Optimization of Energy and Emissions in High-Performance Grid Computing Data Centres

Mikko Majanen, Olli Mämmelä
Seamless Networking Team
VTT Technical Research Centre of Finland
Kaitoväylä 1, Oulu, Finland
Email: mikko.majanen@vtt.fi, olli.mammela@vtt.fi

***Abstract*—At an early stage of information and communications technology and high-performance computing, performance and reliability were two important factors in research and development. It was highly essential that the hardware was able to function adequately, performing strategic operations. Energy consumption was not considered as a serious topic, since the technical characteristics of hardware and software were limited and the amount of computing nodes in a computing cluster, i.e., a data centre was small. Gradually the situation has evolved a lot: nowadays there are multiple data centres located in geographically diverse locations and the software has become more complex. Modern data centres are equipped with a large amount of computing nodes having vast computing power. However, all this progress has not come without a price, since more computing power equals more total power consumption. Consequently, energy consumption has become a major topic nowadays. This work presents two algorithms for optimizing energy and emissions in high-performance grid computing, in which multiple data centres are interconnected to each other. The algorithms are validated in a simulation environment by comparing them to standard round-robin algorithm. Our simulation experiments show that the solution is able to reduce energy consumption and emissions drastically without increase in job turnaround or wait time.**

***Keywords*-HPC; grid computing; energy; emissions.**

I. Introduction

Energy consumption is an increasingly important consideration in computing. Data centres consume substantial amounts of energy, at an increasing financial and environmental cost. In 2006, U.S. servers and data centres consumed around 61 billion kilowatt hours (kWh) at a cost of about 4.5 billion U.S. Dollars [1]. This is equal to about 1.5% of the total U.S. electricity consumption or the output of about 15 typical power plants. High energy consumption naturally causes huge environment pollution. It has been estimated that ICT, as a whole, covers 2% of world's CO_2 emissions [2].

In High-performance Computing (HPC), the ever-growing demand for higher performance seems to increase the total power consumption, even though more flops per watt are achieved. In order to provide even greater computing capabilities, HPC data centres can be interconnected to each other to form larger, federated or HPC grid data centres. The connection is implemented by using special grid software (e.g., UNICORE [3]) that manages the job submissions to all data centres belonging to the grid.

The energy consumption between the data centres may vary radically due to the different characteristics of the centres. For example, the server hardware in each centre may be different and consume different amount of energy. The centres may also locate geographically far from each other and the surrounding climate can cause large differences in the needed cooling, i.e., the Power Usage Effectiveness (PUE) [4] values between different centres may vary due to the surrounding climate. Also, since the energy sources can differ between the centres, the CO_2 emissions of the data centres may vary radically depending on the available energy sources. The differences between the data centres naturally enable optimizations regarding energy consumption and CO_2 emissions. In this paper we introduce two algorithms for selecting the data centre inside the grid in energy- and CO_2-aware manner. The performance of the algorithms is studied by simulations and the results show significant savings in energy consumption and CO_2 emissions.

The rest of the paper is organized as follows: Section II describes the related work. Section III introduces the cluster selection algorithms. Sections IV and V present the simulation model and scenario, respectively. Simulation results are presented in Section VI. Conclusion and future work are presented in Section VII.

II. Related work

As described in [5], several methods for saving energy in single HPC data centres have been studied. The methods include mainly the use of energy-efficient or energy proportional hardware, Dynamic Voltage and Frequency Scaling (DVFS) techniques, shutting down idle hardware components at low system utilizations, power capping, and thermal management. In our prior work [5], we used an energy-aware job scheduler to schedule

 ISBN: 978-1-61208-189-2

the jobs inside single data centres and shut down idle computing nodes whenever possible. We also noted that merely the choice of a different scheduling algorithm can affect the energy consumption of a data centre. In this paper we extend our scope from single HPC data centres to HPC grid data centres, and introduce two algorithms for selecting the data centre inside the grid in energy- and CO_2-aware manner.

To the best of our knowledge, there has not been much previous research that addresses the energy efficiency or CO_2 emissions of the grids from the whole grid perspective; mainly only optimizations inside a single data centre have been studied. Perhaps the most similar approach to our approach is Heterogeneity Aware Meta-scheduling Algorithm (HAMA) [6]. HAMA first selects the most energy-efficient cluster for the job based on the power consumption of the servers and the efficiency of the cooling system. Additionally, when running the job, DVFS is used to reduce the power consumption of the CPU. The simulation results show that HAMA can reduce up to 23% energy consumption in the worst case and up to 50% in the best case as compared to other algorithms (EDF-FQ, which prioritizes jobs based on a deadline and submits jobs to resource sites in earliest start time (FQ) manner with the smallest waiting time). Without DVFS, HAMA can still result in power savings of up to 21%.

Lynar *et al.* [7] have explored the effect on energy consumption by using different resource allocation mechanisms, both in a cluster and in a grid. The results show that different resource allocation methods can result in a significantly different energy usage while computing a stream of tasks. The Pre-processed Batch Auction (PPBA) and batch auctions almost always result in significantly lower energy use than a random resource allocation. By using a simple batch auction allocation method, energy consumption can be reduced up to 37.5%, and possibly even more by using the PPBA method.

Patel *et al.* [8] have presented an energy-aware policy for distributing computational workload in the Grid resource management architecture. They introduce a data centre energy coefficient that is taken into account as a policy when making allocation decisions for compute workloads. This coefficient is determined by the thermal properties of each data centre's cooling infrastructure including regional and seasonal variations. The estimated energy savings in case of three data centres located in two different time zones were large enough to give sufficient reason for the economic viability of the approach.

Shah and Krishnan [9] also analyze the climatic conditions as a means to reducing cooling energy costs. They show that dynamic optimization of the thermal workloads based on local weather patterns can reduce the environmental burden by up to 30% in their case study. Additionally, the data centre operational costs can be potentially reduced by nearly 35%. Due to the variability of fuel mixes encountered in a global grid, they also found that the use of pure energy consumption as a metric for environmental sustainability — a common practice in the ICT literature — can be erroneous.

The GREEN-NET framework [10] consists of an ON/OFF model, which includes prediction heuristics and green advice for the users and takes the decision to switch on or off the nodes, and an adapted energy efficient Resource Management System (RMS) at the grid level.

III. Optimization in the HPC Grid

The optimization algorithm in the HPC grid focuses on optimizing the scheduling process in the UNICORE middleware [3]. The scheduling process is triggered by submitting a job from the UNICORE Commandline Client, or from the UNICORE Rich Client to the UNICORE Workflow Engine. The UNICORE Workflow Engine queries a UNICORE Service Orchestrator (USO), on which cluster the job should be submitted. As a default, the USO uses round-robin algorithm for choosing the cluster. After cluster decision, the job is submitted to the RMS of the chosen cluster. The RMS takes care of executing the job according to the used scheduling algorithm, e.g., FIFO or backfilling.

In this work, we focus on reducing the energy consumption and the CO_2 emissions. The CO_2/energy related optimizations should not affect the current Service Level Agreement (SLA) or QoS agreements, or alternatively, a new green SLA [11] could be used. In HPC, there are no clear SLAs between users and data centres, but a reasonable turnaround time can be seen as sort of a QoS agreement. A possible green SLA for HPC data centres could mean that the users allow certain delay for the execution of their job. As a bonus, they will get some extra computing time for free.

For decreasing CO_2 emissions and/or energy consumption in federated HPC data centres, the optimization algorithm will be used for performing the cluster selection in CO_2/energy-aware manner. In addition to cluster selection algorithms, also energy-aware single site scheduling algorithms will be used. As depicted in Figure 1, the USO in UNICORE receives job requests coming from the users. The jobs include the requirements for the needed resources (e.g., number of nodes/cores, RAM, etc.). If the user wants to use the green SLA, it is also included in the job requirements. The grid optimization algorithm is used to select the most suitable cluster for the job and the job is subsequently submitted to the RMS of the selected cluster. The RMS uses energy-aware job scheduling algorithms to schedule the job and power off idle servers. The energy-aware job scheduling algorithms

 ISBN: 978-1-61208-189-2

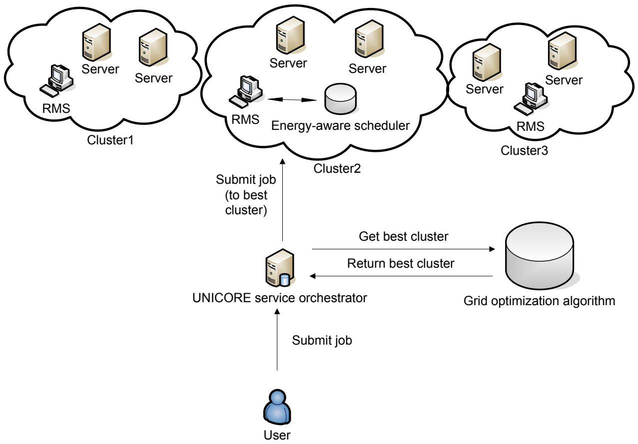

Figure 1. Job submission in a federated HPC data centre

for single site data centres were defined in our previous work [5].

A. Carbon Usage Effectiveness

Carbon Usage Effectiveness (CUE) is a sustainability metric developed by the Green Grid organization [12]. The main purpose of the metric is to address carbon emissions associated with data centres. The CUE can be calculated as follows:

$$CUE = \frac{SiteEmissions}{ICTEnergy}, \quad (1)$$

where $ICTEnergy$ is the energy consumpted by the ICT equipment in the data centre. An alternative approach for calculating the CUE is to multiply the Energy Source Coefficient (ESC) by the data centre's PUE:

$$CUE = ESC * PUE, \quad (2)$$

where PUE is a metric for defining how efficiently the power in the data centre is used, i.e., how much power is actually used by the ICT equipment and how much power is used for cooling and other equipment. ESC is defined as follows:

$$ESC = \sum ESP * EEC, \quad (3)$$

where Energy Source Percent (ESP) indicates the percentage of the energy generation source, and Energy Emission Coefficient (EEC) indicates how many kilograms of CO_2 are emitted per 1 kWh of energy. Example values of the EEC can be found in Table I [13]. By using the formulas described earlier and the values in Table I, we are able to estimate how much emissions are caused by data centres with different energy sources:

$$\begin{aligned} SiteEmissions &= CUE * ICTEnergy \quad (4)\\ &= PUE * ESC * ICTEnergy. \quad (5)\end{aligned}$$

Table I
ENERGY EMISSION COEFFICIENT FACTORS

Generation type	Conversion factor ($kgCO_2$ per kWh)
Closed cycle gas turbine	0.360
Coal	0.910
Electricity, France interconnector	0.083
Electricity, Ireland interconnector	0.699
Non pumped storage hydro	0.0
Nuclear	0.0161
Open cycle gas turbine	0.479
Oil	0.610
Pump storage	0.0
Other	0.610

B. Algorithm/policy description

This subsection describes the functionalities of the default round-robin cluster selection algorithm, as well as the two developed algorithms for optimizations: Fastest possible (FB) that tries to minimize the waiting time, and CO_2-aware (CUE) that tries to minimize the CO_2 emissions.

1) Round-robin: Round-robin (RR) algorithm is generally used in USO for selecting the cluster. Round-robin algorithm balances the number of jobs between different clusters by always choosing the next cluster compared to the previous selection. After the last cluster, the selection is started again from the first cluster.

2) Fastest possible: Fastest possible (FB) cluster selection algorithm tries to select the cluster that could possibly execute the job with minimal waiting time. For this, the algorithm first checks if there are enough idle nodes/cores in some cluster for executing the job. If yes and the cluster's queue is also empty, the job is submitted to that cluster. If not, an estimated waiting time for the job in each cluster is calculated by using the current status of each cluster: number of nodes and cores, status of running jobs, number of jobs in the queue, and walltimes of each queued job. The cluster with the shortest estimated wait time is then selected.

The algorithm relies on the dynamic cluster properties (status of nodes and queues), which can be obtained by a single site monitoring system. Otherwise, this dynamic information is not available for the USO, so the normal cluster selection algorithms can exploit only static cluster information for the decision making.

It should be noted that the wait time can only be estimated. The walltimes of the jobs are given by the users and, in general, they are inaccurate [14], [15]. Also, the used scheduling algorithm affects in which order the jobs are executed (especially backfilling). Thus, it is possible to calculate only the maximum wait times for the jobs, not the exact wait times.

3) CO_2-aware: This algorithm tries to find the cluster with the smallest amount of estimated CO_2 emissions.

 ISBN: 978-1-61208-189-2

The CO_2 emissions of the job are CUE * ICTEnergyOfTheJob. The simplest way is to select the cluster with the smallest CUE value. This works if the clusters have significant differences in their CUE values (CUE = ESC * PUE). If there are only small differences in the CUE values, then additional estimations should be done, since the job may consume different amount of ICT energy in different clusters due to the different computing node properties (CPU, RAM, etc.), and this difference may become a greater factor than CUE for the CO_2 emissions. The ICT energy of the job can be estimated by using the job requirements (number of nodes/cores, walltime) and cluster's computing node properties (CPU, RAM, etc.) as inputs for power consumption models such as those described in [5] and [16].

However, selecting always the cluster with the least amount of estimated CO_2 emissions would cause huge load and queue on the cluster with the least CO_2 emissions. This would mean large delay for the users. Thus, some form of load balancing is needed for this algorithm. In the conducted simulations (described in the next sections), we used a queue size limit: If the queue exceeded its size limit, the job was submitted to the cluster with the second least CO_2 emissions, and so on. In the case of green SLA, the users set a certain deadline for the completion of their job. This limit can be used for load balancing: The estimated completion time for the job can be calculated as a sum of the estimated wait time and walltime of the job. If this is in the limits, the cluster can be chosen. If not, the same calculations should be made to the cluster with the second least CO_2 emissions, and so on. If the user sets too strict a time limit for the job that none of the clusters can fulfill, the job should be either denied or the cluster should be chosen by the Fastest possible algorithm.

If CO_2 emission related information is not available for the cluster, a similar kind of algorithm can be used for selecting the cluster with minimal energy consumption by replacing CUE by PUE.

IV. HPC Grid Simulation model

The simulation model has been developed with the OMNeT++ discrete event network simulator [17] and the INET Framework [18]. The design of the model is similar as in [5], except that the model is extended from a single site scenario to a federated site scenario.

Figure 2 illustrates the network topology used in the simulations. It consists of three backbone routers, three gateway routers, three data centre modules, five clients and a USO module. In this scenario, the clients send HPC job requests to the USO, which is responsible for choosing an appropriate data centre, i.e., an HPC cluster, for executing the job. The USO has been adapted for the simulation so that it is capable of using the developed

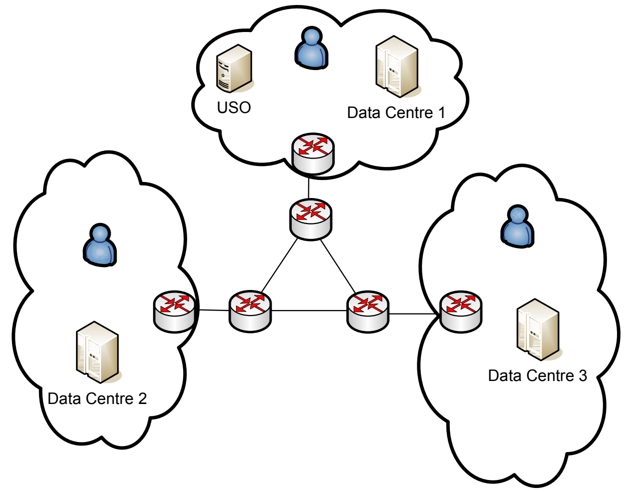

Figure 2. Network topology

optimization algorithms and making decisions based on the dynamic properties of the cluster. Normally, only static information of the cluster is available for the USO.

For the decision making, the USO can query the status and properties of each cluster from the corresponding RMS. Once the cluster is chosen, the USO forwards the job request to the RMS of the chosen cluster. The RMS uses the policies and scheduling algorithms of the cluster to choose suitable servers for job execution. When the job execution finishes, the RMS informs the USO, which again forwards the information to the client that submitted the job for execution.

The data centre module can be seen in Figure 3, which is similar as in the single site scenario. It contains a RMS, a fixed number of servers and a router between them. The RMS handles all incoming job requests arriving to the data centre and allocates the jobs to the servers for execution according to the selected policies and algorithms. Thus, the RMS also functions as a scheduler in the simulation. The RMS supports 6 different scheduling algorithms: standard FIFO, Backfill First Fit and Backfill Best Fit algorithms and their energy-aware counterparts developed previously (see [5] for more information on these).

The RMS module includes parameters for the PUE and the CUE. By using these two values, the USO is able to select a cluster that is the most energy-efficient or produces the least amount of CO_2 emissions.

V. Simulation scenario

In this section, we describe the simulation scenario and parameters. For evaluation we consider a scenario that includes three data centres and 75 clients that are sending job requests to the USO. The simulation is stopped once 1500 jobs have been completed. During the simulation

 ISBN: 978-1-61208-189-2

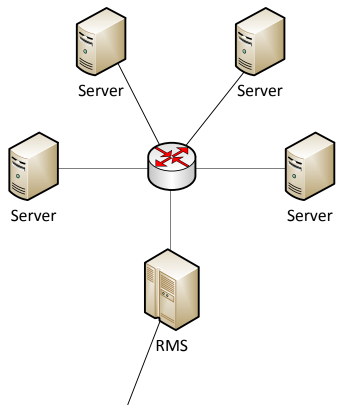

Figure 3. Data centre module

Table II
Simulation parameters

Parameter	Value
Simulation runs	10
Number of jobs	1500
Number of data centres	3
Number of clients	75
Number of gateway routers	3
Number of backbone routers	3
USO cluster selection algorithm	RR, FB, CO_2-aware
RMS scheduling algorithm	FIFO, BFF, BBF
Server memory	4 * 2 GB = 8 GB
Server cores per CPU	2
Server CPUs	2
Server CPU idle power	15 W
Server core voltage	1.2 V
Client job cores	1, 2, 4
Client job load	Uniform(30,99)
Client job nodes	Uniform(1,20)
Client job memory	Uniform(100MB, 2GB)
Client job run time	Uniform(600s, 86400s)

we measure the energy consumed by each data centre and present the obtained results in the next section. General simulation parameters are presented in Table II. Uniform(a,b) means randomly selected value according to a uniform distribution between a and b.

In Table III, we can see the parameters for the three clusters in the considered federated HPC data centre. The clusters have different characteristics, such as, the number of servers, PUE, and ESC. The energy sources (O = Oil, C = Coal, H = Hydro, N = Nuclear) for the clusters were selected so that both extreme ends in terms of ESC were represented in the simulations, while the third one represents something in the middle of them. Also, servers have different operating systems (OS) and processor architectures.

VI. Results

In all of the following figures, the algorithms are shortened as follows:

Table III
Data centre parameters

Parameter	Cluster 1	Cluster 2	Cluster 3
Servers	30	40	50
Energy source	C 50% H 20% N 30%	C 80% O 20%	O 20% H 40% N 40%
PUE	1.5	1.8	1.3
ESC	0.45983	0.85	0.12844
CUE	0.689745	1.53	0.166792
OS	Linux	Windows	Linux
CPU arch.	AMD	Intel	Intel

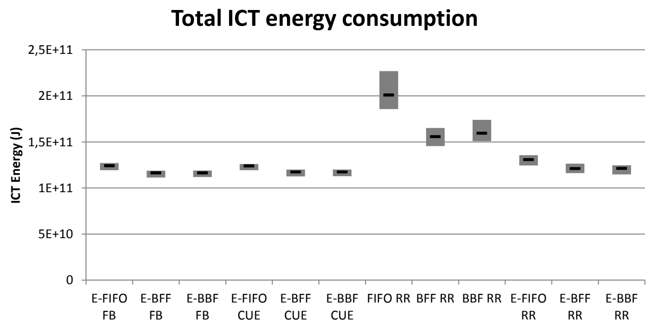

Figure 4. Total ICT energy consumption. Black lines represent the average value and the floating bars show the range of values from minimum to maximum

- FB = Fastest possible USO cluster selection algorithm
- CUE = CO_2-aware USO cluster selection algorithm
- RR = Round-robin USO cluster selection algorithm
- FIFO = First In, First Out job scheduling algorithm
- BFF = Backfilling first fit job scheduling algorithm
- BBF = Backfilling best fit job scheduling algorithm
- E-FIFO, E-BFF, E-BBF = energy-aware counterparts for the job scheduling algorithms (idle nodes are powered off whenever possible)

Figure 4 presents the total ICT energy consumption of the three clusters for different USO cluster selection and job scheduling algorithms. As can be seen, RR with normal job scheduling algorithms consumes the most amount of energy. RR with normal job scheduling algorithms represents a generally used, un-optimized algorithm combination in federated HPC data centres. Thus, it serves as a comparison point when calculating the energy savings and CO_2 emission reductions.

Figure 5 presents the energy savings achieved by using Fastest possible and CO_2-aware USO cluster selection algorithms instead of the default RR algorithm, and by using energy-aware job schedulers on each cluster. The energy-aware job schedulers are compared to their normal counterparts; for example, the first bar (E-FIFO FB) means the savings compared to FIFO RR. The last three bars present the savings when using RR but with energy-aware job scheduling. It can be seen that by using

 ISBN: 978-1-61208-189-2

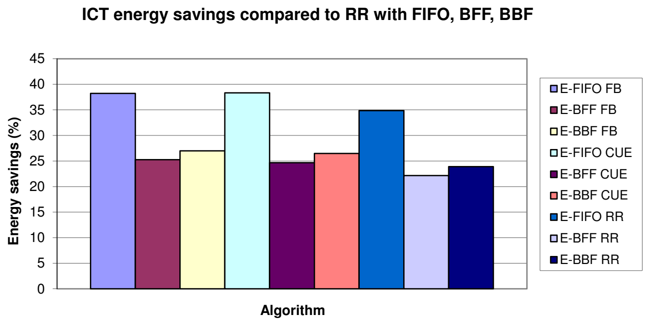

Figure 5. ICT energy savings compared to un-optimized, generally used RR with FIFO, BFF, and BBF

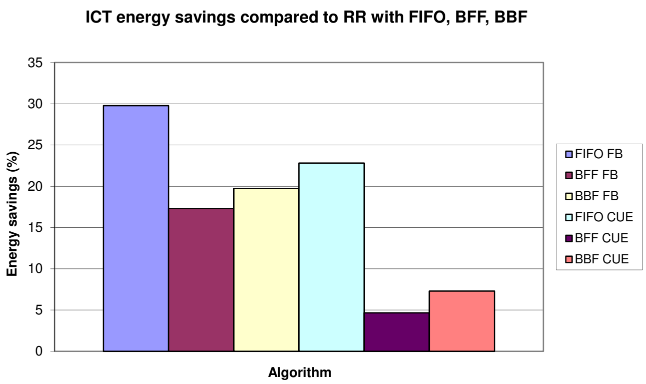

Figure 6. ICT energy savings compared to RR with normal job scheduling

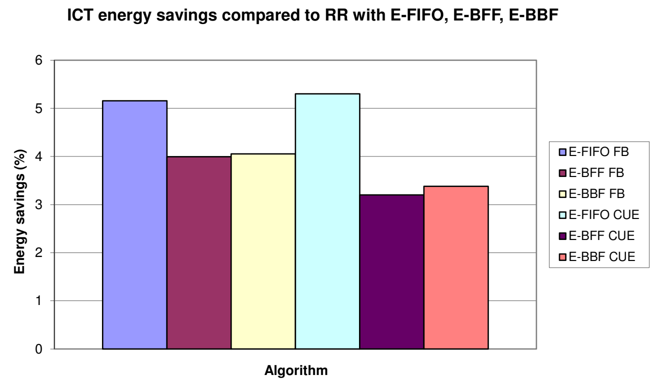

Figure 7. ICT energy savings compared to RR with energy-aware job scheduling

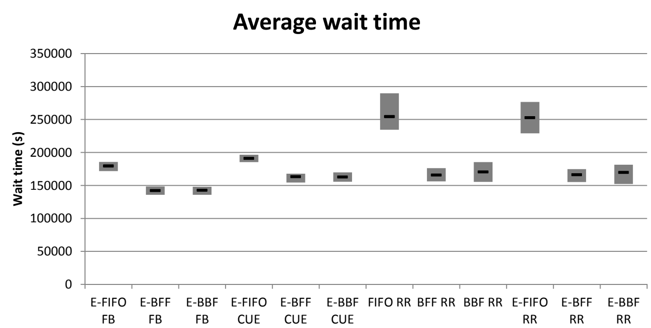

Figure 8. Average job wait times

energy-aware job scheduling, 22% to 35% energy savings can be achieved. Together with FB and CUE cluster selection, the savings are about 25% to 38%. However, if we only change the cluster selection algorithm, and keep the normal job scheduling algorithms, we can see from the Figure 6 that with FB we can save 17% to 30 %. Since the cluster selection is performed before job scheduling, we can say that about 8 % of the total savings are due to the energy-aware job scheduling, while the rest is due to the FB cluster selection. When comparing to RR with energy-aware job scheduling (as depicted in Figure 7), we can see that FB and CUE cluster selection algorithms can save additionally about 3% to 5%. For the explanation, we have to take a look at the jobs' average wait and turnaround times and the simulation duration.

Figure 8 presents the average wait times of the jobs in case of different USO cluster selection and job scheduling algorithms. As can be seen, the average wait time is clearly shorter with the FB USO algorithm. The CUE USO algorithm with backfilling has about the same average queuing time as RR, even though RR with FIFO clearly has the longest waiting time. Also, there are basically no differences between RR with energy-aware and normal job scheduling. This is true also in general, as reported in [5]: energy-aware job scheduling does not cause significant increase in wait time.

Figure 9 depicts the simulation duration, i.e., how long a time it took to execute all the 1500 submitted jobs. The graph shows the same as Figure 8: because the wait times are longer with RR USO cluster selection, also the simulation duration is longer.

Figure 10 presents the average job turnaround times in case of different scheduling algorithms. The story is the same as in previous figures: RR is slower due to the longer queuing time.

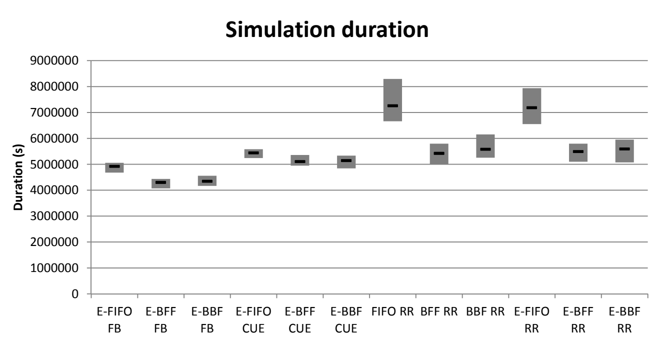

Figure 9. Average simulation duration

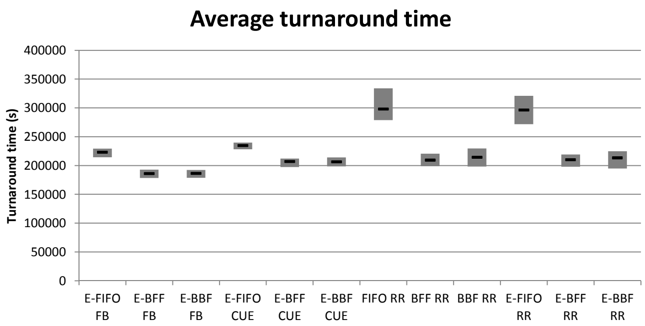

Figure 10. Average job turnaround time

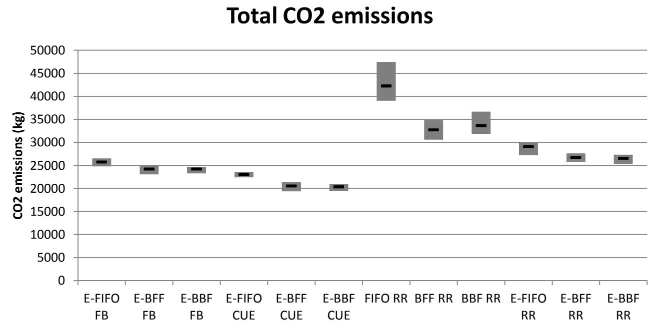

Figure 11. Total CO_2 emissions

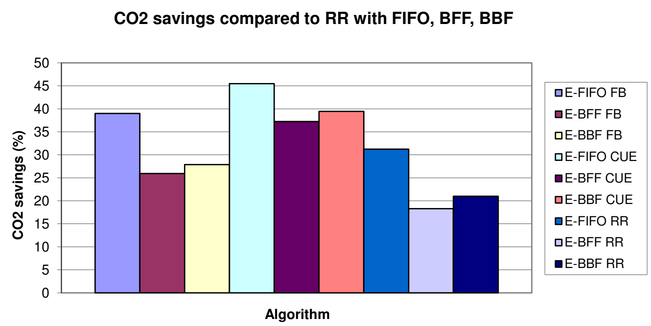

Figure 12. CO_2 savings compared to RR with FIFO, BFF, and BBF

Based on the results above, we can conclude that RR cluster selection with normal job scheduling algorithms can be very inefficient in terms of energy. This is because RR only balances the number of jobs among the clusters. It does not take into account the differences in the clusters (e.g., number of nodes/cores) or the differences in the submitted job characteristics (e.g., number of nodes/cores, walltime estimate). This can lead to a situation where one cluster is over utilized with many jobs waiting in the queue, while the other clusters can be under utilized at the same time, with nodes running idle. The energy-aware job schedulers (E-FIFO, E-BFF, E-BBF) power off the idle nodes whenever possible, and this is why a substantial amount of energy can be saved. On the other hand, the FB cluster selection algorithm inherently takes into account the differences in the clusters and submitted jobs: it always selects the cluster with the estimated minimal wait time, and thus balances the utilization between the clusters. Then fewer nodes are running idle and energy is saved.

Figure 11 presents the total CO_2 emissions of the federated HPC data centre. As can be seen, RR with normal job scheduling causes the largest CO_2 emissions. Using energy-aware job scheduling reduces the emissions due to the reduced energy consumption. Using FB cluster selection reduces the energy consumption still a bit more due to the better load balancing among clusters, and thus the CO_2 emissions are also smaller. CUE cluster selection algorithm favours the cluster with the best CUE value, i.e., least amount of CO_2 emissions, and hence achieves the greatest savings in CO_2 emissions, about 37% to 45% compared to RR with normal job scheduling. The CO_2 savings are depicted in Figure 12 as percentages.

VII. Conclusion and Future Work

The results show that the generally used round-robin cluster selection algorithm can lead to unbalanced utilizations among clusters. This can be very inefficient in terms of energy consumption and CO_2 emissions. Using energy-aware job scheduling to power off idle computing nodes whenever possible greatly enhances the energy-efficiency. Load can also be balanced by replacing round-robin cluster selection by the Fastest possible selection algorithm. This leads to energy savings due to the better utilization of clusters and shorter wait times. Using both energy-aware job scheduling and FB cluster selection simultaneously leads to greater energy savings than using only one of them. The greatest CO_2 emission savings can be achieved by using CUE cluster selection algorithm to favour the cluster with least CO_2 emissions. The actual savings in each case depends on the cluster and job characteristics. In these simulations, for example, the energy sources were chosen so that one cluster had rather small CUE, another one rather big CUE, while the third one was something between them. With smaller differences in CUE, also the possible savings in CO_2 emissions would be smaller.

Based on the simulation results presented above, we propose to use FB cluster selection algorithm for the jobs without green SLA, since it leads to energy and CO_2 emission savings due to the better utilization of the clusters, and to better QoS due to the shorter wait time. For the jobs with green SLA, we propose to use the CUE cluster selection algorithm, since it can lead to even greater CO_2 emission savings than FB, while keeping the QoS (in terms of time) at the user specified level. It can be used also without green SLA, if some other parameter (e.g., queue size limit) is used for load balancing to prevent excessive load on the "greenest" cluster.

Previous research in the energy-efficiency of HPC grid computing has mainly focused on performing optimizations inside a single data centre. This work presented a global view by taking into account the whole grid: the characteristics of the data centres, compute nodes and the computing hardware. The most comparable approach to our work is HAMA, described in [6]. The results of HAMA are similar to our approach: energy savings are between 23% and 50%.

Our future topics include building a high-performance computing test bed and testing our proposed solution in a laboratory environment. In the simulation studies presented in this paper, the PUE was assumed to be constant. However, PUE is not static in the long term. Instead, it changes over time as a function of outside temperature, for example. Future work would be to investigate the impact of dynamic PUE on the energy consumption and explore how the proposed solution is able to cope with it.

Acknowledgment

This work was supported by EU FP7 project FIT4Green [19]. The authors would like to thank all the colleagues working in the project. Special thanks to André Giesler from Jülich Supercomputing Centre for his comments.

References

[1] Y. Liu and H. Zhu, "A survey of the research on power management techniques for high-performance systems," *Softw. Pract. Exper.*, vol. 40, pp. 943–964, October 2010.

[2] Gartner. (2012, Jan.) Gartner Estimates ICT Industry Accounts for 2 Percent of Global CO_2 Emissions. [Online]. Available: http://www.gartner.com/it/page.jsp?id=503867

[3] D. Erwin and D. Snelling, "UNICORE: A Grid Computing Environment," in *Euro-Par 2001 Parallel Processing*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2001, vol. 2150, pp. 825–834.

[4] C. Belady, "Green Grid Data Center Power Efficiency Metrics: PUE and DCIE," Green Grid, Tech. Rep., 2008.

[5] O. Mämmelä, M. Majanen, R. Basmadjian, H. De Meer, A. Giesler, and W. Homberg, "Energy-aware job scheduler for high-performance computing," *Computer Science - Research and Development*, pp. 1–11, 2011. [Online]. Available: http://dx.doi.org/10.1007/s00450-011-0189-6

[6] S. K. Garg and R. Buya, "Exploiting Heterogeneity in Grid Computing for Energy-Efficient Resource Allocation," *Seventeenth Annual International Conference on Advanced Computing and Communications (ADCOM 2009)*, 2009.

[7] T. Lynar, R. Herbert, Simon, and W. Chivers, "Reducing grid energy consumption through choice of resource allocation method," *2010 International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW)*, May 2010.

[8] C. Patel, R. Sharma, C. Bash, and S. Graupner, "Energy aware grid: Global workload placement based on energy efficiency," Hewlett Packard, HP Laboratiories Palo Alto, Tech. Rep., November 2002.

[9] A. J. Shah and N. Krishnan, "Optimization of global data center thermal management workload for minimal environmental and economic burden," *IEEE Transactions on Components and Packaging Technologies*, vol. 31, no. 1, pp. 39–45, March 2011.

[10] G. Da Costa, J.-P. Gelas, Y. Georgiou, L. Lefevre, A.-C. Orgerie, J.-M. Pierson, O. Richard, and K. Sharma, "The GREEN-NET Framework: Energy Efficiency in Large Scale Distributed Systems," *HPPAC 2009 : High Performance Power Aware Computing Workshop in conjunction with IPDPS 2009*, May 2009.

[11] S. Klingert, T. Schulze, and C. Bunse, "GreenSLAs for the eco-efficient management of data centres," in *2nd International Conference on Energy-Efficient Computing and Networking 2011 (E-energy 2011)*, New York, NY, USA, 2011.

[12] C. Belady, D. Azevedo, M. Patterson, J. Pouchet, and R. Tipley, "Carbon Usage Effectiveness (CUE): A Green Grid Data Center Sustainability Metric," Green Grid, Tech. Rep., December 2010.

[13] RealtimeCarbon.org. (2012, Jan.) CO_2 conversion factors. [Online]. Available: http://www.realtimecarbon.org/resources/RealtimeCarbonMethodology.pdf

[14] W. Cirne and F. Berman, "A comprehensive model of the supercomputer workload," in *IEEE International Workshop on Workload Characterization, WWC-4. 2001*, dec. 2001, pp. 140–148.

[15] C. Bailey Lee, Y. Schwartzman, J. Hardy, and A. Snavely, "Are user runtime estimates inherently inaccurate?" in *Job Scheduling Strategies for Parallel Processing*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2005, vol. 3277, pp. 253–263.

[16] R. Basmadjian, N. Ali, F. Niedermeier, H. De Meer, and G. Giuliani, "A methodology to predict the power consumption of servers in data centres," in *Proc. of the ACM SIGCOMM 2nd Int'l Conf. on Energy-Efficient Computing and Networking (e-Energy 2011)*. ACM, 2011.

[17] OMNeT++. (2012, Jan.). [Online]. Available: http://www.omnetpp.org

[18] INET Framework. (2012, Jan.). [Online]. Available: http://inet.omnetpp.org/

[19] FIT4Green project. (2012, Jan.) FIT4Green: Energy aware ICT optimization policies. [Online]. Available: http://www.fit4green.eu