Trends in High-Performance Computing Requirements for Computer-Aided Drug Design

George Vacek[*,1], Dave Mullally[2] and Knute Christensen[2]

[1]*Hewlett-Packard, 2630 East 1st St., Duluth, MN 55812, USA*

[2]*Hewlett-Packard, 3000 Waterview Parkway, Richardson, TX 75080, USA*

Abstract: Computer-aided drug design (CADD) has become a mainstream component of the drug discovery and development process. High Performance Computing (HPC) provides the power that allows CADD researchers to explore more designs in less time, and some of the greatest improvements in CADD result directly from advances in HPC. This paper examines some of the more significant trends in HPC that influence computer-aided drug design (CADD).

Keywords: High-performance computing, blades servers, multi-core processors, grid computing, hardware accelerators.

INTRODUCTION

Any conversation about high-performance computing (HPC) should begin by normalizing the definition of HPC. For our purposes, we consider HPC to be: “Use of computer systems to simulate or model real-world processes that are too big, too small, too fast, too slow, or too expensive to easily observe empirically”. Generally HPC involves describing such processes mathematically and then developing algorithms to solve the math on computer systems.

HPC evolved from huge, expensive computer systems on raised floors manned by computer scientists in white lab coats; the actual science being practiced was secondary to maintenance of the equipment, development of specialized application software, and job turn-around time. At that time, only the government or very large companies could afford the big iron that was necessary to carry out the types of simulations that would make a difference to a business. Today HPC is an integral part of most businesses that have active research and development, with expanding usage. There are two major factors at work in this expansion: (1) the increasing need for models to more accurately describe real-world events, requiring more HPC resources, and (2) the “commoditization” of HPC-class systems, making them more widely available. HPC is now so pervasive, and such an integral part of the business of research, design, and manufacturing, that without it businesses cannot complete enough research innovation to remain competitive. In this paper we will review some of the more important trends in HPC, especially those trends that impact HPC in computer-aided drug design (CADD).

THE NEED FOR MORE

Most HPC problems, including those in CADD, have a common theme–the more computing resources available, the better the answer, i.e. the more accurate a model that can be used (or the larger the system that can be studied, the more simulations that can be run, the longer the timescale that can be modeled, the larger the space that can be searched, etc.). As a result, a typical researcher cannot get enough computational resources–this kind of user will acquire the maximum amount of performance available with their budget constraints.

In industries that employ HPC, model sizes are growing and model complexity is increasing in order to address challenges such as:

- The need to meet more stringent government regulations,
- The need for increased product quality,
- The need for lower product and production costs.

Computer-aided drug design is no exception; in fact, CADD arguably stresses computer architectures across a wider range of problem types than nearly any other HPC discipline. From docking or sequence analysis (which require loosely coupled clusters of smaller servers) to physico-chemical property or dynamics simulations (which require faster interconnect or even shared-memory architectures), CADD scientists can always utilize additional computing power, as shown in Fig. **1**. This diagram breaks computing requirements into a two-dimensional spectrum, and describes an application’s computing requirements. The Y-Axis ranges from I/O and memory intensive to compute intensive, while the X-Axis ranges from high throughput demands to large simulation demands. This simple diagram guides understanding of the type of multiprocessor system best suited for application scalability from distributed-memory clusters to shared memory symmetric multiprocessors (SMP), as well as the necessary qualities for each node (not only processor type, but also I/O and interconnect requirements, local and shared memory and disk, etc.). Memory intensive large simulations, as typified by some quantum chemistry jobs, have best performance on large SMP systems; applications in other quadrants are better served by clusters. For High Throughput jobs that are not compute intensive, e.g. BLAST, a loose cluster of 32-bit x86 processors provides superior price performance. Similar configurations recommendations can be inferred for the other quadrants as well.

---

[*] Address correspondence to this author at the Hewlett-Packard, 2630 East 1st St., Duluth, MN 55812, USA; Tel: +1-218-525-1789;
E-mail: george.vacek@hp.com

1573-4099/08 $55.00+.00

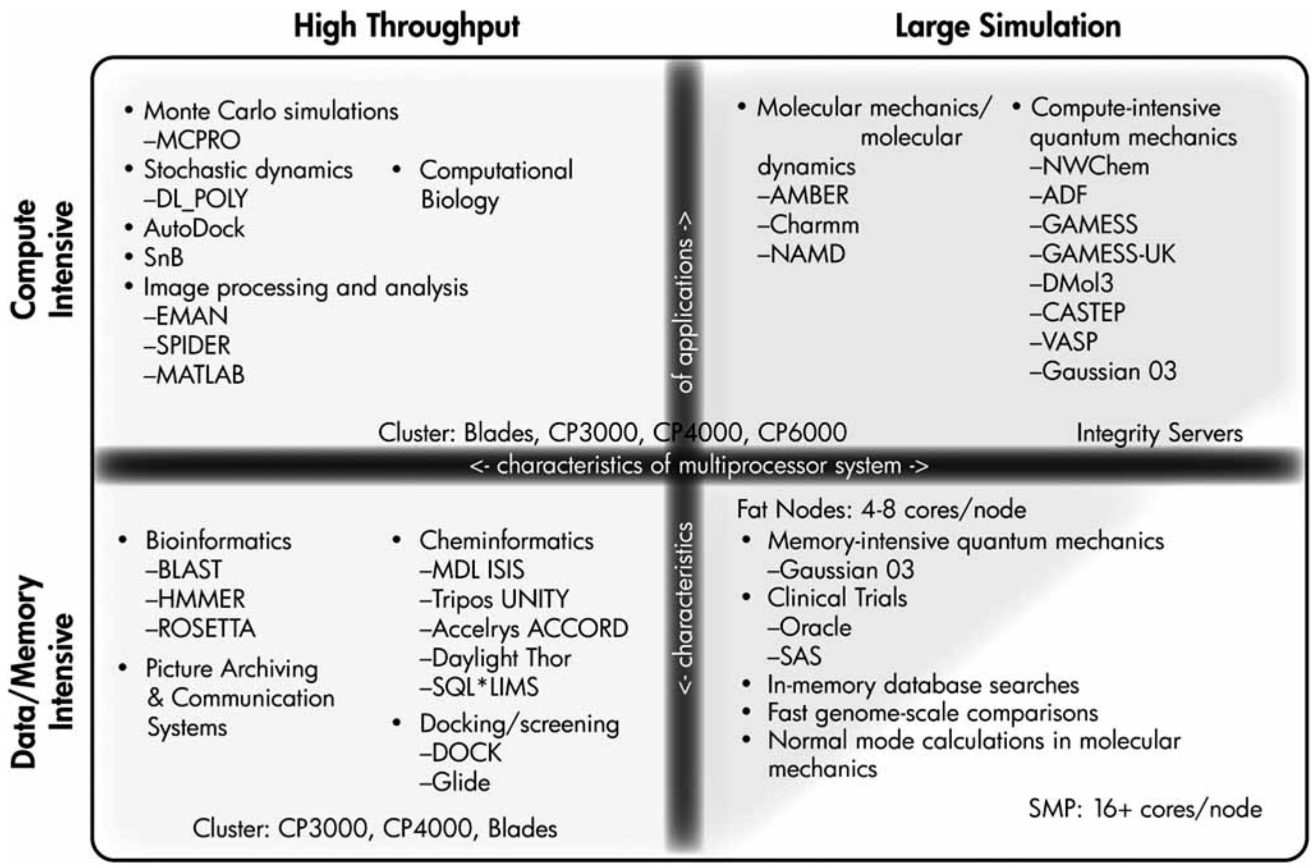


Fig. (1). Computational requirements of various Life and Materials Sciences applications. High throughput jobs are typified by many small tasks that need to be completed as quickly as possible. Large simulation jobs emphasize the completion of a single large, complex job as quickly as possible.

Over the past few decades, computational chemistry has advanced significantly in terms of the accuracy of methods that are routinely applied to molecular systems of interest (Fig. **2**). Complexity has increased to better address the question at hand - increased not only in terms of the level of theory used for single large-simulations, but also in terms of analysis of larger volumes of data, of new data types, and of data from disparate sources. Additional compute power can improve accuracy in other ways, too. For instance, virtual screening yields better results not only by using better hypothesis, but also by screening much larger virtual libraries.

TRENDS IN HPC

Industry Standard Components

Undoubtedly one of the biggest changes to HPC over the past decade has been the adoption of commercial off-the-shelf (COTS) systems as basic building blocks. From microprocessors to enclosures to interconnects, the majority of HPC systems today are composed of industry standard components. Adoption of COTS components in HPC systems began in the very late 1990's as industry-standard processors (e.g., Intel® Xeon® and AMD® Opteron™) increased dramatically in performance, rivaling the performance obtained from more specialized architectures for most applications. As late as 2001, the Top 500 list of supercomputers [1] shows systems based on Intel or AMD processors as 12 of the systems on the list, or 2.4%[1]. Within five years, that number had grown to over 75% [2] (Fig. **3**).

Demand for more specialized and expensive architectures has been dramatically reduced in all but the most niche applications. This trend is expected to continue as industry users reap the benefits from use of "mainstream" enterprise IT systems to solve HPC problems. These benefits include:

- Lower acquisition costs–volume production industry standard system components allows hardware vendors to amortize development costs over a much larger number of units, as well as lower per-unit production costs considerably.
- Increased application availability–software vendors are much more apt to support applications on platforms with a wide customer base, and a standardized instruction set/processor technology.
- Higher system reliability–a much wider installed base results in a more diverse testing base, as well as greater incentives from the (hardware or software) vendor to more extensively test their products. Generally, vendors design-in more robustness to service the more diverse customer base.

An important by-product of the trend towards commoditization is the type of vendor that participates in selling and

---

[1] These are only IA-32 processors, and excludes other RISC (e.g., HP PA-RISC, IBM Power) processors.

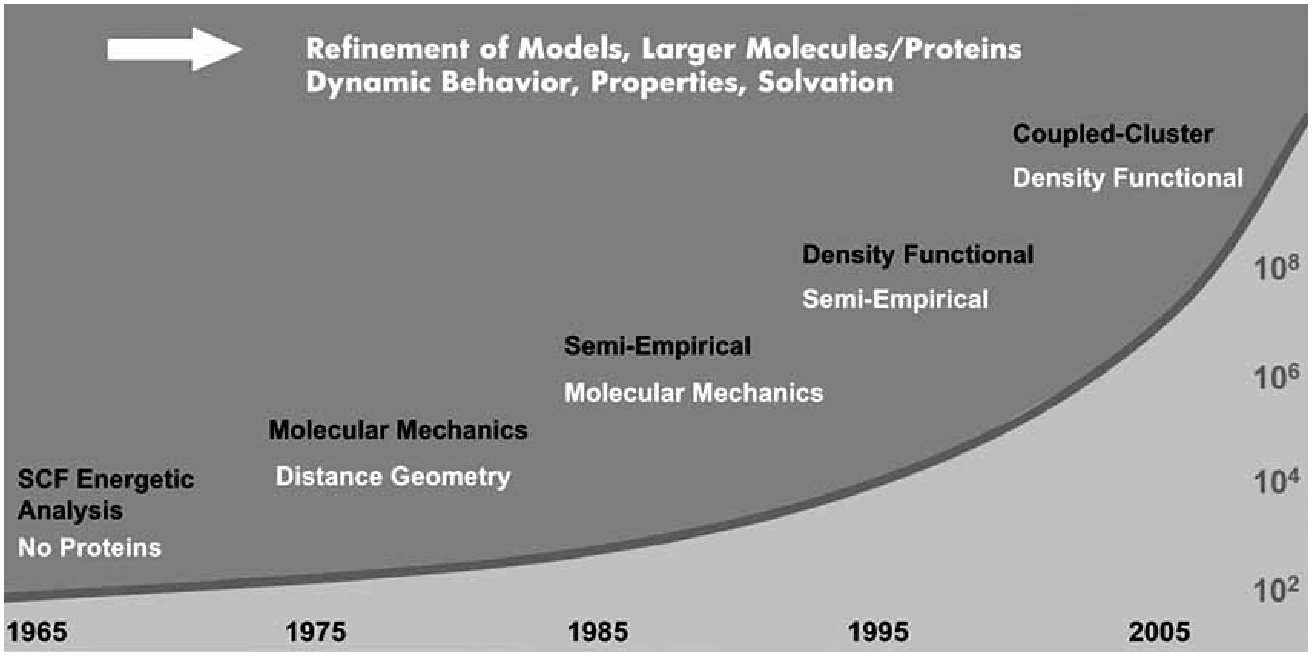


Fig. (2). Increasing refinement and complexity of models requires increased computing resources. Black font indicates state-of-the-art methods for modeling small molecules during each decade; white font indicates same for modeling proteins.

servicing HPC systems. As shown in Fig. **4**, smaller, HPC-specialized vendors are giving way to vendors that design and manufacture systems for commercial enterprises. This trend provides additional benefits to HPC users, including:

- Reduced risk–the widely used commercial products of these vendors are more reliable than those of the smaller vendors that provide specialized products.
- Better support–larger vendors have larger worldwide support staffs and are expected to be around to provide support for the lifetime of the product line.
- Stronger product pipeline–HPC systems benefit by leveraging larger R&D budgets, especially in the areas of system chipsets, interconnect, and system middleware such as job schedulers, management utilities, etc.

Although the curve shown in Fig. **4** may flatten somewhat, we expect this trend to continue due to the advantages cited above. Note that even within the group of enterprise vendors, the above advantages greatly favor the largest–HP and IBM–who together have two-thirds of the HPC market [3].

Linux Domination of the Operating Environment

Another dramatic change in HPC (and in computing in general) started in the 1990's with the adoption of the principles of *open source software*[2]. Open source is based on the premise that a community of developers, collaborating in self-interest on an open system, produces a better result than relatively fewer developers working on a proprietary system.

Linux is the quintessential example of open source software. Introduced in 1991, the operating system whose source code was freely available and modifiable by anyone quickly caught on, especially among academic researchers. The broader HPC community, searching for a portable operating system, followed–and forced somewhat reluctant hardware vendors along. Today Linux is the number one operating environment in HPC [3], and increasing adoption of Linux (especially in clusters) is expected to continue (Fig. **5**).

For HPC users, open source software in general, and Linux in particular, has multiple benefits, including:

- Lower cost of ownership–while open source software may not be free to acquire and support, there are ancillary cost reductions to deploying open source software. For example, users are no longer locked into a proprietary combination of hardware and operating environment, which lowers the price of both. Further, the larger population of developers reduces application development costs.
- Greater selection of applications–since Linux spans vendors, it is the largest operating environment; software vendors preferentially develop for Linux, increasing the size of the application portfolio available.
- Increased application and operating environment reliability–because the entire community has access to the source code, bugs are more quickly found and fixed.

The move to Linux has had a secondary, desirable, effect on systems vendors. Instead of using resources to develop proprietary operating systems, vendors now focus on value-add components such as management tools, clustering capabilities, and performance enhancements.

Traditional UNIX operating environments continue to be used and will be for quite some time (Fig. (**6**)) [4]. Some vendors of established applications are unwilling to (or lack

---

[2] Actually the term "open source" wasn't coined until the late 1990's to replace the somewhat ambiguous term "free software".

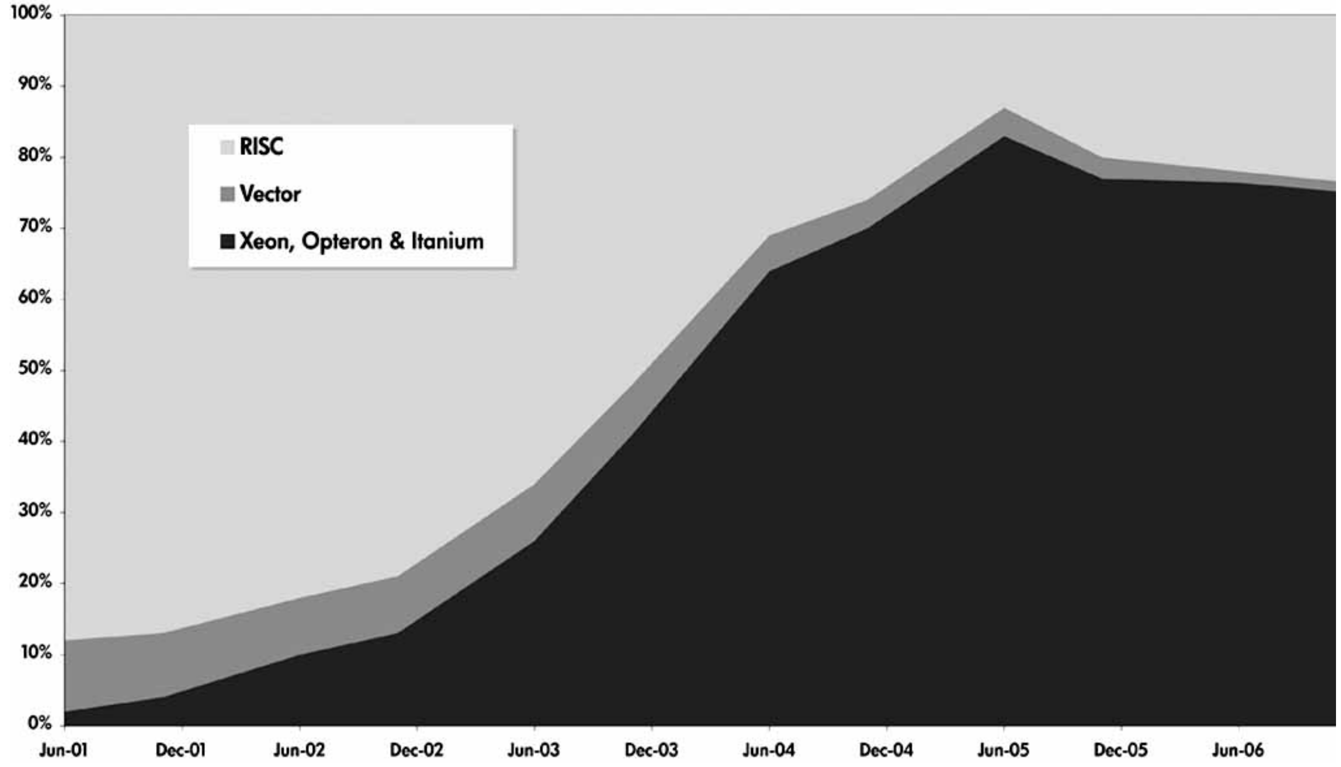


Fig. (3). Dramatic increase of industry-standard processors in the Top500 list of supercomputers over the past five years. Source: top500.org.

resources to) convert their applications to Linux. In some cases their customers are unwilling to switch environments. This is particularly true in clinical trials where governmental validation requirements, such as those from the FDA, create a significant barrier to switching due to time-consuming and expensive revalidation efforts.

An interesting development is the introduction of Microsoft's Windows® Compute Cluster Server (CCS) operating environment [5]. Windows has the advantage of being ubiquitous on the desktop, with users and IT technicians familiar with the environment. In the near term, it is unlikely that there will be a rush to port existing CADD software to CCS from, for instance, Linux. However, for application software currently written for Windows that would benefit from significantly more computational resources than a workstation provides, CCS will be the simplest path foreword.

Increasing Computing Densities

Mutli-Core Processors

In 1965, Gordon Moore (a co-founder of Intel) observed that transistor densities on an integrated circuit double about every two years. The oft-quoted Moore's law has, in fact, held true since 1965 (Fig. **7**), and shows little signs of being broken in the foreseeable future. The first impact of Moore's law on processors is the inclusion of more logic on a semiconductor die. Throughout the 1990's, single-chip microprocessors increased in both clock speed and transistor counts–resulting in geometrically increasing power (and heat dissipation) requirements. While manufactures have access to many different ways of managing heat–including reduced operating voltages, liquid cooling, and sophisticated on-chip logic to manage power–the practical limit in clock speeds has nearly been reached.

In an effort to continue to improve performance, processor designers have turned to placing multiple independent processing units (cores) on a single integrated circuit (processor). The resulting chips have only incrementally improved clock speed and are generally referred to as multi-core processors. Such chips may have a local or shared cache. Since the cores are on the same processor, they necessarily share the bus as well as memory and I/O controller, shown in Fig. **8**. Most new workstations and servers today are already based on multi-core processors (dual- or quad-core) at relatively low clock speeds.

The direct impact of multi-core platforms is that software developers must utilize parallelism to increase performance. The ability of multi-core processors to increase application performance depends on the efficient usage of multiple threads within large simulation applications or multiple processes for high throughput applications. Where CADD applications can take advantage of multiple threads or processes, bandwidth remains a consideration. The shared system bus could force a rethinking of some algorithms and implementations. That said, the current dual- and quad-core processors are relatively well suited to the majority of CADD applications today.

Another important issue with multi-core will be the impact on software licensing. This is especially important with chemistry applications, where, unlike biology, commercial packages often have no realistic open source equivalent. Unfavorable licensing terms can nullify the computational benefit of modern HPC architectures, if jobs must be proc-

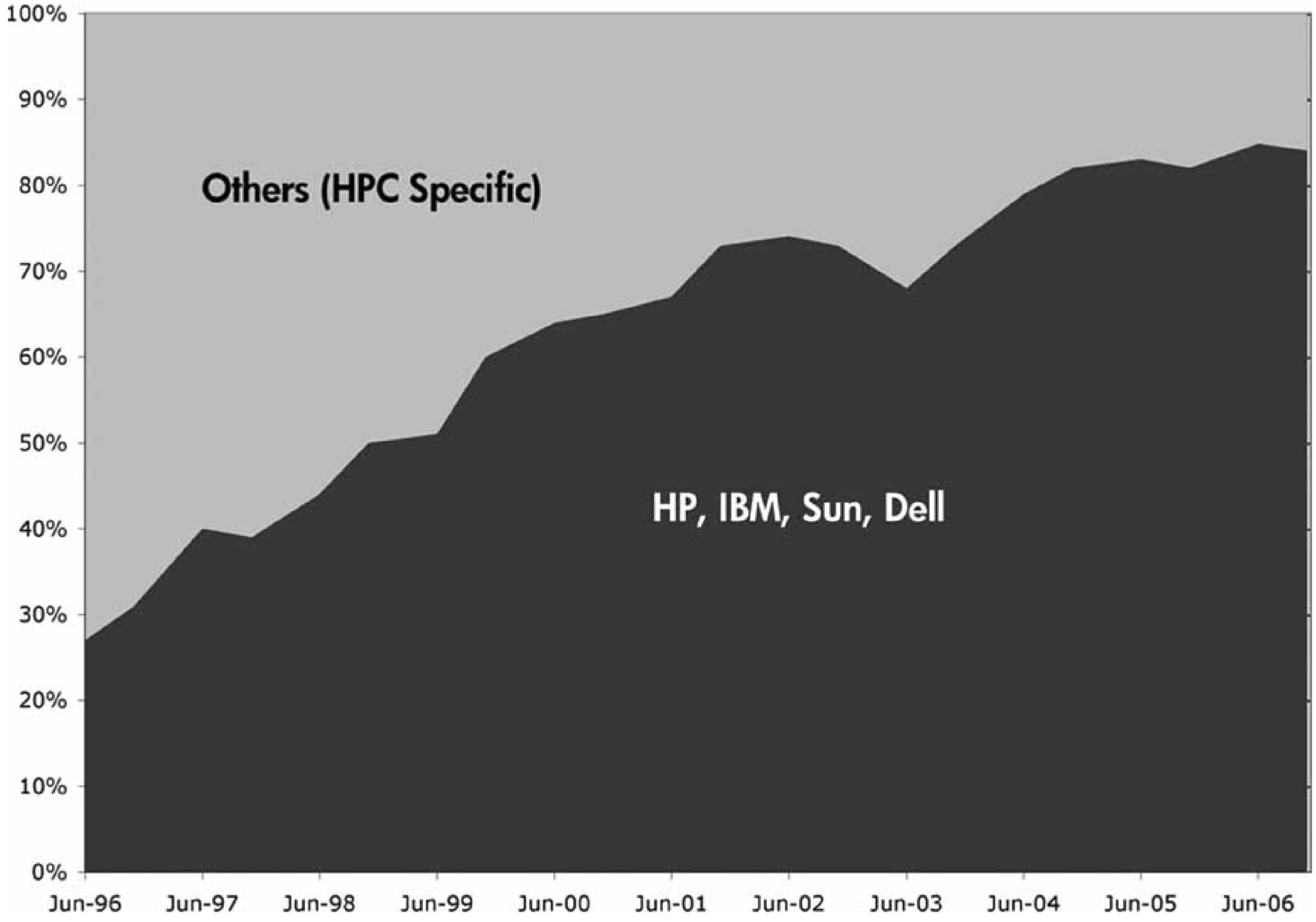


Fig. (4). Increasing dominance of the Top500 over the past 10 years by suppliers of enterprise IT systems as opposed to HPC specialists. Source: top500.org.

essed on a small number of cores due to lack of sufficient licenses, even when more cores exist and the application is capable of parallelizing across the additional cores. Although there are a variety of licensing models, much of the server and application software is still licensed on a "per processor" basis. While some vendors have switched to "per core" the majority seem to consider multi core chips as a single processor. If an application can get the same performance out of half as many dual-core processors, this will effectively cut software costs in half (on a per job basis)–just as previous doublings of chip (clock) speeds cut per use software costs in half. Not all software vendors agree with this philosophy, and it could remain a thorny issue. Even per processor licenses can be restrictive for large clusters or Grid Computing (see below).

Massively Multi-Core (MMC)

Massively multi-core processors (MMC) with hundreds of cores per processor already exist for such uses as network processing and digital signal processing. Furthering the trend of dual- and quad-core, MMC will come to general purpose processing, too.

Multi-core processor chips are shared-memory systems. As the number of cores grows, systems with relatively few MMCs could replace current large symmetric multiprocessor (SMP) systems. Algorithms that need a globally shared memory programming model could fit completely on a single processor, using multiple cores accessing the common memory. However, the MMCs will share a common chip-level cache. When multiple processes run on a chip, cache sharing means that less cache is available for each process. Applications that are sensitive to cache size, such as GROMACS, will run very well as a single process on each chip, but will not scale well across all of the cores.

Originally, implementing parallelism focused on SMP methods that utilize additional processors within a node; the work utilized methods such as pthreads and OpenMP. The tools are relatively simple to implement and allow the program to share data. Parallelism may be fine grain (over individual loops) or coarse grain (over subroutines). For the last decade, most developer focus has been on the use of multiple computers, using tools such as MPI and PVM. These tools are more difficult to implement, often requiring an overhaul of the application. No data is shared, and efficient implementations require coarse grain parallelism with limited communication between processes. MMCs should cause resurgence in the use of SMP methods. Larger numbers of core on a single node will alleviate some of the need for multiple nodes. Easy implementation combined with features that make SMP faster on a single node, such as fine grain paral-

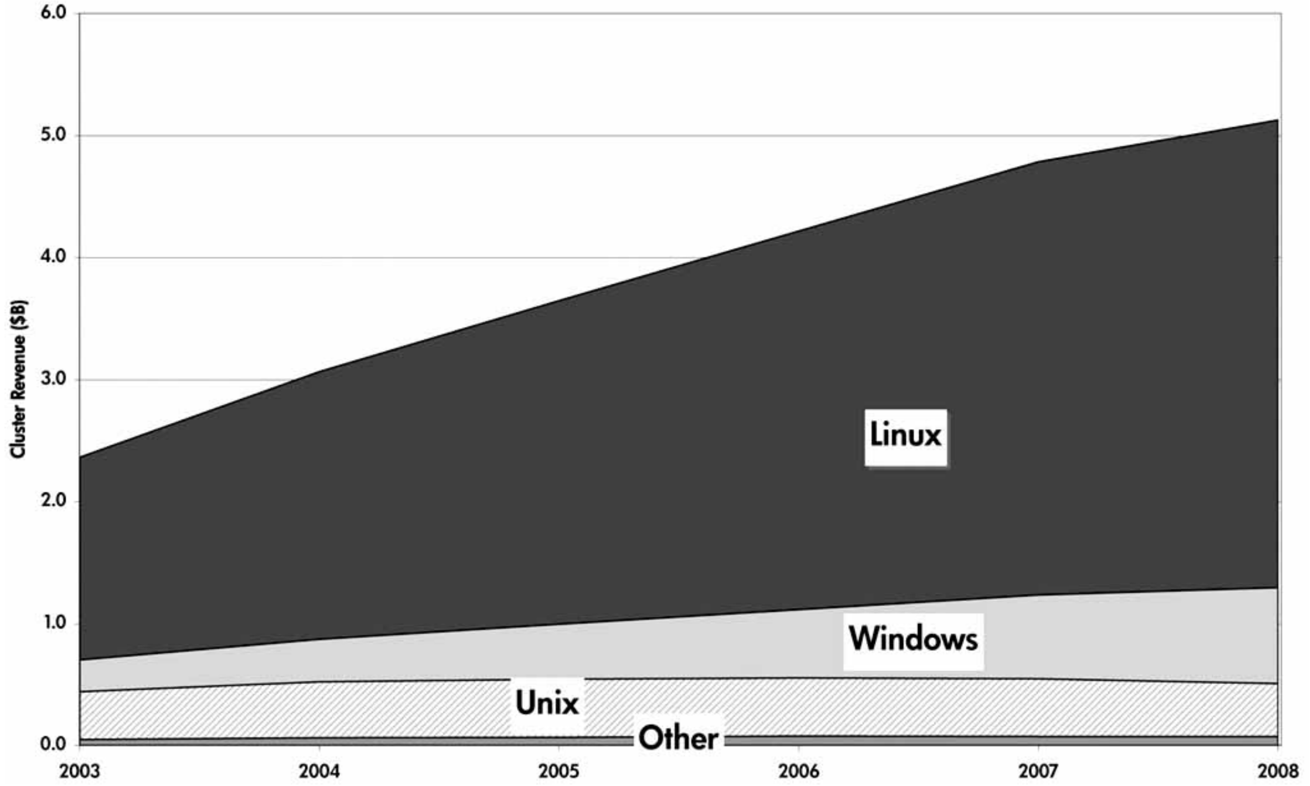


Fig. (5). Linux dominates HPC clustering. Source: IDC Cluster Study 2004.

lelism and shared data should renew interest in these methods.

Regardless of the programming model (distributed *vs* shared memory), application developers will be forced to turn to parallelism to increase performance. The model used to take advantage of parallel performance will depend on the size of jobs (see again Fig. **1**). When an application deals with many jobs that each take a small amount of time, the

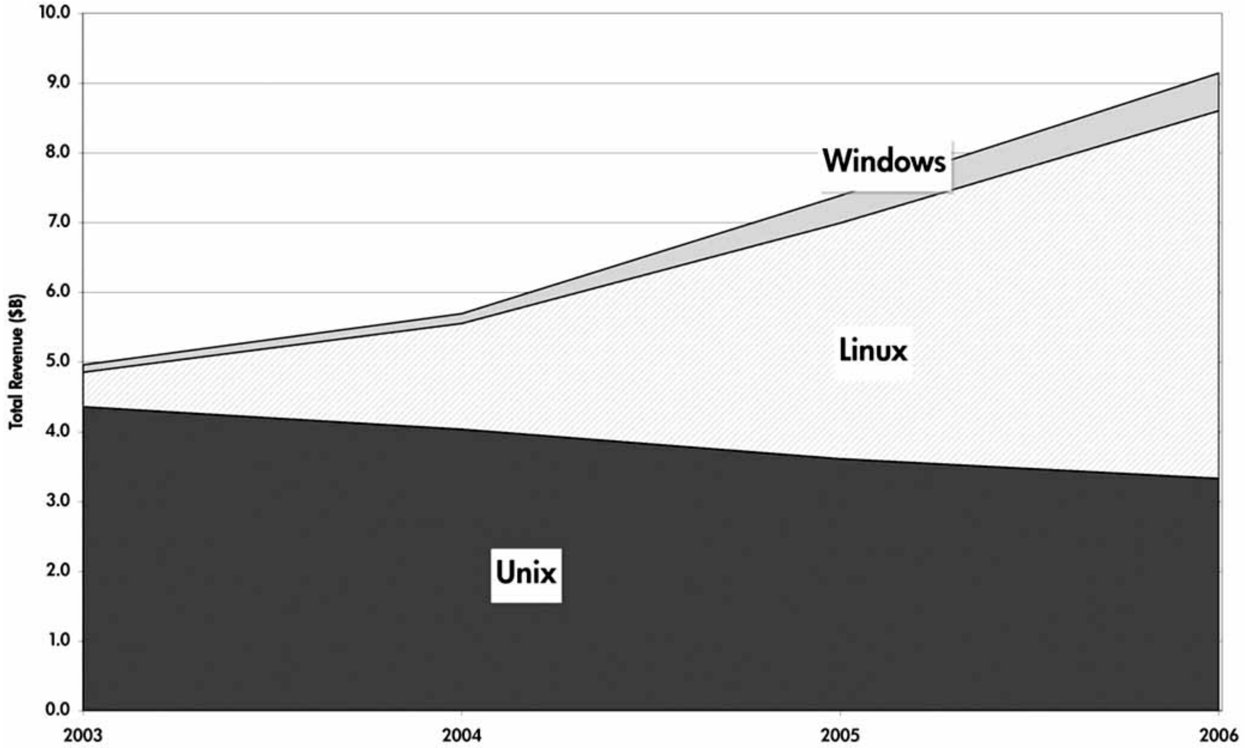


Fig. (6). UNIX will continue to be a viable operating system, especially in non-clustered environments [4]. Source: IDC Cluster Study 2004.

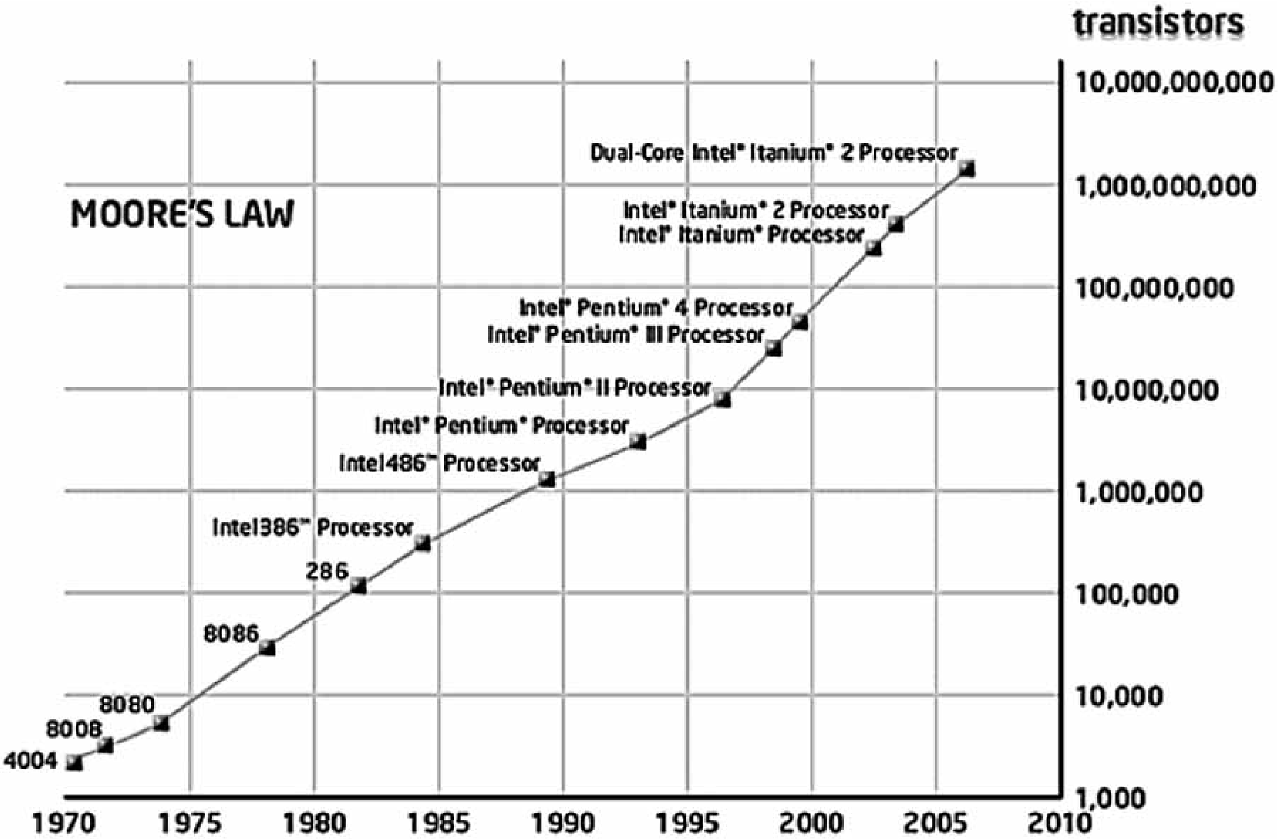


Fig. (7). Moore's Law of increasing integrated circuit density [6]. (Courtesy Intel).

most efficient form of parallelism it to distribute the individual jobs over many cores. mpiBLAST is a good example of an application that takes advantage of this type of parallelism. When a single run can be expected to take a significant amount of time, parallelizing to speed up single jobs is preferred. Many CADD applications like NWChem use this type of parallelism, but most of them do not scale particularly well beyond a dozen or so threads of execution. HP has announced a Multi-Core Optimization Program to work with chip manufacturers and software vendors to research next-generation multi-core optimization technologies [7]. The other major HPC systems vendors are expected to make similar initiatives as well.

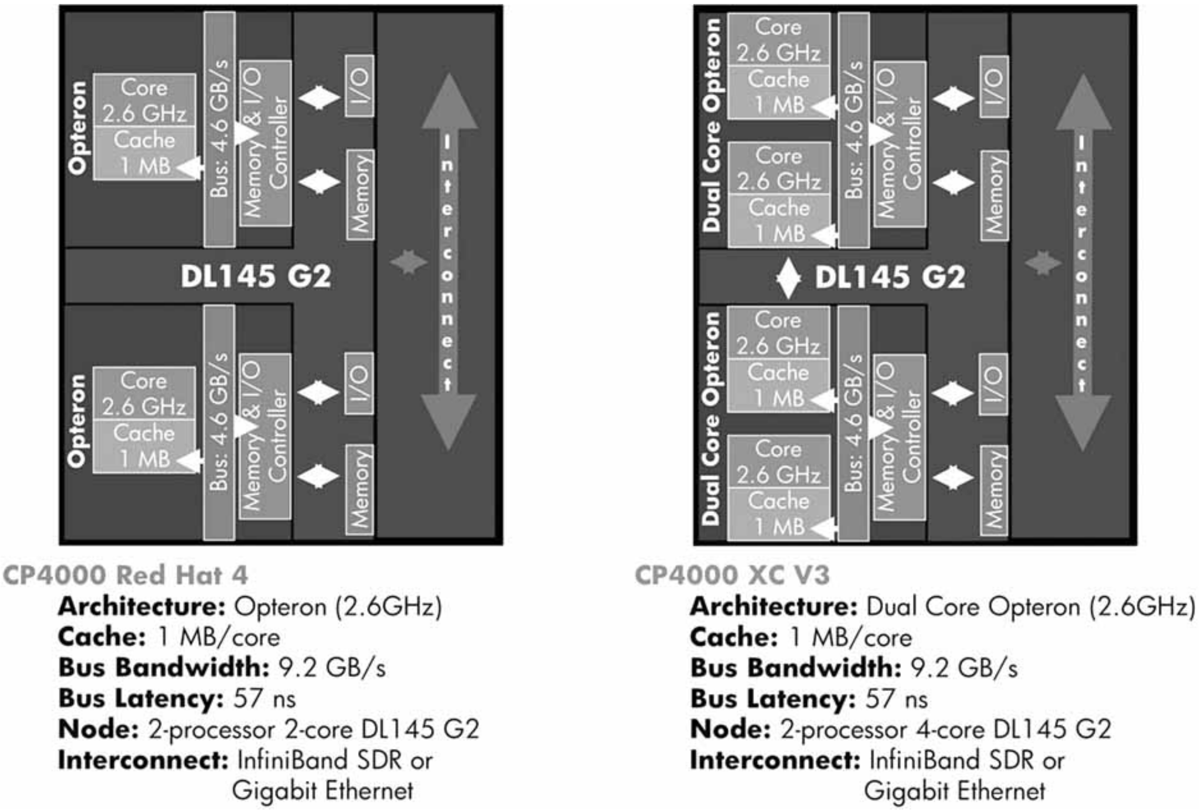
Packaging Densities

Moore's law is an indication of the trend of placing more logic in less space. This applies not only to processors, but also to the rest of the electronic components in a computer system. The result is greater overall computing densities, and corresponding requirements to install, cool, and service such dense computing systems. An important trend, then, is finding innovative ways to manage large, dense installations of computing equipment. Manufacturers will need to develop technologies that reduce overall heat, as well as find ways to automatically and dynamically manage facilities.

A recent trend in HPC is the rapidly increasing use of blade systems. Blades have advantages over traditional racked clusters, including:

- The packaging saves on floor space and minimizes cabling and racking. Because individual blades plug into a backplane, it is much easier to add to the system or replace nodes.
- Facilities costs are lower because much of the infrastructure such as power supplies, cooling, and maintenance hardware can be shared among the blade nodes.
- Modularity, combined with this relatively new idea of virtualization, gives incredible flexibility for change management. Virtualization provides a pool of resources–including compute, storage, and visualization–that can be allocated as needed. The infrastructure is wired once and then configuration changes can be made on the fly without re-wiring, including virtual movement of IP addresses and node names.
- Reduced installation and maintenance costs. Blade systems integrate the interconnect into a midplane board, all but eliminating cabling, at least within each rack. The interconnect switches become just another component in the enclosure–easy to access and monitor, with the optimal amount of cooling, and close to the host nodes. Not only is this kind of setup much cleaner, but it also adds tremendously to the overall stability of the system.

Blades also help solve the time-honored problem associated with managing all of this hardware and software. Typically in clusters, the management software automates processes associated with distributing software, managing users, and controlling the hardware environment. The blade systems being designed today have a host of built-in capabilities



CP4000 Red Hat 4
Architecture: Opteron (2.6GHz)
Cache: 1 MB/core
Bus Bandwidth: 9.2 GB/s
Bus Latency: 57 ns
Node: 2-processor 2-core DL145 G2
Interconnect: InfiniBand SDR or Gigabit Ethernet

CP4000 XC V3
Architecture: Dual Core Opteron (2.6GHz)
Cache: 1 MB/core
Bus Bandwidth: 9.2 GB/s
Bus Latency: 57 ns
Node: 2-processor 4-core DL145 G2
Interconnect: InfiniBand SDR or Gigabit Ethernet

Fig. (8). Sample two processor system architectures using single- and dual-core processors.

that go beyond rudimentary management functions, reducing the manager-to-node count ratio tremendously.

The advantages of blades systems, at no real price premium per node, mean they are being considered in most new HPC purchases [8], particularly in areas like bioinformatics and virtual screening where clusters can be quite large. The Welcome Trust Sanger Institute (Sanger) is in the process of taking their blades farm up to approximately 2500 cores [9].

Access to Data

HPC is not just about computational power–it's also about data. For instance, witness the flood of instrument data that must be analyzed in bioinformatics, and the size of the comparative databases used. At the Welcome Trust Sanger Institute, the trace database alone is 65TB in size [9]. This situation will only get worse with the next generation of sequencing, where sequencers can produce 1 TB of primary image data over a 3 day run [10]. Consider too that these will appear not only in large Sanger-like centers, but also in many smaller labs that will not have IT infrastructure or staffing.

Such data needs to be analyzed, stored, transmitted, visualized, and archived. For the most part, technology has kept pace with volume requirements; that is, disk drives and disk subsystems have increased in capacity as fast as processor performance [11]. However, bandwidth has not kept up. For example, "sneakernet" (physically moving a tape or disk from one location to another) remains a common way to move large amounts of data from where it is generated to where it is analyzed. Clearly one option is not to store all the primary data in a central facility, but to move the computing into the lab–attaching a small cluster to each sequencer.

In clustered computing environments there is still a tendency to overlook storage management as a critical component of overall system performance. For example, many clusters simply use Network Attached Storage (NAS) over Ethernet–in such situations access to disk (bandwidth) can be a substantial bottleneck. To address data bandwidth and availability issues, several vendors have developed scalable filesystems, generally distributed parallel filesystems that stripe and replicate data over multiple storage servers. These systems provide HPC-class storage - tens of thousands of nodes and petabytes of storage - by leveraging parallelization and tiered storage [12]. A recent study showed that inadequate storage lowers cluster throughput, particularly for data intensive applications such as NWChem [13]. The study concluded that object-based clustered architectures of parallel file systems: 1) dramatically increase throughput over internal disks; 2) have comparable throughput to external disk arrays, but are cheaper, more reliable and easier to manage.

Data is also critical to visualization. Visualization techniques such as real-time flythroughs and interactive modeling require huge amounts of bandwidth. Further, global collaboration often requires visualization anywhere in the world, extending bandwidth requirements to the global network.

Grid Computing

The term grid computing means many things to many people, from simple cluster management software, to multi-site, multi-platform distributed computing environments. A reasonable definition of grid computing is

> "Grid Computing involves connecting storage and data as well as CPUs from multiple systems into a centrally managed but flexible computing environment" [14].

HPC organizations are using grid computing (and its cousin, "utility computing") to improve flexibility by leveraging large populations of resources on a "pay as you go" basis. For example, Vijay Pande's folding@home project [15] to understand protein folding does not rely on supercomputers for computing; instead, the primary contributors to the folding@home project are many hundreds of thousands of personal computer users who have installed a small client program. The client runs in the background, utilizing otherwise unused CPU power, or run as a screensaver only while the user is away. Effective use of this computing model is heavily dependent on the cooperation of software vendors. Software licensing schemes vary dramatically ranging from licenses limiting usage to: individual computers, a certain number of processors, a certain number of core, or an entire site. Grid computing requires affordable licensing on a very large number of processors.

Often compared to the model used by public energy utilities, this "IT as a service" model is relatively new, and can be compared to the evolution of energy utilities over the past century. In the early twentieth century, large businesses had to build and operate their own power plants to generate the electricity required to run their businesses and compete in the marketplace. Over time, these businesses realized that it was smarter to offload the burden of managing large internal infrastructures to utility service providers, who could provide electricity when and where it was needed. The same trend is occurring in IT today. HPC directors are realizing the value of a hybrid computing environment–owning a portion of fixed assets for critical jobs and augmenting these resources with utility computing for peak demand or as a long-term, flexible resource (Fig. **9**).

A longer-term trend is to continue to bring grid computing into an enterprise. The experience of Johnson and Johnson in rolling out an enterprise wide grid system shows that the technological barriers can be overcome, however, the biggest challenge remains political and social [16]. In the end they have replaced aging systems, consolidate resources, and were able to reduce costs, while completing a grid that spans multiple lines of business worldwide and a variety of system types, including clusters, servers, and desktops [17].

ON THE HORIZON–HARDWARE ACCELERATORS

One aspect of HPC is that the industry tends to be an incubator for new technology; once the technology becomes established among the early adopting HPC community, it trickles down to become more mainstream in enterprise computing, and eventually more pervasive in the wider computing industry (Fig. **10**). Looking down the road, several hardware accelerator technologies hold promise for HPC and, eventually, as a mainstream component of an enterprise IT infrastructure.

With the continuing increase in semiconductor densities, systems designers are finding innovative ways of using additional logic. One of the ways being explored today is that of employing application-specific accelerators in the system.

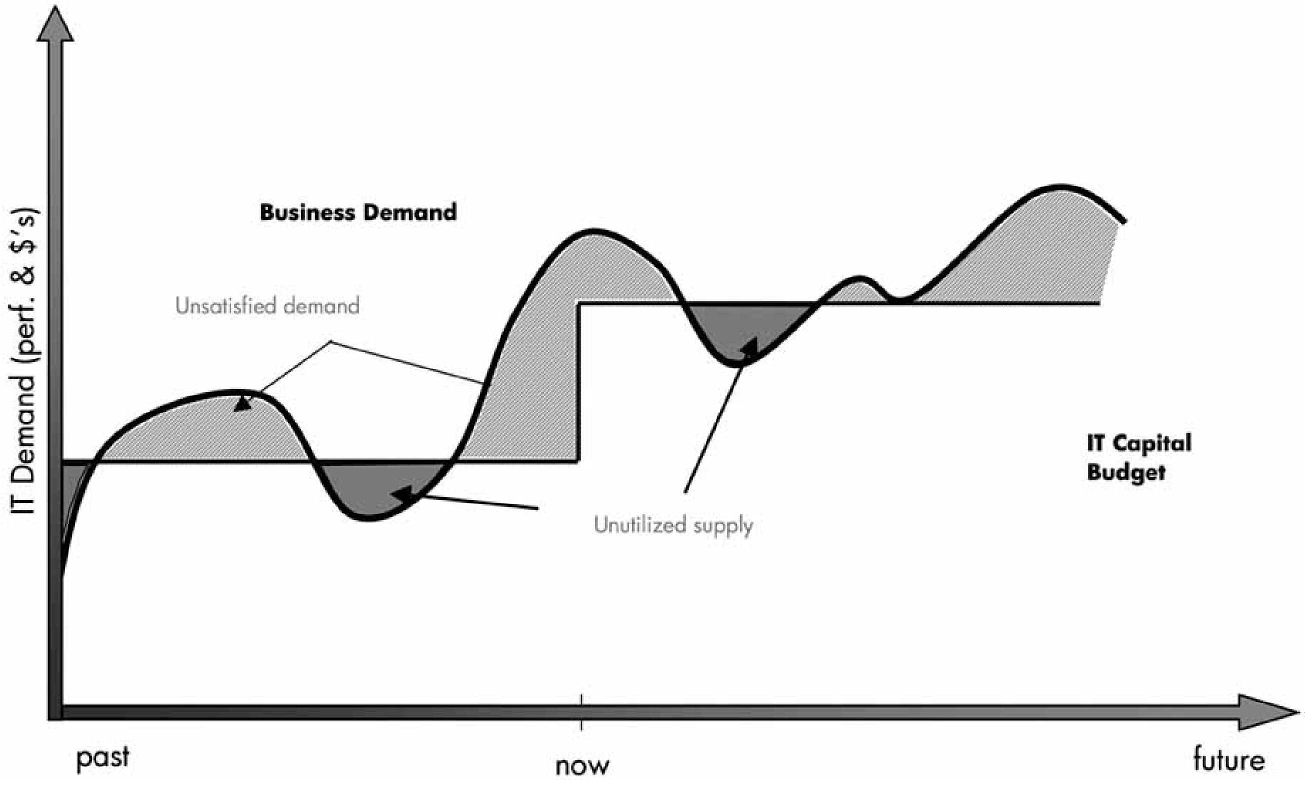


Fig. (9). A decision must be made to buy capacity for average usage or peak. Owning fixed assets to meet sustained demand and taking advantage of utility computing services to meet peak demand is a reasonable solution.

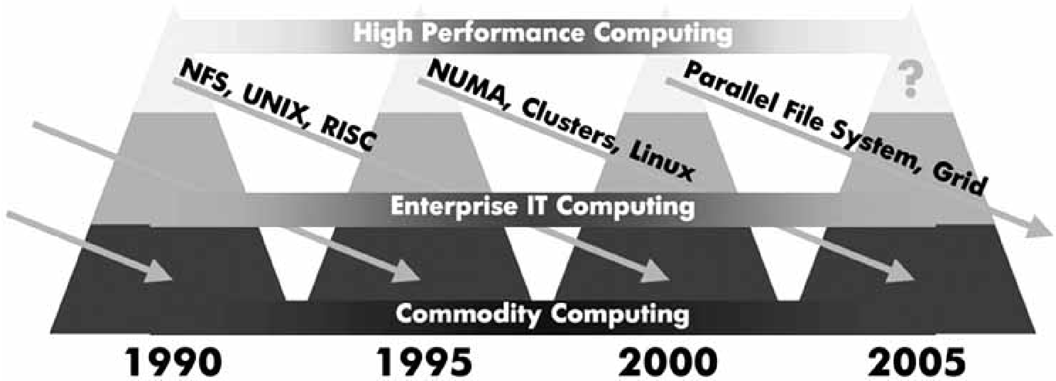


Fig. (10). HPC tends to introduce new technology into "commodity" computing [18].

Such accelerators, embedded in processor logic, on I/O boards, or embedded in the memory subsystem, can be programmed to perform very specific functions at extremely high speeds. For example, a matrix multiply function could be built into part of the memory subsystem, freeing the processor from the computation and putting it closer to physical memory.

An application-specific integrated circuit (ASIC) is an integrated circuit customized for a particular use, rather than intended for general-purpose use. Two examples of ASICs in CADD are IBM's 8-year old BlueGene project and D.E. Shaw Research's molecular dynamics chip, Anton. Anton has been designed but has yet to be manufactured and is expected to execute molecular dynamics programs about 100 times faster than current COTS processors. However, due to economies of scale, it is difficult for systems based on ASICs to surpass industry-standard systems on a price/performance basis. As the reason to design BlueGene and Anton, both projects cited they would enable biomolecular simulations that are orders of magnitude larger than current technology permits. This is a noble goal, important to the CADD community and science in general, but even Shaw acknowledges that it is unlikely to yield economic success [19].

Another hardware accelerator technology is that of field-programmable gate arrays (FPGAs). An FPGA is a semiconductor device containing programmable logic components called "logic blocks", and programmable interconnects. FPGAs are usually slower than their ASIC counterparts, as they cannot handle as complex a design, and draw more power. But their advantages include a shorter time to market, ability to re-program in the field to fix bugs, and lower non-recurring engineering costs. FPGAs are starting to be used in HPC in cases where computational kernels such as FFT or Convolution are performed on the FPGA instead of a microprocessor. The x620 accelerator from Clear Speed, for instance, has been shown to accelerate some Generalized Born methods within AMBER calculations between 3 and 10 fold [20].

Today graphics processing units (GPUs), which are capable of extremely high floating-point calculation rates, are also being considered for offloading CPUs to increase performance of specific algorithms. ATI has teamed with Stanford University to create a GPU-based client for Vijay Pande's folding@Home distributed computing project [16] that claims in certain circumstances to yields results significantly faster than the conventional CPU's. For price performance, GPUs enjoy the economies of scale similar to other COTS CPUs; however, GPUs are designed specifically for graphics and thus are very restrictive in terms of operations and programming. For instance, the implementation of floating point on GPUs is generally not IEEE compliant, is only single precision, and does not match across vendors, making them inappropriate for the accuracy/correctness required for most scientific computing. Thus, the specialized programming environment is likely to keep most commercial applications from being ported anytime soon. Some effort is being made to "correct" the problems of GPUs, to add more control and precision, making them more generally programmable (GPGPUs). That will require a delicate balancing act. In trying to meet the needs of both graphics and scientific computing, they may create something well suited for neither.

CONCLUSIONS

The pace of technology advancements in computing continues at an increasing rate. HPC systems designers have always been, and continue to be, on the forefront of employing these technologies to give scientists and researchers the maximum computing capacity for their money. As HPC technologies mature, they trickle down to more commercial uses, and many eventually become commonplace in computing. At the same time, HPC users, including CADD researchers, tend to stress whatever equipment they have available, constantly increasing problem sizes and refining models to fit the computing power available. This trend–consuming whatever available computing power is available–will continue as long as there is science, and fuels HPC as the proving ground for new technologies. Finally, the benefits of HPC across all areas of an enterprise will continue to grow. HPC is already embedded in most industries–as technology provides more comprehensive functionality, provides every greater problem-solving power, and is easier to access, it will be a crucial component of every enterprise's business infrastructure.

REFERENCES

[1] http://top500.org.
[2] http://top500.org/lists/2006/11.
[3] "IDC Worldwide High-Performance and Technical Computing Server 2007–2011 Forecast", April **2007**.
[4] IDC Cluster Study **2004**.
[5] http://www.microsoft.com/windowsserver2003/ccs/.
[6] http://www.intel.com/technology/mooreslaw.

[7] http://www.hp.com/hpinfo/newsroom/press/2007/070627a.html

[8] IDC Special Study, “The Cluster Revolution in Technical Computing Markets”, May **2006**, Doc. #06C4775.

[9] “Enabling Advanced Genomic Research through Information Technology”, Phil Butcher, HP Health and Life Sciences Symposium, Coranado, California, May **2007**. http://www.hp.com/aquarius/HP_Health&Life_Science_SymposiumIII/S07_Management_Perspectives_Butcher.pdf.

[10] “Of Sequences and Sneakernets”, Chris Dwan, Bio-IT World, June **2007**, p14.

[11] “Kryder’s Law”, Scientific American August, **2005**.

[12] http://www.clusterfs.com.

[13] CAE & Comp Chem Application Performance I: Object-Based, InfiniBand-Connected Storage” www.cse.clrc.ac.uk/disco/publications/M Deegan Tech Report.pdf.

[14] “IDC Worldwide Grid Computing 2006–2010 Forecast: Reality Check”, November **2006**.

[15] http://folding.stanford.edu/.

[16] “Grid Computing Adoption in the healthcare market”, the 451 Group, Grid Adoption Research Service Report 15, June **2007**.

[17] http://h30294.www3.hp.com/hp/changeartists/site/assets/pdf/Grid_Computing.pdf? hit=1798&download=true.

[18] “HPC in Manufacturing: Trends, Paradoxes and Solutions”, Knute Christensen, Applied Biosystems Inform, San Francisco CA, September **2006**.

[19] Algorithms and architectures for millisecond-scale molecular dynamics simulations of proteins, David Shaw, Schrodinger User Symposium, New York, NY, April **2007**.

[20] “Special Chips for Specific Purposes: A look at acceleration technologies for scientific computing”, John Gustafson, Scientific Computing, June **2007**, p.45.

Received: July 18, 2007 **Revised: September 7, 2007** **Accepted: September 10, 2007**