Reflections on 5 Years of the Platform Lab

John Ousterhout
Background

- **Overall interest:** enabling **low-latency computation in datacenters** (at **scale**)

- **Approach to research:**
  - Explore a small number of topics in great depth
  - Build systems that really work

- **Status in May 2015:**
  - RAMCloud project winding down
  - Raft consensus algorithm published in 2014, widespread adoption underway
  - What’s next?

- **Solve infrastructure problems uncovered by RAMCloud:**
  - Threading
  - Network transport
  - Logging
Final RAMCloud Pubs

- Rules-based programming (USENIX ATC 2015)  
  Ryan Stutsman (Utah)

- Implementing exactly-once RPC semantics (SOSP 2015)  
  Collin Lee, Seo Jin Park (MIT)

- SLIK: secondary indexes for RAMCloud (USENIX ATC 2016)  
  Ankita Kejriwal (Google)
Mendel's Student's Projects

- TorcDB: A graph database model implemented on top of RAMCloud

Jonathan Ellithorpe (Google)

- Explored different database models on RAMCloud
  - GraphDB data model looked most promising - irregular, unpredictable access patterns

- TorcDB's performance compared well against commercial graph databases
  - Queries over only a few hops on the graph - Low latency a big win
  - Queries over 3 or more hops – much data- low latency not helpful – a bandwidth problem

- A large RAMCloud client effort – Would have been useful to have:
  - Large consistent reads
  - Transaction support for secondary indexes
  - Snapshotting and bulk data loading
  - List data structure
PL Projects (Granular Computing)

- **Homa transport protocol (SIGCOMM 2018)**
  Behnam Montazeri (Google)
  - Replacement for TCP in datacenters
  - 10-100x better tail latency (especially for short messages)
  - Key ideas: receiver-driven scheduling, use network priority queues

- **Arachne: new threading architecture (OSDI 2018)**
  Henry Qin (Square)
  - Core-aware
  - System allocates cores to applications
  - Application library schedules user-level threads on cores
  - More efficient core utilization
  - Better app latency and throughput
  - Performance isolation
PL Projects, cont’d

- **NanoLog**: world’s fastest logging system (USENIX ATC 2018)
  - *Stephen Yang*
  - Move logging functionality out of runtime hot path (preprocess, postprocess)
  - Log only minimal binary info
  - Simple compression to reduce I/O
  - Result: runtime log overheads 10s of ns, vs. ~1 µs

- **CURP**: using commutativity to reduce replication cost (NSDI 2019)
  - *Seo Jin Park (MIT)*

- **MilliSort and MilliQuery**: experiments in flash bursts
  - *Yilong Li and Seo Jin Park (MIT)*
  - How many servers can be harnessed for computations lasting 1-10 ms?
  - Exposed interesting infrastructure challenge: coordination, shuffles
Personal Projects

- “A Philosophy of Software Design”:
  - Book on software design
  - Published 2018

- “Always Measure One Level Deeper” (CACM 2018):
  - Trying to fix problem in systems community with superficial and misleading evaluations
PhD Graduates

- Ankita Kejriwal, 2017 (Google)
- Benham Montazeri, 2019 (Google)
- Henry Qin, 2019 (Square)
- Seo Jin Park, 2019 (MIT postdoc)
- Stephen Yang, 2020 (Interviewing: syang0@alumni.stanford.edu)
The Next 5 Years

- No longer taking on new PhD students
  - Retirement not here yet, but looming

- A different style of research: personal projects
  - Replace TCP with Homa in the datacenter
  - High-throughput dispatching
  - Many other ideas

- Still interested in engaging with companies
Questions/Discussion