6.5810: Graduate-level Operating Systems

Adam Belay <abelay@mit.edu>

Welcome to 6.5810

- Theme for this semester: Operating systems for datacenters
- This is a **graduate-level**, seminar-style class
  - Includes both lectures and paper presentations by students
- Main focus: OS research
  - Will learn about recent research results
  - Will study datacenter systems developed by Amazon, Facebook, Google, and Microsoft

Background / Prerequisites

- Undergraduate OS course (at MIT this would be 6.S081, now 6.1810)
- Low-level hacking experience (e.g., assembly, C, C++, or Rust)
- Familiarity with reading, interpreting and presenting research papers (e.g., 6.033/6.UAT)

Which class to take?

6.1810

- Basic OS concepts; xv6 (a UNIX teaching OS) is used to illustrate ideas
- Lab assignments on xv6 and RISC-V; page tables, networking, scheduling, etc.

6.5810

- OS research seminar for graduate students and other students who have already taken 6.1810 or an equivalent course
- Research project, typically built on real systems like the Linux Kernel, DPDK, SPDK, etc.

Action items

- Sign up for piazza (see course website)
- Fill out the class survey by the end of the week.
  - Details will be posted on Piazza later today.
  - We will use these responses to help finalize the course schedule
- If you don’t have the right background, register for 6.1810 instead
  - Email us if you have questions

Logistics: Grading

- 10% Class Participation
- 20% Paper Summaries and Presentations
- 20% Lab Assignments
- 50% Final Project

Logistics: Participation

- Read every paper; be prepared to discuss in class
- Interactive lectures are encouraged! It's okay to ask questions anytime
- Submit a question you have about the paper on Piazza the night before

Logistics: Paper presentations

- Each student will present a reading assignment (very likely one paper) and help lead an in-class discussion
- We will meet with you a few days before to provide feedback on a draft of the presentation
- After presenting, your slides (or notes) will be posted online
- The course staff will present through the end of September (and other lectures)
- Sign up for paper presentations by 09/15

Logistics: Final Project

- Group projects encouraged but not required
- We will meet with you regularly to help refine your project
- Three written deliverables: Proposal (10/11), Draft (11/4), and Final Report (12/14)
- More project ideas will be posted soon
- Presentations at the end of class (12/12 + 12/14)
- It’s fine for the project to be related to your current graduate research, but must align with the theme of the course

Logistics: Lab assignments

- Closer to tutorials; intended to introduce tools for research projects

- This year:
  1. **DPDK:** A fast networking library developed by Intel
  2. **SPDK:** A fast storage library developed by Intel
  3. **Sandboxing:** System call filtering for the Linux Kernel

Logistics: Other research platforms

- This course will introduce opensource tools developed in our lab
- Potential for many interesting final projects

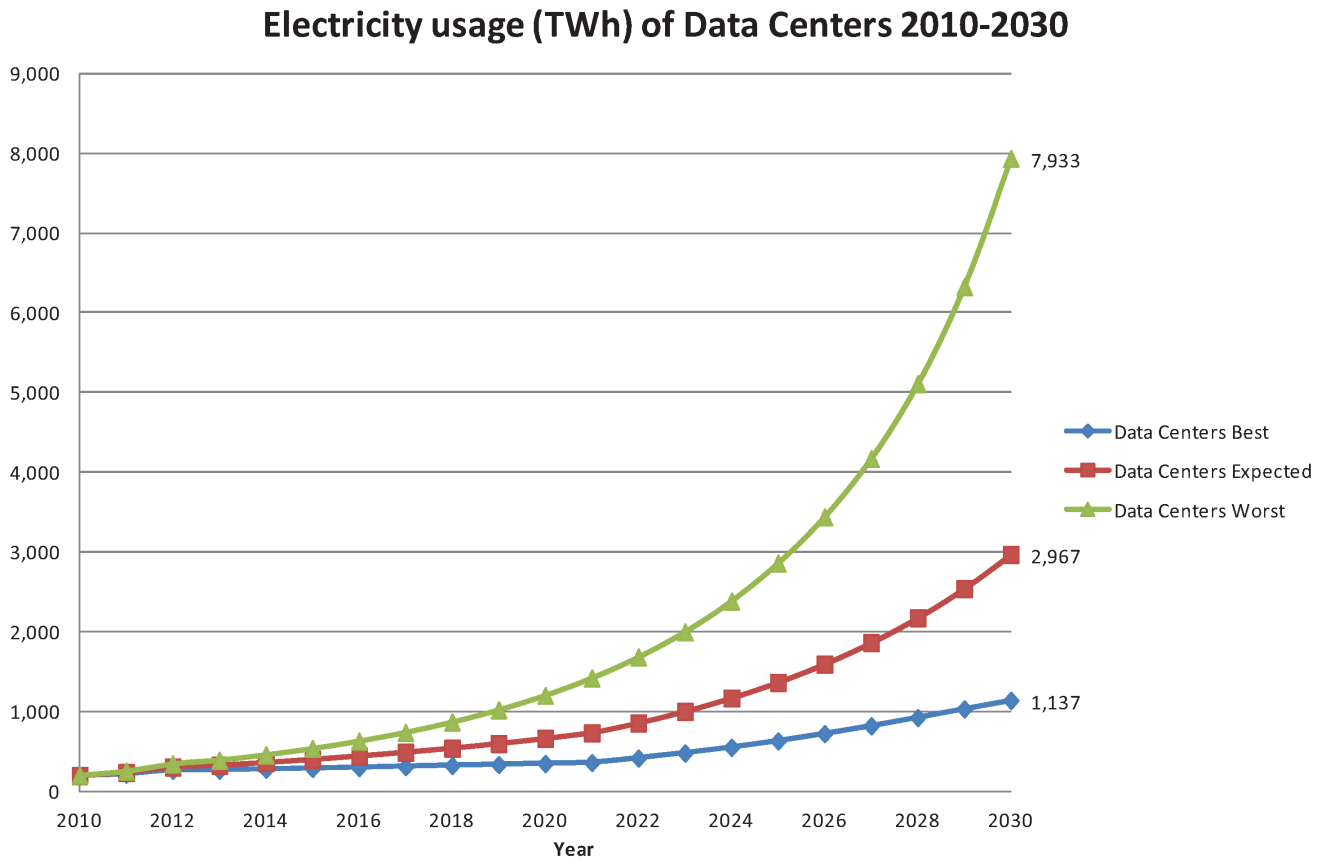
This year:

1. **Dune:** Exposes privileged instructions to userspace safely
2. **Caladan:** A library OS that eliminates tail latency and improves efficiency
3. **Breakwater:** Overload control based on task and network queueing
4. **σOS:** A cluster coordinator with a 9P-style filesystem
5. **Nu:** A distributed runtime that manages resources through rapid migration

Why work on OSes for datacenters?

- Datacenters are rapidly evolving
  - Networks are improving faster than CPUs
  - Flash is replacing mechanical disks
  - Workload scale is increasing
- Small improvements can save tons of carbon and money
- Unique challenges:
  - Tail latency impacts overall performance
  - Resource stranding harms efficiency
  - Tension between security and efficiency

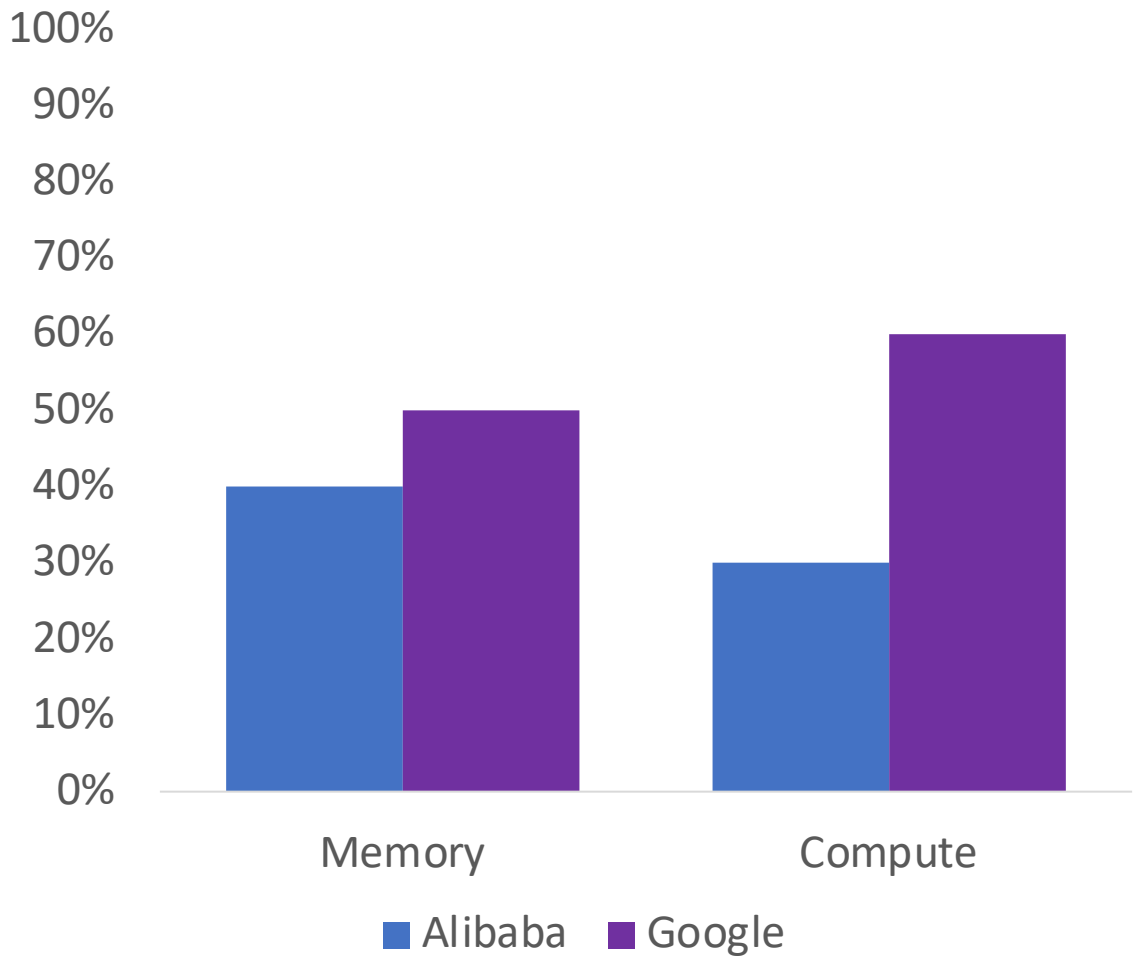
Efficiency is critical



- Current datacenters use over 1% of the world's electricity
- Projected to increase significantly over the next decade

Anders Andrae. On Global Electricity Usage of Communication Technology: Trends to 2030. Creative Commons 4.0.

Datacenters have low utilization



More than **half** of resources left idle

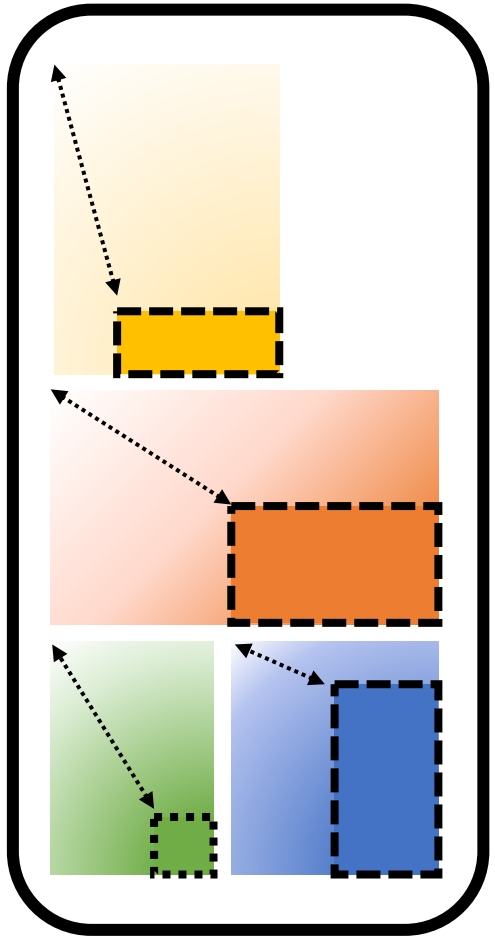
Cheng et. Al. Analyzing Alibaba's Co-located Datacenter Workloads. BigData'18. Reiss et. Al. Heterogeneity and Dynamicity of Clouds at Scale: Google Trace Analysis. SOCC'12.

Provisioning in the cloud today

Instance Size	Cores (vCPUs)	Memory (GiB)
m6i.large	2	8
m6i.xlarge	4	16
m6i.2xlarge	8	32
m6i.4xlarge	16	64
m6i.8xlarge	32	128

* Example instance sizes from AWS EC2 (01/22)

Consequence: Must provision for **peak load**

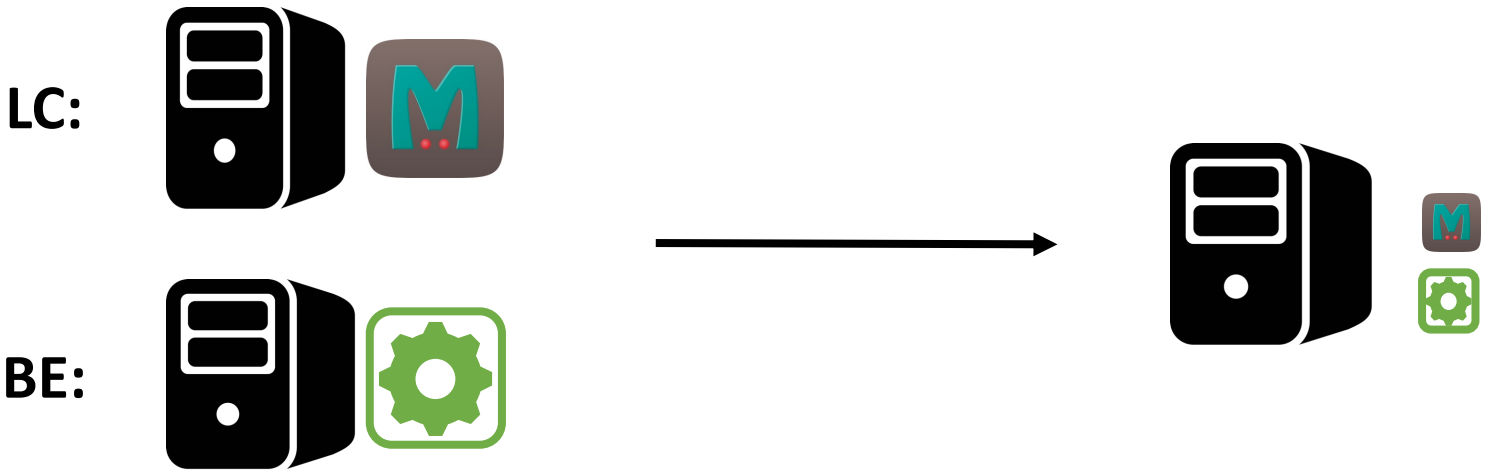


Machine

- Creates gap between instance size and actual use
- Variations in load cause low utilization

Potential solution: Overbooking resources

- Multiplexing between two classes of applications
  - **Latency-critical (LC):** high priority access to resources
  - **Best-effort (BE):** low priority access to resources, fills slack
- Keeps CPU load high under bursts and variability



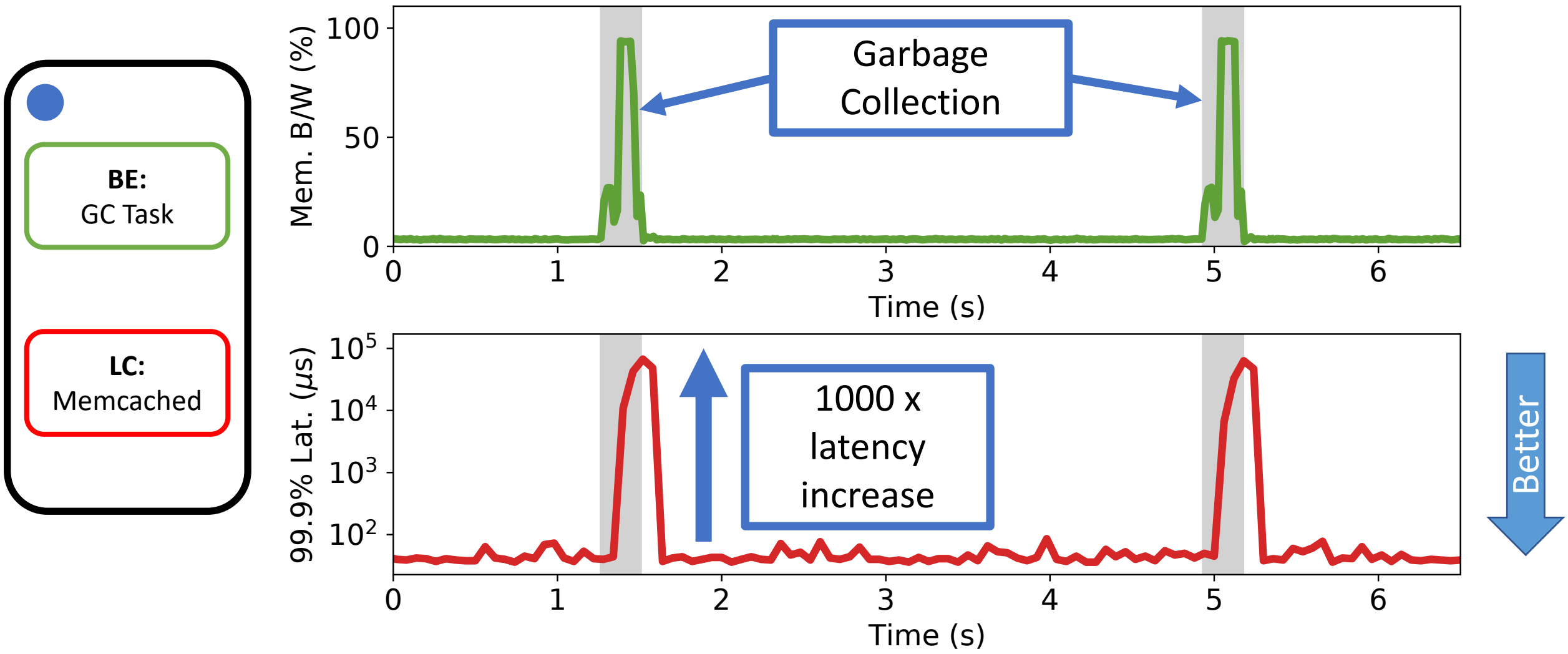
Course themes

1. **Isolation:** How can OSes safely multiplex resources?
2. **Host I/O:** How can OSes keep up with increasing network speeds?
3. **Memory:** How can OSes increase memory utilization?
4. **Applications:** What should OSes optimize for?

#1 Isolation: Many challenges

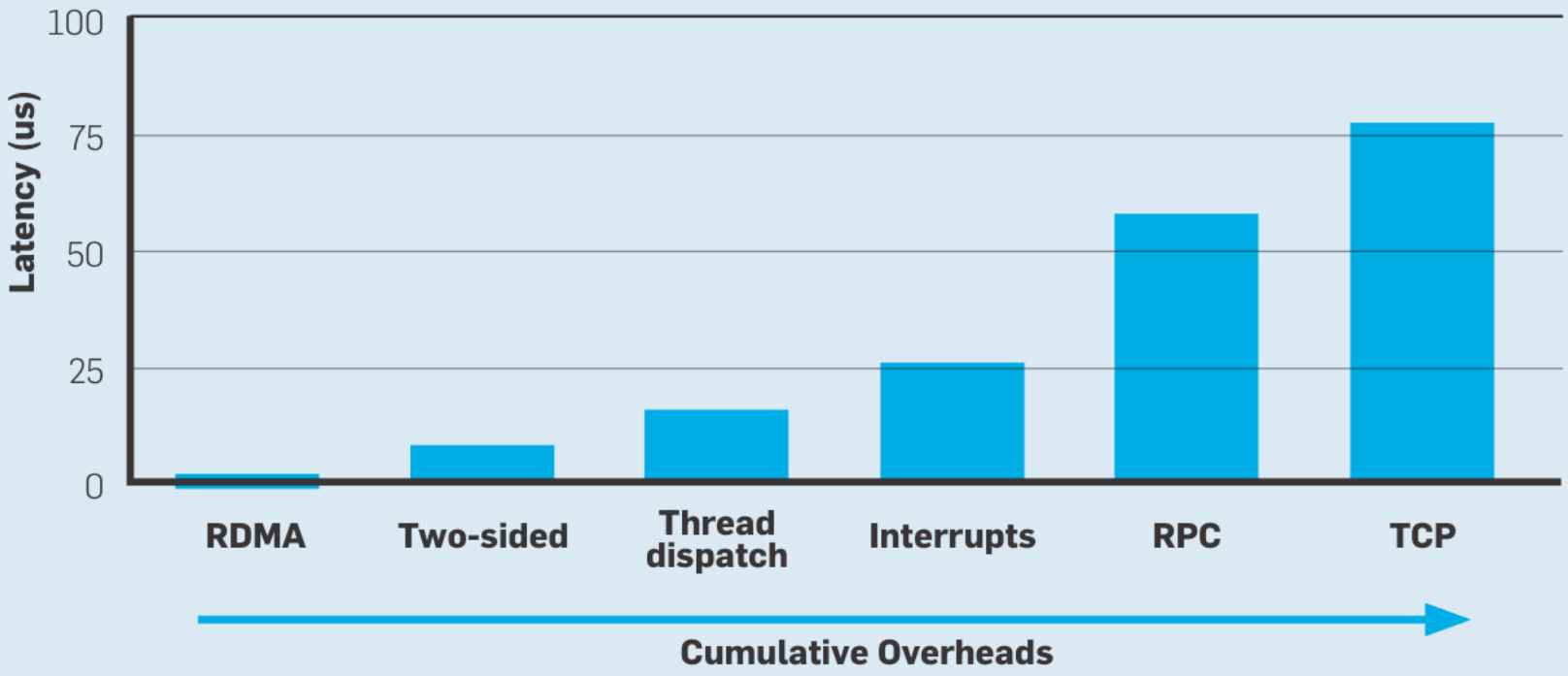
- **Security:** Must be safe to run multiple tasks on the same hardware
- **Overhead and Density:** How many tasks can we pack? How can we minimize waste?
- **Interference:** How can we make performance consistent and isolated from neighbors?
- **Compatibility:** Can we support tasks without code changes or recompilation?

Interference example



#2 Host I/O: Linux is a bottleneck

Figure 1. Cumulative software overheads, all in the range of microseconds can degrade performance a few orders of magnitude.

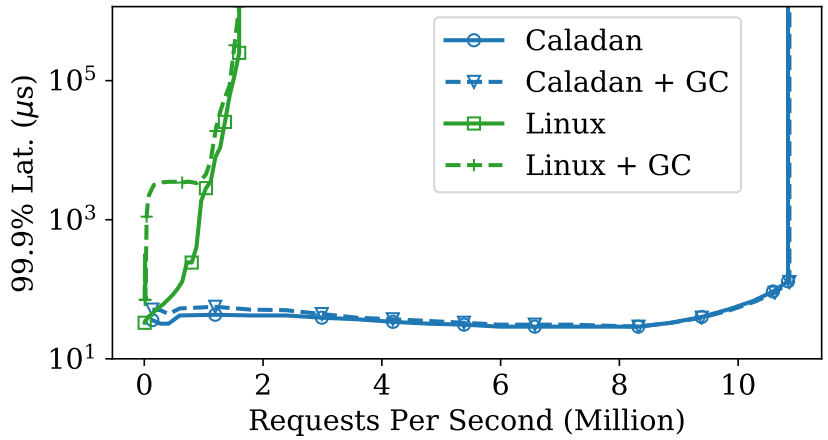


Luiz Barroso et. Al. Attack of the Killer Microseconds. CACM. April 2017

#2 Host I/O: Caladan improves efficiency

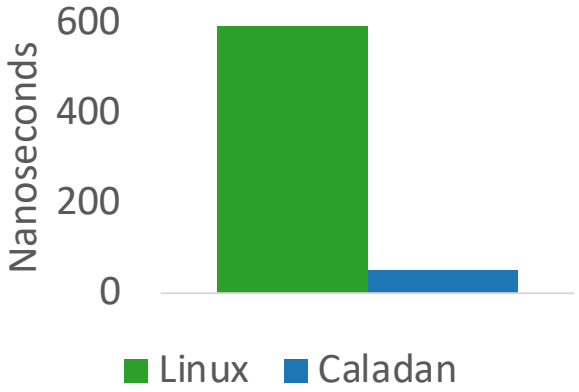
- Significant progress; but many challenges remain

Networking Performance



10x more memcached TCP/IP throughput

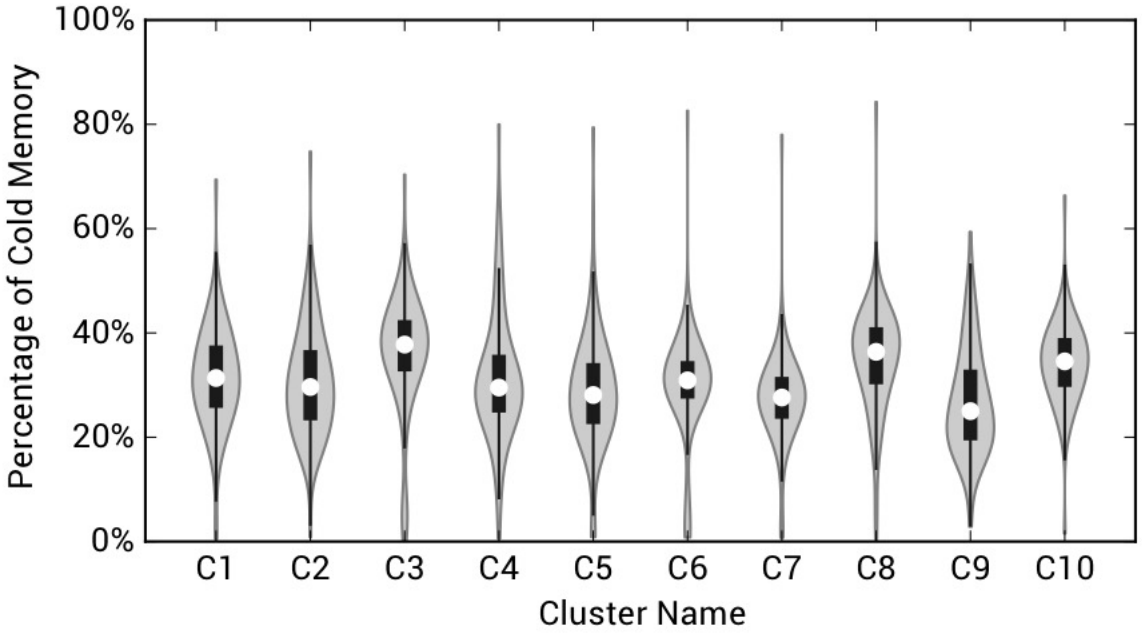
Scheduling Performance



10x less context switching overhead

#3 Memory: Cold memory wastes local RAM

- ~30% of all memory allocated is cold (i.e., rarely or never touched)
- Can we store it somewhere or someway other than local RAM?



Andres Lagar-Cavilla et. Al. Software-Defined Far Memory in Warehouse-Scale Computers. ASPLOS'19

#4 Applications: No efficiency silver bullet?

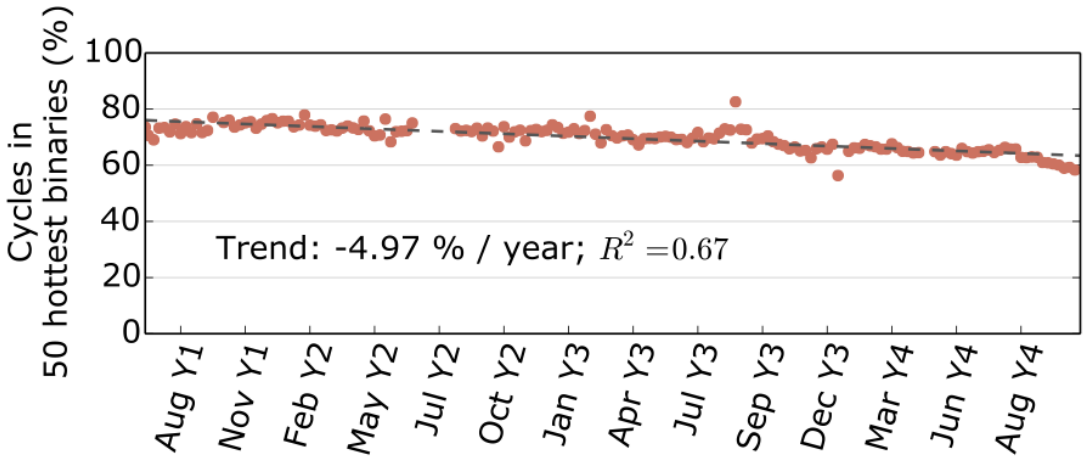


Figure 2: Workloads are getting more diverse. Fraction of cycles spent in top 50 hottest binaries is decreasing.

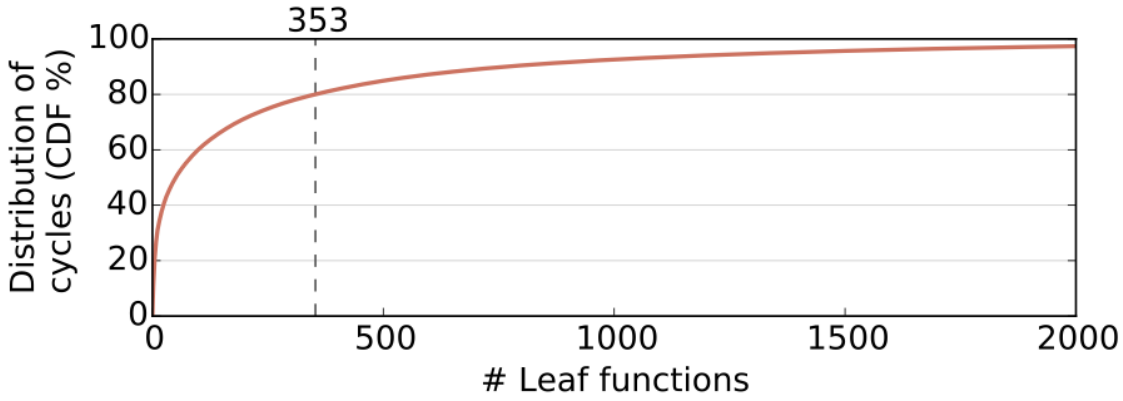


Figure 3: Individual binaries are already optimized. Example binary without hotspots, and with a very flat execution profile.

Svilen Kanev et. Al. Profiling a warehouse-scale computer. ISCA'15

What about better kernels?

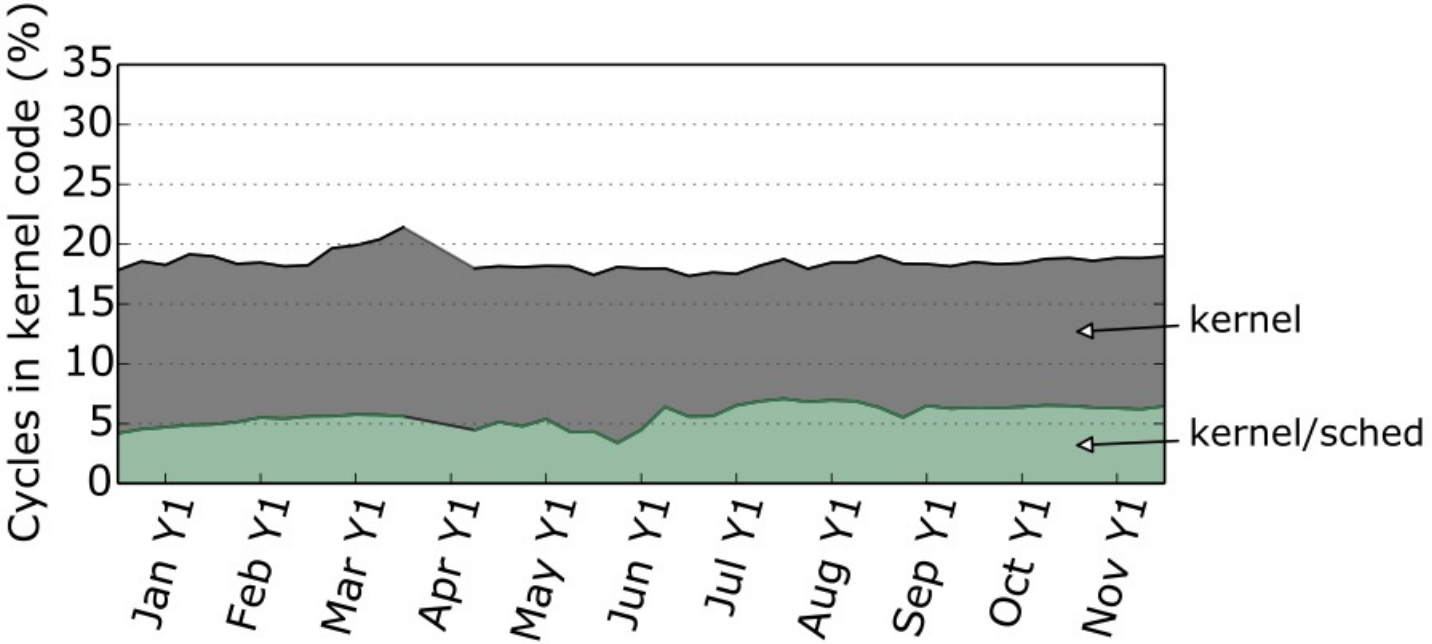


Figure 5: Kernel time, especially time spent in the scheduler, is a significant fraction of WSC cycles.

Svilen Kanev et. Al. Profiling a warehouse-scale computer. ISCA'15

Conclusion

- Better operating systems and systems software needed for datacenters
- Opportunities for large reductions in cost and improvements to energy efficiency
- This class is research focused; emphasis on a significant, open-ended project

Reminder: Fill out the class survey on Piazza