Targeting Skewed Workloads With the Smelter Distributed Database

**Jennifer Lam**, Jeffrey Helt, Haonan Lu*, and Wyatt Lloyd; Princeton University and University at Buffalo*

### Motivation
- Distributed DBs are not designed to handle skewed workloads, which are common in real applications.
- Leads to distributed DBs’ throughputs < single-machine DBs’ throughputs by an order of magnitude.
  - 
  - 
  - 
- Because distributed DBs lack single-machine DBs’ throughput multipliers:
  - Local optimizations are only applicable to systems that exist on one server.
  - Short transaction lifetimes shorten the duration for which conflicting accesses are blocked.

### Design Insight
- Embed a single-machine DB into a distributed DB.
- Distributed DB exploits single-machine DB’s throughput multipliers on skewed workloads.
  - Of servers, s-1 replicated cool shards run a distributed DB.
  - 1 replicated hotshard runs a single-machine DB.
- Co-locate popular, contended hotkeys on hotshard, whose throughput multipliers target the hardest part of the workload.

### Challenges
- **Challenge:** guarantee process-ordered serializability WITHOUT neutralizing hotshard’s throughput.
- **Solution:** Alloy Concurrency Control Protocol.
  - Forcibly non-conflicting orders on both distributed and single-machine DBs.
  - One-Touch commit: txns touch hotshard once, hotshard unilaterally commits txns on both DBs.
- **Challenge:** replicate a single-machine DB (the hotshard) in a distributed setting WITHOUT neutralizing its performance.
- **Solution:** Welder Replication Protocol.
  - Primary replica freely executes txns, decoupled from replication and buffers results until txns are replicated to all backups.
  - Safe timestamp: regularly updated global timestamp determines which buffered txns can be safely returned.

### Contributions
1. **Smelter**, the first distributed database that:
   - Scales storage capacity and throughput for non-skewed parts of a workload, and
   - Approaches the throughput of a networked, replicated single-machine DB under skewed workloads.
2. Novel **dual-DB architecture** that introduces:
   - A specialized concurrency control (CC) protocol that fuses a single-machine DB’s local CC with a distributed DB’s distributed CC, and
   - A specialized replication protocol that replicates a high-throughput single-machine database without neutralizing performance.
3. Evaluation that shows an order of magnitude better throughput than a state-of-the-art distributed database under skewed workloads.

### Evaluation
- How does Smelter’s throughput compare to a baseline state-of-the-art distributed DB?
- How well does Smelter scale throughput, compared to its baseline? Baselines: CockroachDB v20.1.9, Cicada.

### Implementation
- Fuses together:
  - CockroachDB (SIGMOD ’20) is an open-source, production-ready distributed DB written in Go.
  - Cicada (SIGMOD ’17) is a research single-machine DB written in C++.
  - Added networking and Welder replication.

### Results
- **(a) Throughput varying skew**
  - Throughput (tps) varies with different levels of skew.
  - Uniform skew: 19K
  - Medium skew: 13K
  - High skew: 8K
- **(b) Scalability under high skew (zipf s=1.2)**
  - Throughput (tps) scales with increasing numbers of servers.
  - CockroachDB max: 83K
  - Smelter max: 48K
  - Cicada (single-machine DB) max: 8K

---

---