Sustainable Computing: Informatics and Systems 4 (2014) 205–214

Contents lists available at ScienceDirect

Sustainable Computing: Informatics and Systems

journal homepage: www.elsevier.com/locate/suscom

A combined frequency scaling and application elasticity approach for energy-efficient cloud computing

S.K. Tesfatsion*, E. Wadbro, J. Tordsson

Department of Computing Science, Umeå University, SE-90187 Umeå, Sweden

ARTICLE INFO

Article history:
Received 28 February 2014
Received in revised form 4 July 2014
Accepted 11 August 2014

Keywords:
Cloud computing
Energy-efficiency
Quality-of-service
Virtualization
Frequency scaling
Application elasticity

ABSTRACT

Energy management has become increasingly necessary in large-scale cloud data centers to address high operational costs and carbon footprints to the environment. In this work, we combine three management techniques that can be used to control cloud data centers in an energy-efficient manner: changing the number of virtual machines, the number of cores, and scaling the CPU frequencies. We present a feedback controller that determines an optimal configuration to minimize energy consumption while meeting performance objectives. The controller can be configured to accomplish these goals in a stable manner, without causing large oscillations in the resource allocations. To meet the needs of individual applications under different workload conditions, the controller parameters are automatically adjusted at runtime based on a system model that is learned online. The potential of the proposed approach is evaluated in a video encoding scenario. The results show that our combined approach achieves up to 34% energy savings compared to the constituent approaches—core change, virtual machine change, and CPU frequency change policies, while meeting the performance target.

1. Introduction

Cloud computing delivers configurable computing resources on-demand to customers over a network in a self-service fashion, independent of device and location [1]. The resources required to provide the necessary quality-of-service (QoS) levels are virtualized, shared, rapidly provisioned, and released with minimal service provider interaction. Modern data centers use virtualization to provide application portability and facilitate dynamic sharing of physical resources while retaining application isolation. The virtualization technologies enable rapid provisioning of virtual machines (VMs) and thus allow cloud services to scale up and down resources allocated to them on-demand. This elasticity can be achieved using *horizontal* elasticity, where the number of VMs is changed during a service's operation and *vertical* elasticity, where hardware allocation of a running VM, typically in terms of CPU and RAM, are changed dynamically.

Today, the number and size of data centers are growing fast. According to a report [2], there are about 500 000 data centers worldwide and one report [3] estimates that Google runs more than a million servers in total. At present, reducing power consumption is a major issue in the design and operation of these large-scale data centers. A recent survey reports that worldwide, data centers use about 30 billion watts of electricity [4]. The effects of high power consumption manifest in a high operational cost in data centers. Another study presents that the cost of energy has steadily risen to capture 25% of total operating costs, and is among the largest components of the overall cost [5]. In addition, the power consumption in large-scale data centers also raises many other serious issues including excessive carbon dioxide emission and system reliability, e.g., running a single high performance 300-W server during a year can emit as much as 1300 kg CO_2 [6].

Autonomous resource provisioning in the cloud has been widely studied to guaranty system-wide performance, that is, to optimize data center resource management for pure performance [7,8]. *Green cloud computing* is envisioned to achieve not only efficient processing and utilization of a computing infrastructure, but also to minimize energy consumption [9]. This is essential for guaranteeing that the future growth of cloud computing is sustainable. Lowering the energy consumption may result in performance loss and it is thus important to be able to guarantee application QoS while minimizing the energy consumption. To achieve this goal, a well designed trade-off between energy savings and system performance is necessary.

In the research literature, a large body of works applies dynamic voltage and frequency scaling (DVFS) and vary-on/vary-off (VOVO) power management mechanisms. DVFS changes the operating

* Corresponding author. Tel.: +46 722184762.
E-mail addresses: selome@cs.umu.se (S.K. Tesfatsion), eddiew@cs.umu.se (E. Wadbro), tordsson@cs.umu.se (J. Tordsson).

http://dx.doi.org/10.1016/j.suscom.2014.08.007
2210-5379/

frequency and voltage of a given computing resource. VOVO turns servers on and off to adjust the number of active servers according to the workload. Studies show that these techniques can reduce power consumption considerably [10–12,23,47]. However, there is limited work on the use of virtualization capabilities to dynamically scale resources of a service with the main objective of saving energy while meeting the performance target. Moreover, current elasticity implementations offered by IaaS providers, e.g., Amazon EC2 [14], which uses VMs as the smallest scaling unit, may be too coarse-grained and thus cause unnecessary over-provisioning and waste of power. One approach to address the problem of resource provisioning is to provide a fine-grained resource allocation by adapting the VM capacities to the requirements of the application. It is also important to consider the cost of changing resources in terms of performance penalty and system reliability. Therefore, reconfiguration cost should also be considered.

Traditionally, adaptive power management solutions mainly rely on heuristics. Recently, however, feedback and control theory have been successfully applied to power management [15,16,18]. For example, work by Lefurgy et al. [19] shows that control-theoretic power management outperforms a commonly used heuristic solution by having a more accurate power control and better application performance.

In this paper, we present a fine-grained scaling solution to the dynamic resource-provisioning problem with the goal of minimizing energy consumption while fulfilling performance objectives. In contrast to existing works, we combine virtualization capabilities—horizontal and vertical scaling and a hardware technique—scaling the frequency of physical cores and apply control-theoretic approach. In summary, the contributions of this work are:

- An evaluation of performance and power impact of the different management capabilities available in modern data center servers and software.
- Design of an online system model to dynamically determine the relationship between performance and power for the various configurations of the system. Our adaptive model captures variations in system dynamics due to differences in operating regimes, workload condition, and application types.
- An architectural framework for the energy-efficient management of cloud computing environments. Our framework integrates software techniques—horizontal and vertical elasticity and a hardware technique—CPU frequency scaling.
- Design of feedback controller that determines a configuration to minimize energy consumption while meeting performance targets. The controller can be configured to handle trade-offs among energy minimization, guaranteeing application performance, and avoiding oscillations in resource allocations.
- An evaluation of the proposed framework in a video encoding scenario. Our combined approach achieves the lowest energy consumption among the compared three approaches while meeting performance targets.

The rest of the paper is organized as follows. In Section 2, we present the design of the system model. Section 3 describes the architecture of our proposed system. This is followed by a detailed description of the design of the optimal controller in Section 4. In Section 5, we present our experimental setup and the evaluation results. Section 6 surveys related work. Conclusion and future directions are covered in Section 7.

2. System model

In control theory, system models play an essential role in analysis and design of feedback systems [35]. They characterize the relationship between the inputs—control knobs and the outputs of the system—metrics being controlled. In this section, we provide a description of dimensions or knobs that can be used to manage systems in an energy-efficient manner. Then we describe a set of system modeling experiments we performed to analyze the impact of each management action (input) on performance and power consumption (outputs) and design a system model for the dynamic behavior of the application under various configurations.

2.1. Power management dimensions

- *Horizontal scaling*. This technique exploits the hypervisor's ability to add or remove VMs for a running service. The power impact of a VM can be considered in terms of the dynamic power used by components like CPU, memory, and disk. Thus, VM power consumption greatly depends on the extent to which these resources are utilized and also differs from one application type to another.
- *Vertical scaling*. This technique exploits the hypervisor's ability to change the size of a VM in terms of resources like cores and memory. We select core or CPU as a resource for management as it dominates the overall power usage profile of a server. The CPU can draw up to 58% of the relative non-idle dynamic power usage by a data center server [36]. CPU-bound applications, in general, benefit more from this type of scaling. Some research [16,37] show energy consumption savings by modifying the hypervisor's scheduling attributes to change guest's maximum time slice on a core. In contrast, in this work we instead change the number of cores of a VM without modifying the hypervisor scheduler.
- *Hard power scaling*. DVFS has been applied within clusters and supercomputers to reduce power consumption and achieve high reliability and availability [17,34,38,39,18]. In this work, we do dynamic frequency scaling (DFS)—changing the frequencies of the cores based on demand. We also consider turning individual cores on and off in the power scaling decision.

2.2. System model experimentation

After identifying the above dimensions, we performed a set of experiments to understand the relationship between these and their impact on application's performance and power usage. Fig. 1 illustrates an input–output representation of the system we are controlling. The inputs to the system are server CPU frequency, cores, and number of VMs. Two outputs are considered, power usage (W) and performance (throughput, although other performance metrics could also be used). The experimentation was performed to see how performance and power vary with changes in CPU frequency, number of cores, and number of VMs as well as to determine a model for the system behavior.

2.3. Experimental setup

The experiments are performed on a ProLiant DL165 G7 machine equipped with 32 AMD Opteron(TM) Processors (2 sockets, 4 nodes, and 32 cores in total, no hyper-threading) and 56 GB of physical memory. The server runs Ubuntu 12.04.2 with Linux kernel 3.2.0.

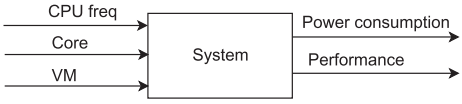

Fig. 1. An input–output model for our considered system.