Maximizing Server Utilization while Meeting Critical SLAs via Weight-Based Collocation Management

Sergey Blagodurov[1], Daniel Gmach[2], Martin Arlitt[2], Yuan Chen[2], Chris Hyser[2], Alexandra Fedorova[1]

[1] School of Computing Science
Simon Fraser University
Vancouver, BC, Canada
{firstname_lastname}@sfu.ca

[2] Sustainable Ecosystems Research Group
Hewlett-Packard Laboratories
Palo Alto, CA, USA
{firstname.lastname}@hp.com

***Abstract*—Servers in most data centers are often underutilized due to concerns about SLA violations that may result from resource contention as server utilization increases. This low utilization means that neither the capital investment in the servers nor the power consumed is being used as effectively as it could be. In this demo, we present a novel method for managing the collocation of critical (e.g., user interactive) and non-critical (e.g., batch) workloads on virtualized multicore servers. Unlike previous cap-based solutions, our approach improves server utilization while meeting the SLAs of critical workloads by prioritizing resource access using Linux cgroups weights. We showcase our work conserving collocation method by utilizing a server to nearly 100% while keeping the performance loss of critical workloads within the specified limits.**

Keywords: collocation; SLA; virtualization; server efficiency

I. OUTLINE

The rising penetration of digital services leads to a rapidly growing demand in data center computing and increases the already substantial global power consumption of data centers. Despite the significant demand for computational resources, servers in data centers tend to be under-utilized. Low utilization means that power is not used as effectively as it could be, as servers use a disproportionate amount of power when idle, relative to when they are busy. For example, energy efficiency (work completed per unit of energy) of commodity server systems at 30% utilization can be less than half that at 100% [1][2]. Low utilization also contributes to over-provisioning of IT equipment and increased capital expenditures (CapEx).

We address the challenge of increasing server utilization while maintaining Service Level Agreements (SLAs) by consolidating critical workloads such as transactional or interactive workloads with non-critical workloads such as batch processing jobs or High Performance Computing (HPC) applications onto each server, enabling a very high utilization level (80% and above). At higher utilizations, the performance of the hosted workloads is likely to degrade due to contention for shared server resources. We prevent this by providing prioritized access to physical resources using Linux Control Groups (cgroups) *cpu.shares* [10]. We demonstrate that critical workloads have preferred access to physical resources such that they exhibit similar performance when consolidated with non-critical workloads as compared to when they are not.

The demonstrator is described in more detail in [3].

II. DEMONSTRATION OVERVIEW

A. Workload Description

Table I describes the workloads that we use for our demo runs. We use RUBiS and Wiki as representatives for critical workloads and combinations of financial, animation, simulation and scientific applications as non-critical workloads [4–8].

Both critical applications are multi-tier Web applications. Wiki represents a fairly large and computational intensive application that triggers multiple queries to the database or Memcached server to create its dynamic Web pages. Wiki response times in our setup are typically on the order of 500 milliseconds (ms) and above. In contrast, RUBiS is a much faster application with response times on the order of tens of milliseconds.

We divide our non-critical workloads into two groups. In Group A we use a set of Swaptions, Facesim, and FDS. These are popular applications representing three typical types of the data center batch loads: finance, animation and simulation. Swaptions and Facesim are multi-threaded, mostly CPU intensive, with Facesim and FDS being partially I/O bound. They have no network communication. Group B is comprised of LU, BT and CG, which are CPU bound, network intensive MPI jobs. They represent a typical HPC workload.

B. Demo setup description

The demo setup is shown on Figure 1 and is comprised of the following modules.

One laptop will dynamically display the performance of the collocated workloads and the CPU utilization of our remote testbed (see below). The workload performance will be represented quantitatively by the measured 99th percentile of the request response time as well as qualitatively, by sharing a functioning web interface of the critical applications. The laptop will be provided by presenters.

The laptop will be connected to the Palo Alto based datacenter through a remote gateway. We will need an internet access to be able to do that. Our remote datacenter testbed consists of five HP ProLiant BL465c G7 servers. Each is equipped with two 12-core AMD Opteron 6176 (2.3 GHz) CPUs. Each server has 5 MB shared cache and 64 GB of main memory distributed across 4 NUMA memory nodes. All servers are connected via a 10 Gb/s Ethernet network and have access to 4 TB of NFS storage.

We consider an environment where applications are hosted within a common pool of virtualized servers. Each application

978-3-901882-50-0

Table I. Critical and Non-critical Workloads used in this Study.

Class	Application	Description
Critical	RUBiS	RUBiS is modeled after eBay.com and implements the core functionality of an auction site: selling, browsing and bidding. It is widely used as a benchmark in research [4]. We used a 3-tier RUBiS setup in our experiments where client requests are first served by an Apache Web server. It answers static HTML pages directly and uses a JBoss application server for dynamic requests. The data for the dynamic pages is stored in a MySQL database.
	Wiki	Wiki is a realistic Web hosting benchmark based on WikiBench [7] and MediaWiki, which is the application used to host wikipedia.org. It uses real Wikipedia database dumps and generates traffic using publicly available traces. Our Wiki setup consists of a workload generator and three tiers: an Apache Web and application server (PHP-based), a MySQL database server and a Memcached server.
Non-critical	Swaptions	Swaptions is a financial analysis benchmark suite from Intel that mimics an RMS (recognition, mining and synthesis) workload that uses the Heath-Jarrow-Morton (HJM) framework to price a portfolio of Swaptions [5]. Swaptions is a multi-threaded application.
	Facesim	Facesim is animation software that takes a model of a human face and a time sequence of muscle activations and computes a visually realistic animation of the modeled face [5]. Facesim is a multi-threaded application.
	Fire Dynamics Simulator	Fire Dynamics Simulator (FDS) is a fire simulator that computes a computational fluid dynamics model of fire-driven fluid flow [6]. FDS is a single-threaded application.
	LU, BT, CG	LU (Lower-Upper Gauss-Seidel solver), BT (Block Tri-diagonal solver) and CG (Conjugate Gradient, irregular memory access and communication) are popular scientific programs written in MPI (Message Passing Interface) programming model [8].

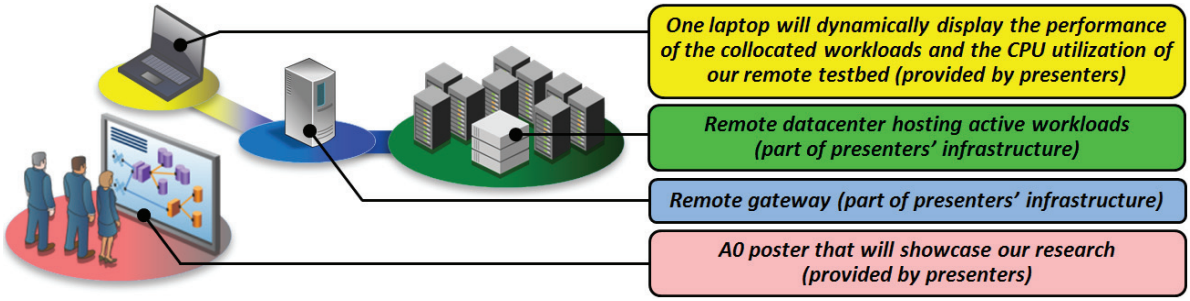

Figure 1. Diagram of the demo setup.

or application tier runs in a virtual machine (VM). The resources of a physical server, including CPU, memory, disk and network I/O bandwidth, are shared by the hosted VMs. In our prototype implementation, we use KVM as the virtualization platform running on Ubuntu 12.04.

We prioritize the access to CPU cycles for the collocated workloads by adjusting the parameter *cpu.shares*, which is a configuration parameter available through Linux Control Groups (*cgroups*). It specifies how VMs are given processor time by the scheduler. A share value twice as high for a process compared to another denotes that it has access to twice as many CPU cycles as the other process. In the literature, this is often referred to as work conserving mode [3].

The demo will additionally include a poster provided by presenters that will showcase our research.

III. DEMO SCENARIOS

Besides explaining our research to the guests, we intend to demonstrate two scenarios with our testbed. In both scenarios, we collocate critical and non-critical workloads from Table I resulting in server utilizations of 80% and above.

In the first scenario, we leave the workloads unmanaged (the default case). We show that the critical performance is unacceptable in this case. In the second scenario, we provide critical workloads with the preferred access to physical resources. We demonstrate that cgroups weights can be used to increase overall server utilization while maintaining the performance of critical workloads. We show that, although the 95th percentile response times of critical workloads increases between 41% and 98% when comparing a solo run and a collocated run with up to 80% server utilization, it is important to note that based on HCI research [9] we anticipate minimal consequences for end users. With our approach we manage to improve server utilization while maintaining the response times of critical applications in similar HCI "categories" [9] (e.g., "crisp" for the RUBiS application), such that typical users will not notice the difference in performance.

REFERENCES

[1] U. Hoelzle and L. A. Barroso. The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines. 2009.

[2] http://cdn.globalfoundationservices.com/documents/MSFTTop10BusinessPracticesforESDataCentersAug12.pdf .

[3] S. Blagodurov, D. Gmach, M. Arlitt, Y. Chen, C. Hyser, A. Fedorova. Maximizing Server Utilization while Meeting Critical SLAs via Weight-Based Collocation Management, IM 2013.

[4] S. Blagodurov and M. Arlitt. Improving the efficiency of information collection and analysis in widely-used IT applications. ICPE 2011.

[5] C. Bienia, S. Kumar, J. P. Singh, and K. Li. The PARSEC benchmark suite: Characterization and architectural implications. Technical report, Princeton University, 2008.

[6] R. Curry, C. Kiddle, N. Markatchev, R. Simmonds, T. Tan, M. Arlitt, and B. Walker. Facebook meets the virtualized enterprise. In 2008 12th International IEEE EDOCC.

[7] http://www.wikibench.eu/

[8] http://www.nas.nasa.gov/publications/npb.html

[9] N. Tolia, D. G. Andersen, and M. Satyanarayanan. Quantifying interactive user experience on thin clients. Computer, 39:46–52, 2006.

[10] http://www.kernel.org/doc/Documentation/cgroups/