Find Your Best Match: Predicting Performance of Consolidated Workloads

Danilo Ansaloni
University of Lugano
Lugano, Switzerland
danilo.ansaloni@usi.ch

Lydia Y. Chen
IBM Research Zürich Laboratory
Rüschlikon, Switzerland
yic@zurich.ibm.com

Evgenia Smirni
College of William and Mary
Virginia, USA
esmirni@cs.wm.edu

Akira Yokokawa
University of Lugano
Lugano, Switzerland
akira.yokokawa@usi.ch

Walter Binder
University of Lugano
Lugano, Switzerland
walter.binder@usi.ch

ABSTRACT

Modern multicore platforms allow system administrators to reduce the costs of the IT infrastructure by consolidating heterogeneous workloads on the same physical machine. To this end, it is important to develop efficient profiling techniques and accurate performance predictions to avoid violating service-level objectives. In this work we present Tresa, a novel tool to automatically characterize workloads and accurately estimate the execution time of different consolidations. These results can be used to optimize consolidations depending on service-level objectives.

Categories and Subject Descriptors

D.4.8 [**Operating Systems**]: Performance—*Modeling and Prediction*

General Terms

Management, Measurement, Performance

Keywords

Workload consolidation, performance modeling

1. INTRODUCTION

Administrators of large data centers and cloud computing platforms often struggle to consolidate sets of heterogeneous workloads on the same physical machine in order to maximize resource utilization without violating service-level objectives, such as maximum execution time. This problem is often challenging because performance interference between consolidated workloads may significantly affect their execution time [1].

In this work we present Tresa, a tool that helps system administrators choosing how to consolidate workloads by providing precise predictions of their execution time. The novel scientific contributions are:

1. a workload characterization technique based on standard, low-overhead tools (i.e., `iostat`, `mpstat`, `sar`, and `vmstat`) available on prevailing UNIX-like systems;

ICPE'12, April 22-25, 2012, Boston, Massachusetts, USA
ACM 978-1-4503-1202-8/12/04.

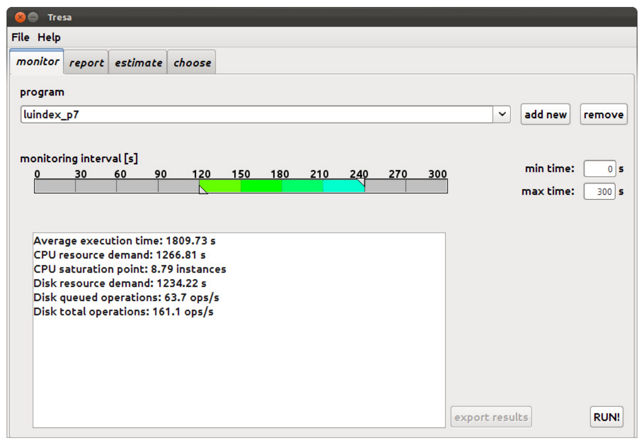


Figure 1: Screenshot of Tresa: interface of the black-box profiler.

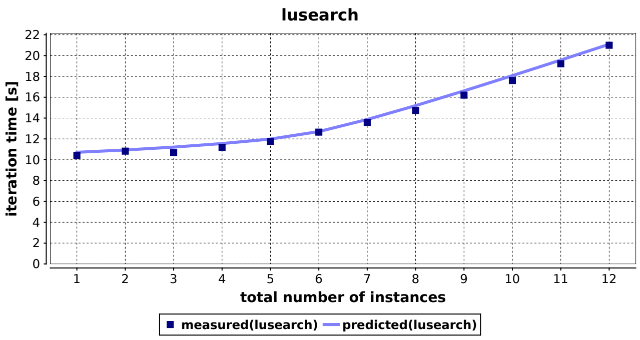
2. a mathematical model based on queueing theory, solved using mean-value analysis;
3. a tool to automate workload characterization and to predict the average execution time of different consolidations.

2. TRESA

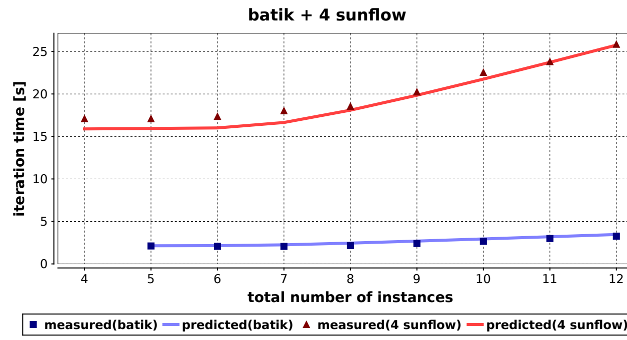
The graphical interface of Tresa presents 4 tabs (i.e., `monitor`, `report`, `estimate`, and `choose`) that allow users to characterize workloads and to choose consolidations that optimize throughput without exceeding a given maximum execution time.

Figure 1 shows the `monitor` tab, which provides an interface to our black-box profiler. Users have to specify how to start the workload and choose the duration of the warm-up phase and of the observation interval. Tresa profiles the execution of a single instance of the specified program and stores all relevant runtime metrics.

In the `report` tab, users can visualize all collected values and derived metrics. This data is internally used by Tresa to compute the *resource demands* of a single program, which are the inputs to our queueing network model. Tresa uses



(a) Homogeneous consolidations of multiple instances of `lusearch`



(b) Heterogeneous consolidations of 4 instances of `sunflow` with multiple instances of `batik`

Figure 2: Measured and predicted iteration time of consolidations of DaCapo benchmarks.

mean-value analysis to precisely predict the execution time of consolidations of various workloads.

The `estimate` tab generates detailed charts (used in Section 3) of the predicted execution time for homogeneous and heterogeneous consolidations, that is, consolidations involving a single class, respectively multiple classes, of workloads.

Finally, the `choose` tab provides a high-level view of all predictions, organized in a table format. By selecting a maximum execution time, it is possible to highlight the consolidations that lead to the highest throughput without violating the constraint on execution time.

3. EVALUATION

We evaluate our predictions on an IBM Power 750 Express server, with a single processor board hosting 8 cores running at 3.00GHz and 64GB of RAM. The system runs AIX 6.1 (64 bit) and IBM J9 JVM SR8-FP1 (64 bit). The observed applications are benchmarks from the DaCapo 9.12[1] suite, executed in a loop within the same JVM process and with external concurrency set to 1. The warm-up phase has a duration of 2 minutes and the profiling interval is 3 minutes.

Figure 2(a) reports the average iteration time of homogeneous consolidations of an increasing number of instances of `lusearch`. As predicted by our tool, consolidations of up to 5 instances of `lusearch` do not noticeably affect the iteration time. After 5 instances, the iteration time starts increasing because all cores are used most of the time (i.e., the CPU utilization is close to 100%).

Figure 2(b) illustrates the average iteration time of heterogeneous consolidations of 4 instances of `sunflow` with an increasing number of instances of `batik`. In this case, up to 3 instances of `batik` can be consolidated with 4 instances of `sunflow` without significantly affecting the iteration time. For both benchmarks, our predictions closely match the measured iteration time, with a maximum error of 8.1%.

To evaluate the overall quality of our predictions, we conducted an exhaustive set of experiments of homogeneous and heterogeneous consolidations. Across 160 considered homogeneous consolidations, the average prediction error is only 6.0%, while it is 8.4% across 400 considered heterogeneous consolidations.

[1]Website: `http://dacapobench.org/`

4. RELATED WORK

Wood et al. developed Sandpiper [5], which implements two profiling approaches, (1) a black-box approach (i.e., fully OS- and application-agnostic), and (2) a gray-box approach exploiting OS- and application-level statistics. Moreover, in [4] the authors use a regression-based model to profile and predict application resource requirements in a virtualized environment. Lu et al. [2] developed a profiling methodology that viewed the problem of physical resource utilization as the source of a separation problem in digital signal processing, and designed a directed factor graph (DFG) to successfully model the dependence relationships among different resources (CPU, memory, disk, network) across virtual and physical layers.

In general, little is known about prediction of execution time for consolidations of multiple classes of workloads. To the best of our knowledge, the only theoretical methodology that focuses on this problem is the one presented in [3]. We depart from prior work by applying the theoretical methodology in [3] to solve the difficult problem of predicting performance in a multicore system where multiple programs are consolidated.

5. REFERENCES

[1] Y. Koh, R. C. Knauerhase, P. Brett, M. Bowman, Z. Wen, and C. Pu. An Analysis of Performance Interference Effects in Virtual Environments. In *Proceedings of the 2007 IEEE International Symposium on Performance Analysis of Systems & Software*, ISPASS 2007, pages 200–209, 2007.

[2] L. Lu, H. Zhang, G. Jiang, H. Chen, K. Yoshihira, and E. Smirni. Untangling Mixed Information to Calibrate Resource Utilization in Virtual Machines. In *Proceedings of the 8th ACM International Conference on Autonomic Computing*, ICAC 2011, pages 151–160, 2011.

[3] E. Rosti, F. Schiavoni, and G. Serazzi. Queueing Network Models with Two Classes of Customers. In *Proceedings of the Fifth International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, MASCOTS 1997, pages 229–234, 1997.

[4] T. Wood, L. Cherkasova, K. Ozonat, and P. Shenoy. Profiling and Modeling Resource Usage of Virtualized Applications. In *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*, Middleware 2008, pages 366–387, 2008.

[5] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif. *Black-box and Gray-box Strategies for Virtual Machine Migration*. In *Proceedings of the 4th USENIX Symposium on Networked Systems Design & Implementation*, NSDI 2007, pages 229–242, 2007.