IBM System x: From Boxes to Workloads

Insight

Gordon Haff
2 March 2010

Pick any point in time during the modern computing era and you'll find some systems designed and sold as pre-integrated, task-specific platform—and others sold largely as piece-parts that buyers have to assemble. The more integrated systems tend to be the more mature ones, and the ones used for more critical jobs. However, new waves of computing gain footholds by being faster and/or cheaper than what came before; customers are often willing to put up with less integrated and self-contained solutions to get speed and economy.

Over time, upstarts often become mainstream, and customers start demanding that they, *too*, come packaged and optimized for specific tasks. There is perhaps no better example than Unix systems. They started out life as a tool for engineers, but became—and remain, in many cases—the back-end systems that run some of the most critical workloads in business today. As they evolved, so, too, did the requirements for how they were packaged, sold, and supported. Unix vendors once presented their systems as the antithesis of the monolithic mainframe; today, they largely emulate mainframe-ish attributes like high scalability, dependability, predictability, and integration.

So it is with x86, which first appeared in the 1980s in servers running in workgroups and other distributed locations. System makers initially mostly just sold boxes. In fact, until relatively recently, they often didn't even pre-assemble the hardware; buying a "rack of servers" meant receiving dozens of separate boxes at your shipping dock. And software? Well, that was something you had to buy from someone else and install yourself—even core software, like the operating system and management tools.

Over the years, x86 came of age and helped usher in an era of horizontal integration whereby different vendors specialized in different layers of the hardware and software "stack." Intel and AMD did microprocessors, Microsoft did an operating system, various disk drive makers did their piece, and so forth. This specialization brought with it often lower acquisition costs relative to products from the vertically integrated soup-to-nuts computer companies that were once commonplace. And it meant that buyers had choices—lots of them. But something that just *works?* Not so much.

System makers responded by developing offerings that bring together more of the pieces needed to accomplish a particular type of task. Moving beyond factory bundles, the idea is to integrate products and technologies that match up well to specific workloads from the get-go. One form this takes is a "solution" or appliance that includes everything from server to software, such as a database appliance. However, between *á la carte* and complete solution are configurations that optimize around requirements of a given workload, but don't necessarily prescribe a single choice of middleware or application. Blade servers, which have come to be as much about integration as density and other attributes touted early on, are one example.

Even more explicitly on point is IBM's workload optimization theme. An integral component of IBM's corporate-level Smarter Planet initiative, workload optimization is also the organizing principle behind IBM's launch of its new x86 server lineup. In short, workload optimization is the lens through which IBM increasingly views its systems. That may not be that much change, except in language, for a System z mainframe. But it's a significantly different look for x86 systems like a System x Intel-based server.

Why and What of Workload Optimization?

Applications—and even the individual modules of a large composite application—can have very different characteristics.

Take an example from IBM's Smarter Planet initiative such as "Smart Traffic."[1] As with many applications that seek to tame the problems of the complex, always-in-motion real world, Smart Traffic is fundamentally about integrating multiple streams, views, and ways of using interrelated data into a single application.

Electronic toll collection for autos and public transit, for example, are transaction processing systems that are themselves part of more complex business applications. They connect directly to consumers' financial instruments such as bank accounts and credit cards, and require correspondingly high degrees of security and transactional integrity; they also often perform best on larger SMP, i.e. "scale-up," servers.

Traffic flow prediction, on the other hand, is a business analytics workload. It involves various types of models, including neural networks, looking for patterns in a large volume of real-time and historical data. This requires enormous processing horsepower, but often lends itself to highly parallel queries using a cluster of smaller servers.

Many of these applications also need a public face. As we move beyond payment interfaces and relatively static information to real-time data, the requirements increase dramatically—especially when data of all sorts is increasingly "mashed up" with other services, especially into location-based applications running on mobile clients.

Virtualized applications are yet another wrinkle. Server virtualization can dramatically improve system utilization. However, as the processor gets driven harder, the load on the rest of the system rises proportionately. What was an appropriately balanced system design in the physical world may no longer be so balanced once workloads are virtualized. Probably the clearest example is that virtualized systems usually need far more memory.

The bottom line is that different workloads tend to align with different types of systems—or, as in the case of eX5 described below—with system options that accelerate or otherwise assist specific usages.

The eX5 Generation

Viewed through a traditional "server box" lens, eX5 is IBM's new portfolio of high-end servers based on Intel's "Nehalem-EX" processor line. It's the successor to IBM's scalable Xeon designs going back to the Summit chipset in 2001.[2]

[1] See our Horses for Courses: Optimizing for Workloads for an expanded look at workload optimization applied beyond x86 architectures.

[2] See our IBM's Uniquely Scalable Xeons

Nehalem-EX's eight cores per chip will double the count of its predecessor. Hyper-Threading, Intel's version of simultaneous multi-threading (SMT), doubles again the number of threads the processor can handle at the same time, to 16.[3] 24MB of shared cache keeps a large pool of data close to the cores working on it. That data can also be brought in from memory more quickly than in previous generations, because Nehalem-EX integrates its memory controllers onto the same die as the processing cores. Nehalem-EX also now connects processors together using the QuickPath Interconnect (QPI), a high bandwidth point-to-point link, rather than the parallel front-side bus (FSB) used in its predecessors. QPI avoids much of the contention for resources that can limit scalability when a bus connects processors with each other and with memory.[4]

IBM, Unisys, and NEC have previously created scale-up Xeon designs based on their own node controller designs and silicon. With Nehalem-EX, each OEM still needs to design its own custom node controller. But it now plugs neatly into a defined architecture, rather than having to integrate into a front side bus that wasn't really designed for that purpose. In IBM's case, its scale-up system is the x3850 X5. (In its workload optimized configurations, IBM calls it the x3950 X5.) It will initially scale to two nodes; at four sockets per node, eight cores per socket, and two threads per core, this 64-core, 128-thread, 3TB of memory system is *enormously* scalable by any historical x86 standard.

The x3850 X5 is clearly the high-end of, and flagship for, the eX5 generation, but it doesn't sail alone. In its armada are the x3690 X5 (a two-socket server that is scalable to four sockets) and the HX5 (a two-socket blade server that is scalable to four sockets).

Faster Storage

While IBM will continue to happily sell server hardware in the usual piece-parts way, with eX5 it is shifting customers toward buying integrated configurations that are optimized for specific types of workloads. The underlying technology, including the basic servers, remains an important part of the ultimate solution, of course. But the focus of eX5 is on delivered value to specific application types, rather than on the servers themselves.[5]

Workload-optimized systems start with two options that target common performance bottlenecks.

The first is the eXFlash, a bundle of up to eight solid state drives (SSD). The x3690 X5 can be configured with up to three eXFlash units (24 SSD) and the x3850 with up to two (16 SSD).

The rationale for eXFlash—and indeed the burgeoning popularity of SSDs for servers in general—is that, for many applications, the number of disk drives has been determined not by capacity requirements but by performance considerations. For an extreme example, one need only peek inside a vendor's performance lab configured for running a TPC-C benchmark, a commonly-used metric that aims to simulate an online transaction processing (OLTP) workload. Rows and rows of disk drives dwarf the system under test. IBM's 6 million transactions per minute result in 2008,[6] for instance, required almost *eleven thousand* disk drives to feed the test system—and they were high-performance 15,000 rpm disks at that.[7]

The traditional knock on SSDs is that they cost a great deal more per GB than spinning media does. That's still true, albeit to a lesser degree than was once the case. However, for environments where I/O per second (IOPS) are the limiting factor, SSD's price premium becomes less important because the

[3] See our Gradations of Threading. Unlike the addition of cores, Hyper-Threading doesn't actually add more execution units. However, it helps a processor avoid idling the execution units it does have while waiting for data to arrive from memory.

[4] Although Intel has not yet formally announced Nehalem-EX, it has publicly disclosed many details in advance of the launch. See e.g. tinyurl.com/phnga9

[5] That shift alone shows how much x86 has matured.

[6] tinyurl.com/ygnqhn3

[7] The number of I/O operations per second (IOPS) that storage based on hard disks can produce depends directly on how many "spindles" (disk units) are in operation, and their rotational speed. In OLTP, one runs out of IOPS long before storage capacity.

far higher throughput of SSDs lets you buy many fewer of them.

How many fewer? Well, as with many things related to performance, it depends. It depends on the mix of read and write operations, the nature of the application, and the data protection requirements. However, IBM estimates that a single eXFlash can provide high speed access for "hot" data at up to 240,000 IOPS, a performance level that could require almost a thousand "spinning rust" disk drives.

The SSDs in the eXFlashes will typically be used to store heavily-accessed, but relatively small, data structures such as database indexes, scratch space, or, in the case of business analytics, pre-set queries. IBM says that it also has Wall Street customers writing in-house applications that make use of an SSD tier. Connecting Flash memory cards via PCI Express is an alternative to SSDs. IBM argues that because the SSDs are front accessible, it is easier to replace them than cards. Hardware RAID is also an option with SSD, and IBM expects that to be used by any customers writing persistent data to SSDs.[8]

In both cases, the placement is currently a manual process, unlike the automated storage tiering products in disk arrays from the likes of HDS and EMC.[9] In part, this reflects that Flash implemented in a storage array, rather than a server, almost *has to* handle data placement more autonomously as it will typically be a common resource for multiple servers and applications. An individual server, on the other hand, can be configured specifically for the application running on it—that's really the idea behind workload optimization.

It's All About the Memory

The second, and arguably more differentiated, eX5 option is MAX5 (Memory Access for eX5). Having more memory benefits many types of heavy-duty workloads. As a result, a well-designed memory subsystem has always been an important aspect of balanced system performance going back to the early days of computing. However, it's the combination of heavy workloads (or many lighter-duty workloads) and virtualization that really pushes the design envelope. And it pushes memory hardest of all. That's the main target for MAX5.

Virtualization is a great way to increase server utilization. Average CPU utilization in the single digits is not uncommon for unvirtualized x86 servers. It was that single statistic which thrust server virtualization into the limelight and made VMware's fortune. Even if near-100 percent utilization isn't especially realistic on x86, even bumping CPU utilization up to fifty percent or so is eminently doable and a big win.

One consequence is that the load on the other parts of the system goes up correspondingly. More VMs mean more operating system images and applications storing and accessing bits in memory. If you're running four virtual machines simultaneously on a server that would otherwise just be running one, you potentially need 4x the amount of memory as well.[10] Virtualization's almost insatiable demand for memory means that memory capacity, rather than CPU horsepower, is often the factor that limits how many VMs can be packed into a single server. Put another way, more memory can reduce the number of physical servers needed for a given application environment—and the number of virtualization software licenses needed to run on those servers.

Taken by itself, Nehalem-EX already boosts memory capacity considerably relative to its predecessors. It uses a "scalable memory buffer" on each of four links out of the processor. Each of these buffers has two DDR3 memory channels with two DIMMs per channel. This provides for a hefty complement of 1TB of memory even on a modest 4-socket server.[11]

[8] PCI Flash cards can use software RAID.

[9] See our EMC rolls out FAST.

[10] In practice, the situation isn't quite so dire. Techniques such as "memory ballooning" help to release memory not current being used by a given VM. Still, virtualized systems are memory hogs.

[11] 4 processors * 4 links/processor * 2 channels/link * 2 DIMMs/channel * 16GB/DIMM = 1TB.

Server	*Storage*	*Use Case*	*Max IOPS*	*Max eXFlash*
x3950 X5	Direct Access	High speed read cache	980,000	3.2TB
x3950 X5	Hardware RAID5	High IOPS redundant data	174,000	3.2TB
x3690 X5	Direct Access	High speed read cache	720,000	4.8TB
x3690 X5	Hardware RAID5	High IOPS redundant data	174,000	4.8TB
HX5	Direct Access	High speed read cache	250000	640GB

Database optimized models with eXFlash

The MAX5 option uses QPI links to connect to a custom silicon chip ("Firehawk") which has its own memory controllers; this chip also acts as the node controller that scales the x3850 X5 to multiple nodes. A MAX5-equipped four-socket rackmount server supports up to 96 DIMMs—half again as many as the 64 DIMMs in an off-the-shelf Nehalem-EX design.[12] Memory-intense deployments such as virtual servers and database engines benefit especially from this expansion.

Workload Optimized Systems

The easiest way to discuss how these concepts and components come together is to look at IBM's specific eX5 offerings.

The database-optimized models are about I/O performance. IBM's basic value proposition here is "less expensive for the same performance." For situations where performance requirements demand many disks, that cost difference can be dramatic; IBM offers up scenarios in which a eXFlash-equipped x3950 X5 can consolidate the equivalent IOPS of 1,600 disks, or 10 racks, of storage into two eXFlashes. (Alternatively, SSDs can be used to increase performance if the cost of a large number of high-performance disks results in configuring fewer spindles than are needed for maximum performance.)

The different models map to different use cases. Read-mostly workloads, such as business analytics, don't necessarily need to protect against a drive failure using full RAID 5, and will tend to prioritize read performance instead. On the other hand, database applications that write data as well as read will typically want to ensure that there's *no* possibility of losing a transaction—notwithstanding that SSDs are considered to be more reliable than spinning media.

IBM also is looking at solutions atop these systems for customers who want a complete IBM stack. IBM Balanced Warehouse, for example, is a business intelligence warehouse for data analysis and forecasting that uses InfoSphere Warehouse. UDB Database provides an integrated DB2 environment with an optional PureScale add-on for continuous availability or the SolidDB in-memory database for accelerating read operations.

Whereas the database-optimized systems focus on I/O (or, more specifically, on I/Os per dollar), the virtualization-optimized systems focus on memory capacity. Raw memory capacity is a technical specification, but that's not really the point. Rather, the point of these MAX5 systems is to increase the number of virtual machines that can cost-effectively run on a physical server. This reduces the number of physical servers and virtualization software licenses required for a given workload.

In other words, this is a statement about cost-effectiveness more than it is about capacity for capacity's sake. Yes, MAX5 can increase system memory capacity by up to 512GB for each node. This translates into support for up to 1.5TB of

[12] MAX5's blade version supports up to 80 DIMMs.

Server	*Environment*	*CPUs*	*Max DIMMs*	*Max RAM*
x3950 X5	VMware ESXi 4.1 or other hypervisors	4	96	1.5TB
x3690 X5	VMware ESXi 4.1 or other hypervisors	2	64	1.0TB
HX5	VMware ESXi 4.1 or other hypervisors	2	40	320GB

Virtualization optimized models with MAX5

memory, a 50 percent increase. However, just as significantly, more DIMM slots means that a customer could configure, for example, a 768GB system using 8GB DIMMs rather than the more expensive 16GB ones and save a significant amount of money. The two-socket x3690 also goes to 1TB with MAX5. Or can hit 256GB with relatively cheap 4GB DIMMs. Using the densest memory parts rarely makes financial sense in the real world unless you have truly extreme requirements or are lucky enough to have an unlimited pocketbook.[13]

Unsurprisingly, given that VMware remains the 800-pound gorilla of the x86 virtualization space, these systems include the latest VMware hypervisor as a primary option. However, IBM's virtualization strategy for x86 is cross-platform, supporting multiple popular hypervisors.

In addition to four-socket configurations, IBM also leverages the two-socket version of Nehalem-EX for configurations that don't require the compute horsepower of a four-socket server but can benefit from its memory capacity and bandwidth, as augmented by MAX5. This reflects the fact that many servers running virtualized workloads are limited by memory rather than processors.

Conclusion

The suggestion that customers in all their diversity want to, or will, universally acquire their computing in a singular manner is, to put it simply, simple-minded. Stories suggesting that Software-as-a-Service, or Windows, or *whatever* will conquer all may get page hits—as is doubtless their intent—but this is not how things work in real-world IT. Horses for courses; different strokes for different folks—pick your metaphor. Almost from the beginning, the IT industry has encompassed many different ways of buying and using computing.

That said, we do see broad patterns. We've moved to a more interoperable world with vendor lock-in much reduced from the historical norm. Many components today are primarily sourced from companies that specialize in their design and manufacture. No one "does it all" in the sense they might have as recently as the 1990s.

One of those broad patterns that we see today is a shift away from piece-parts towards more integrated offerings in which the technology moves to the background and the system is about the workload it's matched to rather than its piece parts..

IBM has long done a better job than most IT vendors at pitching its offerings in terms of customer business value rather than just technology. Yet, even at IBM, System x announcements have still often led with speeds and feeds, capacities and technologies. The eX5 takes a very different approach. It's not that IBM won't (happily!) sell you individual eX5 boxes if that's what you want, but its lede is that something bigger and broader is where value most lies.

[13] The most-dense DIMMs almost always carry a huge price premium over technology a step or two down. The highest-density DIMMs are used more for benchmark configurations than customer apps.