YCSB++: Benchmarking Advanced Features of Cloud Databases

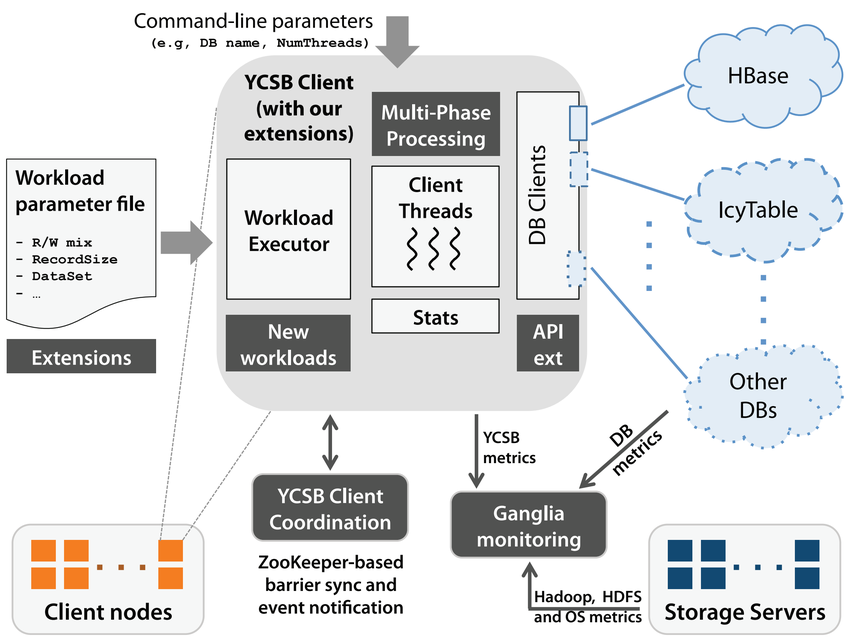
Lin Xiao, Milo Polte, Julio Lopez, Swapnil Patil, Wittawat Tantisiriroj, Kai Ren, Garth Gibson

YCSB++

- **Understand, improve scalable table store systems**
  - **Explore HBase, IcyTable (IBT), etc.**

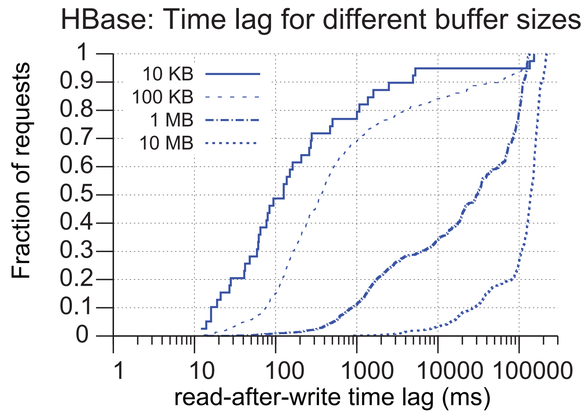
Extend YCSB version 1.3 to add:

- **Synchronized concurrent testers (zookeeper)**
- **High ingest rate exploration**
  - **Vary write behind (batch writer)**
  - **Presplit tablets anticipating lots of inserts**
  - **Bulk load: pre-build "mapfile" then "drop into" table**
- **Read side exploration**
  - **Read-after-write – what latency eventual consistency?**
  - **Offloading filtering to servers**
  - **Security ACLs – what impact on performance?**
- **Monitoring – integrate metrics from all other services**
  - **Built on Otus – per service reporting**

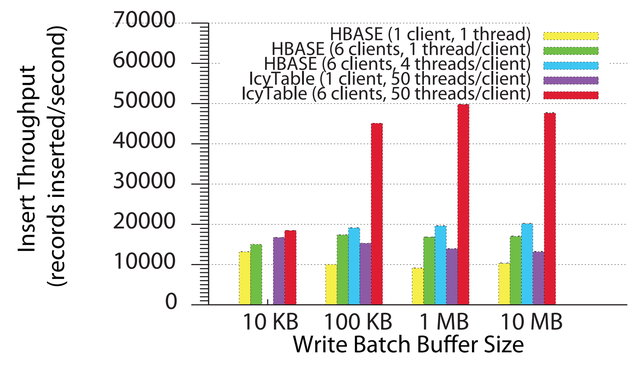


Case Studies (IcyTable, unless marked)

- **Batch insert for throughput vs. eventual consistency lag time**

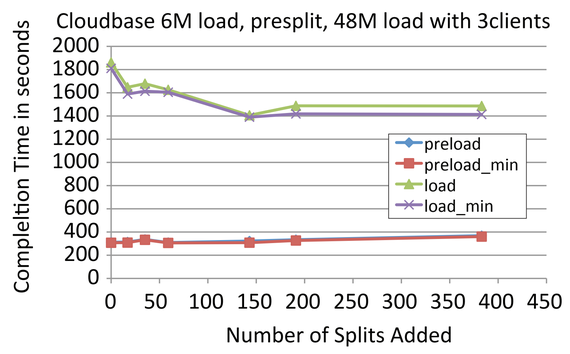


CDF of writes that suffer non-zero lag as seen by another reader

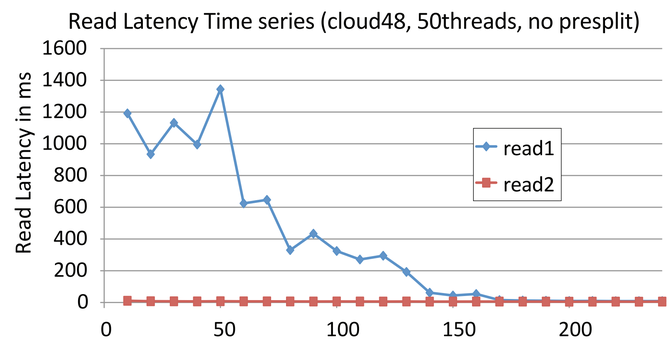


Achieved throughput as a function of DB, batch size, and workload parallelism

- **Benefit of presplitting tablets before insert burst**

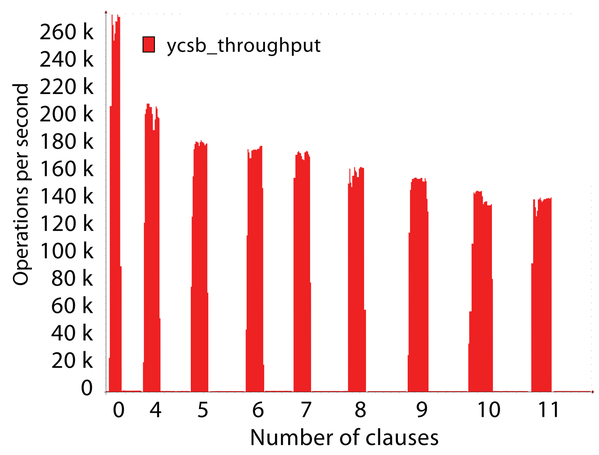


Preload range (0, ..., 1.2B) differs from later, bigger load range (0, ..., 72M), so pre-split between load phases

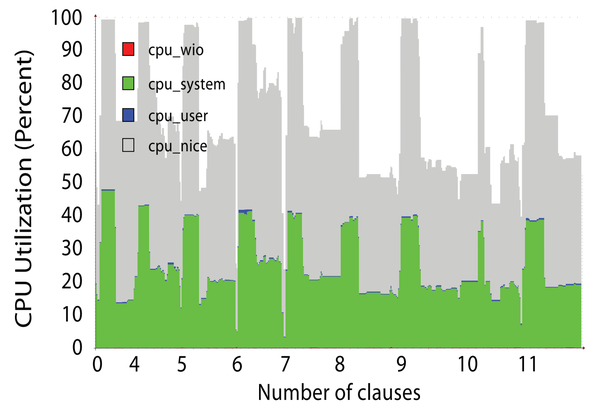


After big load of inserts, low utilization read workload sees long latencies for a few minutes

- **Inserting with credential as function of ACL complexity**
  - **0, 4, 5, ... ACLs, where all non-zero ACLs are the same total size**

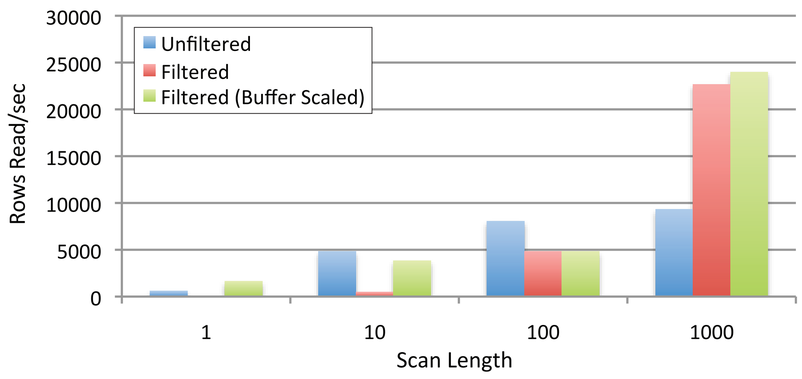


More complex ACLs slows insertion rate



Core problem is tester client CPU because each ACL requires distinct processing

- **Offloading selection filtering to servers**



- from a table with 100 column families, scan sets of rows returning only one column
- filtering at server effective for large scans
- inhibit server read-ahead to avoid wasted work

Carnegie Mellon University

Parallel Data Laboratory

SVD11