Monitoring GreenClouds
Evaluating the trade-off between Performance and Energy Consumption in DAS-4

R.Fontana K.Mparmpopoulou
P.Grosso and K. van der Veldt

University of Amsterdam

System and Network Engineering

February 11, 2013

Abstract

This paper investigates the correlation among two independent systems capable of generating metric information about workload and energy consumption from DAS-4. The systems were operating independently and there were no previous analysis conducted to evaluate a possible trade-off between performance and energy usage. After establishing the main metrics to be evaluated, test scenarios were created to compare both metrics. Conclusions on the aforementioned trade-off were based on final outcomes from the experiments. Power consumption was roughly the same in all experiments. Also, an alternative implementation of the current infrastructure is suggested as an optimization technique to increase the mentioned trade-off.

Contents

1 Introduction

Cloud Computing, the agile IT solution of the 21st century, is a promising technology that has already changed the way in which users receive computing utilities, such as storage capacity, software applications accessing, data processing. Computing resources are distributed over the network and they are granted and dedicated to users as services (worldwide), until users decide to release them.

Cloud Computing is an emerging technology with benefits for both cloud providers and end users. End users, instead of owning and potentially managing their own systems, which means an increase in their expenses, share a large centralized pool of storage, network and computing resources. Recently, many companies and institutes are increasingly shifting from building and maintaining their own information systems to using cloud computing services [1]. From the provider's perspective, making use of resource time sharing and charging users in time or even usage basis bring them back profit from their investments in management and infrastructure.

In the widespread use of cloud computing services, many developments of advanced computing technologies have contributed; high speed networks, virtual computing, grid computing, to name but a few. We can even argue that Cloud Computing is not only the evolution of Grid Computing, but also still relies on it as its backbone [2]. These two technologies, even they differ in many aspects such as the levels of security and abstraction, they still have some similar characteristics like technology and architecture.

Virtualization, one of the main characteristic keys of Cloud Computing, is often used due to the efficient management of resources that offers. The benefits are much more than a safe isolation of multiple consolidating services under a smaller number of computing resources. Cloud service providers can easily migrate and allocate the resources anytime and anywhere, according to the demand. Through resource consolidation and migration virtualization brings effective results in energy saving. Virtualization is an abstraction between a user and a physical resource which provides the illusion that the user interacts directly with the physical resource. [5] A virtual machine is a software implementation of a machine that executes related programs like a physical machine [4]. A hypervisor, on the top of the physical machine, allows for the concurrent execution of multiple virtual machines.

Cloud services is housed in state-of-the-art data centers, which have high deployment operational costs and huge demand of energy amounts. As Cloud Computing rapidly continues to gain popularity, data storage and computational needs increase in accordance, resulting in a raise of energy consumption. It is indeed that consumption which can potentially damage the quality of service in Cloud Environments. It is critical, as well, to pay attention on energy consumption. In the ITC area, energy aware initiatives are recently classified under the term Green Computing. [3]

Although current processors are featured with built-in power saving techniques, CPU remains the main power consumer. The aim of our research is to determine how CPU utilization scaling, affects the total power consumption and the corresponding performance in large-scale server systems, such as cloud environments. The analysis which follows, is based on results that derived after experiments on DAS-4 clusters. DAS-4 consists of more than a hundred of nodes and is also partly cloud-enabled. Covering three aspects: power consumption, CPU utilization and execution time first, our initial approach is to evaluate the Cluster Environment and subsequence the Cloud Environment.

Research Question

- How to evaluate the trade-off between energy and performance in DAS-4?
- How to correlate performance and energy consumption in Cloud Computing Systems?

1.1 Power Management Techniques

In this paper we will initially evaluate the trade-off between performance and power consumption in DAS-4 Cluster nodes. Distinguish among the terms of power and energy is really important, not only for the optimal maintenance of a data center, but especially for applying power management techniques efficiently. Power is defined as the rate at which electrical energy is transferred by a circuit (reference to circuit) with watt (W) being SI unit, while energy is the consumption of power in a period of time and is measured in kilowatt hours (kWh). One, by knowing this exact difference, can make the correct decision. An energy-saving technique, such as suspending idle servers during the night, will definably reduce the electrical bill, but not the peak power demand and therefore neither the cost of the appropriate cooling system.

The total power consumption in CMOS (Complementary Metal-Oxide-Semiconductor) technology is given by the sum of static and dynamic power consumption. Static power consumption, which is also referred to as leakage power, from the fact that occurs due to the leakage current, causes a server to consume a significant amount of power even in the idle period [11]. Various techniques, such as cycle-level simulation, and tools, such as SoftWatt [12], exist for Static Power Management (SPM).

Many techniques have been developed to achieve or improve energy efficiency in Cloud Environments. Temporarily shutting down idle physical machines or sleep scheduling mechanisms are some simple power-aware policies for virtualized data centers that administrators can apply, thus contributing to conserve energy. Energy is consumed in switching, transmission, data processing and data storage. Awano Y. and Kuribayashi S. [4] mention that it is critical to prevent an increase in power consumption, not independently, either in data centers, or in communication or in power networks, by taking coordinated measures. They propose the installing of WAN accelerators as part of cloud resources, which can reduce significantly the communication time and therefore to save power. Another optimization strategy is using Dynamic Voltage/Frequency Scaling (DVFS) which adjust the CPU power according to the workload. However the scope of DVFS optimization is limited to CPUs [6].

Dynamic power consumption, which mostly is due to the charging and discharging of capacitance, contributes greatly to the total power consumption [7]. Minimization of dynamic power can be achieved by reducing supply voltage, eliminating needless computation, switched capacitance, and frequency at which the logic is clocked. Respectively, there is a plethora of Dynamic Power Management (DPM) techniques, such as Dynamic Voltage Frequency Scaling (DVFS), which aim to reduce the dynamic power.

DVFS is applied at runtime, reducing the frequency at which CPU operates, according to the system load. Since clock frequency affects also the required voltage levels in the same direction, the power saving in a DVFS-enabled cluster are, indeed, far from negligible. However, relevant researches [20], have shown that in the presence of virtual machines, DVFS is not efficient, as in case that the number of nodes is fixed, power consumption increases as the number of virtual machines increases.

Turbo Boost is another dynamic feature introduced by Intel, which has the ability to increase the clock speed of each core individually when the processor detects that it is below its power, temperature and current limits, in order to get more performance from the chip. This adaptive per core overclocking makes, indeed, a valuable performance enhancement. Unfortunately in the case of Turbo Boost Technology, the benefits are not delivered without costs. Charles et al. [22] using various benchmark applications from the SPEC CPU2006 suit with diverse qualities, run them either individually or in groups. They found that in a Turbo Boost enabled environment the execution time can be reduced, on average, up to 6, but this gain in performance has also negative impact in power saving, increasing the energy consumption up to 16%.

2 Related Work

Several research efforts have been done in profiling and analysing the energy consumption in Grids and Cloud Environments.

Bruneo et al. [3] presented an extended analysis of energy consumption in cloud computing, considering both public and private cloud environments. Including three different cloud service models, Storage as a Service, Software as a Service and Processing as a Service, their analysis was aimed to evaluate the energy consumption under a set of different conditions. (software as a Storage few users per server, Storage as a Service high Download rates, etc.) In fact, understanding and modelizing the energy consumption of large-scale infrastructures, is a far from easy task [8] due to the non linearity relation between energy consumption and the loads of CPU, disk and network. The fact that a node's power consumption can also be influenced by its position on the rack, namely depends on various and numerous factors complicates, the matters even more.

Virtualization, one of the most important key characteristics of Cloud Computing, raises additional challenges in the field of power metering. Pre-existing power monitoring and metering solutions, such as PDUs, which measure energy at the outlet level, are not longer suitable in a VM environment, for the obvious reason that it's impossible to connect a VM with a power measurement instrument.

Joulemeter [9] is designed to meter VM power by tracking each hardware resource, which the VM uses, and converting the resource usage into power usage based on a power model for the resource.

Chen Qingwen et al. [10] utilizing the three power metrics, i.e. power, power efficiency and energy, built a two-dimensional linear weighted power model for representing the behavior of single work node, including the contribution from the most power-consuming components, i.e. CPU, memory and HDD, to its total power consumption.

It is worth mentioning that they performed all the power benchmarks on the same infrastructure as we performed ours.

Chen Feifei et al. [6] recently presented an energy consumption model in which a single task, running in a Cloud Environment, is looked upon as the fundamental unit for energy profiling. For each task, energy consumption is tightly coupled with task workload. Having focused on storage, computation and communication resources, they propose a further division of energy consumption into two parts; fixed energy consumption (energy consumed during idle time) and variable energy consumption (additional energy consumed by Cloud tasks) respectively.

Besides energy consumption, understanding the performance of a Cloud Environment is equally important and crucial. Except for Cloud Providers, performance is the major decisive factor for cloud customers, which expect to receive a certain level of performance as specified in Service Level Agreement (SLA). Even minor performance improvements translate into huge cost saving [14]. Several efforts have been made to evaluate the performance in Cloud Environments.

Tudoran et al. [15] utilized four metrics within their synthetic benchmarks: data access performance, computation speed, variability and cost model to compare and evaluate performance and monetary cost between public and private cloud deployments. They conducted experiments ,which were focused on scientific HPC (High-Performance Computing) applications, resulting private clouds to show a better computing performance while public clouds to be more cost efficient.

A few systems have been also developed for performance assessing. C-Meter is a portable, for the reason that is platform-independent implemented, [16] and extensible framework for Cloud Environment performance evaluation. Allowing the generation and submission of both real and synthetic workloads to computing clouds, can be used to compare the performance between different configurations or different scheduling algorithms.

RTP (Real Time Performance) [17] is another approach which aims to predict a personalized real-time performance information for cloud components effectively. The motivation of aforementioned research stemmed from the fact that the overall performance of a cloud application may improve by the replacing poor performing components with better ones.

3 Approach

Our approach in this paper is to compare information about workload taken into account metrics obtained from power-monitoring tools. For this comparison, data must be extracted from two different source systems. The first one, Bright Cluster Manager which is an application built on top of Linux, is responsible for providing a range of usage level information from nodes in the cluster. For our experiments, the main metrics to be used are CPU Usage, CPU Idle and Memory usage. The second system is a database with all power and energy metrics collected from Racktivity Power Distribution Units. Basically, to run our evaluation experiments, the information from these two systems must be correlated to build an estimative about the levels of CPU utilization and the energy which is being consumed at that moment. The experiments were executed in the existing environment known as DAS-4.

The above approach was based on the hypothesis that one of the experimental tests might offer a greener result.

4 Experimental Environment

Our analysis were generated by means of experiments over an existing environment known as DAS-4 (The Distributed ASCI Supercomputer 4) [19]. This environment is a six-cluster wide-area distributed system designed by the Advanced School for Computing and Imaging. The purpose of DAS-4 is to provide a common computational infrastructure for researchers within ASCI, who work on various aspects of parallel, distributed, grid and cloud computing, and large-scale multimedia content analysis. The following institutes and organisations are directly involved in the realization and running of DAS-4:

VU University, Amsterdam (VU)

Leiden University (LU)

University of Amsterdam (UvA)

Delft University of Technology (TUD)

The MultimediaN Consortium (UvA-MN)

Netherlands Institute for Radio Astronomy (ASTRON)

The six-clusters are also organized as six head nodes(file servers). These file servers are organized as following:

Cluster	Head node	Compute nodes
VU	fs0.das4.cs.vu.nl	node001-node075
LU	fs1.das4.liacs.nl	node101-node116
UvA	fs2.das4.science.uva.nl	node201-node218
TUD	fs3.das4.tudelft.nl	node301-node332
UvA-MN	fs4.das4.science.uva.nl	node401-node436
ASTRON	fs5.das4.astron.nl	node501-node523

4.1 Cluster Configuration

DAS-4 Cluster is composed by roughly 200 compute nodes divided across the six previous mentioned locations. All cluster nodes have dual-quad-core CPU (primarily SuperMicro 2U-twins with Intel E5650 CPUs), 24 GB memory, 30TB HDDs and roughly 1TB of storage. [10]

About the physical layout, servers in each DAS-4 site follows the same specification and model. These servers are twin like, which were designed to be installed on top of a server rack using U2 space of the chassis. Each server provides two physical nodes and are connected by a single energy source. This means that, if one of the cable is plugged off, both nodes in that server will be turned off. The current setup of the DAS-4 server can be illustrated as the following:

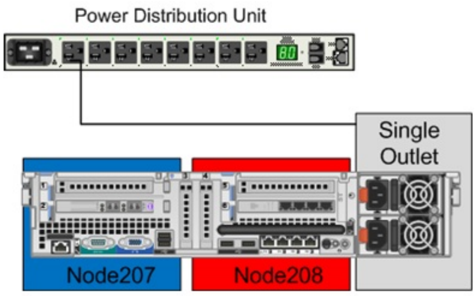


Figure 1: Current Setup

4.2 Cloud Configuration

Only a few nodes on the DAS-4 Cluster provide virtualization, specifically on DAS-4/VU and DAS-4/Delft. Virtualization basically runs on top of physical nodes using their resources to create virtual machines on demand. OpenNebula is the virtualization manager platform used by DAS-4, and is configured to create virtual machines in arbitrary nodes when requested by the user.

The physical node selected to run a new virtual machine is assigned based on resource allocation of the Cloud environment. This means that, with the current configuration and version of OpenNebula, a user can request to create a VM directly in a specific node. However, is not possible to provide a mechanism to provide isolation from concurrent resource share. Logically, this idea to select a specific node to execute a request turns to be against the Cloud perspective to be transparent for the final user.

Another issue using the current DAS-4 Cloud-enable environment to perform our experiment, was the concurrent resource share with other users. To execute the same experiment, the entire node with virtualization capabilities was necessary to be idle, and also the other node in the same PDU outlet. This implies the stop of 2 out of 4 available nodes which offer virtualization.

Considering the environment limitations, a scenario with virtualization of nodes is suggested to approximate the Cluster and Cloud environment. Namely, a node providing virtualization could create two local virtual machines to approximate the same energy bounding (single energy outlet). This indeed shows that the current setup of the DAS-4 Clusters could be interpret as a Cloud environment, with the same idea of abstraction. The figure below can illustrate this setup:

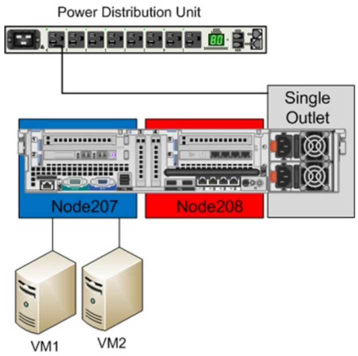


Figure 2: Environment Approximation

4.3 Metrics and Workload

Experiments to evaluate performance and energy trade-off require collecting specific data about multiple sources. Data concerning levels of utilization, non-utilization and memory were collect to build a knowledge base about performance. Regarding energy metrics, information about Power (W) is the main concert to run analysis upon.

Metric	Extraction method
execution time	reported by the job which was running in means of response
power consumption	python scripts were used to extract the power values from PDUs
energy consumption	python scripts were used to extract the energy values from PDUs
CPU load	python scripts were used to extract CPU loads from Bright Cluster Manager.

Table 1: benchmark metrics and their extraction method

The purpose of our analysis is the measurement of power consumption in a cloud environment, in terms of CPU utilization, and subsequent the evaluation of the trade-off, with the corresponding performance. To this end, we characterize and analyze the results of all experiments that have been carried out, using four metrics: execution time, power consumption, CPU Load and energy consumption. For selecting the above measurement metrics, their characteristics, such as linearity, reliability, repeatability and consistency were considered. We believe that these metrics meet our requirements, allowing for accurate comparisons, thus leading to correct conclusions. For extracting measurements, customized scripts were used to read the values from hardware monitoring tools (i.e. Racktivity PDUs) and software monitoring tools (i.e. Bright). Table 1 defines these metrics and for each one of them describes the way in which their corresponding values were derived.

4.4 Cluster Manager

Bright Cluster Manager is a powerful management software stack which allows the monitoring of different metrics in a large scale cluster environment. To retrieve data about CPU loads and execution times, we used script queries which extract data from Bright Cluster Manager. This manager provides roughly all necessary information about CPU Usage, CPU User, CPU Idle and Memory, to track processing state. The figure 3 shows the layout of Bright Cluster Manager:

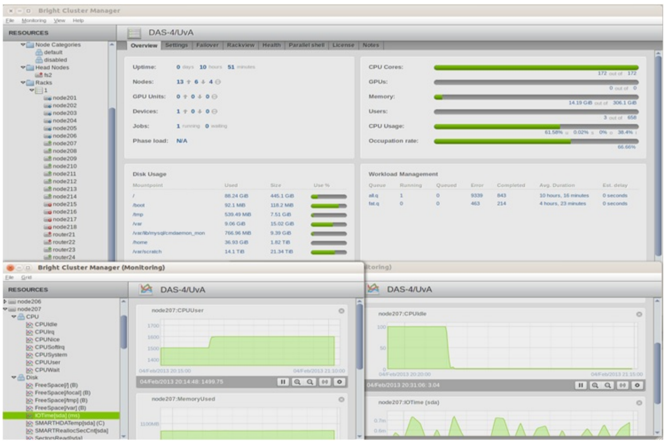


Figure 3: Bright Cluster Manager layout.

Bright was already installed in the nodes of DAS-4 cluster. Accessing all the corresponding heads nodes we could collect all necessary information about CPU Usage and Memory Usage.For all the tests we performed, we collect the CPU data and Memory Usage every 30sec and 120sec respectively.

4.5 Racktivity PDUs

To extract power (W) and energy metrics (Kwh) we use Ractivity PDUs which are connected in the set of the nodes on which we performed our experiments. Each of these Racktivity PDU provides 8 ports to be connected to. In case of DAS-4, the regular nodes are setup in pairs, that combined share either one power supply (UvA Cluster) or two redundant power supplies (VU Cluster). The power unit in the server (chassis) requires both nodes to be plugged in. This implies that anytime, in each of the PDUs outlets, we are able to monitor the power behavior of a pair of nodes, but never the power behavior of one node separately.

To evaluate the trade-off between performance and energy consumption in a Cloud Environment, our initial approach consists of a series of experiments performed in a Cluster Environment. The fact that we need to asses this trade-off for each node, while the retrieved power data comes combined for a pair of nodes, allows us to treat each physical machine with two nodes as a host node with two virtual machines running on it.

4.6 Challenges

In this scenario, information provided by the PDU is entangled, and cannot be used to exactly determine how much power one of the nodes is consuming. This occurs because just one outlet data is provided for the same server with two nodes.

With this entangling feature in context, one can make use of it to calculate the amount of energy of a node when the other node is in the idle state.

By taking this concept in mind, experiments to evaluate the approximate energy consumption can take place when running benchmark tools. Basically, these benchmark tools were executed oscillating the amount of CPU Usage and CPU Idle with the current energy utilization.

Power (W) is the main metric that is going to be taken into account when evaluating the energy consumption of the nodes. Only VU and UvA head nodes are connected to PDU's providing metrics about energy. For this reason, only these two sites will be used for our experiments.

For a final understanding about the Cloud environment, the initial phase is to analyse and evaluate the trade-off in the Cluster perspective. To correlate the results from the Cluster to the Cloud scenario, the entangling aspect can be taken into account. One can argue that, this specific situation of having a server with two nodes providing only one outlet, can be interpreted also as a Cloud node with 2 Virtual Machines.

5 Experiments Results

For the experiments only nodes in the same server were used. The outlet information from the PDU is common for both nodes. This scenario explores the entangling constraint with the system utilization.

Based on the facts about the current environment, different tests were created to experiment possible scenarios that could support our hypothesis about the performance and energy trade-off. The objective is to determine if any of the experiments can be classified as greener. Namely, if one of the experiments consumes less energy with the same or higher performance.

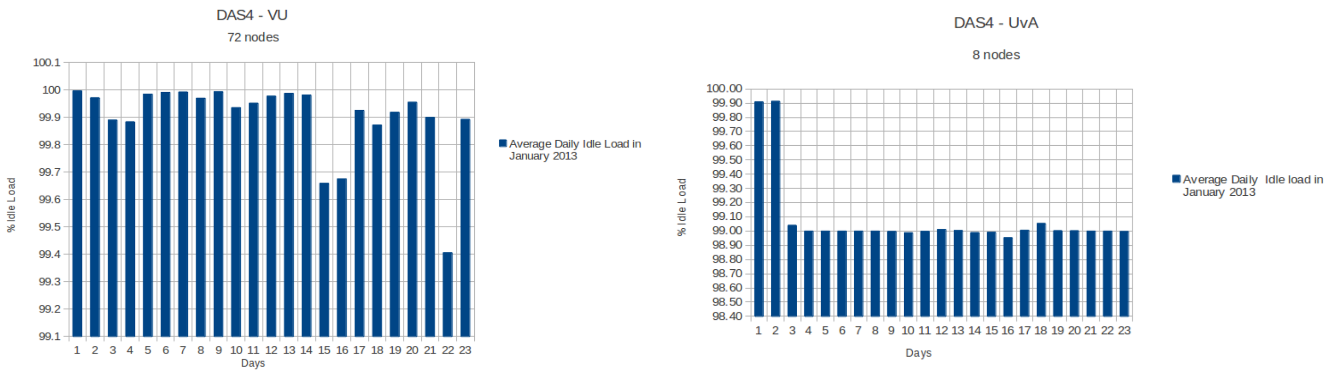
5.1 Idle State

Before conducting any experiment, we monitored the nodes when they were up but inactive, in order to measure the power consumption in idle state. Since nodes are combined in pairs in the power supplies, thus rendering impossible to monitor the power consumption of each one separately, the P_{idle} for one node was defined as:

$$P_{idle} = \frac{P_{idle_n} + P_{idle_{n+1}}}{2} \text{ ,where } n \text{ ,} n+1 \text{ adjacent nodes in the same physical machine.}$$

From the PDUs outlet we monitored the real power consumption in the idle state for several pairs of nodes, both in DAS-4/VU and DAS-4/UvA. The average power of a pair of nodes is $141W$ with a short fluctuation of 1.4%.

During this one month research, besides the experiments that were performed to evaluate the trade-off between power consumption and performance, information on the proportion of nodes that remain in idle state was also collected. These graphs below shows the percentage of idleness of all the nodes in a window of 24 days in the month of January for VU and UvA sites.



(a) (b)

Figure 4: Percentage of Idle State for (a)VU and (b)UvA sites

5.2 LinPack

The first set of our experiments involves the Linpack Benchmark, running in one and thereafter in both of the two nodes of the physical machine. Linpack tests the delivered performance of a computer in floating-point operations. We also considered that such a benchmark, which substantially solves a dense system of linear equations, will also give us precisely the peak of power consumption.

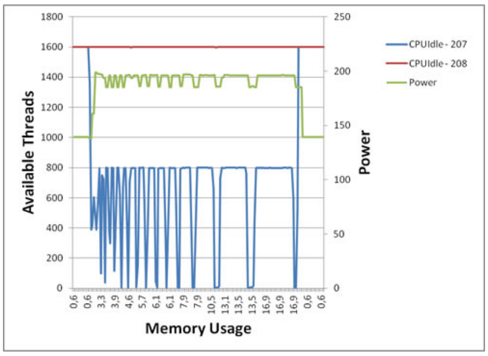


Figure 5: Linpack - Running on Node 207

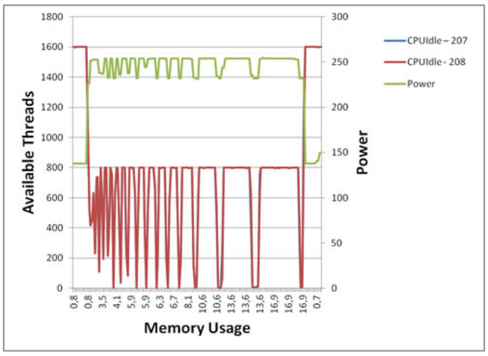


Figure 6: Linpack - Running on both Nodes

Figures 5 and 6 show the correlation between CPU load, Power and Memory Usage during the performance of Linpack. The need for increased memory, as the size of vectors is getting greater, is shown in x-axes. It's remarkable that the CPU load is periodically unstable, but the most interesting observation is that in these small periods, when a heavy load occurs, the power decreases. This happens in both cases, either the Linpack runs on one, or both nodes. The fifteen increasing valleys in CPU load, corresponds to the computations which performed in fifteen increasing size vectors. Since Linpack is a prevalent stress-testing benchmark, we ascribe the large spikes in load, which reach the time of 2 minutes at most, in the time it takes the fan speed algorithm to increase the fan speeds. While the load becomes ever more intense, the CPU temperatures rise above the usual range, however does not cool sufficiently, resulting a reduction in power consumption, in order to keep the permissible limits.

Node Running Linpack	Average Power Consumption(W)	Peak of Power Consumption (W)
node-207	188.52	199
node-207 and node-208	241.37	254.3

Table 2: Average and Peak values of Power during the Linpack execution on one and on both nodes.

In this research, through experiments, we seek to evaluate the trade-off between performance and power consumption in the fluctuation of CPU Load. The fact that Linpack lacks the configuration option to control the amount of resources that it uses, led us to find a more appropriate benchmark for our research.

5.3 Polyphase Filter

The Polyphase Filter script [13], it was possible to configure the workload of CPU utilization by setting up the number of threads allocated to run a specific job. Also, with this script, memory utilization is not expressible to influence the final result of power utilization.

The second cycle of our experiments in the cluster environment involves the running of Polyphase Filter benchmark for multi-core processors [13].

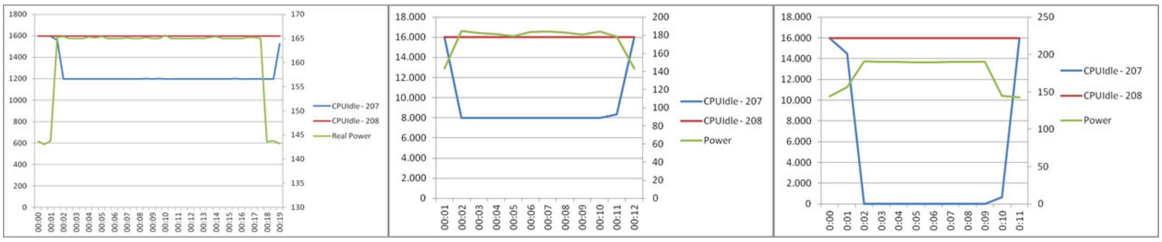
The polyphase filter is a method to isolate and down-sample individual channels. Given a signal as an input, polyphase filter splits it into N individual subsequences of M samples and pass each of these subsequences through a FIR (Finite Impulse Response) filter. DFT uses the outputs to effectively convert each channel to base-band. The polyphase filter for multi-core processors that we use as a benchmark, has been implemented to serve a system, composed of N Stations of N dual-pol channels. Thus, channels polarization defines two interleaved data streams. We believe that this benchmark fits our testing needs, since it is configurable, as regards the number of its runs and the used threads. That allowed us to intervene on the amount of workload and also control the CPU resources.

Table 3 shows the parameters that we are able to modify in polyphase filter, to generate our desired workload.

Adjusting the number of $N_{Samples}$, we can control how many times the polyphase filter runs, namely the amound of the workload. We define two different jobs; job1 and job2, so that job1 causes the double workload of job2. We treat every single job as a unit and measure the power produced by each of them under various rates of CPU utilization.

Parameter	Definition	Value
$N_{Stations}$	Number of Stations	64
$N_{Channels}$	Number of Channels	1024
N_{Taps}	FIR coefficients, also known as tap weights	16
$N_{Samples}$	Number of input samples per channel	varied
$N_{Threads}$	Numbers of threads used	varied

Table 3: Configurable parameters in Polyphase Filter



(a) 25% (b) 50% (c) 100%

Figure 7: Polyphase Filter - Single Node Running the job1 - Percentage of resource allocation

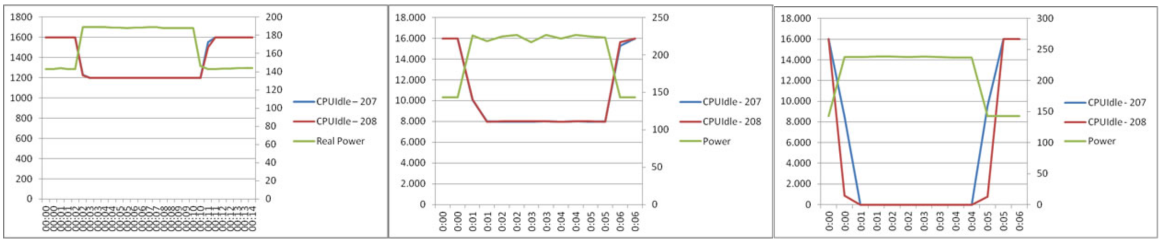
To precisely assess the performance, we set up a coupled in space experimental environment. The first triple of graphs in Figure 7 represents the behavior of the power consumption, while the job 1 is running on node-207 and the adjacent node-208 is idle. The performance can also be captured, with respect to our performance metric; execution time, as the available CPU resources for the node-207 decline. Table 4 demonstrates the execution time for each one of these experiments and the peak values which reached the power during them.

	Execution Time (sec)	**CPU Load node-207**	**CPU Load node-208**	**Peak of Power Consumption during execution (W)**
a	1028	25%	0%	165,4
b	587,6	50%	0%	184
c	530,3	100%	0%	190

Table 4: Executions times and peak values of power while job1 is running on a single node - the adjacent node is idle

Respectively, as shown on the triple of graphs in Figure 8, we examined the case where both nodes, on the same physical machine are up and running the job2. Again the graphs reflect the performance and its instantaneous correlation with the power consumption, as the available CPU resources for the node-207 and node-208 are being used. Table 5 summarises the execution times and the peak values of power for each case.

It is worth mentioning that in order to achieve accurate results, the above experiments were repeated at least four times. After the end of each experiment, we let the machines to return in idle state. This provided us with enough data, capable of returning an average, yet reliable result, and also preventing us from erroneous conclusions.



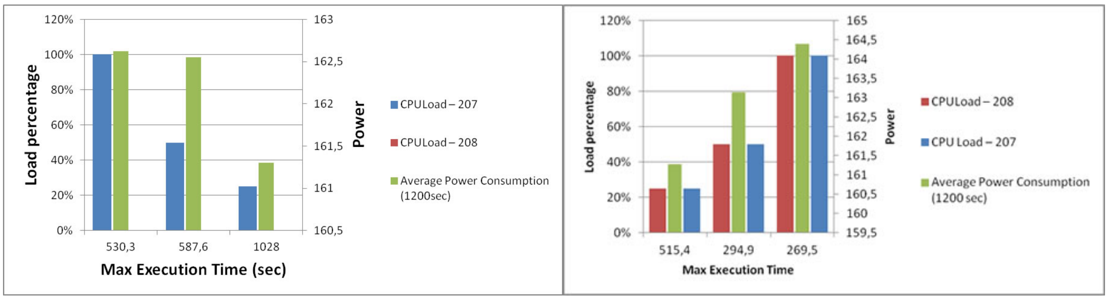
(a) 25% (b) 50% (c) 100%

Figure 8: Polyphase Filter - Two Nodes Running the job2 - Percentage of resource allocation

	Max Execution Time (sec)	**CPU Load node-207**	**CPU Load node-208**	**Peak of Power Consumption during execution (W)**
a	515,4	25%	25%	189
b	294,9	50%	50%	227
c	269,5	100%	100%	239

Table 5: Max Executions times and peak values of power while job2 is running on both nodes.

To evaluate the the trade-off between power consumption and performance for all the above cases, we built a coupled in time environment of 1200 sec. The selection of this specific interval of time made after taking account the longer execution time, which is 1028 sec. The bar graphs in Figure 9 indicate that, although power consumption and CPU usage are tightly associated in the same direction, finally in a short time interval, approximately equal to the longer execution time, the gains in power saving is almost negligible.



(a) (b)

Figure 9: Average Power Consumption in 1200 sec interval of time

Tables 6 and 7, which hold the results of our calculations, show that power consumption is almost the same at a given interval, which is greater than the time of the longer execution. Therefore, if there can not be achieved gains in power saving, the more optimal strategy is to achieve better performance, gaining the delivery of a response in the shortest possible time. Not allowing a single job execution to use all available resources, does not return power saving at all. Instead, allocating all the available resources returns better performance, in terms of execution time, without incurring higher power consumption.

Execution Time in seconds	**CPU Load 207**	**CPU Load 208**	**Avarage Power Consumption(1200sec)**
1028	25%	0%	161,30
587,6	50%	0%	162,54
530,3	100%	0%	162,62

Table 6: Execution time and the corresponding average power consumption when job1 is running on node-207

Max Execution Time in seconds	**CPU Load 207**	**CPU Load 208**	**Avarage Power Consumption(1200sec)**
515,4	25%	25%	161,27
294,9	50%	50%	163,14
269,5	100%	100%	164,39

Table 7: Execution time and the corresponding average power consumption when job2 is running on both nodes

Our intention was also to test the behavior of DAS-4, with respect to the same metrics as above, in the case where a node is already up and running a job, without using all its available resources, while a second job requests to be executed in that node. The implementation of the DAS-4 to execute a request is based on a queue system, which makes impossible another job to run on the same node when the first job execution has not finished. The queue system of DAS-4 uses a job execution scheduler. This scheduler invalidates concurrent job execution.

6 Conclusions

A few conclusions can be drawn from the results of this study. First, idle nodes in DAS-4 clusters always consume the same amount of power: 143W. The fact that no different value was measured indicates that the available power saving techniques were not effectively used, since 143W is the base value for the energy consumption for any given node in idle state.

Second, the study showed that better performance can be achieved without higher power consumption, even when executing a job using all available resources. Executing a job in a node with 100% of the resources allocated, which results in a shorter execution time, does not affect the total energy consumption in the end.

The experiments with the Cluster environment also show that one more improvement concerning the trade-off between performance and energy is to always select nodes which are available in the same physical server. Namely, in definite execution time job, better performance can be achieved using roughly the same amount of power.

The queue scheduler at DAS-4 Cluster, can only grant the execution of a job or request once the previous request has been finished. This means that the scenario to execute two jobs at the same time impossible. Therefore, other experimentations to evaluate the trade-off were impossible due to this policy.

Also, similar experiments over nodes providing virtualization through OpenNebula were impossible to be executed. The current configuration and requirements of the Cloud environment of DAS-4 could not offer a single allocation of resources. Without this policy, one could evaluate power consumption and energy of a single VM while the other VM stays in the idle state.

Considering the above mentioned restrictions and policies, and based on our results for the Cluster Environment, a Node Coordinator is suggested to optimize the current DAS-4 environment.

7 Future Work

Based on this conclusion, we propose that the DAS-4 Cluster can make use of a coordinator to grant nodes on demand. This Coordinator behaves as a job scheduler, granting the best available node to handle user requests. The optimal execution performance this Node Coordinator must fulfill, is to grant nodes which shares the same server and the same outlet, and use the available workload. This coordinator must know about the following informations:

- **Node mapping**;
  This node mapping contains the list of all nodes managed by a single head node;
- **Current workload**;
  For the workload, existing and current information can be extract from Bright Cluster Manager. Information provided from this manager is enough to build a base with CPU System workload, Memory Usage and other comparison metrics;
- **Current power consumption**;
  The PDU's provides energy information by means of outlets. The information contained in a single outlet is entangle, so one outlet provides data about two nodes;

To cover all six head file systems of DAS-4, we also propose that one node coordinator should be implemented in each of the head nodes. Each head node request is executed independently of the other head nodes. In the end, the coordinator will be the one responsible to take decisions on where to run a request, making it transparent for the end user by running on the best trade-off between energy and performance.

Another green feature that can be implemented with the role of a Node Coordinator, is to provide energy saving mechanisms by suspending nodes. In this feature the Coordinator will take decision to change the state of idle nodes to the suspended mode. Obviously this decision to suspend a node which was idle, depends on the system requirements and this should be taken into account when defining the maximum idle time limit. [24]

The figure below can illustrate the behavior of a node coordinator:

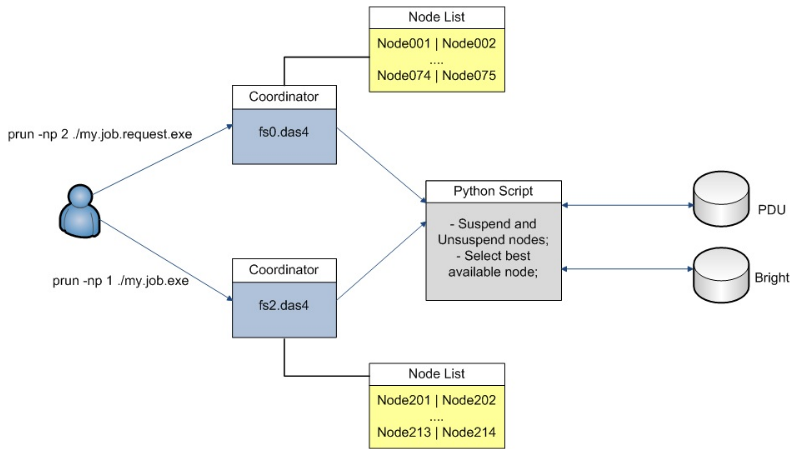


Figure 10: Node Coordinator

The figure 10 shows exactly the proposed scenario to implement the Node Coordinator. An user sends the request to reserve a node in the Cluster to therefore execute a job. Each head node coordinator, making use of a list which maps all the available nodes, handles the user request by means of the script. This script contains the current information about resource usage (workload allocation) and energy consumption. Based on this information, the script grants the optimal node to execute the user request. The script will also maintain the policy to change the state of the nodes from idle to suspend taking into account the system requirements.

References

[1] Y.Awano and S.Kuribayashi. "Reducing Power Consumption and Improving Quality of Service in Cloud Environments",26-28 Sept. 2012,Seikei Univ., Musashino, Japan

[2] Foster, Ian, et al. "Cloud computing and grid computing 360-degree compared." Grid Computing Environments Workshop, 2008. GCE'08. Ieee, 2008.

[3] Bruneo,D., Longo, F.,Puliafito, A. "Evaluating Energy consumption in a Cloud Infrastructure", Dipt. di Mat Univ. degli Studi di Messina, Messina, Italy, 20-24 June 2011.

[4] Wang, Shuen-Tai, Chin-Hung Li, and Ying-Chuan Chen. "Reducing Power Consumption in Cloud Platforms using an Effective Mechanism."

[5] T. Killalea, Meet the virts, Queue, vol. 6, no. 1, pp. 1418, 2008

[6] FeiFei Chen,Schneider, J.;Yun Yang;Grundy, J.;Qiang He, "An Energy Consumption Model and Analysis Tool for Cloud Computing Environments",Swinburne University of Technology,Melbourne, Australia, June 2012

[7] Texas Instruments, "CMOS Power Consumption and Cpd Calculation". 1997

[8] Orgerie, A-C., Laurent Lefevre, and J-P. Gelas. "Demystifying energy consumption in Grids and Clouds." Green Computing Conference, 2010 International. IEEE, 2010.

[9] Kansal, Aman, et al. "Virtual machine power metering and provisioning." Proceedings of the 1st ACM symposium on Cloud computing. ACM, 2010.

[10] Chen, Qingwen, Paola Grosso, Karel van der Veldt, Cees de Laat, Rutger Hofman, and Henri Bal. "Profiling energy consumption of VMs for green cloud computing." In Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on, pp. 768-775. IEEE, 2011.

[11] Liu, Liang, et al. "GreenCloud: a new architecture for green data center." Proceedings of the 6th international conference industry session on Autonomic computing and communications industry session. ACM, 2009.

[12] Gurumurthi, Sudhanva, et al. "Using complete machine simulation for software power estimation: The softwatt approach." High-Performance Computer Architecture, 2002. Proceedings. Eighth International Symposium on. IEEE, 2002.

[13] van der Veldt, K., van Nieuwpoort, R., Varbanescu, A. L., & Jesshope, C. (2012, June). A polyphase filter for GPUs and multi-core processors. In Proceedings of the 2012 workshop on High-Performance Computing for Astronomy Date (pp. 33-40). ACM.

[14] Ren, Gang, et al. "Google-wide profiling: A continuous profiling infrastructure for data centers." Micro, IEEE 30.4 (2010): 65-79.

[15] Tudoran, R., Costan, A., Antoniu, G., & Bouge, L. (2012, April). A performance evaluation of Azure and Nimbus clouds for scientific applications. In Proceedings of the 2nd International Workshop on Cloud Computing Platforms (p. 4). ACM.

[16] Nezih Yigitbasi, Alexandru Iosup, Dick Epema, Simon Ostermann, "C-Meter: A Framework for Performance Analysis of Computing Clouds." ccgrid, pp.472-477, 2009 9th IEEE ACM International Symposium on Cluster Computing and the Grid, 2009

[17] Zhang, Yilei, Zibin Zheng, and Michael R. Lyu. ”Real-Time Performance Prediction for Cloud Components.” Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORCW), 2012 15th IEEE International Symposium on. IEEE, 2012.

[18] CMOS Power Consumption and Cpd Calculation

[19] DAS-4 The Distributed ASCI Supercomputer 4[Online]Available: http://www.cs.vu.nl/das4/about.shtml

[20] von Laszewski, Gregor, et al. ”Power-aware scheduling of virtual machines in dvfs-enabled clusters.” Cluster Computing and Workshops, 2009. CLUSTER’09. IEEE International Conference on. IEEE, 2009.

[21] Racktivity PDU DataSheet.[Online]Available: http://www.racktivity.com/products/energyswitch.php

[22] Charles, James, et al. ”Evaluation of the Intel Core i7 Turbo Boost feature.” Workload Characterization, 2009. IISWC 2009. IEEE International Symposium on. IEEE, 2009.

[23] Fontn, J., et al. ”OpenNebula: The open source virtual machine manager for cluster computing.” Open Source Grid and Cluster Software Conference, San Francisco, CA, USA. 2008.[Online]Available:http://opennebula.org

[24] Alvarruiz, Fernando, et al. ”An Energy Manager for High Performance Computer Clusters.” Parallel and Distributed Processing with Applications (ISPA), 2012 IEEE 10th International Symposium on. IEEE, 2012.