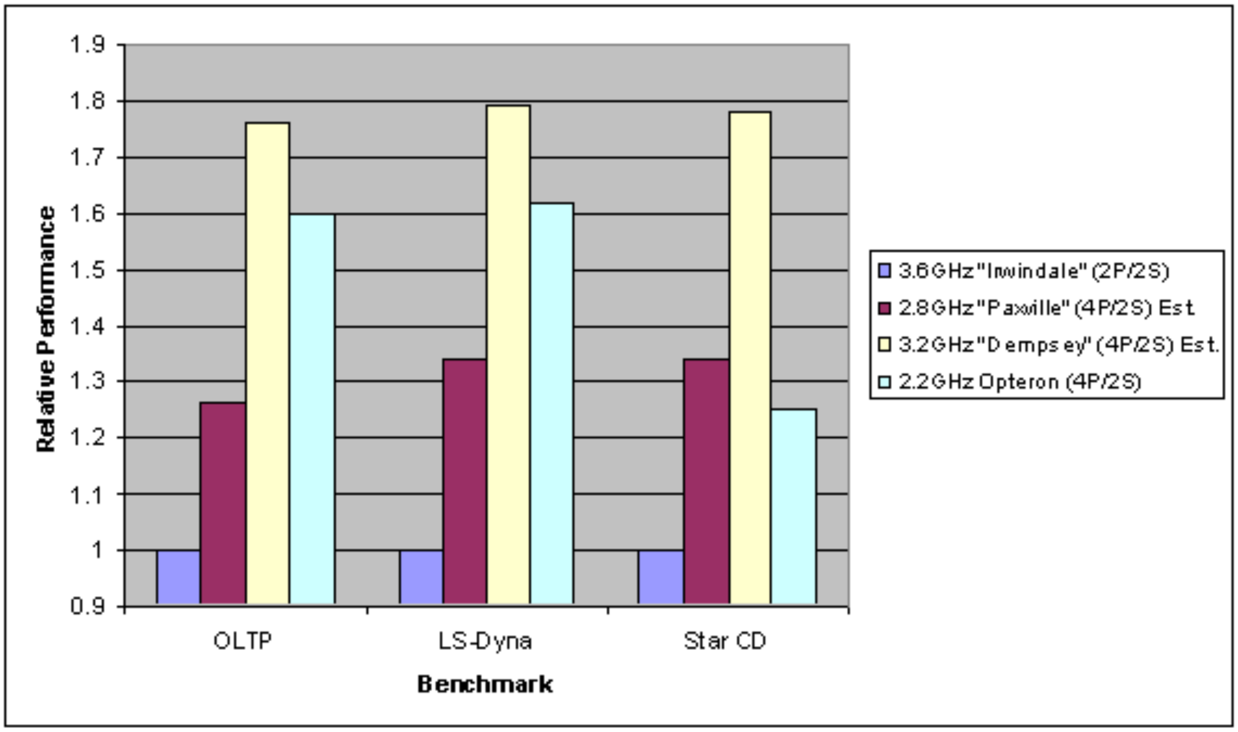


Figure 5 – Relative performance for OLTP, LS-Dyna and StarCD, from Intel

LINPACK is the ever popular dense linear equation solver used in the TOP500 list. It is also available for shared memory systems and for clusters. While all three Intel systems were simulated at problem size N = 10,000, the AMD score was simply taken from their website.

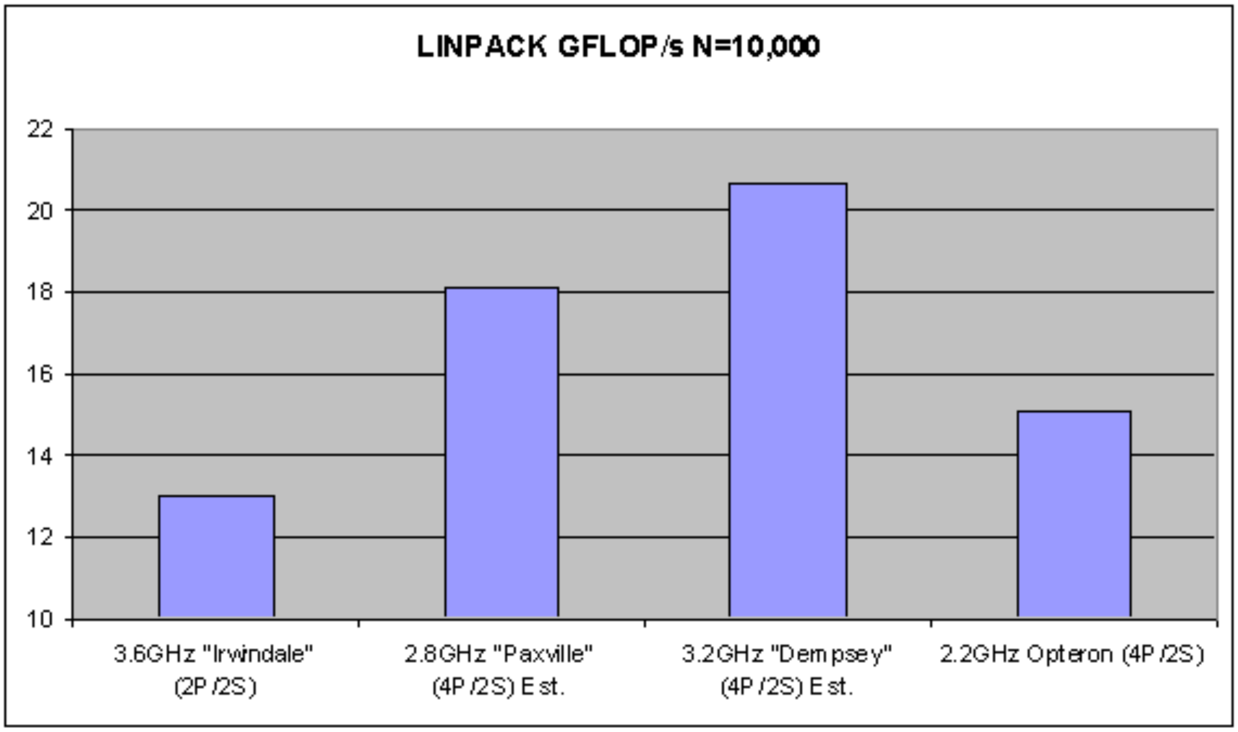


Figure 6 – LINPACK performance, from Intel

It is important to recall that these are performance estimates, and as mentioned in the prior page, there are at least three sources of significant variation. They are not perfect estimates, nor are they unbiased, but they do provide valuable information. RWT will be benchmarking Bensley systems, which will give measured results rather than estimated ones.

Conclusion

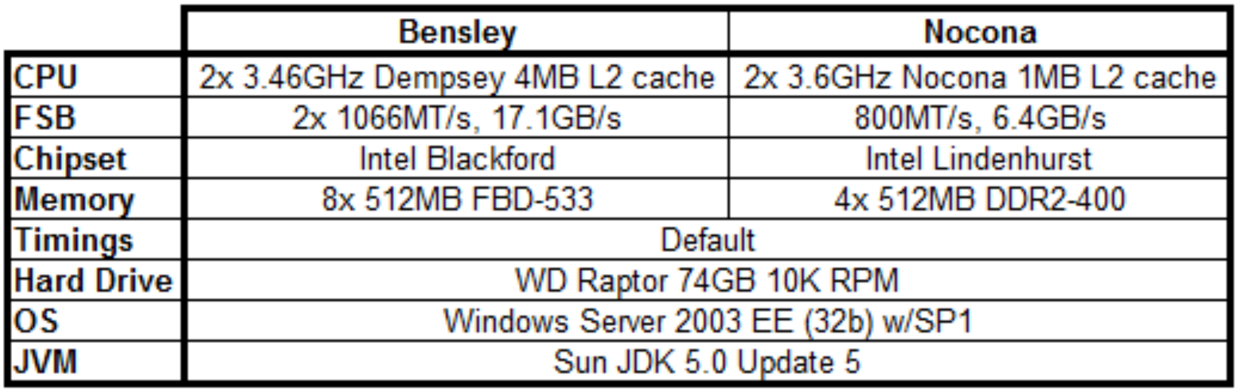
The Bensley platform, to be released by Intel in 1Q06 will be a landmark product launch. Bensley has quite a few technical improvements to look forward to: dual front side buses, a snoop filter and the first implementation of FB-DIMMs. Intel is also beginning to deliver on their promise to provide platforms, not just chips. By any standards, Bensley will be a competitive server and workstation offering and the first good product based on Intel’s dual core MPUs. Performance wise, Intel should be closely matched with AMD, assuming that there are no radical shake ups in the roadmaps. Power consumption and thermal dissipation are an issue; each Dempsey MPU has a 130W TDP at 3.2GHz, and the instantaneous power delivery will need to be higher, but the performance/power is certainly improved over Paxville and Irwindale based systems. Perhaps more importantly, Bensley will offer an upgrade path to the Woodcrest MPU, which is based on Intel’s new microarchitecture. Now that you have seen the performance estimates, the only question left should be “What is the measured performance in a Bensley system?” Fortunately, we are well equipped to answer that question in the second part of this series, which will come later this week or possibly next week.

A Preview of Intel’s Bensley Platform (Part II)

Introduction

In our last article, we outlined the architecture of the Bensley platform with the Blackford chipset and compared it with the prior generation platform with the Lindenhurst chipset and Nocona CPU. We also looked at Intel’s estimates for the performance of fully loaded and highly tuned systems. However, few fully loaded systems are purchased, and many users simply do not have the expertise to tune their entire application stack. Consequently, many of the more prominent industry standard benchmarks represent maximums, rather than reasonably attainable performance measures. Fortunately, Intel was kind enough to provide us with a Bensley platform development kit (PDK) for benchmarking purposes. PDKs are not actual products; they are pre-production systems for key partners to validate hardware, software, drivers, operating systems, compilers and the like. The system we received and tested is fairly modest, and should provide good insights into realistic performance. However, since this system will not be available till March of 2006, the results are an underestimate of actual performance; five months of tuning should improve the performance by a non-negligible amount.

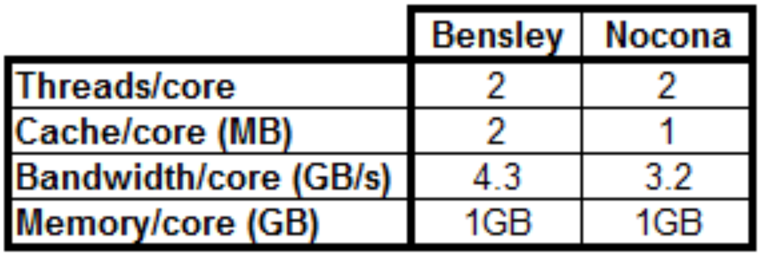
The point of this preview is to examine how the Bensley/Blackford platform will improve over the existing Nocona/Lindenhurst platform. We do not have an AMD K8 system for comparison, nor would it be a particularly insightful; a cutting edge K8 system will likely be slightly dated by the time that Bensley arrives in the first quarter of 2006. Rather than estimating where AMD will be in the future, we will instead compare Bensley with a known quantity. The two systems we use for testing are described below:



	Bensley	Nocona
CPU	2x 3.46GHz Dempsey 4MB L2 cache	2x 3.6GHz Nocona 1MB L2 cache
FSB	2x 1066MT/s, 17.1GB/s	800MT/s, 6.4GB/s
Chipset	Intel Blackford	Intel Lindenhurst
Memory	8x 512MB FBD-533	4x 512MB DDR2-400
Timings	Default	
Hard Drive	WD Raptor 74GB 10K RPM	
OS	Windows Server 2003 EE (32b) w/SP1	
JVM	Sun JDK 5.0 Update 5	

Table 1 – System Configuration

Since we are doing a server comparison, Intel’s Hyper-Threading was enabled for both systems. The Bensley platform appears to have far more resources than the Nocona system, however, that is to be expected; the Nocona system is roughly a year old. Furthermore, the Bensley system has to support twice as many cores, so we would expect it to have roughly twice the bandwidth, memory capacity, etc. More importantly than aggregate statistics, how do the two systems compare on a per core basis?



	Bensley	Nocona
Threads/core	2	2
Cache/core (MB)	2	1
Bandwidth/core (GB/s)	4.3	3.2
Memory/core (GB)	1GB	1GB

Table 2 – Resources per Core

Bensley is certainly a step forward on a per CPU basis, with 34% more bandwidth and twice the cache (which will reduce the bandwidth needs of the CPU, by around &#8730;2).

Our methodology for testing was rather simple. Each system had its own hard drive, and we installed Microsoft Windows Server 2003 and upgraded it to SP1. Then we installed all the relevant software, in particular the JVM and .Net Runtime Beta 2.0. Each benchmark was run 4 times; the first time is to warm up the caches, and then 3 times which were measured. Our results are the averages of these three runs, and after each benchmark, the system is rebooted. Since few benchmarks are perfectly repeatable (i.e. results differ for each run), the standard deviation for the three runs are also reported. Factors that can cause variation include the operating system's scheduling decisions, multi-threading, inter-processor communication, I/O etc. In general, the standard deviations were all very small, and therefore the results can be considered accurate. We will report the standard deviations for all benchmarks but SPECjbb2005, due to licensing concerns.

Cinebench CPU Rendering

Cinebench is a freely available benchmarking suite based on MAXON's CINEMA 4D software, that provides both CPU and graphics benchmarks. Since most servers eschew graphics cards, only the CPU portion will be presented. The CPU test renders "Daylight", a 620×620 scene using Maxon 4D's ray-tracing engine. Daylight has 35 light sources, with 16 shadow maps, according to the documentation. The CPU benchmark can run in both a serial mode, and MP mode. In MP mode, the Bensley system will use 8 threads, and the Nocona system 4 threads. Ray-tracing is an extraordinarily parallel task, and the documentation indicates an 80-90% speed up for the second CPU, and around 10-20% for multithreading on the Xeon. The benchmark is pre-rendered at a small size to move all the data to memory, so there should be no disk accesses during the benchmark. The benchmark can run in two modes, serial and SMP; the latter supports both multithreading and multiple processors. See www.cinebench.com for more details or to download the benchmark.

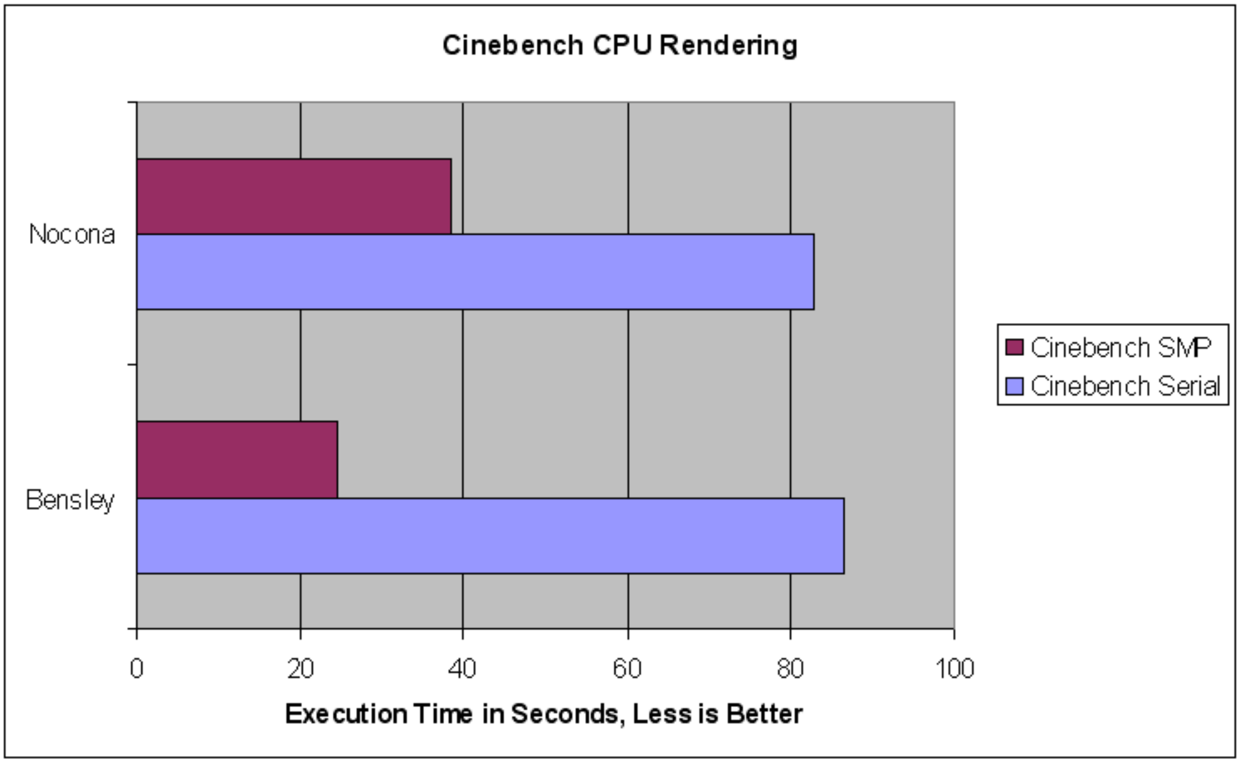


Figure 1 – Cinebench CPU Performance The standard deviations for the Nocona runs were 0.152 seconds for serial and 0.1 seconds for SMP. The Bensley standard deviations were 0.288 seconds, and 0.1 seconds respectively. The Nocona system scaled by a factor of 2.16 from serial to SMP (i.e. execution time for serial was 2.16 times the SMP execution time), while the Bensley system scaled by a factor of 3.51. The Bensley system is slightly slower in single processor performance, likely due to the increased latency of the FB-DIMMs, and the slightly slower clockspeed. However, in SMP operation, which is the relevant comparison, the Bensley system is 50% faster in terms of execution time. This seems low; since rendering is an embarrassingly parallel activity, the improvement should be closer to 70-80%.

POV-Ray 3.7 Beta 9

The Persistence Of Vision Ray tracer (POV-Ray) is a rather famous and commonly used application used for benchmarking many platforms. POV-Ray is freely available from www.pov-ray.org and best of all, the source code can be downloaded for appropriate uses, such as porting to other platforms (currently only Windows, OS X and Linux (i386) are supported). POV-Ray includes a standard benchmarking scene, which was used with unmodified settings. POV-Ray 3.7 is a beta version that adds SMP, SMT and x86-64 support to version 3.6; quite a few changes all at once. The current beta version is 10, but did not work with either platform, so version 9 was used. The benchmark was run in both serial and SMP mode.

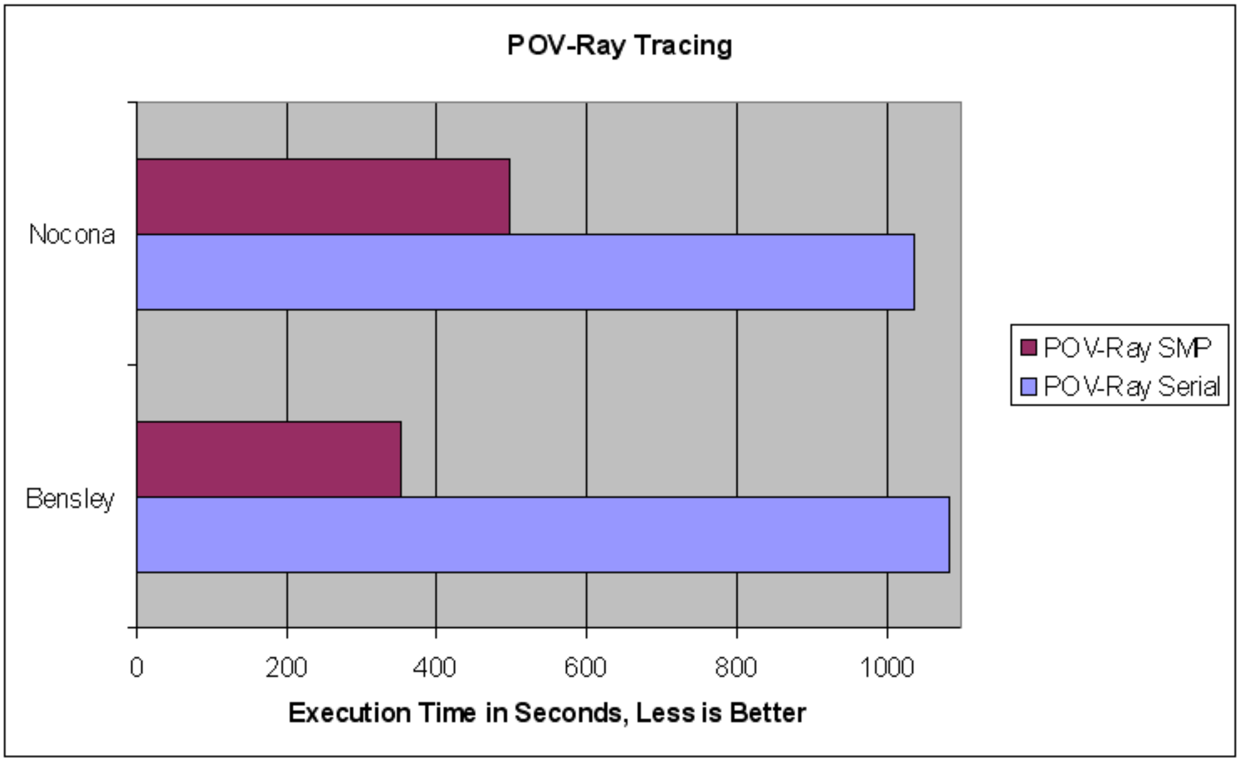


Figure 2 – POV-Ray Performance

The standard deviations for the Nocona runs were 2.065 and 0.813 seconds for serial and SMP modes. The Bensley system had standard deviations of 3.054 seconds, and 4.954 seconds respectively. The Nocona system scaled by a factor of 2.08 from serial to SMP, while the Bensley system scaled by a factor of 3.07. The Bensley system shows an overall speed-up of 41% over Nocona. Again, the Bensley system showed less than expected scaling, although the reasons why are generally unclear.

Black-Scholes Kernel

Black-Scholes is an analytical model for the valuation of stock options, developed by Myron Scholes and Fischer Black, based on the prior work of Robert Merton. Merton and Scholes won the Nobel Prize for their contributions in 1997; Black was ineligible, having passed away in 1995 from cancer. The Black-Scholes model has since been extended to address other sorts of derivatives and is extremely useful because all of the input variables can be observed, which is unusual and convenient, for economics. The Black-Scholes equation can be transformed to the familiar heat equation, which is relatively easy to solve analytically, the most popular methods being Fourier analysis or Green’s Functions.

The Black-Scholes kernel benchmark was supplied by SunGard and Intel in both source form and an executable version (compiled with ICC 9.0). The kernel uses the continued fractions method to approximate a solution to the Black-Scholes equation. The executable takes two inputs: the number of threads to be used, and the number of iterations to be performed. For the benchmark, the number of threads was varied from 1 to 8 by powers of 2, and the number of iterations was fixed at 200,000,000.

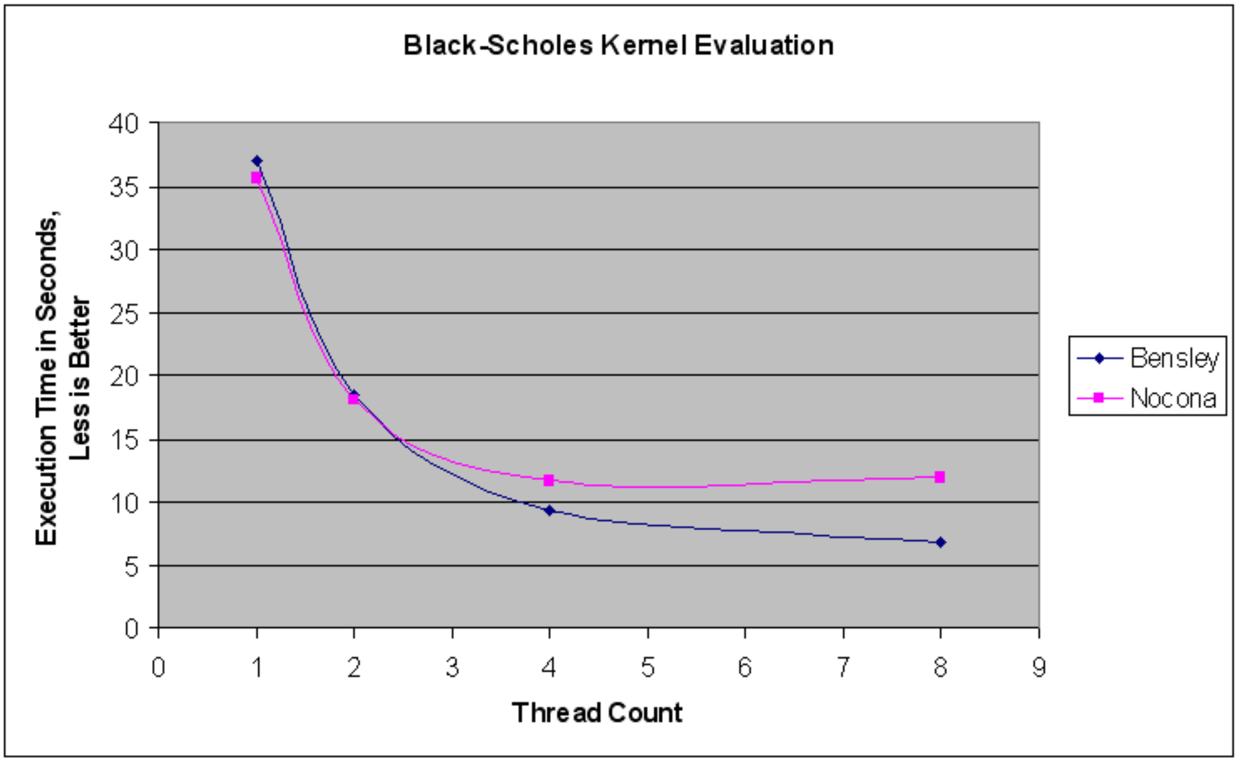


Figure 3 – Black-Scholes Kernel

The standard deviations for the Nocona system were 0.025, 0.321, 0.017 and 0.165 seconds for 1, 2, 4 and 8 threads respectively. The standard deviations for Bensley were 0.015, 0.006, 0.015 and 0.02 seconds. The co-efficient of scaling from 1 to 8 threads for the Nocona system was 2.99, while the Bensley system scaled by a factor of 5.40. The Nocona system scaling is simply astounding, and the Bensley system only slightly less incredible. In order to scale by a factor of 3, Hyper-Threading must add around 50% to nearly perfect scaling from adding another physical processor; the most aggressive estimates that Intel has put forth for Hyper-Threading are 20-30%, so this is quite a surprise. The Bensley system also scales perfectly to four physical processors (which is quite an achievement), and then gets a 35% boost from Hyper-Threading. At eight threads, the Bensley system executed the kernel 74% faster.

SunGard Adaptiv Credit Risk

The SunGard Adaptiv Credit Risk (ACR) benchmark is actually a version of an application from SunGard (www.sungard.com) that was outfitted with a GUI front-end by Intel. ACR uses a proprietary Monte Carlo simulation engine to analyze the risk and return on a hypothetical portfolio of assets. The benchmark portfolio consists of 1428 deals, mostly interest rate swaps and foreign exchange forwards; the portfolio is examined in conjunction with 29 price factors and 19 risk factors. The application decomposes the work across the portfolio, depending on the resources available, so there is a fair amount of parallelism, although some parts are serial or require replication.

While this may sound esoteric, applications like this are extremely common in the financial services industry. Credit risk analyzers are used to issue credit cards, loans, mortgages, bonds, etc. by banks. They are also used by third parties for trading the aforementioned instruments. These applications are considered mission critical by many of the world largest financial companies, including Fannie Mae and Freddie Mac, where they are used to analyze the risk of mortgages.

Unlike the rendering benchmarks, the SunGard benchmark has much finer threading control. The user may set the number of threads at initialization, so that both performance and scaling can be observed. When a Hyper-Threading aware OS is used, preference will be giving towards physical processors. This means that the speed up from Hyper-Threading can be directly measured as the difference between two and four threads for the Nocona system and four and eight threads for the Bensley system.

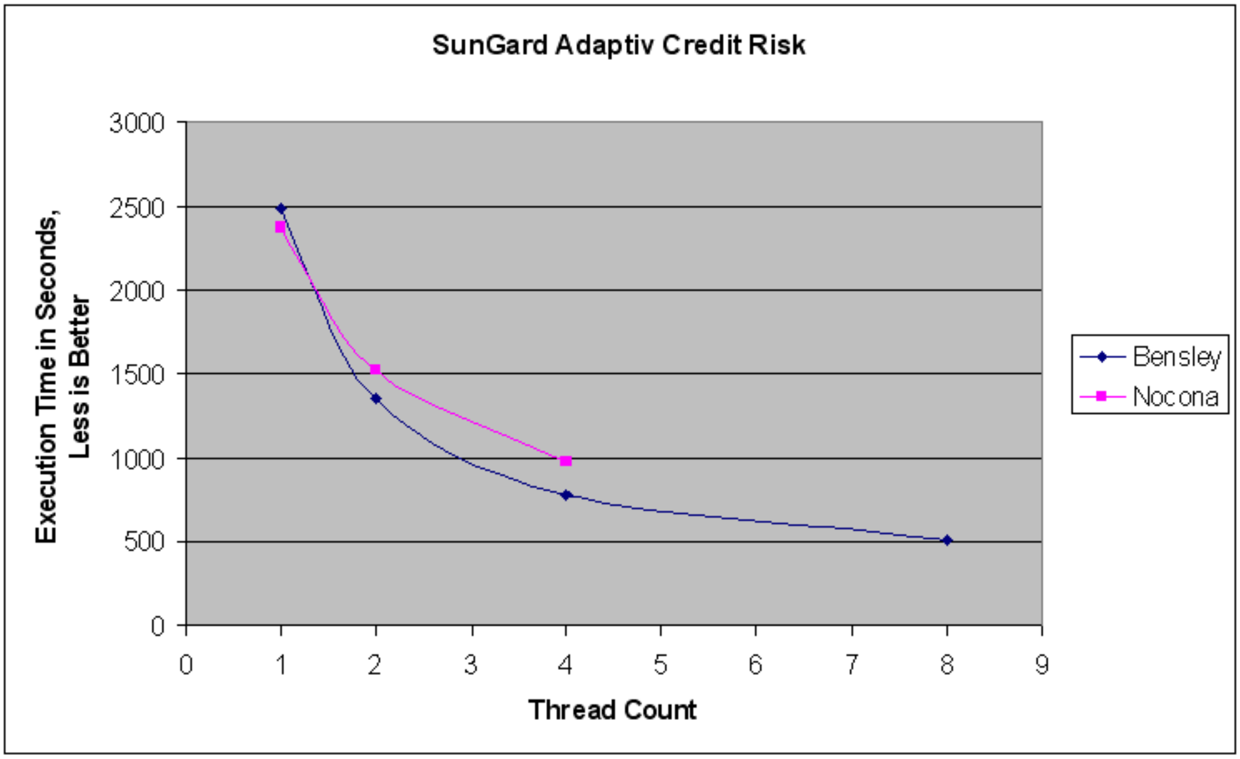


Figure 4 – SunGard Adaptiv Credit Risk Performance

The standard deviations for the Nocona run times were 0.85, 7.33 and 0.66 seconds for 1-4 threads. For Bensley, the deviations were 2.99, 5.16, 2.14 and 2.00 for 1-8 threads respectively. The Nocona system scales by a factor of 2.44, and the Bensley system by a factor of 4.81. The Bensley system at eight threads is 88% faster than the Nocona.

SPECjbb2005

SPECjbb2005 is an industry standard benchmark for evaluating server side Java performance and is a licensed product of the Standards for Performance Evaluation Cooperative (SPEC); SPECjbb is a trademark of SPEC. It is loosely derived from TPC-C, although it forgoes the database aspect and focuses on the performance of the middle tier of a three-tier architecture, implemented in Java.

Quoting from SPEC's website:

"SPECjbb2005 simulates a wholesale company with warehouses that serve different districts. It mimics customer operations such as placing orders or requesting the status of an existing order, and operations within the company, such as processing orders for delivery, entering customer payments, checking stock levels, and requesting a report on recent activity by a given customer.

The benchmark measures throughput of the underlying Java platform, which is the rate at which business operations are performed per second. It steps through increasing amounts of work, providing a graphical view of scalability. Performance is assessed by two metrics: bops (business operations per second), which measures overall throughput for all of the JVMs in a benchmark run, and bops/JVM, which measures the performance and scaling of a single JVM.

The benchmark exercises the implementations of the JVM (Java Virtual Machine), JIT (Just-In-Time) compiler, garbage collection, threads and some aspects of the operating system. It also measures the performance of CPUs, caches, memory hierarchy and the scalability of shared memory processors (SMPs)."

For more information, please visit www.spec.org/jbb2005. Our measurements were done in mid-November using a single JVM on the Nocona and Bensley systems specified in the introduction. The following command line options were used:

```
-server -Xms1500m -Xmx1500m -Xss256K -XX:+AggressiveHeap
```

Each warehouse independently spawns a thread in the benchmark, thus determining the amount of concurrency in the benchmark. Each system has an expected peak number of warehouses (N), which corresponds to the total number of hardware threads supported. The score, measured in Business OPerations per second (BOP/s) is actually an average of the throughputs for N, N+1, N+2…2N warehouses. Due to SPECjbb2005 run rules, we will not be giving actual numbers for the Bensley system.

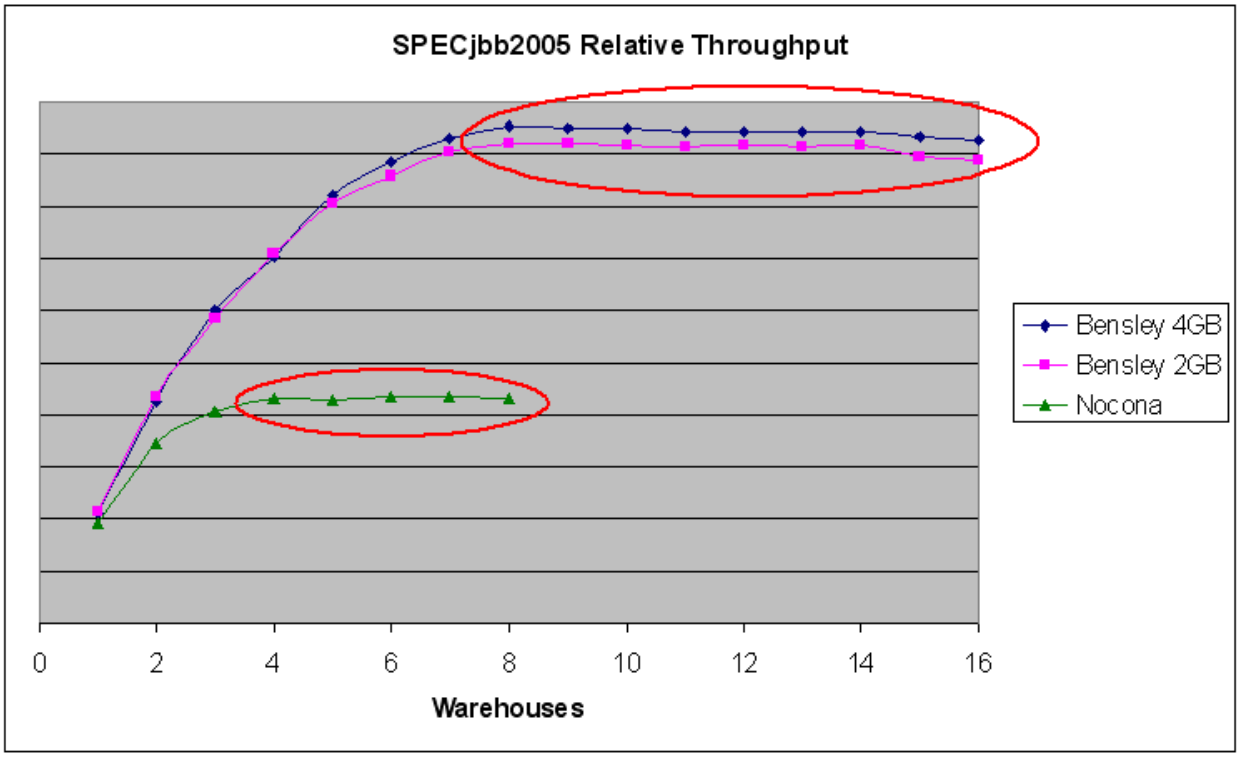


Figure 5 – SPECjbb2005 Relative Throughput

Figure 5 above shows the throughput graphs for three systems: the Nocona, the Bensley (Bensley 4GB) and the Bensley with 1 512MB FB-DIMM per channel (Bensley 2GB). The data points circled in red are those used for calculating actual scores. The Nocona system averaged 21611 BOP/s over three runs, with a standard deviation of 289. Both Bensley configurations score about 2.1 times the throughput of the Nocona system in BOP/s.

Benchmark Discussion and Summary

For this review, five benchmarks were used to evaluate the performance of the upcoming Bensley platform against current DP servers. Two of the benchmarks were rendering applications, one was a kernel of a financial problem, another was a financial application and the last was a Java-based commercial server benchmark. The relative improvement for each benchmark is shown below in Figure 6 and ranges from a little over 40% to slightly under 120%. The relative improvement was calculated as the execution time of the Nocona system divided by the execution time on the Bensley system, minus one (in the case of SPECjbb2005, it is simply the Bensley score divided by the Nocona score, which is equivalent since throughput is the inverse of execution time).

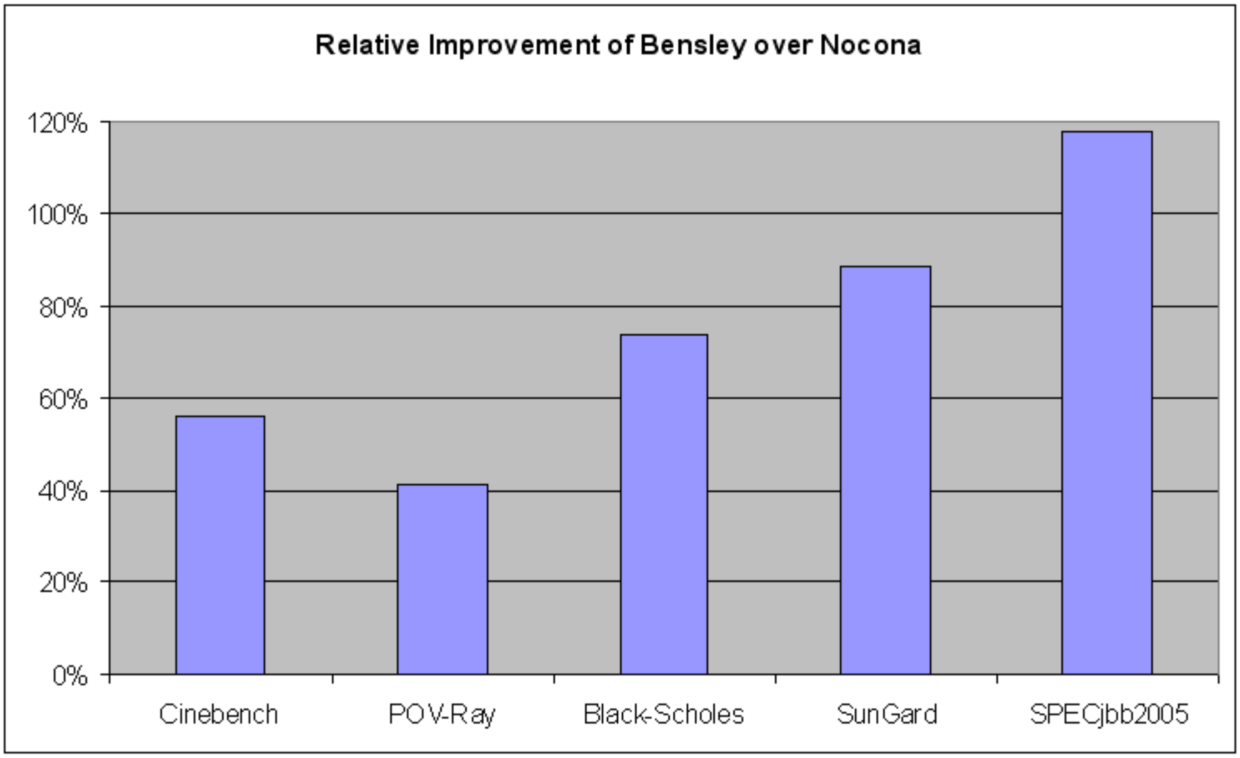


Figure 6 – Summary of Bensley Performance Relative to Nocona

Each of these benchmarks reflects a different sort of workload. The most relevant for the server market is clearly SPECjbb2005, while SunGard and Black-Scholes are good for measuring performance for certain financial applications. Cinebench and POV-Ray fall more into the workstation or Digital Content Creation category; but we are leery of the results. Rendering is an embarrassingly parallel application that is done on clusters; if a workload can efficiently be done on a cluster, then it should scale in a near linear fashion for a shared memory machine. Yet, we observed (and several other reviewers as well) that the scaling falls off after four threads. The most likely explanation is that the algorithms used in these applications were designed only to scale to a certain point. In the past, systems with four threads were the largest commonly sold; larger computers typically were proprietary and rather expensive. Hence it is natural that the authors might choose techniques that are most effective on systems with one to four threads.

These benchmarks supplement Intel's performance estimates by providing realistic measurements. All companies, Intel included, make performance projections for highly tuned systems. Not all servers are so carefully configured, and so our realistic measurements help to elaborate on official claims and estimates. We found a better than 100% performance increase in SPECjbb2005 with two modestly and similarly configured systems. In light of that, Intel's predictions for a 70-90% increase in performance for Java, OLTP and other commercial workloads seem eminently reasonable. Judging from our other benchmark results, the performance improvement is not just limited to commercial server workloads either. Ultimately, our results confirm that the Bensley platform is Intel's first real step into the world of multiple core MPUs, and will provide substantial benefits to many users. As to how it compares with competing offerings, perhaps we will revisit the subject in Q1 next year.