Research Statement:
Rethinking Cloud Computing For Big Data, Machine Learning, and IoT Workloads

Prateek Sharma
University of Massachusetts Amherst

Emerging applications such as large-scale distributed data processing, machine learning, and Internet of Things, are fueled by the computational resources provided by large scale distributed computing platforms. Cloud computing has emerged as a popular platform to run these emerging workloads. While early cloud platforms were used to mainly run web workloads, today it has become commonplace to use cloud platforms to instantiate large clusters comprising of hundreds of servers to process large amounts of data for big data, machine learning, and IoT applications. As a result, today's cloud platforms run ever more complex applications with diverse requirements, resulting in new challenges in efficient use of cloud resources—both from an application and system design perspective.

My research focuses on designing systems and abstractions that make large scale distributed computing platforms more efficient, as well enable applications to effectively harness their resources. While I have broad interests in large distributed systems, my current research looks at the intersection between cloud computing and emerging applications in data processing, machine learning, and IoT. My research vision has centered around exploiting the synergy between large scale distributed systems and data-driven applications and techniques. My thesis research has focused on rethinking cloud platforms and mechanisms to better support data processing, machine learning, and IoT applications, and rethinking how these emerging applications can use cloud platforms more effectively.

In addition to designing new resource management mechanisms and policies, my work has drawn upon techniques from data science and economics to improve system design. As an experimental computer scientist, my research involves designing, building, optimizing, and evaluating software artifacts to understand the behavior of large scale systems. At the same time, I have also used principled approaches such as analytical models, optimization, and machine learning techniques, for understanding and optimizing systems I have built.

1 Transient Computing For Emerging Distributed Applications

A popular use of cloud servers is to run large scale parallel distributed data intensive applications for machine learning, data analytics, deep learning, and scientific computing workloads. Recently, cloud operators have started offering low-cost *transient* servers that can be unilaterally revoked and preempted. Transient servers allow the cloud operators to sell spare capacity at discounted prices (as low as 1/10th the price of conventional servers) to maximize revenue. However, running modern distributed applications on transient computing resources raises a slew of new challenges. Most applications are designed and built with the implicit assumption that its computing resources will continue to be available until relinquished. Transient server revocations can cause loss of application-state, which can result in application downtimes, degraded performance due to failure-recovery, and end-user dissatisfaction in general.

Transiency has a number of unique characteristics that differentiates it from conventional fault tolerance. First, transient servers can become unavailable at a much higher rate than conventional servers, since their availability is controlled by the operational policies of the data-center or the cloud. Second, transient server unavailability is often preceded by an advance warning. And third, transient servers may be available with different cost/availability tradeoffs, requiring the use of

careful server selection using data-driven techniques. One of the thrusts of my thesis research is to design systems that use the unique characteristics of transient servers to ameliorate the effects of server revocations for a wide range of applications.

Using Virtualization and Migration to Mitigate Transiency. One approach to deal with server revocations is to migrate the application and its data to other cloud servers and resume computation. The key challenge is to migrate the state in limited time window, and avoid losing any memory and disk state. Through the use of virtualization techniques such as nested virtualization and bounded-time live-migration, my research [EuroSys '15] has shown that even interactive applications (such as web servers and databases), can run with almost continuous availability on revocable transient cloud servers. Through the use of system-level techniques, we showed that transient servers can be used for *any* application, including interactive ones.

Diversification to Mitigate Transiency. An effective approach to mitigate the impact of revocation is to use the principle of diversification from economics and finance. Running distributed applications on a diverse collection of transient servers, such that not all servers are concurrently revoked, allows applications to continue running in degraded mode. However, due to the large number of different server markets (thousands), and their different prices and availabilities, prior work relied on ad-hoc approaches for server diversification.

We address this problem by modeling the risk of server revocations using their price histories, and construct an "optimal" mix of servers that minimizes both cost and risk of revocation [SIGMETRICS '17, HotCloud '16]. Server portfolios are inspired by (and based on) financial portfolios, which enable investors to methodically construct a financial portfolio from a large number of underlying assets with various risks and rewards. Portfolios enable us to systematically tailor resource allocation to an application's risk and reward tolerances, and have been used to build a practical, general-purpose system that enables transiency-support for multiple application types.

Distributed Data Processing on Transient Servers. Virtualization, migration, and diversification are system level techniques for mitigating transiency. My research has also explored how distributed applications can be made *transiency-aware*, by handling transiency at the application level. Many emerging use-cases now require large scale data processing and machine learning workloads to return results in a timely manner. While cloud transient servers can provide low-cost computation, their revocable nature can sharply increase the running times of big-data workloads. To address this challenge, my research [EuroSys '16] incorporates fault-tolerance policies into the Spark data-processing framework, by intelligently checkpointing intermediate application state, i.e., Spark's resilient distributed datasets. Incorporating transiency-awareness into applications can greatly reduce costs (by up to $10\times$), and improve performance for latency-sensitive in-memory distributed data analytics and machine learning workloads.

Resource Deflation: A New Abstraction for Transient Computing. Traditional cloud platforms and data centers use server revocations to reclaim resources, which can disrupt applications and impose fault-tolerance overheads. My recent work [Submitted '17] proposes a new approach, called resource deflation, that shrinks resource allocations of applications using virtual machine overcommitment techniques. Resource deflation allows applications to gracefully degrade, instead of facing outright preemption. This further expands the use of transient resources to applications that cannot tolerate revocations, such as in-memory caches, latency-sensitive interactive services, and unmodified distributed data processing frameworks without built-in fault-tolerance.

Together, these components of my thesis research have fundamentally altered the tradeoff

imposed by transient resource availability—we have shown that it is indeed possible to mitigate the effects of revocation for many applications *and* run them on low-cost transient servers. They also exemplify my research philosophy of doing relevant systems research (reducing computation costs), using ideas from diverse fields such as classical fault-tolerance, data science, and economics.

2 Virtualization Techniques for Emerging Applications

Large-scale virtualized hosting infrastructures have become the de-facto computing platforms for enterpise and cloud workloads. Virtualization technologies are the foundation for utility-computing, and enable cloud platforms to offer infrastructure, platforms, and software as a service. The computational demands of emerging applications such as large scale low-latency distributed data processing has also resulted in the emergence of new virtualization technologies that have different tradeoffs in performance, security, and management functionality. My research examines tradeoffs in virtualization and proposes solutions to address key performance concerns.

Comparing Virtualization Techniques. Clouds and data centers can employ a number of different virtualization techniques. Conventionally, hardware virtualization techniques are used to create virtual machines. Other forms of virtualization such as operating system virtualization ("containers"), and unikernels, have recently emerged as viable abstractions. The different virtualization techniques have different characteristics and tradeoffs. My research [Middleware '16] seeks to understand these tradeoffs by empirically comparing the performance, isolation, and deployability characteristics for emerging applications.

In addition to providing the missing empirical foundation for comparing the existing virtualization techniques, my research also focuses on developing novel, *hybrid* virtualization techniques that combine both hardware and operating system virtualization. My research will continue to examine hybrid and other virtualization techniques which can provide secure low-overhead virtualization to increase application performance and data center efficiency.

Improving Resource Isolation in Multi-Tenant Environments. Cloud platforms multiplex server hardware resources to run multiple virtual machines, potentially running different applications and belonging to different customers. Thus providing performance isolation is of critical importance so that virtual machines do not starve each other of resources in multi-tenant environments. My research [COMSNETS '16] has identified new sources of implicit resource sharing in modern hypervisors (the hypervisor file-system cache), and proposes a novel page-cache design that carefully partitions memory among multiple VMs. This new design prevents VMs from trampling on each other, and improves the performance isolation among VMs. This ensures that an application's performance is not adversely affected by its neighbours, and is especially beneficial for data-intensive workloads, whose performance is strongly determined by their file-system cache allocation.

Reducing VM Memory Footprint to Increase Multiplexing. My research also looks at virtualization techniques for improving the effectiveness of server multiplexing. In particular, I have looked at how the memory footprint of applications can be reduced by sharing duplicate memory pages between virtual machines and hypervisors [HPDC '12]. A novel aspect of this work is that it was the first system to identify the double-caching problem in VMs, and proposed a black-box exclusive-caching solution to it. This improves application performance, especially for memory and disk intensive workloads such as data-processing applications.

3 Future Research Directions

While my current research has made an initial stab at designing large scale distributed systems for emerging applications, many important research challenges remain to be addressed. These are very exciting times for systems research. New and exciting applications are emerging, such as IoT, connected cars, augmented and virtual reality, etc., in addition to new AI workloads for computer vision, language processing, and robotics, will all require distributed computing to varying degrees.

In the near term, my research will examine the interaction between distributed deep learning applications and transient servers. Transient servers can provide the massive amounts of computational resources at low cost required by deep learning pipelines for tasks such as image and speech recognition. However, the effect of transiency on deep learning workloads that use hardware accelerators (such as GPUs), is interesting, and needs further exploration. Distributed deep-learning frameworks (such as Tensorflow), expose tradeoffs in cost, performance, and accuracy, and I am particularly interested in understanding and exploiting this three-way tradeoff. Specifically, my research will focus on two main areas:

Cloud Architectures for Real-time and Latency-Sensitive Workloads. Applications such as IoT, connected cars, and augmented reality, impose low latency and real-time constraints. The latency, locality, and privacy requirements of these emerging workloads cannot be sufficiently met with current cloud architectures, and new "edge cloud" architectures have been proposed that move computation and data near the edge of the network closer to the users. Due to their highly distributed and resource-constrained nature, edge clouds pose a number of resource management challenges. To support highly dynamic workloads, my recent work [SEC '17] looks at efficiently and transparently migrating virtual machines between edge cloud locations. Going into the future, I intend to expand on this work, and develop systems using migration and computational offloading to improve end-user latency and cloud load balancing.

Operating and Distributed Systems for Emerging Hardware. In addition to changing workloads, we are also seeing emergence of new hardware technologies that require us to rethink many assumptions about operating system design. The emergence of faster networking (100G ethernet and RDMA), storage (persistent memory), and hardware accelerators (GPUs, FPGAs, and ASICs)—has introduced many critical design challenges. The new performance landscape necessitates new design principles and abstractions for operating systems and other systems software, as decades old assumptions need to be re-evaluated. My broad systems background in operating systems, virtualization, resource management, and networking, makes me well placed to handle these challenges. I am especially interested in developing and refining operating systems abstractions that can enable high-performance computing, yet at the same time are expressive and rich enough to meet the needs of both existing and emerging applications.

I shall also explore system designs in which debuggability and observability are first-class design principles, rather than afterthoughts. My research will examine how systems can be dynamically instrumented intelligently by leveraging reinforcement learning techniques, to detect performance anomalies in highly dynamic production environments. With the emergence of new software architectures (edge clouds and new operating system designs), we need "computational microscopes" that help us understand computational processes in detail.

My long term research agenda will focus on the end-to-end design and performance of large scale distributed systems. With the emergence of radically new applications and new hardware technology, we face multiple deep technical transitions and challenges. Addressing these challenges

will require an inter-disciplinary approach, and I intend to collaborate with people in other subfields like AI, architecture, databases, programming languages, etc, as well as application domains such as biology, medicine, and economics, that can benefit from the infusion of systems software and systems thinking. The constant evolution of new applications, computation environments, and ecosystems introduces critical systems research challenges, and my research will continue to address relevant systems problems in novel ways.

4 References

[EuroSys '15] **Prateek Sharma**, Stephen Lee, Tian Guo, David Irwin, and Prashant Shenoy. Spotcheck: Designing a Derivative IaaS Cloud on the Spot Market. In *Proceedings of the Tenth European Conference on Computer Systems (EuroSys)*, pages 16:1–16:15. ACM, 2015

[SIGMETRICS '17] **Prateek Sharma**, David Irwin, and Prashant Shenoy. Portfolio-driven Resource Management for Transient Cloud Servers. In *Proceedings of ACM on Measurement and Analysis of Computing Systems (SIGMETRICS)*, volume 1, pages 5:1–5:23. ACM, June 2017

[HotCloud '16] **Prateek Sharma**, David Irwin, and Prashant Shenoy. How Not to Bid the Cloud. In *Proceedings of the 8th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud)*. USENIX, June 2016

[EuroSys '16] **Prateek Sharma**, Tian Guo, Xin He, David Irwin, and Prashant Shenoy. Flint: Batch-Interactive Data-Intensive Processing on Transient Servers. In *Proceedings of the Eleventh European Conference on Computer Systems (EuroSys)*, pages 6:1–6:15. ACM, 2016

[Submitted '17] **Prateek Sharma**, Ahmed Ali-Eldin, and Prashant Shenoy. Resource Deflation: A New Abstraction For Transient Computing.

[Middleware '16] **Prateek Sharma**, Lucas Chaufournier, Prashant Shenoy, and Y. C. Tay. Containers and Virtual Machines at Scale: A Comparative Study. In *Proceedings of the 17th International Middleware Conference*, pages 1:1–1:13. ACM, 2016

[COMSNETS '16] **Prateek Sharma**, Purushottam Kulkarni, and Prashant Shenoy. Per-VM Page Cache Partitioning for Cloud Computing Platforms. In *2016 8th International Conference on Communication Systems and Networks (COMSNETS)*, pages 1–8, Jan 2016

[HPDC '12] **Prateek Sharma** and Purushottam Kulkarni. Singleton: System-wide Page Deduplication in Virtual Environments. In *Proceedings of the 21st international symposium on High-Performance Parallel and Distributed Computing (HPDC)*, pages 15–26. ACM, 2012

[SEC '17] Lucas Chaufournier, **Prateek Sharma**, Franck Le, Erich Nahum, Prashant Shenoy, and Don Towsley. Fast transparent virtual machine migration in distributed edge clouds. In *Proceedings of the second IEEE/ACM Symposium on Edge Computing*, page 12, October 2017