The Parallel Supercomputer Industry in Light of the Top500 Data

Dror G. Feitelson

School of Computer Science and Engineering

The Hebrew University of Jerusalem

91904 Jerusalem, Israel

Abstract

The Top500 list lists the 500 most powerful computers installed worldwide, and has been updated semiannually for the last 10 years. Analyzing this data enables an impartial assessment of the supercomputer industry and the challenges which it faces.

The Top500 List

The Top500 list grew out of the Linpack benchmark. Originally, this set of linear algebra routines was used to gauge the capabilities of various models and configurations of computers manufactured by different vendors [5]. But in 1993 this was changed. Instead of ranking computer models, the Top500 list was created to rank computer installations. This added a dimension of market success, and removed (or at least significantly reduced) the clutter caused by numerous endeavors that did not pan out. The list of the top ranking 500 installations worldwide has been updated twice a year, in June and November, ever since [6]. In the following analysis, we use the November lists, and denote the maximal computation rate achieved on the Linpack benchmark by Rmax.

Over the years the Top500 list has gained in reputation, with national labs vying to host machines that contend for the top spot. The list has been used to gauge the standing of different vendors and different countries in terms of the production and use of supercomputers. It has also been used to comment on the state of the industry as a whole [7, 3].

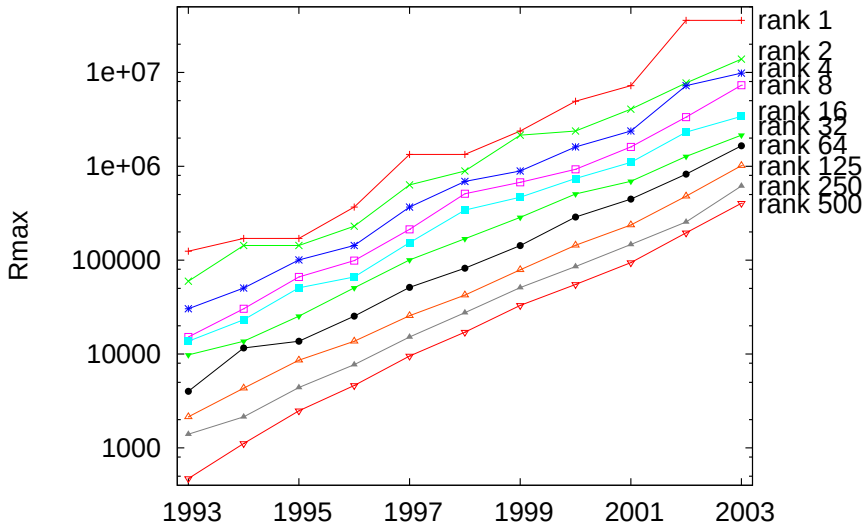

Figure 1: *Increase in computational power with time, as measured using Linpack.*

However, it is not without its shortcomings. Chief among them is the use of the Linpack benchmark to rank the machines. This benchmark focuses on dense matrix operations, and tends to report optimistic performance figures that are closer to peak performance than those typically observed in practice using production applications. Moreover, it may be that the discrepancy between the Linpack measurements and actual performance is different for different classes of computers, making their relative ranking suspect. In addition, there is no regard for the efficiency of the work done or to loss of resources to fragmentation when multiple jobs are executed concurrently [4, 8, 10, 12, 9].

Nevertheless, Linpack has the advantage that it is easily ported and measured, making the Top500 list the only list of its kind. And the accumulated data from 10 years allows for some interesting insights, that are at least partly independent of the actual mechanism used for the ranking.

Ranking and Performance Predictions

It has been widely observed that the performance of machines at a certain rank, and of the list as a whole, grows exponentially with time. This can be seen by tabulating the log of Rmax values as a function of year for a given rank, which leads to a straight line (Figure 1). The slope, however, depends on rank: performing a linear regression on data from 1997

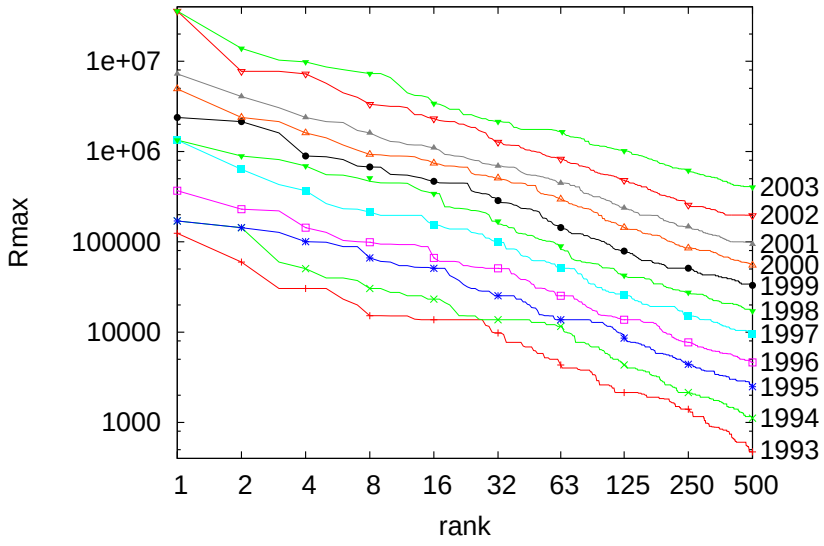

Figure 2: *Distribution of computing power across the lists.*

to 2003, Rmax doubles every 13.5 months for the rank 500 machines, but only every 16.6 months for the rank 16 machines (the data at the very highest ranks is too noisy for an accurate estimate). In other words, the bottom of the list is improving faster than the top. In any case, the rate across the list is somewhat lower than the doubling every year reported in the past [7]. But it is still faster than the doubling every 18 months predicted by Moore's Law.

by tabulating the Rmax values as a function of rank on log-log axes, we find that the computational power drops off polynomially with rank (Figure 2). Discarding the top 31 entries in each list, and performing a least-squares regression on the rest, we find that the slope of the line is growing slightly smaller with time (except in 2000, when it grew). Again, this means that the bottom of the list is actually growing at a slightly faster rate than the top. The current value is about -0.67. This means that as we double the rank, the computational power drops to about 63% of what it was at the original rank.

Based on these regularities, we can actually make crude predictions of the Rmax values at different ranks in future years. We know Rmax grows exponentially with time. Assuming an average time constant of about 1.15 years, we can write

$$Rmax(r,t) = Rmax(r,t_0)\, 2^{(t-t_0)/1.15}$$

where $Rmax(r,t)$ is the Linpack rating at rank r at time t, $Rmax(r,t_0)$ is the rating at this

year	*new machines*
1993	110
1994	206
1995	243
1996	253
1997	319
1998	276
1999	325
2000	307
2001	292
2002	306
2003	333

Table 1: *The number of new entries in the list each year.*

rank at time t_0, and t and t_0 are measured in years. We also know that log(Rmax) drops linearly with log(rank), with a slope of about -0.7,

$$\log(Rmax(r)) = C - 0.7 \log r$$

which leads to

$$Rmax(r) = C'/r^{0.7}$$

Using this to calculate the Rmax value at a rank that is a factor of α higher we get

$$Rmax(\alpha r) = C'/(\alpha r)^{0.7} = Rmax(r)/\alpha^{0.7}$$

implying that Rmax changes by a factor of $1/\alpha^{0.7}$. Putting all the above together we then get the approximate expression

$$Rmax(r, t) = Rmax(r_0, t_0)\ (r_0/r)^{0.7}\ 2^{(t-t_0)/1.15}$$

As an example, let's use the Rmax of the rank 500 machine in 1997, which is 9513, to estimate the Rmax of the rank 100 machine in 2003. The formula gives $9513\ (500/100)^{0.7}\, 2^{6/1.15} = 1,091,848$. The true value is 1,142,000, so the error is less than 5%.

If the slope of log(Rmax) by log(rank) does indeed decrease, the consequence is that machines will tend to fall off the list faster, and the list will end up dominated by new

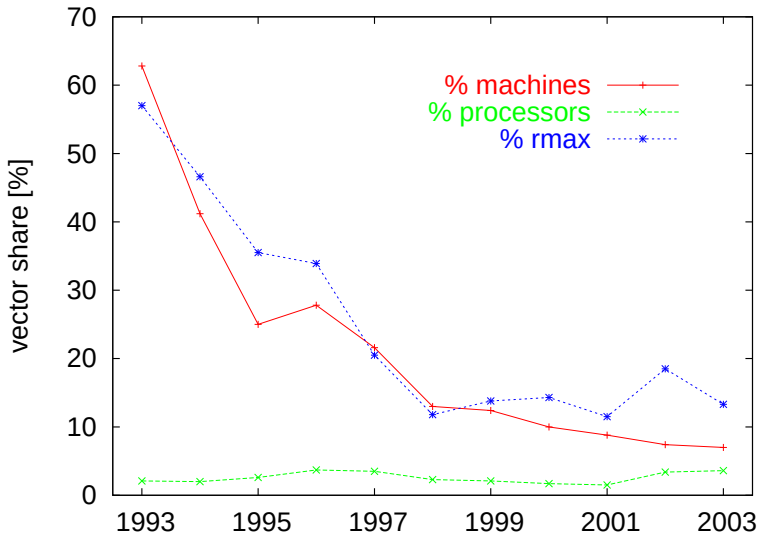

Figure 3: *Share of machines, processors, and cycles held by vector machines.*

machines. To a degree, this is already the case, and has been for a long time: The number of machines in the list drops off exponentially with age [7], and more than half the list is new each year. However, the number of new machines has been relatively static at around 300 at least since 1997 (Table 1). If it grows, it may make sense to increase the length of the list to some number higher than 500. This is also desirable since the prevalence of certain machine models and configurations leads to long stretches of constant values in the current lists.

Vectors and Micros

Much publicity had been given last year to the capture of the top spot by Japan's Earth Simulator [1], after several years of dominance by American ASCI machines. This has also re-ignited the controversy regarding the use of vector processors as opposed to commodity scalar microprocessors. NEC, the creator of the Earth Simulator, has pursued the vector path originally developed by Cray Research, as have other Japanese firms. Most American companies have preferred to use commodity microprocessors, including Cray itself in its T3D and T3E models. The Top500 list has been used repeatedly to show that vector machines are loosing ground to those based on commodity microprocessors (e.g. [3]).

The truth of the matter is a bit more complex. Figure 3 shows the share of machines, processors, and cycles held by vector machines in the last 10 years. The number of vector machines has indeed dropped precipitously. But the share in cycles, as expressed by the Rmax performance values, has stabilized at about 12–15% of the total since 1998, and surpassed 18% when the Earth Simulator was introduced in 2002 (and, if we accept the claim that Linpack over-estimates the capability of microprocessor-based machines, these numbers should actually be larger). Interestingly, the share of installed processors has remained relatively constant throughout the 10-year period. As vector processors are generally much more powerful, there were always many fewer of them, even when vector machines dominated the list. In the mid '90s the other processors were part of massively parallel SIMD machines, and now they are the commodity microprocessors installed in large parallel machines and clusters.

Other interesting data is provided by focusing on the bottom end of the list, and tracking the minimal number of processors needed to make the list. The results, shown in Figure 4, indicate that for microprocessor-based machines this number doubles every three years (as predicted based on much less data in [7]). This explains the finding that the power of installed supercomputers grows faster than that of desktop machines: it is the product of improvements due to Moore's law and using increasing numbers of processors (note also that even vector machines have required multiple processors to make the list since 1997). This applies across the list, as witnessed by the fact that the ratio of the Rmax value of the rank 100 machine to that of the last machine has been remarkably stable at around 3 since 1996.

Comparing the minimal size of microprocessor-based machines with the minimal parallelism in the list, we find that the latter grows faster than the former (Figure 4). As the minimal parallelism is invariably achieved using vector processors, the higher slope indicates that vector processors are improving at a slower rate than microprocessors. If they continue at the same rate, the two lines are expected to cross around 2009. Alternatively, the two types may converge as innovations initially developed for vector processors are incorporated in microprocessor designs.

Limiting Factors

Figure 4 also shows the biggest and top-ranking machines in the Top500 list. Since 1997, it seems that there is a sort of glass ceiling that prevents systems with more than about 10,000

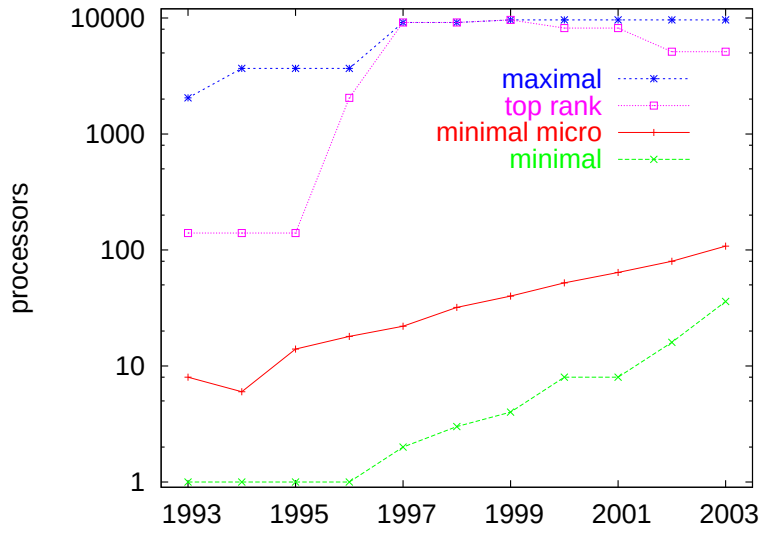

Figure 4: *Maximal and minimal degrees of parallelism represented in the list compared to the minimal number of microprocessors needed and to the parallelism at the top rank.*

processors.

The capabilities of top-ranked machines are the product of how many processors they have and what each one can do. As noted above, vector processors are still more capable than microprocessors, despite having a similar clock rate; but their progress is slower than that of microprocessors. The technology needed to achieve higher performance is and always has been a popular research topic, both in industry and in academia. A major component of this is the question of whether or not Moore's Law will keep working, as chip feature sizes are approaching the atomic scale and manufacturing costs are in the billions of dollars [11]. But for now it is, which implies that the 10,000 processor limit is limiting the growth at the top of the Top500 list. The bottom of the list, meanwhile, is growing faster by increasing the number of processors used.

There are three main factors that limit the number of processors used in large-scale machines. One possible factor is cost. Large scale machines obviously cost a lot of money, with Japan's Earth Simulator coming in at about 350 million dollars. The cost issue is the main motivation for the growth of clusters as an alternative to vendor machines [3]. A few years ago clusters were small-scale endeavors constructed by individual research groups. In the 2003 Top500 list, they occupy 7 of the top 10 slots, and account for more than 40% of the listed systems. As such large-scale clusters are much less expensive than previous

vendor machines, cost cannot explain the 10,000 processor limit.

The second limiting factor is management. This includes two components. One is general management of the machine by its operators, including configuration management, identifying and replacing faulty parts, etc. The other is on-line management by a combination of daemons and operating system services, including runtime support and scheduling. It seems that these management issues may be the main factor that imposes the 10,000 processor limit (which, incidently, is also the number of machines used by Google). Improved management, and especially improved fault tolerance, are needed in order to surpass this number.

A third limiting factor is effective usage by applications. Current programming practices are too rigid, and only scale effectively for very regular problems. There is a need for more dynamic handling of imbalance and failures (which may be seen as an extreme case of imbalance: a failed component has zero capacity, equivalent to infinite load). The problem is basicly one of the programming model employed. Current practices are not flexible enough to deal with massive resources and possible reallocations due to failures or changes in load.

To maintain progress, and in particular to outperform the Earth Simulator using commodity microprocessors, the 10,000 processor bound will have to be broken. A current effort to do so is the BlueGene/L project [2], which has a design point of 65,536 processing nodes. One component of the design is to adopt a SIMD-like style, in which large blocks of processors are used en masse. In addition, nodes will not run a full operating system, but rather a simplified kernel. Interestingly, these design choices are re-incarnations of ideas that were common some 10 years ago, and have been disused in the time since.

Conclusions

The above results (and updates to [7]) are summarized in Table 2, together with predictions based on them. In summary, a large number of parameters have remained relatively static over several years. This does not mean that the supercomputing industry is stagnating, as some of the constants are exponential growth rates. But their stability over time makes it possible to make predictions of how things will change in the future.

Apart from allowing predictions, the analysis also allows us to identify major challenges. An increasingly important challenge is how to produce more efficient management and scheduling mechanisms, that approach the high utilization that has been typical of vec-

invariant	*prediction*
Rmax grows exponentially doubling every 14 to 17 months depending on rank	rank 500 will achieve 1 teraflop in 2005, and rank 1 will achieve 1 petaflop in 2008 or 2009
power drops polynomially with rank, with exponent of about -0.7	machine at rank r will remain on the list $7.2 - \log_{2.365} r$ years
minimal parallelism grows exponentially doubling every 3 years for microprocessor-based machines, and every 1.5 years for vector machines	in 2009 microprocessors will have the same power as vector processors, and about 512 will be needed to make the list
maximum usable parallelism is 10,000	this limit will drive progress towards better management, programming, and reliability
about 15% of the total Rmax is due to vector processors	vectors might make a modest comeback
age drops exponentially in list, with around 300 new machines each year	current growth rate meets current needs; new markets needed to expand usage of parallel supercomputers
about 50% of machines are used by industry	

Table 2: *Invariants and predictions from analysis of the Top500 list.*

tor machines in the past, and at the same time use more than 10,000 processors effectively [9, 10]. This is partly a problem of designing better runtime systems, and partly a problem of devising more flexible programming models. These issues may be more important for long-term progress than the architectural innovations required to achieve a petaflop.

Another challenge is to increase the use of parallel machines outside of the research community. Investigating the fraction of machines installed in industry locations shows that the industry share of machines has doubled since the mid '90s, and peaked at over 52% in 2001. This is typically taken as a good sign, showing the maturity of the supercomputing field. However, in the last two years industry share has dropped by 10 percentage points. Moreover, the industry share of Rmax has grown at a slower rate than its share of machines, and now stands at 27%. Overall, these findings indicate that new markets have to be found to further increase the use of parallel machines.

References

[1] "*Earth Simulator*". URL http://www.es.jamstec.go.jp/.

[2] N. R. Adiga et al., "*An overview of the BlueGene/L supercomputer*". In *SC2002*, Nov 2002.

[3] G. Bell and J. Gray, "*What's next in high-performance computing?*". *Comm. ACM* **45(2)**, pp. 91–95, Feb 2002.

[4] G. Cybenko and D. J. Kuck, "*Revolution or evolution?*". *IEEE Spectrum* **29(9)**, pp. 39–41, Sep 1992.

[5] J. J. Dongarra, "*Performance of various computers using standard linear equations software*". *Comput. Arch. News* **18(1)**, pp. 17–31, Mar 1990.

[6] J. J. Dongarra, H. W. Meuer, and E. Strohmaier, "*Top500 supercomputer sites*". URL http://www.top500.org/. (updated every 6 months).

[7] D. G. Feitelson, "*On the interpretation of Top500 data*". *Intl. J. High Performance Comput. Appl.* **13(2)**, pp. 146–153, Summer 1999.

[8] J. Gustafson, "*Teraflops and other false goals*". *IEEE Parallel & Distributed Technology* **2(2)**, pp. 5–6, Summer 1994.

[9] J. P. Jones and B. Nitzberg, "*Scheduling for parallel supercomputing: a historical perspective of achievable utilization*". In *Job Scheduling Strategies for Parallel Processing*, pp. 1–16, Springer-Verlag, 1999. Lect. Notes Comput. Sci. vol. 1659.

[10] L. Rudolph and P. Smith, "*Valuation of ultra-scale computing systems*". In *Job Scheduling Strategies for Parallel Processing*, pp. 39–55, Springer Verlag, 2000. Lect. Notes Comput. Sci. vol. 1911.

[11] T. N. Theis, "*Beyond the silicon transistor: personal observations*". *Comput. in Sci. & Eng.* **5(1)**, pp. 25–29, Jan/Feb 2003.

[12] A. Wong, L. Oliker, W. Kramer, T. Kaltz, and D. Bailey, "*System utilization benchmark on the Cray T3E and IBM SP2*". In *Job Scheduling Strategies for Parallel Processing*, pp. 56–67, Springer Verlag, 2000. Lect. Notes Comput. Sci. vol. 1911.