Validated Simulation for Cloud Scheduler Engineering

Alexander Pucher, Rich Wolski, and Chandra Krintz
Department of Computer Science
University of California, Santa Barbara
{pucher, rich, ckrintz}@cs.ucsb.edu

***Abstract*—Researchers have contributed promising new techniques for allocating cloud resources in more robust, efficient, and ecologically sustainable ways. Unfortunately, the wide-spread use of these techniques in production systems has, to date, remained elusive as their evaluation frequently relies on exploratory, model-driven simulation only.**

We present a new methodology that complements existing model-driven simulation with platform-specific and statistically validated quantitative results. We simulate systems at scales and on time frames that are testable, and then, based on the statistical validation of these simulations enable investigation of scenarios beyond those feasibly observable in practice.

I. Introduction

Cloud computing realizes the long-held vision of computing as a utility and is increasingly successful in both public provider and private enterprise settings. Public cloud providers such as Amazon AWS [1] and Google Cloud Platform [2] often hold the specifics of their implementation as trade-secrets and validation of academic research results in production settings is difficult. Fortunately, private enterprise clouds are often based on open-source IaaS cloud frameworks such as Eucalyptus [3], [4] and OpenStack [5] which allow for in-depth analysis, customization, and optimization and have attracted the interest of the research community. Software deployed in mission-critical infrastructure, however, must be reliable. Thus, modifications and extensions to open-source clouds require production quality engineering and thorough testing before they can be deployed.

As a specific example, when introducing new software components into Eucalyptus IaaS, decision makers are faced with the question of whether an engineering effort will pay off, which breaks down into two parts:

- Will this intrusive modification break the system or threaten its stability in corner cases?
- Do its benefits under real-world conditions out-weigh the investment required for engineering and quality-assurance?

Validated simulation provides reliable answers to both questions to better inform decision making by engineering leadership. Furthermore, validated models can guide the design and implementation process and ease the transfer of research artifacts into production. We emphasize this approach is complementary to existing simulators, as it trades off ease of modification and flexibility for quantitative accuracy.

Our method for building validated simulation models is inspired by Perturbation Theory [6]. We employ an end-to-end modeling approach that starts with a parsimonious analytical model of the system. We then incrementally "perturb" the model by adding empirical noise terms to improve accuracy.

II. Creating Accurate End-To-End Simulation Models

The goal of our methodology is to use an, alternative, "top down" approach to simulation that models only those parameters that are necessary to capture the behavior of the component of interest with sufficient accuracy. Our work explores an approach rooted in *perturbation theory* [6] that focuses on validation of simulated results against empirical measurements (at the cost of flexibility and extensibility) as a way of addressing the engineering needs that cloud developers and practitioners have. Identifying the parameters of this model requires an understanding of the fault isolation properties of the platform which, in our example use case, comes from source code inspection. The fault isolation properties establish the independence of our model parameters which is required for trustworthy scaling of our simulations.

The approach is to:

1) start with the most parsimonious model of end-to-end behavior that is possible, identified via white-box inspection of the framework architecture,
2) perturb the model using statistical sampling technique to represent unmodeled behavior,
3) test the model by comparing its outputs generated in simulation to measurements taken from the "real world" system,
4) if the model is insufficiently accurate, add terms, adjust the perturbation, and repeat.

Thus every addition of a variable to our model of the cloud should be justified by a necessary increase in accuracy. Variables that only contribute marginally to the aggregate result are omitted and modeled in aggregate as "perturbing" error terms. Since this "noise" is not deterministic our approach takes the form of Monte-Carlo simulation with statistical results.

We then extract performance information from a running private cloud to fit and evaluate the simulation model. This requires two types of measurements: those we use to introduce perturbations (e.g. VM start-up, termination, etc.) and those that we use to validate the model predictions (the aggregated CPU-time and up-time).

TABLE I: Utilization per Node (Log-normal trace)

	All	A	B	C	D	E	F
sim (mean)	0.3974	0.8555	0.7305	0.5140	0.1960	0.0665	0.0217
sim (sd)	0.0045	0.0024	0.0053	0.0082	0.0039	0.0001	0.0023
real (mean)	0.3985	0.8550	0.7223	0.5213	0.2025	0.0696	0.0202
real (sd)	0.0043	0.0022	0.0046	0.0059	0.0050	0.0037	0.0036

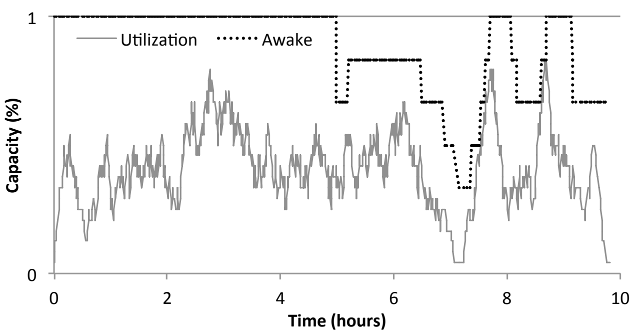

Fig. 1: Workload trace with power-optimizing scheduler activated at the 5 hour mark. The dotted line shows the fraction of online nodes.

III. Fitting and evaluating the model

For our empirical measurements we use a seven node commodity hardware cluster. Each node runs on CentOS v6.5 and holds four cores, 8 GB ram, and a 500 GB hard drive and is connected to the network via two 1 Gbit Ethernet links. We set up Eucalyptus v3.4.2 with a dedicated head and storage node and six nodes serving as instance hosts.

We first execute a benchmark trace on a single node, which has been separated from the seven node Eucalyptus cloud to collect "noise" distributions without interfering with the system at large. Specifically, we collect data on instance launch delay and the termination delay.

To evaluate the accuracy of our simulation model we then execute an independent test trace on both, our simulator and our real-world testbed. The workload has a duration of 10 hours and a mean utilization of 1/3 of the 6 node cluster and activates the power-management feature of the scheduler at the 5 hour mark. The trace is generated from a log-normal distribution for inter arrival time and durations. The mean inter arrival time is 81 seconds and the mean duration is 785 seconds.

Our results show agreement between prediction and real-world measurement as scale. We repeat simulation and real world runs 12 times (a total of 120 hours) for each trace separately and compute the averages. For visualization, an exemplar run from the benchmarks is shown in the graph in Figure 1. The figures depict the activity of the power management over time. The y-axis represents the number of cores used, normalized to maximum capacity. The x-axis represents time in one hour (3600 seconds) intervals. The solid line shows the number of cores occupied by instances in the cluster.

The average utilization per node and the respective standard deviation for the trace can be found in Table I. We find a good match between simulation and real world observation, with the largest per-node difference of less than 1 percent.

IV. Related Work

We base this paper on an original version [7] that appeared in the IEEE International Conference of Cloud Engineering. Our work leverages a significant body of work on distributed systems technologies and on methodologies and best practices for validation of large scale systems using real-scale experiments, emulation, benchmarking, and simulation [8]. The authors of this survey discuss the importance of ab initio (high-level, imprecise, easily composable, and extensible simulation for use in comparative analysis and exploration) and validated simulation (simulation that produces behavior that matches that of a real system with low error).

We also gained insights about the specific workloads in private clouds (which we use to drive our synthetic workload generation) from our previous work described in [9].

V. Conclusions and Future Work

Simulation plays a key role in performing experimental exploration into large scale systems and has significant potential for facilitating research and experimentation with cloud systems. Existing approaches aim for ab-inito, exploratory simulation of cloud systems only. The transfer of technology into production deployments, however, requires a degree of quantitative accuracy they do not provide.

In this work, we present a new, complementary methodology for facilitating trust in the quantitative accuracy of cloud simulation inspired by a tool employed in the physical sciences for simulation called perturbation theory. We find that the perturbation approach achieves high accuracy for simulating an existing system and creates a foundation for evaluating the efficiency of cloud schedulers ahead-of-time.

Acknowledgments

This work was funded in part by NSF (CNS-0905237, CNS-1218808, and ACI-0751315) and NIH (1R01EB014877-01).

References

[1] "Amazon Web Services home page," http://aws.amazon.com/.

[2] "Google Cloud Platform," [Online; accessed Aug-2014] "https://cloud.google.com/".

[3] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, "The eucalyptus open-source cloud-computing system," in Cluster Computing and the Grid, 2009. CCGRID'09. 9th IEEE/ACM International Symposium on. IEEE, 2009, pp. 124–131.

[4] Eucalyptus Systems Inc. (2013, Jun.) http://www.eucalyptus.com. [Online]. Available: http://www.eucalyptus.com

[5] "OpenStack," [Online; accessed Aug-2014] "http://www.openstack.org/".

[6] "Perturbation Theory," http://en.wikipedia.org/wiki/Perturbation_theory.

[7] A. Pucher, E. Gul, C. Krintz, and R. Wolski, "Using Trustworthy Simulation to Engineer Cloud Schedulers," in Cloud Engineering (IC2E), 2015 IEEE International Conference on, March 2015.

[8] J. Gustedt, E. Jeannot, and M. Quinson, "Experimental Validation in Large-Scale Systems: a Survey of Methodologies," Parallel Processing Letters, vol. 19, no. 3, pp. 399–418, 2009.

[9] R. Wolski and J. Brevik, "Using Parametric Models to Represent Private Cloud Workloads," Computer Science Department of the University of California, Santa Barbara, Santa Barbara, CA 93106, Tech. Rep. UCSB-CS-2013-05 (to appear in IEEE Transactions on Services Computing), October 2013.