Tailoring resources: the energy efficient consolidation strategy goes beyond virtualization

Jordi Torres, David Carrera, Vicenç Beltran, Nicolás Poggi, Kevin Hogan, Josep Ll. Berral, Ricard Gavaldà, Eduard Ayguadé, Toni Moreno and Jordi Guitart.

Barcelona Supercomputing Center (BSC) - Technical University of Catalonia (UPC) - Barcelona (Spain)
torres@ac.upc.edu

Abstract

Virtualization and consolidation are two complementary techniques widely adopted in a global strategy to reduce system management complexity. In this paper we show how two simple and well-known techniques can be combined to dramatically increase the energy efficiency of a virtualized and consolidated data center. This result is obtained by introducing a new approach to the consolidation strategy that allows an important reduction in the amount of active nodes required to process a web workload without degrading the offered service level. Furthermore, when the system eventually gets overloaded and no energy can be saved without loosing performance, we show how these techniques can still improve the overall value obtained from the workload. The two techniques are memory compression and request discrimination, and were separately studied and validated in a previous work to be now combined in a joint effort. Our results indicate that an important improvement can be achieved by deciding not only how resources are allocated, but also how they are used. Moreover, we believe that this serves as an illustrative example of a new way of management: tailoring the resources to meet high level energy efficiency goals.

1. Introduction

In this paper we present how two simple and well-known techniques can be combined to dramatically increase the energy efficiency of a virtualized and consolidated data center. Increased energy efficiency is obtained through the introduction of a new approach to the consolidation strategy by combining: memory compression and request discrimination. Combining these techniques enables an important reduction in the amount of active nodes required to process a web workload by dynamically classifying and shaping the workload, without degrading the offered service level. Furthermore, when the system eventually gets overloaded and no energy can be saved without loosing performance, we show how request discrimination can still improve the overall value obtained from the workload. The two techniques were separately studied and validated in a previous work to be now combined in a joint effort. Memory compression is used to convert CPU power into additional system memory. The amount of extra memory produced using this technique can potentially go beyond consolidation through virtualization by allowing the placement of an extra application that did not fit in a node before, therefore reducing node underutilization. Request discrimination is introduced to classify web requests according to the value they have to the system with the purpose of prioritiezing those requests that add more value to the system, in overload conditions.

Notice that both of the techniques described here can produce a similar effect in a system: reducing the number of nodes necessary to meet a certain service level criteria. This extra consolidation is achieved through memory compression by increasing the number of application instances that can be placed in a node, and through request discrimination by reducing the load on the system, thus allowing more options to collocate applications.

1.1 Benefits of memory compression

We consider a scenario composed of 3 identical servers and 4 different web applications. Neither allocation restrictions nor collocation restrictions are defined, but placement is still subject to resource constraints, such as the node memory and CPU capacity. We also consider 4 different applications, in which application 1 can not be placed together with any other application because of the memory constraints, whereas all its CPU demand can be satisfied by one single node. Applications 2, 3 and 4 can be collocated, but only two of them can be placed together in each node.

Figure 1 shows the memory and CPU consumption observed in our experiments. Notice how memory constraints lead to a situation where the three nodes are clearly underutilized in terms of CPU power. At this point, we introduce the use of memory compression to increase the memory capacity of a node on demand. The memory is produced at a cost in terms of CPU power. In the scope of this experiment, we assume an achievable compression factor of 47% (see [1] for more details), and will use an 50% increased memory capacity for each active node at a cost of 1320MHz of CPU power. The resulting CPU and memory consumption for the same workload when memory compression is applied can be seen in Figure 2. Initially, the four applications are placed in node 3, until point A is

978-0-7695-3175-5/08 $25.00 © 2008 IEEE
DOI 10.1109/ICAC.2008.11

reached. Notice how 2 nodes can be switched off before that point. At that point, more CPU power is denadned, and application 1 is migrated to a second node which is switched on for this purpose. Finally, at point D, aggregated CPU demand for all the applications can be satisfied again with one single node and thus all the applications are placed again in server 3.

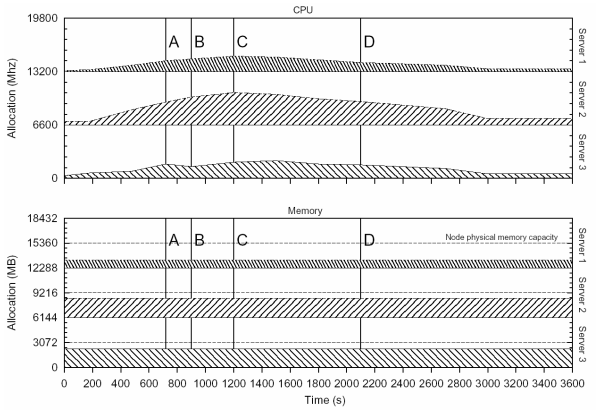

Figure 1. Per-node CPU and memory consumption without memory compression

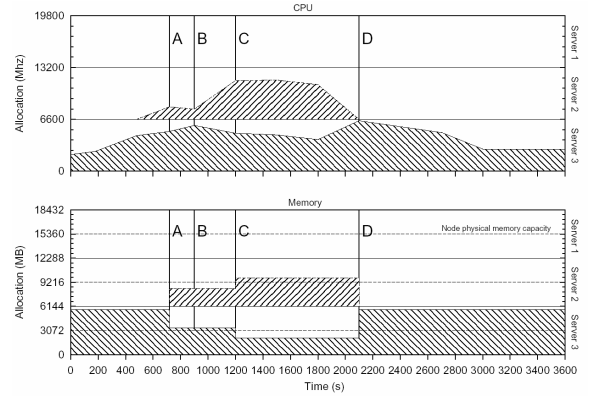

Figure 2. Per-node CPU and memory consumption with memory compression

.

1.2 Benefits of request discrimination

Once a placement is decided, the resources allocated to each application are determined. After that, the application can be either overloaded or not, depending on the observed load. In the case an application is overloaded and not all the demand can be satisfied, a portion of the received requests must be dropped. Even in this hard scenario, request discrimination can be useful to increase the aggregated value of the system by considering the value associated to each request. Based on real access logs obtained from a top national travel company, Figure 3 shows the total number of requests corresponding to buying clients present in the workload (solid line), as well as how many of these requests are accepted when overloading requests are rejected following a random policy (dashed line) and a machine-learning-based policy (dotted line) as described in [2,3]. It can be observed from the figure, that in periods of time when the system is not overloaded all the requests are accepted and thus all the requests corresponding to purchasing sessions are processed. On the other hand, when the system is overloaded, using a good discrimination technique can improve the obtained result by dropping first the requests corresponding to non-purchasing.

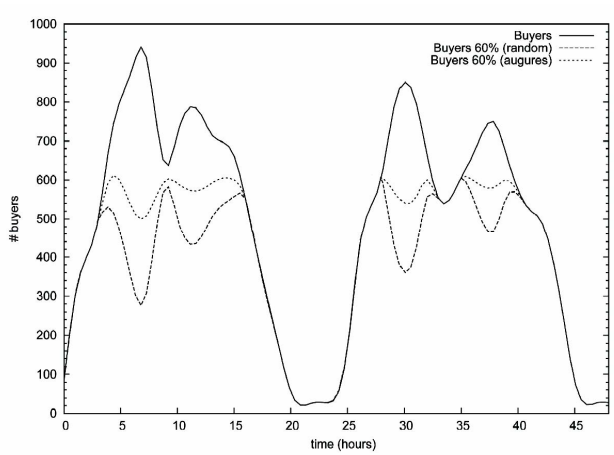

Figure 3. Total requests corresponding to buyer clients Vs accepted buyers with and without request discrimination

Acknowledgments

This work is supported by the Ministry of Science and Technology of Spain and the European Union (FEDER funds) under contracts TIN2007-60625 and TIN2005-08832-C03-03

References

[1] V.Beltran, J. Torres and E. Ayguade "Improving Disk Bandwidth-Bound Applications Through Main Memory Compression" MEDEA Workshop MEmory performance: DEaling with Applications, systems and architecture. Brasov, Romania. Held in conjunction with PACT 2007 Conference Sept. 15-19 2007

[2] N. Poggi, J.L. Berral, T. Moreno, R. Gavaldà and J. Torres. "Automatic Detection and Banning of Content Stealing Bots for E-commerce". In Workshop on Machine Learning in Adversarial Environments for Computer Security (NIPS 2007).British Columbia, Canada. Dec. 2007

[3] N.Poggi, T. Moreno, J. Berral, R. Gavaldà, J. Torres. "Web Customer Modeling for Automated Session Prioritization on High Traffic Sites". In 11th International Conference on User Modeling. Corfu, Greece, June, 2007.