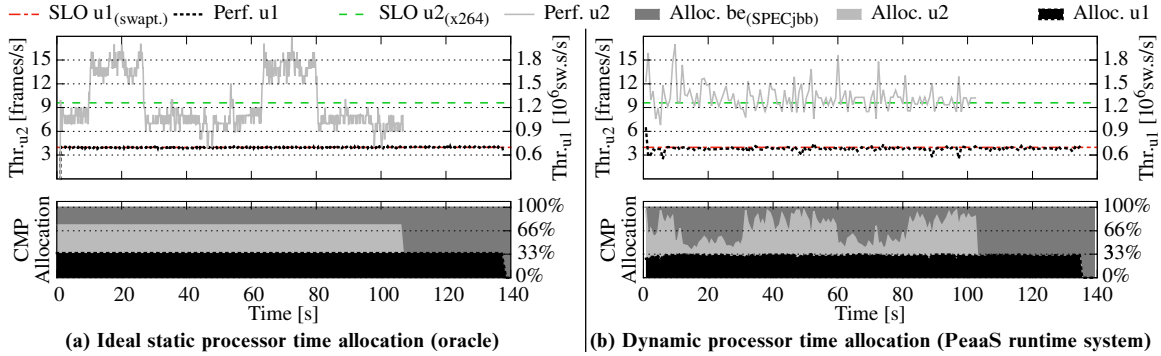
Towards a Performance-as-a-Service Cloud

D. B. Bartolini, F. Sironi, M. Maggio, G. C. Durelli, D. Sciuto, M. D. Santambrogio
Politecnico di Milano



(a) Ideal static processor time allocation (oracle) **(b) Dynamic processor time allocation (PeaaS runtime system)**

Figure 1. Performance and allocations for two performance-sensitive applications (by users u1 and u2) co-located with a batch best-effort (be) workload. The *PeaaS* runtime system allocates the CMP to match SLOs and maximize utilization.

Motivation While the pay-as-you-go model of Infrastructure-as-a-Service (IaaS) clouds is more flexible than an in-house IT infrastructure, it still has a resource-based interface towards users, who can rent virtual computing resources over relatively long time scales. There is a fundamental mismatch between this resource-based interface and what users really care about: performance.

This mismatch affects both users and providers. Users need to fine-tune resource allocations for each virtual machine (VM) or accept the inefficiency of worst-case provisioning [5, 7]. Providers face the problem of mapping virtual resource requests onto the heterogeneous data center infrastructure [4] so as to optimize utilization and maintain consistent performance despite resource sharing [6]. The fine-grained (units of virtual resources) and fast (order of seconds) resource trading scenario that is likely to emerge [1] will magnify these issues.

Contribution We propose to solve the resource-performance mismatch through a Performance-as-a-Service (PeaaS) model. In the PeaaS model, users do not rent virtual resources, but state service-level objectives (SLOs) for performance-sensitive VMs.

We are building a prototype runtime system that automatically allocates resources as-needed to satisfy SLOs; this layer is the enabling technology for the PeaaS model. We are building this runtime system under three primary goals: (1) dispense users with the need for laboriously tuning resource allocations, (2) enable providers to optimize infrastructure utilization, (3) take advantage of fine-grained, high-frequency virtual resource trading.

Initial Results Similarly to the state of the art [8], we leverage a two-level control schema based on application-level controllers and node-level brokers. In contrast with previous work, we simplify the controllers to obtain faster control frequency (sub-second versus tens of seconds); a faster controller allows to react to fast workload variations that may otherwise go unobserved and cause long tail latency distributions [3].

We initially focus on allocating a chip multiprocessor (CMP) to compute-bound applications. Figure 1(b) evaluates our prototype with two performance-sensitive applications (i.e., *swaptions* and *x264*, from the PARSEC benchmark suite [2]) by two users, compared to the case of static allocations (Figure 1(a)) that meet SLOs on average, but cannot adapt to workload variations (see *x264*). We support the co-location of batch best-effort (be) workloads (here we use SPECjbb2005 [9]) to maximize node-level utilization. The PeaaS runtime system is able to enforce SLOs, adapt allocations to varying resource demands, and keep the node fully utilized.

Discussion and Work in Progress Our initial results validate the PeaaS approach on automatically managing processor allocation to competing compute-bound applications. We are extending our prototype to support more resources and application types and we are studying broader issues such as fair pricing in the PeaaS model.

SOCC '13, Oct 01-03 2013, Santa Clara, CA, USA
ACM 978-1-4503-2428-1/13/10.
http://dx.doi.org/10.1145/2523616.2525933

References

[1] O. Agmon Ben-Yehuda, M. Ben-Yehuda, A. Schuster, and D. Tsafrir. The Resource-as-a-Service (RaaS) Cloud. In *Proceedings of the 4th Workshop on Hot Topics in Cloud Computing*, HotCloud '12, pages 12–12, Berkeley, CA, USA, 2012. USENIX Association.

[2] C. Bienia. *Benchmarking Modern Multiprocessors*. PhD thesis, Princeton University, 2011.

[3] J. Dean and L. A. Barroso. The Tail at Scale. *Commun. ACM*, 56(2):74–80, Feb. 2013. doi: 10.1145/2408776.2408794.

[4] B. Farley, A. Juels, V. Varadarajan, T. Ristenpart, K. D. Bowers, and M. M. Swift. More for Your Money: Exploiting Performance Heterogeneity in Public Clouds. In *Proceedings of the 3rd Symposium on Cloud Computing*, SoCC '12, New York, NY, USA, 2012. ACM. doi: 10.1145/2391229.2391249.

[5] D. Gmach, J. Rolia, and L. Cherkasova. Selling T-shirts and Time Shares in the Cloud. In *Proceedings of the 12th International Symposium on Cluster, Cloud and Grid Computing*, CCGRID '12, pages 539–546, Washington, DC, USA, 2012. IEEE Computer Society. doi: 10.1109/CCGrid.2012.68.

[6] S. Govindan, J. Liu, A. Kansal, and A. Sivasubramaniam. Cuanta: Quantifying Effects of Shared On-chip Resource Interference for Consolidated Virtual Machines. In *Proceedings of the 2nd Symposium on Cloud Computing*, SoCC '11, New York, NY, USA, 2011. ACM. doi: 10.1145/2038916.2038938.

[7] J. Mars, L. Tang, R. Hundt, K. Skadron, and M. L. Soffa. Bubble-Up: Increasing Utilization in Modern Warehouse Scale Computers via Sensible Colocations. In *Proceedings of the 44th International Symposium on Microarchitecture*, MICRO '11, pages 248–259, New York, NY, USA, 2011. ACM. doi: 10.1145/2155620.2155650.

[8] P. Padala, K.-Y. Hou, K. G. Shin, X. Zhu, M. Uysal, Z. Wang, S. Singhal, and A. Merchant. Automated Control of Multiple Virtualized Resources. In *Proceedings of the 4th European Conference on Computer Systems*, EuroSys '09, pages 13–26, New York, NY, USA, 2009. ACM. doi: 10.1145/1519065.1519068.

[9] SPEC. SPECjbb2005. `http://www.spec.org/jbb2005/`.