High Performance Computing Today

Jack Dongarra
Computer Science Department
University of Tennessee

Hans Meuer
University of Mannheim

Horst Simon
NERSC
Lawrence Berkeley National Laboratory

Erich Strohmaier
University of Tennessee

In last 50 years, the field of scientific computing has seen a rapid change of vendors, architectures, technologies and the usage of systems. Despite all these changes the evolution of performance on a large scale however seems to be a very steady and continuous process. Moore's Law is often cited in this context. If we plot the peak performance of various computers of the last 5 decades in Figure 1 that could have been called the `supercomputers' of there time we indeed see how well this law holds for almost the complete lifespan of modern computing. On average we see an increase in performance of two magnitudes of order every decade.
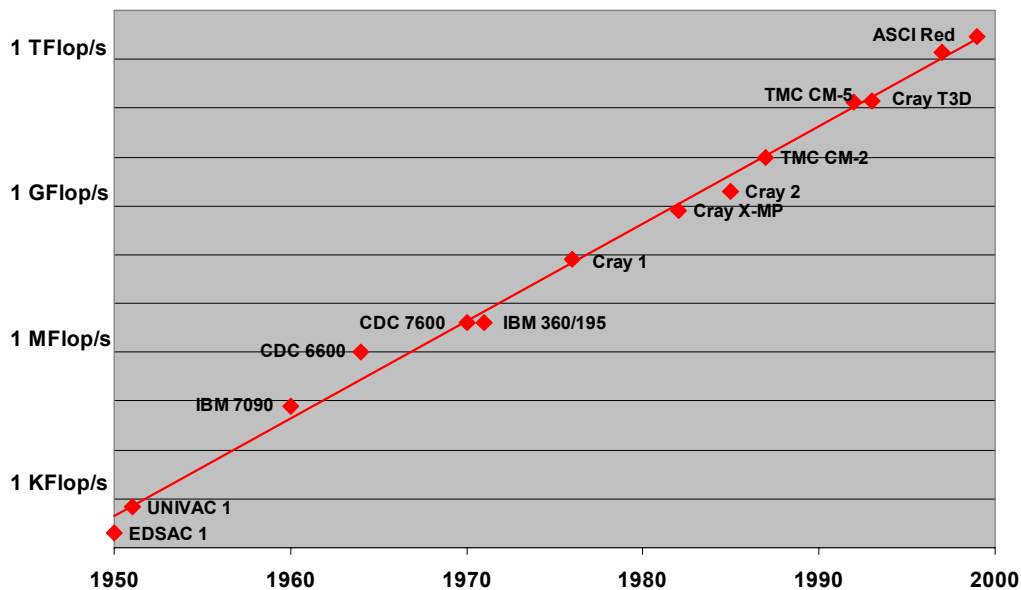

Figure 1. Moore's Law and Peak Performance of Various Computers Over Time

In the second half of the seventies the introduction of vector computer systems marked the beginning of modern Supercomputing. These systems offered a performance advantage of at least one order of magnitude over conventional systems of that time. Raw performance was the main if not the only selling argument. In the first half of the eighties the integration of vector system in conventional computing environments became more important. Only the manufacturers that provided standard programming environments, operating systems and key applications were successful in getting industrial customers and survived. Performance was mainly increased by improved chip technologies and by producing shared memory multi processor systems.

Fostered by several Government programs massive parallel computing with scalable systems using distributed memory got in the focus of interest end of the eighties. Overcoming the hardware scalability limitations of shared memory systems was the main goal. The increase of performance of standard microprocessors after the RISC revolution together with the cost advantage of large-scale productions formed the basis for the ``Attack of the Killer Micro''. The transition from ECL to CMOS chip technology and the usage of ``off the shelf'' microprocessor instead of custom designed processors for MPPs was the consequence.

The acceptance of MPP system not only for engineering applications but also for new commercial applications especially for database applications emphasized different criteria for market success such as stability of system, continuity of the manufacturer and price/performance. Success in commercial environments is now a new important requirement for a successful Supercomputer business. Due to these factors and the consolidation in the number of vendors in the market hierarchical systems build with components designed for the broader commercial market are currently replacing homogeneous systems at the very high end of performance. Clusters build with components of the shelf also gain more and more attention.

Beginning of the nineties while the MP vector systems reached their widest distribution, a new generation of MPP system came on the market with the claim to be able to substitute of even surpass the vector MPs. To provide a better basis for statistics on high-performance computers, The Top500[1] list was begun.
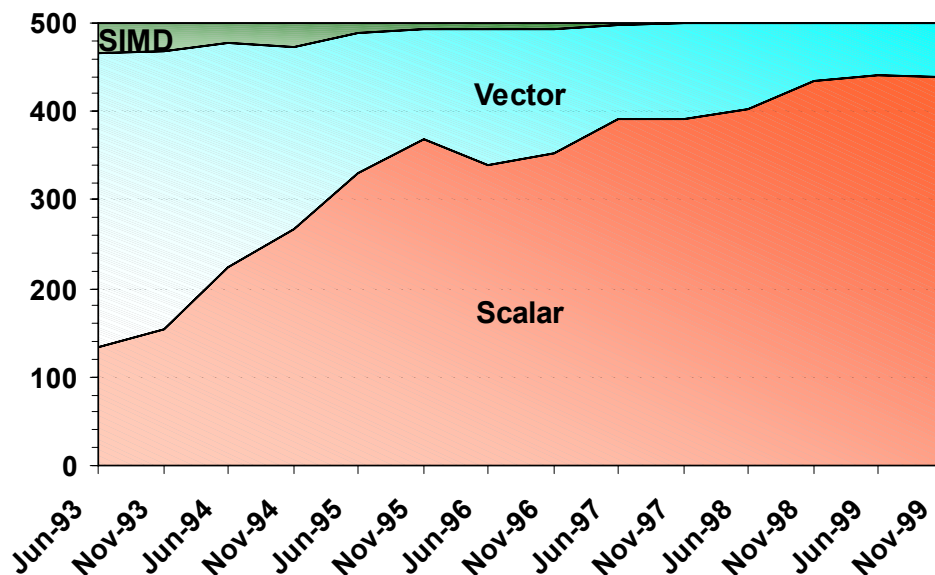
Figure 2 Process Design Used as Seen in the Top500

This report lists the sites that have the 500 most powerful computer systems installed. The best Linpack benchmark performance [2] achieved is used as a performance measure in ranking the computers. The TOP500 list has been updated twice a year since June 1993. In the first Top500 list in June 1993 there were already 156 MPP and SIMD systems present (31% of the total 500 systems).

The year 1995 saw some remarkable changes in the distribution of the systems in the Top500 for the different types of customer (academic sites, research labs, industrial/commercial users, vendor installations, and confidential sites) Until June 1995, the major trend seen in the Top500 data was a steady decrease of industrial customers, matched by an increase in the number of government-funded research sites. This trend reflects the influence of the different governmental HPC programs that enabled research sites to buy parallel systems, especially systems with distributed memory. Industry was understandably reluctant to follow this step, since systems with distributed memory have often been far from mature or stable. Hence, industrial customers stayed with their older vector systems, which gradually dropped off the Top500 list because of low performance.

Beginning in 1994, however, companies such as SGI, Digital, and Sun started to sell symmetrical multiprocessor (SMP) models of their major workstation families. From the very beginning, these systems were popular with industrial customers because of the maturity of these architectures and their superior price/performance ratio. At the same time, IBM SP2 systems started to appear at a reasonable number of industrial sites. While the SP initially was sold for numerically intensive applications, the system began selling successfully to a larger market, including database applications, in the second half of 1995.
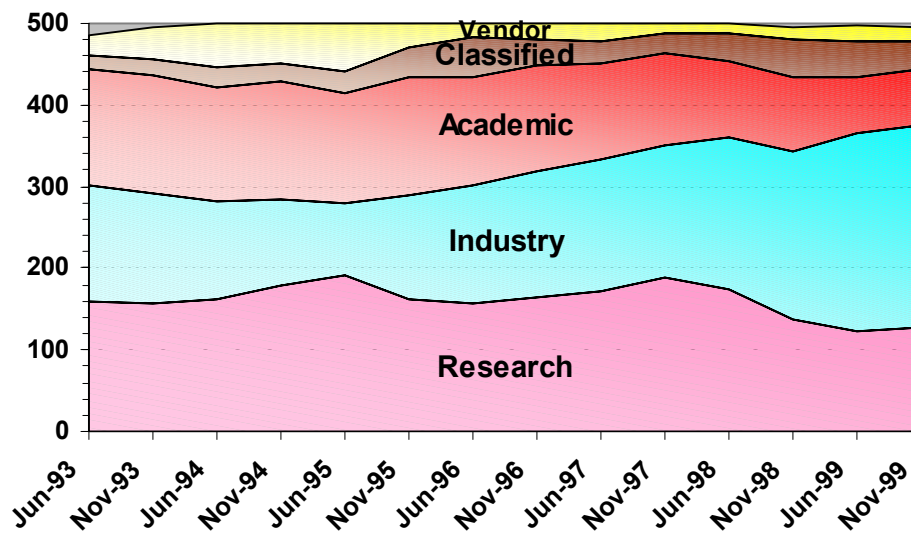
Figure 3: The Number of Systems on the Different Types of Customers Over Time

Subsequently, the number of industrial customers listed in the Top500 increased from 85, or 17%, in June 1995 to about 241, or 48.2%, in June 1999. This appears to be trend because of the following reasons.

- The architectures installed at industrial sites changed from vector systems to a substantial number of MPP systems. This change reflects the fact that parallel systems are ready for commercial use and environments.

- The most successful companies (Sun, IBM and SGI) are selling well to industrial customers. Their success is built on the fact that they are using standard workstation technologies for their MPP nodes. This approach provides a smooth migration path for applications from workstations up to parallel machines.

- The maturity of these advanced systems and the availability of key applications for them make the systems appealing to commercial customers. Especially important are database applications, since these can use highly parallel systems with more than 128 processors.

While many aspects of the HPC market change quite dynamically over time, the evolution of performance seems to follow quite well some empirical laws such as Moore's law mentioned at the beginning of this chapter. The Top500 provides an ideal data basis to verify an observation like this. Looking at the computing power of the individual machines present in the Top500 and the evolution of the total installed performance, we plot the performance of the systems at positions 1, 10, 100 and 500 in the list as well as the total accumulated performance. In Figure 4 the curve of position 500 shows on the average an increase of a factor of two within one year. All other curves show a growth rate of 1.8 +- 0.07 per year.
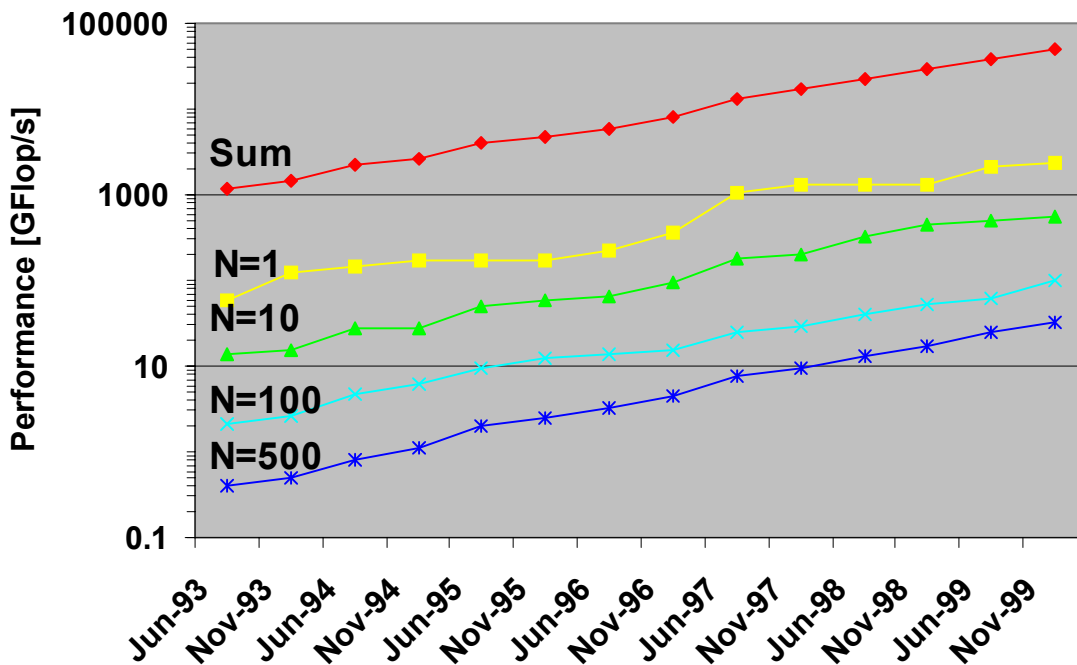
4

Figure 4: Overall growth of accumulated and individual performance as seen in the Top500

To compare these growth rates with Moore's Law we now separate the influence from the increasing processor performance and from the increasing number of processor per system on the total accumulated performance. To get meaningful numbers we exclude the SIMD systems for this analysis, as they tend to have extreme high processor numbers and extreme low processor performance. In Figure 4 we plot the relative growth of the total processor number and of the average processor performance defined as the quotient of total accumulated performance by the total processor number. We find that these two factors contribute almost equally to the annual total performance growth factor of 1.82. The processor number grows per year on the average by a factor of 1.30 and the processor performance by 1.40 compared 1.58 of Moore's Law.
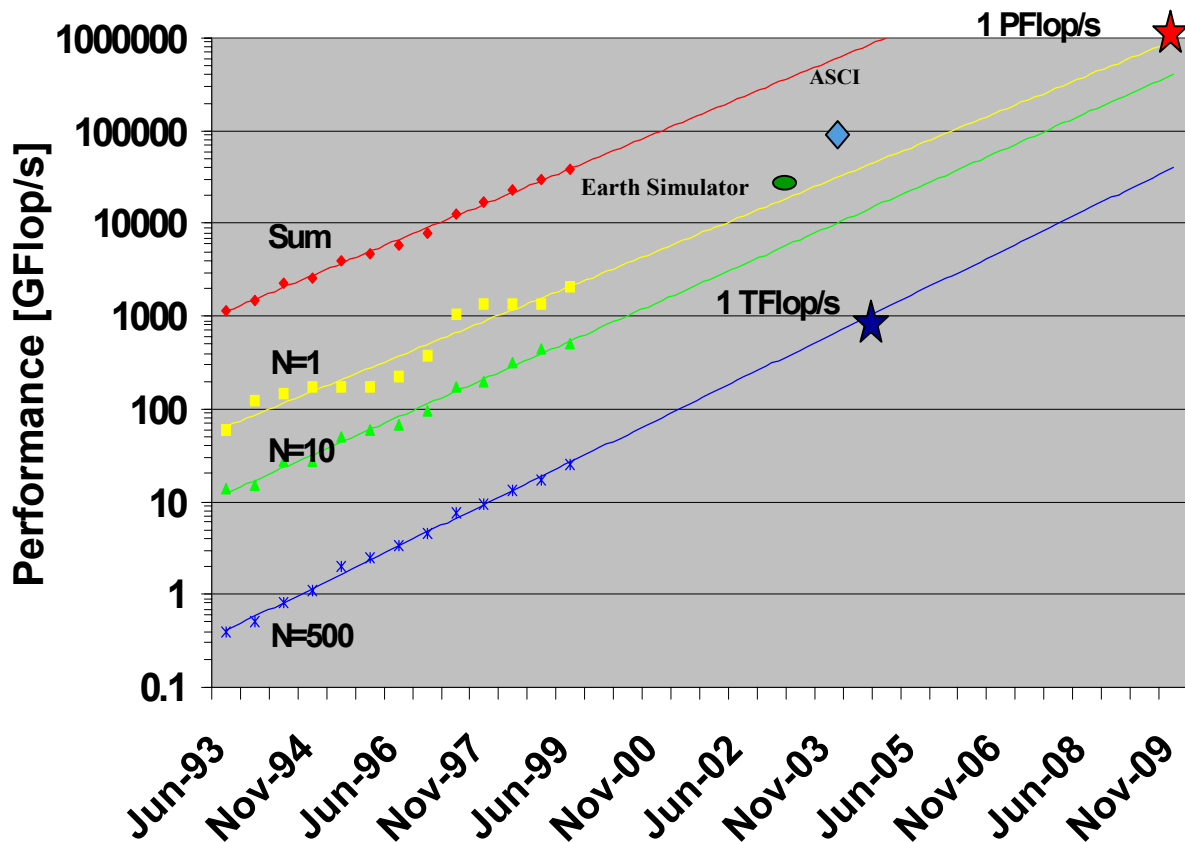
Figure 5: Extrapolation of recent growth rates of performance seen in the Top500

Based on the current Top500 data which cover the last 6 years and the assumption that the current performance development continue for some time to come we can now extrapolate the observed performance and compare these values with the goals of the mentioned government programs. In Figure 5 we extrapolate the observed performance values using linear regression on the logarithmic scale. This means that we fit exponential growth to all levels of performance in the Top500. These simple fitting of the data shows surprisingly consistent results. Based on the extrapolation from these fits we can expect to have the first 100~TFlop/s system by 2005 which is about 1--2 years later than the ASCI path forward plans. By 2005 also no system smaller then 1~TFlop/s should be able to make the Top500 any more.

Looking even further in the future we could speculate that based on the current doubling of performance every year the first Petaflop system should be available around 2009. Due to the rapid changes in the technologies used in HPC systems there is however at this point in time no reasonable projection possible for the architecture of such a system at the end of the next decade. Even as the HPC market has changed its face quite substantially since the introduction of the Cray 1 three decades ago, there is no end in sight for these rapid cycles of re-definition.

Computational Grids

Two things remain consistent in the realm of computer science: i) there is always a need for more computational power than we have at any given point, and ii) we always want the simplest, yet most complete and easy to use interface to our resources. In recent years, much attention has been given to the area of Grid Computing. The analogy is to that of the electrical power grid. The ultimate goal is that one day we are able to plug any and all of our resources into this Computational Grid to access other resources without need for worry, as we do our appliances into electrical sockets today. We are developing an approach to Grid Computing called NetSolve[3]. NetSolve allows for the easy access to computational resources distributed in both geography and ownership. We also describe a parallel simulator with support for visualization that runs on workstation clusters and show how we have used NetSolve to provide an interface that allows one to use the simulator without obtaining the simulator software or the tools needed for visualization.

Network Enabled Solvers

The NetSolve project, under development at the University of Tennessee and the Oak Ridge National Laboratory, has been a successful venture to actualize the concept of Computational Grids. Its original motivation was to alleviate the difficulties that domain scientists usually encounter when trying to locate/install/use numerical software, especially on multiple platforms. NetSolve is of a client/agent/server design in which the client issues requests to agents who allocate servers to service those requests; the server(s) then receives inputs for the problem, does the computation and returns the output parameters to the client. The NetSolve client user gains access to limitless software resources without the tedium of installation and maintenance. Furthermore, NetSolve facilitates remote access to computer hardware, possibly high-performance supercomputers with complete opacity. That is to say that the user does not have to possess knowledge of computer networking and the like to use NetSolve. In fact, he/she does not even have to know remote resources are involved. Features like faulttolerance and load balancing further enhance the NetSolve system. At this point, we offer a brief discussion of the three aforementioned components. The NetSolve agent represents the gateway to the NetSolve system. It maintains a database of servers along with their capabilities (hardware performance and allocated software) and usage statistics. It uses this information to allocate server resources for client requests. The agent, in its resource allocation mechanism, balances load amongst its servers; it is also the primary component that is concerned with fault tolerance. The NetSolve server is the computational backbone of the system. It is a daemon process that awaits client requests. The server can be run on all popular strains of the UNIX operating system and has been ported to run on almost any architecture. The server can run on single workstations, clusters of workstations, or shared memory multiprocessors. It provides the client with access to software resources and also provides mechanisms that allow one to integrate any software with NetSolve servers.

The NetSolve client user submits requests (possibly simultaneously) and retrieves results to/from the system via the API provided for the language of implementation. NetSolve currently supports the C, FORTRAN, Matlab, and Mathematica programming interfaces. The functional interface completely hides all networking activity from the user. NetSolve-1.2 can be downloaded from the project web site at www.cs.utk.edu/netsolve.

There are many research projects in the area of grid based computing. Recently, a book with a number of contributors was published on computational grids. Ian Foster and Carl Kesselman, the originators of Globus, were the editors for this book. It was just published by Morgan Kaufman and is titled *The Grid: Blueprint for a New Computing Infrastructure[4]*. It contains ideas and concepts by many people associated with this new area of computational grids, trying to build up an infrastructure that will allow users to get access to the remote users in a way that makes sense.

References:

1. Dongarra, J.J., H.W. Meuer, and E. Strohmaier, *Top500 Supercomputer Sites*. 1999, University of Tennessee, CS Tech Report, UT-CS 99-434: Knoxville, Tennessee.
2. Dongarra, J., *Performance of Various Computers Using Standard Linear Equations Software*. 2000, University of Tennessee, Computer Science Tech Report CS-89-85: Knoxville.
3. Casanova, H. and J. Dongarra, *Applying NetSolve's Network Enabled Server*. IEEE Computational Science & Engineering, 1998. **5**(3): p. 57-66.
4. Foster, I. and C. Kesselman, eds. *The Grid: Blueprint for a New Computing Infrastructure*. 1998, Morgan Kaufman Publishers: San Francisco, CA. 677.