Shoot4U: Using VMM Assists to Optimize TLB Operations on Preempted vCPUs

Jiannan Ouyang, John Lange
University of Pittsburgh

Haoqiang Zheng
VMware Inc.

VEE'16
04/02/2016

CPU Consolidation in the Cloud

CPU Consolidation: multiple virtual CPUs (vCPUs) share the same physical CPU (pCPU).

Motivation: Improve datacenter utilization.

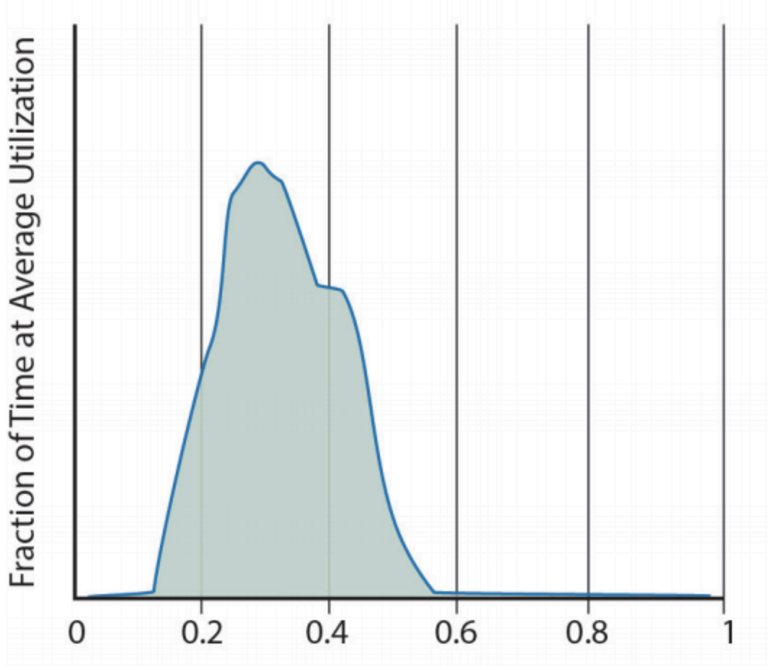
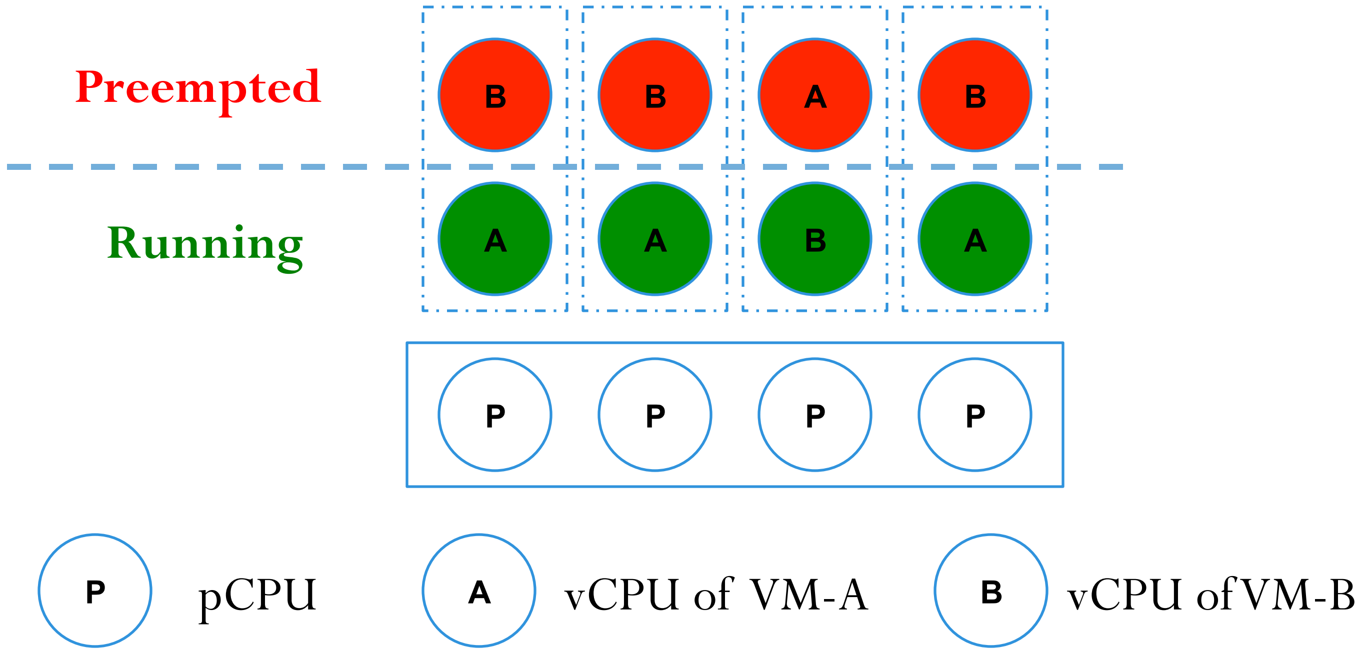


Figure 1. Average activity distribution of a typical shared Google clusters including Online Services, each containing over 20,000 servers, over a period of 3 months [Barroso 13].

Problems with Preempted vCPUs



Performance problems:

Busy-waiting based kernel synchronization operations

- Lock *Holder* Preemption problem
- Lock *Waiter* Preemption problem
- *TLB Shootdown* Preemption problem

Lock Holder Preemption

Lock holder preemption [Uhlig 04, Friebel 08]

- A preempted vCPU is holding a spinlock
- Causes dramatically longer lock waiting time
  - context switch latency + CPU shares allocated to other vCPUs

Scheduling Techniques

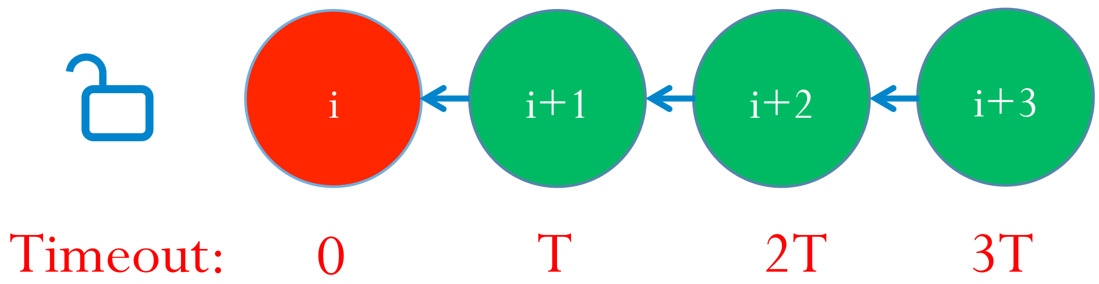
- co-scheduling, relaxed co-scheduling [VMware 10]
- Adaptive co-scheduling [Weng HPDC11]
- Balanced scheduling [Sukwong EuroSys11]
- Demand-based coordinated scheduling [Kim ASPLOS13]

Hardware Assisted Techniques

- Intel Pause-Loop Exiting (PLE) [Riel 11]

Lock Waiter Preemption [Ouyang VEE13]

Linux uses a FIFO order fair spinlock, named *ticket spinlock*



Lock waiter preemption

- A lock waiter is preempted, and blocks the queue
- P(waiter preemption) > P(holder preemption)

Preemptable Ticket Spinlock

- **Key idea: proportional timeout**

TLB Shootdown Preemption

KVM Paravirt Remote Flush TLB [kvmtlb 12]

- VMM maintains vCPU preemption states and shares with the guest.
- Use conventional approach if the remote vCPU is running.
- Defer TLB flush if the remote vCPU is preempted.
- Cons: preemption state may change after checking.

TLB shootdown IPIs as scheduling heuristics [Kim ASPLOS13]

Shoot4U

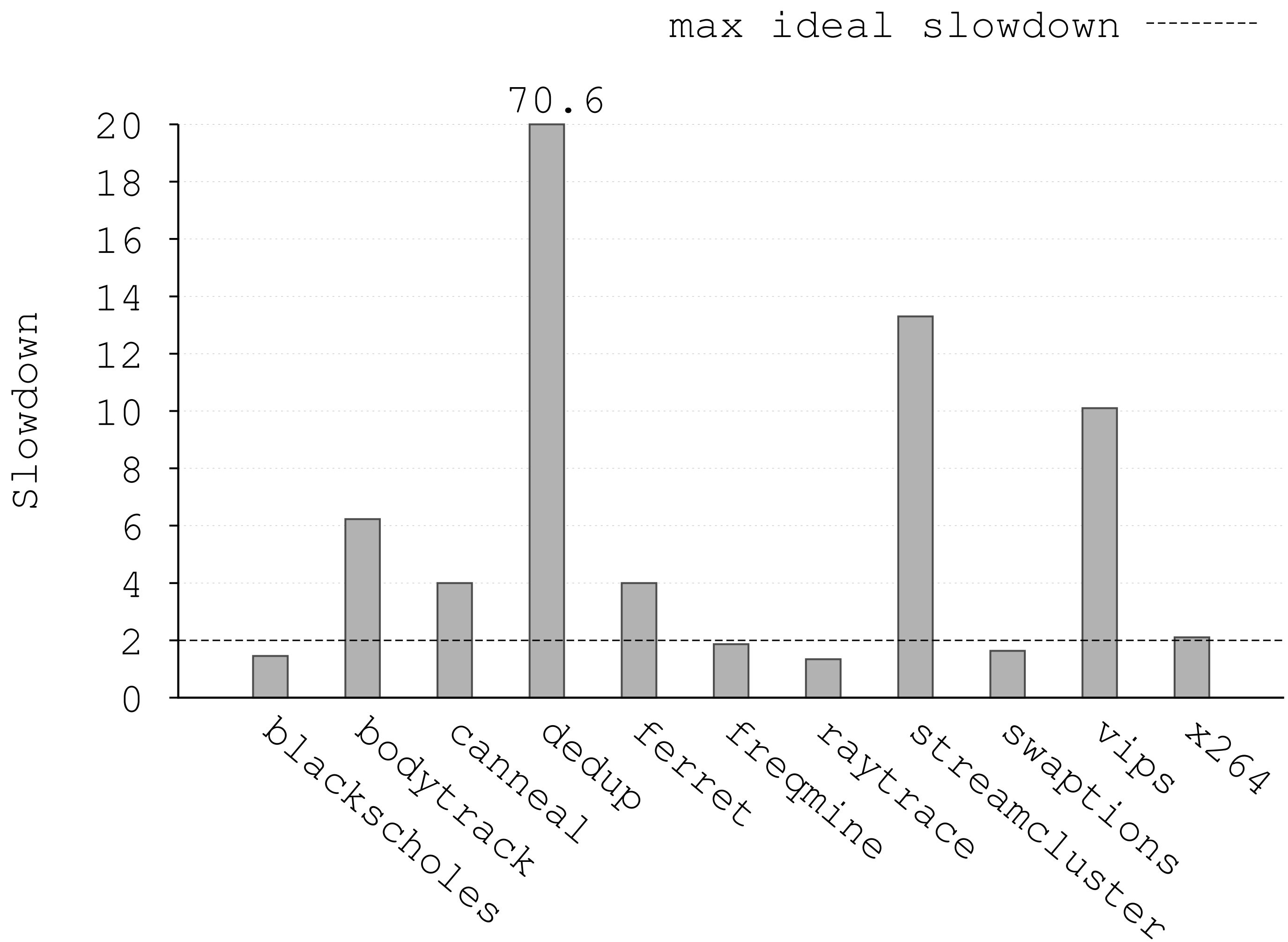
- Goal: eliminate the problem
- Key idea: invalidate guest TLB entries from the VMM

Contributions

- An analysis of the impact that various low level synchronization operations have on system benchmark performance.
- Shoot4U: A novel virtualized TLB architecture that ensures consistently low latencies for synchronized TLB operations.
- An evaluation of the performance benefits achieved by Shoot4U over current state-of-art software and hardware assisted approaches.

Performance Analysis

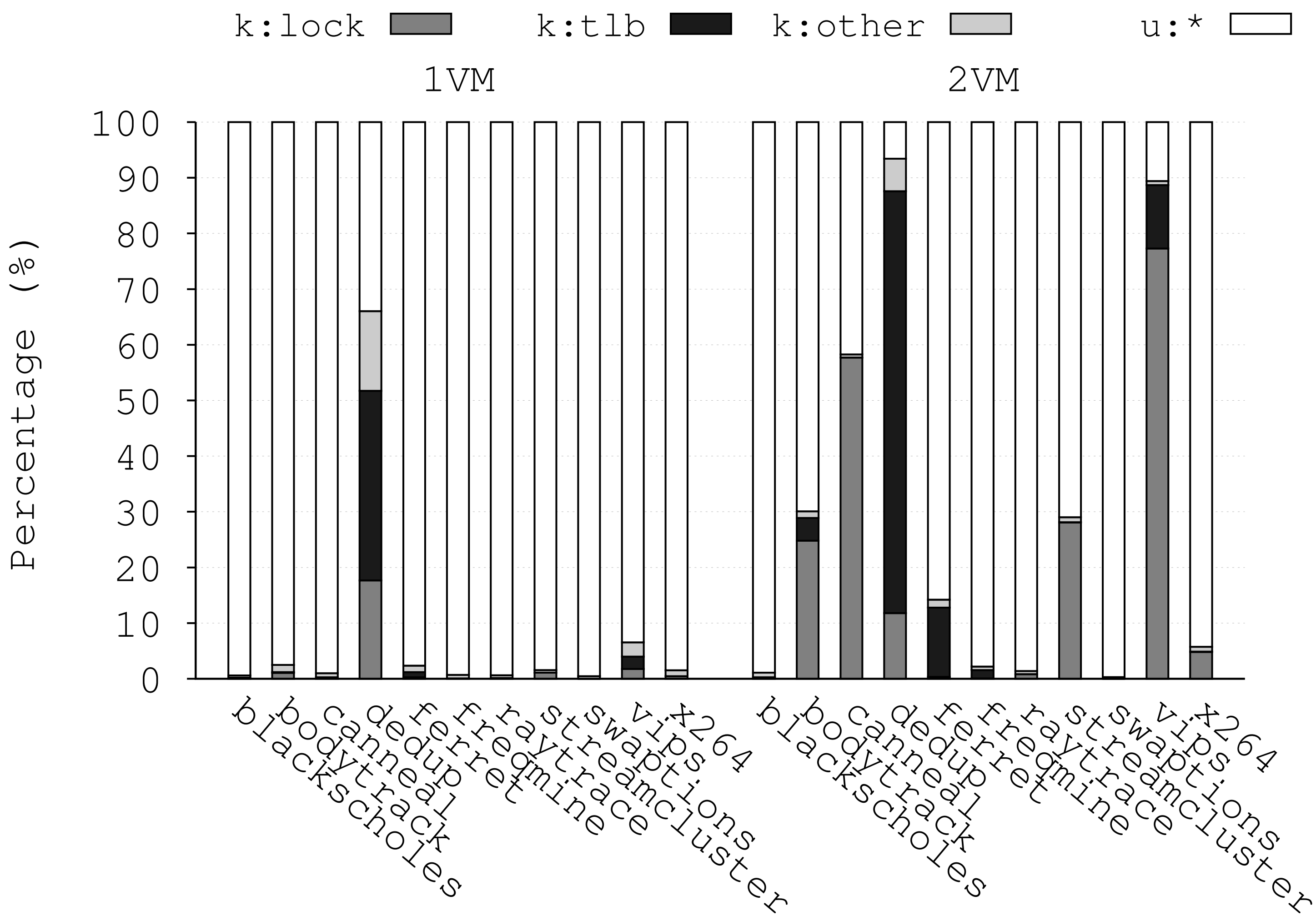
Overhead of CPU Consolidation



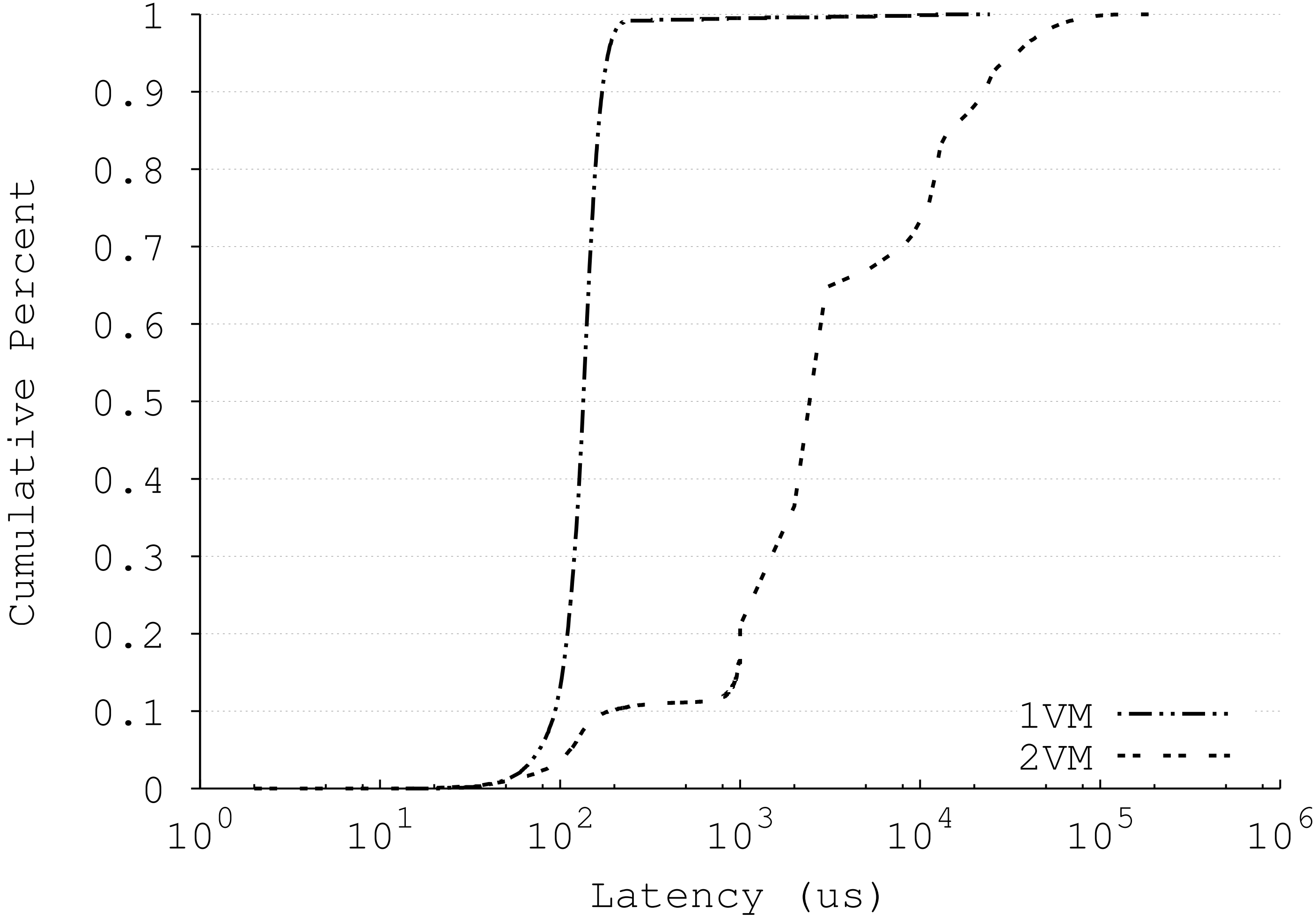
PARSEC Runtime with co-located VM over running alone

(12-core VMs, measured on Linux/KVM, with PLE disabled)

CPU Usage Profiling (perf)

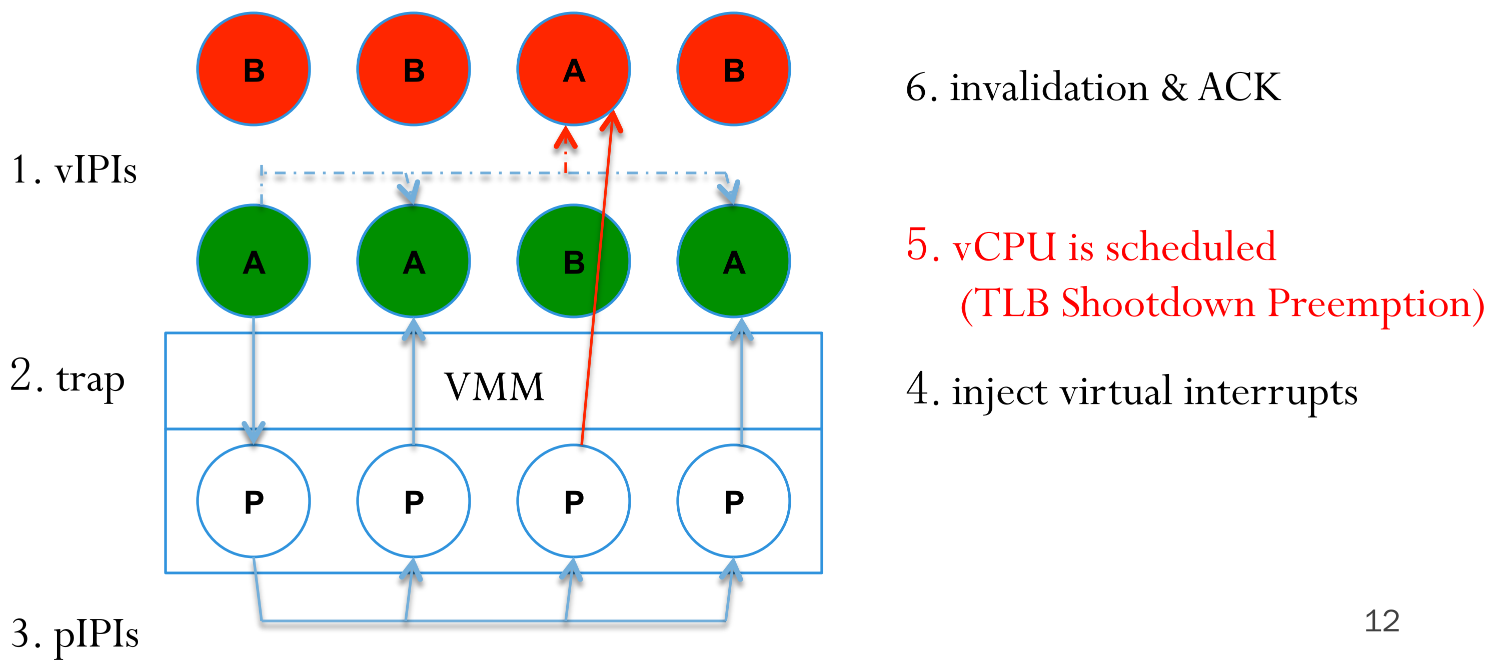


CDF of TLB Shootdown Latency (ktap)



How TLB Shootdown Works in VMs

- TLB (Translation Lookaside Buffer)
  - a per-core hardware cache for page table translation results
- TLB coherence is managed by the OS
  - TLB shootdown operations: IPI + invlpg
- Linux TLB shootdown is busy-waiting based



Shoot4U

Shoot4U

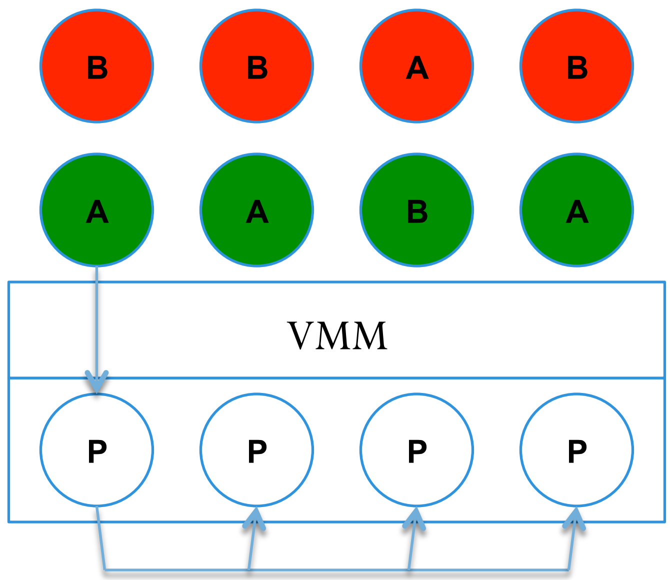
Observation: modern hardware allows the VMM to invalidate guest TLB entries (e.g. Intel *invpid*)

Key idea: invalidate guest TLB entries from the VMM

- Tell the VMM what TLB entries and vCPUs to invalidate (hypervall)
- The VMM invalidates and returns, no interrupt injection and waiting

1. hypercall
<vcpu set, addr range>

2. pIPIs



3. invalidation and ACK

Implementation

Shoot4U API

```
kvm_hypercall3(unsigned long KVM_HC_SHOOT4U, unsigned long vcpu_bitmap,
    unsigned long start_addr, unsigned long end_addr);
```

KVM/Linux 3.16, ~200 LOC (~50 LOC guest side)

- https://github.com/ouyangjn/shoot4u

Guest

- use hypercall for TLB shootdowns

VMM

- hypercall handler: vCPU set => pCPU set, and send IPIs
- IPI handler: invalidate guest TLB entries with invpid

Evaluation

Dual-socket Dell R450 server

- 6-core Intel "Ivy-Bridge" Xeon processors with hyperthreading
- 24 GB RAM split across two NUMA domains.
- CentOS 7 (Linux 3.16)

Virtual Machines

- 12 vCPUs, 4G RAM on the same socket
- Fedora 19 (Linux 4.0)
- VM1: PARSEC Benchmark Suite, VM2 sysbench CPU test

Schemes

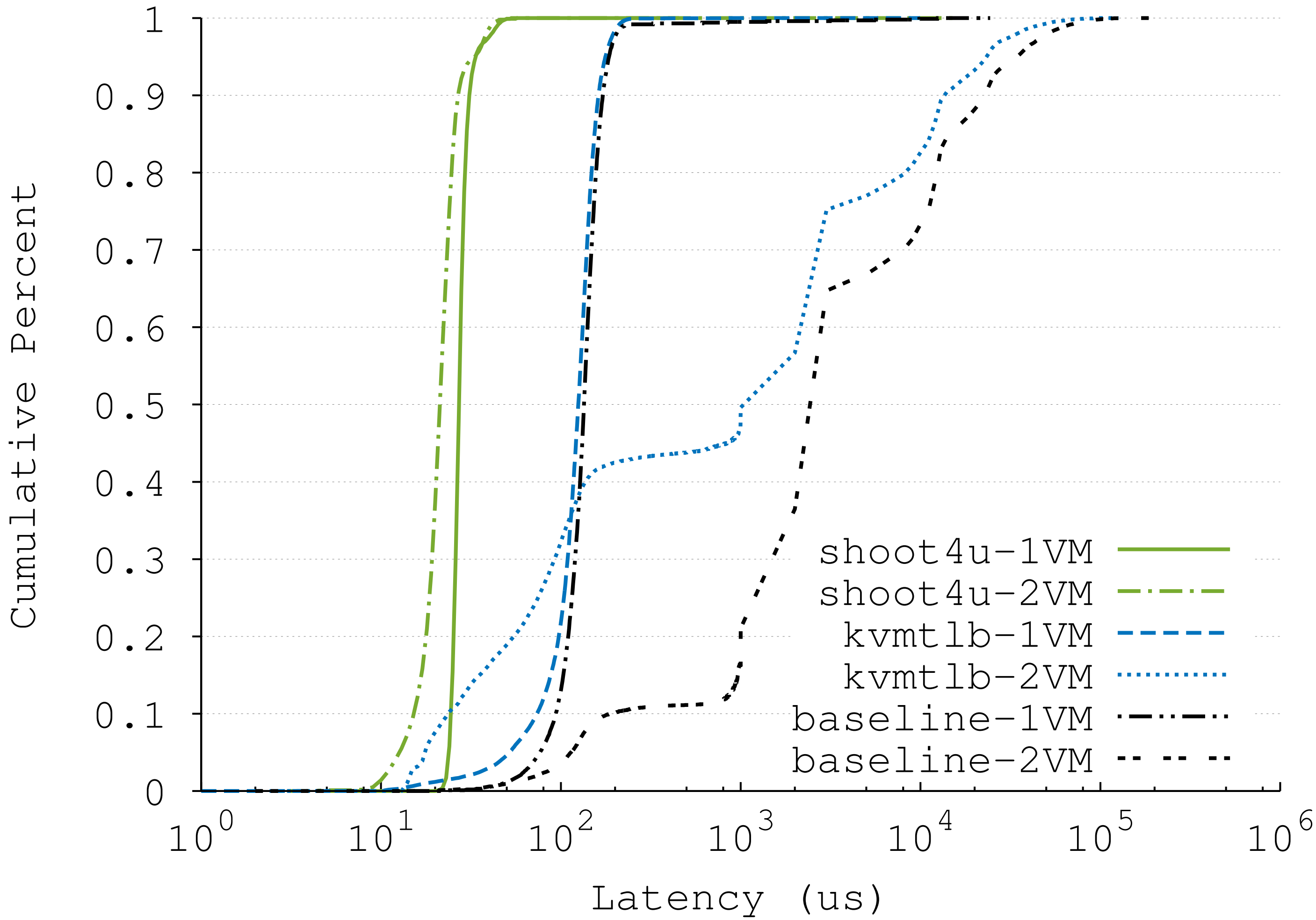
- baseline: unmodified Linux kernel
- kvmtlb [kvmtlb 12]
- Shoot4U
- Pause-Loop Exiting (PLE) [Riel 11]
- Preemptable Ticket Spinlock (PMT) [Ouyang VEE '13]

TLB Shootdown Latency (Cycles)

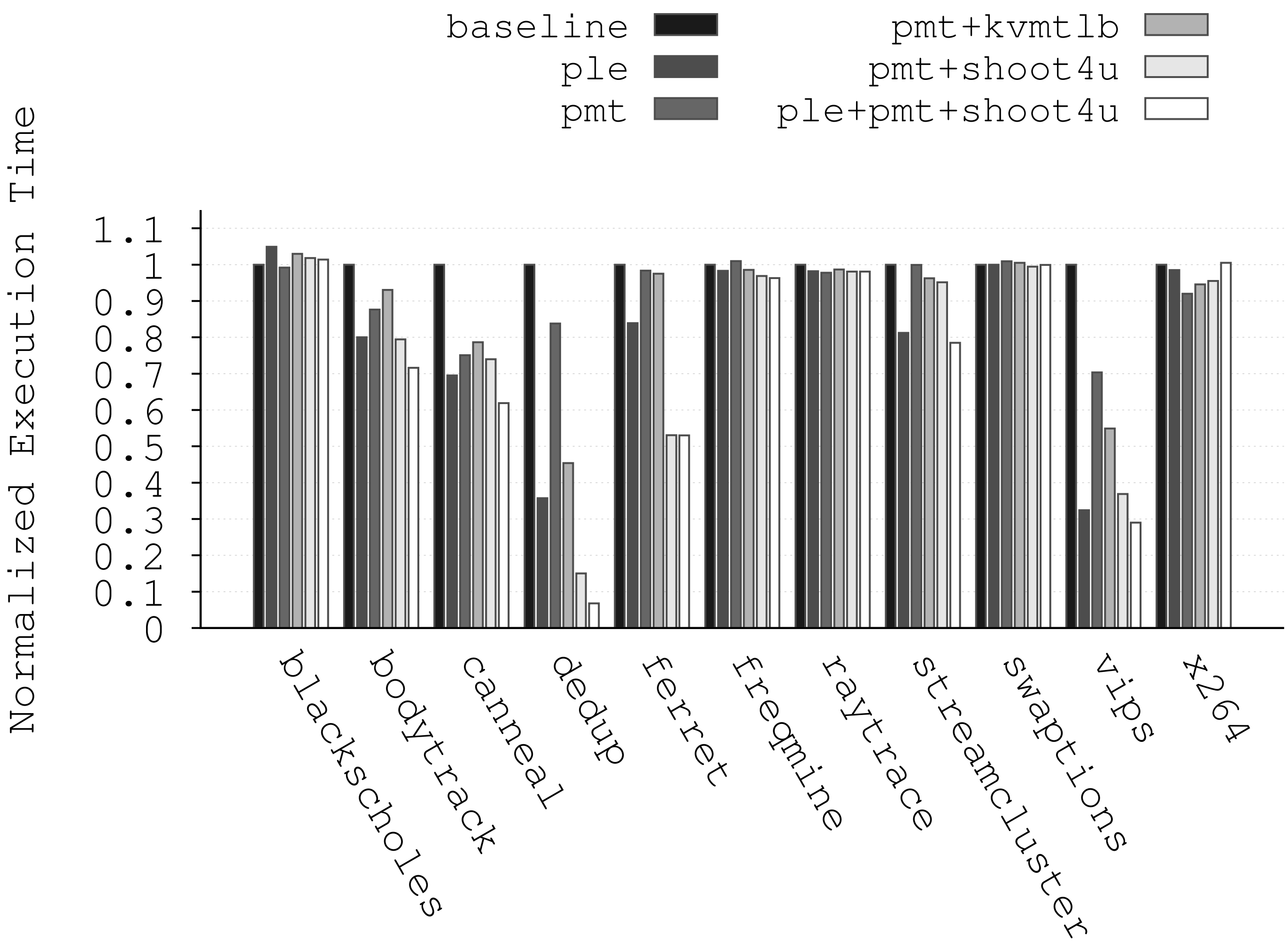
		baseline	kvmtlb	shoot4u
1VM	Mean	166	122	28
	Max	24,428	9,953	453
2VM	Mean	9,048	5,401	22
	Max	194,108	126,923	15,034

Order of magnitude lower latency

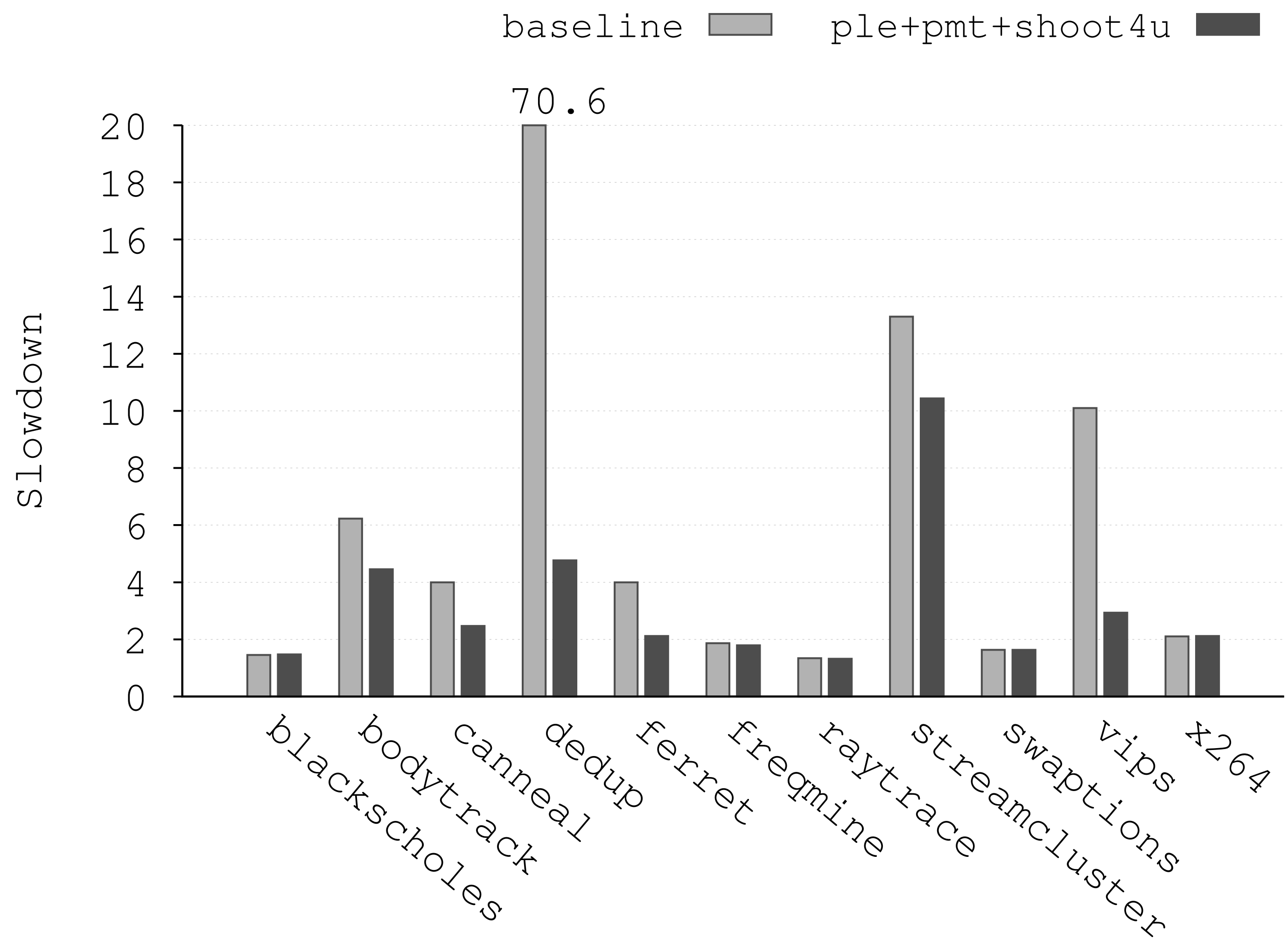
TLB Shootdown Latency (CDF)



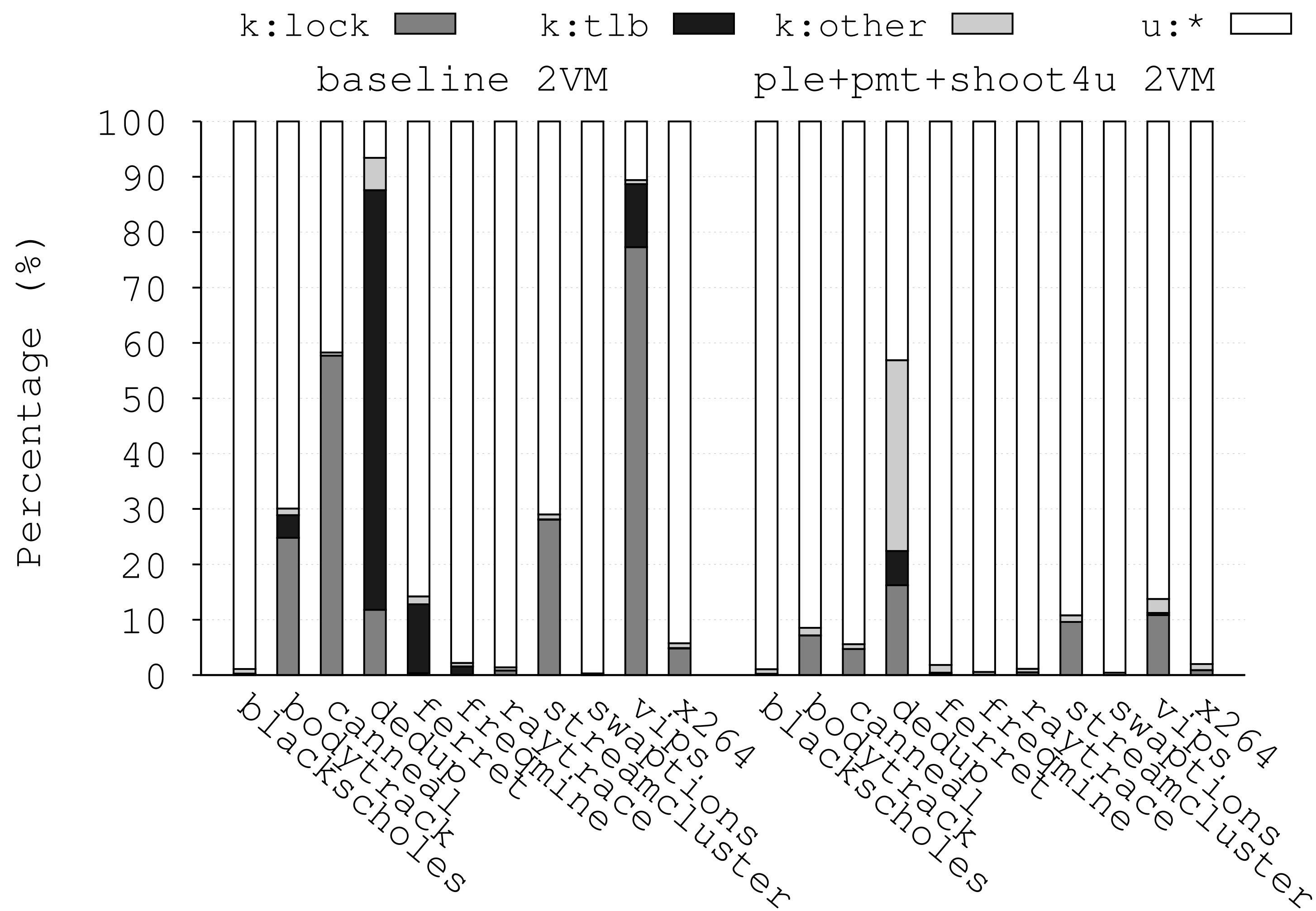
Parsec Performance (2-VMs)



Revisiting Performance Slowdown



Revisiting CPU Usage Profiling



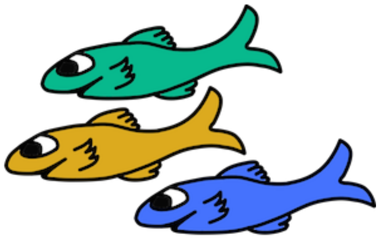
Conclusions

- We conducted a set of experiments in order to provide a breakdown of overheads caused by preempted virtual CPU cores, showing that TLB operations can have a significant impact on performance with certain workloads.
- We Shoot4U, an optimization for TLB shootdown operations that internalizes TLB shootdowns in the VMM and so no longer requires the involvement of a guest's vCPUs.
- Our evaluation demonstrates the effectiveness of our approach, and illustrates how under certain workloads our approach is dramatically better than state-of-the-art techniques.

https://github.com/ouyangjn/shoot4u

Q & A

Kitten Lightweight Kernel



Pisces Co-Kernel

Jiannan Ouyang

Ph.D. Candidate

University of Pittsburgh

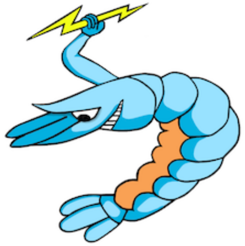
ouyang@cs.pitt.edu

http://www.cs.pitt.edu/~ouyang/

The Prognostic Lab

University of Pittsburgh

http://www.prognosticlab.org



Palacios VMM

References

- [Ouyang 13] Jiannan Ouyang and John R. Lange. Preemptable Ticket Spinlocks: Improving Consolidated Performance in the Cloud. In Proc. 9th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE), 2013.
- [Uhlig 04] Volkmar Uhlig, Joshua LeVasseur, Espen Skoglund, and Uwe Dannowski. Towards scalable multiprocessor virtual machines. In Proceedings of the 3rd conference on Virtual Machine Research And Technology Symposium - Volume 3, VM'04, 2004.
- [Friebel 08] Thomas Friebel. How to deal with lock-holder preemption. Presented at the Xen Summit North America, 2008.
- [Kim ASPLOS'13] H. Kim, S. Kim, J. Jeong, J. Lee, and S. Maeng. Demand- based Coordinated Scheduling for SMP VMs. In *Proc. Inter- national Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2013.

- [VMware 10] VMware(r) vSphere(tm): The cpu scheduler in vmware esx(r) 4.1. Technical report, VMware, Inc, 2010.
- [Barroso 13] L. A. Barroso, J. Clidaras, and U. Holzle. The Datacenter as a Computer: An Introduction to the Design of Warehouse- Scale Machines. *Synthesis Lectures on Computer Architec- ture*, 2013.
- [Weng HPDC'11] C. Weng, Q. Liu, L. Yu, and M. Li. Dynamic Adaptive Scheduling for Virtual Machines. In *Proc. 20th International Symposium on High Performance Parallel and Distributed Computing (HPDC)*, 2011.
- [Sukwong EuroSys'11] O. Sukwong and H. S. Kim. Is Co-scheduling Too Expensive for SMP VMs? In *Proc. 6th European Conference on Com- puter Systems (EuroSys)*, 2011.

- [Riel 11] R. v. Riel. Directed yield for pause loop exiting, 2011. URL http://lwn.net/Articles/424960/.
- [kvmtlb 12] KVM Paravirt Remote Flush TLB. https://lwn.net/Articles/500188/.