Power-Efficient Server Utilization in Compute Clouds

Jörg Lenhardt, Wolfram Schiffmann
Faculty of Mathematics and Computer Science
Computer Architecture Group
University of Hagen
Joerg.Lenhardt,Wolfram.Schiffmann@FernUni-Hagen.de

Patrick Eitschberger, Jörg Keller
Faculty of Mathematics and Computer Science
Parallelism and VLSI Group
University of Hagen
Patrick.Eitschberger,Joerg.Keller@FernUni-Hagen.de

High performance servers of data centers for cloud computing consume immense amounts of energy even though they are usually underutilized because they provide huge computing capabilities. In times when not all of those computing capabilities are needed the task to be solved is how to distribute the load in a power-efficient manner. The research question is: How should a requested compute load be mapped to the available physical servers so that it is executed with the minimum power consumption? The requested load is measured in operations per seconds and changes over time. In this work, we assume that it is *divisible* which means that portions of the requested load can be freely assigned to different servers. This assumption is plausible because the load of a typical compute cloud consists of many virtual machines (VM). Our investigations are based on the SPECpower benchmark, retrieved Jan 9, 2013. SPECpower relies on Server Side Java (SSJ) for measuring power consumption of servers at different load levels running Java applications [7].

Even though a current VM load on a server can't be correlated directly to a corresponding SSJ-operations' rate, SPECpower results can be used to model the relationship between power consumption and the computational load of the VMs running on used servers. E.g., higher VM's service levels correspond to a larger number of SSJ-operations and thus the server will consume more power. Contemporary server systems include enhanced technologies (like DVFS) for power management. In general, the operating system in connection with on-board firmware take care for adapting the power consumption to the current load. Thus, we have to view a single server as a black box. Its power consumption solely depends on the assigned load and the relationship between both values is given by its SPECpower benchmark data. In order to minimize the power consumption of a cloud system we must find an appropriate distribution of the overall load to the available servers.

In our work we developed algorithms which calculate a power-efficient distribution of a divisible workload among multiple, heterogeneous physical servers. The power consumption of a single server is described by fitting a cubic power function to the real data measurements from the SPECpower benchmarks. We assumed a fully divisible load to calculate an optimized utilization of each server and devised four strategies to distribute a given load in a power-efficient manner on a collection of heterogeneous servers. The first three strategies are straightforward while the fourth is more complex. We concentrate on a collection of servers which process a given computational demand of o operations per second.

Relative load balancing (rlb) distributes the load o to all systems in such a way that all servers are equally loaded. So if the demand is 10% of the overall load each system is assigned a fraction of o so that it runs at 10% load. **Absolute load balancing (alb)** distributes o so that all systems process the same amount of operations per second. This works fine till the least powerful system is fully loaded. Then the remaining load is equally distributed among the remaining systems and so on. **Best performance to power ratio first (bppf)** sorts the systems in decreasing order according to their performance to power ratio at 100% load. At first the system with the highest performance to power ratio is filled up to 100% load. After that, the system with the second highest ratio is used and so on until o is 0. **Adaptive load distribution (ald)** is a more sophisticated approach to calculate the distribution of a specific load on n servers. The key idea is to assign each server a portion α_i of the requested load. The total power consumption of the load request o can be calculated by summing up the partial power of the different servers as $p_{total}(o) = \sum_{i=1}^{n} p_i(o \cdot \alpha_i)$. The values of the scaling factors α_i are constrained in the following way: 1. $\alpha_i \geq 0$, 2. $\alpha_i \cdot o \leq o_i^{max}$, and 3. $\sum_i \alpha_i = 1$. Constraint 1 and 2 ensure that the parameter to the power function p_i is in its domain of definition. o_i^{max} represents the maximum operations per second that a server is able to perform when its load is at 100%. Constraint 1 ensures that no negative amount of operations is assigned to a server. Constraint 2 ensures that no server will be overloaded. The third constraint ensures that the requested load is completely distributed. Of course o must not exceed $\sum_i o_i^{max}$. We considered four configurations consisting of 16 nodes each. Every node can be a single server or a cluster of identical servers for which a single power function exists. Here we just present the results for a typical configuration of cloud system that evolutionary developed over a period of six years. Of course, the number of nodes can be scaled by an arbitrary factor in order to model a real big data center.

The **2006-to-2012-cloud configuration** consists of 12 single systems and 4 clusters. There is one 4-server cluster, one 16-server cluster, one 18-server cluster and one 32-server cluster. The SPEC performance to power ratio is between 268 and 5,521 ssj_ops/Watt. Power consumption is at about 5.3 to 19.1 kW. The server hardware got available between November

2006 and October 2012. Two systems are equipped with AMD Opteron processors (2356 and 8376HE), 14 systems with Intel Xeon processors (5160, E5420, E5440, E5462, E5472, 2 x L3360, 2 x L5420, 2 x L5430, L7345, 2 x X5570). Main memory is 4 GB (2 servers), 8 GB (34 servers), 12 GB (4 servers), 16 GB (3 servers), 24 GB (34 servers), 32 GB (3 servers) or 128 GB (2 servers). Each server is equipped with one, two or eight processors containing two, four, six or eight cores. Eight systems running Windows 2008 Enterprise Edition, six systems Windows 2003 Enterprise Server, one system SuSE Linux ES 10, one system Red Hat EL 5.3. The best performance to power ratio was achieved by an Intel Xeon E5-2470 16-node cluster at 2.3 GHz running Windows 2008. The worst performance to power ratio was achieved by an Intel Xeon 3040 system at 1.8 GHz running Windows 2003.

In Fig. 2 we see that *ald* and *bppf* perform best. Only at 10% load *bppf* leads to a much lower power consumption, at 50% and 60% to a slightly lower power consumption. On higher load levels over 80% *bppf* has the worst results. Both, *rlb* and *alb* are worse than the other two strategies up to 70% load. *ald* led to 18.2% less power consumption than *rlb* at 30% load. The average is at about 8.9%. This configuration covers systems that were installed over a pretty long period of time from November 2006 til October 2012. A great variety of different hardware and power reduction facilities is present and we can achieve the most advantages compared to a simple load balancing (*rlb*) by *ald* or *bppf*. The greatest benefit is achieved at lower load levels up to 50%.

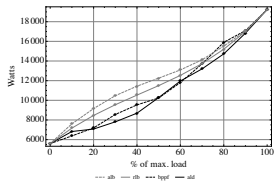


Figure 1. Results for the 2006-to-2012-cloud configuration

Power management can be implemented on node level [5] or on cluster level [4] that belongs to our approach. Moreover, low-power and energy-aware computing can be distinguished. While the former one relies on a large number of low-power processors [8], we are focusing on the second approach where the power consumption of the nodes is controlled usually by means of DVFS [2], [9] and automatically adapts to the requested load. While [6] only considers homogeneous cluster nodes, our approach can also manage heterogeneous clusters which will be found in recent cloud computing data centers.

Our contribution consists mainly in modeling the total power consumption based on recent SPECpower_ssj2008 measurements and the partitioning of a given workload to a collection of commercially available servers. Moreover, we devised a switching off algorithm that can further reduce power consumption already achieved by the partitioning. The switching off approach identifies redundant servers and redistributes their load share to the remaining nodes. By switching off those servers, their idle power can be saved which results in a significant reduction of the overall power consumption.

REFERENCES

[1] Shekhar Borkar and Andrew A. Chien. The future of microprocessors. *Communications of the ACM*, pages 67–77, 2011.

[2] Jason Cong and Bo Yuan. Energy-efficient scheduling on heterogeneous multi-core architectures. In *Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design*, ISLPED '12, pages 345–350, New York, NY, USA, 2012. ACM.

[3] E. Feller, C. Morin, D. Leprince, et al. State of the art of power saving in clusters and results from the edf case study. 2010.

[4] E. Feller, C. Rohr, D. Margery, and C. Morin. Energy management in iaas clouds: A holistic approach. In *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, pages 204–212. IEEE, 2012.

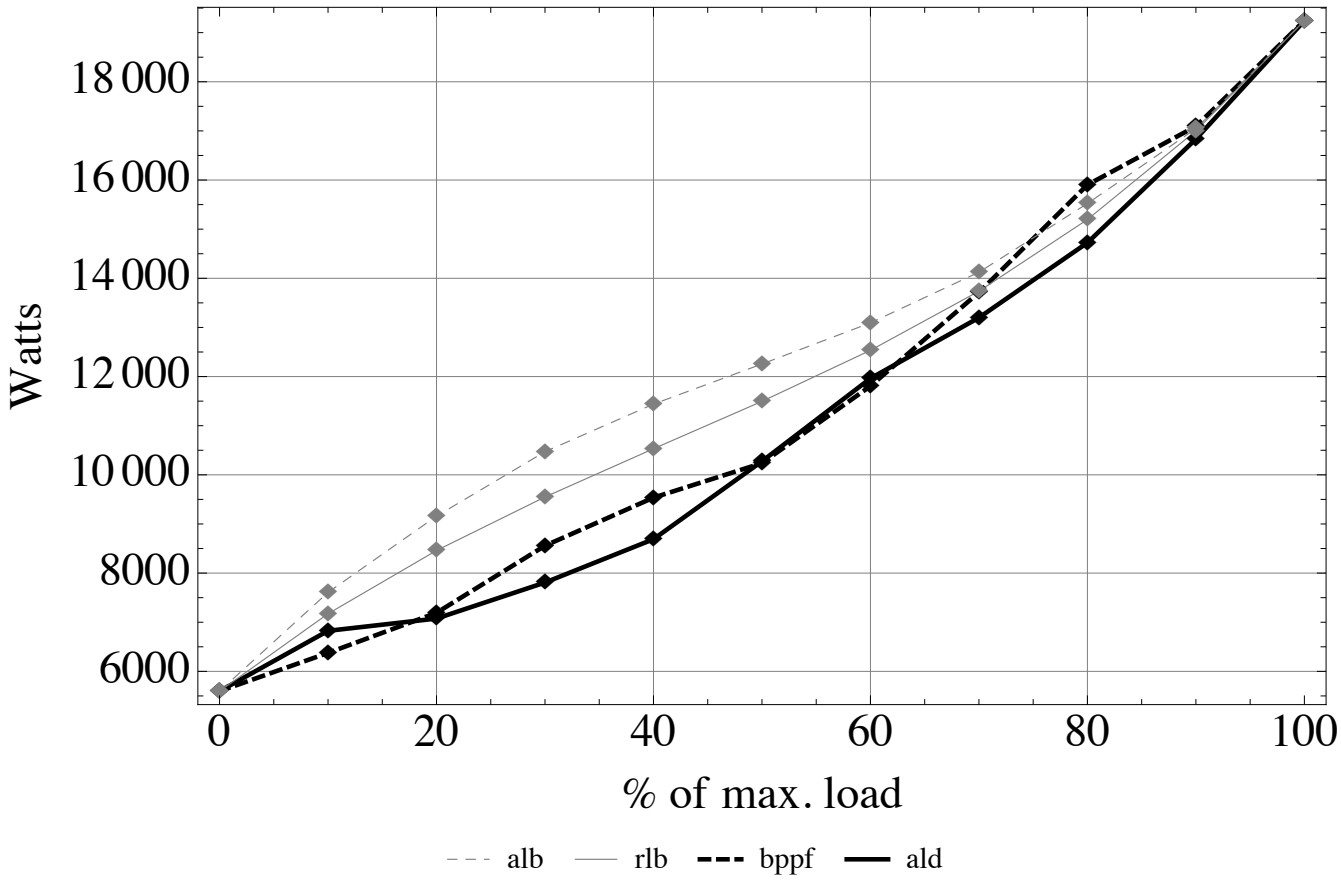
[5] J. Howard and et al. A 48-core ia-32 message-passing processor with dvfs in 45nm cmos. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, pages 108 –109, feb. 2010.

[6] Eduardo Pinheiro and Ricardo Bianchini. Load balancing and unbalancing for power and performance in cluster-based systems. *Proceedings of the Workshop on Compilers and Operating Systems for Low Power*, 2001.

[7] SPEC. *SPEC — Power and Performance, Design Document, SSJ Workload, SPECpower_ssj2008, rev1137*, 2012.

[8] Ibrahim Takouna, Wesam Dawoud, and Christoph Meinel. Energy efficient scheduling of hpc-jobs on virtualize clusters using host and vm dynamic configuration. *SIGOPS Oper. Syst. Rev.*, 46(2):19–27, July 2012.

[9] G. von Laszewski, Lizhe Wang, A.J. Younge, and Xi He. Power-aware scheduling of virtual machines in dvfs-enabled clusters. In *Cluster Computing and Workshops, 2009. CLUSTER '09. IEEE International Conference on*, pages 1 –10, 31 2009-sept. 4 2009.


18 000
16 000
14 000
12 000
10 000
8000
6000
Watts
0
20
40
60
80
100
% of max. load
alb
rlb
bppf
ald