Optimizing Energy Efficiency and Quality of Service in Large Scale Web Server Environments

Simon Kiertscher
University of Potsdam
Potsdam, Germany
kiertscher@cs.uni-potsdam.de

Bettina Schnor
University of Potsdam
Potsdam, Germany
schnor@cs.uni-potsadm.de

ABSTRACT

Energy-aware resource-management strategies for web-server clusters use on/off strategies which are based on thresholds. The basic idea is to turn off currently unused nodes, and on again when the load increases. The biggest challenge for these strategies is to provide enough compute power in an unexpected peak load situation. We investigate this challenge using trace-driven simulation for different scenarios like using Suspend-to-RAM capable nodes, over-provisioning, load forecasting, and anti-flapping strategies. We evaluate the results using well known metrics like quality of service and energy consumption. Within different scenarios we are able to achieve a saving of 28.08 % of energy while retaining a quality of service of 96.99 % even in a peak scenario.

1. INTRODUCTION

Web server clusters are an important infrastructure for many Internet applications from eCommerce to Cloud Computing and contribute to the immense energy consumption of data centers. In 2010, the U.S. Environmental Protection Agency (U.S. EPA) estimated 3 % of all electricity consumed in the U.S. to be used in running data centers [3]. Hence, an energy efficient resource management has become a significant issue.

We developed the energy saving daemon *CHERUB* [2] which applies an on/off-strategy for unused resources to react dynamically on load changes. To evaluate on/off-strategies in a bigger setup, the only feasible approach is simulation. The simulator *ClusterSim* is able to simulate a large cluster running the Apache web server. The validation of the simulator was presented in [1]. *ClusterSim* is able to mimic a web-server cluster of arbitrary size and interacts with the original *CHERUB* daemon, i.e. not a simulation of *CHERUB*, but the real daemon. *ClusterSim* models a webserver cluster running Apache with the MPM-prefork module on the back-ends. The architecture of *ClusterSim* and the coupling with *CHERUB* is illustrated in Figure 1. The blue components are part of *ClusterSim*. *CHERUB* is coupled via a separate *State Server* which manipulates the virtual cluster according to the commands given by *CHERUB*, and gives *CHERUB* all the information about the current load and the status of the simulated nodes. The trace and the request library must be provided by the user. The trace contains the workload. For each request in the workload, the request library contains its duration.

In this paper, we investigate the scaling of *CHERUB*'s on/off-strategy for a cluster of 100 virtual nodes using *ClusterSim*.

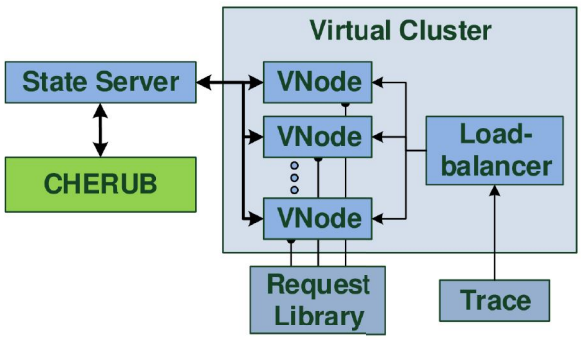

Figure 1: Coupling *ClusterSim* with *CHERUB* [1].

Further, we investigate the influence of the node's boot duration and consider also servers which utilize the *Suspend-to-RAM* (STR) technology for a nearly instant availability after shutdown. STR is part of the *Advanced Configuration and Power Interface* (ACPI) specification [1] which defines various power related features. With STR, the current state of the operating system and the running applications is stored in memory, and hold there after shutdown.

The rest of the paper is structured as follows. Section 2 describes the metrics, parameters and factors as well as the workload of our experiments. Section 3 presents and discusses the results. Section 4 concludes the paper.

2. DESCRIPTION OF EXPERIMENTS

We use *ClusterSim* 1.07.00 to simulate a virtual cluster composed of 100 homogenous nodes and *CHERUB* 1.3.11 to manage this cluster.

2.1 Used Metrics

We are interested in the following five metrics.

1. Quality of Service (QoS) in percentage.
2. Request duration (RD) in milliseconds.
3. Energy consumed (EC) in Wh / Energy saved (ES) in percentage.
4. Number of physical state changes (PSCs). One PSC is defined as the process to either turn on or turn off a node.

[1] `http://www.acpi.info/spec50a.htm`
Accessed February 2016