Towards a novel Smart and Energy-Aware Service-Oriented Manager for Extreme-Scale applications

Mohammed el Mehdi DIOURI, Olivier GLÜCK, Laurent LEFÈVRE
Laboratoire de l'Informatique du Parallélisme
UMR 5668 CNRS, ENS Lyon, INRIA, Université de Lyon
46, Allée d'Italie, 69364 Lyon Cedex 7, FRANCE
{*mehdi.diouri, olivier.gluck, laurent.lefevre*}*@ens-lyon.fr*

***Abstract*—Exascale supercomputers will gather hundreds of million cores. The main problem to take care for running applications on such platforms is energy consumption since it is one major limitation if we consider that the currently fastest supercomputer consumes more than 12MW for a maximum performance of 10PFlops. Besides, we also need to overcome important challenges related to fault tolerance and data management in such extreme-scale systems. Thus, we need to take into consideration these challenges from an energy consumption point of view and to propose them as energy-aware services for exascale applications. At this end, we propose in this paper to provide accurate estimations of the energy consumption due to these services and offer some green and energy efficient solutions, leading to a smart and energy-aware service-oriented manager for exascale applications.**

***Keywords*-Extreme-scale supercomputers; Energy efficiency; Smart grids; Services; Exascale applications.**

I. Introduction

A supercomputer is a machine built from a collection of computers performing tasks in parallel, in order to achieve very high performance. Since the early 90s, these supercomputers have grown rapidly, we can see in TOP500 list [1] that about every 11 years, the performance of supercomputers experiences a growth factor of 1000. 1 GFlop/s was reached in 1985 by the Cray 2, 1 TFlop/s was reached in 1997 by the ASCI Red system [2], 1 PFlop/s was reached in 2008 by Roadrunner [3]. According to the Top500 list published in November 2011, the most powerful supercomputer is the K-Computer, [4] a machine with more than 700,000 cores and able to perform 10 PFlop/s.

Supercomputers are now used to run a wide range of scientific applications, including manufacturing with the design of cars and aircraft, and environment with the prediction of tsunami damage and seismic waves. In order to meet new scientific challenges, designing exascale systems is identified by the high performance computing (HPC) community as a real need in roughly by the 2018 time frame. An exascale machine is a supercomputer capable of performing more than 10^{18} floating point operations per second (1 EFlop/s).

However, to ensure the transition to the exascale era, we must be able to address several challenges that will become even more problematic for exascale systems. Among these challenges, power/energy consumption is recognized as one of the most significant concerns to build exascale supercomputers.

This paper presents a novel smart and energy-aware service-oriented manager for extreme-scale applications. We mean by service an algorithm or a protocol that is performed to satisfy a need or to fulfill a demand. This framework considers the application features and the user requirements in terms of performance and level of service in order to:

- provide an accurate estimation of the energy that will be consumed by running different services used by the application;
- propose some green and energy efficient solutions for an optimized energy consumption.

This paper is organized as follows. Section 2 describes previous works. Section 3 presents future challenges at the Exascale. Section 4 presents energy aware services at Exascale. In Section 5, we present our novel smart and energy-aware service-oriented manager for exascale applications. Section 6 presents the conclusion and future work.

II. Related work

The issue of energy efficiency in distributed platforms has been seen mainly in the context of grids, datacenters, or cloud computing. Combined with hardware optimizations that offer the manufacturers, approaches to reduce the energy consumption of distributed platforms can be organized into two distinct classes, as follows.

The shutdown approach consists in dynamically turning off unused resources and turning them back only when they are needed. Many works like [1], [2] are based on this approach and suggest using on and off algorithms in order to avoid that machines consume energy while they are idle. However, to implement on and off algorithms, we need to migrate tasks from one node to another [3] or migrate virtual machines on other physical nodes [4], [5]. These

[1]Top500 list: http://www.top500.org/
[2]ASCI Red: http://www.sandia.gov/ASCI/Red/index.html
[3]Roadrunner: http://www.lanl.gov/roadrunner/
[4]K-Computer: http://www.top500.org/system/10587

978-1-4673-2154-9/12/$31.00

migrations generate an overhead in terms of energy and performance, which is essential to take into account while computing efficiency. In [3], Orgerie et al. propose to take into account in their approach the peaks of consumption experienced during node rebooting. At this end, they present a model that predict the future use of resources and define a minimum duration from which a resource consumes less energy if it is turned off during this time duration. If we predict that a resource will be idle for a period greater than this minimum threshold, then it has to be switched off during this time. Otherwise, it has to remain on. The shutdown approach gives satisfying results in terms of energy savings. However, its implementation requires the definition of a reliable enough prediction model and involves managing the loss of connectivity with the resources off.

The slowdown approach consists in dynamically adjusting the performance level of a resource according to the performance level the application and users really need. Again, many studies are based on this approach and propose to use DVFS techniques (Dynamic Voltage Frequency Scaling) for processors [6] or ALR techniques (Adaptive Link Rate) for the network interfaces [7]. In techniques based on DVFS, it is proposed to adapt as accurately as possible the clock speed of the processor depending on the performance required by the application. These techniques have inspired the definition of different energy states characterized by the CPU frequency, voltage and power consumption. Techniques based on ALR are similar to DVFS, as it consists also in adapting the performance of the network according to the importance of communications that take place on that network. These techniques are more and more taken into account in the implementation of processors today especially with the ACPI [5].

III. Future challenges at Exascale

In order to run exascale applications on extreme-scale systems, reliable fault tolerance mechanisms and optimized data management are mandatory. As we also need to manage the power and energy consumption in such systems, we consider that we need to take them into account in our energy-efficient framework.

A. Fault Tolerance: Checkpoint/Restart

One of the main challenges in designing exascale machines is fault tolerance. An exascale supercomputer will typically gather from half a million to several millions of CPU cores running up to a billion of threads. From the current knowledge and observations of existing large systems, it is anticipated that exascale systems will experience various kind of faults many times per day [8]. Thus, to reach exascale application termination, fault tolerance is mandatory.

Several techniques are used to implement fault tolerance in high-performance computing. We distinguish two main categories of protocols: uncoordinated and coordinated protocols. Both categories of protocols rely on checkpointing and in order to obtain a coherent global state, this checkpointing is associated with message logging in uncoordinated protocols [9] and with process coordination in coordinated protocols [10]. Hybrid protocols such as the one proposed by Ropars et al. in [11] propose to use coordinated protocol within a same cluster and message logging for messages exchanged between clusters.

In uncoordinated protocols, the crashed processes are reexecuted from their last checkpoint image to reach the state immediately preceding the crashing state in order to recover a coherent state with non crashed processes [12]. In coordinated protocols, all the processes must rollback to the previous coherent state, meaning to the last full completed coordinated checkpointing.

B. Data Management

Exascale infrastructures will also face important challenges related to data processing. Exascale applications will involve large volumes of data: hundreds of exabytes of data are expected by 2018. Consequently, we should be able to:

- scatter data over several or all processes;
- gather data from several or all processes;
- locate specific data among all processes;
- visualize data coming from all processes.

In order to optimize data movement cost in exascale applications, we should maximize data locality by allocating processes in an optimal way: the processes that communicate the most should be located on the closest cores. To increase performance in exascale applications, significant improvements in the data management algorithms are necessary.

C. Power/Energy consumption

Another challenge to consider in designing exascale supercomputers is the issue of high energy consumption caused by the incessant increase of performance. In [13], the issue of power consumption is considered as a potentially limiting factor to the future growth in high performance computing (HPC). These high costs including infrastructure installation and maintenance costs become predominant factors in the total cost of ownership (TCO) of a supercomputer.

Both the DOE and DARPA target a maximum of 20 MJ per second for a single exaflop [14], which corresponds to achieve an energy efficiency of 50 GFlop/J. However, when we take the most energy efficient petascale machine according to the list established by Green 500 [6], we find that its energy efficiency is lower than 1 GFlop/J.

Besides, by projecting today's technology, a study of exascale computing in the U.S.A came to the conclusion

[5]Advanced Configuration and Power Interface: http://www.acpi.info/DOWNLOADS/ACPIspec40a.pdf

[6]Green 500 list, november 2011: http://www.green500.org/lists/2011/11/top/list.php?from=1&to=100

that an exascale computer would require 120 MW of power [13], which is far from the 20MJ per second! This target of 20 MJ/s is less than twice the power consumption of today's most performant supercomputer: the K-computer, with 10 PFlop/s and 12MJ/s. Since 1 EFlop/s represents more that 100 times the performance of this supercomputer, then a lot of energy-saving engineering has to be developed over the next years in order to hit that 20MJ per second target.

IV. From Challenges to Energy-Aware Services

As we have seen previously, future challenges are unavoidable in exascale applications. In order to overcome these challenges, we should enable several services to run harmoniously together with extreme scale scientific applications. In order to run harmoniously, the power/energy consumption issue must always be regarded as a main concern whatever the service considered. In Figure 1, we present the services for which we should take into consideration energy consumption.

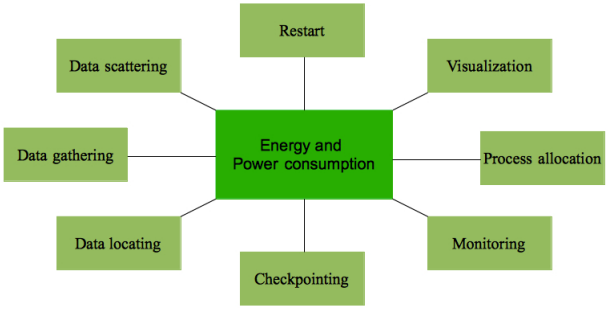

Figure 1. Services where energy consumption should be considered

As concerns fault tolerance, we propose two services. On the first hand, one is dedicated to the checkpointing step that is performed during the normal execution of the application. It consists in storing a snapshot image of the current application state. On the other hand, one is dedicated to the restart in case of failure. It consists in restarting the execution of the application from the last checkpoint. We also need a service for monitoring the resources that are involved in the extreme-scale system in order to visualize in real time the energy consumption, and to detect failures. As concerns data management, we need a service for scattering data, one for gathering data, one for retrieving a specific data and one for visualizing data. We also need a service for allocating processes in order to maximize the data locality.

A. Energy consumption Estimation

For each service presented before, several implementations are possible. For instance, as concerns fault tolerance, applications can be run either with coordinated or uncoordinated checkpointing. Depending on the application considered, the most efficient fault tolerance protocol in terms of energy consumption can be one or the other protocol. Hence, the first step to consume "less", is to take into account the application features and the user requirements in order to provide an accurate energy evaluation of the different implementations of each service. Indeed, this will enable the user to choose the less consuming protocol.

However, making an accurate estimation of the energy consumption due to a specific implementation of a service is really complex as it depends on several parameters:

- hardware:
  - architecture: number of nodes, number of sockets per node, number of cores per socket, network topology, memory architecture, etc.
  - features: network technologies (Infiniband, Gigabit Ethernet, proprietary solutions, etc), type of hard disk drives (SSD, SATA, SCSI, etc), etc.
- application specifications:
  - computation: ratio of time spent in computation, number of float operations per second, etc.
  - networking: the number of processes, number and size of messages exchanged between processes, ratio of time spent in networking, etc.
  - I/O: type of storage media used (RAM, HDD, NFS, etc), volume of data written/read by each process, etc.

For instance, if we consider data broadcasting over all the processing cores, the energy consumption of broadcasting depends on the number of processes, the broadcasting algorithm, the volume of data to broadcast, the storage media where data comes from, the storage media where data would be stored. It also depends on the architecture and the features of the supercomputer. Therefore, in order to compute accurate energy estimations, we should obtain all hardware and application information.

Performance and power consumption depend strongly on the hardware used in the extreme scale supercomputer. For example, broadcasting time depends on the exascale architecture (number of nodes, number of cores per node) and on the network interface used (Infiniband, Gigabit Ethernet, Proprietary, ...). In order to take into account the hardware specifications, we need to calibrate our estimations depending on the hardware used. At this end, we developed a set of benchmarks that extract the power consumption and the time execution of a given service. These measurements serve as a knowledge base that calibrate our estimations depending on the hardware used in the extreme scale supercomputer. Although this knowledge base has a significant size, it needs to be done only once. In order to get application features, we provide preliminary interaction with the user and the application in which we gather all the information we need as concerns the considered application and the execution context.

B. Energy-efficiency Management

Energy consumption of resources is rarely proportional to their usage. Even if processing nodes are completely idle, they consume significant power consumptions. A typical server in Lyon site of Grid5000 [7] consumes 175W when it is idle and 225W at its peak usage.

However, processes can often be idle or worse actively waiting for a synchronization with other processes. Therefore, energy is consumed inefficiently. To give a concrete illustration, in case of failure with the uncoordinated checkpoint, the crashed processes rollback to their last checkpoints while the non-crashed processes stay waiting until the restart of crashed processed is done.

To achieve important energy efficiency, we must rely on a very fine-grained cooperation between hardware resources and the application. To implement this resources/application cooperation, we propose to shutdown or slowdown resources during their idle and active waiting periods. The shutdown approach is promoted only if the idle or active waiting period is long enough, greater than the minimum threshold from which it becomes gainful to turn off a resource and turn it on again [3]. The shutdown and slowdown approaches are proposed at the component level, meaning that we consider to switch off or slowdown CPU/GPU cores, network interfaces, storage medium.

In order to know when to apply green levers (shutdown and slowdown approaches) for each service, we propose to provide predictions of idle and active waiting periods. These predictions are done based on the service features, such as the checkpointing interval as concerns the uncoordinated protocol provided in the fault tolerance service. Indeed, thanks to such information, we can anticipate that in case of failure, non-crashed processes will be actively waiting in average for half of the checkpointing interval.

In order to perform these green levers for each service, we suggest to ask the supercomputer administrator about its rights to act on available resources either if they will need to be turned off or slowed down during some periods of time. Besides, these energy efficient solutions should also be evaluated in terms of energy consumption.

V. Towards a Smart and Energy-Aware Service-Oriented Manager for Extreme-Scale applications: SEASOMES

A. SEASOMES components

As there are various services that are needed at Exascale, we believe that designing an unified energy-aware framework is essential for enabling a coordination of the several techniques used to improve their energy efficiency. In Figure 2, we present the main components of this framework. Some of them have been introduced in the previous section.

[7]Grid5000 platform is 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)

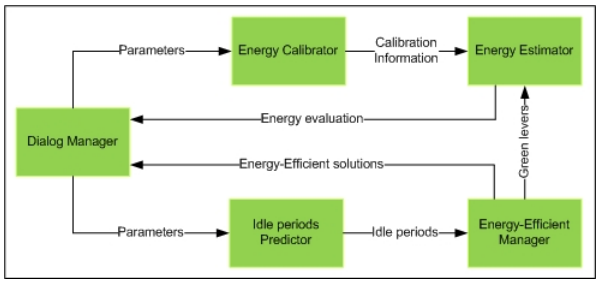

Figure 2. SEASOMES components

First of all, SEASOMES must be calibrated in order to take into account hardware specifications as concerns the energy consumption estimations. Then, each time that a user would like to run an application, its user provides some general information about the application and its personal requirements. This information is processed by SEASOMES in order to compute accurate energy consumption estimations of all the unoptimized implementations of the services wanted by the user, and to predict idle and active waiting periods known during the considered services. Depending on the idle and wanted periods discovered and on the rights assigned by the supercomputer administrator to the user, the energy-aware manager proposes the possible green levers that aim a maximization of the energy efficiency of the resources involved. These green solutions are evaluated by the energy estimator in order to inform the user about the significance of the energy-efficiency of the green solutions suggested. The dialog manager is the component processing all the information in and out of SEASOMES.

B. Solutions for "consuming better"

The energy efficient solutions previously proposed in this paper aim to "consume less" energy, meaning a reduction of the energy consumed. We must also look for solutions to "consume better", that is to say to consume at a lower cost the same amount of energy and/or to consume as much as possible a green energy. That's why we propose in the following subsection some new solutions to "consume better". To "consume better", we advise supercomputer users and administrators to run theirs applications whenever possible, during off-peak periods the applications requiring the highest energy consumption so that energy is cheaper and greener. For that, users can also provide a flexible execution planning so that administrators can adjust these executions during off-peak periods. Indeed, energy providers usually offer price per kWh depending on the time and day when energy is consumed. In [15], the authors proposed to schedule jobs in order to maximize the green energy consumption while respecting the jobs' deadlines. However, this work did not expose electricity pricing to the users.

For example, EDF [8], the French energy provider, shows

[8]EDF: http://bleuciel.edf.com

prices in euro per kWh depending on whether it is a "blue day", a "white day" or a "red day", but also depending on whether it is an off-peak hour (from 10pm to 6am) or a full hour (from 6am to 10pm). The price per kWh on "red days" and in full hours is more than 7 times the price per kWh on "blue days" and in off-peak hours, which is far from being negligible! Why not take the opportunity to run its HPC applications in off-peak periods?

In order to include this suggestion in our framework, SEASOMES gathers price information from the energy provider enabling to estimate the financial cost of the energy consumption depending on the period of time when the application would be executed. Thus, SEASOMES provides the financial cost for each slot in the agenda fixed by the user. Therefore, the user can choose the most convenient time slot of its agenda by considering together the termination date of the application, the energy consumption (in kJ) and the financial cost.

Beyond our suggested energy efficient solutions, we propose to implement a "smart grid" to better establish communication between energy suppliers and their most important customers, which can be both profitable for energy suppliers and for the most consuming clients, especially those who will run exascale applications.

Indeed, if we can estimate the future energy consumption of HPC computing applications and then we are able to send information on projected energy consumption to energy suppliers through permanent communication flows, then energy suppliers can more easily procure green energy and therefore offer it at reduced rates.

C. External interactions of SEASOMES

On Figure 3, we show the external interactions between SEASOMES, the user with its exascale application, the resource manager and the extreme-scale supercomputer, the supercomputer administrator and the energy provider.

Initially, SEASOMES launches a series of experiments via the resource manager to calibrate the energy estimator to take into account the architecture and physical characteristics of the supercomputer. Then, it gathers price information from the energy supplier in order to enable itself to provide financial estimation costs depending on the period of time.

Before launching an exascale application, the user specifies on the one hand the suitable agenda for the execution of its application. On the other hand, he informs about its requirements in terms of services needed, performance and of financial costs by providing the maximum execution time and the price that he doesn't want to exceed at every application launch.

Besides, SEASOMES gathers the application features either through the user or via a trace tool that automatically provides the needed information. Moreover, it consults the administrator to learn about the user rights as regards the use of green levers. Then, SEASOMES promotes as much as possible off-peak periods and ask the resource manager to involve the minimum possible resources to achieve the performance required by the user.

By taking into account this information and relying on the energy calibration, SEASOMES estimates for the best time slots in the agenda of the user, the energy consumption and the financial cost of the various services required by the user. By taking into consideration the user rights and whenever it is possible, it also proposes energy-efficient solutions that suggest to apply green levers on resources unnecessarily overused while services are running. If however the application reaches too high peak consumption, the energy supplier is notified in order to enable him to provide on time enough power and a green energy whenever possible.

In sum, it is a double negotiation on the one hand between SEASOMES and the user and on the other hand between SEASOMES, and the energy supplier. Concerning the user/SEASOMES negotiation, it is about finding a compromise that satisfies both:

- SEASOMES by providing energy estimations and energy efficient solutions that vary depending on the period of utilization, on the importance of involved resources, on the services required and the application features.
- the user who wants to run its application with the best performance at lower financial cost;

Regarding the energy supplier/SEASOMES relationship, it is about informing the energy supplier about peak demands so that he can regulate its procurement, which justifies the fact that the SEASOMES negotiates preferential rates and possibly allows the energy supplier to deliver a greener energy.

VI. Conclusion

In this paper, we addressed the issue of energy efficiency for exascale supercomputers. At this end, we proposed a smart and energy-aware service-oriented manager for exascale applications: SEASOMES. This framework aggregates the various energy-efficient solutions to "consume less" energy and to "consume better". It involves both internal and external interactions with the various actors interfering directly or indirectly with the supercomputer. On the one hand, we recommended a more fine-grained collaboration between application and hardware resources in order to reduce energy consumption and provide sustainable exascale services. On the other hand, we suggested a cooperation between the user, the administrator, the resource manager and the energy supplier for the purpose of "consuming better". In our future work, we plan to lean on this green framework to enrich the knowledge base of SEASOMES in order to incorporate a multitude of services.

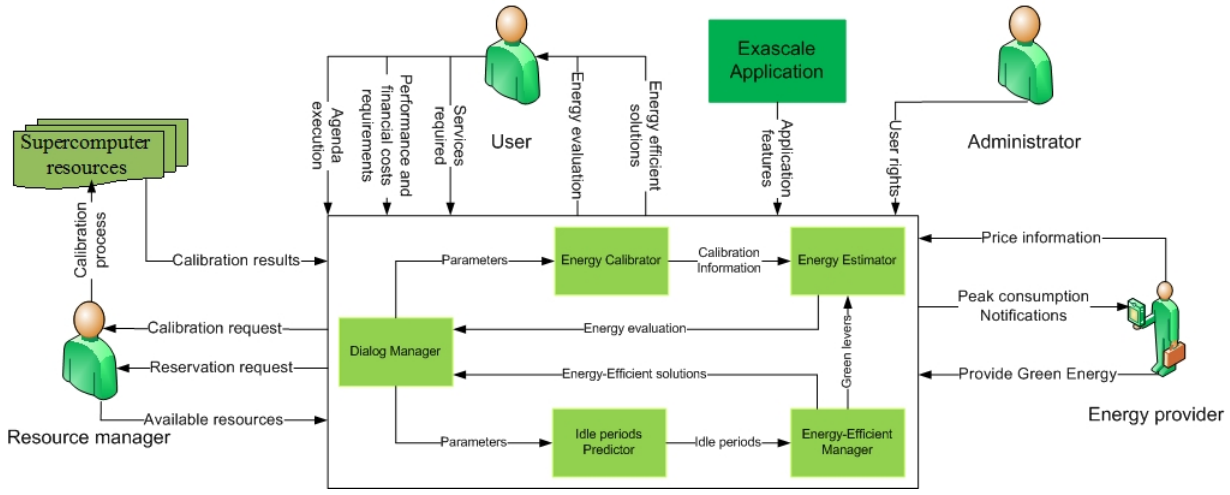

Figure 3. External interactions involving external actors

REFERENCES

[1] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, and R. P. Doyle, "Managing energy and server resources in hosting centers," in *Proceedings of the eighteenth ACM symposium on Operating systems principles*, ser. SOSP'01. Banff, Alberta, Canada: ACM, October 2001, pp. 103–116.

[2] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *Proceedings of the 34th annual international symposium on Computer architecture*, ser. ISCA '07, New York, NY, USA, 2007, pp. 13–23.

[3] A.-C. Orgerie, L. Lefevre, and J.-P. Gelas, "Save Watts in your Grid: Green Strategies for Energy-Aware Framework in Large Scale Distributed Systems," in *ICPADS 2008 : The 14th IEEE International Conference on Parallel and Distributed Systems, Melbourne, Australia*, December 2008.

[4] F. Hermenier, N. Loriant, and J.-M. Menaud, "Power Management in Grid Computing with Xen," in *Frontiers of High Performance Computing and Networking - ISPA 2006 International Workshops*, vol. 4331, Sorrento, Italy, December 4-7 2006, pp. 407–416.

[5] A.-C. Orgerie and L. Lefèvre, "When clouds become green: the green open cloud architecture," in *Parco2009 : International Conference on Parallel Computing*, Lyon, France, September 2009.

[6] Y. Hotta, M. Sato, H. Kimura, S. Matsuoka, T. Boku, and D. Takahashi, "Profile-based optimization of power performance by using dynamic voltage scaling on a pc cluster," in *Proceedings of the 20th International in Parallel and Distributed Processing Symposium, IPDPS 2006*, 2006.

[7] C. Gunaratne and K. J. Christensen, "Ethernet adaptive link rate: System design and performance evaluation," in *The 31st IEEE Conference on Local Computer Networks, Tampa, Florida, USA, 14-16 November 2006*. IEEE Computer Society, 2006.

[8] F. Cappello, A. Geist, B. Gropp, S. Kale, B. Kramer, and M. Snir, "Toward exascale resilience," *International Journal of High Performance Computing Applications*, vol. 23, pp. 374–388, November 2009.

[9] A. Guermouche, T. Ropars, E. Brunet, M. Snir, and F. Cappello, "Uncoordinated checkpointing without domino effect for send-deterministic mpi applications," in *25th IEEE International Symposium on Parallel and Distributed Processing, IPDPS 2011, Anchorage, Alaska, USA, 16-20 May, 2011*, pp. 989 –1000.

[10] K. M. Chandy and L. Lamport, "Distributed snapshots: Determining global states of distributed systems," *ACM Trans. Comput. Syst.*, vol. 3, no. 1, pp. 63–75, 1985.

[11] T. Ropars, A. Guermouche, B. Uçar, E. Meneses, L. V. Kalé, and F. Cappello, "On the use of cluster-based partial message logging to improve fault tolerance for mpi hpc applications," in *Euro-Par 2011 Parallel Processing - 17th International Conference, Bordeaux, France, August 29 - September 2, 2011, Proceedings, Part I*. Springer-Verlag, pp. 567–578.

[12] A. Bouteiller, T. Hérault, G. Krawezik, P. Lemarinier, and F. Cappello, "Mpich-v project: A multiprotocol automatic fault-tolerant mpi," *IJHPCA*, vol. 20, no. 3, pp. 319–333, 2006.

[13] H. D. Simon, "Is HPC Going Green ?" ScientificComputing.com, May/June 2008.

[14] P. M. Kogge and et al, "ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems," in *DARPA Information Processing Techniques Office*, Washington, DC, September 28 2008, p. pp. 278.

[15] I. Goiri, R. Beauchea, K. Le, T. D. Nguyen, M. E. Haque, J. Guitart, J. Torres, and R. Bianchini, "Greenslot: scheduling energy consumption in green datacenters," in *Conference on High Performance Computing Networking, Storage and Analysis, SC 2011, Seattle, WA, USA, November 12-18, 2011*. ACM, 2011, p. 20.