

Action: A New Metric for Evaluating the Energy Efficiency on High Performance Computing Platforms (ranked on Green500 List)

E. M. KARANIKOLAOU, M. P. BEKAKOS
Dept. of Electrical & Computer Engineering
Democritus University of Thrace
67100 Xanthi
GREECE

Abstract: - The need for new and more reliable metrics is always in demand. In this paper, a new metric is proposed for the evaluation of high performance computing platforms in conjunction with their energy consumption. The aim of the new metric is to reliably compare different HPC systems concerning their energy efficiency. The metric provides a mean to rank supercomputers of similar capabilities, avoiding the misleading results of metrics like performance-per-watt, currently used for ranking systems, as in the Green500 list, where systems with totally different sizes and capabilities are ranked consecutively. An example of this misuse for two adjacent systems in the Green500 list, is discussed. A comparative study for the energy efficiency of three high performance computing platforms, with different architectures, using the proposed metric is presented. This paper highlights the cases where a metric, like the one that is used in the Green500 list, may produce erroneous results in the ranking of the most energy efficient supercomputers.

Key-Words: Energy efficiency, metrics, supercomputer ranking, HPC, distributed/shared memory platforms, manycore platforms, Green500.

Received: July 5, 2021. Revised: November 8, 2021. Accepted: November 22, 2021. Published: January 5, 2022.

1 Introduction

The evaluation of high performance computing platforms is a complicated issue and is a function of multiple correlated factors. These correlated factors include the application itself, the algorithm, the problem size, the programming language, the implementation, the amount of human effort for optimization, the compiler's version as well as its capability for optimization, the operating system used, the system's architecture, the load from other users or processes, the hardware specifications (CPU, size of cache, memory bandwidth, GPU), as well as the specifications of the interconnection network.

In order to find the best performance of an algorithm on a specific High Performance Computing (HPC) system, the algorithm has to be tested for the maximum size of a problem that could fit into the system's memory. Normally, an amount of memory has to be taken into account for the operating system processes. A good practice for estimating the maximum size of a problem that can fit into the memory is to calculate the problem size for a percentage of 80% of the total system's memory [6].

Beyond performance, computer architects face a new significant challenge, the need for energy

efficiency. Energy becomes an apparent obstacle to realize performance scaling; thus, low power techniques and algorithms for multicore systems, such as Dynamic Voltage Scaling (DVS) [18], have been a major design trend over the last years [3, 4]. Moreover, in an era where energy is a very valuable resource, there is a crucial need to define the tradeoffs between performance and energy consumption on HPC platforms. Energy is as important as performance and the list for ranking the world's most energy efficient supercomputers, the Green500 list [5], is an effort to encourage supercomputing stakeholders to ensure that supercomputers are only simulating climate changes and not creating them [19]. However, the current evaluation metrics used for energy efficiency, as well as for performance measurements, are not optimal. Hence, new evaluation metrics are introduced -over time- measuring overall performance, while taking into account power and energy as well, in order to provide a better framework for ranking high performance computing systems [1, 2].

2 The Current Energy Efficient Ranking of HPC Systems

The Green500 list ranks the world's most energy efficient supercomputers that are selected either from the submitted results to the list or from the Top500 list [6]. This list is released twice a year and the ranking is made using the performance-per-watt metric. A comparative study based on the energy efficiency of a distributed memory, a many-core and a shared memory platform is given in [7, 8]. According to the experimental results and the analysis carried out, the conclusion from the systems' comparison, when using a metric like performance-per-watt or even performance-per-joule, has not produced a clear winner, and depends upon the metric used and the base of the comparison.

Claims of improved performance-per-watt may be used to mask increasing power demands. For instance, though newer generation GPU architectures may provide better performance-per-watt, continued performance increases can negate the gains in efficiency since the GPUs continue to consume large amounts of power. Also, energy required for the climate control of the computer's surroundings often is not counted in the wattage calculation, but still remains quite significant. While performance-per-watt is useful, absolute power requirements are also important.

Moreover, the terms *performance*, *watt*, as well as the division operator, that are used in order to form the metric *performance-per-watt*, may sometimes easily mislead when high performance platforms of totally different sizes and capabilities are compared. It is in the nature of "*ratio*", to sometimes obtain the same results for both a large and a small HPC infrastructure, as far as the energy efficiency is concerned. This can be proved by comparing a pair of two adjacent supercomputers in the Green500 list. For example, the #9 and #10 ranked supercomputers in the Nov.'14 list. The "*Piz Daint*" (#9) achieves an energy efficiency of 54.84 MFlops/W, more than "*romeo*" (#10). The "*Piz Daint*" is an MPP (Massively Parallel Processing) system with a total of 115,984 cores that is ranked in place #6 into the Top500 list and achieves 5,587,000 TFlops with a power consumption of 1753.66 kW (*3185.91Mflops/Watt*). On the contrary, "*romeo*" is a computer cluster with a total of 5,720 cores and is ranked in place #221 into the Top500 list and achieves 254,900 TFlops with a power consumption of 81.41 kW (*3131.06 Mflops/Watt*). Thus, while these two supercomputers are completely different in order of magnitude,

nevertheless they are ranked consecutively in the list Green500. The aforementioned pair is not an isolated case. The same applies also for the pairs ranked in places #11-#12, #12-#13, #22-#23 in the specific, but not limited to, Green500 list and so forth.

3 Introduction of the New Metric

In accordance with the previously discussed paradox, the need to define a new, more fair and more reliable metric for ranking energy efficient supercomputers becomes apparent. The introduction of such a metric must certainly satisfy specific requirements, as for example, the real energy consumption, E , which must not be hidden, i.e., the new metric has to indicate the energy consumption and not conceal it, as it happens in the case of the performance-per-watt metric. The amount of energy required for climate control of the supercomputer's surroundings, also must be considered.

On the other hand, the performance metric to be used has to be valid and reliable; time complexity, t , is the only valid measure of computer performance [14]. Thus, the performance metric must focus upon the time complexity instead of others (MFLOPS, MIPS, etc.) and must correspond to the total time complexity of an algorithm or a benchmark run in order to assess the whole system's performance, incorporating both processing (CPUs) and accelerating (GPUs) capabilities.

Herein, a metric for comparing the energy efficiency of supercomputers, that correlates both the terms *energy consumption* and *time complexity*, is proposed. This metric satisfies the aforementioned requirements forming an accurate, reliable and indisputable metric for ranking energy efficient high performance computing systems. The new metric, named *Action*, S , is expressed as the product of a system's consumed energy (*times*) the time complexity achieved for the solution of a given problem. *Action*, can be utilized either for the comparison of different systems or for the comparison of different algorithms or even for complete benchmarks, like HPCG [2].

The fact that, solely, the energy consumption of an energy efficient system has to be minimum, in conjunction with the fact that the time complexity that a supercomputing system needs for solving a given problem has to be also minimum, leads to the conclusion for the most energy efficient system. It is based on the *minimum* product between the energy consumption and the time complexity that a system needs to solve a given algorithm or to run a benchmark.

The new energy-efficiency metric named “*Action (S)*”, is defined as:

$$S = E x t \quad (1)$$

with unit of measurement the *Joule x second*, in *S.I.* [9].

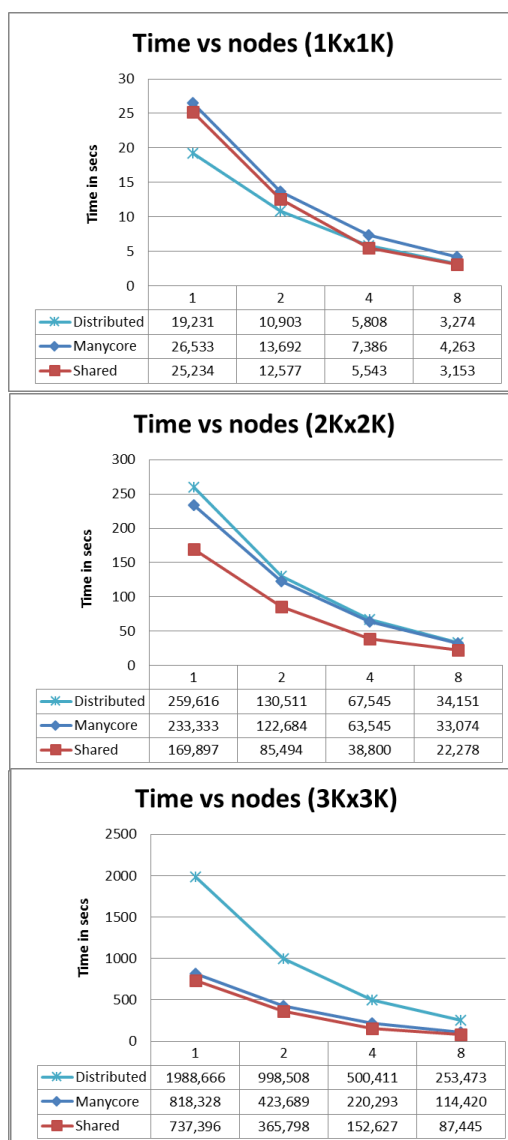
Action borrowed its’ name from physics and this was done for two main reasons. Firstly, the conclusion for a system that is more energy efficient than another is based upon the least value of the metric, corresponding to the *principle of least action* [16], which is fundamental to many areas of physics. Secondly, the unit of measurement for the metric, *Joule x second*, is similar to that of physics quantity “*action*” [15].

4 Experimental Results

The following experimental results were obtained using three different high performance computing platforms, which will be referred to as *Distributed*, *Manycore* and *Shared platforms*. The evaluations for the distributed and manycore platforms were carried out using the MPICH [10] implementation of the Message Passing Interface (MPI) standard, as the programming paradigm. The distributed memory platform was a homogeneous cluster of uniprocessors (Intel P4, 2.8GHz CPU clock speed), while the manycore platform was based upon Opteron [13] processors (having a CPU clock speed at 2.2GHz). The shared memory platform’s evaluation was carried out using the OpenMP [11] standard, which provides a portable, scalable model for developers of shared memory parallel applications. The shared memory platform was a Tyan [12] advanced platform, based upon multiple Opteron multicore processors with a common address space. The experimental vehicle used was the parallel matrix multiplication algorithm, since it exhibits a high level inherent parallelism and offers various parallelization percentages according to the problem size selected. All the experiments were carried out in an isolation mode; namely, the platforms in hand were inaccessible from other users and/or processes. In accordance with the parallelized percentage each time, the performance scalability to the power/energy consumption was calculated. The energy consumption for each platform was calculated based on the analytical models introduced in [7, 8]. In all the diagrams that follow, the three platforms in hand are compared by the execution of the algorithm on the selected problem sizes utilizing different number of cores.

4.1 Total Parallel Execution Time Complexity

In Fig. 1, it is shown the total parallel execution *Time Complexity*, in *secs*, according to the selected problem size and the number of cores used for the three above mentioned platforms in hand (*Distributed*, *Manycore*, *Shared*). The representative problem sizes presented herein, concern matrix dimensions from 1K up to 5K, which progressively scale up, in steps of 1K. The evaluated values of time, resulted taking the average of multiple program executions. A problem size of $mK \times mK$ represents the multiplication of two square matrices of size $m \times 1000$.



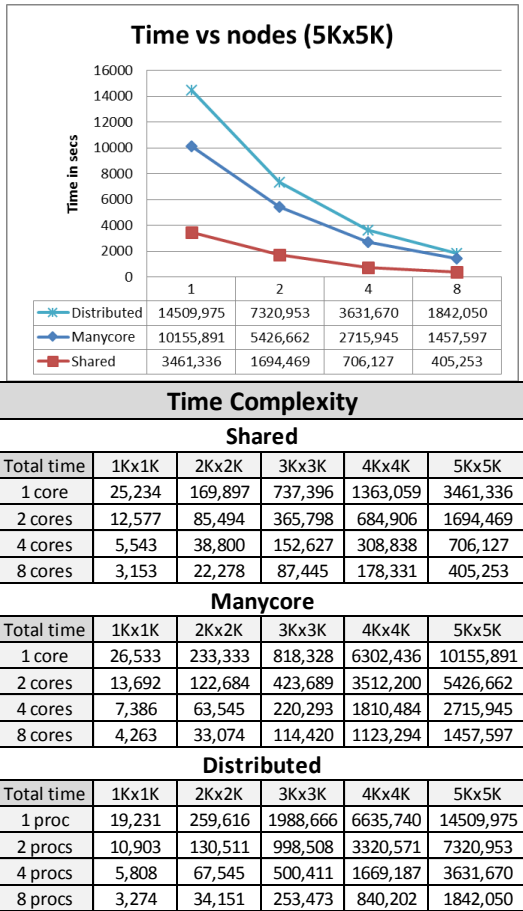
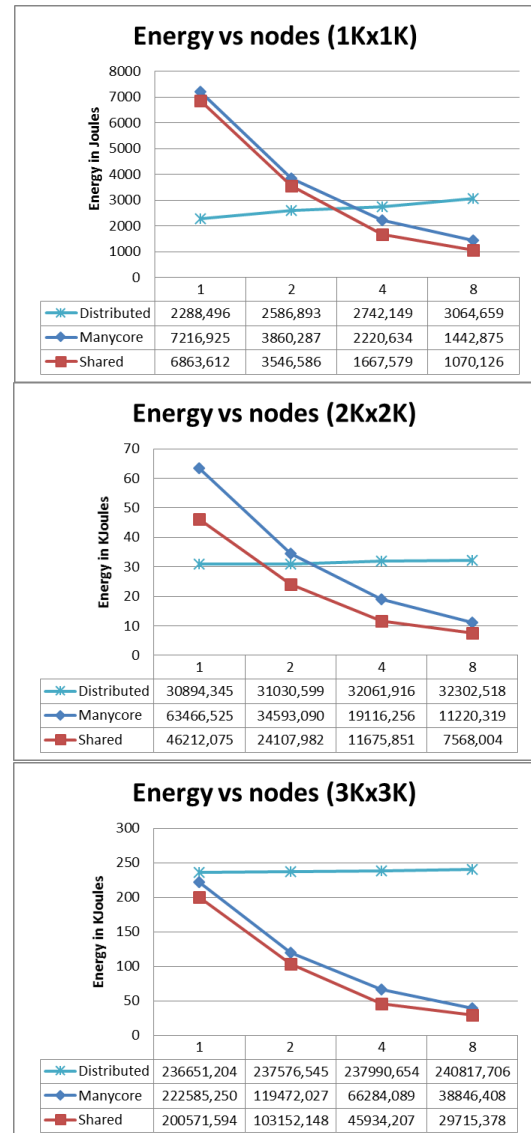


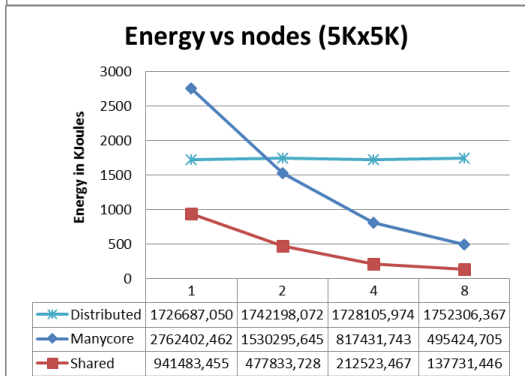
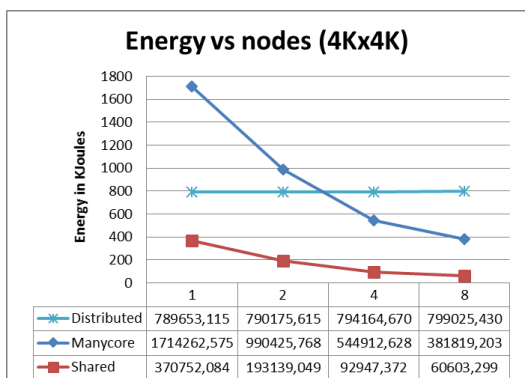
Fig. 1: Evaluation of *Time Complexity* metric, in *sec*, according to the number of cores used for increasingly selected problem sizes

From diagrams (a) to (e) of Fig. 1, it is observed that, for all problem sizes, the execution time complexity decreases as the number of cores used increases, which shows that the algorithm is scalable, and this applies to all three platforms. From the same diagrams it is also clear that the fastest platform for the specific algorithm is the *Shared* platform. The main reason for this is the reduced processors' communication time, due to the shared memory architecture. It is worth noting that the different time results from the execution of the algorithm on the *Shared* and on the *Manycore* platform using one core is different, due to the reason that while the underlying hardware is the same, the software is different (MPI vs OpenMP), thus forming two different parallel systems. This fact, which is emphasized in Fig. 1(d,e), is mainly caused due to the increased overhead of the MPI routines from one side versus the capability of OpenMP compiler for better parallelization, which may also occur to improved cache hit ratio, from the other side.

4.2 Total Energy Consumption

The other factor that has to be known in order to evaluate the values of the *Action* metric is the total *Energy* that a platform consumes, upon executing the algorithm for a specific workload (i.e. problem of size). To evaluate the total consumed energy (with or without taking into consideration the required energy for the cooling of the equipment) for the execution of an algorithm, a central energy-meter that takes account the whole of the equipment is demanded. Alternatively, wherever this is not an option, the energy consumption can be evaluated upon specific analytic models for each computing platform, respectively [7, 8]. For the three platforms in hand, using as experimental vehicle the parallel matrix multiplication algorithm for the selected problem sizes, the evaluated results for the Energy consumption, using the analytical models in [7,8], are presented in Fig. 2.

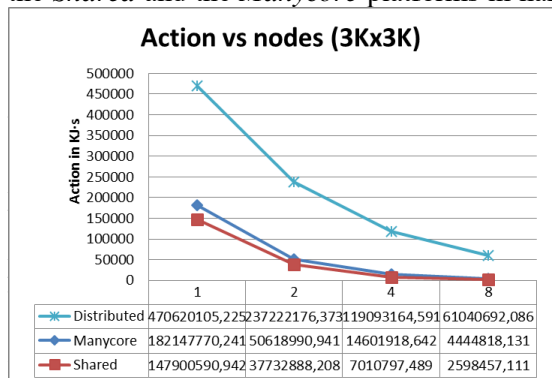




Energy				
Shared				
Energy	1Kx1K	2Kx2K	3Kx3K	5Kx5K
1 core	6863,612	46212,075	200571,594	370752,084
2 cores	3546,586	24107,982	103152,148	193139,049
4 cores	1667,579	11675,851	45934,207	92947,372
8 cores	1070,126	7568,004	29715,378	60603,299
Manycore				
Energy	1Kx1K	2Kx2K	3Kx3K	5Kx5K
1 core	7216,925	63466,525	222585,250	1714262,575
2 cores	3860,287	34593,090	119472,027	990425,768
4 cores	2220,634	19116,256	66284,089	544912,628
8 cores	1442,875	11220,319	38846,408	381819,203
Distributed				
Energy	1Kx1K	2Kx2K	3Kx3K	5Kx5K
1 proc	2288,496	30894,345	236651,204	789653,115
2 procs	2586,893	31030,599	237576,545	790175,615
4 procs	2742,149	32061,916	237990,654	794164,670
8 procs	3064,659	32302,518	240817,706	799025,430

Fig. 2: Evaluation of Energy metric, in Joules, according to the number of cores used for increasingly selected problem sizes

From the respectful diagrams, it is observed that for the Shared and the Manycore platforms in hand, the

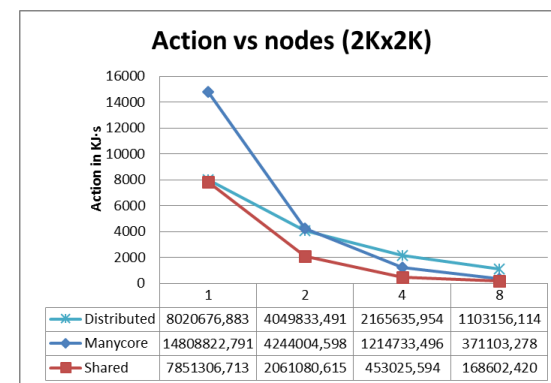
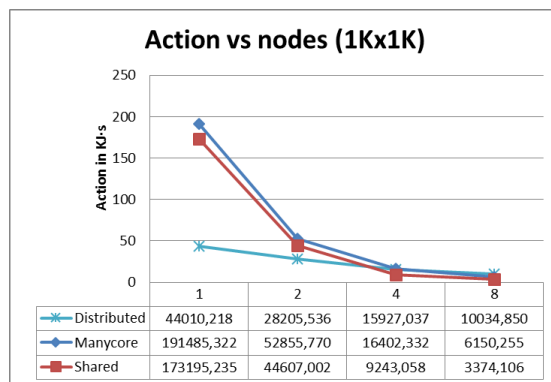


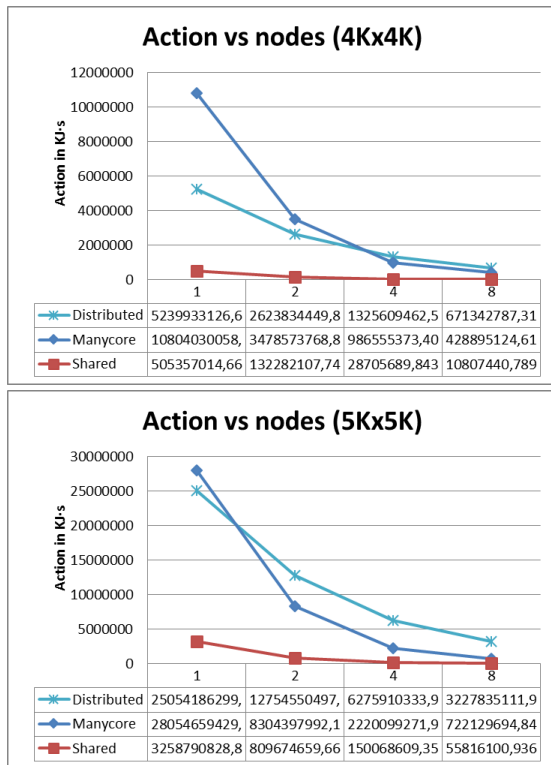
more execution. Watts, cores of er, as tate, it e total n full many- y the cluded

and measured by a Wattmeter. From the other side, the Energy consumption for the Distributed platform in hand is larger and “seems” to be constant as it can be seen from the diagrams. The Energy consumption is respective to the operating Power, which for a distributed environment is analogous to the number of nodes that take part into the problem execution [7]. In addition, one must take into account that, the total execution time, which is also remarkable higher in most of the cases for the Distributed platform, is the second term that defines the calculation of Energy.

4.3 The Action Metric

Using the total Time Complexity and the consumed Energy, for the platforms in hand, the evaluation of the new metric, Action, is feasible now. In Fig. 3 there are shown, in a comparative base, the values for the three different systems that were evaluated using the Action metric, in J-s, according to the number of cores used for increasingly selected problem sizes.





Action					
Shared					
Action	1Kx1K	2Kx2K	3Kx3K	4Kx4K	5Kx5K
1 core	173195,235	7851306,713	147900590,942	505357014,662	3258790828,873
2 cores	44607,002	2061080,615	37732888,208	132282107,740	809674659,664
4 cores	9243,058	453025,594	7010797,489	28705689,843	150068609,357
8 cores	3374,106	168602,420	2598457,111	10807440,789	55816100,936
Manycore					
Action	1Kx1K	2Kx2K	3Kx3K	4Kx4K	5Kx5K
1 core	191485,322	14808822,791	182147770,241	10804030058,991	28054659429,231
2 cores	52855,770	4244004,598	50618990,941	3478573768,899	8304397992,186
4 cores	16402,332	1214733,496	14601918,642	98655373,403	2220099271,912
8 cores	6150,255	371103,278	4444818,131	428895124,619	722129694,846
Distributed					
Action	1Kx1K	2Kx2K	3Kx3K	4Kx4K	5Kx5K
1 proc	44010,218	8020676,883	470620105,225	5239933126,636	25054186299,071
2 procs	28205,536	4049833,491	237222176,373	2623834449,870	12754550497,913
4 procs	15927,037	2165635,954	119093164,591	1325609462,578	6275910333,976
8 procs	10034,850	1103156,114	61040692,086	671342787,313	3227835111,920

Fig. 3: Evaluation of *Action (S)* metric, in *Joule x sec*, according to the number of cores used for increasingly selected problem sizes

From diagrams (a) to (e) of Fig. 3, it becomes apparent that, in conjunction with the performance, the most energy efficient platform in hand for the specific algorithm is the *shared* memory platform. This is due to the fact that, especially for the cases where the maximum number of cores are used, the values obtained for the action metric were minimum, implying that the particular platform achieved the least value for the product of the *Time* required to solve the problem and the *Energy* which consumed, compared to the other platforms, in proportion to the principle of least action.

5 Application of Action Metric on Green500 List

On the experimental results presented above, the *Action* metric was evaluated for specific problem sizes, which is common, in general, whenever comparisons take place, since in order to compare a system/platform to another some quantity must be kept constant (e.g. the problem size in this case). In case of comparing a few number of systems, this method is accepted but if one want to use it to create a large list with characteristics such that of the Green500, then the platforms' performances are used indirectly (e.g. through faster execution time) and any direct use of a platform's maximal achieved performance is left behind. In Green500 list, the results arise from the ratio of the values *Rmax* and *Power* which are mainly feeded from the Top500 list (except those results that are exclusively submitted to the Green500 list). These values are platform specific, with *Rmax* being the maximal LINPACK performance achieved, in *TFlop/s*, and *Power* being the electric power that a system needs in order to operate, in *kW*. The Linpack Benchmark is a measure of a computer's floating-point rate of execution. It is determined by running a computer program that solves a dense system of linear equations. By measuring the actual performance for different problem sizes *n*, a user can get the maximal achieved performance *Rmax* for a problem size *Nmax* [6]. Therefore, *Nmax* is different for almost any *Rmax* that each systems achieves.

On Green500 list, the ranking is made without keeping any quantity constant (e.g. the problem size, the value of *Rmax*, the amount of Energy consumption, the operational Power), but the benchmark. In the context of comparing and ranking supercomputers as in the Green500 list, one can start by evaluating the *Action* metric which can be used selecting a specific problem size *n*. In order for any system to be able to run the LINPACK benchmark, the best candidate as a problem size would be the minimum *Nmax* of the list, so as even the smallest system could run it. Then, after re-running the benchmark and obtaining the total execution *Time Complexity*, the evaluation of the *Energy* that any system consumes is feasible, since the *Power* consumption of any system is prior known. Finally, by calculating the results of the *Action* metric a new ascending ranking list can be formed. The higher place a system holds the better it is, in terms of performance in conjunction with energy consumption. This list would eliminate the erroneous ranking that the current Green500 list now presents, ranking consecutively

supercomputing systems with totally different characteristics, as noted in Chapter 2.

6 Extension of the Action Metric

In order to specifically take into consideration the maximal achieved performance R_{max} of any system, the Action metric could be extended in order to form a more general metric, let say $S_{G(i)}$, for each system i , with the following definition:

$$S_{G(i)} = \frac{S_{(i)N_{max}(min)}}{R_{max(i)}} \quad (2)$$

where $S_{(i)N_{max}(min)}$ is system's i evaluated S metric value for the minimum problem size, N_{max} , of the Green500 list and $R_{max(i)}$, in TFLOPS, is system's i maximal evaluated LINPACK performance achieved, which can be found in the Top500 or the Green500 list.

In that terms, not only the systems will be ranked appropriately according to the minimum specific problem size via the Action metric, but also they will be ranked according to their overall performance, eliminating the aforementioned problems that exist in all Green500 lists, due to the currently selected metric still used today.

Ultimately, in order to compare systems for a given problem size, one can use the Action, S , metric, while for more generalized comparisons, where R_{max} is available, the more generalized Action metric, S_G , could be used.

7 Conclusions

Herein, a metric for comparing the energy efficiency of supercomputers, that correlates the terms energy consumption and time complexity, was introduced. A comparative study for the energy efficiency of three high performance computing platforms of different architectures, was also discussed. The new metric is capable to compare and reliably rank high performance computing systems, in a direct way, as far as their energy efficiency is concerned, ranking systems of similar size and computational capabilities. The outcome for the most energy efficient system is based upon the minimum value of the presented Action metric. An example of two adjacent systems ranked into the Green500 list was used in order to highlight how misleading can be a metric like performance-per-watt, which is widely used today, for ranking systems into the Green500 list. From the samples discussed, it becomes obvious that recent Green500 lists may contain

more than one cases where systems of totally different computational capabilities are ranked consecutively.

While performance-per-watt is useful, absolute power requirements are also important. The invention and the assessment of new metrics, such as Action, that concern the performance, the energy efficiency of the current and future high performance computing systems and the qualitative as well as the social contribution of parallel processing into the modern way of life, will always be a hot research area.

References:

- [1] Subramaniam, B., & Feng, W. C. (2012). The green index: A metric for evaluating system-wide energy efficiency in hpc systems. In *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International (pp. 1007-1013)*. IEEE.
- [2] Heroux M.A., Dongarra J. (2013). Toward a new metric for ranking high performance computing systems, *Sandia National Laboratories, SAND2013-4744*.
- [3] Yang, L., Lin, M., Yang, T. (2012). Multi-core Fixed Priority DVS Scheduling, Algorithms and Architectures for Parallel Processing, *Lecture Notes in Computer Science*, v.7439, pp. 517-530.
- [4] Lu, J., Guo, Y. (2011). Energy-Aware Fixed-Priority Multi-core Scheduling for Real-Time Systems, *RTCSA-IEEE 17th International Conference Proceedings*, v.1, pp. 277-281.
- [5] The Green500 List, <http://www.green500.org>
- [6] Top500 F.A.Q., <http://www.top500.org/resources/frequently-asked-questions>
- [7] Karanikolaou, E.M., Milovanović, E.I., Milovanović, I.Ž., Bekakos, M.P. (2014). Performance scalability and energy consumption on distributed and many-core platforms, *The Journal of Supercomputing*, v.70.1, pp. 349-364.
- [8] Karanikolaou, E.M., Bekakos, M.P. (2014). Performance scalability and energy consumption on a Shared Memory Platform, *Neural, Parallel and Scientific Computations*, Vol. 22, pp. 623-638.
- [9] International System of Units, http://en.wikipedia.org/wiki/International_System_of_Units

- [10] High-Performance Portable MPI (MPICH), <http://www.mpich.org>
- [11] The OpenMP® API specification for parallel programming, <http://openmp.org>
- [12] TYAN - server motherboards, server barebones for HPC, GPU, Cloud Computing and embedded applications, <http://www.tyan.com>
- [13] AMD Server Processors, <http://www.amd.com/en-us/products/server>
- [14] Hennessy J.L., Patterson D.A. (1998). *Computer Organization and Design (2nd edition)*, Chapter 2: The Role of Performance, Morgan Kaufmann.
- [15] Action (physics), [http://en.wikipedia.org/wiki/Action_\(physics\)](http://en.wikipedia.org/wiki/Action_(physics))
- [16] Lanczos, C. (1986). *The Variational Principles of Mechanics, 4th edition*, Dover Publications, New York.
- [17] Marowka, A. (2017). Energy-Aware Modeling of Scaled Heterogeneous Systems, *International Journal of Parallel Programming*, v.45.5, pp. 1026–1045.
- [18] V. Hanumaiah and S. Vrudhula (2014). "Energy-Efficient Operation of Multicore Processors by DVFS, Task Migration, and Active Cooling," in *IEEE Transactions on Computers*, vol. 63, no. 2, pp. 349-360, Feb. 2014.
- [19] U. Wajid *et al.* (2016). On Achieving Energy Efficiency and Reducing CO₂ Footprint in Cloud Computing, in *IEEE Transactions on Cloud Computing*, vol. 4, no. 2, pp. 138-151.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0
https://creativecommons.org/licenses/by/4.0/deed.en_US