Towards Managing Variability in the Cloud

Ali Anwar, Yue Cheng, Ali R. Butt
Virginia Tech
{ali, yuec, butta}@cs.vt.edu

***Abstract*—Performance variability in advanced computing systems, such as those supporting the cloud computing paradigm, is growing intractably and leads to inefficiency and resource wastage. A key requirement in large-scale virtualized infrastructure, e.g., Amazon EC2, Microsoft Azure, etc., is to provide a guaranteed quality of service to cloud tenants, especially in today's multi-tenant cloud environments. This generally involves using past information and prediction of the probability distribution of requests to match resources that meet service-level agreements. The variability in systems performance hinders the cloud service providers' ability to effectively guarantee SLAs, and thus efficiently meet user demands.**

In this paper, we propose innovative methodologies for resource management, which leverages the understanding of performance variability in high performance computing systems to exploit new opportunities for tradeoffs between system stability and performance in the cloud. This would help cloud providers better provision and design their infrastructure, as well as ensure meeting provider-tenant SLAs. Moreover, the approach also leads to improved cloud service costs, as tighter bounds on variability could be codified in cost structures bundled in operations or directly offered to cloud tenants.

I. Introduction

The cloud computing model has emerged as the de facto paradigm for efficiently providing infrastructure, platform, and application services. Performance variability in cloud based systems is growing intractably and leads to inefficiency and resource wastage. Such variability has also been cited as a significant barrier to exascale computing [40]. However, the impact of variability on cloud systems has not been explored thus far. Unfortunately, variability is both ubiquitous and elusive as its causes pervade and obscure performance across the systems stack from hardware to middleware to applications to large-scale systems.

Cloud computing environments comprise a complex array of compute, storage, networking, and I/O components. That coupled with the on-demand nature of cloud services instantiation may result in unpredictable performance when utilizing cloud-based services. Recent studies [15], [39] have shown that performance unpredictability acts as one of the major obstacles for cloud computing. Cloud users expect consistent performance for their applications at any time, independent of the current workload of the cloud; this is quite important for research community as well, as the repeatability of results is highly desirable. As more and more users build their customer-facing services using cloud-based backend components, the performance consistency requirements are becoming paramount. Variability in performance does not only affect cloud tenants but also the cloud service providers. For example, to meet a certain Service Level Agreement (SLA), cloud providers are expected to make Quality of Service (QoS) guarantees. Variability in performance hinders cloud providers' ability to effectively guarantee SLAs based on performance features.

Cloud service provider	Memory bandwidth	Across instances	Within instances
Rackspace [6]	6.7 GB/s	$+/-10\%$	$+/-7\%$
Digital Ocean [3]	6.3 GB/s	$+/-10\%$	$+/-8\%$
HP Cloud [4]	5.9 GB/s	$+/-7\%$	$+/-4\%$
Amazon m1.medium [1]	5.2 GB/s	$+/-25\%$	$+/-4\%$
Windows Azure [8]	3.6 GB/s	$+/-5\%$	$+/-6\%$

TABLE I: Memory performance variability of different cloud service providers. Across instances: performance measured from multiple instances of the same type; Within instances: performance measured from a single instance.

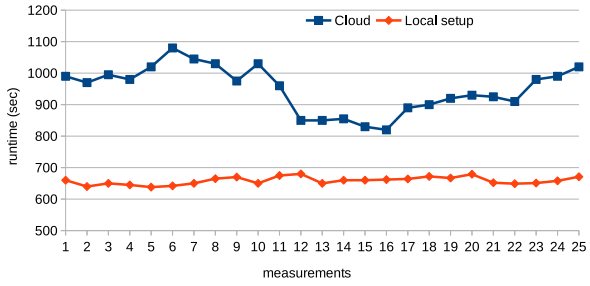

Fig. 1: Variation in runtime of a MapReduce job when executed in a cloud based setup vs. a local setup. A total of 25 measurements were collected.

To illustrate the impact of variability on system performance, Table I [7] shows the memory bandwidth variation (measured using the Cloudlook benchmark [5]) across virtual machine (VM) instances and within single VM instances of several well-known cloud service providers. While the average memory performance ranges from 6.7 GB/s to 3.6 GB/s, we observe non-trivial performance variation both across VM instances and within VM instances. Consider the example of `m1.medium` type VM in Amazon EC2. Here, the variability was observed to be highest and the standard variation exceeded around 50% $(+/-25\%)$ of the average. Similarly, variability was observed to be relatively low within the VM instances, but can still be as high as 16% $(+/-8\%)$ for Digital Ocean.

To further demonstrate the impact of variability in virtualized infrastructure, a sample set of measurement on a data-intensive batch application (MapReduce [22]) were collected at different times within a single day on a 20-node cluster, both on a cloud environment (Amazon EC2) and a local cluster setup. Figure 1 shows the results. We see that the performance on the cloud based setup varies considerably. A series of factors contribute to the significant variability of the multi-tenant cloud based performance, e.g., memory bandwidth variability, CPU sharing, and network contention. For the local cluster test, among various reasons for the

relatively smaller performance inconsistency, contention for non-virtualized resources (e.g. network bandwidth) is perhaps the major reason.

The hypothesis of this paper is that *with a deeper understanding of the relationship between system configurations, e.g., VM instance, network contention, and the resulting performance variability, e.g., expressed as a probability density function (PDF), we can better manage as well as mitigate the effects of variability on cloud systems.*

II. Background and Motivation

Variability challenges in traditional advanced computing systems, such as High Performance Computing (HPC), pervade every aspect of the systems stack and require a holistic solution. In the Cloud era, the scale, complexity, and heterogeneity of large-scale virtualized infrastructure systems require fundamental new solutions that are applicable to a broad class of systems and applications. This requires innovative adaption of existing techniques that have been traditionally applied to HPC to the cloud environment.

Variability in HPC Systems. HPC systems have been explored for variability and its effects. Hoefler et al. [24] recently summarized the state of the practice for benchmarking in HPC and suggested ways to ensure repeatable results. Similarly, HPC researchers have been examining variance for a long time, e.g., in the 1990s IBM observed variance in uniprocessors [35], and Kramer et al. [29] explored variation in large distributed memory systems more than a decade ago. Such well-cited studies establish the existence of variability, and the need for mitigating its impact in HPC.

Reproducibility and Repeatability. Studies aimed at improving the ability to verify experimental results can directly or indirectly address variability. For example, recent studies have shown that minor aspects of experimental setups can have a significant impact on system performance. The high degree of sensitivity to experimental design can potentially invalidate conclusions. This is further complicated by lack of reproducibility, corroboration and repeatability, thereby, propagating invalid conclusions in academia [44]. Such complications delay the adoption of academic results, e.g., in industrial systems. To this end, there is a growing need to perform deep analyses to identify the root causes of observed results. Similarly, introducing determinism to achieve reproducibility also needs to be explored using environment categorization [37], statistical modeling [42], or variance-aware algorithm design [16].

Variability in Real Time Operating Systems. Runtime predictability is a key concern in Real-time Operating System (RTOS) design, where mission critical application deadlines have to be met with high accuracy. Hence, RTOSs sacrifice throughput/performance for predictability [38]. Furthermore, RTOS approach can not be simply used in cloud applications because: (i) In RTOSs, only a limited number of tasks can run at the same time, hence limiting the achievable level of multitasking [34]. Cloud systems of scale routinely handle hundreds of thousands to millions of threads. (ii) By limiting the supported set of real-time threads and applications, such systems tend to have a higher ratio of resources to demands in order to meet deadlines, i.e., cores, memory and networking bandwidth are scaled to meet real-time design requirements. Applying these same principles to the cloud, would likely be prohibitively expensive. (iii) The algorithms and scheduling techniques at the heart of a RTOS are primarily designed to meet deadlines. Many general purpose applications common in cloud computing environments, do not benefit from the same types of optimizations. (iv) RTOS systems are hard to develop and many cloud/web frameworks do not run just out of the box on them.

Variability in Computer Architecture. Computer architecture research has explored variability in studying and realizing new hardware, arising mainly from heterogeneity in the chip-level architecture. These works are mainly focused on the consequences of the chips having a limited amount of power [17], [11]. The resulting issues such as leakage current, cross-wire-signaling, and real-estate budgeting, result in expected performance to vary significantly. If these techniques affect software performance variability (e.g., VMs controlled by hypervisors for providing cloud services), then they can be captured and are orthogonal to our proposed work. Otherwise, such variations are likely to be small relative to the variations observed much higher in the systems software stack.

Variability in the Cloud. Moving from HPC to the cloud imposes new challenges in solving the variability issues. The use of virtualized infrastructure in the cloud is likely to demonstrate higher load imbalance and performance fluctuations [19] due to the factors such as resource sharing across multiple tenants, compute/memory resource over-subscription, etc. This is challenging task for cloud service providers as tenants' behaviors can be treated as a large set of unpredictable variables, and a weak model may end up with cloud providers lose their tenants and thus profit [10], [20]. A line of research [41], [27], [31] attempts to provide effective performance isolation in multi-tenant datacenters to minimize the performance unpredictability and variability. Researchers have focused on providing a robust framework [9] in cloud datacenter for effectively handling variability caused by multi-tenancy and resource contentions, which are beyond a cloud tenant's control.

III. Mitigating Variability in Cloud Systems

We aim to manage and mitigate the variability in cloud systems. Our approach is rooted in experimentally studying the said variability, and developing models to utilize the information in designing variability-aware resource managers for the cloud.

A. Measuring Variability in Computing Environments

Both the current load on a cloud as well as the underlying architecture used for launching a cloud instance play an important role in defining the performance of a cloud instance launched on that setup, e.g., [39] shows that MapReduce jobs perform better on EC2 when using a larger percentage of Xeon-based systems than Opteron-based systems. Similarly [28] identifies the problem of performance variability in their study, however solutions to address the problem or measure the variability have not been developed.

The unique facets of the performance variability in the cloud are the multi-tenant use of the shared resources, and that different cloud service providers utilize different resource scheduling and allocation mechanisms. Thus, the impact on the performance of different workloads is different for different cloud providers. For example, in a multi-tenant environment, priority based scheduling [32], [33] chooses to offer more resources for workloads with higher priority, which may affect the performance of a certain workload. On the other hand, an environment equipped with load-balancing scheduling [25], [23], [43] may affect the same workload differently, e.g., loss of data locality. Such different scheduling choices and their impact on performance variability needs to be studied in detail to quantify the impact.

Similarly, cloud storage offers customers the key benefits of on-demand elastic scalability and usage-based pricing [21], [30]. However, variation in the performance of cloud storage services can lead to cloud applications violating their SLAs [46]. The main source of variability in cloud storage performance has been identified to be interference from co-located tenants. As the logical partitioning between different tenants' data does not map to separate physical partitions, applications from different tenants could contend for the same disk resources, which result in lower overall IOPS for both of the co-located tenants [41], [26]. In addition, even if cloud storage services achieve even distribution of data across their deployed hardware, the skew in the demand of individual data objects will result in unfairness in the usage of storage devices [41].

B. Leveraging Variability for Cloud-based Service Design

The cloud providers need to comply with their customers SLAs by scaling up their setup according to the load on the systems [14], [12]. Accurate measurements of variability provide information that can be useful for the cloud providers to offer tighter performance-based SLA guarantees to users, and hence scale their setup in a more cost-aware manner [13].

A recent study [18] shows a clear trade-off between latency and cluster utilization for MapReduce applications in cloud environments. Such trade-off is caused by different and sometimes conflicting optimization goals of the application developers and cloud service providers. Different cloud environment variability yields new conflicts that might influence application performance in different ways. On the other hand, the diversity of MapReduce usage scenarios makes it hard to develop a single solution for all applications and environments. With the knowledge of variability measurement, an application could choose the cloud based on its characteristics. For example, end-user services that are susceptible to performance variability, such as video streaming services, high-speed trading, etc., would choose more stable environments. At the same time, users on a tighter budget can make a more informed decision about whether to employ cheap services such as spot cloud instances, if the variability characteristics of such instances are known (or can be readily determined).

Similarly, to mitigate the impact of variability on cloud storage performance, tenants resort to employing redundant resources via replication and additional layers of load distribution to minimize the effect of this variability on the IOPS achieved by the applications [45]. For such tenants, the ability to measure the variability in IOPS as envisioned by the proposed work will provide for a powerful paradigm that users can leverage to determine the amount of redundancy in resources required to achieve a prespecified quality of service (QoS) for their applications. This, in turn will lead to better resource planning and stronger SLA guarantees. Good measures of variation can also help cloud providers with implementing better isolation and fairness mechanisms [41] as well.

C. Variability-proofing Cloud-based Services

The variability in a cloud caused by differences in underlying architecture can be minimized by allowing cloud tenants to choose the underlying physical hardware configuration, e.g., network locality, processor, memory type, storage device, etc. To help reduce the variability in multi-tenant or shared cloud environments, it is important to accurately predict the future resource usage by understanding the usage pattern of a cloud setup. Also, different tracking mechanism can be used to keep tabs on performance variability so that performance variability can be studied and made more predictable.

A number of research projects explore how to optimize MapReduce based on underlying cloud environments from variability perspectives [48], [47]. One of such optimizations can be adopted for a specific underlying environment to help reduce the performance variability if we know the dominant causes of the variability. For example, if the variability is mainly caused by heterogeneity, we might consider applying Longest Approximate Time to End (LATE) [48]; if the variability is mainly caused by data locality, more advanced data placement schemes [47], [30], [36] are worth consideration.

Such techniques can be used to build variability-proof applications atop cloud services. The motivation for such services is the Chaos Monkey toolkit [2] used by Netflix Inc. to shield their services against vagaries of a cloud. At a high level, we propose to identify the variability in the system, and then introduce sufficient redundancy within the instantiated services so as to mitigate the expected loss in availability and performance. For instance, a compute instance can be enhanced with additional resources, a storage layer can be given added copies, and I/O can be over-provisioned accordingly.

IV. Conclusion

The cloud environment comprises a plethora of resources, each with its own performance characteristics and stability, which leads to a high degree of variability in the overall system. The cloud service provides need means to mitigate and manage such variability if they are to support guaranteed quality of service for the users. Studying and quantifying such variability serves as the first step in this direction. We have identified the problem of variability and how it impacts overall performance, as well as discussed ways in which the information can be used to design better and efficient cloud services. In our future work, we aim to exploit such information in designing cloud resource management solutions.

Acknowledgments: This work was sponsored in part by the NSF under CNS-1405697 and CNS-1422788 grants.

REFERENCES

[1] Amazon ec2 instances. https://aws.amazon.com/ec2/instance-types/.

[2] ChaosMonkey application description. https://github.com/Netflix/SimianArmy/wiki/Chaos-Monkey.

[3] Digital ocean. https://www.digitalocean.com/.

[4] Hp cloud. http://www.hpcloud.com/.

[5] Live Benchmarks from the Cloud. http://www.cloudlook.com/.

[6] Rackspace cloud. https://www.rackspace.com/en-us/calculator.

[7] The Secret Guide to Cloud Performance. http://www.slideshare.net/gidgreen/cloudlook-secret-guide-to-cloud-performance.

[8] Windows azure. https://azure.microsoft.com/en-us/.

[9] S. Angel, H. Ballani, T. Karagiannis, G. O'Shea, and E. Thereska. End-to-end performance isolation through virtual datacenters. In *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, pages 233–248, Broomfield, CO, Oct. 2014. USENIX Association.

[10] A. Anwar, Y. Cheng, A. Gupta, and A. R. Butt. Taming the cloud object storage with mos. In *Proceedings of the 10th Parallel Data Storage Workshop*, PDSW '15, pages 7–12, New York, NY, USA, 2015. ACM.

[11] A. Anwar, K. Krish, and A. R. Butt. On the use of microservers in supporting hadoop applications. In *Cluster Computing (CLUSTER), 2014 IEEE International Conference on*, pages 66–74. IEEE, 2014.

[12] A. Anwar, A. Sailer, A. Kochut, and A. R. Butt. Anatomy of cloud monitoring and metering: A case study and open problems. In *Proceedings of the 6th Asia-Pacific Workshop on Systems*, page 6. ACM, 2015.

[13] A. Anwar, A. Sailer, A. Kochut, C. O. Schulz, A. Segal, and A. R. Butt. Cost-aware cloud metering with scalable service management infrastructure. In *Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on*, pages 285–292. IEEE, 2015.

[14] A. Anwar, A. Sailer, A. Kochut, C. O. Schulz, A. Segal, and A. R. Butt. Scalable metering for an affordable it cloud service management. In *Cloud Engineering (IC2E), 2015 IEEE International Conference on*, pages 207–212. IEEE, 2015.

[15] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, et al. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.

[16] J.-Y. Audibert, R. Munos, and C. Szepesvári. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009.

[17] K. Bowman, J. W. Tschanz, S.-L. L. Lu, P. Aseron, M. M. Khellah, A. Raychowdhury, B. M. Geuskens, C. Tokunaga, C. B. Wilkerson, T. Karnik, et al. A 45 nm resilient microprocessor core for dynamic variation tolerance. *Solid-State Circuits, IEEE Journal of*, 46(1):194–208, 2011.

[18] Y. Chen, A. S. Ganapathi, R. Griffith, and R. H. Katz. Towards understanding cloud performance tradeoffs using statistical workload analysis and replay. *University of California at Berkeley, Technical Report No. UCB/EECS-2010-81*, 2010.

[19] Y. Cheng, A. Gupta, and A. R. Butt. An in-memory object caching framework with adaptive load balancing. In *Proceedings of the Tenth European Conference on Computer Systems*, EuroSys '15, pages 4:1–4:16, New York, NY, USA, 2015. ACM.

[20] Y. Cheng, M. S. Iqbal, A. Gupta, and A. R. Butt. Cast: Tiering storage for data analytics in the cloud. In *Proceedings of the 24th International Symposium on High-Performance Parallel and Distributed Computing*, HPDC '15, pages 45–56, New York, NY, USA, 2015. ACM.

[21] Y. Cheng, M. S. Iqbal, A. Gupta, and A. R. Butt. Pricing games for hybrid object stores in the cloud: Provider vs. tenant. In *7th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 15)*, Santa Clara, CA, July 2015. USENIX Association.

[22] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[23] J. Gu, J. Hu, T. Zhao, and G. Sun. A new resource scheduling strategy based on genetic algorithm in cloud computing environment. *Journal of Computers*, 7(1):42–52, 2012.

[24] T. Hoefler and R. Belli. Scientific benchmarking of parallel computing systems. In *Proceedings of the 2015 ACM/IEEE Conference on Supercomputing*, SC '15, New York, NY, USA, 2015. ACM.

[25] J. Hu, J. Gu, G. Sun, and T. Zhao. A scheduling strategy on load balancing of virtual machine resources in cloud computing environment. In *Parallel Architectures, Algorithms and Programming (PAAP), 2010 Third International Symposium on*, pages 89–96. IEEE, 2010.

[26] A. Iosup, N. Yigitbasi, and D. Epema. On the performance variability of production cloud services. In *Cluster, Cloud and Grid Computing (CCGrid), 2011 11th IEEE/ACM International Symposium on*, pages 104–113. IEEE, 2011.

[27] R. Kapoor, G. Porter, M. Tewari, G. M. Voelker, and A. Vahdat. Chronos: Predictable low latency for data center applications. In *Proceedings of the Third ACM Symposium on Cloud Computing*, SoCC '12, pages 9:1–9:14, New York, NY, USA, 2012. ACM.

[28] D. Kossmann, T. Kraska, and S. Loesing. An evaluation of alternative architectures for transaction processing in the cloud. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 579–590. ACM, 2010.

[29] W. T. Kramer and C. Ryan. *Performance variability of highly parallel architectures*. Springer, 2003.

[30] K. Krish, A. Anwar, and A. Butt. hats: A heterogeneity-aware tiered storage for hadoop. In *Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on*, pages 502–511, May 2014.

[31] K. Krish, A. Anwar, and A. R. Butt. [phi] sched: A heterogeneity-aware hadoop workflow scheduler. In *Modelling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2014 IEEE 22nd International Symposium on*, pages 255–264. IEEE, 2014.

[32] Z. Lee, Y. Wang, and W. Zhou. A dynamic priority scheduling algorithm on service request scheduling in cloud computing. In *Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on*, volume 9, pages 4665–4669. IEEE, 2011.

[33] B. Li, A. M. Song, and J. Song. A distributed qos-constraint task scheduling scheme in cloud computing environment: model and algorithm. *AISS: Advances in Information Sciences and Service Sciences*, 4(5):283–291, 2012.

[34] M. Melkonian. Get by without an rtos. *Embedded Systems Programming*, 13(10), 2000.

[35] R. Mraz. Reducing the variance of point to point transfers in the ibm 9076 parallel computer. In *Proceedings of the 1994 ACM/IEEE conference on Supercomputing*, pages 620–629. IEEE Computer Society Press, 1994.

[36] B. Palanisamy, A. Singh, L. Liu, and B. Jain. Purlieus: locality-aware resource allocation for mapreduce in a cloud. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, page 58. ACM, 2011.

[37] R. Ricci, G. Wong, L. Stoller, K. Webb, J. Duerig, K. Downie, and M. Hibler. Apt: A platform for repeatable research in computer science. *ACM SIGOPS Operating Systems Review*, 49(1):100–107, 2015.

[38] S. Rostedt and D. V. Hart. Internals of the rt patch. In *Proceedings of the Linux symposium*, volume 2, pages 161–172. Citeseer, 2007.

[39] J. Schad, J. Dittrich, and J.-A. Quiané-Ruiz. Runtime measurements in the cloud: observing, analyzing, and reducing variance. *Proceedings of the VLDB Endowment*, 3(1-2):460–471, 2010.

[40] J. Shalf, S. Dosanjh, and J. Morrison. Exascale computing technology challenges. In *Proceedings of the 9th International Conference on High Performance Computing for Computational Science*, VECPAR'10, pages 1–25, Berlin, Heidelberg, 2011. Springer-Verlag.

[41] D. Shue, M. J. Freedman, and A. Shaikh. Performance isolation and fairness for multi-tenant cloud storage. In *OSDI*, volume 12, pages 349–362, 2012.

[42] D. Skinner and W. Kramer. Understanding the causes of performance variability in hpc workloads. In *Workload Characterization Symposium, 2005. Proceedings of the IEEE International*, pages 137–149. IEEE, 2005.

[43] W. Tian, Y. Zhao, Y. Zhong, M. Xu, and C. Jing. A dynamic and integrated load-balancing scheduling algorithm for cloud datacenters. In *Cloud Computing and Intelligence Systems (CCIS), 2011 IEEE International Conference on*, pages 311–315. IEEE, 2011.

[44] J. Vitek and T. Kalibera. Repeatability, reproducibility, and rigor in systems research. In *Proceedings of the ninth ACM international conference on Embedded software*, pages 33–38. ACM, 2011.

[45] J. Wang, P. Varman, and C. Xie. Avoiding performance fluctuation in cloud storage. In *High Performance Computing (HiPC), 2010 International Conference on*, pages 1–9. IEEE, 2010.

[46] J.-z. Wang, P. Varman, and C.-s. Xie. Optimizing storage performance in public cloud platforms. *Journal of Zhejiang University SCIENCE C*, 12(12):951–964, 2011.

[47] J. Xie, S. Yin, X. Ruan, Z. Ding, Y. Tian, J. Majors, A. Manzanares, and X. Qin. Improving mapreduce performance through data placement in heterogeneous hadoop clusters. In *Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on*, pages 1–9. IEEE, 2010.

[48] M. Zaharia, A. Konwinski, A. D. Joseph, R. H. Katz, and I. Stoica. Improving mapreduce performance in heterogeneous environments. In *OSDI*, volume 8, page 7, 2008.