REVISING RESOURCE MANAGEMENT AND SCHEDULING SYSTEMS

Mehdi Sheikhalishahi and Lucio Grandinetti
Department of Electronics, Computer and System Sciences, University of Calabria, Rende (CS), Italy

Keywords: Green Computing, Resource Management, Performance, Computing Paradigms, Scheduling, Technologies, Resource Contention.

Abstract: With the explosive growth of Internet-enabled cloud computing and HPC centers of all types, IT's energy consumption and sustainability impact are expected to continue climbing well into the future. Green IT recognizes this problem and efforts are under way in both industry and academia to address it. In this paper, we take into account green and performance aspects of resource management. Components of resource management system are explored in detail to seek new developments by exploiting contemporary emerging technologies, computing paradigms, energy efficient operations, etc. to define, design and develop new metrics, techniques, mechanisms, models, policies, and algorithms. In addition, modeling relationships within and between various layers are considered to present some novel approaches.

1 INTRODUCTION

Climate Change and Global Warming are the two most important challenging problems for the Earth. These problems pertain to a general increase in world temperatures caused by increased amounts of carbon dioxide around the Earth. Researchers in various fields of science and technology in recent years started to carry out research in order to address these problems by developing environmentally friendly solutions. Green IT and in particular Green Computing are two new terms introduced in ICT community to address the aformentioned problems in ICT.

At the first sight of exploring green technologies and capabilities, we identify power management operations such as DVFS and IDLE-states capabilities at the processor level and virtualization technology at the middleware layer of computing systems as immediate solutions to address energy efficiency goals i.e. energy consumption minimization and heat-dissipation (limitations in wasted energy-removal).

On the other hand, resource management as the main middleware management software system plays a central role in addressing computing system problems and issues. Energy efficient operations, new technologies for green computing, and emerging computing paradigms should be exploited at the resource management to coordinate multiple elements within a system for manifold objectives. We believe green and performance objectives converge to the same point. In the evolution of resource management systems, we should take into account this note as we demonstrate this insight in this paper. In addition, resource management design and architecture should evolve according to the advances in contemporary technologies, computing paradigms, and energy efficient operations to provide new techniques, algorithms, etc. For instance, a comparison between cloud and other paradigms provides some guidelines and insights in the design and development of resource management components. Economic model and accuracy of allocations' parameters (requests') are the main two different characteristics of HPC and Cloud paradigms which highly impact scheduling.
In Cloud computing, pay-as-you-go on a utility computing basis is the economic model so users pay based on how much time they used cloud resources, while in HPC paradigm there is no general or specific economic model. Similarly, the requested allocation time and capacities of resources as allocation parameters are not accurate in HPC whereas they are precise and accurate in Cloud computing model. In fact, this is the result of economic model.

In HPC environment, a scheduler makes decisions and reservations according to the requested runtime of jobs which is an estimate of the real runtime; this means jobs might finish earlier or later than the specified requested runtime, however in Cloud the re-

Sheikhalishahi M. and Grandinetti L..
REVISING RESOURCE MANAGEMENT AND SCHEDULING SYSTEMS.
DOI: 10.5220/0003955401210126
In *Proceedings of the 2nd International Conference on Cloud Computing and Services Science* (CLOSER-2012), pages 121-126
ISBN: 978-989-8565-05-1

quested runtime is the actual runtime.

Therefore, in cloud model users provide which resources they need, the capacity of those resources and precisely for how much time. We might take into account these differences in these two paradigms to design and develop resource management components.

There are many approaches, mechanisms and algorithms in the literature on these problems and issues; however, most of them are special purpose. A complete approach should be a multi-level and general-purpose (holistic) approach that is architected over all layers of computing paradigms and systems.

For instance, in cloud paradigm from the highest level of resource management stack i.e. Cloud pricing strategies and Admission control policies to the lower levels i.e. policies (to direct a scheduler in making various decisions e.g. host selection for a specific job), and finally core scheduling algorithms are some of the research work which could be carried out in holistic scheduling approach. Such an approach should model the relationship between these layers, for example core scheduling information about jobs and resources might be considered in designing Cloud pricing strategies, Admission control policies and so on.

In brief, our paper lays some groundwork for future researchers in the field of resource management and scheduling to easily define, design and develop new objectives and it makes the following contributions in the field:

- Evolution of resource management and scheduling as new technologies, paradigms, etc. emerge
- Finding and establishing relationships within and between various layers of resource management
- Being general rather than special purpose solution for all computing paradigms i.e. Cluster, Grid and Cloud

The next parts of this paper are structured as follows. In Section 2, we describe computing system problems from the resource management point of view. Section 3 to Section 7 discuss and to some extent explore resource management components. Then, Section 8 points out to some key notes. Finally, Section 9 presents our conclusions and future work.

2 RESOURCE MANAGEMENT

In computing systems from resource management and scheduling point of view in general, there are the main problems and issues such as Low Utilization, Overload, Poor Performance, Resource Contention. In addition, if we consider Energy Efficient computing the High Energy Consumption is another issue. Moreover, sustainability and reliability are other issues to be addressed by resource management to some extent.

Figure 1 depicts the whole picture of our research including resource management components, contemporary technologies, computing paradigms, Green IT and their relationships. In this paper, in order to address these problems components of resource management system are explored in detail to seek new developments and evolutions and in parallel this process exploits contemporary emerging technologies, computing paradigms, energy efficient operations, etc. to define, design and develop new metrics, techniques, mechanisms, models, policies, and algorithms. Furthermore, finding and establishing relationships within and between various components is a key consideration in this development process. For example, a metric could be modeled as an approximate function of some other (well defined) metrics. A model could take advantages of several techniques, etc. A consolidation policy might exploit resource contention and utilization metrics in order to address Resource Contention and Utilization issues i.e. to achieve distribution and packing at the same time, respectively. Thus, in this case consolidation policy is modeled as a function of utilization and resource contention metrics.

We can enumerate many other relationships in resource management. However, in some relationships there are tradeoffs, we should model these tradeoffs as well. For example, if we improve utilization, we might face some performance issues such as Resource Contention and Overload. On the other hand, if we improve resource contention, utilization might degrade.

The solution to address the aforementioned problems and issues is all about to answer complex resource management questions which start by When, Which and Where with the help of well established relationships. For instance, Which types of applications might be consolidated together in a server?, When some workloads should be migrated to other servers?, Where a workload should be placed?, and lots of other general and specific questions.

2.1 Metrics

The first step in evolving resource management system is to define, model and develop new metrics to address some problems in better ways than already developed well known metrics. For example, in green computing Energy Consumption is a good metric to address energy related issues.

Utilization, Wait time, Slowdown, QoS, SLA

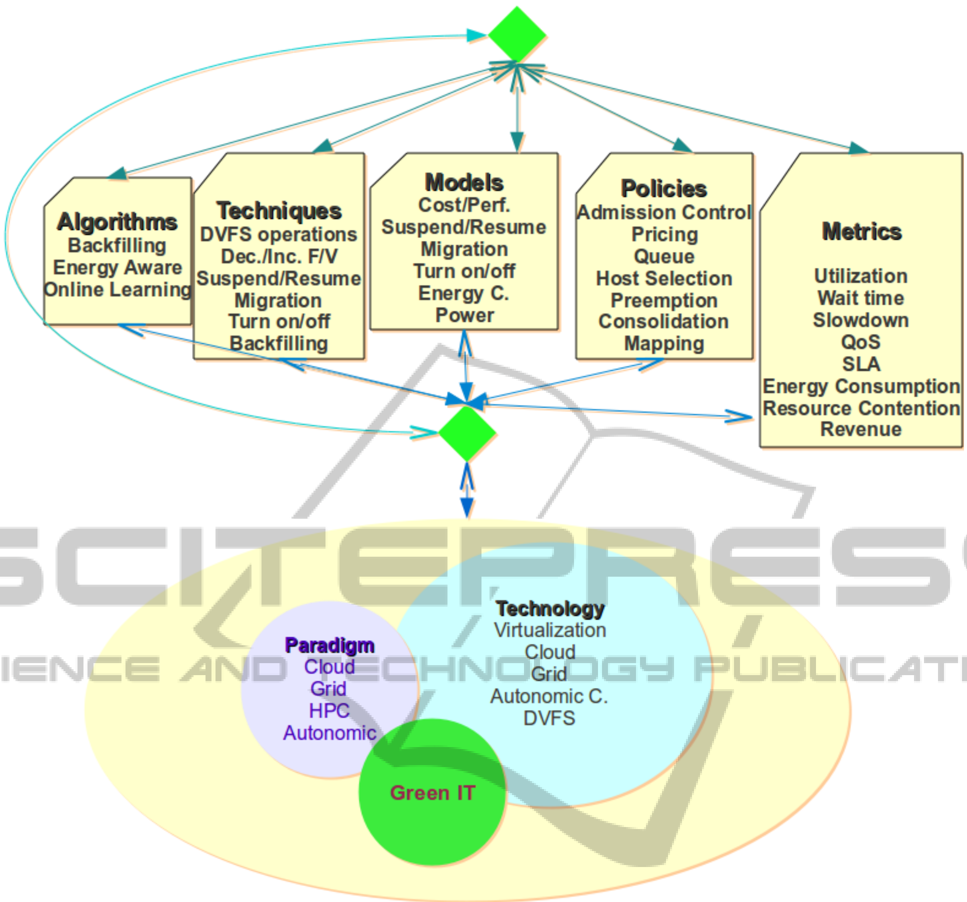

Figure 1: Resource management evolution.

are some traditional performance oriented metrics in HPC, Grid and Cloud paradigms. Resource Contention is another quasi-performance oriented metric (Sheikhalishahi et al., 2011b) and Revenue is an economic metric introduced by Cloud paradigm. In addition, more attention is paid to Energy Consumption metric since the appearance of the GreenIT term.

There are many metrics to be considered in resource management. We seek to establish some relationships between some of these metrics in order to have a better model, understanding of system behaviour, manifold objectives, etc. For instance, in (Srikantaiah et al., 2008) it is demonstrated that Utilization, Poor Performance, and Resource Contention as the main performance metrics and High Energy Consumption as the main energy efficient metric are directly inter-related to each other.

Resource Contention is widely recognized as having a major impact on the performance of computing systems, distributed algorithms, etc. Nonetheless, the performance metrics that are commonly used to evaluate scheduling algorithms do not take into account resource contention since researchers are interested in improving the conventional well known performance metrics such as utilization, slowdown and wait time.

On the other hand, in simple terms addressing Resource Contention issue will implicitly address Poor Performance, and High Energy Consumption issues as well as slowdown and wait time metrics; therefore in some environments Energy Consumption optimization and performance issues could be modeled as an approximate function of Resource Contention resolution. In fact, the following approximate function represents some portion of energy consumption in terms of Resource Contention and Poor Performance:

$$EnergyConsumption \simeq f(ResCont, PoorPerf) \quad (1)$$

and the following approximate equation represents poor performance:

$$PoorPerf \simeq g(ResCont) \quad (2)$$

so that, we simply model energy consumption as a relative approximate function of resource contention:

$$EnergyConsumption \simeq f(ResCont) \quad (3)$$

Therefore, we optimize resource contention metric (Sheikhalishahi et al., 2011b) to achieve energy optimization.

2.2 Techniques

Emerging technologies, paradigms, and energy aware actions provide various techniques to be exploited in resource management. Virtualization technology provides some heavyweight operations such as suspend/resume/migrate and start/stop on virtual machines, these operations are used in many recent research work to improve various metrics such as utilization.

For example, in (Sotomayor et al., 2008), authors demonstrated when using workloads that combine best-effort and advance reservation requests, a VM-based approach with suspend/resume can overcome the utilization problems typically associated with the use of advance reservations, even in the presence of the runtime overhead resulting from using VMs.

A DVFS-enabled processor provides some energy aware operations i.e. decrease/increase frequency/voltage, transitioning to an idle-state and transitioning to a performance-state. Time scaling is a technique as a result of DVFS operations which might be exploited in resource management to improve energy consumption and utilization metrics. For instance, by combining energy aware operations we can have some optimization in resource and energy usage that is to fill out free spaces in availability window of a scheduler by extension of or reduction of jobs' running times with the help of increasing/decreasing frequency of processors.

On the other hand, many devices provide the capability to transition to one or more lower-power modes when idle. If the transition latencies into and out of lower power modes are negligible, energy consumption can be lowered simply by exploiting these states. Transition latencies are rarely negligible and thus the use of low-power modes impedes performance. To minimize this impact, it is necessary to alter the workload so that many small idle periods can be aggregated into fewer large ones. This is a workload batching technique to be exploited in such cases.

In addition, techniques for dynamically balancing MPC (Memory accesses per cycle), IPC (Instructions per cycle), utilization and also dynamically scaling the frequency of processors with the help of online learning algorithms (Dhiman and Rosing, 2009) or other mechanisms are among the other techniques within this domain.

In depth study and research on scheduling strategies (Shmueli and Feitelson, 2005) in particular backfilling mechanisms, revealed that inaccurate estimates generally lead to better performance (for pure scheduling metrics) than accurate ones. This observation proposes the development of new scheduling techniques in HPC and Cloud paradigm according to the differences between these paradigms.

2.3 Models

A model quantifies some parameters in terms of some other parameters such as performance, energy, power and cost. For instance, in green computing a formal cost model quantifies the cost of a system state in terms of power and performance. Sleep states' power rate and their latency i.e. the time required to transition to and from the performance/power state, are examples of parameters in modeling cost vs. performance. In addition, models should specify how much energy will be saved in state transitions and how long it takes for state transitions.

Cost/Performance, Performance/Energy, Cost/Energy, Cost/Power, Suspension/Resumption, Migration, Turn on/off, Energy Consumption, and Power models are examples of some emerging models. Some models emerge as a result of some techniques such as Suspension/Resumption.

The models we seek to design and parameterize in green computing should relate to power consumption and computation rate (performance) or energy consumption and completion time simultaneously. These models are exploited by the scheduling algorithms to select the best state of a processor, memory, disk and network.

Power management actions may affect performance in complex ways because the overall computation rate is a net result of the speed and coordination of multiple elements within the system. For example, doubling the CPU speed may do little to increase the computation rate if the memory transactions do not move any faster. This indicates that models for the study of energy-computation tradeoffs would need to address more than just the CPU.

Models are also architecture and infrastructure dependent e.g. internal of Multicore and NUMA systems have different features and characteristics to be considered.

In addition, exploiting technology requires models. For instance, we can use the suspend/resume/migrate capability of virtual machines to support advance reservation of resources efficiently (Sotomayor et al., 2008), by using suspension/resumption as a preemption mechanism, In (Sotomayor et al., 2009) authors presented a model for predicting various runtime overheads involved in using virtual machines in order to support advance reservation requests. It adequately models the time and resources consumed by these operations to ensure that preemptions are completed before the start of a reservation.

2.4 Policies

We categorize policies into frontend and backend policies. Admission control and pricing are frontend policies whereas consolidation, host selection, mapping and preemption belong to backend policies.

Job requests pass through frontend policies before queueing. At the highest level in the cloud interface, we have pricing strategies such as Spot Pricing in Amazon (Amazon, 2010) and recent pricing approaches in Haizea (Sotomayor, 2010) or game theory mechanisms. These mechanisms apply cloud policies that are revenue maximization or improving utilization. Almost these policies have the same goals, and they are energy efficient since they keep cloud resources busy by offering various prices to attract more cloud consumers. A dynamic pricing strategy like offering cheaper prices for applications that will lead to less energy consumption (or higher performance) based on the current cloud status (workloads and resources) compared to the others is an energy efficient pricing schema. Pricing strategies implement cloud administrators' objective i.e. revenue maximization, utilization improvement, etc.

Backend policies could be categorized into three types: general-purpose policies (Dhiman et al., 2009), architecture-specific (or infrastructure-specific) policies (Hong et al., 1999) application-specific (or workload-specific) policies (Kim et al., 2007).

General-purpose policies are those that can be applied to most of the computing systems. For instance, CPU/cache-intensive workloads should run at high frequencies, since by increasing frequency the performance scales linearly for a CPU/cache-intensive workload. However, if a task is memory-intensive, the performance improvement is relatively insensitive to increase in frequency, so that a memory-bound workload favors by running at a lower frequency to reduce energy consumption.

Architecture-specific policies are defined based on the architecture or the infrastructure in which computation happens. Also application-specific policies are defined around applications' characteristic.

Technically, workload consolidation (Hermenier et al., 2009) policy is a sort of policy at the intersection of the last two mentioned policies. Bundling various types of workloads on top of a physical machine is called consolidation. Furthermore, consolidation-based policies should be designed in such a way to be an effective consolidation. In fact, effective consolidation (Srikantaiah et al., 2008) is not packing the maximum workload in the smallest number of servers, keeping each resource (CPU, disk, network and memory) on every server at 100% utilization, such an approach may increase the energy used per service unit.

Placement of jobs is a critical decision to address Resource Contention. Consolidation policies are one of the sources of information for effective placement of jobs in computing paradigms. In (Sheikhalishahi et al., 2011b), we developed effective energy aware consolidation policies. We designed consolidation policies according to the resource contention model (metric) and implemented them as host selection policies of a distributed system scheduler.

2.5 Algorithms

Algorithms implement techniques, models and policies. For instance, algorithms based on cost/performance models are part of the scheduling to model cost vs. performance of system states. In addition, core scheduling algorithms deal with implementing various backfilling mechanisms, etc. to improve utilization and other optimizations at core of a scheduler.

In (Sheikhalishahi et al., 2011a) we have proposed a novel autonomic energy efficient resource management and scheduling algorithm (architected on different levels of resource management stack). The proposed autonomic scheduling approach answers some When questions to improve energy consumption as it is modeled by resource contention metric. For that, it models interaction between queue mechanism and core scheduler information (about jobs and resources), as a result according to the system state, jobs are reordered and those jobs which satisfy necessary energy aware conditions get admitted to the wait queue and the others are delayed for the next scheduling cycles. From an autonomic scheduling point of view, there are some loops between queue mechanism, scheduling function, end of a job event and core scheduler information. This is an example of an multi-level algorithm over various resource management components.

3 CONCLUSIONS AND FUTURE WORKS

In this paper, we have taken into account green and performance aspects of resource management systems. First, we enumerated the main issues in computing systems i.e. Low Utilization, Overload, Poor Performance, Resource Contention, High Energy Consumption. We highlight that these issues should be addressed by the resource management of computing systems. Then, resource management components are explored in detail to seek new developments by exploiting contemporary emerging technologies, computing paradigms, energy efficient operations, etc. to define, design and develop new metrics, techniques, mechanisms, models, policies, and algorithms.

As an example, we reduced some problems to resource contention problem. In addition, we modeled energy consumption optimization and performance resolution as an approximate function of Resource Contention resolution.

Modeling relationships within and between various layers are considered to present some novel approaches such as in resource contention metric design and autonomic energy efficient scheduling algorithm.

In particular, this paper lays some groundwork for future researchers in the field of resource management and scheduling, as a result, a lot of future work in the framework of this paper could be conducted.

Of some notes, although power and energy are often used interchangeably, there are important distinctions. Transactional applications are better described in terms of throughput and average power, whereas completion time and energy consumed are more meaningful for nontransactional applications. Power consumption can often be increased over short periods to accelerate computation or accumulate more data and thereby minimize overall energy consumption. On the other hand, in resource management systems, we are interested in energy consumption metric not power metric or power average metric (as over time power draw is variable), since this study is over a time horizon not one second. Power metric is not revealing any information about consumption, since it is not consumption. In fact, Watt is the rate of energy consumption within one second. In all, energy calculations become much more practical with Joules. In other words, since in a cloud model we pay for what we consume, so that we pay for Joules.

REFERENCES

Amazon (2010). Amazon ec2 spot instances. http://aws.amazon.com/ec2/spot-instances/.

Dhiman, G., Marchetti, G., and Rosing, T. (2009). vgreen: A system for energy efficient computing in virtualized environments. In *the 14th IEEE/ACM International Symposium on Low Power Electronics and Design*. ISLPED '09.

Dhiman, G. and Rosing, T. (2009). System-level power management using online learning. *IEEE Transactions on CAD'09*.

Hermenier, F. et al. (2009). Entropy: a consolidation manager for clusters. In *VEE'09*.

Hong, I., Kirovski, D., Qu, G., Potkonjak, M., and Srivastava, M. B. (1999). Power optimization of variable-voltage core-based systems. *IEEE Trans. Computer-Aided Design*, 18(12):1702–1714.

Kim, K. H., Buyya, R., and Kim, J. (2007). Power aware scheduling of bag-of-tasks applications with deadline constraints on dvs-enabled clusters. In *CCGRID*, pages 541–548.

Sheikhalishahi, M., Grandinetti, L., and Laganà, D. (2011a). Autonomic energy efficient scheduling. preprint (2011), to Future Generation Computer Systems.

Sheikhalishahi, M., Llorente, I. M., and Grandinetti, L. (30 August - 2 September, 2011b). Energy aware consolidation policies. In *International Conference on Parallel Computing*.

Shmueli, E. and Feitelson, D. G. (2005). Backfilling with lookahead to optimize the packing of parallel jobs. *J. Parallel Distrib. Comput.*, 65:1090–1107.

Sotomayor, B. (2010). *Provisioning Computational Resources Using Virtual Machines and Leases*. PhD thesis, Department of Computer Science, University of Chicago.

Sotomayor, B., Keahey, K., and Foster, I. (2008). Combining batch execution and leasing using virtual machines. In *Proceedings of the 17th international symposium on High performance distributed computing*, HPDC '08.

Sotomayor, B., Montero, R. S., Llorente, I. M., and Foster, I. (2009). Resource leasing and the art of suspending virtual machines. In *Proceedings of the 2009 11th IEEE International Conference on High Performance Computing and Communications*.

Srikantaiah, S., Kansal, A., and Zhao, F. (2008). Energy aware consolidation for cloud computing. In *USENIX HotPower'08: Workshop on Power Aware Computing and Systems at OSDI*.