Evaluating the Need for Complexity in Energy-aware Management for Cloud Platforms

Pooja Ghumre, Junwei Li, Mukil Kesavan, Ada Gavrilovska, Karsten Schwan
Center for Experimental Research in Computer Systems (CERCS)
Georgia Institute of Technology, Atlanta
{poojaghumre, junwei.li}@gatech.edu, {ada, mukil, schwan}@cc.gatech.edu

Abstract

In order to curtail the continuous increase in power consumption of modern datacenters, researchers are responding with sophisticated energy-aware workload management methods. This increases the complexity and cost of the management operation, and may lead to increases in failure rates. The goal of this paper is to illustrate that there exists considerable diversity in the effectiveness of different, potentially 'smarter' workload management methods depending on the target metric or the characteristics of the workload being managed. We conduct experiments on a datacenter prototype platform, virtualized with the VMware vSphere software, and using representative cloud applications – a distributed key-value store and a map-reduce computation. We observe that, on our testbed, different workload placement decisions may be quite effective for some metrics, but may lead to only marginal impact on others. In particular, we are considering the impact on energy-sensitive metrics, such as power or temperature, as corresponding energy-aware management methods typically come with greater complexity due to fact that they must consider the complex energy consumption trends of various components in the cloud infrastructure. We show that for certain applications, such costs can be avoided, as different management policies and placement decisions have marginal impact on the target metric. The objective is to understand whether for certain classes of applications, and/or application configurations, it is necessary, or it is possible, to avoid the use of complex management methods.

1. INTRODUCTION

It is well known that projected increases in the power consumption of datacenters and servers in the US and worldwide continue to grow, with recent estimates [3] reporting staggering 20% increase in 2012 alone, to represent approximately 2% of the world's electricity consumption at an annual cost of $44.5B and placing an estimated peak load on the power grid of 31GW. In addition to this rising demand, Gartner has estimated that if the cost of power continues to rise, the cost of datacenter power may outstrip the cost of computing hardware. These trends are unsustainable, even when taking into account expected IT efficiency improvements [7].

In response to these trends, one solution with wide adoption is that of consolidating workloads on fewer virtualized physical resources. This model, further adopted by Cloud Computing, is based on the observation that typical resource utilization levels are low – in few 10s of percent, however hardware is not "power proportional", thereby utilizing significant power/energy levels in spite of the relatively low consumption of IT resources. In this context, researchers in industry and academia are actively developing sophisticated solutions to manage how consolidated cloud workloads use the underlying cloud platform, so as to ensure increased energy efficiency, while still satisfying application/client performance demands [11, 2, 13, 16, 18, 17, 15, 8, 5, 6, 1].

While these management stacks have been demonstrated to lead to improvements in the power consumption on the target platforms, and for the target workloads, they also lead to increased complexity. It is a well known fact that increases in software complexity lead to increases in deployment costs, runtime overheads, and more importantly, decreased reliability of the software due to increased possibility for software bugs and failures. At increasing scales, and at increased diversity of target platforms and workloads, it becomes even more important to assess the efficiency of the management methods and their cost/benefit ratio.

Toward this end, the goal of our research is to illustrate the differences which exist in the impact of management methods on different applications, application configurations or loads. We are particularly targeting popular classes of cloud applications – a distributed key-value store, represented via Voldemort [19], the key-value store used by LinkedIn and other companies, and a map reduce application, exemplified through one of the Hadoop MapReduce benchmarks [9]. We explore the effects of different management decisions (i.e., different VM placements) and illustrate their impact on a range of management metrics, from those concerned with purely CPU utilization, to more complex metrics taking into consideration performance and power utilization, energy usage and temperature. In this manner, our goal is to gain an understanding of the circumstances under which it is necessary/beneficial to incur increased costs related to energy-aware management.

In summary, our technical contributions include:

- illustration of the non-linear relationships among different types of IT (e.g., CPU and memory usage) and environ-

mental (power utilization and temperature) resources in datacenter systems,
- experimental evaluation of the impact of different workload placement decisions on a range of metrics, from CPU utilization to performance/Watt for representative cloud workloads, and
- demonstration of the opportunities to avoid management complexity for certain classes of applications or for certain management metrics.

2. TESTBED

We first describe the physical testbed used in our experiments and software infrastructure used for monitoring and analyses.

Physical Testbed. We use compute nodes from one rack of a 3000+ core fully instrumented prototype datacenter testbed. All nodes in the datacenter are currently virtualized with VMware's ESX hypervisor, and managed by management infrastructure based on VMware's vSphere stack. In the vSphere architecture, datacenter nodes are organized in subclusters, each under the control of a Virtual Center. On each node, platform-level sensors data is gathered at the hypervisor-level (i.e., ESX) and then propagated and logged at the Virtual Center responsible for that node.

All racks in the datacenter are outfitted with temperature, pressure, humidity sensors, anemometers, and remotely accessible managed power strips with outlet-level power monitoring and switching. In this testbed, external SNMP-accessible power sensors (PDUs) measure power utilization on per chassis' power domain. Each chassis or Blade Center (BC) consists of 14 blades, grouped in two power domains of 6 and 8 blades, respectively. Platform-level sensors on each node provide information regarding fan speeds, CPU temperature, as well as other information available through board-level management controllers such as utilization of CPU or memory resources.

Monitoring Infrastructure. We leverage our group's substantial investment in developing cloud monitoring and management software. In order to deal with the scale of information generated via the various sensors described above, we use an implementation based on existing scalable open-source components. For online data collection and aggregation, we leverage an open source distributed data transport service – Flume – deployed within our datacenter, and collecting data from the vSphere VirtualCenter monitoring modules and the physical PDUs. This permits continuous collection and analysis of power, load, performance and thermal properties of the datacenter systems, of the cooling infrastructure, and of the applications being run.

We augment the online monitoring and analysis with a database to store historic data necessary to better understand power consumption, the relationships between application performance, resource usage, and thermal effects, to aggregate the energy usage information across all workload components, and to improve the models used for runtime management. For this, we use the open-source HBase [10] database to scalably store the data gathered through the myriad of relevant sensors and monitoring interfaces.

On top of this data storage layer, we run map-reduce Hadoop jobs to aggregate the per-VM or per-set of VMs energy usage over time, to gain insights into certain datacenter or system behaviors, and to validate models that can be used at runtime to manage such systems [4, 5].

3. NEED FOR COMPLEXITY IN ENERGY-AWARE MANAGEMENT

Next, for a single rack in our datacenter, we perform analysis on the relationship between the utilization level of the nodes' compute resources (i.e., CPU and memory), and the resulting power consumption and temperature. For these measurements, we use the wileE benchmark developed by our group [14] which permits us to generate desired level of CPU or memory load in a given VM. We perform repeated experiments with workload configurations which generate uniform but different CPU loads across the rack, and generate "datacenter maps" for each of the measured resources. Figure 1 shows the maps for one representative run. Darker coloring of a block corresponds to higher usage levels and vice versa.

As can be observed from the figure, the rather uniform levels of CPU and memory load per blade, and the increasing load pattern across the rack, do not correspond to similar patterns in the power utilization or temperature distribution. Regarding power, we observe that one part of the rack continuously operates at higher power level, across all blades. This is due to the design of our datacenter infrastructure, where the nodes in each BC are divided unevenly among two power domains, with one power domain also powering fans and other BC components. Second, regarding CPU temperature per blade, we observe an even more divergent pattern, compared to the CPU and memory maps for the rack, with middle and bottom nodes typically reporting even substantially higher temperature readings. This behavior is attributed to a well-known fact that, as a result of the complex air-flow dynamics and rack construction, bottom and inner parts of the rack are hotter than others.

These observations lead to the conclusion that there are complex relationships between the measured usage of different resource types – CPU, power, temperature – in current datacenter platforms. As a result, management tasks aimed at optimizing metrics involving multiple parameters, such as balancing CPU usage and temperature distribution, must understand and consider this complexity. For instance, a management method aimed at maintaining performance *and* balanced power utilization, cannot simply look at the nodes' CPU, memory and other resources' usage, but instead must consider the design of the specific hardware platform where such management is conducted [5, 4]. Or, management methods concerned with balancing CPU usage and avoiding hotspot creation, must be aware of the infrastructure topology and physical properties of the environment.

The goal of the research presented in this paper is to understand the extent to which the consideration of such complexity is necessary, particularly when considering classes of emerging cloud workloads. The experimental results and analysis of our findings are discussed next.

4. IMPACT ON CLOUD WORKLOADS

Given the complex relationships between different resource types illustrated in the previous section, we next consider the behavior of popular cloud applications on such platforms. Specifically, we focus on two types of cloud workloads: a distributed key-value store using the Voldemort benchmark, and a CPU intensive map-reduce computation using the Hadoop open-source map-reduce framework.

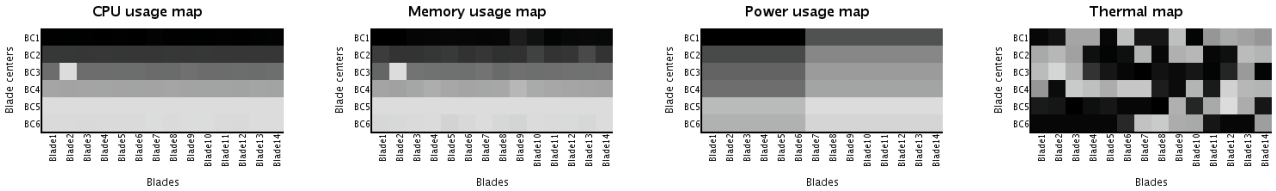


Figure 1: Datacenter maps for different resource usage.

For each workload we statically choose multiple configurations, i.e., multiple management decisions, and measure a range of performance metrics, such as (i) average throughput or response time, reported by the application, (ii) average CPU and memory utilization, obtained through the ESX hypervisor, and (iii) average power usage per power domain and average temperature per blade, gathered from the various sensors in the environment. In addition, we also compute more complex metrics to consider the imbalance in a particular resource, such as temperature or CPU usage, or to consider multiple types of resources, such as application performance per W, used for energy-aware management methods.

In the remainder of this section, we present the results from our experiments, by comparing the results from three representative runs for each application. In each case, one configuration is chosen as a default configuration, and the other two are obtained by migrating some number of VMs to other nodes in the system, for instance, as in response to some management policy. For these results, so as to have greater control over the experiments, we manually choose the target VMs and destination nodes, as opposed to deploying a particular management algorithm to drive such reconfigurations.

The experiments are conducted on a rack in the same instrumented datacenter used in Section 3. Few of the 84 nodes in the rack had permanent failures. For our experiments, we are using Blade Centers BC1, BC2 and BC4, running low, medium and high loads, respectively. In addition, Blade Centers BC3, BC5 and BC6 run infrastructure services, and their load is not controlled by our experiments. We do include their readings in our maps, but not in computing averages or other aggregate metrics for the experimental platform.

4.1 Key Value Store

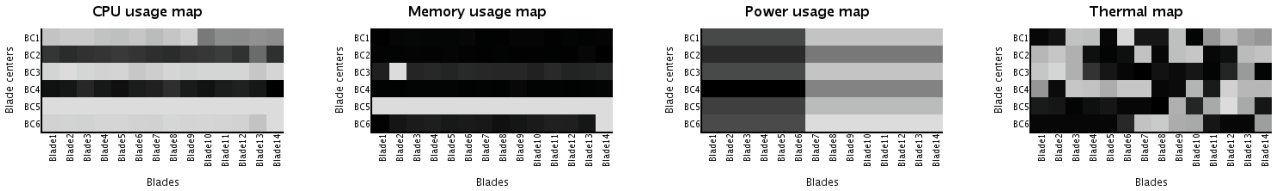
We first present the results for the Voldemort key-value store benchmark. Key-value stores allow users to store replicated data in a schema-less fashion with relaxed consistency models, and rely on efficient indexes to quickly locate data. As a result, some of the Internet's largest applications running in cloud deployments, such as Facebook Photo Store, Google App Engine, Amazon S3, etc. use these stores to serve vast amounts of data to millions of users. For our characterization we use the Voldemort [19] key-value store, known for its use at LinkedIn, and drive load to it using the Yahoo Cloud Serving Benchmark. Voldemort is characterized with relatively low CPU usage and high memory usage [12].

The usage maps in Figure 2 correspond to three different configurations obtained for a mix of three Voldemort instances deployed simultaneously on the platform. The three instances differ in the number of operations, value size, number of threads and target throughput, and result in different load levels. The top graphs correspond to the 'default' configuration (A), where BC1 runs the lowest load instance, BC2 the medium load instance, and BC4 runs the highest load instance. The middle graphs correspond to a configuration where the load is re-distributed within a BC so as to balance out the power across the two power domains (configuration B). In this configuration, a high-load VM is migrated within the same Blade Center but to a different power domain. The third configuration corresponds to a policy which migrates load across nodes in the rack so as to balance out the power usage across the rack (configuration C). Here, to increase resource utilization and reduce power usage, a low-load VM is consolidated with another low-load VM and then a high-load VM is migrated to that idle blade.

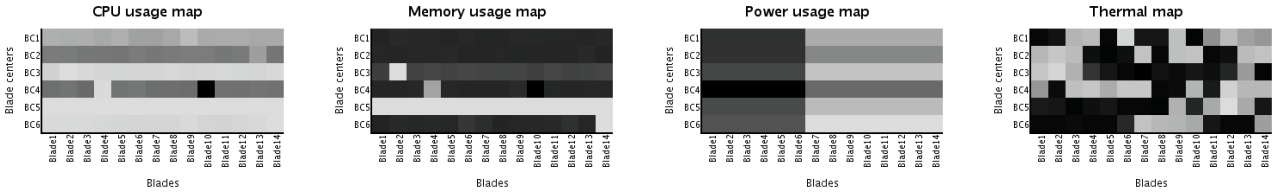
We next compute the effect of these different management decisions on a range of metrics. The results are shown in Table 1. We observe that, on our platform, for this workload distribution pattern, different management methods can lead to significant differences in metrics such as average CPU utilization or CPU imbalance (computed as $(max - min)/max$), but when considering energy-aware metrics, such as average power, temperature imbalance ($max - min$), or application performance/Watt (application performance here is the average throughput), the differences are not as dramatic. This indicates that for the target platform and workload configuration, for workloads such as Voldemort, it is not necessary to invest in complex energy-aware management methods, as the benefits may be very limited. For instance, across multiple runs, we obtain maximum gains in performance/Watt of $\tilde{6}\%$ and even less for power usage or temperature.

4.2 Map Reduce Computation

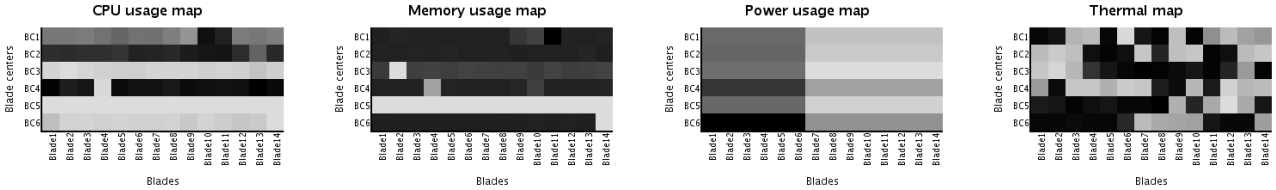
Next, we perform similar analysis for a representative map reduce task. The map-reduce framework and its open source implementation, Hadoop, have emerged as the de-facto standard for parallel computation and data analysis in the cloud. We use one representative map-reduce benchmark which estimates the value of Pi using the Monte-Carlo method. The benchmark takes 2 inputs: the number of mappers and the number of samples, which determines the accuracy of the Pi value. Similarly to the Voldemort measurements, our load consists of three instances of this benchmark, with different load levels. For each instance, we use 20 billion samples per map, but vary the number of mappers, thereby changing the resulting CPU utilization. Again we perform multiple runs for different workload configurations, to represent the result of potentially different management decisions. We represent the results for three configurations: default configuration A, where BC1 has 13, BC2 has 26 and BC4 has 52 mappers (a master is present in each BC), configuration B, with migrations happening across the Blade Center, and configuration



(a) Configuration A - Initial VM placement involving no migration



(b) Configuration B - Migrate a high-load VM within the same Blade Center but to a different power domain



(c) Configuration C - Consolidate two low-load VMs and migrate a high-load VM to the low-load Blade Center

Figure 2: Datacenter maps for different configurations of Voldemort workload.

Metrics	Config A	Config B	Config C
Avg CPU	15.88%	13.59%	11.39%
CPU imbalance	20.3%	32.3%	38.4%
Avg Memory	77.12%	76.41%	75.64%
Memory imbalance	2.99%	2.81%	14.6%
Throughput (ops/s)	195.98	201.7	185.9
Avg Read time (ms)	4.74308	4.32	5.12
Avg Write time (ms)	14.02	13.56401	15.12
Avg Power (Watt)	141.38	141.64	139.55
Avg Temperature (C)	52.36	52.52	52.00
Temp imbalance (C)	58	59	58
Performance/Watt	1.39	1.42	1.33

Table 1: Metric sensitivity for Voldemort

Metrics	Config A	Config B	Config C
Avg CPU	51.34%	50.68%	50.74%
CPU imbalance	35.5%	66.9%	65.5%
Avg Memory	65.35%	65.05%	64.29%
Memory imbalance	8.3%	7.3%	20.0%
Throughput	352316520	243257644	348292905
Avg Power (Watt)	154.0	152.5	151.0
Avg Temperature (C)	53.6	53.8	53.6
Temp Imbalance (C)	53	55	55
Performance/Watt	2287770	1595132	2306576

Table 2: Metric sensitivity for MapReduce

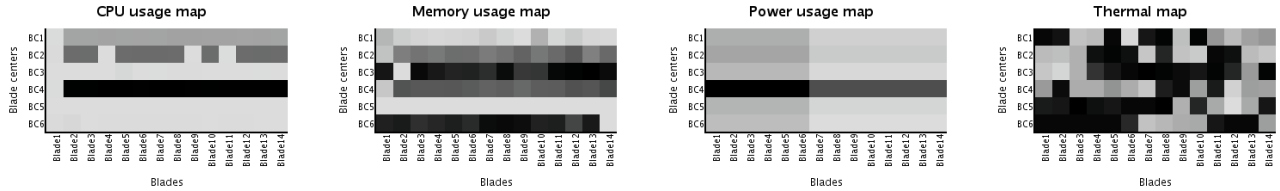
C, when additional consolidations of multiple VMs per node take place.

The results for the map-reduce example are summarized in Figure 3 and Table 2. We observe quite different trends regarding the energy-aware metrics. For instance, when considering a metric such as performance per Watt (performance here is samples per second), we measure that different management decisions can lead to differences of several tens of percents. Such potentially substantial gains can offset any incurred costs due to the increased overheads, complexity and resource requirements of the management infrastructure. It is therefore worth considering the corresponding investment.

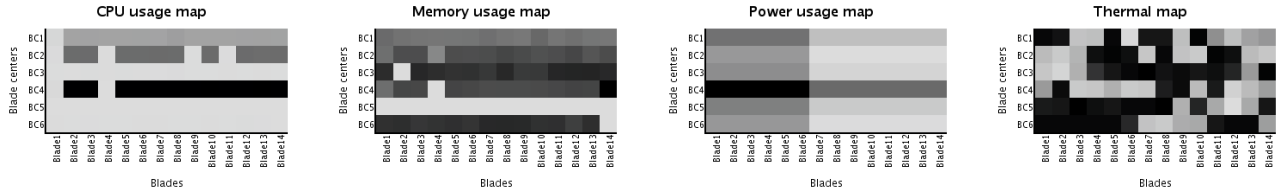
5. CONCLUSIONS AND FUTURE WORK

In this paper, we first experimentally illustrate the complex relationships which exist between the utilization of compute resources, such as CPU and memory, and the corresponding power utilization or temperature (heat generation) of the datacenter platform. These trends lead to development of complex energy-aware management methods. The question being addressed with the experimental approach presented in this paper, is whether increased complexity is always necessary. We choose two representative cloud workloads, and illustrate that different management decisions can have quite different impact on metrics capturing the energy-effectiveness of the management method. As a result, for certain classes of workloads it may be not be desirable to incur the added costs and complexities of such 'smart' management, since they lead to only marginal improvement. More generally, we illustrate the importance of understanding the impact that a management method can have on a target workload for the given platform, so as better assess its cost benefits. The methodology and infrastructure presented in this paper can be applied to provide such insights.

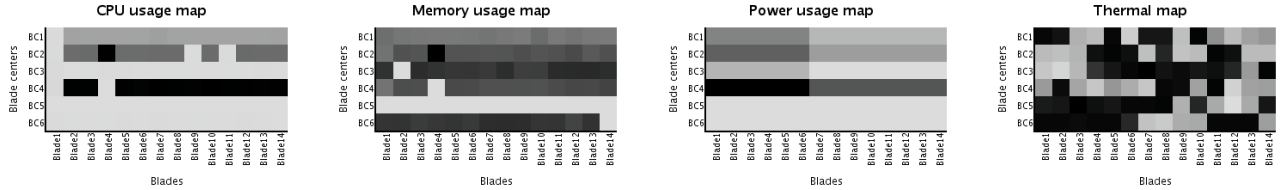
Our next steps are focused on providing greater understanding of the observations presented in this paper. The results presented here indicate that, on the hardware in



(a) Configuration A - Initial VM placement involving no migration



(b) Configuration B - Migrate a high-load VM within the same Blade Center but to a different power domain



(c) Configuration C - Migrate a high-load VM to a different Blade Center but in the same power domain

Figure 3: Datacenter maps for different configurations of MapReduce workload.

our testbed, for memory-intensive workloads such as the Voldemort key-value store, different management policies and workload placement decisions have little impact on metrics such as average power usage or performance/Watt, whereas for CPU-intensive workloads the impact can be substantial. However, additional experimental results, not shown here for brevity, indicate that this may further depend on the aggregate load level (low vs. high), the imbalance in the CPU resource usage, and other factors. Therefore, further understanding of these observations, requires, on one end, more detailed characterization of the required management complexity, by experimentally building models to capture the relationship between the utilization levels of the compute resources (e.g., CPU, memory, etc.) and the corresponding power, temperture, fan speeds, etc., at node, chassis (i.e., Blade Center) and rack-level. On the other end, it requires more rigorous characterization of the workloads' resource utilization and corresponding performance levels. By gaining detailed understanding of the observed behavior on our platform, our goal is to further generalize a methodology that datacenter operators can easily apply to assess the costs and benefits of different management policies for their typical workload patterns, on their platform, and for their target metrics.

6. REFERENCES

[1] Amur, H., et al. Robust And Flexible Power-proportional Storage. In *SOCC'10*.

[2] Bohra, A., and Chaudhary, V. Vmeter: Power Modelling for Virtualized Clouds. In *IPDPS'10*.

[3] Campbell, S. Data Center Power Consumption to Grow 20 Percent in 2012.

[4] Chen, H., et al. Spatially-aware Optimization of Energy Consumption in Consolidated Datacenter Systems. In *InterPACK'11*.

[5] Chen, H., Xiong, P., Gavrilovska, A., Schwan, K., and Xu, C. A Cyber-Physical Integrated System for Application Performance and Energy Management in Data Centers. In *IGCC'12*.

[6] Vmware Distributed Power Management Concepts and Use. www.vmware.com/ĭňĄles/pdf/DPM.pdf.

[7] Fanara, A. Report to congress on server and data center energy efficiency. Tech. rep., U.S. Environmental Protection Agency, Energy Star Program, August 2007.

[8] Goiri, I., et al. GreenSlot: Scheduling Energy Consumption in Green Datacenters. In *SC'11*.

[9] Apache Hadoop. hadoop.apache.org.

[10] Apache Hbase. hbase.apache.org.

[11] Kansal, A., Kothari, N., and Bhattacharya, A. Virtual Machine Power Metering and Provisioning. In *SOCC'10*.

[12] Kesavan, M., Gavrilovska, A., and Schwan, K. Xerxes: Distributed Load Generator for Cloud-scale Experimentation. In *7th OpenCirrus Summit* (2012).

[13] Koller, R., Verma, A., and Neogi, A. WattApp: An Application Aware Power Meter for Shared Data Centers. In *ICAC'10*.

[14] Krishnan, B., Amur, H., Gavrilovska, A., and Schwan, K. VM Power Metering: Feasibility and Challenges. In *GreenMetrics'10*.

[15] Kusic, D., Kephart, J., et al. Power and Performance Management of Virtualized Computing Environments via Lookahead Control. In *ICAC'08*.

[16] Lim, H., Kansal, A., and Liu, J. Power Budgeting for Virtualized Data Centers. In *USENIX ATC'11*.

[17] Nathuji, R., and Schwan, K. VPM Tokens: Virtual Machine-Aware Power Budgeting in Datacenters. In *HPDC'08*.

[18] Stoess, J., Lang, C., and Bellosa, F. Energy Management for Hypervisor-based Virtual Machines. In *USENIX ATC'07*.

[19] Project Voldemort. A Distributed Database. project-voldemort.com.