The GREEN-NET Framework: Energy Efficiency in Large Scale Distributed Systems

Georges Da Costa[3], Jean-Patrick Gelas[1], Yiannis Georgiou[2], Laurent Lefèvre[1]
Anne-Cécile Orgerie[1], Jean-Marc Pierson[3], Olivier Richard[2], Kamal Sharma[4]

[1]INRIA RESO - Université de Lyon - École Normale Supérieure
46, allée d'Italie - 69364 LYON Cedex 07 - FRANCE ,
laurent.lefevre@inria.fr, {annececile.orgerie|jean-patrick.gelas}@ens-lyon.fr
[2] MESCAL, Laboratoire Informatique et Distribution (ID)-IMAG
ZIRST 51, avenue Jean Kuntzmann 38330 Montbonnot Saint Martin - FRANCE,
{Yiannis.Georgiou|Olivier.Richard}@imag.fr
[3] IRIT, Université Paul Sabatier
118 Route de Narbonne - F-31062 TOULOUSE CEDEX 9,
{Jean-Marc.Pierson|Georges.Da-Costa}@irit.fr
[4] Indian Institute of Technology Kanpur
kamals@iitk.ac.in

Abstract

The question of energy savings has been a matter of concern since a long time in the mobile distributed systems and battery-constrained systems. However, for large-scale non-mobile distributed systems, which nowadays reach impressive sizes, the energy dimension (electrical consumption) just starts to be taken into account.

In this paper, we present the GREEN-NET[1] framework which is based on 3 main components: an ON/OFF model based on an Energy Aware Resource Infrastructure (EARI), an adapted Resource Management System (OAR) for energy efficiency and a trust delegation component to assume network presence of sleeping nodes.

1. Introduction

The question of energy savings is a matter of concern since a long time in the mobile distributed systems. However, for the large-scale non-mobile distributed systems, which nowadays reach impressive sizes, the energy dimension just starts to be taken into account.

Some previous work on operational Grids [**?**] shows that grids are not utilized at their full capacity. We focus on the utilization and the energy analysis of experimental Grids by relying on the case study of Grid5000[**?**][2], a french experimental Grid. Based on this analysis, we propose the GREEN-NET software framework which allows energy savings at large scale.

Figure 1 presents the GREEN-NET framework with the main three components :

- an Energy Aware Resource Infrastructure (EARI) which collects energy logs from distributed autonomic energy sensors. EARI enforces Green decisions to the scheduler and requests some network presence decisions to the Network Presence Proxies. Moreover, EARI proposes some "Green advices" to the Grid end users;

- an adapted Resource Management System (OAR) which provides: a workload prediction module for automatic node shut down during cluster 'under-utilization' periods and a new PowerSaving type of jobs for device energy conservation.

- a trust evaluation component: when some nodes are switched OFF for energy reduction choices, this component evaluates and choses trusted target Network

[1]This research is supported by the GREEN-NET INRIA Cooperative Research Action: http://www.ens-lyon.fr/LIP/RESO/Projects/GREEN-NET/

[2]Some experiments of this article were performed on the Grid5000 platform, an initiative from the French Ministry of Research through the ACI GRID incentive action, INRIA, CNRS and RENATER and other contributing partners (http://www.grid5000.fr)

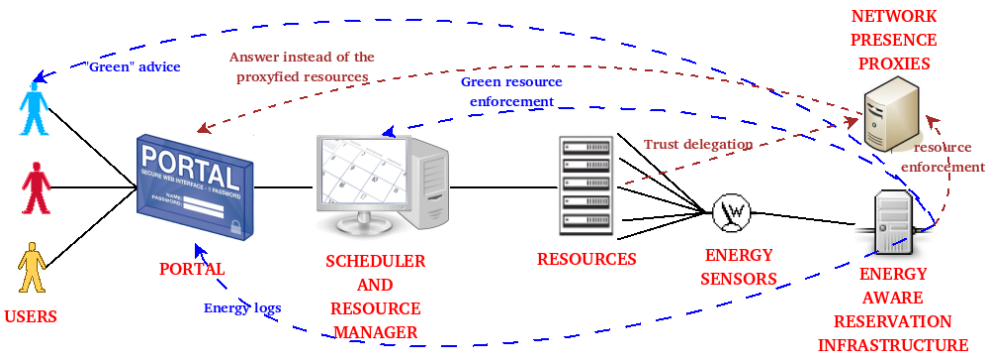

Figure 1. The GREEN-NET framework

Presence Proxies where basic services can be migrated.

Section 2 presents our approach on understanding the usage in large scale experimental Grids over a one year period. In the next sections, we describe the GREEN-NET framework with focusing on three main components: EARI (section 3), Adapted Scheduler (section 4) and Trust Delegation framework (section 5). We link our approach with some related works in section 6. Section 7 concludes this paper and presents some future works.

2. Understanding Large Scale Experimental Distributed Systems usage

Lots of computing and networking equipments are concerned by overall observations on the waste of energy: PCs, switches, routers, servers, etc, because they remain fully powered-on during idle periods. In a grid context, different policies can be applied depending on the level we want to make savings: node level, data center level or network level.

The Grid5000 platform is an experimental testbed for research in grid computing which owns more than 3400 processors geographically distributed in 9 sites in France. This platform can be defined as a highly reconfigurable, controllable and monitorable experimental Grid equipment. Its utilization is specific : each user can reserve in advance some nodes and use them as super user in order to deploy his own system image. The node is entirely dedicated to the user during his reservation. So Grid5000 is different from an operational Grid (exclusive usage, deployment, etc.), but the energy issue is still the same and we can propose solutions which fit for both experimental and operational Grids as well.

Currently, we are monitoring 18 nodes on Grid5000: 6 in Lyon, 6 in Toulouse and 6 in Grenoble. The electric consumption of these nodes is available in live on line with some graphs as shown in Figure 3. We collect one data per second for each node and we provide different views (hour, day, week, month and year) for each node. This monitoring allow us to conduct power experiments between these three sites.

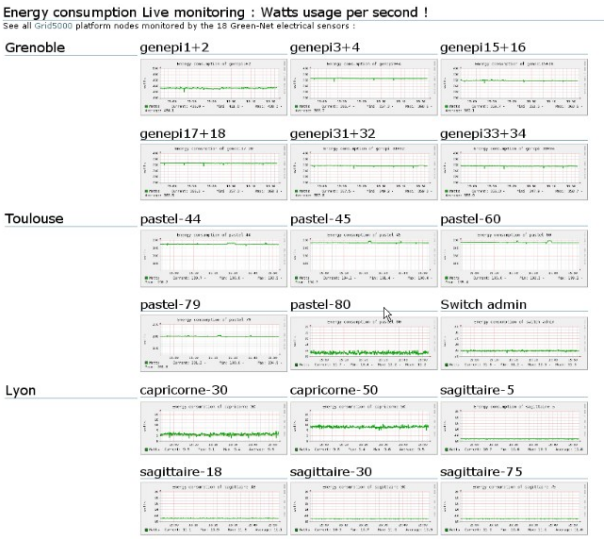

Figure 3. Monitoring of 18 nodes sensors

3. EARI: an ON / OFF model for benefiting of gaps in usage

3.1. EARI model

In order to reduce the electric consumption, we have designed an Energy-Aware Reservation Infrastructure (EARI), which is detailed in [**?**]. The global idea is to design an infrastructure that works like a garbage collector: it

Site	nb of reservations	nb of cores	nb of core per reservation	mean length of a reservation	real work
Bordeaux	45775	650	55.50	5224.59 s.	47.80%
Lille	330694	250	4.81	1446.13 s.	36.44%
Lyon	33315	322	41.64	3246.15 s.	46.38%
Nancy	63435	574	22.46	19480.49 s.	56.41%
Orsay	26448	684	47.45	4322.54 s.	18.88%
Rennes	36433	714	54.85	7973.39 s.	49.87%
Sophia	35179	568	57.93	4890.28 s.	51.43%
Toulouse	20832	434	12.89	7420.07 s.	50.57%

Figure 2. Grid5000 usage over the 2007 period7

switches off the unused nodes, and switches them on again when a user makes a reservation on them. A reservation is a reservation of some resources by a user during a certain period of time.

Our infrastructure is based on three ideas:

- to switch off the unused resources;
- to predict the next reservation;
- to aggregate the reservations.

We want to predict the next reservation in order not to switch off resources that will be used in a really near future. Indeed, such a behavior would consume more energy than keeping the resources powered on. We define T_s as the minimum time which ensures an energy saving if we turn off a resource compared to the energy we use if we let it powered on. Figure 4 illustrates this definition.

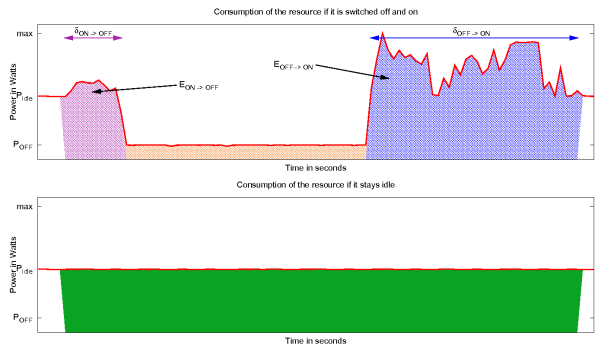

Figure 4. The definition of T_s

The top part of the figure presents the case where the resource is switched off and then booting when required. The bottom part of Figure 4 shows the case where the resource is left powered on, but idle. Then, T_s corresponds to the time that makes equal the colored area of the two solutions.

We need to define an imminent reservation: it is a reservation that will start in less than T_s seconds in relation to the present time. So the infrastructure maintains an agenda of the reservations.

Our prediction models are described in [**?**]. They use average values of the last few reservations. We also aggregate reservations in order to avoid frequent switching between off and on. Aggregate means that we will try to "glue" the reservations in terms of time and resources. So, when a reservation will arrive, we will try to place it after or before (in terms of time) a reservation which is in the agenda. In order to do that and by assuming that the user gives a wished start time, we have defined six different policies:

- *user*: we always select the solution that fits the most with the user's demand (we select the date asked by the user or the nearest possible one);
- *fully-green*: we always select the solution that saves the most energy (where we need to boot and to shut down the smallest number of resources);
- *25%-green*: we treat 25% of the submissions, taken at random, with the previous *fully-green* policy and the remaining ones with the *user* policy;
- *50%-green*: we treat 50% of the submissions, taken at random, with the *fully-green* policy and the others with the *user* policy;
- *75%-green*: we treat 75% of the submissions, taken at random, with the *fully-green* policy and the others with the *user* policy;
- *deadlined*: we use the *fully-green* policy if it doesn't delay the reservation from the initial user's demand for more than 24 hours, otherwise we use the *user* policy.

So, we expect that the *fully-green* policy is the most energy efficient (ie. consumes the less).

3.2. Experimental evaluation

To evaluate our model, we conduct experiments based on a replay of the year 2007 traces of Grid5000 (see section 2).

Figure 5 presents our results for 4 different sites, so 4 different workloads. These diagrams show the percentages of energy consumed with EARI compared to the present consumption (when all the nodes are always fully powered

on). To do these experiments, we have used the consumption measurements done previously with our wattmeter. The theoretical lower bound is what we have called *all glued*; it respresents the case where we glue (aggregate) all the reservations at the end of the year for example and let the nodes off for the remaining time. This lower bound is unreachable.

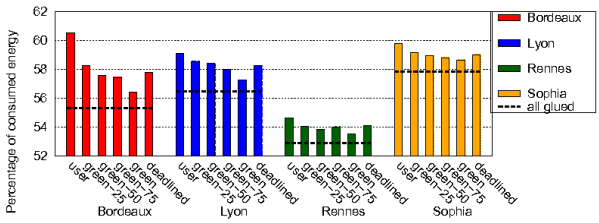

Figure 5. Results of EARI for 4 sites

We see that our *fully-green* policy is always the best and could lead to huge energy savings (almost 44% on average for these four examples).

4. Adapting a Resource Management System for energy efficiency

Another approach to deal with the management of energy upon a cluster or a grid is to provide a solution through the main software that is in charge of resources and their allocation upon the different jobs. This software usually called Resource Management System or Batch Scheduler provides the standard mean of communication between the administrator (utilization rules), the users (applications and workloads) and the resources of the cluster or grid (actual energy consumption).

OAR [**?**] is an open source Resource Management System implemented in ID-IMAG Laboratory. Initially developed as a tool for research upon the area of Resource Management and Batch Scheduling, this software has evolved towards a certain 'versatility'. It provides a robust solution, used as a production system in various cases (Grid5000 [**?**], Ciment [3]). Moreover, due to its open architectural choices, based upon high level components (Database and Perl/Ruby Programming Languages), it can be easily extensible to integrate new features and treat research issues [**?**].

To adapt OAR with energy efficient functionalities we decided to treat the problem with two complementary approaches.

Initially we have to deal with the waste of energy when the cluster is 'under-utilized' (functioning with no or few jobs to treat). This drives a need to create an automated system to manage the energy demand of the cluster. The system adapts to 'under-utilization' periods of the cluster and takes appropriate actions.

In parallel, we decided to deal with energy concious users and clever applications that are aware of which devices are going to be in use during the computation. Hence, OAR provides a way to specify the usage of specific node devices per job, so as to consume less energy.

Concerning related work upon energy efficient Resource Management Systems; It seems that two proprietary solutions LSF[4] and Moab[5] already provide adapted energy efficient techniques. From the opensource world, as far as we know, only SLURM[6] seem to provide an option for energy saving through the cpufreq command for changing the processor frequency.

4.1 Prediction based energy efficient scheduling

Energy demand in cluster environment is directly proportional to the size of the cluster. The typical usage of the machines varies with time. During daytime, the load is likely to be more than during night. Similarly, the load drastically decreases over the weekend. Ofcourse workloads can change upon different cluster configurations and utilizations. Energy saving can occur if this pattern can be captured.

Hence a need for a prediction model arises. Here, we explore this behavior of load cycles to power down nodes when idle time period is large. A past repository aids in maintaining the periodic load of the system.

Our prediction model is based upon an algorithm which scans for current and future workload and tries to correlate with the past load history. The Algorithm 1 explains a high level flow of events to decide to power off a specific number of nodes. The time window is one of the parameters used in the algorithm that is decided based on the cluster configuration.

Figure 6 shows the energy saving feature working on a cluster environment. The module considers the past history and the job queue to perform the energy conservation on the nodes.

4.2 PowerSaving Jobs

Nowadays users have become more energy concious and want to be able to control the energy consumption of the

[3] https://ciment.ujf-grenoble.fr/cigri

[4] http://www.platform.com

[5] http://www.clusterresources.com/solutions/green-computing.php

[6] https://computing.llnl.gov/linux/slurm/power_save.html

Algorithm 1 Prediction based Scheduling

N - Total nodes in the Grid
N_{down} - Nodes currently switched off
N_{up} - Nodes currently in active mode
N_{hib} - Nodes in hibernate mode
W - Lookahead time window(e.g say 1 hr)
Q - Queue of the Grid
$PastJob_{nodes}$ - Past history of peak node utilization of job arriving in Grid based on (hour,day) over W.
$\triangle$ - Buffer machines kept in case of sudden load

1: **if** (Q full for W) **then**
2: **if** (maxnodesusage(Q_{jobs})=N) **then**
3: $LookHibernate(Q)$
4: **end if**
5: **end if**
6: $N_1 \leftarrow N - maxnodesusage(Q_{jobs})$
7: $N_2 \leftarrow N_1 - PastJob_{nodes} - \triangle$ /* Look for nodes that can be shut down */
8: **if** ($N_2 > 0$) **then**
9: $SwitchDown(N_2 - N_{down})$ /* if $N_2 < N_{down}$ indicates nodes are already down */
10: **end if**

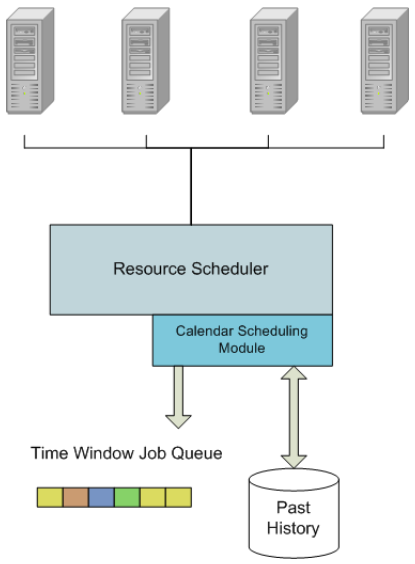

Figure 6. Prediction based Energy Saving Scheduler

Devices	Options	Description
CPU	cpufreq:min/max	The user can select the minimum or maximum frequency of the processor.
GPU	gpustate: <*state number*>	The user can provide a power state which defines the frequency of the GPU
Network	nw:on/off nwspeed: <*nwspeed*>	There may be more than one network card present in the system. The user can switch it on/off or provide a speed for the network card.
Hard Disk	hdd:standby/sleep	The user can spin down or power off the disk. Frequent switchovers could cause the problem of the hard disk.

Table 1. OAR PowerSaving Jobs Supported Devices

cluster during their computation. At the same time, applications can be programmed to provide monitoring functions if a device is not needed or if it can function slowly. Hence, a new type of jobs called 'powersaving' has been introduced to allow users and applications to exploit those new energy saving possibilities.

Our choices of the hardware devices that can be treated, were defined by the fact that they have to be either parameterized to function slower, consuming less energy, or provide the possibility of a complete power off. Thus the following table shows the supported devices for energy saving along with the relevant options for each one of them.

OAR supports different kind of jobs, like besteffort jobs (lowest priority jobs used for global computing [**?**]) or deploy type of jobs (used for environment deployment [**?**]). The implementation of the new powersaving type of job allows the user to control the device power consumption of the computing nodes during their job execution. The open architecture of OAR along with its flexibility permitted us to integrate this feature with a rather straightforward manner. Unlike most Resource Management Systems, in OAR there is no specific daemon running on the computing nodes of the cluster. Nevertheless, during the execution the server communicates with every node (participating in the job) where it can obtain root privileges and perform all the demanded power saving modifications. The specific device modifications are stored into the database as different device power states. At the end of the job all computing nodes return to their initial power states.

Experiments are on the way to measure the energetical gain of each power state of every device, considering real-life applications and workload conditions.

5. Trust delegation to support network presence

The On/Off model described in the previous section deals with computing nodes. Such strategy can be adapted for infrastructure services too. Depending on the current usage of the grid, site services such as the scheduler, the resource manager, visualization tools, etc... may need each a whole dedicated server or can be moved around and share a node with other services in order to shut down a server. For services with low reactivity requirements it is even possible to move them on other remote sites of the grid.

Figure 7 shows the required steps to move a service from a server to another one.

Before moving a service, it is important to choose carefully the new server. For instance, some servers can have an history of regular crashes and should be avoided. Due to the complexity of the structure of large scale grids, it is difficult to know where a service can be moved as each site does not always have information on all other sites composing the grid. Reducing the possible migration hosts to only directly known and trusted servers would reduce the achievable energy gains.

In [**?**] we define the Chameleon architecture that allows to provide nomadic users access to local resources, based on the trust path that can be established between the home domain of the user and the domain where the local resources resides. The process is based on the evaluation of a trust chain between domains and the propagation and aggregation of trust values along the chain. All the domains involved into the chain don't have to be connected directly (a Peer to Peer approach is used). We will not detail here the nuts and bolts of the trust value propagation but information can be found in [2]. One noticeable point of the approach is that it is fully distributed (no central server is needed), and the trust evaluation formula ensures that despite the local trust setting on each domain, the process still gives consistent trust value that can be compared and aggregated among the whole system. Trust values and trust chains are embedded in so-called `X316` certificates that allow the secure, non-repudiable and non-modifiable transmission of these data (indeed if a user could change the embedded trust values, the evaluation would be useless). The certificates are delivered to individual users during their roaming that present them to the visited domains to gain access.

We adapted this work to match the Green-Net architecture requirements. In our context, domains are individual nodes that may accept some migrating services (*i.e* users of our former work). The main difference we face with this adaptation is that no service is at first actually moving from one node to the next one, that enables the iterative construction of the trust chain between the source of the migration to a potential target node. A second difference is the understanding of the trust values: These can be understood in our context with traditional security meaning (if a node does not have a high confidence in another one, the trust value from the first to the second will be low; for instance a high value is given to all nodes at the same site, and lower values to nodes on remote sites), as a reliability meaning (if a node is not reliable and crashes often, then its trust value will be small for its direct neighbors), or any other metric that can be aggregated and propagated along a chain.

In our deployment, each node is hosting a trust service. Its role is twofold:

- It sets the trust values of a neighboring nodes logical set: the trust values represents the trust that one node is giving to a set of other nodes. This setting is automated to ease the production of the neighborhood.
- It manages requests and responses about trust values and trust chains in `X316` certificates.

A source node that wants to migrate a service must first construct a list of target nodes (Network Presence Proxies) for this service (potentially all the other nodes of the system, but that can be restricted to respect some performance, quality of services or availability criteria). Then, the node evaluates the trust it can have in each potential target, to select the most trustful one (*i.e.* the one with the best trust value).

The process of this evaluation is the following: the source node first contacts a local trust service. If this one knows the target (direct link), then the trust value is returned. In other cases, the local trust service sends requests in X316 certificates to the trust services running on the set of logical neighbor nodes. These will propagate the request among their neighbors until the target is found. When the target is reached, response to the request is forwarded back to the source site (remember that the trust path, thus the propagation path of the request is in the certificate).

It must be noted that several possible trust chains can exist between two arbitrary sites. Thus, several responses to the same request may come back to the source from different paths. Threshold on the acceptable trust value and timeout mechanisms ensure some limited (in time and space) but still consistent answers. The choice between the diverse returned trust values depends on the node settings: A paranoiac node will prefer to wait and to take into account the lower trust value, a trustful node will prefer the higher trust value while a hurried node will take the first response coming out to make a decision.

6. Related works

Although energy has been a matter of concern for sensor networks and battery constrained systems since their cre-

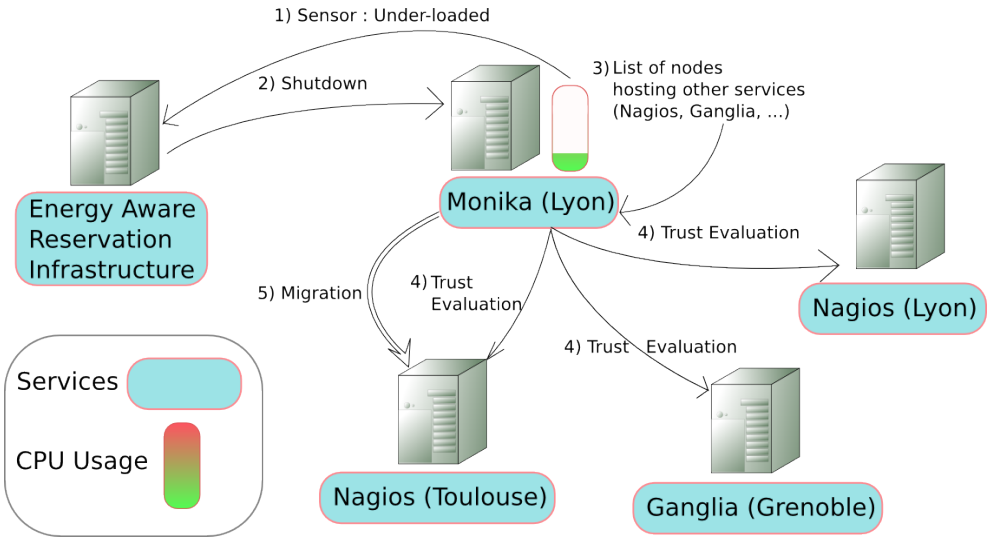

Figure 7. When a server is under-loaded, its services are moved to target trusted nodes (Network Presence Proxies) to shut down this server and reduce global energy consumption

ation, energy issues are recent for plugged systems. A first problem that occurs is how to measure the consumption. We have considered an external watt-meter to obtain the global consumption of a node. A different approach consists of deducing it from the usage of the node components, by using event monitoring counters [**?**] for example. Lot of other work on server power management based on on/off algorithm has been done [**?**]. The main issue in that case is to design an energy-aware scheduling algorithm with the current constraints (divisible task or not, synchronization, etc).

Concerning the adapted energy efficient techniques for Resource Management Systems. Techniques exist to low the node power according to workloads. In [**?**], a minimal set of servers is chosen to handle the requested load and rest of nodes are put in low power mode. Other techniques in [**?**], try to adapt the cluster based on the incoming load behaviour.

Concerning the Device Energy Conservation a known area where energy saving is present is by the use of DVS(Dynamic Voltage Scaling) in accordance with the job submission. Method adopted in [**?**] present a static algorithm to adjust the processor frequency based on the jobs. For hard disks, research work in [**?**] exploits the potential of disk spin-down whenever drives are not used for long period of time. Other approaches [**?**, **?**] to reduce disk consumption is by reorganization of data on the drive. Network cards can consume significant amount of energy. Modern machines especially laptops and servers, can have multiple network connectivity. Studies in [**?**, **?**] make use of multiple states of the network card for energy conservation.

Trust delegation for service migrations in grids already attracted large momentum [**?**, **?**]. These approaches does not evaluate the trust path between nodes participating in the migration and restrict their usage to the actual mechanisms of trust delegation. Our approach is complementary as it focuses on the trust evaluation part. To the best of our knowledge, trust propagation techniques have never been used in the context of service migration among a set of sites. For references in trust propagation and evaluation technics, the reader will refer to [**?**].

7. Conclusion and future works

This paper presents a first step of our work whose goal is to better understand the usage of large-scale distributed systems and to propose methods and energy-aware tools to reduce the energy consumption in such systems.

The GREEN-NET framework is based on 3 distinct software components:

- a ON/OFF model which includes prediction heuristics and *green* advice for the users and takes the decision to switch on or off the nodes;
- an adapted energy efficient Resource Management System ;
- a trust delegation framework which allows proxying techniques to ensure the network presence of the sleeping nodes.

We are currently monitoring 18 nodes on three different sites of Grid5000, but our medium-term goal is to monitor an entire site and our long-term goal is to monitor the whole platform. So we will have an entirely monitored grid and thus we will be able to conduct power experiment at a large scale Grid.

References

[1] N. Capit, G. D. Costa, Y. Georgiou, G. Huard, C. M. n, G. Mounié, P. Neyron, and O. Richard. A batch scheduler with high level components. In *Cluster computing and Grid 2005 (CCGrid05)*, 2005.

[2] F. Cappello et al. Grid'5000: A large scale, reconfigurable, controlable and monitorable grid platform. In *6th IEEE/ACM International Workshop on Grid Computing, Grid'2005*, Seattle, Washington, USA, Nov. 2005.

[3] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, and R. P. Doyle. Managing energy and server resources in hosting centers. *SIGOPS Oper. Syst. Rev.*, 35(5):103–116, 2001.

[4] Y. Georgiou, O. Richard, and N. Capit. Evaluations of the lightweight grid cigri upon the grid5000 platform. In *E-SCIENCE '07: Proceedings of the Third IEEE International Conference on e-Science and Grid Computing*, pages 279–286, Washington, DC, USA, 2007. IEEE Computer Society.

[5] D. P. Helmbold, D. D. E. Long, and B. Sherrod. A dynamic disk spin-down technique for mobile computing. In *MobiCom '96*, pages 130–142, New York, NY, USA, 1996. ACM.

[6] A. Iosup, C. Dumitrescu, D. Epema, H. Li, and L. Wolters. How are real grids used? the analysis of four grid traces and its implications. In *7th IEEE/ACM International Conference on Grid Computing*, Sept. 2006.

[7] R. Krashinsky and H. Balakrishnan. Minimizing energy for wireless web access with bounded slowdown. *Wirel. Netw.*, 11(1-2):135–148, 2005.

[8] J. Lee, C. Rosenberg, and E. K. P. Chong. Energy efficient schedulers in wireless networks: design and optimization. *Mob. Netw. Appl.*, 11(3):377–389, 2006.

[9] A. Merkel and F. Bellosa. Balancing power consumption in multiprocessor systems. *SIGOPS Oper. Syst. Rev.*, 40(4):403–414, 2006.

[10] R. Mishra, N. Rastogi, D. Zhu, D. Mossé, and R. Melhem. Energy aware scheduling for distributed real-time systems. In *IPDPS '03: Proceedings of the 17th International Symposium on Parallel and Distributed Processing*, page 21.2, Washington, DC, USA, 2003. IEEE Computer Society.

[11] A.-C. Orgerie, L. Lefèvre, and J.-P. Gelas. Chasing gaps between bursts : Towards energy efficient large scale experimental grids. In *PDCAT 2008 : The Ninth International Conference on Parallel and Distributed Computing, Applications and Technologies*, Dunedin, New Zealand, Dec. 2008.

[12] A.-C. Orgerie, L. Lefèvre, and J.-P. Gelas. Save watts in your grid: Green strategies for energy-aware framework in large scale distributed systems. In *14th IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, Melbourne, Australia, Dec. 2008.

[13] E. Pinheiro, R. Bianchini, E. V. Carrera, and T. Heath. Dynamic cluster reconfiguration for power and performance. pages 75–93, 2003.

[14] L. B. Rachid Saadi, Jean-Marc Pierson. Security in distributed collaborative environments: limitations and solutions. *Book Chapter in Emergent Web Intelligence, to be published by Springer Verlag (series Studies in Computational Intelligence)*, 2008.

[15] R. K. Sharma, C. E. Bash, C. D. Patel, R. J. Friedrich, and J. S. Chase. Balance of power: Dynamic thermal management for internet data centers. *IEEE Internet Computing*, 9(1):42–49, 2005.

[16] S. W. Son, G. Chen, and M. Kandemir. Disk layout optimization for reducing energy consumption. In *ICS '05: Proceedings of the 19th annual international conference on Supercomputing*, pages 274–283, New York, NY, USA, 2005. ACM.

[17] S. W. Son and M. Kandemir. Integrated data reorganization and disk mapping for reducing disk energy consumption. In *CCGRID '07: Proceedings of the Seventh IEEE International Symposium on Cluster Computing and the Grid*, pages 557–564, Washington, DC, USA, 2007. IEEE Computer Society.