ARTICLE IN PRESS

Future Generation Computer Systems ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Energy-efficient virtualized clusters

Mihai Carabaş, Pantelimon George Popescu *

Computer Science and Engineering Department, Faculty of Automatic Control and Computers, University Politehnica of Bucharest, Splaiul Independenţei 313, 060042, Bucharest (6), Romania

HIGHLIGHTS

- Mathematical proof of a virtual–physical machine mapping.
- New upper and lower bounds for the power consumption.
- A new test scenario which takes particular mappings.
- Validate applicability of the new theoretical results in practice.

ARTICLE INFO

Article history:
Received 24 July 2015
Received in revised form
12 October 2015
Accepted 26 October 2015
Available online xxxx

To Nicolae Ţăpuş on his 65th birthday

Keywords:
CPU load
Power reduction
Scheduling
Virtualization
Jensen inequality
Bounds

ABSTRACT

In this paper we provide the state of the art for the virtualization techniques and means to reduce power consumption using it. Virtualization allows us to answer all the requirements with many-core servers and thus eliminate the *one size does not fit all* issue. The resulting pool of resources is beneficial from an economic as well as environmental point of view. It brings benefits of scale to all logistic elements of the problem: power supply, cooling, floor space. When talking about virtualization and power consumption, one important aspect to be taken into account is data center's heterogeneity from the hardware architecture point of view (e.g., X86, PowerPC). Mapping virtualized operating systems on hardware nodes in order to minimize power consumption is still an open issue that will be addressed throughout this paper: given a number of physical machines, we try to map on them the available virtual machines (called virtual machine assignment) in order to have an efficient system when relating to power consumption. We expose new general bounds for the power consumption of a virtual machine assignment based on Jensen inequality. The lower bound has been previously obtained and used into literature, so here we only rediscover it in a simplified and more clear manner. The upper bound is new and general. Furthermore we practically evaluate some discrete cases and we proposed some graphics with the power consumption and its bounds for some particular real cases.

1. Introduction

Operating system virtualization is a debated subject in modern Computer Science and Engineering, even if their origins are very old from the '60s [1]. Over the last years, the philosophy of hardware development has been modified from high frequency processors (CPUs) to multiple core processors [2]. This trend enforced the acceleration of the virtualization development, in 1999 being released the first product from VMware [3]. One of the advantages in having multiple operating systems running on top of a single hardware than running multiple processes in only one instance is related to the isolation degree. For example, it is a much secure environment having an operating system for each project, than having only one instance for all projects. Virtualization helps running multiple instances of an operating system on the same hardware. Another advantage of virtualization is related to power-efficient data centers. Through virtualization's elasticity one can easily move services (operating system instances) from one hardware to another in order to minimize the power consumption (e.g., group all services on a hardware partition of the data center and shutdown the remaining machines).

Modern microprocessors have become so powerful that assigning a server to a single service or application does not make economic sense. Nowadays, even the smallest microprocessors tend to be multi-core ones [4] and will continue to evolve to act as multiprocessors or even clusters-on-a-chip. Not surprisingly, system administrator has migrated their services from the old servers

* Corresponding author.
E-mail addresses: mihai.carabas@cs.pub.ro (M. Carabaş), pgpopescu@yahoo.com (P.G. Popescu).

http://dx.doi.org/10.1016/j.future.2015.10.018

(many times single-core ones) into a single unit with multiple cores and a more sophisticated architecture, providing a superior performance than the old grid architecture. The virtues of physical isolation provided by downsizing were nevertheless preserved by using virtualization techniques. At the expense of a minimal overhead, hardware assistance and hypervisor-based system software ensure that the malfunction of one application, or the demise of a service, would not propagate to the whole system.

Nowadays, virtualization is being present, at a large scale [5], in servers, desktop computers and even in embedded devices. All these systems have a common requirement: reducing energy consumption and maintain productivity, where productivity is transposed with a different meaning in each category:

- servers—do the same number of jobs in a given time; to reduce energy one will shut-off the standby servers;
- desktop computers—do not affect the user experience even with demanding applications; to reduce energy one will use the auxiliary processors (e.g., GPU) only when needed;
- embedded devices—do not affect the user experience and increase battery lifetime; to reduce energy consumption one will shut-off unused cores or scale-down CPU frequency.

The majority of the mechanisms for reducing power consumption presented above can be addressed in an easier manner using virtualization techniques.

When talking about servers, the data centers have become a large power consumer [6], in many cases one can compare the consumption level on a per-day basis with the one of a city [7]. Along the last years there have been employed multiple techniques in order to minimize the power consumption: local scheduling policies on each server or cluster scheduling algorithms in order to maximize the CPU usage on a group of servers and shut down the others. Basically create virtual machines for each needed service and use the concept of migration to move it on any server in the cluster. This technique is also called *server consolidation* with a mathematical approach given in [8]. This second approach raises the problem of maintenance: how do you remotely shut down servers and boot them on demand? They need to have a separate management console for this and even then, the intervention of the administrator is mandatory. What if we can create a mapping set of a pool of virtual machines using all the physical servers that minimizes the power consumption without being necessary to shutdown any servers?

The main contributions of this paper are a mathematical proof of a virtual–physical machine mapping that is trying to minimize the power consumption: for a given number of virtual machines that must run on top of the physical ones we propose a distribution of the virtual machines which would minimize the power consumption. To verify the sustainability of the proof we created a test scenario which takes particular mappings and compare their power consumption to the discovered lower and upper bounds. Thus we validate the fact that our proof is applicable in practice.

The paper is structured as follows. Section 2 summarizes the background elements for the virtualization mechanism and the power consumption problem. Section 3 presents the related work regarding the power efficiency using virtualization techniques. Section 4 shows new general bounds for power consumption and Section 5 presents some practical virtual machine assignments for which we have computed the power consumption and its bounds.

2. Background

This section presents a brief background about virtualization in operating systems and power consumption issues in commercial platforms.

2.1. Virtualization

Virtualization means abstracting an implementation of an object through different software techniques in order to run multiple instances of that object in the same environment, preferably with minimal overhead.

Operating system virtualization refers to the fact that multiple instances of operating systems can run simultaneously on the same physical device (hardware). Based on the changes needed to be operated on the guest operating system, virtualization can be classified into two main categories, Full-virtualization and Paravirtualization.

In each type of virtualization there is a need for an entity to moderate the access to hardware. This is generally called a **hypervisor**.

2.1.1. Full-virtualization

The main **advantage** of *full-virtualization* is that the operating systems are running directly over the hardware and there is no need for modifying it. The **disadvantage** in this situation is given by the fact that you need to have hardware assisted virtualization support in the CPU. Another **disadvantage** is that hardware offer support for virtualization only for CPU privilege levels and for the Memory Management Unit (MMU), not for any other devices. One has to emulate the peripherals which bring a cost in performance.

2.1.2. Paravirtualization

In contrast with **full-virtualization**, when talking about *paravirtualization*, the operating system needs to be modified in order to use the interface provided by the hypervisor, instead of using hardware interface. This fact is providing a clear **disadvantage** (e.g. you cannot run a proprietary OS like Windows as a virtual machines). Using *paravirtualization* is necessary when the underlying hardware platform does not support extensions for full-virtualization. A clear **advantage** for paravirtualization is that it can provide paravirtualized peripherals which are very performant compared with the emulated ones.

In general a mixed of *full-virtualization* and *paravirtualization* is used: the first one is used to virtualize the CPU privilege levels and MMU and the second one is used to virtualize the peripherals.

In practice, it exists one more type of virtualization called *OS-level* or *container-based*. It is not included in the classification list because it is more of a *sandboxing* technique [9], being built with a single kernel which is common to all instances running at a given time. OS-level technique has the smallest overhead, but does not provide sufficient isolation between different operating systems. Another drawback is the fact that one cannot run multiple type of operating systems, being the same kernel at the base.

Virtualization is a widely spreaded technique in servers environment, as all the hosting providers [10] are selling virtual machines as dedicated servers. There exist various implementations, like KVM (Kernel-based Virtual Machine) [11], HyperV from Microsoft, XEN [12] which are working in a production environment. Not the same thing one can say about embedded devices like phones or automotive microchips, which is a new domain of research in virtualization, not being present at a large scale nowadays.

As stated in [13], the virtualization was extended in runtime systems to ensure the following prerequisites:

- support to virtualize the set of processors through the use of multi-threading and dynamic task migration;
- support for memory system virtualization, including object caching and migration.

Virtualization brings a lot of flexibility in system and data center administration. Features that fulfill the flexibility are strongly

Please cite this article in press as: M. Carabaş, P.G. Popescu, Energy-efficient virtualized clusters, Future Generation Computer Systems (2015), http://dx.doi.org/10.1016/j.future.2015.10.018