HOW TO RANK THE TOP500 LIST?

Lin-Wang Wang
Computational Research Division
Lawrence Berkeley National Laboratory, Berkeley, CA, 94720, USA
e-mail address: lwwang@lbl.gov

With the increasing interest in green computing, power efficiency in supercomputer becomes an important issue. In the high performance computing (HPC) community, the TOP500 list of the fastest 500 computers in the world has been published twice a year for 15 years. So far, the ranking of this list is based solely on the total speed of the computer. With the increased emphasize on power efficiency, one question emerged: how to rank the computers according to both the speed and the power efficiency? One can of cause produce two lists, one according to speed, another according to power efficiency, but there are many advantages to produce a single ranking based on both properties. In this report, we present one approach to sort the TOP500 list. We also show the results using the recently published TOP500 list of June 2008 [http://www.top500.org/lists/2008/06].

The goal of the method is to avoid human factors as much as possible, and to make the formalism as nature as possible. Let's first assume we have selected N=500 best computers from the world. This initial selection can be made based on speed. Our task is to sort these N computers from a given set. Note, we will not produce an absolute "greatness" number for each computer, independent of the rest of the computers in this N computer set. Instead, for the purpose of the sorting, we will produce a score for each computer, "normalized" against the whole computer set. Thus, the value of the score for each computer depends on the whole set. But for the sorting purpose, that is appropriate.

Let's use $P_m(i)$ to denote the property of computer i (i=1,N), and m=1,q is the index of the property, e.g., speed, memory, power efficiency, etc. We will assume, larger $P_m(i)$ means better. Now, given the q properties, how can we sort the N computers? The question is: how to combine $P_m(i)$ into a single score function S(i).

Our first task is to normalize $P_m(i)$. Since different m means completely different properties, with different units and distributions, it is difficult to combine them. We like to convert $P_m(i)$ into $A_m(i)$. One requirement for $A_m(i)$ is that, it must be unit-less. Besides, different properties might have different distributions. Some property might all cluster around a single value, while another property can have a wide distribution. In order to use these properties to distinguish different computers, one likes $A_m(i)$ to have similar width of distribution. To make $A_m(i)$ unit-less, it is reasonable to ask its average from the whole set to be 1. To make it having a given width of distribution, we can require the distribution of $A_m(i)$ to have a given standard deviation around its average value. After some tests, we believe a standard deviation of one is a reasonable choice. Now, we will use the following simple power law to map $A_m(i)$ from $P_m(i)$:

$$A_m(i) = \alpha_m P_m(i)^{\beta_m} \qquad (1)$$

and we have the following requirement:

$$\frac{1}{N}\sum_{i=1,N} A_m(i) = 1 \qquad (2)$$

$$\frac{1}{N}\sum_{i=1,N} [A_m(i) - 1]^2 = 1 \qquad (3)$$

It is easy to determine the α_m and β_m from equations (2) and (3) for a given m.

Fig.1 shows the distribution of $A_1(i)$ for speed (maximum flops) and $A_2(i)$ for power efficiency (maximum flops divided by total powers (watts)). We called them normalized scores based on speed and power efficiency respectively. The original $P_m(i)$ is taken from the recently published June 2008 TOP500 list. Unfortunately, only 247 computers have power efficiency data. So, instead of sorting the original 500 computers, here we only deal with the 247 computers. Thus, N=247. For speed (m=1), we found β_1=0.6136. For power efficiency (m=2), we found β_2=1.321. The α_m values can be found easily as the normalization factors. The distribution of speed $A_1(i)$ is between 0.5 to 10, while the distribution of power efficiency $A_2(i)$ is between 0.1 to 5.

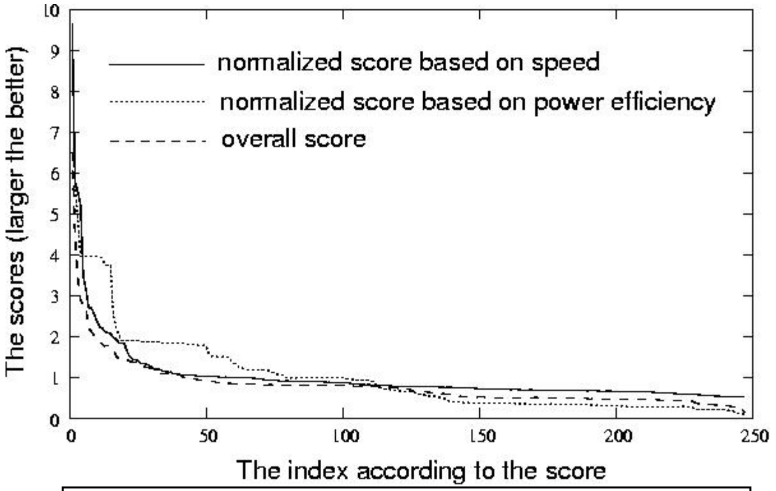

Figure 1, the distribution of the normalized properties (scores) and the overall score.

Our next task is to combine $A_1(i)$ and $A_2(i)$ into a single score (here we only have two properties. But the formula works the same if more properties exist). For this purpose, we propose to use a weighted harmonic mean. That is:

$$S(i) = [\sum_{m=1,q} \frac{W_m}{A_m(i)}]^{-1} \qquad (4)$$

Here $\sum_{m=1,q} W_m = 1$. The harmonic mean is used, so in order to get the high overall score S(i), every individual score $A_m(i)$ for all m need to be high. We have also used a weight function W_m. These weight functions represent the emphases on different properties in the HPC community. It should be decided based on community consensus. In the following, we will discuss a way to estimate the community consensus on W_m. Here however, let's first simply assume $W_1=W_2=0.5$, which means we emphasize equally on the computer total speed and power efficiency. We like to know how does it change our sorting.

In Figs.2, 3, 4, we show the correlations between the overall score S(i) and the score $A_1(i)$ based on computer speed, and score $A_2(i)$ based on power efficiency. We see that, there is not much correlation between $A_1(i)$ and $A_2(i)$ in Fig.2, but as expected, there are some correlation between $A_1(i)$, $A_2(i)$ and S(i) as shown in Figs.3 and 4 respectively. Interestingly, as shown in Fig.2, for those top speed computers, the power efficiency is reasonable. There is no one with very low power efficiency, probably representing the fact that for large computers, power efficiency becomes important. On the other hand, for the low speed computers, some of them have very low power efficiency, although there are also ones with very high power efficiency.

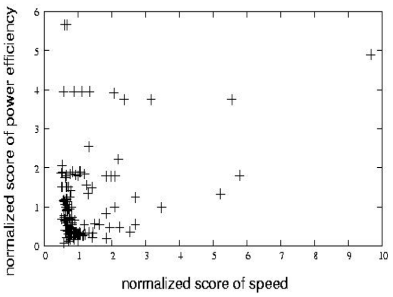

Figure 2, correlation between the normalized scores of speed $A_1(i)$ and power efficiency $A_2(i)$. Each cross represents one computer.

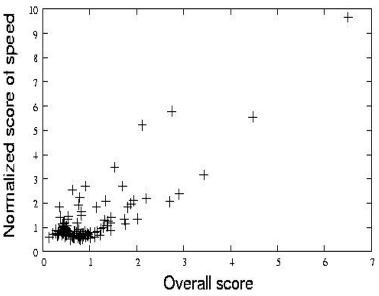

Figure 3, correlation between the normalized scores of speed $A_1(i)$ and overall score S(i).

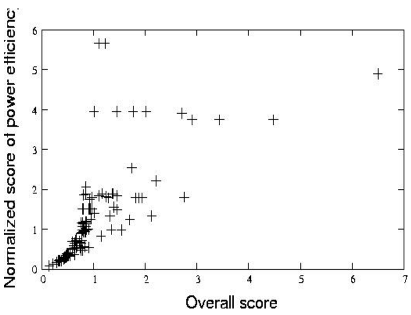

Figure 4, correlation between the normalized scores of power efficency $A_2(i)$ and overall score S(i).

Finally, the new ranking is shown in Fig.5. This ranking is compared with the original ranking which is solely based on the total computer speed. We see that, although there is still a strong correlation between the new and old ranking, the ranking for individual computer has been changed dramatically. This is especially true for the low ranking computers. In Table.I, we show the top 10 computers and their new rankings. We also listed their flops and power efficiencies [$P_1(i)$ and $P_2(i)$].

In the above discussion, we have chosen $W_1=0.5$ and $W_2=0.5$, representing equal emphases for these two properties. An intriguing question is: whether we can determine W_1,W_2 in a natural way. One way to ask this question is: whether there is a community consensus about the value of W_1,W_2. Instead of doing a survey, here we will try to calculate their values based on some economic and cost/benefit assumptions.

The choice of W_1 and W_2 represents a balance between these two properties. We will use the monetary cost to improve $P_1(i)$ and $P_2(i)$ as a measure for this balance. We will assume, for each machine i, the user (builder of the machine) has reached their own balance. Thus, we will first find for each machine, what is its corresponding $w_1(i)$, $w_2(i)$. Then we will take an average over all the machines to get the overall W_1 and W_2 for the whole community. For a given machine i, there is a fixed budget. If more money is spent to improve $P_1(i)$, there will be less money to improve $P_2(i)$. So, let's use $x_m(i)$ to denote the money used for $P_m(i)$. Let's assume that the users (builders) know what they are doing.

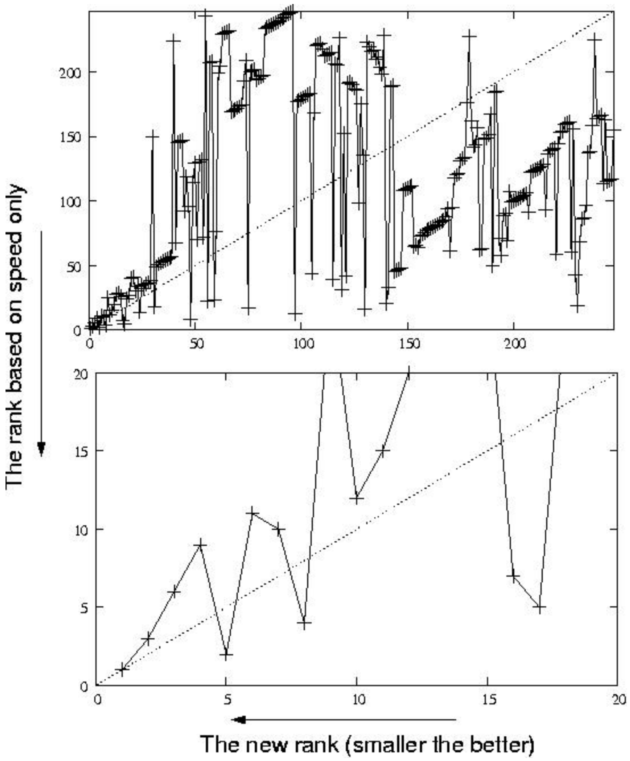

Figure 5, the new ranking (horizontal axis) versus old ranking (vertical axis). W_1=0.5, W_2=0.5. Each point represents one computer. The lower panel is a blow up of the upper panel at the left-lower corner. Index 1 means the first computer, 2 means the second computer, so on.

Table.I, the old and new rankings for the first 10 computers.

Old rank (based on speed)	New rank (W_1=0.5) (W_2=0.5)	New rank (W_1=0.92) (W_2=0.08)	Rmax (Gflops)	Rmax/power (Mflps/Watt)
1	1	1	1,026,000	437.433
2	6	3	478,200	205.271
3	2	2	450,300	357.380
4	9	4	326,000	163.000
5	20	6	205,000	129.689
6	3	5	180,000	357.142
7	16	7	133,200	154.591
8	54	12	132,800	82.884
9	4	8	112,500	357.143
10	7	9	106,100	240.045

We can assume that they are maximizing their machine's score (overall greatness) S(i) by adjusting money $x_m(i)$ among different properties m while keeping the overall money $\sum_{m=1,q} x_m(i)$ fixed. They are doing this maximizing using (consciously or subconsciously) the weight values $w_m(i)$ proper to their problems. Thus we have the condition for the maximum:

$$\frac{dS(i)}{dx_m(i)} = \lambda(i) \tag{5}$$

Here λ (i) is the Lagrangian multiplier for the constant $\sum_{m=1,q} x_m(i)$ constraint. Now, using Eqs.(4) and (1), we have:

$$\frac{dS(i)}{dx_m(i)} = \frac{w_m(i)}{S^2(i)\theta_m(i)} \tag{6}$$

and here,

$$\theta_m(i) = \frac{A_m(i)}{\beta_m \frac{1}{P_m(i)} \frac{dP_m(i)}{dx_m(i)}} \tag{7}$$

In deriving Eq(6), we have ignored the derivatives of α_m and β_m respect to $x_m(i)$. Because α_m and β_m depend on all the computer properties within a N computer set. The change caused by a single computer is small, thus their derivatives by $x_m(i)$ can be ignored. Now, combine Eq(5) and (6), and notice that $\sum_{m=1,q} w_m(i) = 1$, we have:

$$w_m(i) = \frac{\theta_m(i)}{\sum\limits_{m=1,q} \theta_m(i)} \tag{8}$$

After all the $w_m(i)$ are obtained for all the computer i, we can take an average to get the global W_m. The average should be weighted by the overall score S(i) of the computer, so larger the computer, more weight it has on taking this average:

$$W_m = \frac{\sum\limits_{i=1,N} w_m(i) S(i)}{\sum\limits_{i=1,N} S(i)} \tag{9}$$

Note that, since S(i) calculated from Eq(4) depend on W_m, thus Eq(4) and Eq(9) have to be solved together iteratively. Fortunately, a simple direct iteration converges the problem quickly, and $w_m(i)$ do not depend on S(i).

The above formalisms are rigorous. But the final result depends critically on $\frac{1}{P_m(i)} \frac{dP_m(i)}{dx_m(i)} = \frac{d \ln P_m(i)}{dx_m(i)}$. This derivative denotes how much the property $P_m(i)$ can be improved by one dollar (or a thousand dollar, the unit is not important, it will cancel out in Eq(8)). Note that, Eq.(7) makes sense in that, if this derivative is small (which means it is costly to make improvement), but despite of that, the $A_m(i)$ for that property m is still high, then it means this property m is important in this computer i builder's mind, thus the corresponding $\theta_m(i)$ [hence $w_m(i)$] will be large.

Let's now try to estimate $dP_m(i)/dx_m(i)$. If we assume there is a market value, then we can drop index i, i.e, this derivative is machine independent. For computer speed, nowadays, one can buy a 100Tflop computer by about 20M dollar (or say a petascale computer by 200M dollar). Thus, roughly we have

$$\frac{dP_1}{dx_1} = \frac{100Tflops}{\$20M} = 5Tflps/\$M \tag{10}$$

It is more difficult to estimate what is the cost to improve the power efficiency, since it might depend on R&D for new technology. Nevertheless, let's assume that there are readily available technologies to improve the power efficiency, but it just cost money to do it. There is a balance between the up front cost on these techniques and the electric bill saving over the life time of the computer (~3 years). If the up front cost is smaller than the saving, then we can assume that the builder will use those technologies, thus, the efficiency keeps increasing, until further technology improvement are more expensive than the electric bill saving. Thus, at this crossing point (which corresponds to the current computer configuration), the cost of further improving the power efficiency should be equal to the additional electric bill saving. Let's use ΔP_{tot} to denote the total power saving. For each kwatt saving, over three year's life time, and assuming 10 cents per kwatt hour (an international average), the electric bill saving is about \$ 2.628K. That should be the Δx_2. Actually, there might be some other benefits by improving the power efficiency besides the electric bill, for example, removing the need to build a power station, special power line, or other non-monetary reasons, e.g., to develop new technology, save global warming, or to race to the greenest computer. Thus, it is probably safe to say that the cost of improving the efficiency is probably higher than the possible electric bill saving for most of our current computer. Thus, we can add a factor of 2 to take into account of those factors. Thus, our estimation is:

$$\frac{dP_{tot}}{dx_2} = -\frac{1Mwatt}{2.628 \times 2\$M} \tag{11}$$

Note P_{tot} is the total power, not the power efficiency. Thus, we have used a minus sign in Eq(11) to represent the fact the investment on power efficiency will reduce the total power. The power efficiency equals to $P_2 = P_1 / P_{tot}$, here the P_1 is the total speed of the computer. Thus, we have:

$$\frac{dP_2}{dx_2} = -\frac{P_2}{P_{tot}}\frac{dP_{tot}}{dx_2} = \frac{P_2^2}{P_1} \times 0.19Mwatt/\$M \tag{12}$$

Now, using Eq(10) and Eq(12), we can plug into Eq(7) to calculate $\theta_m(i)$, thus $w_m(i)$, and eventually W_m.

Using the June 2008 TOP500 data, and the above formalism, we get W_1=0.92, and W_2=0.08. This is far from treating them equally. In other words, at this moment, the HPC community believes that total speed is still far more important than the power efficiency. If in the future, the computer electric power cost becomes more expensive (not only electric bill, but also the cost to build new power transition station, power lines, or even dedicated power plants), then the weight factor for power efficiency will increase. Using the above new W_m, we have calculated the new ranking. The result is shown in Fig.6. This time, the ranking change is not as dramatic as in Fig.5. Nevertheless, it has changed the ranking of many of the computers. The new rankings are also listed in Table.I for the first 10 computers.

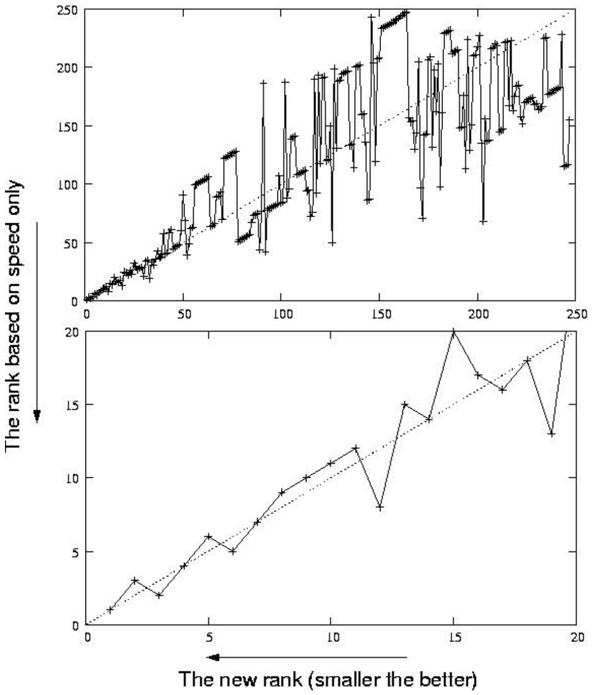

Figure 5, the new ranking (horizontal axis) versus old ranking (vertical axis) using W_1=0.92, W_2=0.08. The lower panel is a blow up of the upper panel at the left-lower corner.

The advantage of the above approach to calculate W_m is that, it doesn't depend on any human decision. Instead, the computer systems, the cost to improve different components, and the market price determine the importance (W_m) of each property. Further investigation for Eq(11) and (12) might be necessary. For example, when the power efficiency is already high, it might be difficult to further improve it due to technology bottleneck, thus there might be a P_2 dependence in Eq.(11). There might also be a P_{tot} dependence in Eq.(11) representing the nonlinear dependence of the electricity cost due to the need for additional equipments (special power transition station, power line, etc).

There could be another way to think about Eqs.(7) and (8). Note that, Eq.(7) can also be written as $\theta_m(i) = A_m(i)\frac{dx_m}{d\ln A_m(i)}$. This is basically how much money it will be involved (cost or saved) if property $P_m(i)$ [hence $A_m(i)$] is changed by some amount. The factor $A_m(i)$ is interesting, it comes because we calculated the overall score S(i) by a harmonic mean. It is logic to use this (the money involved) as the measure of importance for different properties. As the result, in the following discussion [e.g., Eqs.(11) and (12)], it is more proper to think about dx_m/dP_m, i.e., for speed, this will be how much it will cost to increase the speed by a certain Mflops, and for power efficiency, this will be how much it will save (electric bill) to increase the power efficiency by certain amount. Thus, the calculation for these will be more certain, and have more straight forward meaning (although the end result will be the same as above). For example, if the electric bill is extremely cheap, then the power efficiency should not be an issue. Another even more straight forward way is to use the total amount of money currently spend on the speed (to be approximated by the up front cost for the whole computer), and the total amount of money to pay the electric bill in the computer's life time, add them up separately for all the computers in the set, and use the ratio of these two total number to determine W_1 and W_2. If we use $P_1 \times \$M / 5Tflps$ to calculate the total computer cost, and $P_{tot} \times 2.628 \times 2\$M / 1Mwatt$ to calculate the electric bill (we have kept our factor of 2 to take into account the other electricity related cost), then we end up with W_1=0.814, W_2=0.186. This means, for the community as a whole, we have used about 20% of our money on electricity related costs. The resulting W_2 is twice as large as the W_2 calculated from Eq.(8) due to different ways of sampling things. But overall, they are in the same order. It is however this author's belief that Eq.(8) is probably more appropriate because it takes into account the proper scaling and distributions of different properties, etc. Using Eq.(8) makes the whole approach more self-consistent.

Acknowledgments

This work was supported by DOE-SC-ASCR office under Contract No. DE-AC02-05CH11231, and National Energy Research Scientific Computing Center (NERSC) at Lawrence Berkeley National Laboratory. The author likes to thank Dr. Erich Strohmaier for providing the TOP500 data.