Towards a Scalable Data Center-level Evaluation Methodology

David Meisner
meisner@umich.edu

Junjie Wu
wujj@umich.edu

Thomas F. Wenisch
twenisch@umich.edu

Advanced Computer Architecture Lab
The University of Michigan

***Abstract*—As the popularity of Internet services continues to rise, the need to understand the design of the data center systems hosting these workloads becomes increasingly important. Unfortunately, research in this area has been stifled, primarily due to a lack of tools, workloads, and rigorous evaluation methodology. Traditional tools, such as architectural simulators, do not directly address data center-level issues and do not scale to simulate the thousands of machines needed for data center research.**

We introduce,* Stochastic Queuing Simulation (SQS), *our methodology for characterization and evaluation of data center systems. By leveraging techniques from stochastic modeling, queuing theory and statistical sampling, SQS uses discrete-event simulation to drive models that scale to tens of thousands of machines. Whereas detailed architectural simulations can last hours or days, SQS turnaround time is typically on the order of tens of minutes to an hour. Furthermore, computation can be distributed across cores and machines, achieving speedup using commodity clusters.

I. Introduction

Recently, there has been an explosive growth in Cloud-based Internet services, greatly influencing both software and hardware architectures. Small mobile devices connected to large data centers are becoming increasingly important, quickly overtaking traditional workstations. The design of data centers themselves has shifted from smaller collocation centers to massive Warehouse-Scale Computers (WSC), housing many thousands of servers.

The lack of scalable simulation tools has limited past WSC research to either measurement studies of existing deployments, or analysis via theoretical and statistical models. Measurement studies, though valuable, can explore only existing architectures and require access to multi-million dollar facilities. Even for the few academic and industrial research teams with access to such facilities, experimentation is typically limited to non-intrusive monitoring, since these facilities host the mission-critical operations of their owners. Analytic approaches typically require numerous simplifying assumptions and cannot capture detailed interactions among the components of a WSC. Moreover, even well-understood modeling approaches, for example queuing networks (on which our methodology is based), rapidly become analytically intractable as the size and complexity of the model grows.

This study presents our data center-level evaluation methodology, *Stochastic Queuing Simulation* (SQS), targeted specifically to investigate issues of data center design at scale. At its core, SQS is a methodology for system characterization and discrete-event simulation to enable quantitative exploration of data center-level challenges, such as performance optimization, power provisioning, power management, distributed data placement, and fault-tolerant design. SQS incorporates a number of techniques from stochastic modeling, queuing theory and statistical sampling to provide simulations that are fast enough to handle multi-thousand server complexity and provide probabilistic guarantees on its estimates.

Our methodology hinges on the observation that designers must raise the *level of abstraction* for data center-scale simulation. Rather than simulate workloads at the granularity of an instruction, memory, or disk access as in conventional simulation tools, SQS is built on the theoretical framework of queuing theory, where the fundamental unit of work is a *task* (a.k.a job). Tasks are characterized by a set of statistical properties—random variables that describe their length, resource requirements, arrival distribution, or other relevant properties—which are collected through observation of real systems, similar to [1]. SQS abstracts the data center as an interrelated network of queues and power/performance models describing the relevant behaviors of software/hardware components. The discrete event simulation uses a variety of statistical sampling techniques to provide estimates of selected output variables (e.g., 95th-percentile response time) with quantifiable measures of confidence, while enabling parallel simulation to provide strong scaling to reduce turnaround time.

SQS is not a replacement for conventional simulators; whereas existing simulation tools are still needed to refine the design for an individual server within a data center, SQS provides a framework for investigating behaviors that emerge at scale with rapid turnaround time.

II. Stochastic Queuing Simulation

We demonstrate the utility of SQS by simulating cluster-level power-capping [2] – using power management to limit the aggregate power consumption of a collection of machines. Figure 1 provides an overview of the SQS methodology, which comprises two parts: characterization and simulation.

Characterization. In the characterization step, we construct empirical models of workloads and systems that are used during the simulation step. A workload model comprises, at a minimum, task interarrival and service distributions. The workload model may also include distributions for other critical task parameters (e.g., tasks' network traffic if modeling network links). The system model modulates service rates and relates tasks to output variables of the simulation (e.g., in our

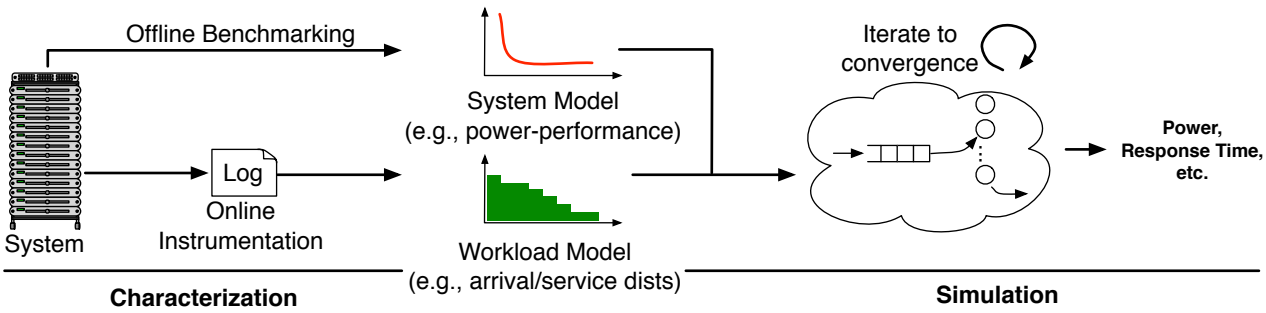

Fig. 1: Overview of the SQS methodology: A system is a) instrumented to derive workload interarrival and service time distributions and b) characterized to create a model of system behavior (e.g., power-performance settings). From these inputs, SQS simulations derive estimates for new system designs and/or configurations.

running example, it captures the power-performance curve for DVFS states).

Characterization involves both an online and offline component. We construct empirical models of workloads online, by instrumenting a live system. Typically, this process involves instrumenting a binary such that the timing of task arrivals and their duration are recorded. Later, these traces can be processed to derive the desired distributions. It is necessary to capture these workload models online, under live traffic, because interarrival processes depend greatly on the users of an internet service.

In the offline component of characterization, real systems are benchmarked to capture their modes of operation and construct system models. For our power-capping example, one would capture a server's power-performance behavior under the available DVFS settings. The model records the relative service rate and power consumption of the system as a function of frequency setting. Typically, this part of characterization must be performed offline because it would disrupt production systems.

Simulation. During simulation, SQS derives estimates for hypothetical data center configurations. For our DVFS example, various frequency transition policies for a rack of servers could be evaluated such that both latency constraints are met and rack-level power stays within a budget.

The simulation itself is a discrete-event simulation of the queuing network representing the data center. Typical events represent a high-level phenomenon such as a task entering or exiting a server, a power-performance state changing, and so on. The core functionality of the SQS discrete event simulator does not differ substantially from other tools for simulating queuing networks. SQS augments conventional queuing networks with system models, such as the power-performance model used in our example.

Workload Models. Rather than requiring an executable binary, as in a traditional simulator, SQS workloads are defined statistically by empirical interarrival and service time distributions. This approach allows workloads to be represented compactly—a typical distribution occupies less than 1 MB, whereas event traces often require multi-gigabyte files. Furthermore, in contrast to binaries, which industry is often loathe to disseminate, public dissemination of interarrival and service distributions is significantly easier, as they do not require releasing proprietary software.

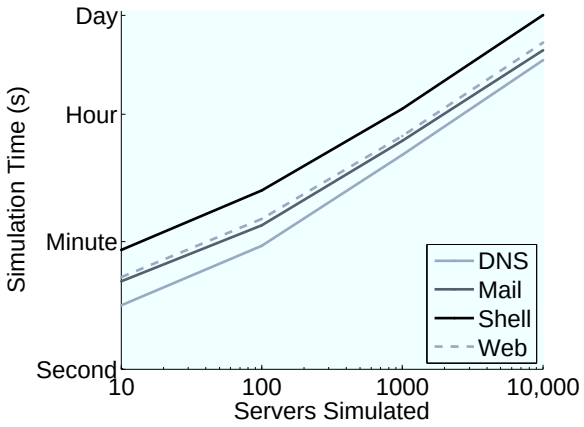

Fig. 2: Simulation Time Scaling: Simulation time required for convergence scales roughly linearly with the number of servers simulated. Scaling simulation size typically does not increase the variance of the output variables, so the required sample size does not increase significantly. Instead, the overhead of maintaining the discrete-event-simulation state is the main cause of increased runtime.

III. Results

Figure 2 illustrates the how simulation time varies with the number of simulated servers for four different workloads. Simulation of tens of servers takes only minutes. Larger cluster sizes increase simulation time in a near-linear fashion. Importantly, simulation time of appropriately large systems (1,000-10,000 clusters) is hours to a day. Such studies would be prohibitively expensive with existing architectural tools.

References

[1] L. Eeckhout, S. Nussbaum, J. Smith, and K. De, "Statistical simulation: Adding efficiency to the computer designer's toolbox," *IEEE Micro*, pp. 26–38, 2003.

[2] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," *ISCA '07: International Symposium on Computer Architecture*, 2007.