Quantifying the Impact of Variability and Heterogeneity on the Energy Efficiency for a Next-Generation Ultra-Green Supercomputer

Quantifying the Impact of Variability and Heterogeneity on the Energy Efficiency for a Next-Generation Ultra-Green Supercomputer

Francesco Fraternali, Andrea Bartolini, Carlo Cavazzoni, Luca Benini, *Fellow, IEEE*

Abstract—Supercomputers, nowadays, aggregate a large number of nodes featuring the same nominal HW components (eg. processors and GPGPUS). In real-life machines, the chips populating each node are subject to a wide range of variability sources, related to performance and temperature operating points (i.e. ACPI p-states) as well as process variations and die binning. Eurora is a fully operational supercomputer prototype that topped July 2013 Green500 and it represents a unique 'living lab' for next-generation ultra-green supercomputers. In this paper we evaluate and quantify the impact of variability on Eurora's energy-performance tradeoffs under a wide range of workloads intensity. Our experiments demonstrate that variability comes from hardware component mismatches as well as from the interplay between run-time energy management and workload variations. Thus, variability has a significant impact on energy efficiency even at the moderate scale of the Eurora machine, thereby substantiating the critical importance of variability management in future green supercomputers.

Index Terms—Green500, High-Performance Computing, Hardware Variability, Energy-Efficient Software Design, Energy-Aware Computing, Green Supercomputer, Heterogeneous Supercomputer, Dynamic Resource Management, Hardware Accelerator, DVFS.

1 Introduction

While integrated computing architectures are facing Power/Thermal/Utilization walls that are limiting the performance benefits of technology scaling, the demand for more powerful supercomputers continues to increase. TOP500 rankings in the last twenty years show an exponential growth of peak performance that is predicted to enter the ExaFLOPS (10^{18}) at the latest by 2023 [2]. Today's most powerful supercomputer, TaihuLight, reaches 93.01 PetaFLOPS with 15.37 MW of power dissipation whitout event considering the cooling infrastructure. This data shows that exascale supercomputers cannot be built by simply expanding the number of processing nodes and leveraging technology scaling, as power demand would increase unsustainably (hundreds of MW of power). According to [4], an acceptable value for an Exascale supercomputer is 20 MW. To reach this target, current supercomputer systems must achieve an energy efficiency "quantum leap", pushing towards a goal of 50 GFLOPS/W. With the aim to push supercomputers to improve energy efficiency, the Green500 list ranks Top500 supercomputers by their energy efficiency [3]. In contrast with TOP500, the Green500 list looks into an energy efficiency metric, the GFLOPS per Watt (GOPS/W), for computers "big enough" to be considered supercomputer-class, i.e. passing the threshold of being part of Top500. From the Green500 perspective, the current fastest supercomputer (TaihuLight) delivers only 6.05 GFLOPS/W. It is clear that the design of future exascale machines must take energy efficiency as a primary design goal and address the challenges coming from power bounds in direct manner, not just as an afterthought. Variability is one of the key challenges that must be addressed when designing a green supercomputer. This is especially true because energy efficiency boosters like voltage scaling and aggressive power management have a quite dramatic impact on variability [5], [7]. Process variation is the deviation of transistor parameters from their nominal design values, which is caused by both systematic (e.g., lithographic inconsistencies) and random effects (e.g., varying dopant concentrations) [25]. Variation effects have been proven to worsen with the process scaling [7]. After silicon fabrication, providers use speed binning [6] to cluster in the same product family devices that share similar performance and silicon quality and to sell them with the same nominal speed. Even if this mitigates the end-user product variability, in a supercomputer that includes thousands of CPUs of the same bin the effect of process variability can become relevant. In addition to process variability, the same device can operate at different frequency and voltage levels (DVFS, ACPI states) [9]. The Linux operating system does this by mean of SW governors. The default one is called "on-demand" and it adapts the frequency to the CPU load [10]. In Intel machine this governor is called "intel_pstate" and has a similar behavior. Even more than with just advanced fabrication technologies, green supercomputers will achieve high energy-efficiency (GFLOPS/Watts) at the architectural level, by exploiting HW heterogeneity as they embeds in the computing node parallel accelerators. Looking at the top 20 most energy efficient supercomputers according to the June 2016 green500 list, it can be observed that 19 over

- *F. Fraternali, A. Bartolini and L.Benini are with the Department of Electrical, Electronic and Information Engineering (DEI), University of Bologna, Italy.*
  E-mail: {francesco.fraternali, a.bartolini, luca.benini}@unibo.it
- *A. Bartolini and L. Benini are with the Integrated Systems Laboratory, ETH Zurich, Switzerland. E-mail: {barandre, lbenini}@iis.ee.ethz.ch*
- *Carlo Cavazzoni is with SCAI, CINECA, Italy. E-mail: c.cavazzoni@cineca.it*

Manuscript received April 19, 2005; revised September 17, 2014.

20 supercomputers use a heterogeneous design composed by a data-parallel accelerator (NVIDIA GPU, AMD Firepro, PEZY-SCnp), while one (TaihuLight) embeds as processing core a custom designed many-core CPU based on RISC cores. In addition HW accelerators have the capability of changing operating state to trade-off performance with power. This opens interesting design points when a hybrid workload runs on heterogeneous HW. More than 86% of TOP500 supercomputers are based on a scalable architecture where a "node" is replicated many times and in almost 90% of today's supercomputers the node embeds x86 CPUs. Moreover 80% of nodes use Intel Xeon E5 series (36.8% SandyBridge, 15.2% IvyBridge and 28% Haswell) components. Almost the entire TOP500 supercomputers (94.4%) use Linux O.S. Finally supercomputers run a wide variety of workloads and scientific computational kernels and thus are affected by software variability which can impact the overall energy-performance trade-off.

The Eurora Supercomputer prototype, developed by Eurotech and Cineca [16] has ranked first in the Green500 list in July 2013, achieving 3.2 GFLOPS/W on the Linpack Benchmark with a peak power consumption of 30.7 KW, and improving by almost 30% the performance of previous green supercomputers. Eurora has been supported by PRACE 2IP project [20] and it serves as testbed for next generation Tier-0 systems. Its energy efficiency performance is achieved by adopting a heterogeneous architecture and a direct liquid cooling system that enables hot water cooling, that is suitable for hot water recycling and free-cooling solutions [18]. For its characteristics Eurora is a perfect vehicle for testing and characterizing next-generation "greener" supercomputers. As the majority of Supercomputers, Eurora nodes embeds NVIDIA Kepler GPUs and Intel Xeon Phi accelerators, uses Intel Xeon processors with linux O.S.

1.1 Contributions

In this paper we analyze the impact of different variation sources (HW and SW) on Eurora in terms of performance and energy metrics. We show that the whole system has significant optimizations margins and that optimization has a sizable impact at the scale of the entire supercomputer. The main contributions of the paper are:

- We measured up to 15% of energy variation among nodes at the same operating condition and under the same workload. This amount of variability is measured on a relatively small system with just 32 nodes, and thus is expected to increases in larger systems and more advanced technology nodes.
- In real supercomputer applications we measure 27% energy saving w.r.t. the default turbo mode for the CPUs devices and 26% energy saving in the GPUs accelerators w.r.t. the maximum running frequency.
- We quantify that optimal voltage and frequency operating point selection (VFS) can lead to an energy saving ranging between 18% and 50% by using the only CPUs devices and a further 17% energy saving by extending the optimal VFS selection to the GPUs accelerators. These results point out that the vast majority of workloads achieve significantly higher energy efficiency when they do not run at peak performance. Hence, new management strategies for allocating machine resources to workload are needed for energy-constrained supercomputers at the expense of pure performance.
- We measure a further improvement up to 6% in the energy efficiency by applying the optimal VFS on both CPUs and GPUs on hybrid workloads. Hence, we further show that to reduce the energy of the system the highest CPUs and GPUs speed are not the best option and only an application dependent operating point configuration for both GPUs and GPUs gives the best energy efficiency.

1.2 Related Work

In the last decade, variability has been widely studied in computer-architecture, VLSI and EDA fields on the hardware and software viewpoints. Authors in [29], review the literature for reliability and process-variation aware VLSI design to find that the design of reliable circuits with unreliable components is a significant challenge that is likely to remain relevant for all circuit designs from now on. Other industrial players like TI and IBM adopts reliability-aware design methodologies at various stages of the design process [48]. Authors in [28] show that in order to account for parameter variations during the design phase, the designers will endure an average of 11% increase in area. On the software side, process variation has also been deeply investigated and several countermeasures and approaches have been studied and implemented. As an example, authors in [8], [11] show that operating system can be designed to take advantage of process variation to differentiate the peak performance of processing elements while ensuring the same target lifetime of the device.

In addition, Paterna et al. propose an ILP formulation to minimize the energy consumption of a multimedia multicore platform affected by variability [12], [13]. Sharing the same assumption Rudi et al. introduced an ILP formulation which can couple thermal prediction with hardware heterogeneity to optimize the overall system performance under thermal constraints [23]. The techniques presented in [32] rely on the characterization of the power consumption measured by B. Balaji et al in [38] that uses detailed power measurements to show the part to part variability for a variety of representative single-threaded and multithreaded application workloads. Balaji utilizes six Core i5-540M laptop processors and the Linux userspace CPU governor to control four out of a total of ten available frequencies 1.2Ghz (lowest),1.73Ghz, 2.13Ghz and 2.63Ghz (highest). In the results, they measured processor power variation of 7-17% depending on configuration and application between identical processors at the same frequency of operation. [38] shows that commercial multiprocessors are affected by process variability and it is not clear how this will impact the final performance and energy efficiency of a large scale HPC system which integrates a large number of them. Indeed, the exascale system [2] will likely contain hundreds of thousands of nodes and billion-way parallelism and authors in [36], [37] are pointing out the importance of variability modeling for large scale clusters. In particular, they find that inter-node variability in homogeneous clusters leads to different models and for high-fidelity cluster power models, the choice of model predictors will vary from node to node. Indeed, by simply multiplying the power prediction of a single node with the number of nodes in a cluster, it could yield to a worst-case dynamic range errors up to 150%. Increasing system size bring a complementary challenge on power and energy availability and costs, with projected systems expected to consume tens of mega-watts of power [40]. To overcome this problem, Torsten et al. discuss in [39] existing node power variations in two real HPC homogeneous systems. They introduce three energy-saving techniques and quantifies possible savings for each technique. Energy saving results are based on simulation and in the best theoretical case a combined savings of the two best practical techniques show an energy saving of just 0.5%.

Hence, to address the mentioned power and energy efficiency problem, authors in [33] and [34] exploit the variations in manufacturing processes that cause the transistors on each chip to differ resulting in many-core chips being inherently heterogeneous. In-fact, due to process variations, frequency and power consumption profiles of cores can span a wide range and this make optimal scheduling of applications under a power budget computationally difficult. In particular, authors in [33] propose an integer linear programming (ILP) based approach for selecting optimal configuration of a chip showing savings in energy consumption on an average of 26% and 10.7% for two HPC mini-applications. Analogously, [34] proposes a scheduling framework using ILP, which enables efficient scheduling based on the application, the properties of the chip, and power and performance constraints. Results show that their framework finds configurations that are up to 2.5 times faster than the ones obtained from simple heuristics. Both the mentioned works are based on the use of Sniper Multicore Simulator [49] for simulating chips with heterogeneity. Validations of the Sniper simulator [49] against real hardware show average absolute errors within 25% for a variety of multi-threaded workloads. All these works are based on simulation results. A detailed assessment of the impact of variability on power and energy in a real large-scale high-performance computing (HPC) system is currently missing in the open literature. In recent times, the new dominating trend in energy-efficient HPC is toward heterogeneous architectures coupling processors with accelerators (typically GP-GPUs). The importance of heterogeneity at the architectural level is claimed by [35] that stresses the need for architectures that can tolerate application variability without performance loss in a Warehouse-Scale Computer (WSC). Presenting a detailed micro-architectural analysis of live data-center jobs, measured on more than 20,000 Google machines over a three year period, and comprising thousands of different applications, [35] found common low-level functions (datacenter tax), which show potential for specialized hardware in a future server SoC and making heterogeneous architectures also beneficial in the server domain.

To the authors' knowledge the work presented in this paper is the first open study on the impact of variability and heterogeneity on the energy efficiency of a full-scale green supercomputer. Typically, manufacturers are the only owner of such information and it is non-trivial to find variability investigation even on single devices. A similar attempt has been pursued by the author in [45], but the variability has been quantified only on general-purpose CPUs and hence, it is missing the accelerators part to include the whole supercomputer.

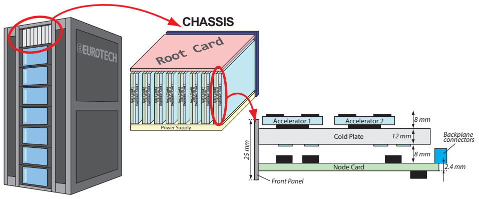


Figure 1. Eurora architecture

In the remainder of the paper, Section 2 presents an overview of the Eurora Platform, Section 3 provides a taxonomy of the variation sources in a supercomputer system and Section 4 shows the workloads and tests for our characterization. Finally, Section 5 shows the results obtained and the conclusions and future work are reported in Section 6.

2 EURORA

The Eurora system consists of a half-rack containing 8 stacked chassis, each of them designed to host 8 node cards and 16 expansion cards (see Fig. 1). The node card is the basic element of the system and consists of 2 Intel Xeon E5 series (SandyBridge) processors and 2 expansion cards configured to host an accelerator module. One half of the nodes use E5-2658 processors including 8 cores with 2.1 GHz clock speed (Max Turbo Frequency 2.8 GHz), 20 MB caches, and 95 W maximum TDP. The rest of the nodes use E5-2687W processors including 8 cores with 3.1 GHz clock speed (Max Turbo Frequency 3.8 GHz), 20 MB caches, and 150 W maximum TDP. The accelerator modules can be Nvidia Tesla (Kepler) with up to 24 GB of GDR5 RAM and up to 2 TFlop peak DP and 250 W TDP, or, alternatively, Intel MIC KNC with up to 16 GB of GDR5 RAM and up to 1.4 TFlop peak DP and 245 W TDP.

Each node of Eurora runs a SMP CentOS Linux distribution version 6.3. The kernel is configured with NOHZ function disabled, hyper threading HW support disabled and on-demand power governor [10]. The linux governor allows users with specific rights to change at run-time the clock frequency of each CPU by writing the target frequency value in the */sys/dev*. The clock frequency of GPUs can be scaled at run-time too by mean of specific APIs of the NVIDIA driver. This mechanism can be exploited to precisely control the frequency of the studied system and it has been used to perform the analysis as described in the following sections.

Eurora interfaces with the world through a dedicated *login node*, physically positioned outside the Eurora rack. This node executes the batch job dispatcher (PBS) [46] and connects to the same shared file system directly accessible from all the computing nodes. In this paper, we encapsulated our tests in PBS jobs so that we can exploit the job dispatcher features to exclusively test all the nodes of Eurora while running in a production environment. We remark that all these settings are quite commonplace for high-performance supercomputers.

Eurora features an integrated and low-overhead monitoring

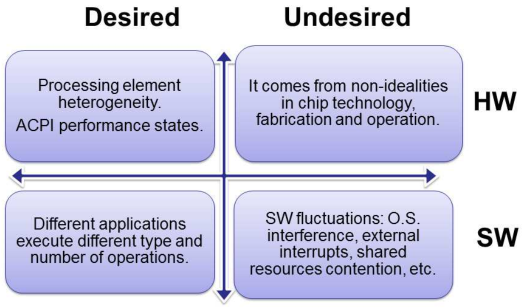

Desired
Undesired
Processing element heterogeneity. ACPI performance states.
It comes from non-idealities in chip technology, fabrication and operation.
HW
Different applications execute different type and number of operations.
SW fluctuations: O.S. interference, external interrupts, shared resources contention, etc.
SW

Figure 2. Relevant Variability Sources in Supercomputers

system made-up by a set of software daemons and parsing scripts. The SW daemons run periodically (every 5 seconds) on each node to collect traces of the processing elements (CPUs, GPUs, Xeon Phy) activity by mean of HW performance counters. For each core, the monitoring system gathers values from the Performance Monitoring Unit[1] as well as the core temperature sensors, and the time-step counter. In addition, for each CPU it gathers the monitoring counters (power unit, core energy, DRAM energy, package energy) present in the Intel Running Average Power Limit (RAPL) interface. The parsing scripts process off-line the raw log of the performance counters to generate performance metrics (CPI, Load, Temperature, Power, etc.) and relate them with the job running on the node.

3 Variability in supercomputers

In this Section we classify variability sources that impact supercomputer performance (i.e. power, execution time, energy). We classify the variability sources from the user-perspective (*desired* vs. *undesired*) and from their nature (*hardware* vs. *software*) as shown in Figure 2.

Desired Hardware Variability
The use of different hardware components (e.g. CPU, GPU) and different operating points (e.g. clock frequency) produces different and user-expected performance. Hence, we label both the heterogeneity in the processing elements and in between the ACPI performance states as *desired hardware variability*. Eurora nodes, similarly to others, are built using different computational units including HW accelerators and different series of the same CPU family (speed binning). In particular, as mentioned in Section 2, Eurora is composed by 32 nodes that use Intel Xeon E5-2658 *with 2 Intel Xeon Phi 5120D,* while the remaining 32 nodes that use Intel Xeon E5-2687 *with 2 Nvidia Tesla K20s GPU*. Furthermore, both the aforementioned CPUs and GPUs devices are capable of scaling their own voltage and frequency by means of O.S. governors and driver.These knobs can be also used by the system admin and by the final user to modulate the energy-performance trade-off. In the Eurora system, the nominal 2.1GHz CPUs can scale their frequencies from 1.2GHz to 2.1GHz with 100MHz step. Instead, the nominal 3.1GHz CPUs can scale their frequencies from 1.2GHz to 3.1GHz with 200MHz step. With regard of CPUs, if the fastest state is selected, then the turbo mode is enabled and the HW can overclock the frequency if this is thermally and power sustainable [15]. To scale the frequency of the GPUs accelerators, both the memory and the graphics clocks can be controlled. In particular, the frequency of the memory clock can be configured to 2600MHz and 324MHz while the graphic clock can range between 758MHz and 324MHz in 6 steps (758, 705, 666, 640, 614 and 324MHz). The graphic clock can be configured to the lower value only if the lower level of the memory clock is selected as well.

Undesired Hardware Variability
By executing the same benchmark on the same family hardware-nodes, the presence of undesired hardware variability determines different performance for each node. Hence, this class groups all the variability sources that come from non-idealities in chip technology, fabrication and operation. Due to process variation, ambient conditions and manufacturing variability, different instances of the same nominal device operate at different PVT points and ambient conditions. This may lead to observe different power and temperature values for different devices (CPUs and GPUs) of the same family while executing the same workload at the same operating point.

Desired Software Variability
This class accounts for the fact that different applications execute different type and number of operations and have a different usage of resources. This may reflect in a variable performance and energy consumption and different sensitivity to hardware variability sources.

Undesired Software Variability
This class accounts for all the software fluctuations that introduce variations in energy and performance of the same code that runs on the same node multiple times. Those variation sources include the operating system interference, external interrupts, the effect of shared resources contention, etc.

Next Section 4 proposes a methodology for quantifying the impact of the presented variability sources. Section 5 measures their impact on the CPUs and accelerators of the Eurora supercomputer.

4 Variability Exploration Methodology

We propose a methodology for evaluating and quantifying the impact of the above mentioned variability sources. Our methodology is based on a combination of scripts, real applications and synthetic benchmarks which target different variability sources. When dealing with undesired variability sources (the HW and SW) it is necessary to adopt well controlled benchmarks. Indeed supercomputer applications are characterized by the composition of several computational kernels, complex communication patterns and I/O accesses which may hide the targeted variability. Hence, in our methodology we introduce on purpose ”synthetic benchmarks” for two reasons: (i) to measure Hardware and Software Undesired Variability; (ii) to measure the corner cases of desired Hardware and Software variability. In our framework this is done with two synthetic benchmarks, a CPU bound and a Memory Bound one.

- *SYNT CPU*: this synthetic benchmark is composed by a number of threads equal to the number of cores.

[1] i.e. UnHalted Core Cycles, Instructions Retired and UnHalted Reference Cycles

Each thread is bound to a specific core with thread affinity to avoid migrations. Each thread consists of a loop where an ALU operation is executed on a circular buffer. At each iteration, a read-write to an entry of circular buffer is executed that moves with an incremental step of one cache line. In this particular case, we used 2^{37} iterations and a buffer dimension of 4KB per core, that fits the L1 cache emulating a CPU bound application (the L1 size for both the Intel Xeon E5-Series is 64KB). In fact, by doing that, the *SYNT CPU* is capable of hitting always in the L1 cache.

- *SYNT Mem*: this benchmark is similar to the *SYNT CPU* but it uses a circular buffer of 4MB per thread and 2^{33} iterations. Both the Intel Xeon E5-Series used for our tests, present a L3 shared cache of 20 MB. As each thread has its own circular buffer, the overall memory footprint (32MB) exceeds the L3 size. This is sufficient to let each memory access miss in the L1, L2, L3 cache and hit in the DRAM, emulating a strongly memory bound task execution.
- *SYNT GPU*: this synthetic benchmark allows to simulate a worst-case scenarios for the GPUs devices [22]. This program forks one process for each GPU. Each GPU process allocates 90% of the free GPU memory, initializes 2 random 1024*1024 matrices, and continuously performs efficient CUBLAS matrix-matrix multiplication routines on them and finally stores the result across the allocated memory. Both floats and doubles are tested. In this way, the GPUs are 100% busy while the CPUs remain in idle. Using this benchmark we can simulate a strong GPUs bound task execution.

Furthermore, in our methodology we use "real benchmarks" to evaluate the effect of desired hardware and software variability in practical applications. When dealing with Desired Hardware Variability, real benchmarks gives us a realistic and practical evaluation of the metrics involved, unveiling the behavior of a real application in the range marked by the synthetic benchmarks. In our methodology we choose Quantum Espresso (QE)[2]. QE is a Computational Material Science community code, publicly available and it is one of the currently "hot applications" for high-end supercomputers[3]. QE main computational kernels include dense parallel linear algebra and 3D parallel FFT, which are both relevant in many HPC applications. Hence, QE is a good candidate to evaluate HPC architectures, and it is included in many benchmark suites [19].

- *QE-*Al^2O^3: QE main computational kernels include dense parallel linear algebra and 3D parallel FFT, which are both relevant in many HPC applications. We configure QE to calculate the electronic structure of the Al^2O^3 in 3K points. The code is parallelized with 16 threads and we configure the GOMP_CPU_AFFINITY to use all the available cores within one node.
- *QE-*SiO^2: This Quantum Espresso benchmark allows to calculate the band structure of the Silicon along the main symmetry. We use *QE-*SiO^2 as it contains a larger linear algebras that the Al^2O^3 and this can be better exploited by the *QE GPU* version which mostly accelerates the linear algebra but not the FFT [21]. By using this benchmark in the GPU and CPU, we are able to compare the performance of the *QE-*SiO_2 using only the CPUs and both the GPUs and CPUs devices. As in previous benchmarks, the code is parallelized with 16 threads and we configure the GOMP_CPU_AFFINITY to use all the available cores within one node.
- *QE GPU*: the aim of this benchmark is to exploit the capabilities of the NVIDIA GPU graphics cards. The *QE GPU* uses the Quantum ESPRESSO SiO_2 suite [21] to exploit new hybrid CPU+GPU high performance computing systems. In this way, we are able to compare the performance of the same benchmark *QE-*SiO_2 using only the CPUs and both the GPUs and CPUs devices.

Desired HW Variability Methodology Calculation: In addition to the QE CPU and GPU versions, to quantify this variability source, we have designed a PBS script that first scales equally the frequencies for all the cores of the node in which is running and then executes (with N equal to five in our experiments) the same benchmark. At the beginning and at the end of each benchmark run, we save the initial time and the end time. The script iterates these operations for all the available DVFS states and for all the benchmarks considered. The script is then executed in all the nodes of Eurora. Off-line the log information are used to navigate the traces generated by the Eurora monitoring framework (Section 2). To verify the Desired HW Variability, we ran both the synthetic benchmarks and the real applications.

Desired SW Variability Methodology Calculation: By using the synthetic and real benchmarks we are able to precisely define the number and kind of operations executed for each single test. In this way, we test the Desired Software Variability by monitoring the different usage of resources by different software on each test executed. The use of the systematic benchmarks allow to verify the extreme corner case usage of the Eurora machine.

Undesired HW Variability Methodology Calculation: Again, the use of the synthetic benchmarks produce stable results allowing to separate the different variability-components. Hence, to calculate the Undesired HW variability we run for 5 times the synthetic benchmarks on the all available nodes while changing the clock frequency for each run. The average of the 5 runs at each nodes gives a good understanding of the energy efficiency of the given node. The difference between all the nodes are then calculated allowing to extract the only Undesired Hardware Variability Component.

Undesired SW Variability Methodology Calculation: To quantify this variability source we executed 5 times on the same node and for all the nodes the synthetic

[2] Quantum ESPRESSO is an initiative of the DEMOCRITOS National Simulation Center (Trieste) and of its partners, in collaboration with the CINECA National Supercomputing Center in Bologna, the Ecole Polytechnique Federale de Lausanne, Princeton University, and the Massachusetts Institute of Technology.

[3] www.quantum-espresso.org

benchmarks. In this way, we are able to precisely evaluate the energy consumed on each node and monitor the only variations in the nodes due to software fluctuations and noise.
It is important to mention that we restricted our analysis to the GPU accelerators only due to the hw limitations of the Intel Xeon Phi accelerators in scaling the clock frequency. We leave the analysis of the Intel Xeon Phi accelerator to future works. Considering these benchmarks and tests, in Section 5, we quantify the impact of all variability sources defined in Section 3 in terms of energy/performance.

5 Experimental Results

This Section presents the results of our analysis which quantifies the effects of different sources of variability in the Eurora Supercomputer. In particular, we are going to present the results obtained for the general purpose processing and GPU accelerator units.

5.1 Desired Hardware and Software Variability:

Next paragraph compares the standard operation (i.e turbo mode) of the two classes of nodes (with 2.1GHz CPUs and with 3.1GHz CPUs) while in the second paragraph we conduct a similar analysis for the accelerators units.

5.1.1 General Purpose Processing Units

To evaluate the impact of desired software and hardware variability on the energy/performance metrics for the different benchmarks, we report in Figure 3 the execution time, power and energy for the two classes of nodes (with 2.1GHz CPUs and with 3.1GHz CPUs) at the different DVFS states. Each dot in the figures reports the average among all the nodes of the same class and among the five runs of the same benchmark. Looking at the graphs we can clearly notice that in all the plots the 2.1GHz nodes lay on top of the 3.1GHz for all the frequency settings. From the same figure we can see, as expected, that by lowering the frequencies all the four different benchmarks decrease their power and increase their execution time. The energy instead is not monotonic and behaves differently for the four cases.

Table 1 quantifies for the four benchmarks the energy/performance improvements when the nodes operates at the optimal frequency for the energy consumption w.r.t standard operating conditions (i.e. turbo mode). For the *SYNT CPU*, *QE-*Al^2O^3 and *QE-*SiO^2 the minimum energy consumption happens at around 2.0GHz, while the *SYNT MEM* has minimum energy at the lower frequency. Choosing the optimal frequency for minimum energy leads to energy savings up to 18% for *SYNT CPU* and savings of the 50% for the *SYNT MEM* benchmark. *QE-*Al^2O^3 instead saves the 27% while *QE-*SiO^2 saves the 21%. Furthermore, from Table 1 we can notice that both the two QE benchmarks show a behavior similar to the CPU bound benchmark (*SYNT CPU*) and that not only the application matters for the power and energy consumption but also the data-set that need to be executed since this could lead to a difference up to 10% in terms of energy saving between the two QE benchmarks. Indeed from Figure 6 we can see that the speed-up due to frequency increase in *QE-*Al^2O^3 saturates before *QE-*SiO^2.

Nodes	Optimal [MHz] Frequency	Ex Time [%] Overhead	Energy [%] Saving
Benchmark SYNT CPU			
2.1GHz	1900	-11	+2
3.1GHz	2000	-70	+18
Benchmark SYNT Mem			
2.1GHz	1200	-18	+18
3.1GHz	1200	-23	+50
Benchmark QE Al^2O^3			
2.1GHz	1700	-20	+3
3.1GHz	1800	-65	+27
Benchmark QE-SiO^2			
2.1GHz	1800	-18	+3
3.1GHz	1800	-79	+21

Table 1
Energy Optimization Margin

To understand why this energy efficiency trade-offs happens we modeled the power and execution time of Figure 3a,b) with a regression fit. In particular, the regression model for the power (1) calculates a polynomial curve based on the input data and returns a matrix describing the curve. The equation used to model the execution time is shown in (2):

$$P_{ower} = a * x^2 + b * x + c \tag{1}$$

$$E_{xectionTime} = d/x + e \tag{2}$$

where: x is the clock frequency; *a*,*b* and *c* are the coefficients that corresponds to each polynomial value of x while; *d* and *e* are the coefficient used to model the execution time. After extracting the curves for both power and execution time of Figure 3a,b) we multiplied the two models to extract the analytical energy curve. By exploiting iterative methods we were then able to find the analytical energy minimum reported in Table 2. Hence, in this way we are capable to analytically extract the frequency that minimize the energy for each benchmark.

Nodes	Optimal [MHz] Frequency	a d	b e	c f
Benchmark SYNT CPU				
2.1GHz	2000	1.238064e-05 970544.5528	0.005529 -2.703800959	48.008
3.1GHz	2000	01.87949E-05 969715	-0.02 -1.713393	70.429
Benchmark SYNT Mem				
2.1GHz	1200	7.756455e-06 228131	0.0219 326.095	61.626
3.1GHz	1200	1.536898e-05 170744.5744	-0.00775 365.2372204	87.093
Benchmark QE Al^2O^3				
2.1GHz	1700	1.263489e-05 827629	0.006741 65.632	47.916
3.1GHz	1800	1.765919e-05 826059	-0.0147 74.215	67.95
Benchmark QE SiO^2				
2.1GHz	1800	1.75475E-05 869489	-0.0056 26.970	59.31
3.1GHz	1800	1.381937e-05 871497	0.00864 27.89	48.59

Table 2
Energy Minimum by Analytical Model

Looking at Table 2 it becomes clear why the energy minimum of the benchmarks are located in certain positions for each benchmark. The reason must be found on the value of the coefficient *d* that indicates the slope of the execution

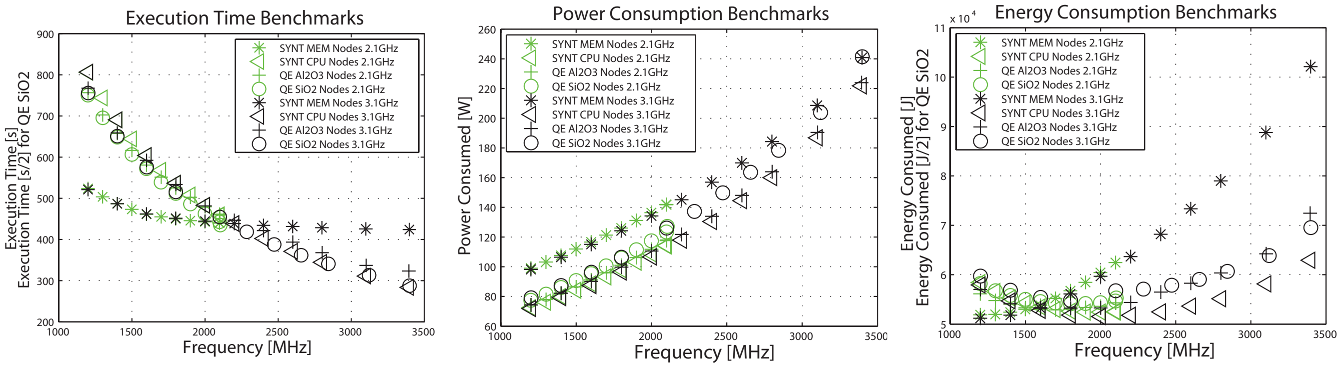


Figure 3. 3.1GHz nodes vs 2.1GHz nodes - Mean Ex Time, Power, Energy

time while varying the clock frequency. In-fact, from the Figure 3b) we can notice that the power trend is very similar and linear for the different benchmarks. On the contrary, the slope of duration's is changing. Hence, from our regression models we expect that the energy minimum is going to move based on the duration's slope. This intuition is confirmed by the results reported in Table 2: the lower value of d for the benchmark *SYNT Mem* compared to the value of d for the *SYNT CPU* determines a lower energy minimum for the *SYNT Mem* benchmark. Interesting, the *SYNT CPU* and *SYNT Mem* have respectively the higher and the lower slopes while the Al^2O^3 and QE SiO^2 stand in the middle of the two corner cases.

From the analysis of these results it becomes clear that the vast majority of workloads achieve significantly higher energy efficiency when they do not run at peak performance. Hence, for energy-constrained supercomputers new management strategies for allocating machine resources to workload can be conceived that reward energy efficiency with respect to pure performance.

Figure 4 shows in details the breakdown of the power consumption for all the main components of the General Purpose Processing Units for the CPU benchmarks varying the frequency of the system. From the analysis of the Figure, it appears obvious that the power consumption of the memory of the *SYNT Mem* benchmark is drastically higher (22% Power_Dram) when compared with the power consumed by the *SYNT CPU* benchmark (7% Power_Dram). In this graph we represent with *Power_TOT* the sum of the power consumed by the package (Power_Pkg) and the power consumed by the memory of the nodes (Power_Dram).

It is interesting to notice that for all the benchmarks considered the power consumption of the node when operating at the optimal frequency is around 100 watts, which is 3x less than the TDP. This suggests that designing supercomputers which operate at the most energy-efficient point does not only reduces the power bill but also eases the thermal design which in the case of Eurora would have been downsized up to a factor of 3x. This has the potential to reduce cooling infrastructure cost and enabling simplified and yet more efficient cooling design.

5.1.2 GPU Accelerators

In the second test we have conducted the same analysis of the previous paragraph, on the nodes equipped with the General Purpose accelerators (GPUs). Table 3 shows the different energy/performance metrics in executing the two benchmarks for GPUs (*SYNT GPU* and *QE GPU*). The second and third rows of the table are used to compare the performance of the same SW benchmark ($QESiO_2$) while using different HW components: the only CPUs and the CPUs+GPUs processors. The values are averaged among five repetitions of the same workload. Whereas table 3 reports for $QESiO_2$ the performance and power for both the case with CPU + GPU and CPU and GPU idle, for SYNT GPU we report only the case with CPU + GPU as the benchmark is designed for work with the GPU.

Nodes Nodes	Power [W]	Ex Time [s]	Energy [KJ]
Benchmark SYNT GPU			
3.1GHz +GPU	439 103+336(GPU)	175	76.5
Benchmark QE CPU (SiO^2)(GPU idle)			
3.1GHz +GPU idle	271 244+27(GPU)	590	160
Benchmark QE GPU (SiO^2) (CPU at 3.1GHz)			
3.1GHz +GPU	306 (+13%) 202+104(GPU)	502 (-15%)	156 (-3%)

Table 3
Desired Hardware Variability Results GPU

From Table 3 we can notice that the use of the GPUs devices strongly influences the performance and energy of the benchmark execution. In-fact, even if the use of the GPUs increase the power consumption, there is an important reduction of the execution time up to 15% that bring the energy consumption to decrease up to 6% with respect to the execution of the same benchmark with the only CPUs device. It is important to remember that the *QE-*SiO^2 is a real benchmark for which only the 30% of the workload is managed by the GPUs accelerators. This means that the *QE GPU* benchmark is mostly CPU bound and thus a more GPU friendly code or a better coding can results in larger speed-ups and energy savings. This can also be noticed by the comparisons with the *SYNT GPU*benchmark for which the GPUs consumes up to 336Watts which is 3x more than the power consumption of the GPU for the *QE-*SiO^2 benchmark. Overall, it is important to notice that the energy efficiency is increased by GPUs even if their utilization is partial. From our results we can notice that a 30% of utilization of the GPU is an empirical safe-threshold which programmers should aim for, to achieve energy-savings in GPU's accelerated code. Higher utilization increases the

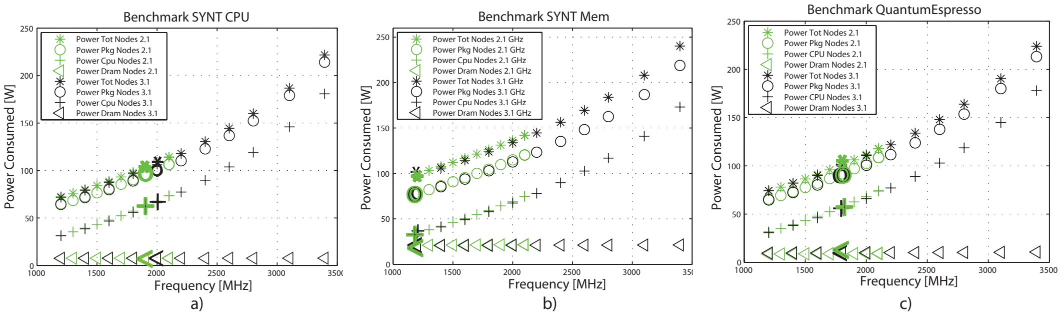


Figure 4. 3.1GHz nodes vs 2.1GHz nodes - Mean Power for the main components of the CPU nodes.

energy-gain introduced by the GPU usage.

In Figure 5 we further analyze the performance and power trade-offs of the GPU nodes by reporting for the SYNT GPU benchmark its execution time, power and energy breakdown while vary the frequency selection. Indeed, also the GPU can scale its own clock frequency independently to the CPU clock one. From Figure 5a we can notice that the execution time increases super-linearly as the GPU's clock frequency reduces. The node power consumption decreases significantly for the *SYNT GPU* benchmark, since the GPU computation and power dominates the entire budget. When we look at the energy we notice that it decreases when the GPU clock frequency is decreasing as well, but the energy minimum is not at the minimum frequency point, but at an intermediate one. This suggests that the optimal frequency selection has not a trivial solution also for GPU accelerators.

To analytically understand where and why the optimal frequency selection happens we modeled the total power and execution time of Figure 5a,b) with a regression fit. As for the CPU Section, the regression model for the power and the execution time are reported in (1) and (2). After extracting the two models, we multiplied them to extract the analytical energy curve. Then, by exploiting iterative methods we were able to analytically find the frequency that minimize the energy for each benchmark (Table 4). Looking at Table 4 it becomes clear why the *SYNT GPU* has a higher frequency optimum compared to the *QE GPU* benchmark. Again, the reason must be found on the value of the coefficient d that indicates the slope of the execution time while varying the clock frequency. In-fact, the slope of the duration equation is different for the two benchmarks: Figure 6a) shows that the duration's slope of the *SYNT GPU* is higher compared to the one of the *QE GPU* benchmark (Figure 7a)) that instead is almost flat. Hence, from our regression models we expect that the energy minimum is going to move based on the duration's slope. This is confirmed by the results reported in Table 4: the lower value of d for the benchmark *QE GPU* compared to the value of d for the *SYNT GPU* determines a lower energy minimum for the *QE GPU* benchmark.

To evaluate the impact of DVFS states on the energy performance metrics of the two GPUs benchmarks, we report in Figure 6 and Figure 7 the execution time, power and energy consumption of the *SYNT GPU* and *QE GPU* benchmark at different DVFS states. In addition to changing the GPUs frequency, we repeated the same test varying also the frequency of the CPUs device. In this way, we were able to explore an hybrid CPU+GPU DVFS configuration and evaluate its impact on the energy-performance trade-off. In particular, in addition to the GPUs DVFS, we run different tests changing the frequency of the CPUs to 1200, 2000, 3100 and 3100+turbo(3101) MHz. With the values selected we can explore all the DVFS CPU's range including the turbo boost modality. Each dot in the figures reports the average value among all the nodes which embeds GPUs and among five runs for the mentioned CPUs and GPUs frequency.

Optimal [MHz] Frequency	a d	b e	c f
Benchmark SYNT GPU			
614	0.0003243 274139	0.3086208 -223.73	66.33
Benchmark QE CPU			
324	-0.0003142 30298	0.48 451.99	119.12

Table 4
Energy Minimum by Analytical Model

Looking at the graphs we can clearly notice that by changing the CPUs and GPUs frequency we have different behaviors in the execution time, power and energy consumption due to the different nature of the two presented benchmarks. In particular, the *SYNT GPU* is a GPU bound benchmark and the frequency variation of the CPUs is not altering its execution time as we can see from Figure 6a. Instead, from Figure 6b we can see that the power is affected by the CPUs frequency and by decreasing the CPUs speed we significantly reduce the overall power and total energy consumption (Figure 6c). It results that the minimum CPU's frequency leads to the minimum overall power and energy consumption. If we look at the GPU frequency we can notice instead that even if the benchmark is GPU centric the maximum energy efficiency is not achieved at the maximum frequency nor at the minimum one. The optimal frequency is obtained with minimum CPU frequency and intermediate GPU frequency. To the best of authors knowledge, this is the first work in showing the evidence that consensus in the DVFS policy for the accelerator and host processor can lead to significantly higher energy saving with respect to state-of-the-art accelerator agnostic power management policies. Hence, from the analysis of these results we clearly notice that it is possible to achieve significantly higher energy efficiency by not running the *SYNT GPU* benchmark at peak performance.
Figure 7 shows the results of the same analysis applied to

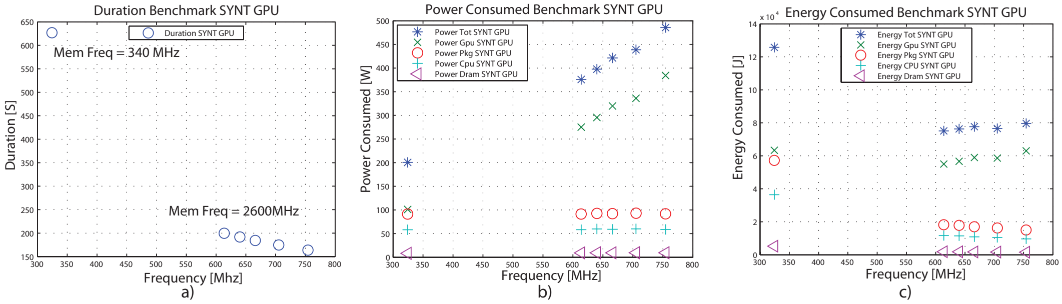


Figure 5. 3.1GHz+GPU nodes - Mean Ex Time, Power, Energy for all the components of the GPU nodes.

the *QE-*SiO^2 benchmark and shows significantly different trade-offs. Here the execution time strongly depends on both the GPUs and CPUs clock frequencies used, confirming the hybrid nature of the *QE GPU* benchmark which contains both GPUs and CPU centric computational phases. Figure 7a reports the execution time of the benchmark for the different configurations of the CPU's and GPU's clock frequency. From the same figure we can notice that the execution time is more sensitive to the CPU clock frequencies underlying a dominant effect of the use of the CPUs in the *QE GPU* benchmark. A similar behavior is present in Figure 7b which shows the power consumption of the *QE GPU* benchmark. Also in this case the power increases mostly as effect of CPU's frequency variations. Finally in Figure 7c we can see the effects of the CPU and GPU frequency variation for the overall energy consumption of the *QE GPU* benchmark.
Interestingly, even if this benchmark has a clear dominance of the CPU load in the power and execution time and thus is significantly different from the previous SYNT GPU benchmark, the minimum energy efficiency as well as the best energy efficiency trade-offs are achieved with an intermediate value for the CPUs frequency. As a matter of fact, also in this case we show that to reduce the energy of the system the highest CPUs and GPUs speed are not the best option and only a careful analysis of the benchmark determines the best choice.

Table 5 quantifies for the two benchmarks the energy/performance improvements when the GPUs and both the CPUs+GPUs processors operate at the optimal frequency w.r.t standard operating conditions (i.e. turbo mode for CPUs and GPUs running at the maximum frequencies). Numbers in brackets reports the optimal frequency calculated with the polynomial regression model. By selecting the optimal GPU frequency while maintaining the standard CPUs operating mode to achieve the minimum energy consumption leads to energy savings up to 6% for *SYNT GPU* and savings up to 13% for the *QE GPU* benchmark. If we consider an holistic DVFS policy which exploits at the same time the CPU and GPU DVFS capabilities we can increase the energy savings for the *SYNT GPU* benchmark up to 17% and up to 26% for the *QE GPU* benchmark.

As a matter of fact there is a large opportunity for today's and future heterogeneous supercomputer to increase their energy efficiency by leveraging synergies in between the CPU and GPU and the accelerator's power management policy.

Optimal Frequency [MHz]	Ex Time [%] Overhead	Energy [%] Saving
Benchmark SYNT GPU [CPU F_MAX - GPU F_OPT]		
3101CPU+614GPU	-21	+6
Benchmark *QE-*SiO^2 GPU [CPU F_MAX - GPU F_OPT]		
3101CPU+324GPU	-11	+13
Benchmark SYNT GPU [CPU F_OPT - GPU F_OPT]		
1200CPU+614GPU	-23	+17
Benchmark QE GPU [CPU F_OPT - GPU F_OPT]		
2000CPU+324GPU	-64	+26

Table 5
Energy Optimization Margin GPU

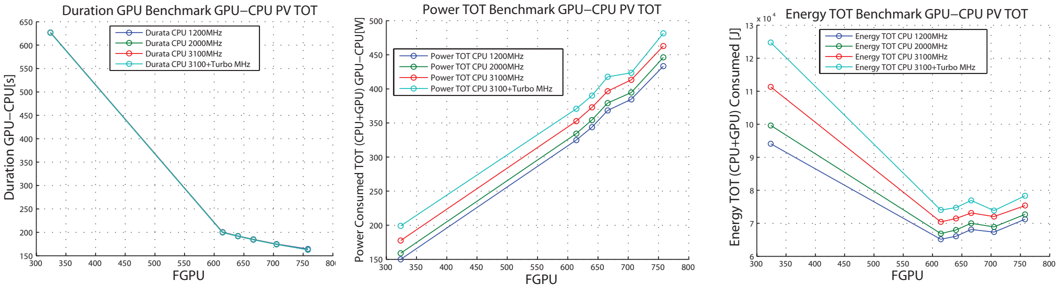
5.2 Undesired Hardware Variability

Whereas the previous explorations were conducted considering the average among the different nodes of the same class HW, in the following analysis we will compare the energy/performance metrics of single nodes. To improve the quality results for each node and for each benchmark we consider average of the five runs.

5.2.1 General Purpose Processing Units

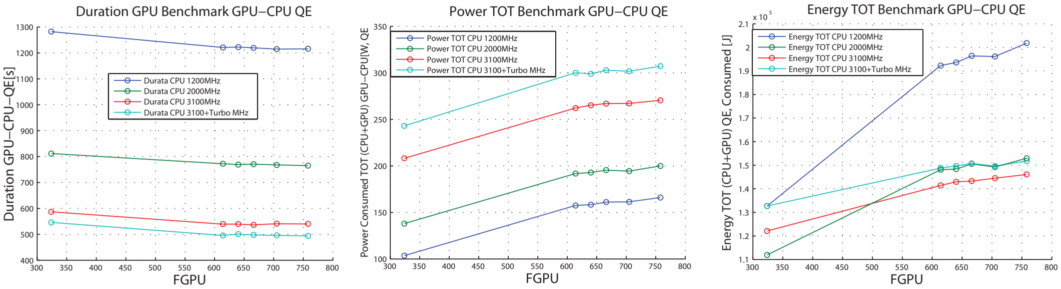
In the following analysis we focus only on the synthetic benchmarks as they highlight the corner cases and are characterized by lower variability among repetitions (Table 6). Figure 8 for each node and benchmark (*SYNT CPU*, *SYNT MEM* and *IDLE*) shows on the x-axis the average total power consumption and on the y-axis the average core temperature. From the figure we can notice that both the CPUs have a similar thermal resistance. This is expected as they share the same packaging solution and cooling solution. Furthermore, from the same plot we can notice two important effects: first, for each benchmark the power-to-temperature relation is linear. This means that positive feedback loop in between the absolute temperature and the materials resistance are negligible in a real setup in the range of allowed operating temperatures.

Secondly, different benchmarks when consuming a similar amount of power, have different impact on the core temperature. This can be clearly see by the different slopes for the different benchmarks plots. This suggests that thermal management needs to be done based on actual temperature measurements and cannot be performed based only on the core power consumption. In addition to that we notice that nodes of the same class at the same DVFS level shows significantly different thermal/power behavior. Moreover, the effects of variability on thermal resistance grows at higher power consumption. This is also the condition at which temperature becomes more critical. Furthermore, as



a) b) c)

Figure 6. SYNT GPU: DVFS impact for Execution Time, Power and Energy Consumption



a) b) c)

Figure 7. QE GPU: DVFS impact for Ex Time, Power and Energy Consumption

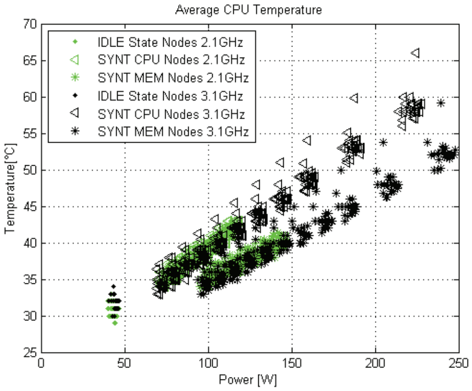


Figure 8. Average CPU Temperature

it can be expected, even if *SYNT MEM* has a higher power consumption than *SYNT CPU* due to DRAM power its core temperature is significant lower ($\approx 5^oC$).

Figure 9a and 9b visually quantify the effect of process variability among Eurora nodes. As highlighted previously, in Figure 3 the energy consumption has a minimum at a lower frequency than the maximum one. This is true for all the nodes. In the same figure we can see that different nodes show a significant variability on energy consumption that carries over all DVFS operating points along the different DVFS values. Figure 9c quantifies the maximum variation between all nodes of the same class at the different frequencies while executing the same benchmark. We can notice that the energy variation is totally due to the power variation as the execution time variation is negligible. Energy variation can reach almost 9% and its average is a non-negligible 7%. Figure 10a and 10b show the Eurora Energy map trend for the memory bound benchmark. Here we can see an outlier (node57). This node has DRAM clock half of the other nodes one even if it is nominally the same. This is the reason why the outlier was not present in Figure 9a and 9b. We removed this node in Figure 10c that quantifies the maximum variation between all the nodes of the same class at the different frequencies while executing the same benchmark. From the plot we can see that memory bound applications incur in higher variability w.r.t CPU bound ones, with peak variation of 15% and average of 8%. This can be explained by higher sensibility to the DRAM variability. It must be noted that this value are computed on 32 nodes and thus are expected to increases in larger systems and more advanced technology nodes.

5.2.2 GPU Accelerators

In this section we focus only on the synthetic GPU benchmark as it is characterized by lower variability among repetitions (Table 7). Figure 11a and 11b visually quantify the effect of process variability among the Eurora nodes equipped with GPUs. As mentioned in the Desired Hardware Variability Section, Figure 6 shows the energy consumption, revealing a minimum that does not match the maximum frequency available. This happen within all the nodes running the *SYNT GPU* benchmark. By the analysis of Figure 11a and 11b we can see that different nodes show a significant variability on energy consumption that carries over all DVFS operating points along the different DVFS values. Moreover, Figure 11c quantifies the maximum variation between all nodes at the different frequencies while executing the same benchmark. These results show that the energy variation is totally due to a power variation between the nodes as the execution time variation is always very low compared to the power variation. Also by using GPUs device the Energy variation can reach almost 9% and its average is a non-negligible 6.9%.

Thus our measurements show that, even in today's technology, node-to-node variability cannot be neglected and

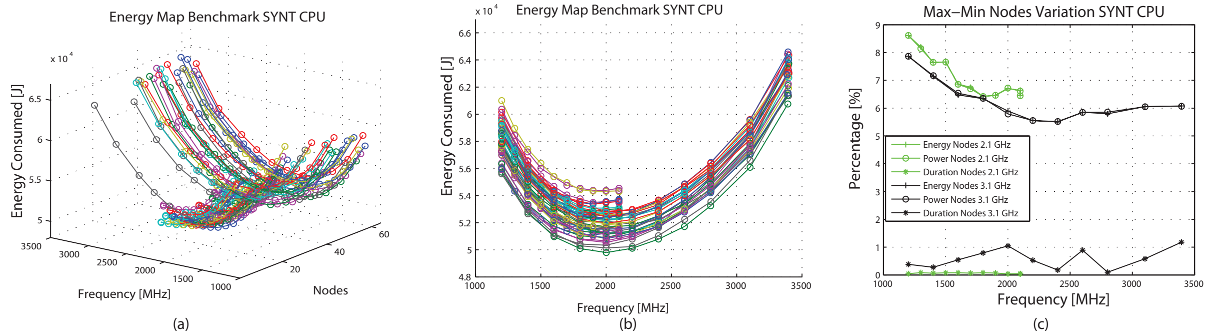


Figure 9. Energy Map SYNT CPU Results

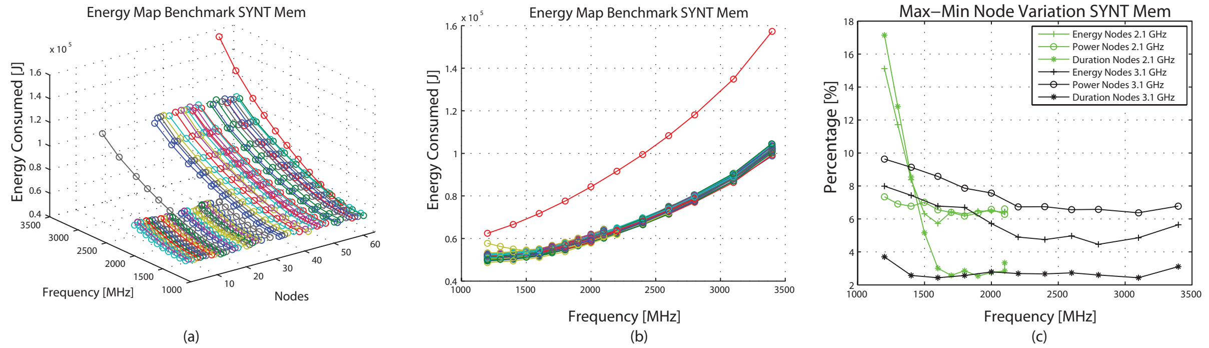


Figure 10. Energy Map SYNT MEM Results

according to all technology projections, it will increase as we move to scaled-down nodes. Hence variability managements in software (e.g. through proper workload allocation and scheduling by a variability-aware job dispatcher) will become a necessity for future Exascale systems.

5.3 Undesired Software Variability

This accounts for the operating system and system interferences. We evaluated it by launching several times the same benchmark on the same hardware node.

5.3.1 *General Purpose Processors Units*

In Table 6 we quantify the percentage of the variability between five different runs on the same Eurora nodes using only CPUs.

Nodes	Power[%]	Ex Time[%]	Energy[%]
Benchmark SYNT CPU			
2.1GHz	0.23	0.04	0.24
3.1GHz	0.97	2.21	1.77
Benchmark SYNT Mem			
2.1GHz	0.41	1.27	1.20
3.1GHz	0.70	1.55	1.36
Benchmark *QE-Al*^{2}*O*^{3}			
2.1GHz	0.36	0.99	1.05
3.1GHz	1.93	6.40	4.70

Table 6
Software Variability Results

In Table 6 we can notice that synthetic benchmarks show less SW variability as their computational patterns are simpler and more regular. *QE-Al*^{2}*O*^{3} shows instead a significantly higher SW variability up to 5% in terms of energy consumption. This should be considered as unavoidable process noise and managed properly when designing feedback-based energy management techniques using model-predictive formulations.

5.3.2 *GPU Accelerators*

In Table 7 we quantify the percentage of variability between five different runs on the same Eurora nodes equipped with GPUs accelerators. Similar consideration as for the Table 6 can be noticed looking at Table 7. The *QE GPU* benchmark shows higher SW variability almost up to 2% in terms of power consumption. However, in this case the SW variability is more similar between the two GPU benchmarks, showing that the GPUs device are less affected from undesired software variability.

CPU - GPU Freq[%]	Power[%]	Ex Time[%]	Energy[%]
Benchmark SYNT GPU [CPU F_MAX - GPU F_MAX]			
3101CPU+758GPU	1.02	0.03	1.02
Benchmark QE GPU [CPU F_MAX - GPU F_MAX]			
3101CPU+758GPU	1.68	1.06	1.31
Benchmark SYNT GPU [CPU F_MIN - GPU F_MAX]			
1200CPU+758GPU	0.75	0.61	1.07
Benchmark QE GPU [CPU F_MIN - GPU F_MAX]			
1200CPU+758GPU	1.13	0.43	1.38

Table 7
Software Variability Results GPU

Summing up, we believe that SW variability should be a strong driver to derive programming guidelines and APIs for better controlling it and avoiding its blowup in future exascale systems.

6 Conclusion

As a result of our analysis we derive design guidelines for future heterogeneous supercomputers based on large numbers of nodes hosting general purpose processors and parallel accelerators:

1) General purpose processors: for a given process technology and processor family, high-speed devices are less energy efficient than the slow ones.

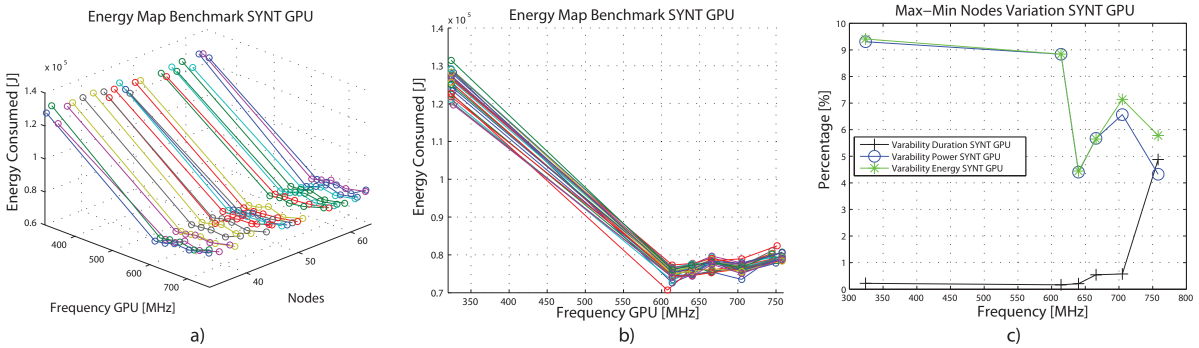


Figure 11. Energy Map SYNT GPU Results

Moreover, we discover that the majority of workloads achieve higher energy efficiency when they do not run at peak performance. Thus, green supercomputers should either be designed with a small portion of high clock frequency processors w.r.t low speed ones to sustain sporadic CPU bound jobs or use only high-speed processors if peak parallel performance is at absolute premium, but downclocking them for the vast majority of real-life workloads.

2) We found out that in real applications, GPUs can effectively reduce the energy consumption of the overall benchmark even when partially used. We empirically found out an energy-efficient safe threshold for code usage of the GPU of the 30%, percentage at which the GPU accelerated version of the code start to gain energy efficiency with respect to the CPU one.
3) We quantified the impact of combined CPUs and GPUs dynamic voltage and frequency scaling in real HPC workload. We discover that there is a large opportunity for today's and future heterogeneous supercomputer to increase their energy-efficiency by leveraging synergies in between the host's and the accelerator's power management policy.
4) We show that (i) in real scenario for a large set of workload and power levels the power-to-temperature relation is linear, suggesting that thermal resistance of the materials involved is not influenced by the absolute temperature; (ii) different benchmarks which consume a similar amount of power have different core temperatures. This suggests that thermal control cannot be performed by solely monitoring and controlling the power consumption, i.e. thermal control done by mean of power capping is sub-optimal; (iii) different nominally equal nodes have different package thermal resistance, and that the impact of thermal resistance variability grows at higher power consumption. This leaves opportunities for self-calibrating thermal controllers.
5) In addition to thermal variation there is a significant node-to-node energy variability, which is totally induced by a power consumption variability. We measured an energy variation on the CPUs up to 9% which increases to the 15% in DRAM centric benchmarks and the 9% for GPUs centric benchmarks. This result shows that todays supercomputer can benefit from system-level variability-aware resource management solutions. Their impact will become more relevant in future installations, as variability is foreseen to worsen as the technology scales.
6) We introduced a specific methodology to quantify variability sources that allowed to measure up to 5% of variability on the energy consumption of multiple run of the same real supercomputer benchmark (QE) on the same node with controlled operating point. This variation is 3 times larger than the one present in synthetic benchmarks. This should be considered as unavoidable process noise and managed properly when designing feedback-based energy management techniques using model-predictive formulations.

As a final remark, we believe that variability monitoring infrastructure and closed loop variability management will be essential tools to build sustainable future supercomputers. Our future work will focus on building a scalable observation and control infrastructure for future exascale machines featuring hundreds of thousands of heterogeneous computing nodes.

Acknowledgments

This work was partially supported by the FP7 ERC Advance project MULTITHERMAN (g.a. 291125), the H2020 FETHPC ANTAREX (g.a. 671623) and by the YINS RTD project (no. 20NA21 150939), evaluated by the Swiss NSF and funded by Nano-Tera.ch with Swiss Confederation financing.

References

[1] J. Dongarra. Visit to the National University for Defense Technology Changsha, China. *Technical report*, University of Tennessee, 06 2013.

[2] J. J. Dongarra, H.W. Meuer, E. Strohmaier, et al, "Top500 supercomputer sites." *Supercomputer*, 13:89-111, 1997.

[3] Feng, Wu-chun, and Kirk W. Cameron, "The green500 list: Encouraging sustainable supercomputing," *Computer* 40.12 (2007): 50-55.

[4] Bergman, Keren, et al., "Exascale computing study: Technology challenges in achieving exascale systems," in *Defense Advanced Research Projects Agency Information Processing Techniques Office* (DARPA IPTO), Tech. Rep 15 (2008).

[5] K. A. Bowman, et al., "Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration," in *IEEE J. Solid-StateCircuits*,vol.37,no.2,pp.183-190,Feb.2002.

[6] Datta, Animesh, Swarup Bhunia, Jung Hwan Choi, Saibal Mukhopadhyay, and Kaushik Roy, ”Speed binning aware design methodology to improve profit under parameter variations,” in *Design Automation, 2006. Asia and South Pacific Conference on*, 2006, pp. 712-717.

[7] Pang, Liang-Teck, Kun Qian, Costas J. Spanos, and Borivoje Nikoli, ”Measurement and analysis of variability in 45 nm strained-Si CMOS technology,” in *Solid-State Circuits, IEEE Journal* of 44, no. 8 (2009): 2233-2243.

[8] P. Mercati, A. Bartolini, F. Paterna, T. S. Rosing, and L. Benini, ”Workload and user experience-aware dynamic reliability management in multicore processors,” In *Proceedings of the 50th Annual Design Automation Conference*, DAC ’13, pages 2:1-2:6, New York, NY, USA, 2013. ACM.

[9] ACPI, ”Advanced Configuration and Power Interface Specification,” Available at: http://www.Intel.com/products/processor/manuals/

[10] V. Pallipadi and A. Starikovskiy, ”The ondemand governor: past, present and future,” In *Proceedings of Linux Symposium, vol. 2, pp. 223-238*, 2006.

[11] C. Zhuo, D. Sylvester, and D. Blaauw, ”Process variation and temperature-aware reliability management,” in *Design, Automation Test in Europe Conference Exhibition (DATE), 2010*, pages 580-585, march 2010.

[12] F.Paterna, et al, ”Variability-Aware Task Allocation for Energy-Efficient Quality of Service Provisioning in Embedded Streaming Multimedia Applications,” in *IEEE Transactions on Computers*, 61(7) July 2012

[13] Paterna, Francesco, Andrea Acquaviva, and Luca Benini. ”Aging-aware energy-efficient workload allocation for mobile multimedia platforms.” in *Parallel and Distributed Systems, IEEE Transactions on* 24.8 (2013): 1489-1499.

[14] G. Dhiman, G. Marchetti, and T. Rosing, ”Green: A System for Energy-Efficient Management of Virtual Machines,” in *ACM TODAES*, 2010.

[15] D. Lo, et al., ”Dynamic Management of TurboMode in Modern Multi-core Chips,” in *Proc. HPCA*, 2014, Orlando, USA

[16] C. Cavazzoni, ”EURORA: a European architecture toward exascale,” in *Proceedings of the Future HPC Systems: the Challenges of Power-Constrained Performance*(FutureHPC ’12), ACM, New York, NY, USA, , Article 1 , 4 pages.

[17] Krisztin Flautner and Trevor Mudge, ”Vertigo: automatic performance-setting for Linux,” in *Proceedings of the 5th symposium on Operating systems design and implementation*(OSDI ’02). ACM, New York, NY, USA, 105-116.

[18] J. Kim, M. Ruggiero, and D. Atienza, ”Free cooling-aware dynamic power management for green datacenters.” in *Proc. HPCS International Conference on*, pages 140-146, 2012.

[19] P. Giannozzi, et al, J.Phys.:Condens.Matter, 21, 395502 (2009). http://dx.doi.org/10.1088/0953-8984/21/39/395502.

[20] PRACE. Partnership for Advanced Computing in Europe.

[21] Pascolo, E. et al., ”High Performance Computing Simulation (HPCS),” in *International Conference on High Performance Computing & Simulation*, 2014.

[22] Multi-GPU CUDA stress test. http://wili.cc/blog/gpu-burn.html

[23] Rudi, Andrea, et al. ”Optimum: Thermal-aware task allocation for heterogeneous many-core devices,” in *High Performance Computing & Simulation (HPCS), 2014 International Conference on.* IEEE, 2014.

[24] Kuhn, Kelin J., ”CMOS transistor scaling past 32nm and implications on variation,” in *IEEE journal of Advanced Semiconductor Manufacturing Conference*(ASMC). 2010.

[25] Borkar, Shekhar, et al. ”Parameter variations and impact on circuits and microarchitecture,” in *Proceedings of the 40th annual Design Automation Conference.* ACM, 2003.

[26] The Green500 List - June 2015, http://www.green500.org/news.

[27] Borkar, Shekhar. , ”Designing reliable systems from unreliable components: the challenges of transistor variability and degradation,” in *Micro, IEEE 25.6*(2005): 10-16.

[28] Hassan H. et al,”MOS current mode circuits: analysis, design, and variability,”in *Very Large Scale Integration (VLSI) Systems*, IEEE Transactions, 2005.

[29] M. Alam, ”Reliability- and process-variation aware design of integrated circuits,” in *19th European Symposium on Reliability of Electron Devices, Failure Physics and Analysis* (ESREF 2008).

[30] Ping Yang, et al, ”An Integrated and Efficient Approach for MOS VLSI Statistical Circuit Design,” in *Computer-Aided Design of Integrated Circuits and Systems*, IEEE Transactions, January 1986.

[31] Abu-Rahma, et al, ”Variability in VLSI Circuits: Sources and Design Considerations,” in *IEEE International Symposium on Circuits and Systems*, 2007.

[32] Wanner, Lucas, et al. ”NSF expedition on variability-aware software: Recent results and contributions,” in *it-Information Technology* 57.3 (2015): 181-198.

[33] Akhil Langer, et al. ”Energy-efficient computing for HPC workloads on heterogeneous manycore chips,” in *Proceedings of the Sixth International Workshop on Programming Models and Applications for Multicores and Manycores.* (PMAM ’15),ACM, 2015.

[34] Totoni, Ehsan, et al. ”Scheduling for HPC Systems with Process Variation Heterogeneity,” *PPL Technical Report*, 2014. http://charm. cs. uiuc. edu/media/14-35.

[35] Svilen Kanev, et al, ”Profiling a Warehouse-Scale Computer,” in *International Symposium on Computer Architecture* (ISCA). June 2015.

[36] Davis,et al, ”Accounting for variability in large-scale cluster power models,” in *Exascale Evaluation and Research Techniques Workshop*(EXERT), 2011.

[37] Davis, John D., et al, ”Including Variability in Large-Scale Cluster Power Models,” in *Computer Architecture Letters* 11.2 (2012): 29-32.

[38] Balaji, Bharathan, et al., ”Accurate characterization of the variability in power consumption in modern mobile processors,” in *Proceedings of the 2012 USENIX conference Power-Aware Computing and Systems*, HotPower. Vol. 12.

[39] Wilde, Torsten et al, ”Taking Advantage of Node Power Variation in Homogenous HPC Systems to Save Energy,” in *Springer International Publishing 2015.*

[40] Geist, Al, and Daniel A. Reed, ”A survey of high-performance computing scaling challenges,” in *International Journal of High Performance Computing Applications* (2015).

[41] Kramer, William, and David Skinner, ”Consistent Application Performance at the Exascale,” in *International Journal of High Performance Computing Applications* 23.4 (2009).

[42] Schroeder, Bianca, and Garth Gibson, ”A large-scale study of failures in high-performance computing systems,” in *Dependable and Secure Computing*, IEEE Transactions on 7.4 (2010).

[43] Di Martino et al, ”Lessons learned from the analysis of system failures at petascale: The case of blue waters,” in *Dependable Systems and Networks (DSN), 2014 44th Annual IEEE/IFIP International Conference on*. IEEE, 2014.

[44] Wang, Ke and Kulkarni, Abhishek and Lang, Michael and Arnold, Dorian and Raicu, Ioan ”Exploring the Design Tradeoffs for Extreme-Scale High-Performance Computing System Software,” in *Parallel and Distributed Systems, IEEE Transactions on*, Volume:PP , Issue: 99, (2015).

[45] Francesco Fraternali, et al, ”Quantifying the impact of variability on the energy efficiency for a next-generation ultra-green supercomputer,” in *Proceedings of the 2014 international symposium on Low power electronics and design*, ISLPED ’14.

[46] Feng, Hanhua, Vishal Misra, and Dan Rubenstein, ”PBS: a unified priority-based scheduler,” in *ACM SIGMETRICS Performance Evaluation Review*. Vol. 35. No. 1. ACM, 2007.

[47] Reliability simulation in integrated circuit design a whitepaper. $www.cadence.com/whitepapers/5082_{R}eliabilitySim_{F}NL_{W}P.pdf$

[48] Goda, Ananth Somayaji, and Gautam Kapila, ”Design for degradation: CAD tools for managing transistor degradation mechanisms,” in *Quality of Electronic Design. ISQED 2005. Sixth International Symposium on.* IEEE, 2005.

[49] Carlson, T.E. et al, ”Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulation ,” in *High Performance Computing, Networking, Storage and Analysis* (SC), 2011 International Conference for , vol., no., pp.1-12, 12-18 Nov. 2011.

[50] The raw data and more information is available at the following two URLs: www.pdl.cmu.edu/FailureData/ and www.lanl.gov/projects/computerscience/data/, 2006

PLACE PHOTO HERE

Francesco Fraternali Francesco Fraternali received a Master degree in Electronic Engineering from the University of Bologna, Italy, in 2012. He is currently a PhD student at Computer Science and Engineering Department at University of California, San Diego (USA) after one year as a Research Assistant in the Department of Electronic and Computer Science (DEIS) at University of Bologna. His research interests include energy efficiency in multi-core platforms, occupancy detection in smart buildings and IoT.

PLACE PHOTO HERE

Andrea Bartolini Andrea Bartolini received a Ph.D. degree in Electrical Engineering from the University of Bologna, Italy, in 2011. He is currently a postdoc researcher in the Department of Electrical, Electronic and Information Engineering Guglielmo Marconi (DEI) at the University of Bologna. He also holds a post-doc position in the Integrated Systems Laboratory at ETH Zurich. His research interests concern dynamic resource management ranging from embedded to large scale HPC systems with special emphasis on software-level thermal and power-aware techniques. His research interest also includes ultra-low power design strategies for biosensors nodes operating in near-threshold.

PLACE PHOTO HERE

Carlo Cavazzoni Carlo Cavazzoni graduated in Physics at the University of Modena in 1994 and got a PhD cum laude at the International School for Advanced Studies (ISAS-SISSA) of Trieste in 1998. He is presently a staff member of the CINECA SCAI Department in charge of the coordination of technical and operational activities. He collaborate with different user communities to enable applications on massively parallel systems and innovative architecture solutions. He is responsible for the parallel design of Quantum ESPRESSO suite of codes. He is author and coauthor of several papers published in prestigious international reviews including Science, Physical Review Letters, Nature Materials, and many others. In PRACE 2IP he was responsible for the EURORA prototype.

PLACE PHOTO HERE

Luca Benini Full Professor at the University of Bologna and he is the chair of Digital Circuits and Systems at ETHZ. He has served as Chief Architect for the Platform2012/STHORM project in STmicroelectronics, Grenoble in the period 2009-2013. Dr. Beninis research interests are in energy-efficient system design and Multi-Core SoC design. He is also active in the area of energy-efficient smart sensors and sensor networks for biomedical and ambient intelligence applications. He has published more than 700 papers in peer-reviewed international journals and conferences, four books and several book chapters.