HPC-Oriented Power Evaluation Method

Feng Zhang
Department of Computer Science and Technology
Tsinghua University
Beijing, China
Email: feng-zhang12@mails.tsinghua.edu.cn

Liang Chen*
Schlumberger Beijing GeoScience Center
Schlumberger Technologies (Beijing) Ltd.
Beijing, China
Email: cl_sml_light@hotmail.com

***Abstract*—In the field of high performance computing (HPC), energy consumption is an increasingly important consideration. The Standard Performance Evaluation Corporation (SPEC) Power benchmark and the Green500 are two well-known power evaluation methods. However, the former focuses on datacenters and the latter concentrates on compute-intensive applications. This paper focuses on the power evaluation of single multi-core HPC servers. We analyze the limitations of these existing evaluation methods and construct a novel evaluation method using the High-Performance Linpack (HPL) and NAS Parallel Benchmarks-Embarrassingly Parallel (NPB-EP) programs. We conduct experiments on three HPC servers to test the evaluation method. The results from our evaluation method differ from the results of the Green500, and are more general and close to real-world HPC applications. We also build a regression model of power to assist in the analysis. We use the HPC Challenge Benchmark (HPCC) to train the model and use the NPB to perform verification. The R^2 representing similarities for the B and C classes of the NPB are 0.634 and 0.543, indicating that the model satisfies most cases.**

I. INTRODUCTION

Energy consumption is becoming a serious problem in the High Performance Computing (HPC) industry. As computers are continually developed and improved, many fields grow more reliant on them. As this is happening, datacenters are being used more while the cost of computer hardware is decreasing. That is, the cost of hardware for building up large-scale clusters and data centers is decreasing. However, electricity costs and cooling costs are increasing dramatically. For example, the cost of energy consumption and cooling can exceed the cost of adding new equipment [15]. Energy consumption in the HPC industry is tremendous and there are demands for related power evaluation methods. The Standard Performance Evaluation Corporation (SPEC) Power benchmark [16] and the Green500 [7] are two major power evaluation methods.

SPEC established SPECpower_ssj2008 [6] to drive energy efficiency initiatives. However, our research indicates that SPECpower_ssj2008 does not apply to the HPC field for the following reason. SPECpower_ssj2008 is the first industry standard benchmark for testing the power of servers, but it simulates workloads for a datacenter, which differs from workloads in the HPC field in three key aspects. First, the memory usage remains at a low level (less than 14%) while the memory usage in HPC programs is relatively high. Second, the CPU usage declines with a decrease in workload while the CPU usage in HPC programs remains high for most problem sizes. Third, the number of cores involved in execution is not configurable while a convenient HPC power benchmark provides configuration for cores.

We also investigated the Green500 and found that it only represents the characteristics of a small fraction of HPC programs. The Green500 uses High-Performance Linpack (HPL) [2], which is a highly parallel computing benchmark. HPL is an extremely compute-intensive program and can reach nearly 90% of the theoretical peak performance. For example, the theoretical peak performance of the server Xeon-4870 is 384 GFLOPS and the HPL result is 344 GFLOPS in our experiments. However, we applied the NAS Parallel Benchmarks (NPB) [8], which is a set of programs representing real performance of HPC programs, and found that most programs fail to reach that performance. Moreover, HPL requires more power to run than the NPB does.

We perform the power experiments for the NPB and HPL, and prove that their statistics are irreplaceable. The configuration for NPB-EP (where EP refers to the Embarrassingly Parallel kernel) is flexible and EP has contrary performance and power characteristics to the characteristics of HPL, and we use these two programs, HPL and NPB-EP, to obtain the power characteristics for servers. We demonstrate that the number of cores and memory usage are decisive factors for power evaluation. This paper mainly focuses on single multi-core servers. We choose HPL and EP as evaluation programs and test the system power in five states: (1)Idle; (2)Full CPU usage and full memory usage; (3)Half CPU usage and full memory usage; (4)Full CPU usage and half memory usage; and (5)Half CPU usage and half memory usage. We conduct experiments on three servers and provide comparison results with existing power evaluation methods. Our results are different from the results using only HPL (The Green500 method). Our results of HPL and EP represent the power characteristics for general programs.

Since we found that the number of cores and memory used in experiment are the two main decisive factors for power, we divide the total power into the following three sections: the power of CPUs, the power of memory, and the power of other parts. Note that the power of other parts is considered constant and denoted C. We use the HPC Challenge Benchmark

---

*Liang Chen did this work during postgraduate in Tsinghua University.

(HPCC) [11] to build a multiple linear regression model of power. The R Square value of this model is 0.94, indicating that the selected indicators are strongly correlated with power. Moreover, we input the results of the NPB using classes B and C, which denote the sizes of test problems, into our model and we define the evaluation criteria. The R^2 representing similarities for NPB-B and NPB-C are 0.634 and 0.543, which is greater than 0.5, indicating the results are satisfactory for most cases.

We make three main contributions in this work:

- First, we quantitatively study the limitations of the SPEC Power and Green500, using the NPB as a control. We show the difference between the two well-known benchmarks and general HPC programs, indicating the necessity for a new general power benchmark for HPC servers.
- Second, we propose a power evaluation benchmark which combines HPL and EP, with the CPU and memory required to perform in multiple states. Our result for three tested servers is different from the Green500 result, with ours being more general.
- Third, we provide a power regression model to assist the analysis. We train the model using the HPCC benchmark and verify it using the NPB. The R^2 is greater than 0.5, showing the regression results and measured results are close.

The remainder of this paper is organized as follows. Section II details the architectures of the servers Xeon-E5462, Opteron-8347, and Xeon-4870. Section III describes the relevant evaluation methods - the SPEC Power benchmark, the Green500, and the NPB. Section IV reviews the limitations for the SPEC Power and Green500, and provides the motivation for building a new HPC benchmark. Section V presents the proposed power evaluation method and performs experiments on three servers to compare it to the SPEC Power and Green500. Section VI shows the regression model of power. Section VII discusses related work. Section VIII provides the conclusion.

II. HIGH-END COMPUTING PLATFORMS

We list the servers used in this paper in Table I, using the processor names to represent the server names.

A. Server Xeon-E5462

The server Xeon-E5462 uses the Xeon E5462 [3] running at 2.8 GHz. Each core processes 4 operations per cycle and the performance for each core is 11.2 GFLOPS (Giga Floating-point Operations per Second). This server has 4 cores in total so the theoretical peak server performance is 44.8 GFLOPS. For Level 1 caches, the processor has 4 x 32 KB 8-way set associative instruction caches and 4 x 32 KB 8-way set associative write-back data caches. The Level 2 caches of this processor are 2 x 6 MB 24-way set associative shared caches.

B. Server Opteron-8347

The server Opteron-8347 has four Opteron 8347 processors [1] running at 1.9 GHz. Each core has 7.6 GFLOPS and the theoretical peak server performance is 121.6 GFLOPS. Level 1 caches for each processor are 4 x 64 KB 2-way associative instruction caches and 4 x 64 KB 2-way associative data caches. Level 2 caches are 4 x 512 KB 8-way set associative caches. Level 3 caches are 2 MB 32-way set associative shared caches.

C. Server Xeon-4870

The server Xeon-4870 has 4 Xeon E7-4870 processors [4] running on 2.4GHz. Each core has 9.6 GFLOPS and the theoretical peak server performance is 384 GFLOPS. Level 1 caches for each processor are 10 x 32 KB 4-way set associative instruction caches and 10 x 32 KB 8-way set associative data caches. Level 2 caches are 10 x 256 KB 8-way set associative caches. For Level 3 caches, each chip has a 30 MB 24-way set associative shared cache.

III. BENCHMARKS

We focus on two main power benchmarks, the SPEC Power and the Green500, as well as an HPC evaluation benchmark for servers, the NAS Parallel Benchmark suite (NPB). We use these to test the processors and memories, and provide insight into the relation between power and performance.

A. SPEC Power

The SPEC Power benchmark suite [16] measures the power characteristics for servers and SPECpower_ssj2008 is the first industry standard power benchmark used to test the power statistics of servers. As e-commerce thrives and investment in HPC increases, server energy consumption has gradually increased. Thus, there is increasing need for a unified test platform to evaluate systematic power performance. SPECpower_ssj2008 simulates real data center workloads and provides a unified power performance test method, allowing the comparison of test results among different systems.

SPECpower_ssj2008 includes four parts: Server Under Test, Control and Collect System, Power Analyzer, and Temperature Sensor. The procedures are as follows: First, the Control and Collect System goes through three calibration phases and measures the peak number of requests for the system. Second, the Control and Collect System progressively decreases the data requests according to 10% of the peak magnitude. This decline occurs over a fixed time. During this process, the system collects the ssj_ops (server side Java operations per second) data and energy consumption data. This step repeats until the final number of requests is decremented to 0%. Third, the system summarizes the ssj_ops and energy consumption for each stage, and uses the ratio ssj_ops/Watt (server side Java operations per second per watt) as a final evaluation.

TABLE I

SYSTEM CHARACTERISTICS OF THE SERVERS USED.

Model	Server Xeon-E5462	Server Opteron-8347	Server Xeon-4870
Processor Type:	Xeon E5462	Opteron 8347	Xeon E7-4870
Processor Characteristics:	Quad-Cores	Quad-Core	10
CPU Frequency (MHz):	2800	1900	2400
Core(s) Enabled:	4cores, 1 chips, 4 cores/chip	16cores, 4 chips, 4 cores/chip	40 cores, 4 chips, 10 cores/chip
Hardware Threads / chip:	4	4	20
Primary Cache / chip:	4x32KB icaches and 4x32KB dcaches	4x64KB icaches and 4x64KB dcaches	10x32KB icaches and 10x32KB dcaches
Secondary Cache:	6MB (12MB total)	512KB per core	256KB per core
Tertiary Cache:	0	2048KB per processor	30MB per processor
Memory Amount (GB):	8	32	128
Memory Details:	DDR2	DDR2	DDR2
Power Supply Quantity and Rating (W):	1 x Unknown	1 x Unknown	3 x Unknown
Disk DriveGB:	400	444	152
Disk Controller:	Integrated SAS controller	Integrated SAS controller	Integrated SAS controller
Network Speed (Mbit):	1000	1000	1000

B. Green500

The Green500 uses performance per watt (PPW) as its rank reference. Linpack is a program for performing linear algebra problems and parallel High-Performance Linpack (HPL) is used as a benchmark to rank supercomputers. The performance measure is the maximum performance result achieved by the Linpack benchmark test and it is represented by R_{max}. The power measure is the average system power during the maximum performance execution of Linpack and it is represented by $P_{avg}(R_{max})$. Then, PPW is defined as

$$PPW(GFLOPSPerWatt) = \frac{R_{max}(inGFLOPS)}{P_{avg}(R_{max})(inWatt)} \qquad (1)$$

The Green500 does not have a Control and Collect System. The Green500 includes the following stages: First, the power meter data logger software is launched. Second, the Linpack benchmark is launched using the input file which can guarantee that the system has peak performance. Third, the power samples begin recording and then stop recording after the Linpack test. Fourth, the Linpack performance result and the average power result are used to compute the PPW. The first and last few samples can be ignored for this calculation to prevent inaccurate records of power meters.

C. NPB

NPB refers to the NAS Parallel Benchmarks. Although Linpack can reach nearly 90% of the peak performance of the system, it cannot represent the real performance for HPC programs as most programs do not reach that high peak performance. The NPB benchmark suite includes eight programs, which are five kernels and three pseudo-applications. Its test results represent the real performance of HPC programs. The five kernels are Integer Sort (IS), Embarrassingly Parallel (EP), Conjugate Gradient (CG), Multi-Grid (MG), and discrete 3D fast Fourier Transform (FT). The three pseudo-applications are the Block Tri-diagonal solver (BT), Scalar Penta-diagonal solver (SP), and Lower-Upper Gauss-Seidel solver (LU). [12] We use the MPI version of the NPB.

The NPB defines six problem sizes (W/A/B/C/D/E) to simulate varied demands for computation and execution time of different workloads. Problem sizes D and E consume excessive memory and are not intended for single servers, so we omit these. Moreover, problem size W is extremely small and the execution time is short, so it is also omitted from this study. The NPB has limitations for the number of processes.

IV. MOTIVATION

In this section, we analyze the limitations of the SPEC Power and Green500, using the NPB as a control.

We regroup programs to find a suite that can be used as benchmarks in the field of HPC, thus achieving a comprehensive evaluation of the relationship between performance and power, and providing a set of unified evaluation criteria for the effectiveness of different systems.

A. SPEC Power

HPC programs are usually scientific computing programs, such as HPL. The main difference between SPECpower_ssj2008 and scientific computing programs is that the latter usually maintain a high memory usage rate or high CPU usage rate. However, from Figure 1 which presents the results of the server Xeon-E5462, the variation of workload sizes of SPECpower_ssj2008 has little effect on the memory utilization, and the memory utilization remains at a low level of less than 14%. We only present results from the server Xeon-E5462 because we obtained similar results for the other servers.

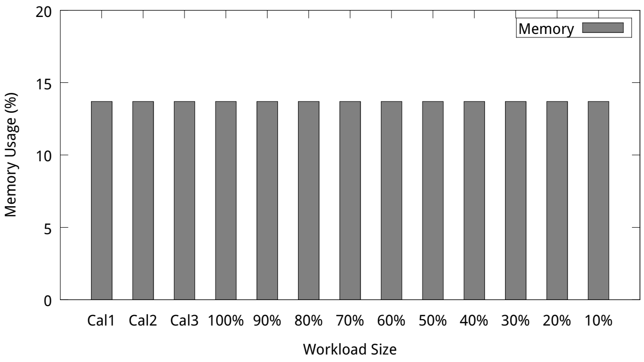


Fig. 1. Memory Usage Test for SPECpower_ssj2008 on Server Xeon-E5462.

Decreases in workload size for each core show a corresponding reduction of CPU utilizations in Figure 2. These kinds of CPU performance characteristics are vastly different for scientific programs. Scientific computing programs generally have high CPU utilizations in different workloads and have efficient data operations. Moreover, SPECpower_ssj2008 is written in Java which needs to be compiled into intermediate code and then interpreted by the virtual machine. Therefore, SPECpower_ssj2008 cannot take advantage of the maximum computing performance of the system.

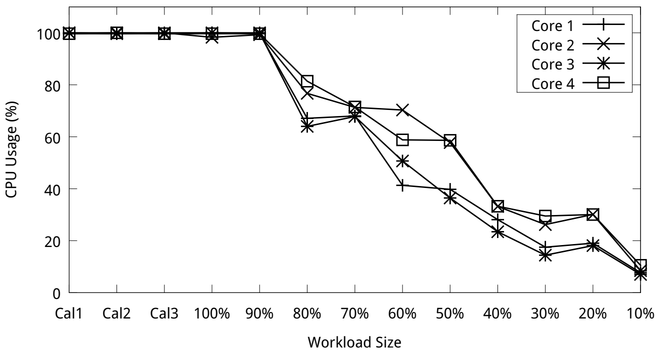


Fig. 2. CPU Usage Test for SPECpower_ssj2008 on Server Xeon-E5462.

B. Green500

The Green500 uses the Linpack, which is a standard scientific computing program. We show the Green500 results along with the NPB results for three servers in Section IV-C.

C. NPB

The results of the SPECpower_ssj2008, HPL and NPB (C scale) from the server Xeon-E5462 are shown in Figure 3 when the number of processes equals four, two, and one. CG.C.2 and CG.C.4 cannot run because the memory required is beyond the maximum memory of the server. The figure shows the power characteristics for each program are different. Among them, HPL reaches the maximum power when the number of processes is four and the single process EP has the lowest power. Under the premise of equal number of cores, EP always has the lowest power and HPL has the highest power when the number of processes is four and two. However, HPL does not consume the highest energy when the process number is one.

The results from the server Opteron-8347 are similar to the results from the server Xeon-E5462 and are shown in Figure 4. When the process number is 16, HPL reaches the highest power. EP has the lowest power in most cases. HPL has the highest growing speed when we increase the number of processes. In contrast, EP has the lowest growing speed.

We show the results from the server Xeon-4870 in Table II. We change the process number from 1 to 40 on the server Xeon-4870 and note that only EP works on all configurations of process numbers.

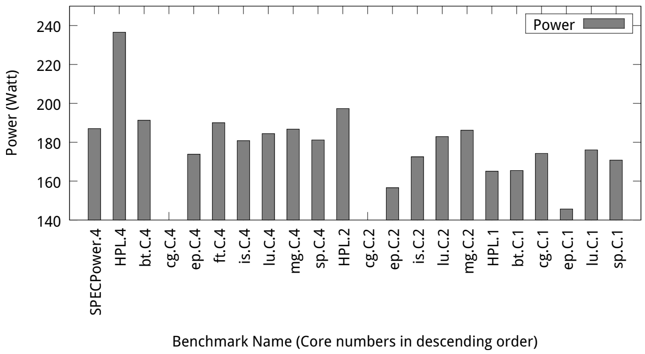


Fig. 3. Power Test on Server Xeon-E5462.

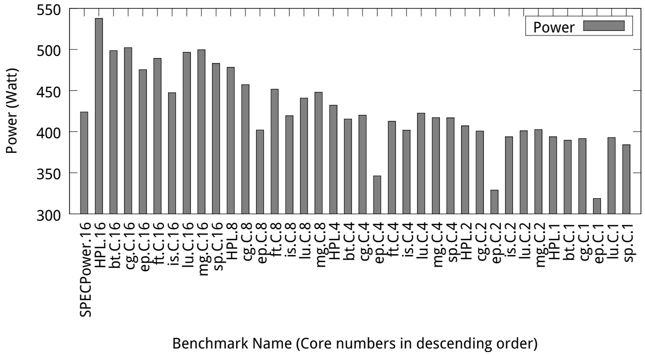


Fig. 4. Power Test on Server Opteron-8347.

D. Limitation

The SPECpower_ssj2008, HPL and NPB each exhibit different power characteristics under different circumstances and thus are mutually irreplaceable. For HPC programs, the number of cores used in the test should be configurable, and this requirement is unable to be met except by EP in the NPB. Therefore, there is a need to design a new set of benchmarks for the HPC field, and we can use the following findings to assist in its construction.

(1) With the growth of the process number, the power of HPL increases dramatically. The power of HPL is close to the highest power in other programs.
(2) With the growth of the process number, the power of EP has the rate of growth, and the power of EP is close to the lowest power in other programs.
(3) HPL and EP suit a wide range of core numbers, which satisfies the need for different cores among the HPC industry. Other programs are unable to do this.
(4) With the same process number, the powers of programs are covered in the range between the powers of EP and HPL.

TABLE II
POWER TEST ON SERVER XEON-4870.

Process Number	HPL	BT	EP	FT	IS	LU	MG	SP	SPEC Power
1	0.45	0.45	0.44			0.45		0.46	
2			0.46		0.47	0.47	0.48		
4	0.50	0.51	0.48	0.51	0.51	0.50	0.52	0.52	
8	0.55		0.49	0.53	0.55	0.52	0.56		
9	0.56	0.54	0.51					0.56	
16	0.61	0.57	0.52	0.58	0.56	0.57	0.62	0.61	
25	0.68	0.64	0.43					0.67	
32	0.71		0.56	0.68	0.60	0.66	0.71		
36	0.72	0.70	0.58					0.74	
39	0.73		0.59						
40	0.74		0.60						0.87

V. POWER EVALUATION IN HPC

The HPL and NPB are common test procedures in HPC. Since HPL has many parameters, we first analyze the HPL configuration parameters to determine the significantly effective parameters. Second, we analyze the relationship between power and NPB problem sizes, and provide an evaluation method for single multi-core servers that is better than the SPEC Power and Green500. Finally, we analyze and compare the evaluation method on three servers.

A. HPL Analysis

HPL uses a fixed workload of calculation to test the peak floating point performance of the system. HPL has many parameters and we need to perform parameter tuning before performance evaluation to ensure peak performance. This is mainly affected by the HPL parameters of the problem sizes (Ns), the data block sizes of LU decomposition (NBs), and the processor mesh size (decided by P and Q). We conduct the test on all three servers. As the results had similar characteristics, we only present the results of the server Xeon-E5462.

1) Ns: Different Ns correspond to different memory usage. We vary the Ns to obtain the relationship between memory usage and power of HPL in Figure 5. The number of cores has a decisive relationship with the power, but the impact of memory utilization to power is limited. This figure also shows the optimization space for memory design.

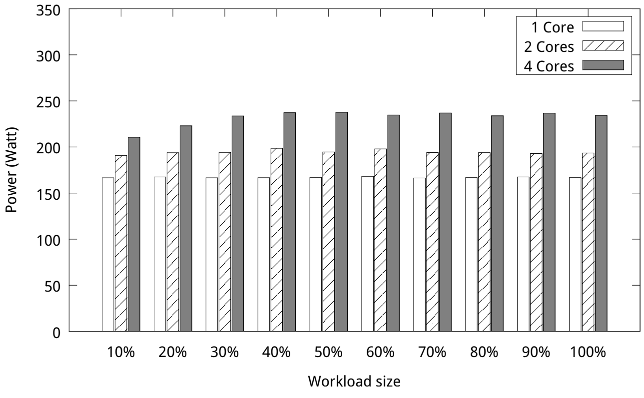


Fig. 5. Ns influence on Server Xeon-E5462.

2) NBs: The variation of NBs has a minimal influence on the power when we fix N, P, and Q in Figure 6. The power of HPL is mainly influenced by the number of cores involved in computing because the power curves of different numbers of cores in Figure 6 do not intersect.

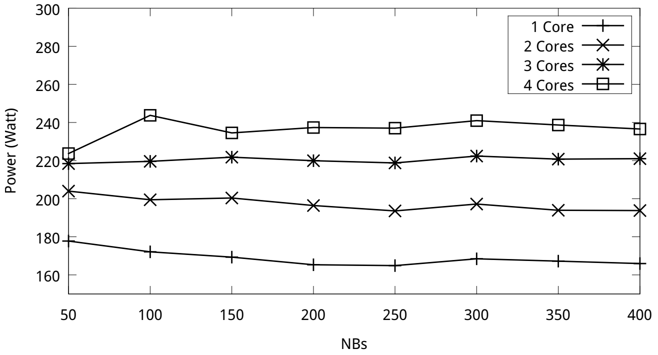


Fig. 6. NBs influence on Server Xeon-E5462.

3) P and Q: When NB is relatively small, P and Q affect the power. The power when NB equals 50 is 10W smaller than the power with other NBs. The combination of P and Q affects power minimally for most cases. Overall, P, Q, and NBs have little influence on power with the majority of power values are in the range from 230W to 245W. In our test, N is 30,000. NBs equal 50/100/150/200/250/300/350/400. We also change the P and Q using 1x4, 2x2, and 4x1 in Figure 7.

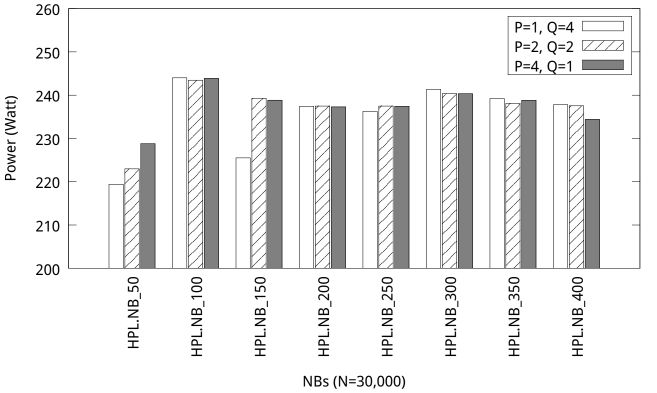


Fig. 7. P and Q influences on Server Xeon-E5462.

4) Conclusion: Although HPL has many parameters, the most significant impact on power is the number of processes. The combination of other parameters also affects the power, but their influence is negligible compared to the influence of the process number. We can provide a simplified method for the HPL power test to reduce the complexity of the performance-power relationship through this phenomenon. In later sections, we first perform parameter tuning and then perform the power test based on this phenomenon.

B. NPB Analysis

Although the NPB exhibits the performance of general applications, using the NPB as a power load model in HPC has the following problems. First, the NPB is not as effective as HPL with regard to floating point peak performance. Second, the NPB has eight sub-tests. The power feature similarities of the eight programs need further study. Third, programs in the NPB, except EP, have special requirements for the number of processes. Therefore, using all programs in the NPB has limitations. We analyze the scale of A/B/C and focus on EP. We only present the results of the server Xeon-E5462 since the other servers gave similar results.

1) Scale: Figure 8 shows the influence of different problem scales. Memory usage is decided by the problem scale, regardless of the number of processes. FT consumes the largest memory footprint under the same circumstances. The memory usage of FT has the highest rate of growth with increasing scales of workloads. EP occupies the minimal memory and the memory footprint of EP has the slowest growth rate with increasing scales of workloads. CG cannot run on this system, but we list it for completeness.

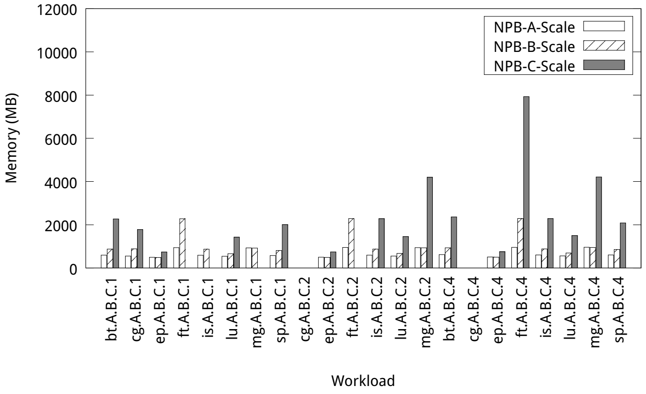


Fig. 8. Memory Usage for A/B/C Scales on Server Xeon-E5462.

The power values in different scales are shown in Figure 9. Power values do not increase with the memory usage significantly, such as FT. In the circumstance that the system uses the same number of cores, EP always has the minimum power than other programs have. The NPB has similar results to HPL, and the power increases along with the number of cores. In this performance test, some of the programs finish quickly due to the small scale of A. For example, the duration of LU.A.2 and MG.A.2 are 1.01s and 2.45s, respectively. The stability and accuracy are difficult to maintain.

2) EP: EP has the lowest memory usage and power among the NPB. We analyze the power characteristics for EP separately because it has no special requirement for the number of cores. Under the C scale, the power and PPW (Performance Per Watt) increase with the number of processes. The energy formula is

$$Energy(KJ) = Power(KWatt) * Time(Second) \quad (2)$$

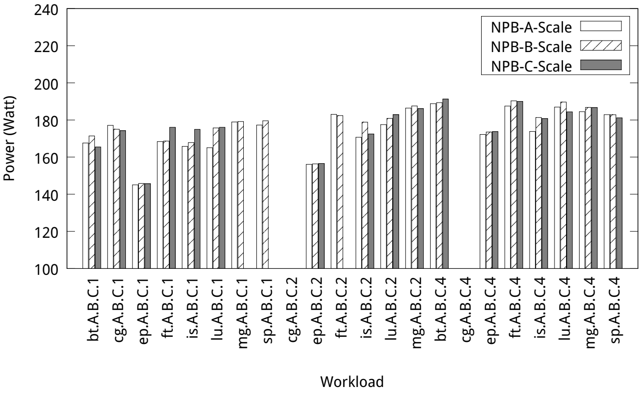
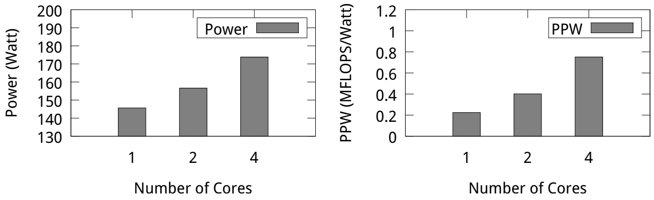


Fig. 9. Power Usage for A/B/C Scales on Server Xeon-E5462.



(a) EP Power (b) EP-PPW

Fig. 10. Power Profiling for EP.

We show the results of power and PPW in Figure 10 and the results of the energy calculations in Figure 11. Both power and PPW increase, but the increase in PPW plays a key role. Therefore, the energy decreases because of the time decrease. Multiple cores reduce the total energy consumption of a calculation, given the same scale of problem. Therefore, improving the parallelism can not only improve the computing performance, but also reduce energy consumption. Both performance and energy consumption are improved by increasing the degree of parallelism.

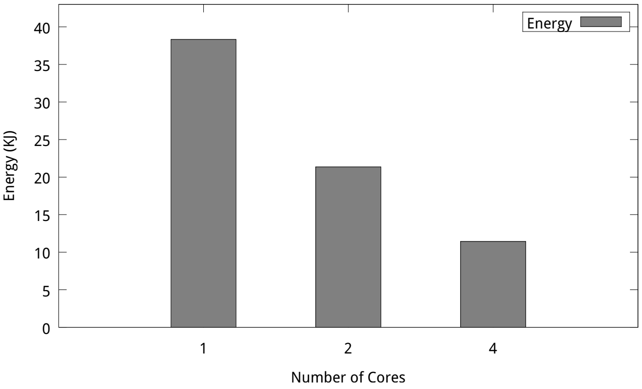


Fig. 11. Energy Analysis for EP.

C. Power Evaluation Design

Based on the previous power tests, we propose a power evaluation method in this section.

1) Evaluation Indicator: According to the analysis of previous experiments in this paper, the workload and power relationship PPW is mainly influenced by the number of cores. The variation of memory utilization has little influence on power, which indicates that the systems do not perform power optimization for memory utilization. In the case that some part of memory is not utilized by the system, the unused component is still in a high power state. However, the situation of high idle power characteristics of memory will be improved with new manufacturing processes. We still consider the memory usage as an evaluation indicator for power evaluation to support the development of memory technologies. Evaluation results of benchmark programs need to reflect the performance indicators of systems to exhibit the power influence of memory usage and the number of cores involved in computation. Evaluation results of benchmark programs need to reflect the following performance indicators:

(1) No-load power
(2) PPW for full CPU usage and full memory usage
(3) PPW for full CPU usage and half memory usage
(4) PPW for half CPU usage and full memory usage
(5) PPW for half CPU usage and half memory usage

These indicators draw on program design experience of load scale changes in SPECpower_ssj2008. We change not only the memory utilization, but also the number of cores. This change can serve the purpose of measuring the load and power relationship, and it reflects the combination of power and performance better than the Linpack test does.

2) Methodology: We choose the HPL and NPB-EP.C as evaluation procedures based on the performance indicators developed by Section V-C1. The number of cores of HPL and NPB-EP.C meet the requirement of flexible configurability. Changing the Ns parameter can support different memory utilizations for HPL. We can simulate the power features for other HPC programs by configuring HPL as the state of half CPU usage or half memory usage. We select the C scale in EP mainly due to its stable measurement time. The test method is indicated in Table III. We use an external power meter WT210 in our test and we do not control the temperature.

TABLE III
TEST METHOD.

Program	Number of Core	Memory Usage
Idle	0	0
NPB-EP.C	1/half/full	C Scale
HPL	1/half/full	50%,90% - 100%

The test procedure is as follows:

(1) Share the power data directory on a PC.
(2) Mount the shared directory to the test server.
(3) Synchronize the clock of the server and the PC.
(4) Start WTViewer (WT210 PC client software) to record power data on the PC.
(5) Start test programs on the server.
(6) Start the NPB-EP.C and HPL according to the configuration files. The configuration follows the rules in Section V-C1.
(7) Acquire the memory information at regular intervals (1s) during the test.

After the test, the program automatically performs the following procedure for data analysis:

(1) Copy CSV files which contain power data to the server and merge them into one file.
(2) Extract the power information for each program according to the execution time.
(3) Analyze the memory and power for each program, and remove the initial 10% data and the final 10% data.
(4) Obtain the arithmetic average power and arithmetic average memory usage.
(5) Divide the average performance (GFLOPS) by the average power (Watt) to obtain the PPW for each program.
(6) Calculate the arithmetic average for PPWs to provide the PPW result for the system.

3) Power Evaluation Result: We perform the power evaluation for Server Xeon-E5462, Opteron-8347, and Xeon-4870. The power evaluation result of the servers Xeon-E5462, Opteron-8347 and Xeon-4870 are shown in Tables IV, V and VI, respectively.

TABLE IV
PPW ON SERVER XEON-E5462.

Program	Performance (GFLOPS)	Power (Watt)	PPW (GFLOPS/Watt)
Idle	0.0000	134.3727	0.0000
ep.C.1	0.0319	145.4889	0.0002
ep.C.2	0.0638	156.9150	0.0004
ep.C.4	0.1237	174.0141	0.0007
HPL_P1_Mh	10.5000	168.4366	0.0623
HPL_P2_Mh	20.2000	203.8387	0.0991
HPL_P4_Mh	36.1000	231.3697	0.1560
HPL_P1_Mf	10.6000	168.1937	0.0630
HPL_P2_Mf	20.3000	204.9486	0.0990
HPL_P4_Mf	37.2000	235.3179	0.1580
Average	13.5000	182.2896	
(GFlops/Watt)/10			**0.6390**

TABLE V
PPW ON SERVER OPTERON-8347.

Program	Performance (GFLOPS)	Power (Watt)	PPW (GFLOPS/Watt)
Idle	0.0000	311.5214	0.0000
ep.C.1	0.0126	392.6666	0.0000
ep.C.4	0.0836	427.6455	0.0002
ep.C.8	0.1394	476.9047	0.0003
HPL_P1_Mh	3.8900	408.8880	0.0095
HPL_P8_Mh	26.3000	485.6727	0.0542
HPL_P16_Mh	32.0000	535.5574	0.0598
HPL_P1_Mf	3.9500	412.7283	0.0096
HPL_P8_Mf	27.1000	484.0001	0.0560
HPL_P16_Mf	32.7000	529.5337	0.0618
Average	12.6000	446.5118	
(GFlops/Watt)/10			**0.0251**

The result of the power evaluation for the three servers is as follows (Greater than sign indicates better result):

$$XeonE5462(\mathbf{0.639}) > Xeon4870(\mathbf{0.0975}) > Opteron8347(\mathbf{0.0251})$$

TABLE VI
PPW ON SERVER XEON-4870.

Program	Performance (GFLOPS)	Power (Watt)	PPW (GFLOPS/Watt)
Idle	0.0000	642.2300	0.0000
ep.C.1	0.0187	667.2800	0.0000
ep.C.20	0.3400	706.7800	0.0005
ep.C.40	0.7590	730.9800	0.0010
HPL_P1_Mh	8.9100	676.1600	0.0132
HPL_P20_Mh	162.0000	963.8000	0.1680
HPL_P40_Mh	339.0000	1118.5400	0.3030
HPL_P1_Mf	8.0800	676.3700	0.0119
HPL_P20_Mf	164.0000	965.2900	0.1700
HPL_P40_Mf	344.0000	1119.6000	0.3070
Average	103.0000	826.7030	
(GFlops/Watt)/10			**0.0975**

However, when we use the evaluation method of HPL peak Performance Per Watt, the conclusion is different. GREEN500 uses this method and the result is as follows:

$$Xeon4870(\underline{0.307}) > XeonE5462(\underline{0.158}) > Opteron8347(\underline{0.0618})$$

Our evaluation is different from the above conclusion, which implies that the peak condition does not represent the overall performance or power characteristics. Our evaluation includes load measurements. As there is no operation with no load, the PPW is 0. Therefore, load consumption is reflected in the average system consumption, and is not reflected in the final Performance Per Watt result.

We also perform SPECpower_ssj2008 for the servers and the result is as follows:

$$XeonE5462(247) > Xeon4870(139) > Opteron8347(22.2)$$

Although our evaluation conclusion is the same as the evaluation conclusion of SPECpower_ssj2008, they reflect different performance and power relationships. Our evaluation result demonstrates the performance and power relations (GFLOPS/Watt) in HPC. The SPECpower_ssj2008 result provides the system performance and power relations (ssj_ops/Watt) when the servers are used as data center servers.

Workloads in the HPC field have their own characteristics. An evaluation method can only appropriately reflect the performance and power relationship if it is based on an HPC workload.

VI. POWER MODEL

We study the relationship of computing characteristics, memory access features, and power of HPC programs in this section. We construct a power model for single multi-core servers, and study the relationship between workload characteristics and power through multiple linear regression analysis.

A. Multiple Linear Regression Model for Power

Multiple linear regression models generally use historical samples as inputs, and obtain regression coefficients by solving linear equations. These regression equation coefficients are used to build up regression equations. We pass newly collected samples into the regression equation whose coefficient is determined. We use the forward stepwise [9] to forecast the target y.

1) Multiple Linear Regression Model: We can divide the power for servers as follows, where the total power equals the sum of powers for each part.

$$P_{Total} = P_{CPU} + P_{Mem} + P_{Motherboard} + P_{Harddrive} + P_{Mouse} + \ldots \tag{3}$$

We consider the motherboard, hard drives, fans, mouse and other peripherals substantially independent of workload changes. The main indicators of PPW are memory usage and the number of used cores, which are also decisive factors affecting the performance in HPC. Therefore, we further simplify the power model as follows. The total system power equals to the sum of processor power, memory power, and a constant C.

$$P_{Total} = P_{CPU} + P_{Mem} + C \tag{4}$$

2) HPCC Power Regression Model: The HPCC benchmark [11] consists of seven types of procedures, and they are selected to evaluate workloads which are compute-intensive, memory access intensive, network transmission intensive, and so forth. We use the Performance Monitoring Unit (PMU) of processors [5] to collect data that reflects the system state. We divide program executions into instruction executions, cache accesses, and memory accesses. The power model should cover a wide variety of load characteristics in the HPC field to obtain general relations of HPC workload and power characteristics. The HPCC load characteristics satisfy this requirement. We use the forward stepwise to choose the following indexes. We assume that other indices have little impact on power, and use R Square [17] to test for their influences.

(1) X_1: WorkingCoreNum
(2) X_2: InstructionNum
(3) X_3: L2CacheHit
(4) X_4: L3CacheHit
(5) X_5: MemoryReadTimes
(6) X_6: MemoryWriteTimes

We obtain the further developed equation:

$$P_{Total} \approx b_1 * X_1 + b_2 * X_2 + \ldots + b_6 * X_6 + C \tag{5}$$

We use a similar method in Section V-C2 to obtain and process the power data. Test scripts sequentially start the seven HPCC programs from single core to full cores. We collect PMU data according to a certain interval (10s) during the execution. We integrate the PMU data with the average power data according to the time stamp and perform normalization to unify the dimensions of different variables.

B. Results on Server Xeon-4870

We perform the experiment on Server Xeon-4870 and show the regression result in Table VII. R Square is close to 1, which shows strong correlation between the selected indicator and power.

TABLE VII
REGRESSION RESULT ON SERVER XEON-4870.

Name	Value
Multiple R	0.969706539
R Square	**0.940330771**
Adjusted R Square	0.940271585
Standard Error	0.244393975
Observation	6056

We calculate the regression coefficient and obtain the indices for the regression equation. The index values are shown in Table VIII. The values of b_1 and b_2 are high, which indicates the number of used cores and executed instructions are more influential than other indices.

C. Regression Results Verification

We use NPB scales of B and C to perform regression verification. Since the results are similar, we only provide the results of scale B. We show the measured values and regression values in Figure 12. Although there is a gap at some points, the regression waveform has a considerable degree of predictability. This predictability shows that the power is closely related to the real-time load characteristics and memory access. The reason for the points with a large gap is that the power may have different sensitivity with different load characteristics, and these related features may not be totally included in the indicators selected in Section VI-A2.

The measured value minus the regression value equals the difference which represents the fitting degree. The closer the difference is to zero, the better the prediction is. The results of these calculations are shown in Figure 13.

We also use the following fit formulas to obtain the fit for measured values and regression values. x_i denotes measured value and $\tilde{x}_i$ denotes regression value. $RSS(x)$ denotes the residual sum of square and $TSS(x)$ denotes the total variation. R^2 denotes the fitting coefficient of determination. The R^2 for NPB-B is about 0.634 and the R^2 for NPB-C is about 0.543, which demonstrates the similarity between the predicted and the measured is greater than 50%. EP and SP have unsatisfactory results and this relates to their programming characteristics. EP essentially has no communication while SP has the most communication. We can combine EP and SP into the training set to reinforce the load forecast for the regression equation.

$$R^2 = 1 - \frac{RSS(x)}{TSS(x)} \tag{6}$$

$$RSS(x) = \sum_{i=1}^{n}(x_i - \tilde{x}_i)^2 \tag{7}$$

$$TSS(x) = \sum_{i=1}^{n}(x_i - \frac{1}{n}\sum_{i=1}^{n} x_i)^2 \tag{8}$$

VII. RELATED WORK

The SPEC Power provides a method that associates performance with power and is the first industry standard benchmark. Wang et al. [21] used the SPEC Power benchmark to study computing on green data centers, but not for HPC workloads. Ryckbosch et al. [18] used SPECPower_ssj2008 to prove that more energy can be saved. In our paper, we use other programs as our workload and do not adhere to the SPEC Power.

The Green500 is another well established standard for performance and power. Feng et al. [13] discussed improving energy efficiency on supercomputers. Ge et al. [14] show the measuring tutorial for the Green500. Subramaniam et al. [20] showed how power measurement techniques can be applied to supercomputers. Our work uses HPL as part of the evaluation method and we focus on single servers.

Singh et al. [19] used Performance Monitoring Counters to estimate power via analytic models. Contreras et al. [10] provided a power prediction model for Intel XScale Processors. The models are different, and we perform a simplification to reduce the difficulty. Moreover, we use HPCC to train our model and use NPB to validate it.

VIII. CONCLUSION

The field of high-performance computing (HPC) faces the challenges of both speed and energy-savings. Performance per power has become the standard for hardware platforms and software environments. The SPEC Power benchmark and Green500 are two mainstream evaluation benchmarks, but both have limitations. In this paper, we showed that the two benchmarks cannot represent the characteristics for general HPC programs and demonstrated a need for a new power benchmark which better represents the HPC programs.

The two main influencing factors for the power and performance relationship are the number of cores in computation and memory usage. We proposed a new power evaluation method that involved shielding the tuning for HPL parameters and simplifying the problem where the NPB has multiple programs, which reduced the power evaluation complexity. Our power evaluation method showed different power characteristics from the Green500 and the SPEC Power benchmark. We examined our evaluation method on three HPC servers and provided the performance per power indicators for general HPC programs.

We proposed a power model to support further research and provided a theoretical basis for power evaluation. We used a multiple linear regression method to confirm that the power of single multi-core servers can be predicted by the characteristics of load operations.

IX. ACKNOWLEDGEMENT

This paper was supported by the National High Technology Research and Development Program of China (2012AA012602) and Natural Science Foundation of China project (61472201).

TABLE VIII
INDEX ON SERVER XEON-4870.

Index	b_1	b_2	b_3	b_4	b_5	b_6	C
Value	0.121595997	0.836925677	-0.008648267	-0.007731074	0.087493111	-0.070519444	2.37E-14

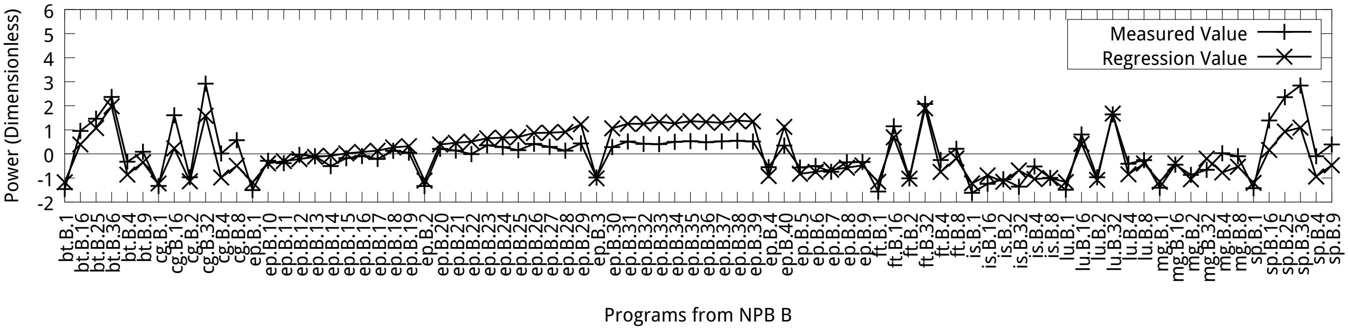


Fig. 12. Regression Results.

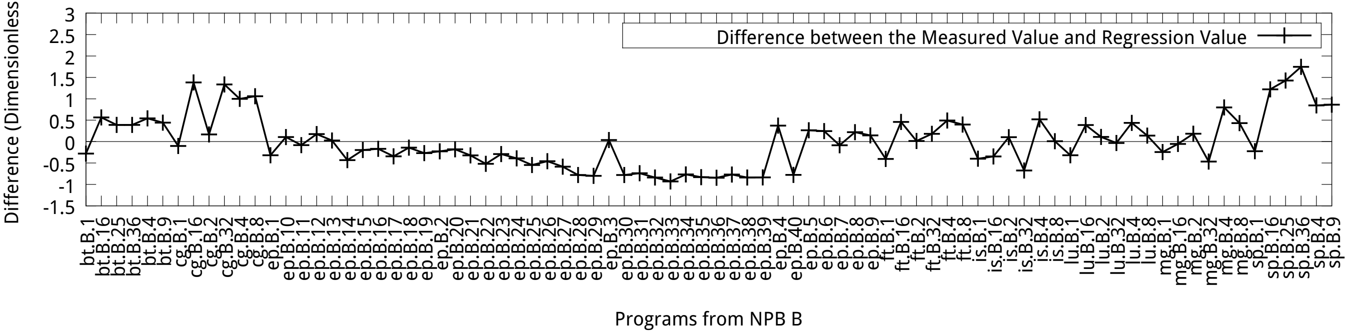


Fig. 13. Difference Results.

REFERENCES

[1] AMD Opteron 8347 specifications. http://www.cpu-world.com/CPUs/K10/AMD-Third%20Generation%20Opteron%208347%20-%20OS8347WAL4BGE.html.

[2] HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers. http://www.netlib.org/benchmark/hpl/.

[3] Intel Xeon Processor E5462. http://ark.intel.com/products/33084/Intel-Xeon-Processor-E5462-12M-Cache-2_80-GHz-1600-MHz-FSB.

[4] Intel Xeon Processor E7-4870. http://ark.intel.com/products/53579/Intel-Xeon-Processor-E7-4870-30M-Cache-2_40-GHz-6_40-GTs-Intel-QPI.

[5] PMU Performance Monitoring PerfMon. https://software.intel.com/en-us/tags/18842.

[6] SPECpower_ssj 2008. https://www.spec.org/power_ssj2008/.

[7] The GREEN500. http://www.green500.org/.

[8] D. H. Bailey, E. Barszcz, J. T. Barton, D. S. Browning, R. L. Carter, L. Dagum, R. A. Fatoohi, P. O. Frederickson, T. A. Lasinski, R. S. Schreiber, et al. The NAS parallel benchmarks. *International Journal of High Performance Computing Applications*, 5(3):63–73, 1991.

[9] R. B. Bendel and A. A. Afifi. Comparison of stopping rules in forward stepwise regression. *Journal of the American Statistical Association*, 72(357):46–53, 1977.

[10] G. Contreras and M. Martonosi. Power prediction for Intel XScale® processors using performance monitoring unit events. In *Low Power Electronics and Design, 2005. ISLPED'05. Proceedings of the 2005 International Symposium on*, pages 221–226. IEEE, 2005.

[11] J. Dongarra and P. Luszczek. HPC challenge benchmark. In *Encyclopedia of Parallel Computing*, pages 844–850. Springer, 2011.

[12] A. Faraj and X. Yuan. Communication characteristics in the NAS parallel benchmarks. In *IASTED PDCS*, pages 724–729, 2002.

[13] W.-c. Feng and K. W. Cameron. The green500 list: Encouraging sustainable supercomputing. *Computer*, 40(12):50–55, 2007.

[14] R. Ge, X. Feng, H. Pyla, K. Cameron, and W. Feng. Power measurement tutorial for the Green500 list. *The Green500 List: Environmentally Responsible Supercomputing*, 2007.

[15] J. Humphreys and J. Scaramella. The Impact of Power and Cooling on Data Center Infrastructure. IDC, 2006.

[16] K.-D. Lange. Identifying Shades of Green: The SPECpower Benchmarks. *IEEE Computer*, 42(3):95–97, 2009.

[17] N. J. Nagelkerke. A note on a general definition of the coefficient of determination. *Biometrika*, 78(3):691–692, 1991.

[18] F. Ryckbosch, S. Polfliet, and L. Eeckhout. Trends in server energy proportionality. *Computer*, 44(9):69–72, 2011.

[19] K. Singh, M. Bhadauria, and S. A. McKee. Real time power estimation and thread scheduling via performance counters. *ACM SIGARCH Computer Architecture News*, 37(2):46–55, 2009.

[20] B. Subramaniam and W.-c. Feng. Understanding power measurement implications in the green500 list. In *Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom)*, pages 245–251. IEEE, 2010.

[21] L. Wang and S. U. Khan. Review of performance metrics for green data centers: a taxonomy study. *The Journal of Supercomputing*, 63(3):639–656, 2013.