# ACTS: Adaptive Control Techniques for Virtualized Servers

Xiaoyun Zhu  Zhikui Wang  Sharad Singhal
Hewlett-Packard Laboratories
Palo Alto, CA 94304, USA
{xiaoyun.zhu, zhikui.wang, sharad.singhal}@hp.com

*Abstract*— **This poster presents the ACTS (Automated Control and Tuning of Systems) project at HP Labs that aims to develop adaptive control techniques to simplify service level management of enterprise applications while increasing resource utilization of virtualized servers.**

## I. Introduction

The need for increasing return on IT investment has in the past few years driven an emerging interest on server virtualization that allows common server resources to be shared across multiple resource containers used by different applications or organizations. For example, HP offers the Virtual Server Environment (VSE) that aims to help enterprise customers optimize server resource utilization based on business priorities. Products in the VSE portfolio include the HP-UX Process Resource Manager (PRM), the Virtual Partitions (vPars), and the Integrity Virtual Machines for server partitioning with varying degrees of isolation and flexibility, as well as the HP-UX Workload Manager (WLM) for automatic control of CPU resource based on prioritized service level objectives (SLOs). Similar products from other IT vendors include the IBM Enterprise Workload Manager, VMware, Xen, and Microsoft Virtual Server. Server consolidation practices exploiting such technologies have resulted in significant savings not only in hardware, but also in software licenses, physical cost (power, cooling, space), and operational cost. The SASU (Shared Application Service Utility) project in HP IT is a good example where multiple BEA WebLogic Application Servers are hosted on a common HP-UX server using PRM/WLM partitions and offered as a service to business-critical applications within the company.

As a result, there is an urgent need for dynamic resource allocation techniques for adjusting capacities of resource containers in real time, for three reasons. First, enterprise applications have been observed to have resource demands that vary over time. While long-term trends or repeatable patterns in these demands can be managed by offline capacity planning, dynamic resource allocation in shorter time-scales is required to ensure that applications' SLOs are met in face of unanticipated changes in business priorities, system conditions, or user demands. Second, in contrast to static provisioning, dynamic allocation takes advantage of statistical multiplexing of resource demands between co-hosted applications, thereby resulting in higher resource utilization overall. Third, APIs (i.e., actuators) for dynamic allocation of system resources such as CPU, memory, disk I/O, and network I/O are readily available in either all or some of the formerly mentioned products. In response to this need, we have created ACTS: Automated Control and Tuning of Systems, a project that aims to develop adaptive control techniques for increasing resource utilization of virtualized servers and simplifying service level management for the hosted applications.

## II. Results

The ACTS project uses formal control theory as the foundation for developing feedback control algorithms and techniques in order to achieve more predictable performance on resource utilization and application service levels. Consider a virtualized server that has m resource containers, each hosting an application. The resource controller interacts with each container i through two modules, Ai and Si, where Si is the sensor that periodically measures the resource utilization of container i or the service level (e.g., response time or throughput) for application i, and Ai is the actuator that dynamically allocates a portion of the shared resource to container i according to the output of the resource controller. At the beginning of every control interval, the resource controller computes the needed resource allocation to container i based on its measured utilization or the service level from the past interval as well as the desired values for these metrics, and passes it to Ai for actuation.

We have developed a suite of adaptive control algorithms to dynamically allocate CPU resource to a container, for regulating either the mean response time as a service metric (IM2005), or the utilization level of a container (DSOM2005), or a combination of both using nested control loops (ACC2006), where controller parameters vary automatically as the system operating condition changes. We also developed predictive control algorithms exploiting periodicity and time correlation in an application's workload (NOMS2006). Although all these controllers are for CPU allocation to individual resource containers, the algorithms should be extensible to managing multiple containers at the same time.

We have implemented the above control algorithms on two testbeds, one using HP-UX PRM to create resource partitions on an HP 9000 server, the other using Xen to create virtual machines on a DL360 Linux server. In both cases, an Aapche Web Server is used as the application hosted inside a container. We have also built a demonstrator for comparing results of different allocation schemes and different algorithms. Compared to the current practice of uisng static allocation, our dynamic allocation scheme can increase the utilization of allocated resource while delivering same or better service levels. Compared to other feedback controllers with fixed parameters, our adaptive controller was able to offer more stable and predictable service levels and reduce the overhead of manual tuning by the operator.

We can run the demo on a laptop next to the poster at the workshop if possible.