Advanced Cloud Resource Management For Performance and Power Saving

Alexandre Kandalintsev
University of Trento
Trento, Italy
alexandre.kandalintsev@disi.unitn.it

ABSTRACT

In the last few years cloud computing has emerged as a popular mean addressing scalability problem of distributed large-scale applications. The ever-growing demand for such technology brought up the problem of efficient power management and resource allocation in clusters. Several software and hardware technologies were developed to address this limitation. Software methods have shown to be quite efficient and can be applied to a whole cluster of hosts.

However, software methods do not have control over low-level hardware circuit modules. Built-in hardware methods have very fine-grained control, but their impact is limited to a specific microchip unit.

In this study we seek to address this problem by developing algorithms that improve interoperability and combine the benefits of both software and hardware approaches delivering significant energy savings.

Keywords

cloud, cluster resource management, crm, power, power-saving

1. INTRODUCTION

The development of the modern Internet is inseparably coupled with the development of cloud technologies. Companies of the time like amazon, dropbox, GoGrid, Joyent and many others build their business on top of the clouds.

Clouds are not only turned the way applications are developed and deployed. They emerged the new application paradigms. Smashing the boundaries between local and remote applications, they significantly raised the quality and usability of the last.

The most noticeable features of cloud applications are device and location independence, access device offload, data vitality and extended battery life.

From customers' point of view clouds provide an efficient way to manage resource ("pay as you go") and enhanced security through centralized updates.

This is how cloud computing changed the world. Yet, this technology is still on the early stage of the development and its full potential is still covered in the clouds.

2. RESOURCE MANAGEMENT

Technically speaking, a cloud is a portion of cluster resources capable of growing and shrinking (auto-scale) to accommodate the load changes. Cloud resources are controlled on the three *independ* levels: cluster level, node level and hardware level.

The cluster level of power management is represented by cluster resource manager, a software complex that manages resources and tasks in a cluster in order to maintain its efficiency. A CRM is responsible for creation and deletion of clouds.

The node-level power management is done by an operating system(OS). An OS controls the high-level state of equipment. For instance, to save energy operating system can put a processor(CPU) into the sleep state or spin-down disks.

Modern CPUs consist of many modules, which may not be permanently involved in an operation. Therefore, unused modules can be switched off [2]. This is done by a special circuit responsible for internal power management of the CPU. All management is done on the hardware level without involving an operating system. This circuit forms the hardware-level power management.

3. PROBLEM STATEMENT

A CRM makes decisions based on resource consumption [3]. The basic rule is to provide enough resources for all tasks to avoid starvation. It does not take into account *how* resources are used. This means that different tasks with different requirements can be put together. In certain cases it may be a case of contradiction.

Let us assume the following scenario. A crm manages two clouds: cloud database and computational cloud. Resources for these two clouds are allocated from the same resource pool. Traditional CRMs do not know about fine distinctions between clouds; a program doing numerical computations can be placed together with a database. Such dissimilar programs have different execution algorithms. In this situation the CPU can try to change power saving strategy on each task switching. This will lead to frequent toggling of circuits eliminating all positive effect of hardware power management [2]

This situation is shown on Figure 1. The color of programs represents types of CPU activity. Program "task 3" is running together with "task 1", although it's better to couple it with "task 2".

Moreover, an operating system can constantly move tasks to balance CPU load [1]. This also may reduce a positive effect of CPUs caches and slow down program execution [4].

Obvious solution would be a mechanism providing an operating system with comprehensive information about CPU internal state. This would allow us to schedule the execu-

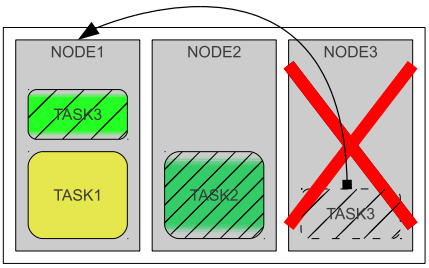

Figure 1: Cluster consolidation algorithm.

tion tasks in a more power-efficient way. Furthermore, this information can be exported to a CRM for better load balancing. Using this information as a feedback, a CRM could initiate the migration of tasks to improve overall power consumption.

However, it is not possible with proprietary solutions. Modern CPUs lack methods of fine-grained control over built-in features. Detailed description of a CPU implementation can expose commercial secrets. Thus, we do not have methods of fine-grained control over built-in features.

4. HINTS FOR A SOLUTION

To avoid contradictions between different levels of power management we need to establish connectivity from lower levels to upper. It can be done through a collection of valuable information concerning our tasks that is to be passed to the upper levels. Though we do not have this detailed information we can reconstruct it.

We propose two methods of reconstruction of missing information. The first proposal is based on our knowledge of internal structure of CPUs. The second proposal uses hidden Markov model(HMM). These methods have individual advantages and disadvantages; they can be used together to "join efforts".

4.1 Direct task Classification

Each CPU instruction is run by a specific execution block. Using debugging tools we can dump and inspect instruction flows from all processes. Then we label all processes with special marks describing CPU usage. These data are to be parsed by a special analyzing tool that can classify processes by used CPU modules[1]. The output of this program should be a classification information matching the processes and CPU resources need to execute them. Using this information scheduler can schedule processes with similar characteristics to the same CPU of the same server.

In case of small number of processes changes to group processes by this criteria is not very hight. The resulting arrangement may show little to no effect because processes may be rather different. However, obtained statistical data can be propagated to the CRM. Using these values the CRM can move tasks in a cloud to combine them in a more energy-efficient way.

Direct task classification is very simple but it has the following disadvantages:

- Our knowledge concerning CPU structure is very limited. This limits the accuracy of methods based on inspection of instructions. For instance, out of order execution makes execution flow very unpredictable.
- Processes can change (e.g. on external conditions) execution profile in the future due to their dynamic nature.
- We do not take into account the order of instructions. CPU may detect that specific block remains unused for some short period of time and switch it off.

Despite these shortcomings this method can still be useful as it is simple, relatively fast and, to the best of our knowledge, should work perfectly in simple cases.

4.2 Machine-learning Classification

The issues of the previous method are addressed by the following proposal that uses a machine-learning approach.

We propose to use Hidden Markov Model(HMM) to reconstruct missing information. In this method CPU instructions and consumed power are the observable variables, and CPU blocks are hidden states that we would like to estimate.

Machine-learning approach requires initial training, additional equipment (power meter connected to the CPUs) and more resources for computation. However, it gives more precision and does not depend on a specific CPU architecture.

5. RESEARCH PLAN

Our research line consists of three major parts. The first part is dedicated to verify the main concepts and principles used in this research. We plan to assemble testbed consisting of a PC and power meeter for this step. The testbed will be used for the testing purposes and preliminary results.

For the second part we plan to implement the machine-learning classification approach and deploy it on the university cluster to train the HMM-model.

The final step is to integrate the classifiers with an existing CRM. It will allow us to verify compare results with the state of the art solutions.

6. EXPECTED OUTCOMES

We expect that this new information unveiled by the proposed methods will considerably improve power efficiency and performance of large cluster installations.

Further steps include the research activity for the mobile sector where power budget is very limited and efficiency is very important.

7. REFERENCES

[1] S. Derr. Cpusets, 2011.

[2] H. Li, S. Bhunia, Y. Chen, T. N. Vijaykumar, and K. Roy. Deterministic clock gating for microprocessor power reduction. In *In Proc. of 9 th Int'l Symp. on High Performance Computer Architecture (HPCA)*, pages 113–122, 2003.

[3] E. Pinheiro, R. Bianchini, E. V. Carrera, and T. Heath. Dynamic cluster reconfiguration for power and performance, 2002.

[4] T. Technologies. White paper: Processor affinity, 2003.

[1]The processes with the same name can have very different initial conditions and therefore cannot be aggregated into a single entity.