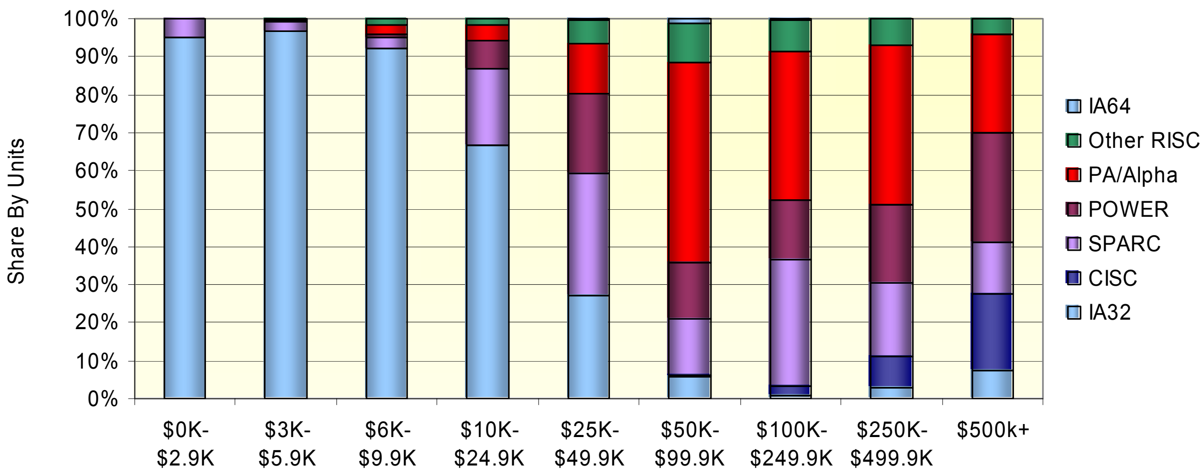
64-Bit Market Outlook

Transcript of Presentation by Nathan Brookwood, Principal Analyst, Insight 64
Delivered at AMD Developer Symposium, Sunnyvale, Calif., September 24, 2002

This morning, I'm going to talk about the 64-bit systems market, a 30 billion dollar market, a very large market by any standard. Today, virtually all the hardware and operating system software that's sold into this market is based on proprietary designs. These systems use proprietary processors including Sun's SPARC, IBM's POWER, HP's PA-RISC and the DEC/Compaq/HP Alpha. They use proprietary operating systems like Sun's Solaris, and IBM's AIX. With Linux and 64-bit industry-standard processors, you have an opportunity to change that. Open systems, based on industry-standard hardware and open software, will enable customers to acquire systems from a range of suppliers, without needing to completely rebuild their infrastructure if they want to mix and match systems from multiple vendors. That's what we're talking about here. It's not just "64-bits is twice as many as 32." Let me show you how I see this playing out over time.

I'm Nathan Brookwood, the principal analyst at Insight 64, a market research firm that focuses on semiconductors used in high-end general purpose computing systems, and the applications for those systems. Here's what I plan to cover this morning. First, I'll give you a little bit of statistical overview on the server market. Then I'll talk about why you are hearing all the fuss about 64-bit computing. Next, I will describe how industry standard processors will change the dynamics of the high-end market. Finally, and I hope that this comes across objectively, I'm going to compare two industry standard 64-bit processors. I will evaluate these processors along a several dimensions and see where they fit.

For the last six months, Intel has been using a chart like the one below to explain why they're interested in 64-bit computing. IDC analyzes the server market by price bands, with categories from under $3,000, $3,000 to $6,000, and so forth, all the way up to systems priced over $500,000. The IDC data shows pretty clearly that x86 architecture dominates the market for systems priced below $25,000; it accounts for 90-plus percent of units sold. There are a few Solaris and AIX systems; a few customers didn't get the message and bought inexpensive RISC systems from Sun and IBM. Basically the low-end is x86 territory. Once you pass the $25,000 price point, you start to see the SPARCs, POWERs, Alpha's and PA RISCs. When you get to the far right, systems priced over $500,000, you see IBM mainframes, high performance clusters and super computers from suppliers like Cray and SGI. These architectural splits across the price bands look pretty much the same, whether you break the data down by units, as we've been doing up till now, or by system revenues. That's a characteristic of price band analyses. The averages in each band are about the same. So units and dollars correspond, more or less. Whether you count by units or dollars, x86-based systems dominate the low-end server market. Nevertheless, proprietary 64-bit systems dominate high end system revenues.



Source: IDC

There's one problem with these charts like these; they normalize each price band to 100 percent. Each segment is the same size, exactly 100 percent, whether it contains one thousand or one million units. You don't get a sense of the differences in the sizes of the segments. When look at number of units in each price band, some really amazing insights jump out. Virtually all of the unit shipments fall into the lower price bands. This is just mind boggling data from my perspective. 95 percent of all servers sold, as tracked by IDC, sell for prices of less than $25,000. There were 4.3 million servers sold in the 12 months that ended in March 2002. Approximately four million of them sold for less than $25,000; only 200,000 units, more or less, sold for over $25,000. So, if you are developing software and you're interested in a market that is measured in units, because every seat, every machine, represents a possible sale, who cares about the high-end segments? If you're Oracle, and you can charge $50,000 per processor for a software license, that high-end might look interesting. But if you can't charge $50,000 or $100,000, that high-end doesn't even show up on your radar screen. As a matter of fact, you scratch your head and you ask why anybody would ever bother to develop these high-end systems.

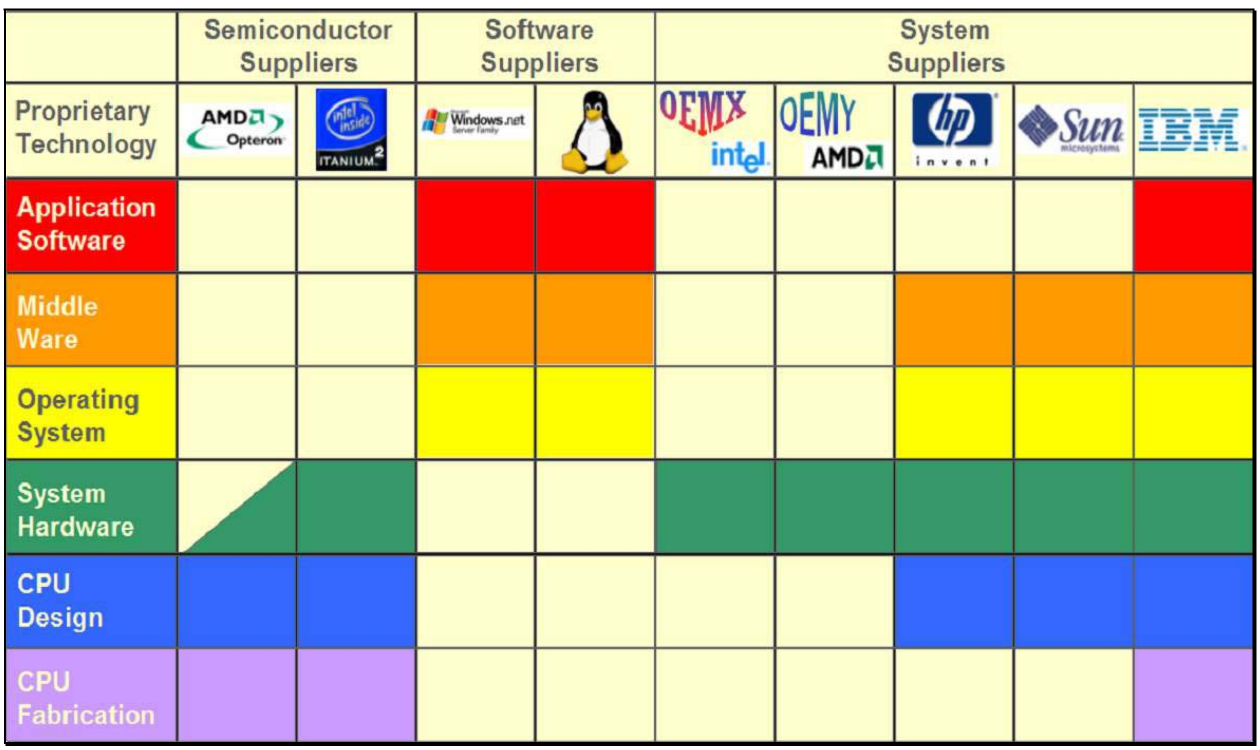
Well, let's take a look at this same kind of price-banded analysis, applied to system revenues. Viewed from this perspective, system revenues are evenly distributed, with about 10 percent of revenues in each band, except at the very high end, where systems selling for half a million dollars and up account for approximately 30 percent of all server revenues. In fact, 65 percent of all the end user spending for server hardware goes into systems selling for over $25,000. Another mind boggling phenomenon. For system suppliers, the high end market represents an attractive revenue opportunity. Software developers with high ticket packages also find the high end market attractive. Most semiconductor suppliers count on high volume, rather than high prices, to drive their revenues. If your business success depends on high unit volumes, then the high-end market won't look all that attractive to you.

Since many of you are Linux developers, let's take a look at the Linux segment. Linux accounts for about 11 percent of the market, as measured by unit shipments. IDC forecasts that it will grow to about 25 percent of unit shipments over the next five years. That is a 35 percent per year growth rate, making it the fastest growing segment of the server market in terms of units. The overall server market is only growing about 14 percent per year, so Linux unit shipments are growing almost three times as fast as the overall market. We see the same story, although not quite as dramatic, if we measure the market in terms of end user spending instead of units. Today, Linux-based systems account for four percent of server revenues. IDC is projecting this figure will go to 11 percent, almost tripling over the next five years. That's a compound growth rate of 26 percent per year, an impressive rate, given that overall server system revenues are only growing at about 4 percent per year. So, you're in the right place with Linux. Next, I want to show you why you are also in the right place by focusing on 64-bit industry standard platforms.

Why is everyone making such a fuss about 64-bit computing? The answer is rather straightforward. A program running on a 32 bit processor can directly address up to 4,294,967,296 bytes of main memory, an amount we commonly refer to as four gigabytes. Programs running on 64-bit processors can address up to 18,446,744,073,709,551,616 bytes of data and program code, more or less, an amount technically referred to as 16 exabytes. It has nothing to do with CISC, RISC, EPIC, new architectures, or old architectures. It has everything to do with how high you can count in the binary system with 32 bits and with 64 bits. It's that simple. Ten years ago, four gigabytes of DRAM memory sold for approximately $168,000. Few system buyers could afford that much memory, not to mention that back then you would have needed 4,096 one megabyte memory modules to complete the configuration. Today, four gigabytes of memory sells for about $1,000, and fits on four one gigabyte modules. As a matter of fact, I was just checking these prices last night in anticipation of today's talk. On PriceWatch, four gigabytes of PC133 DRAM with ECC sells for about $800. Four gigabytes of PC2100 DDR memory, goes for about $1,200. Both speed grades fit on four 1GB DIMMs. Four gigabytes of memory has become very affordable. We know that memory prices typically go down over time. There's only one period in the last ten years when they haven't behaved this way. You can see a little bump in the data from 1993 to 1995, when prices actually went up. There was an industry-wide shortage and it took a while for suppliers to add capacity. The memory industry refers to that period as the "good old days," and they keep waiting for it to come back.

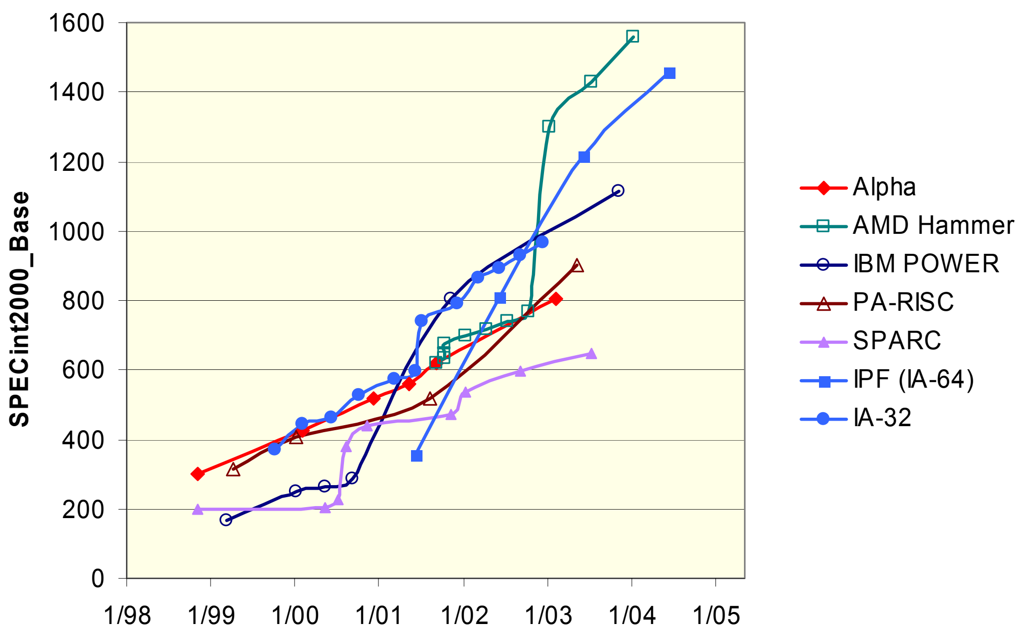
Who's going to use all this memory? Do people really need systems that big? Today you can buy high-end Sun and HP systems with up to 576 gigabytes of main memory. Sun tells me a few of their customers complain that that's not enough. I don't know what people do with all this memory. Maybe you do. Once upon a time I wrote code for a living. I cut my eye teeth on a 12-bit PDP-8 minicomputer. Some of my friends call that my "Insight 12" period. In any event, four thousand 12-bit words of memory for that computer sold for $4,000. A dollar a word. We polished every line of code. We could do some amazing things in 4K of memory. I developed a time sharing system that supported 16 on-line users in 16K of memory. It sold for $50,000, a very attractive price at the time. Given what we could do back then with such limited memory resources, I really can't explain why people need all this memory today. Of course, back then we didn't have graphical user interfaces, TCP/IP, HTML, Java, SSL, and all the other modern conveniences. We barely had high-level languages at all. All I know is that every time memory gets cheaper, customers use more of it. Once they put it their systems, they want their programs to be able to address it and use it in ways that improve performance or enhance system utility. Today's supercomputers, mainframes and mid-range systems are all delivered with memory configurations far greater than 4 gigabytes. Even entry level servers will cross the 4 gigabyte barrier in two or three years, assuming normal erosion in memory pricing. Once that happens, buyers will ask "Why did I buy a 32 bit processor that can't address all the memory in my system?" The four gigabyte barrier will become as much of a constraint on advanced applications as the old 640KB DOS limitation was in the 1980's. That's when the pressure for 64-bit processors, even in low-end systems, will really begin to build.

Now I'm going to shift gears and look at the way the computer systems industry landscape is changing. Modern computer systems incorporate lots of advanced technology, including microprocessors, system hardware (chip sets, motherboards, chasses and such), operating systems, middleware, and application software. Without useful applications, these systems look nice and work well as room heaters, but are otherwise useless. The manner in which system suppliers acquire the technology they need for their systems is undergoing a substantial change. If you look on the right in the chart below, you can see a column labeled "IBM." IBM produces everything that goes into its proprietary systems. They design and manufacture the microprocessors, they design the chasses, they build the motherboards, they design the operating systems, they design the middleware, and they design the applications. The only high-tech components IBM doesn't design these days are the hard drives and the DRAM. Everything else is designed, engineered, and manufactured, by IBM. Amazingly, the entire computer industry followed a similar structure until the early 1980's. The invention of the microprocessor in 1971 began a process that gradually restructured the industry, and today only a small part of the industry continues to operate in this manner.



A few years ago, Sun decided it didn't need to manufacture processors. They designed their own SPARC CPUs, and had TI manufacture them. They designed their own proprietary system hardware and operating software, but encouraged third parties to develop applications for their OS. They changed the model a bit, and reduced the degree of vertical integration they needed to support their system business. HP, with its PA RISC and inherited Alpha-based systems, pretty much aligns with Sun's model. The advent of 64-bit industry standard processors like the upcoming AMD Opteron™ and Itanium suddenly enables an entirely new mode of operation. System OEMs can buy the processor technology they need from Intel or AMD, and they can wrap systems around these processors with core logic, motherboards, chasses, and so forth. They can get software from Microsoft or any number of Linux suppliers. They can obtain applications from a range of software developers. Today's OEMs, working with AMD or Intel, only need to contribute a system hardware platform to deliver a complete system to an end user. Everything else can be outsourced. More and more suppliers will follow this model. The vertical integration model, as practiced by IBM and to a lesser extent by Sun, will soon be placed on the endangered species list.

Performance often plays an important role in system purchasers' buying decisions. If you look at the performance of the popular proprietary and industry-standard processors, and make a few projections, you get a chart like the one shown below. This is based on the SPECint2000 data, which I use as a proxy for general purpose computing performance. Some suppliers prefer TPC-C (and others avoid it completely), but the TPC-C benchmark is very costly to run, and is subject to many factors beyond raw CPU speed. Some testers use lots of small and fast disk drives to elevate their TPC-C scores, although no sane user would ever deploy such a configuration. SPECint2000 tests the processor, main memory and caches. The SPECint2000 data demonstrate that in the past, proprietary processors like Alpha, POWER, and PA-RISC provided superior performance. Over the next year, the upcoming AMD Opteron and Intel's Itanium will dominate these tests. Once they get there, there's no looking back.



Is performance the only buying criterion for high-end system selection? Absolutely not! If it were, Alpha-based systems would dominate the market and Sun would have a very low market share. Obviously, other factors come into play. I lump many of those factors into the category of "economics." Let's take a quick look at the economic advantages industry standard processors have over their proprietary brethren. These advantages fall into two broad categories – those that accrue primarily to processor and system suppliers and those that accrue to the system purchasers. The suppliers get to reduce their R&D expense per processor sold. It doesn't cost any less to develop industry-standard processors, but they sell in much larger volumes. HP spends a lot of money developing PA-RISC processors, and they have to amortize that expense over the limited volume of PA-RISC systems they sell. AMD and Intel amortize their R&D expenses for industry-standard processors over a much larger volume. Industry-standard processors also have lower manufacturing costs than their proprietary cousins. Most proprietary processors are produced in such low volumes that it doesn't make sense to invest in much up-front effort aimed at optimizing production yields, a task that involves running dozens of lots of test wafers. Instead, producers have to take whatever yield they can get from their manufacturing partners.

System buyers also gain economically, in ways that have nothing to do with the R&D or manufacturing costs of the processors in their systems. Their first advantage relates to the increased availability of complementary third party hardware and software. Customers who purchased Alpha systems couldn't find a wide range of applications running on Tru64 UNIX. They couldn't get device drivers for a variety of exotic peripherals; often the hardware supplier would charge a fee to port a device driver over to the Alpha system. This constrained their choices with regard to hardware devices. Economists label the advantages a product gains if it works with a wide range of complementary products "network effects." Successful industry-standard processors typically gain a lot of industry support. They attract lots of partners who in turn develop hardware, software, drivers, and so forth. This is an important advantage. No proprietary processor has ever been able to amass the networks effects that accrue to successful industry standard devices.

End users also gain from reduced "switching costs." This is a polite way to say that buyers of proprietary systems become locked into their system suppliers. Once they have acquired or adapted their software to work on a particular proprietary platform, it is difficult and expensive to move to a different platform. New versions of software must be developed, and users must be retrained. There may be equipment interoperability issues between the systems of the old and new suppliers. Networks may need to be reconfigured. Economists lump all these issues under the category of "switching costs." The low end of the market doesn't operate this way. Users can mix x86-based servers from IBM, Dell and HP, or even unbranded "white box" systems, and run the same software operating systems and applications everywhere. Only those people who go into the computer room and inspect the equipment frames know the specific brands of the hardware they are using. This turns out to be a very compelling model for IT buyers and nets out to improved utility for the end user. You can't deliver this utility at any price if you base your systems on proprietary processors, even if you give away the processor for free in your system configurations.

I believe that as a result of (a) their good performance and (b) their superior economics, industry standard processors will dramatically gain share from the proprietary products that form the norm today-- the Sun's, the IBM's, and so forth. By the end of the decade, industry standard processors will account for about 60 or 65 percent of all the servers that are sold, as measured by dollars. They will account for an even larger percentage, probably around 80 percent, as measured by units. Not surprisingly, the average prices for systems based on industry-standard processors will be far lower than systems based on proprietary processors, accounting for the bigger market share as measured by units, compared with the share as measured by dollars.

Now, this does not necessarily mean that high-end 64-bit systems will become commoditized, the way the PCs have become. It's hard to tell the difference between a Dell PC and a Sony PC or an HP PC. They run the same software, they have similar levels of performance, and they use the same peripherals. There several reasons why the high-end space can retain a higher degree of differentiation between vendor offerings. The functional requirements for desktop PCs are pretty straight-forward so there's not a lot of room for innovation. In the high-end space, there is room for lots of variation. Suppliers can optimize for two-way, four-way, 64-way, or even 512-way expansion capability. There continues to be considerable debate with regard to I/O standards like PCI-X, PCI Express, iSCSI, Infiniband and FibreChannel. Will clustered approaches, SMP or NUMA approaches prevail? All these variations affect system design and give designers more degrees of freedom, and thus more ways to differentiate their products. This opens a range of architectural opportunities for innovation. Since these mid-range and high-end systems sell at high prices, they generate lots of gross margin dollars that can be used to fund R&D efforts. This situation differs substantially from the R&D spending that low-end suppliers can afford. Low-end systems sell for five or ten thousand dollars, with tight margins, so suppliers have little choice but to outsource their technology. All in all, the nature of the mid-range and high-end market, the price points and the margins, will lead suppliers to invest in differentiated products. Since both end-users and suppliers gain from the use of industry-standard processors, the systems that emerge will feature differentiated platforms wrapped around industry standard processors.

Let's explore the range of choices available in the 64-bit industry-standard processor category. There are just two options at this point. AMD and Intel are pursuing radically different approaches with regard to their 64-bit strategies. Intel's Itanium and Itanium II are based on IA-64, an entirely new architecture based on EPIC, an entirely new architectural construct. Intel defines EPIC as "Explicitly Parallel Instruction Computing," but others suggest the acronym might be expanded into "Expect Perfectly Intuitive Compilers," because EPIC places a great deal of emphasis on the advanced compiler technology needed to deliver high levels of performance. EPIC differs from x86 architecture in almost all regards. Itanium retains the ability to execute x86 code for backward compatibility, but its performance when executing this 32-bit code is less than thrilling. It is Intel's slowest x86 processor. We all knew it was slow, but we were never sure how slow. Last summer, my friends at HP put out a white paper on Itanium II performance. The document touts Itanium II's impressive 64-bit performance, but if you read the entire document, you can find a brief note that quantifies its 32-bit performance as well. According to HP, the Itanium II, running 32 bit code, compares approximately to a "350 megahertz Pentium Pro processor." I'm quoting that document exactly. This is a pretty dismal level of performance today. If you try to donate a 350 megahertz PC to an organization that can reuse such systems, they'll probably tell you to put it in the trash instead. So Itanium II's x86 performance is far from exciting.

The current Itanium designs take a more or less conventional approach to the partitioning of system logic. You need a north bridge with a memory controller, and a south bridge that ties into I/O buses. In multiprocessor systems, each Itanium OEM takes its own approach to managing cache coherency and memory switching. This allows OEMs to differentiate their systems, but requires them to make substantial R&D investments in these platforms. Despite all this, Intel has been able to gain lots of industry support for the Itanium family. They have over 20 major OEMs, starting with companies like IBM and HP, and going all the way down the list Fujitsu-Siemens, NEC, Hitachi, SGI and a raft of others that have committed to delivering Itanium based platforms and peripherals. Many of these suppliers have already demonstrated their systems, although only a few are shipping at this point. If you were at LinuxWorld, you would have seen a bunch of them. If you were IDF, you may have seen even more.

AMD has a taken very different approach. Their upcoming AMD Opteron processor, based on X86-64, is a classic CISC; it's like the 32-bit x86 in almost all regards. AMD suggests that anyone who knows how to program an x86 will be able to learn how to program their x86-64 processors in an afternoon. This afternoon, by the way, is when you'll learn how to do it. More interestingly, AMD claims their AMD Opteron will be its fastest 32-bit *and* 64-bit processor. It can run off-the-shelf 32-bit operating systems and applications with no apologies needed for performance. AMD took a very unconventional approach to system partitioning, and integrates many core logic functions onto the CPU itself. It's got on-chip memory controllers and HyperTransport™ interfaces that make it easier for OEMs to construct multiprocessor platforms. These may not be the world's most differentiated platforms with regard to other AMD Opteron platforms, but they will be well differentiated from the Xeon boxes used in most current x86-based server installations. The one wild card here is how much industry support AMD can gain for its upcoming AMD Opteron processor. This Developer Symposium is a good start. We'd like to see another session like this for hardware developers, and another one with IBM, HP and other system vendors in the audience, a la IDF. AMD needs to demonstrate that it can get industry support behind this processor.

Let's compare these two processors. We have two nominees in the 64-bit industry standard processor category. On your left, we have Itanium II, 400 square millimeters, 150 watts, selling at prices of $2,000 to $4,000. On your right, we have the upcoming AMD Opteron, with as yet unknown size, thermal characteristics and pricing. AMD has said it's a pretty small dye, so it might be more cost effective in that regard. We're going to compare these two entrants in terms of performance, scalability, reliability, software availability, network effects and economics, and then we'll try to net all this out.

With regard to performance: Itanium II has more than respectable, maybe even great 64-bit performance. But, as we noted earlier, its 32-bit performance is nothing to write home about. The AMD Opteron should be good on both 32-bit and 64-bit codes, and has a shot at delivering the highest SPECint2000 scores in the industry when it is introduced next year.

With regard to scalability: Intel places the burden of building scaleable platforms on the OEM, and these OEMs plan to deliver systems ranging from four-way SMP solutions up to 128-way NUMA configurations. SGI is even adapting its Origin 3000, which can accommodate up to 512 processors. The upcoming AMD Opteron should be really good at two and four-way processor configurations. Tomorrow, Fred Webber will show you how it can be stretched to eight. A few startups are talking about 64 processor systems in a few years. Given that AMD is still finding its way in the server market, and has yet to establish its reputation with these very conservative buyers, if AMD could deliver a 64 way system with great performance tomorrow, I suspect most buyers would still be a little reluctant to go with it.

With regard to reliability. Both chips include error correcting code (ECC) on all critical paths and both include machine check architectures that can detect hardware problems and turn off key parts of the chip.

With regard to software availability: Intel claims over one hundred 64-bit applications have already been released on Itanium, with more to come over the next 12 months. These include a range of Unix, Linux and Windows environments, along with many tool suites, database systems and utilities. Given the performance of Itanium's 32-bit mode of operations, Intel needs to ensure that virtually all key software applications migrate from its x86 platforms in order to secure Itanium's long term success. The AMD Opteron can claim that virtually all current 32-bit applications will run with acceptable performance, and we've seen commitments for 64-bit operating system support from Microsoft and several Linux suppliers. AMD still needs to persuade the developers of high-end 64-bit packages that run on today's proprietary RISC platforms to port their software to the upcoming AMD Opteron. It's unclear how well they are doing in this regard, but the attendance of so many Linux developers at today's session is certainly an encouraging sign.

With regard to economics: Itanium is a big chip that is sold in relatively low volumes. The R&D investments Intel makes in its Itanium line do little to bolster its desktop Pentium products, and vice versa. The AMD Opteron is a small chip that will be sold in high volumes. AMD gets to apply much of the R&D effort it puts into the AMD Opteron line across its desktop processor roadmap as well. The economic advantage clearly goes to AMD.

Overall, despite its problems with 32-bit performance, Itanium will do well against the 64-bit RISCs, given its impressive 64-bit performance and the economic advantages that come from its position as an industry standard processor. I expect Itanium, Itanium II, III, and IV, to substantially impact the proprietary system businesses of Sun and IBM. AMD, with its upcoming AMD Opteron, should prove an effective competitor to Intel's Xeon line. But it will not make substantial inroads here unless it gets the support of at least one of the 'Tier One' server suppliers.

Let's review what I've discussed this morning. Today the 64-bit market is dominated by high-end proprietary systems. This will change dramatically over the remainder of the decade. Industry-standard 64-bit architectures will move down to the low-end of the market and up into the mid-range and high-end segments as well. The people who understand this, and who adapt their applications for these industry standard 64-bit boxes, will win big-time when these systems become available. The folks who ignore this trend and continue to focus their efforts on proprietary platforms will soon be in trouble, if they're not there already.

Thank you. I've enjoyed the opportunity to be here today. I hope you've learned something from this. I will be around for the rest of the day. Feel free to grab me during the breaks, and tell me what it is you're doing with all this memory that the DRAM suppliers are trying to shove down our throats.

Disclaimer: The views and opinions stated in this transcription do not necessarily reflect those of AMD. Additionally, AMD does not vouch for the accuracy of the claims,

benchmarks and forward-looking statements included in the document. It is provided for informational purposes only and does not constitute an endorsement or recommendation of any kind.