A Case For Adaptive Datacenters To Conserve Energy and Improve Reliability

Peter Bodik
Michael Paul Armbrust
Kevin Canini
Armando Fox
Michael Jordan
David A. Patterson

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2008-127
http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-127.html

September 26, 2008

Acknowledgement

RAD Lab research is supported by foundational sponsors Google, Microsoft, and Sun Microsystems, and by affiliate sponsors Fujitsu America, HP, IBM, Nortel Networks, and Oracle. Their support is enhanced through the U.C. Discovery Grant and Industry-University Cooperative Research Program of the state of California. In addition to our RAD Lab colleagues, we thank Luiz Barroso of Google, Ricardo Bianchini of Rutgers, Eric Brewer of UC Berkeley, George Candea of EPFL, Dave Douglas of Sun, Jeff Mogul of HP, and Partha Ranganathan of HP for their comments and Joe Gebis of UC Berkeley for his IC physics expertise.

A Case For Adaptive Datacenters To Conserve Energy and Improve Reliability

Peter Bodík, Michael Armbrust, Kevin Canini, Armando Fox, Michael Jordan, and David Patterson,
{*bodikp, marmbrus, kevin, fox, jordan, pattrsn*}*@cs.berkeley.edu*
UC Berkeley RAD Lab

Abstract

Although there is prior work on energy conservation in datacenters, we identify a new approach based on the synergy between virtual machines and statistical machine learning, and we observe that constrained energy conservation can improve hardware reliability. We give initial results on a cluster that reduces energy costs by a factor of 5, reduces integrated circuit failures by a factor of 1.6, and disk failures by a factor of 5. We propose research milestones to generalize our results and compare them with recent related work.

1. Problem: Energy Efficiency vs. Performance

Power is ranked as the #5 top concern of IT executives [Sca06], with availability and performance being #10 and #11 respectively.[1] This is not surprising given that powering and cooling a datacenter now rivals the cost of the hardware: each $1 spent on servers in 2005 required an additional $0.48 to power and cool it, expected to rise to $0.71 by 2010 [Sca06]. Also, new problems arise as power becomes the constraining resource in datacenters:

1) Space limited by power budget: The University of Buffalo's $2.5M Dell datacenter caused a brownout when switched on, because the operators had failed to arrange for extra power circuits to handle the greater load [Cla05]. Their response was to underutilize the datacenter to avoid another brownout. Similarly, a datacenter at Ask.com is only 2/3 full due to limited power availability [Gil06].

2) Power emergencies: Large power consumers must handle on-demand "agile conservation" such as Pacific Gas & Electric ordered to prevent rolling blackouts during peak power demand [Bra06].

3) Thermal emergencies: Cooling systems can be overtaxed by "hot spots" in a datacenter, leading to downtime or ruined equipment.

At the same time, workload peaks can exceed average utilization by 5x, and datacenters are provisioned for these peaks because compliance with service-level agreements (SLAs) trumps average efficiency [Sca06]. Hence, if we could dynamically turn off underutilized equipment *and* guarantee no impact on meeting SLAs, we could save, say, 4/5's of the power used by datacenters.

Our position is that statistical machine learning (SML) will be the key enabling technology for making policy decisions about turning equipment on and off, and virtual machine technology (VM) will be the enabling mechanism.

[1] The top concerns are security, system management tools, virtualization, product road map, power consumption, ease of deployment, interoperability, scalability, features and functionality, availability, performance, and product portfolio breadth.

In this paper we motivate this position, present initial proof-of-concept results, and propose research milestones. We also address the concern that power cycling equipment reduces reliability, explaining how careful power cycling decisions could *improve* component reliability.

2. The Promise of SML for Making Policy

The promise of statistical machine learning rests in part on recent theoretical progress and in part on the fact that techniques languishing since the 1960's have become practical due to computers becoming 100,000 times faster. Moreover, cheaper computers make resources for monitoring and analysis affordable. This is fortunate, because we find four reasons for enthusiasm about SML.

First, SML techniques work well in dynamic environments where transients (e.g., resource reallocation times) impact performance in complex ways. For example, Tesauro et al. [TJDB06] found that Reinforcement Learning (RL) performed better than queuing theory for making dynamic server allocation decisions, in part because RL learned to "ride out" the short-term effects of transition behavior. Accomplishing the same result using queuing theory would have required avoiding sampling during transitions between steady states, yet performance during these transitions is critical. Queuing theory also cannot inform *tradeoffs* between power and performance.

Second, techniques such as RL can handle arbitrary non-linear cost/reward functions. For example, to shape behavior of datacenter operators, Pacific Gas & Electric plans to pick (on short notice) a handful of days each year during which the cost of electricity will be many times higher than the average, and then slightly drop the cost the remaining days [Bra06]. Adaptive operators could achieve major savings if they could exploit these complex reward functions without sacrificing performance.

Third, SML techniques adapt as hardware/software configurations change. For example, eBay and Amazon push 100 software changes in a typical month [BFJ+06]. Whereas SML can automatically adapt to some of these changes, as demonstrated by the use of ensembles of SML models to capture changing system behavior [ZCM+05], queuing models must be redone by experts when the system configuration changes. The problem of managing a datacenter is too large for constant manual intervention; hence, we need techniques that can adapt to such changes, as SML can.

Lastly, because conserving energy is an optimization and not a guarantee, useful progress can be made even though SML models, algorithms and heuristics are imperfect, as long as performance is preserved. Indeed, we demonstrate this by simulation in section 5.

3. VMs as an Enabling Mechanism

Good operational practice requires that applications be fault-isolated and resource-isolated from each other. In addition, different applications may require different versions of the OS kernel, libraries, drivers, or, in extreme cases, even a different OS. The problems of application diversity, resource containment, and fault isolation arise even in datacenters of a single operator. For example, Amazon's datacenters host hundreds of independent services that cooperate to produce a single user experience [Gray06].

Previously, isolation and application diversity were supported by putting applications on separate machines. Today, virtual machines achieve these goals without requiring hardware isolation, and recent advances in VM technology have made overhead of virtualization low enough that VM's have become widely used in the datacenter. Given the popularity of network-attached storage, to a first order, we can now treat all comparable computers with the same ISA as equivalent for the purposes of resource allocation. This interchangeability greatly simplifies the scheduling problem faced by SML in deciding which machines to turn off.

4. Conserving Energy and *Improving* Reliability

Although we will present initial results that SML+VM can achieve power savings without SLA violations, a separate concern is that turning equipment on and off frequently could hurt reliability, because expansion and contraction due to thermal fluctuations can accelerate mechanical failure. Hence, another requirement is to guarantee no lowering of reliability due to power cycling.

Surprisingly, we hypothesize that it may be possible to *improve* reliability by turning equipment off when it is not needed. Within reasonable limits, we can increase reliability by increasing the *average time components are off* while respecting a *maximum number of on/off transitions*.

For example, disks rotate (causing wear and tear) even when they are not reading or writing, so disk manufacturers' stated reliability rates are based on the number of hours a disk is powered on. Turning a disk off should extend its expected lifetime; keeping it off half the time should cut annual failure rates roughly in half. Obviously, while you can't reduce the failure rate by a factor of 99 by leaving the disk off 99% of the time, this guideline should work for, say, factors of 2 to 6. Disk manufacturers also specify a limit on the number of times a disk can be turned off and on before failure—typically 40,000 to 50,000 start/stops, which works out to about one start/stop per hour over the nominal 5-year lifetime of a disk.

As another example, chips that save power using dynamic voltage scaling can change voltage as fast as tens of milliseconds. Recent results suggest that for newer chips with smaller geometries, the thermal cycling resulting from very aggressive changes in power can accelerate failure rates of integrated circuits by a factor of three, to 10% per year [CSRM+06]. In contrast, the same paper suggests the reliability benefits of turning chips off for longer periods at a time, which reduces chip failure due to electromigration and time-dependent dielectric breakdown. Thus, if we can remove thermal cycling as a concern, a chip off half the time should be more reliable by roughly a factor of 1.5.

It is not simple to calculate how many on/off cycles it would take before thermal cycling becomes important; it is a function of the materials, the difference in temperature between on and off, and other factors. However, even if we assume a very wide temperature difference, integrated circuits can likely tolerate many more on-off cycles than disks. We therefore make the simplifying assumption that since servers include both integrated circuits and disks, observing the disk limit of on/off cycles of about once an hour is sufficient to remove thermal cycling as a reliability issue for ICs.

We recognize that the datacenter's software and infrastructure must be designed to accommodate component failures without disrupting service. Nonetheless, improving component reliability means fewer hassles due to failed parts, thereby lowering costs of operation.

5. A Simulated Experiment

Our preliminary experiments using the 1998 World Cup trace [Arl98] suggest a potential energy savings of nearly 80% and potential reliability improvements of 1.6 to 5 while satisfying a 100 ms SLA for more than 99.9% of requests in the trace. In this section, we sketch the experimental approach and discuss the results and caveats; the subsequent section suggests research milestones for further progress.

5.1. Experimental Setup and Approach

The publicly available World Cup workload is an HTTP trace in which 98% of the requests are for nearly-fixed-size, static-content objects. This workload is simple by today's standards; in section 6 we propose concrete research milestones not only for generalizing our results to more complex workloads, but also for providing enabling technologies to improve researchers' access to realistic traces.

We ran a "spiky" 29-day subset of these traces against a cluster of six dual-CPU 1GHz x86 workstations running VMware ESX server and Apache on each VM. Consistent with industrial practice, we define SLA compliance as the 90th percentile of response time meeting a threshold of 100 ms. We used the measured behavior of this setup to calibrate a Matlab simulation in which VM's are suspended and resumed as workload fluctuates. We assume that machine power-down and reactivation (reboot, resume VM, reconnect to load balancer) take 2 minutes each; others have reported 45 seconds [PBCH03] so this is conservative.

Our approach relies on two predictor functions. The first predicts near-future workload based on previous workload. For the World Cup trace, trial and error revealed that linear regression on the last 5 minutes of workload yielded accurate prediction of the next 2 minutes, the necessary lead time to power up a machine. The burstier the workload, the more challenging it would be to make this predictor accurate.

To calculate the number of machines required to handle the predicted workload, we first estimate the maximum

number of requests per second *Rmax* that each machine can handle. We adjust *Rmax* every 20 seconds to avoid SLA violations and (as a safety margin) keep CPU utilization below 80%. From Figure 1, which shows these two constraints, we see that this simple model is sufficient for our single-tier application and simple workload. In section 6.1 we discuss more complex applications and workloads.

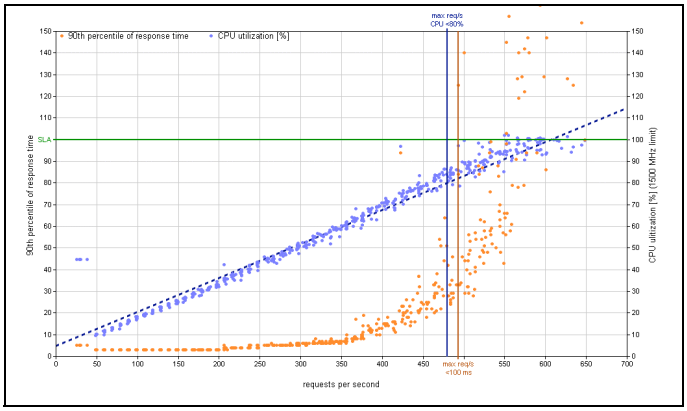

Figure 1. Data from an experiment on the cluster used to compute *Rmax*. The two vertical lines represent two constraints: 1) no SLA violations (orange line) and 2) CPU utilization below 80% (blue line). rps_{max} was set to the lower of these two numbers.

Given *Rmax*, a naive policy would maintain exactly ⌈W/*Rmax*⌉ machines turned on, where W is the predicted workload. To reduce potential thrashing, we tried simple hysteresis and use ⌈W/*Rmax* + buffer⌉ machines, where buffer is 0.1 (i.e., 1/10 of a machine) for turning a machine on, and 0.5 for turning a machine off. Hence, this policy is turns on conservatively and turns off very conservatively.

5.2. Power Savings and Reliability Improvement

To determine power savings, based on hardware measurements [JS06] we estimated our machine's power consumption in steady state as (110+0.6*R*) watts, where *R* is the number of requests per second the machine is serving.

Table 1 shows that we can get nearly 80% power savings while penalizing only 1 in 1,000 to 1 in 100,000 requests. The bottom row shows that we can get most of this savings while dramatically reducing the number of policy changes. This is important because each policy change corresponds to toggling a machine from on to off or vice-versa, and excessive power cycling can reduce reliability. In addition, reducing the total number of policy changes also mitigates the inherent "risk" of powering up a new system—extra time required for cache warming, reintegrating the new machine back into the network infrastructure, and other hard-to-predict warm-up effects. Table 2 shows the resulting reliability improvement from reducing the number of policy changes. Note that even the naive model limits changes to 1 power cycle per disk per hour.

Model	% SLA violation	Energy (kWh)	Normalized energy	# Policy changes
Everything on	0	629.0	1.00	0
Naive model	0.124%	121.9	0.19	1012
Hysteresis	0.005%	139.1	0.22	143

Table 1. Matlab simulation of three policy models for adaptive power conservation (1998 World Cup trace, days 40-69). Adding hysteresis greatly reduces number of policy changes while still conserving most of the energy.

Model	Disk Reliability	IC Reliability	Max/Avg # on/ offs/hour /server
Everything on	1.0	1.0	0 / 0
Naive model	5.2	~1.6	0.69 / 0.18
Hysteresis	4.5	~1.6	0.05 / 0.02

Table 2. Estimated improvement in disk and integrated circuit reliability using data in Table 1. The estimate for an improvement in chip reliability of 1.6 with an average off fraction of 0.79 is an extrapolation from an estimate of 1.4 with an average off fraction of 0.50 [CSRM+06]. Note that the maximum on/offs per hour for the Naïve policy is close to the recommended limit for disks.

Although hysteresis reduces the number of on/offs per server, for reliability we are also concerned about the *distribution* of those changes across servers. Although the average of 0.18 is within our safety margin even for the naive policy, the maximum is a factor of 4 worse. A better approach would exploit VM migration and network-attached storage to decouple the decision of *how many* servers to use from the decision of *which* servers to use, and distribute the power-cycles more evenly.

6. Research Milestones

Our enthusiasm for our preliminary results is tempered by the limitations of our simplified setup:

- We argued that real workloads are complex and dynamic, yet ours is simple. Indeed, complex realistic workloads are hard to obtain, yet critical for credibility of results.
- We emphasized the need to plan for multiple applications sharing a datacenter, yet we used only a single application.
- We argued that the problem scale is thousands of computers per datacenter, yet we used a trivial-sized cluster.

The research milestones we propose are an attempt to make progress on each of these fronts.

6.1. More Complex Applications and Workloads

Although our simple computation of *Rmax* using only response time and CPU utilization was sufficient for capacity planning, the performance of modern applications depends on relationships among collections of many low-level metrics. Zhang et al. [ZCM+05] showed that a Bayesian network that accurately predicts SLA violations could capture these relationships. We will repeat our experiments using complex applications and workloads using such techniques to develop the necessary predictor functions, and see how this affects both energy savings and SLA compliance.

6.2. Modeling Multiple Applications & Workloads

Companies are understandably reluctant to release workload traces to academic researchers, and even when traces are available, it is impractical for researchers to deploy hundreds of applications against which to playback the traces. We propose to use SML both to characterize real workloads in a way that enables synthetic generation and to characterize the *effect* of workloads on applications in a way that an "application simulator" can faithfully reproduce.

Specifically, we plan to investigate using Kernelized Canonical Correlation Analysis to cluster trace requests that elicit similar system-level behavior. The cluster parameters, plus the observed distribution of cluster instances in the original trace, could be given to researchers who would then synthesize a similar workload and scale it up using stratified sampling. Since these clusters amount to descriptions of system effects (e.g. requests of type X result in 70% CPU resource utilization + 100Kbyte network traffic), we could then build an "application simulator" (similar to [RLI+06]) that, say, mimics serving a file by consuming the appropriate resources and emitting the right number of bytes. Researchers could then use synthetic but realistic workloads and simulated but realistic applications to validate their results.

6.3. Scaling Up to 10,000-Processor Datacenter

How can universities convincingly emulate a datacenter of 1,000–10,000 processors? RAMP (Research Accelerator for Multiple Processors) [WPOK+2006] is a multi-university project using FPGA's to build affordable emulation hardware for large-scale multiprocessors. We believe we can repurpose RAMP to emulate key aspects of datacenters for $10–$100 per processor; we have already used it to emulate the performance and thermal behavior of disks using Mercury/Freon [HCG+06,Tan06]. RAMP's low cost makes it a viable *dedicated* emulation facility (unlike PlanetLab, for example), allowing more experiments.

6.4. Empirical Evaluation of Reliability Claims

While the reliability hypothesis of section 4 is sound in principle, we know of no empirical data to support it, in part because today's datacenters generally do not power down unused equipment. We are pursuing empirical evaluation not only on research clusters such as DETER [Det04], in real datacenters. While ambitious, this is consistent with existing practice in real datacenters of deploying "risky" changes incrementally by first testing them on a small part of the datacenter [AWL05]. Such experiments could also validate the power savings and performance guarantees.

6.5. Storage System Design for Adaptive Datacenters

Since our strategy involves turning off whole machines, we are assuming it's OK for those machines' disks to go away. Many, but not all, three-tier applications are structured so that local disks are used for caching and temporary storage, so application correctness is not affected by turning them off (though performance may be). A more thorough investigation of storage system design to facilitate adaptive datacenters would be another important milestone.

6.6. Cross-Datacenter Migration for Power Efficiency

In Tokyo, electricity costs 30¢/kWh at peak, compared with 4¢/kWh near a US hydroelectric plant—more than a factor of seven difference. This suggests dynamically moving work *across* datacenters, essentially doing global load balancing for power [PSB+03]. The tradeoff is the higher cost of migration and possible higher latency for some users, but the same techniques we propose for making migration decisions within a datacenter should generalize readily to the cross-datacenter case.

7. Related Work

Pinheiro, Bianchini *et al.* [PBCH03] propose turning off unused machines for an Internet service in a datacenter. However, they do not rely on VMs, and their model is a single application per datacenter. Single-company datacenters run tens or hundreds of independent "services" [BFJ+06], so any solution must generalize to this case.

A queueing-theoretic approach to managing datacenter resources, including power, is proposed in [CDQ+05]. They assume an adaptation interval large enough that the steady-state results of QT are valid, that the time to service a request varies linearly with the request frequency, and that the overall service capacity allocated to an application is directly related to the number of servers allotted to it. The first assumption ignores performance during transitions, which may not be acceptable. The second assumption generally holds for CPU-bound applications but not for more complex applications and workloads [ZCM+05a]. The third assumption may hold for single-tier applications, but in multi-tier applications it is rarely obvious which tier should have machines added to it in order to improve overall application performance. The paper also accounts for the cost of turning a power cycling a disk by charging essentially $1/50{,}000^{th}$ of the cost of the disk for each on/off cycle. This point is different from Section 4 that argues keeping components off for longer periods can improve reliability.

Some have advocated spreading load equally across all machines and operating each machine in a lower-performance (therefore lower-power) mode, which does not require software or OS involvement and allows very rapid transitioning back to full-power/high-performance mode. However, the potential power savings are limited: Chase et al. [CAT+01] find that an idle x86 server consumes 60-74% of the power it needs when fully utilized; it was 70% for our cluster [JS06]. Others have suggested using voltage scaling to save power on less-utilized processors, but as mentioned above, rapid thermal cycling can lower component reliability [CSRM+06] whereas leaving components off can potentially improve reliability.

A related topic is avoiding hotspots in datacenters in order to lower the costs of cooling [MCR05]. We can leverage SML to add hotspot-awareness to a scheduling algorithm for

turning machines off, and exploit VM migration to decouple this decision from physical hardware.

Measurements of server energy consumption have focused primarily on batch-style benchmarks [ERKR06], which fail to characterize transitions to/from steady state or the dynamic nature of datacenter workloads and may have limited utility in predicting interactive-service power consumption.

8. Conclusions

We believe statistical machine learning combined with low-overhead virtual machine technology can not only save significant power in datacenters but also potentially increase the reliability of the hardware, by turning off portions of it for significant fractions of a day. VM's provide software isolation without hardware isolation and advances in SML yield techniques that can set policy based on a large number of dynamically changing inputs and a complex cost function.

Based on 2005 data [Sca06], reducing server power consumption by 30% would yield savings of 150 billion kWh, corresponding to US$15B and almost 100 million metric tons of CO_2 emissions. To put that into perspective, California passed a law to cut emissions in the state by 175M metrics tons by 2020. Adaptive datacenters by themselves might be able to save half that amount much sooner.

Acknowledgments

RAD Lab research is supported by foundational sponsors Google, Microsoft, and Sun Microsystems, and by affiliate sponsors Fujitsu America, HP, IBM, Nortel Networks, and Oracle. Their support is enhanced through the U.C. Discovery Grant and Industry-University Cooperative Research Program of the state of California.

In addition to our RAD Lab colleagues, we thank Luiz Barroso of Google, Ricardo Bianchini of Rutgers, Eric Brewer of UC Berkeley, George Candea of EPFL, Dave Douglas of Sun, Jeff Mogul of HP, and Partha Ranganathan of HP for their comments and Joe Gebis of UC Berkeley for his IC physics expertise.

References

[Arl98] M. Arlitt and T. Jin. 1998 World Cup Web Site Access Logs. *http://ita.ee.lbl.gov/html/contrib/WorldCup.html*

[AWL05] Mike Afergan, Joel Wein, Amy LaMeyer. Experience with some principles for building an Internet-scale reliable system. Proc. *2nd Workshop on Real, Large Distributed Systems*, Dec. 2005

[BFJ+06] Peter Bodík, Armando Fox, et al. Advanced tools for operators at amazon.com. In *Hot Topics in Autonomic Computing (HotAC)*, 2006.

[Bra06] Mark Bramfitt, PG&E. Personal communication, 2006.

[CAT+01] Jeff Chase, Darrell C. Anderson et al. Managing energy and server resources in hosting centers. In *Symposium on Operating Systems Principles (SOSP)*, 2001.

[CDQ+05] Y. Chen, A. Das et al. Managing server energy and operational costs in hosting centers. In *Proc. ACM SIGMETRICS,* June 2005.

[Cla05] Don Clark. *Power Hungry Computers Put Data Centers In Bind.* Wall Street Journal Online, Nov. 14, 2005, page A1.

[CSRM+06] A. K. Coskun, T. Simunic Rosing, K. Mihic, G. De Micheli, and Y. Leblebici.Analysis and Optimization of MPSoC Reliability. *J. Low Power Electronics*, 2(1): 56–69, 2006.

[Det04] DETER, A Laboratory for Security Research, Testbed Overview. http://www.isi.edu/deter/docs/testbed.overview.htm

[Gray06] Jim Gray. *A Conversation With Werner Vogels.* ACM Queue 4(4), May 2006.

[ERKR06] Dimitris Economous, Suzanne Rivoire, Christos Kozyrakis, and Parthasarathy Ranganathan. Full-system power analysis and modeling for server environments. In *Workshop on Modeling, Benchmarking, and Simulation (MoBS),* June 2006.

[Gil06] George Gilder. The information factories. *Wired Magazine,* 14.10, October 2006.

[HCG+06] Taliver Heath, Ana Paula Centeno et al. Mercury and Freon: Temperature emulation and management for server systems. In *Architectural Support for Programming Languages and Operating Systems (ASPLOS),* 2006.

[JS06] Pallavi Joshi and Mark Shlimovich. Power and performance of virtual machines. Class proceedings, UCB CS294-1, Fall 2006.

[MCR05] Justin Moore, Jeff Chase, Parthasarathy Ranganathan, Ratnesh Sharma. "Making Scheduling Cool: Temperature-aware Resource Scheduling," *Proceedings of the 2005 Annual Usenix Conference*, April 2005.

[PBCH03] Eduardo Pinheiro, Ricardo Bianchini et al. Dynamic cluster reconfiguration for power and performance. In *Compilers and Operating Systems for Low Power (COLP),* Kluwer Academic Publishers, Sept. 2003.

[PSB+03] C.D. Patel, R. Sharma et al. Energy aware GRID: global service placement based on energy efficiency of the cooling infrastructure. Proc. *ASME Intl. Mech. Eng. Conf.,* June 2003.

[RLI+06] Ensemble-level Power Managementfor Dense Blade Servers. In Proc. *ISCA 2006.*

[Sca06] Jed Scaramella. *Worldwide server power and cooling expense 2006 – 2010 forecast.* IDC, 2006.

[Tan06] Zhangxi Tax, Disk and Thermal Emulation using RAMP, 2006, class proceedings, UCB CS294-1, Fall 2006.

[TJDB06] Gerald Tesauro, Nicholas Jong, Rajarshi Das, and Mohamed Bennani. A hybrid reinforcement learning aproach to autonomic resource allocation. In *International Conference on Autonomic Computing (ICAC),* 2006.

[WPOK+2006] J. Wawrzynek, D. Patterson, M. Oskin, S.-L. Lu, C. Kozyrakis, J.C. Joe, D. Chiou, and K. Asanovic, "RAMP: A Research Accelerator for Multiple Processors," U.C. Berkeley technical report, EECS-2006-158, 2006.

[ZCM+05] Steve Zhang, Ira Cohen et al. *Ensembles of Models for Automated Diagnosis of System Performance Problems.* In *Proc. Intl. Conf. on Dependable Systems and Networks* (DSN) 2005.

[ZCM+05a] Steve Zhang, Ira Cohen et al. *Clustering, Indexing and Retrieving System History.* Proc. SOSP 2005.