A Power-Saving Strategy for Grids

Chris Develder, Mario Pickavet, Bart Dhoedt, and Piet Demeester

Ghent University – IBBT, Dept. of Information Technology (INTEC), IBCN
Gaston Crommenlaan 8, bus 201, BE-9050 Gent, Belgium
{chris.develder,mario.pickavet,bart.dhoedt,piet.demeester}@intec.ugent.be

Abstract. In light of recently stirred energy consumption concerns, we investigate the opportunities for power consumption reduction in Grids. Considering real life Grid traces, we note considerable fluctuations in load. We consider a peak load dimensioning strategy to derive how much servers to install in computational Grids. In lower loaded periods, there is a potential to save energy by dynamically powering on/off servers to address the actual demand for computational capacity. An appropriate scheduling and power-saving scheme can, under such lower-load conditions, considerably reduce energy consumption. The price paid is that some jobs are executed at sites more remote than closer powered-down ones. Yet, the resulting penalty in consumed bandwidth is rather limited and is expected not to cancel the power consumption advantage.

Key words: Grids, Green ICT, Power-awareness, Grid scheduling.

1 Introduction

Despite the environmentally friendly image of Information and Telecommunications Technology (ICT; cf. tele-working, e-commerce etc.), energy consumption does raise some concerns [4]. ICT usage (excl. production cost) accounted for about 8% of global electricity consumption in 2007 (forecast to 14% in 2020) [6]. Grids can help, since their capability of serving high computational and storage demands perfectly fits a thin client scenario, allowing for replacing PCs with less power hungry and longer living client machines [8] delegating jobs to (Grid) servers. Considering also the low power cost per bit/s of e.g. optical networks, Grids form an attractive paradigm. Yet, power-saving mechanisms can help to cut energy consumption further. Indeed, Grids are designed to successfully cope with the overall system's peak load, but in lower-load periods excess server capacity can be turned off. Since PC and data center equipment currently accounts for a large fraction (ca. 35%) of the ICT energy, reducing their power can have an important impact. Yet, shutting off servers causes jobs to be sent to more remote servers still powered on, and the resulting network traffic increase should be limited (network equipment consuming 14% of ICT power in 2007 [6]).

In this paper, we address energy consumption in Grids. We motivate our research in Section 2, including real world measurement data. In Section 3, a power-saving strategy for Grids is proposed, which is evaluated in the subsequent Section 4. Conclusions are summarized in Section 5.

2 Opportunities for green ICT in Grids

Grids originated from eScience dealing with large experimental data sets (particle physics, astrophysics etc.): to meet computational and storage demands, cluster centers were interconnected via networks to achieve a huge common resource pool to process the tasks (jobs). Yet, also business/consumer oriented applications can benefit from Grid infrastructure (see also cloud computing). A prime example would be thin client computing, where the end user's device is basically just an input/output device, delegating all processing to a (Grid) server. From an environmental viewpoint [7, 8], such a thin client system has quite some advantages.

In this paper we focus on reducing energy consumption of Grid server sites. Recent studies of the usage of Grid resources shows that the usage of a Grid site may significantly vary (between less than 20% to over 90%) on a day-per-day basis [5, Fig. 2]. Similarly, there is substantial variation in the hourly job arrival rates at sites of the EGEE/LCG Grid sites, as shown in Fig. 1. This means that there is an opportunity for energy saving mechanisms to automatically switch on and off servers (e.g. by power distribution units controllable via IP) to match the available server capacity to actual computational demands.

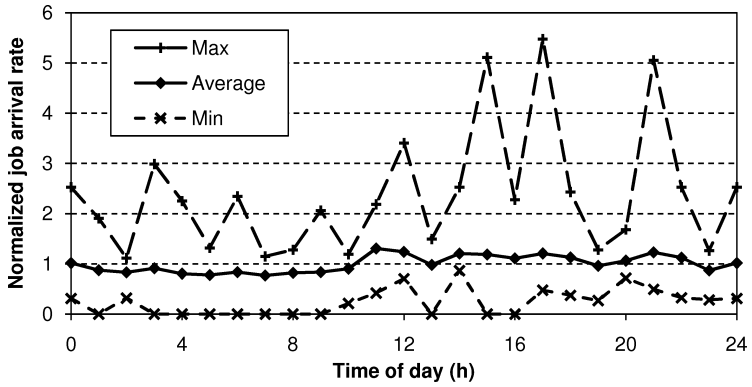

Fig. 1. Normalized hourly job arrival rates $\lambda_{h,norm} = \lambda_h / \text{avg}_h(\lambda_h)$, where λ_h is the arrival rate in a 1-hour period ($h = 0 \ldots 23$). Averages, minima and maxima are taken over 24 EGEE/LCG Grid sites' $\lambda_{h,norm}$ values. The maximal hourly arrival rate at a particular site can be up to 5 times larger than the average hourly rate.

In the following, we propose a simple power saving mechanism and assess it through a case study using realistic power consumption figures. As reported in [7], power consumption of a server can be captured by a linear model (Eq. 2). We assume multi-core processors, where each server CPU comprises C processors. A certain amount P_{idle} of power is consumed even if the server is doing nothing. For each of the cores, the additional power increases linearly with the load $\rho_c(t) \in [0, 1]$ up to a maximum of P_{core} for a fully loaded core. Network communication powers calls for an additional amount of power P_{netw} for each

unit of bandwidth $BW_{serv}(t)$ transferred. Each of the n_{start} times we power on a server server, we need to account for a certain startup time T_{start}, during which the server is not able to process a job (yet) and consumes power P_{start}. From power measurements on a modern Linux server, we derived the parameters in Table 1. These measurements (confirmed by similar results in [5]) illustrate that a server in use does not consume much extra power compared to an idle one (cf. small extra P_{core} when busy). Also, if jobs last long enough ($\gg T_{start}$), the penalty of starting up a new server will be limited.

$$E_{serv} = E_{start} + E_{op} \quad \text{with} \quad \begin{cases} E_{start} = n_{start} \cdot P_{start} \cdot T_{start} \\ E_{op} = \int_t P_{op}(t)\,\mathrm{d}t \end{cases} \tag{1}$$

$$P_{op}(t) = P_{idle} + \sum_{c=1}^{C} \rho_c(t) \cdot P_{core} + BW_{serv}(t) \cdot P_{netw} \tag{2}$$

Table 1. Power Measurements for a 2GHz Dual-Core AMD Opteron^{TM} Processor 2212 platform, running a linux operating system (Debian GNU/Linux 4.0).

Symbol	Value	Meaning	Symbol	Value	Meaning
P_{idle}	183.26 W	Idle CPU power.	P_{start}	201.20 W	Startup power.
P_{core}	19.53 W	Extra power needed to fully load a core.	T_{start}	89 s	Startup duration.

3 A dynamic power-scheme for Grids

In Grid systems, users do not really care where exactly their job ends up being executed. In view of energy consumption, it means we can choose to offload a job to a remote site with an available processor, rather than turning on a nearby server. To reduce energy consumption, a Grid system needs (a) a power-aware job scheduling mechanism, and (b) a power-saving strategy deciding when to turn servers on/off. The scheme combining (a) and (b) will be noted as *PA*.

For (a), existing scheduling algorithms that choose a free server out of a set S can be used: first consider only the servers S_{on} powered on, and only if none is available use the same algorithm to choose one among those S_{off} powered off (and turn it on). In this work, we consider a shortest-path strategy: for a job arriving at site i, the free server closest to i is chosen to execute it (thus minimizing network usage [3]).

For (b), we propose a straightforward approach: a server will be turned off a time D (termed power-saving delay) after a job finished, if at that time it is not running any other jobs. The reason for introducing this power-saving delay D is to avoid frequent switches between on- and off states: if within a reasonably short period of time a job arrives, it makes sense to leave the server on. (E.g.

if job arrivals follow a Poisson process, the chance that no job arrives during a period of duration D decreases exponentially with D.)

4 Case study

We adopted Poisson traffic—shown to accurately fit real world Grid-wide job traces [1]—and artificially generated job arrival rates at each site of the considered topology depicted in Fig. 2 (for peak load, each site i's job arrival rate λ_i was with 30% chance uniformly chosen in $[1\mu, 15\mu]$ and 70% from $[30\mu, 60\mu]$). The average job service time was set to $T = 1/\mu = 14249$ s, based on trace data gathered from the EGEE/LCG Grid.

The servers are assumed to be multi-core processors, with $C = 1, 2, 4$ cores per server. Jobs are assumed to occupy a single core completely (in Eq. 2, $\rho_c(t) = 1$ when core c is in use), and hence we will not schedule more than 1 job/core. (Note that in this paper we do not model job interdependencies for e.g. user tasks comprising multiple jobs.)

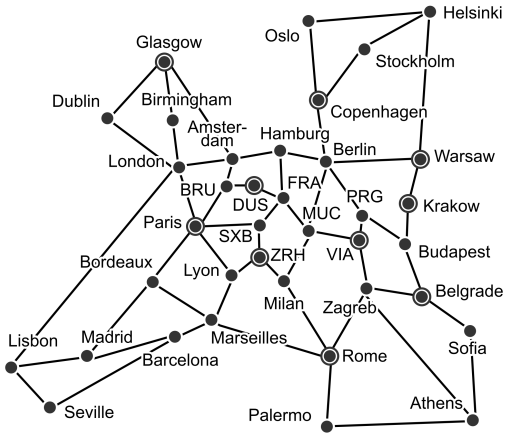

Fig. 2. A European backbone network topology [2], comprising 37 nodes and 57 links, with an average shortest path hop count of 3.62.

4.1 Server site dimensioning

We dimensioned the server sites for the aforementioned peak load. Given the assumed Poisson traffic, in a buffer-less system, the number of cores N_c required to achieve a tolerable probability L that a job cannot be accommodated, follows from solving the ErlangB formula for N_c:

$$L = \text{ErlangB}\left(\lambda, \mu, N_c\right) = \frac{\left(\lambda/\mu\right)^{N_c}/N_c!}{\sum_{n=0}^{N_c}\left(\lambda/\mu\right)^{n}/n!}. \tag{3}$$

The total number of servers N_s depends on the number of cores per server, and can be easily found as $N_s = \lceil \frac{N_c}{C} \rceil$. Targeting a 95% acceptance rate (hence $L = 0.05$), for the case study at hand, we find $N_s = 800, 400, 200$ for resp. $C = 1, 2, 4$ cores per server. We distribute these N_s servers over $K = 10$ sites (dimensioning studies show it is beneficial to choose a limited number of server sites [3]). Solving the Integer Linear Program (ILP) outlined in Fig. 3 gives the 10 best sites—aiming at minimizing network load—encircled in Fig. 2.

The N_s servers were spread over the K sites using the *prop* strategy from [3], limiting offloading to remote sites and hence minimizing network load. It sets the number of servers $N_{s,k}$ at site k to be proportional to the job arrival rates of its closest sites (with S_{jk} as defined in the ILP of Fig. 3):

$$N_{s,k} = \frac{\lambda_k^*}{\sum_i \lambda_i^*} \cdot N_s \quad \text{with} \quad \lambda_k^* = \sum_j \lambda_j \cdot S_{jk}. \tag{4}$$

Binary variables: $\begin{cases} T_j = 1 & \text{if and only if } j \text{ is chosen as server location} \\ S_{ij} = 1 & \text{if and only if } j \text{ is the server location closest to } i \end{cases}$

Given constants: $\begin{cases} H_{ij} = & \text{hop count from site } i \text{ to } j \\ \lambda_i = & \text{job arrival at site } i \end{cases}$

Objective: $\min \sum_i \sum_j \lambda_i \cdot H_{ij} \cdot S_{ij}$

Conditions:

$$\begin{cases} \sum_j T_j = K & \text{(only } K \text{ server locations)} \\ S_{ij} \le T_j, \quad \forall i, j & \text{(only send jobs to server sites)} \\ \sum_j S_{ij} = 1, \quad \forall i & \text{(simplifying assumption: all traffic to closest server)} \end{cases}$$

Fig. 3. Integer Linear Program (ILP) for choosing K server locations.

4.2 Energy savings

Realistically modeling power consumption, we used the measurement data of Table 1. To judge the potential of power saving techniques, we considered the Grid sites dimensioned for peak load ρ_{peak} as described above. Simulating varying fractions f of that peak load, we compared energy consumption between power-aware (*PA*, for varying power-saving delay D; see Section 3) and *non-PA* cases. We also considered the *ErlB* benchmark, dimensioning the server site capacity for the reduced load $f \cdot \rho_{peak}$ and $L = 0.05$, and using the *non-PA* strategy. In our discrete event simulation, we tracked the performance parameters of Table 2. The energy E_{serv} consumed by the Grid servers was calculated as follows (excluding network power): $E_{serv} = T_{proc} \cdot P_{core} + T_{on} \cdot P_{idle} + E_{start}$.

Table 2. Power parameters tracked in the energy savings simulation case study.

Symbol	Meaning
n_{start}	Number of times a server is started up.
T_{on}	Total on-time summed over all (multi-core) servers.
T_{proc}	Total processing time of jobs.

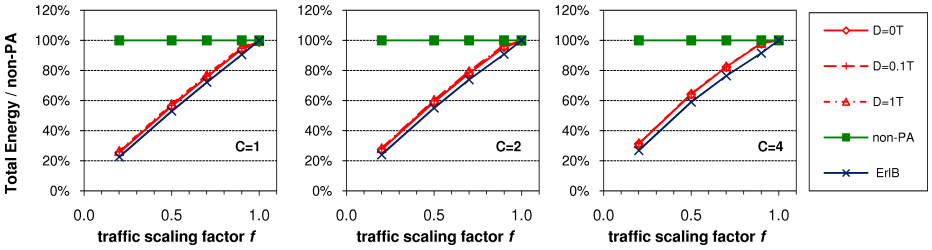

Fig. 4. A power-aware Grid deployment achieves almost the minimal amount of energy consumption given by the *ErlB* lower bound. (Note that the curves for different values of D overlap.)

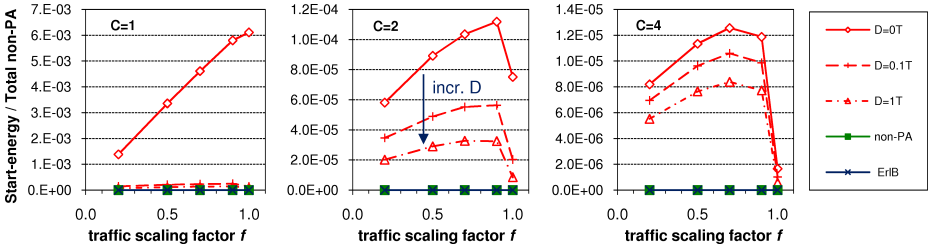

Fig. 5. By increasing the power-saving delay D, the start-energy E_{start} is reduced.

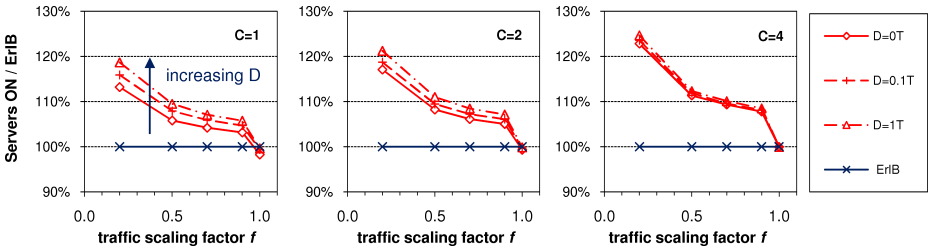

Fig. 6. A power-aware approach has slightly more servers powered on (averaged over time) compared to the *ErlB* lower bound, and slightly more so for larger values of the power-saving delay D.

Results plotted in Fig. 4, show that a power-aware Grid approach is able to reduce the energy to almost the lower bound given by *ErlB* ($E_{serv,PA}/E_{serv,ErlB} - 1 \approx$ 13-18%). The power-saving delay D has only a minimal influence—looking at the numeric values, we note a slightly higher power consumption for larger D (keeping idle servers longer on)—given the long duration of jobs ($T \gg T_{start}$).

The major influence of D lies in the number of times servers are powered on and hence start-energy E_{start}: Fig. 5 plots E_{start} relatively to the total E_{serv} of the *non-PA* case. For increasing power-saving delay D, servers are powered on and off less frequently. This implies idle servers are kept on for a longer time, resulting in a higher average number of servers on: Fig. 6 shows that slightly more servers are on than strictly necessary for the maximal job loss of $L = 0.05$ (as given by the *ErlB* bound). Note however that in the power-saving cases (as well as *non-PA*), the job loss lies far below *ErlB*'s $L = 0.05$: we found that for $f \leq 0.8$, job loss drops well below 10^{-6}.

4.3 Network load

Since in a power-saving approach (*PA*) a job may have to be sent to a remote site (rather than a nearby one powered off), we expect that the network load will increase compared to *non-PA*. This can be observed in Fig. 7. By increasing the power-saving delay D, servers are turned of less frequently, thus reducing the amount of off-loading to remote sites (i.e. lower job hop counts). Also, when using multi-core servers, chances of finding all C cores idle—allowing to power a server down—decrease, and therefore also the off-loading to remote sites.

This extra *PA* network load could incur an energy penalty stemming from the interconnecting network. (The client and server's network interface cards will need to send and receive the job's data anyway: BW_{serv} of Eq. 2 will not change.) However, this network will not be dedicated to Grid traffic, and therefore we consider it unlikely that when using shorter paths for the Grid jobs some links could be powered down. Hence, sending Grid jobs an extra hop further is deemed unlikely to have a noticeable impact on total network power consumption (e.g. measurements on a Force10 gigabit-ethernet switch with 656 Gb/s throughput indicate a power cost of only about 1 W/(Gb/s), versus an idle power—i.e. for load $BW(t) = 0$—in the order of 3.8kW or $3.8 \cdot 10^3/656 \approx 6$ W per potential Gb/s).

5 Conclusions

Measurement data indicates that Grid sites often experience periods of under-utilization: the server capacity foreseen to cope with a certain peak load is at other times not fully used. Thus a power-saving strategy turning off idle servers can reduce energy consumption. We proposed such a power-aware (*PA*) Grid scheme, and assessed its performance through simulations. Our results indicate that in lower-load scenarios, the power consumption can be reduced to a level

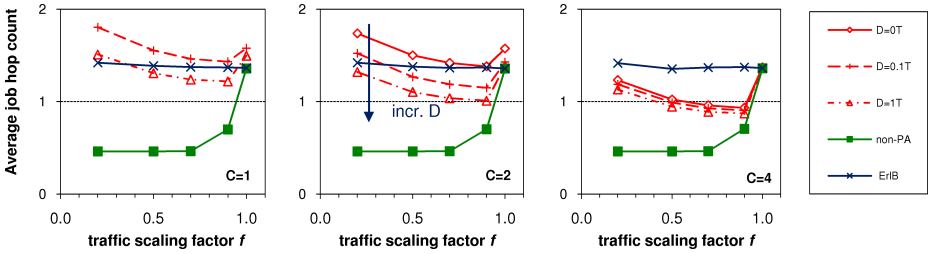

Fig. 7. A power-aware approach *PA* increases the hop count jobs have to travel compared to a *non-PA* scheme. This extra network load is limited for higher power-saving delays D or multi-core servers ($C > 1$).

close to that of a system dimensioned for that low load. The price paid is a slight increase in network utilization (compared to a *non-PA* scheme). Yet, it is limited and not expected to outweigh the server energy savings.

Acknowledgments This work has been supported by the European Commission through the Phosphorus and BONE projects, and by the Flemish Government through the Research Foundation (FWO). C. Develder is a post-doctoral fellow of the FWO, and thanks prof. B. Mukherjee and his group for the valuable discussions during a stay at UCDavis, CA with a travel grant from the FWO.

References

1. K. Christodoulopoulos, E. Varvarigos, C. Develder, M. De Leenheer, and B. Dhoedt, "Job demand models for optical grid research," in *Proc. 11th Int. IFIP TC6 Conf. on Optical Netw. Design and Modeling (ONDM2007)*, May 2007.
2. S. De Maesschalck *et al.*, "Pan-european optical transport networks: An availability-based comparison," *Photonic Network Commun.*, vol. 5, no. 3, pp. 203–225, 2003.
3. C. Develder *et al.*, "On dimensioning optical grids and the impact of scheduling," May 2008, submitted to *Photonic Network Commun.*
4. J. Koomey, "Estimating total power consumption by servers in the U.S. and the world," Analytics Press, Oakland, CA, 15 Feb. 2007. [Online]. Available: http://enterprise.amd.com/Downloads/svrpwrusecompletefinal.pdf
5. A.-C. Orgerie, L. Lefèvre, and J.-P. Gelas, "How an experimental grid is used: The Grid5000 case and its impact on energy usage," in *Proc. 8th IEEE Int. Symp. on Cluster Computing and the Grid (CCGrid2008)*, 19–22 May 2008.
6. M. Pickavet *et al.*, "Energy footprint of ICT," in *Proc. Broadband Europe*, 3–6 Dec. 2007.
7. W. Vereecken *et al.*, "Energy efficiency in telecommunication networks," in *Proc. 13th European Conf. on Networks & Optical Commun. (NOC2008)*, Jul. 2008.
8. E. Weidner *et al.*, "Environmental comparison of the relevance of PC and thin client desktop equipment for the climate, 2008," 24 Apr. 2008. [Online]. Available: http://it.umsicht.fraunhofer.de/TCecology/docs/TCecology2008_en.pdf