Lessons learned while evolving Box’s Database Infrastructure

Tamar Bercovici, Senior Director of Engineering
@TamarBercovici
85K Customers

69% Fortune 500
Lessons learned while evolving Box’s Database Infrastructure | @Tamar Bercovici

7 years ago

- 6 person Backend Engineering team
- No DBAs
- 100s of millions of data records
- PHP codebase pushing 20k+ QPS
- Single active MySQL DB + MC cache pool

Today

- Dedicated Database Dev team
- Dedicated Database Operations team
- 100s of billions of data records
- Multiple clients pushing 1M+ QPS
- 100s of servers spanning MySQL, MC, Redis, HBase…
Lessons learned while evolving Box’s Database Infrastructure | @Tamar Bercovici

For more details: The Story of Sharding at Box, AYB13 Conference
Lessons learned while evolving Box's Database Infrastructure | @Tamar Bercovici

Starting to scale  
Nirvana  
Post-sharding  
Building resiliency  
Armageddon
Scaling ahead of the business is hard

- The ground is constantly shifting underneath you
- You’re often hurtling towards a cliff, but don’t realize it until you have one foot over the abyss
Goal: Put out the fire *fast* while making stability *sustainable* for the long term
Goal: Reduce the frequency and impact of DB outages

- Client optimizations
- Automated Failover
- Read Replica Framework
- Connection Management
- Rate Limiting
- Remove SPOFs
- De-coupling
Lessons learned while evolving Box’s Database Infrastructure   |   @Tamar Bercovici

Leveraging our Read Replicas

• Reading from replicas is tricky

  • Lag can lead to inconsistent reads
  • Failing on lag can lead to high error and retry rates

• Introduced a new read API with a master checkpoint parameter

Master

Replica

Execute WRITE
Grab checkpoint

Failed READ

Successful READ retry

Fast read after write consistency

Offload 75% of our read traffic
Automating Failovers

- Mean Time to Remediate (mttr) > monthly SLA allotment
- **Spatula**: Automatic database flipper
- Correctly assessing database health is key
- CPU load, lock timeouts, error rates... are all secondary signals
- Executed simple queries as healthcheck
Rate limiting in Credence
Gossip-based rate limiting

- Requirements:
  1. Local decision making
  2. Distributed state
  3. Fast convergence

- Adapted the Push-Sum protocol presented in
  Gossip-Based Computation of Aggregate Information, David Kempe, Alin Dobra, Johannes Gehrke

\[
\text{sum} = 7
\]
Streaming Push-Sum

- Nodes gossip about a sequence of “time” buckets containing usage data
- Leader node creates new bucket every set interval
Lessons learned while evolving Box's Database Infrastructure  |  @Tamar Bercovici

- Investing in our Platform
- Post-sharding
- Managing user traffic
- Building resiliency
- Starting to scale
- Armageddon
When meeting your SLA become the norm
It’s been quite a ride…
It’s tempting to think of “Nirvana” as a fixed point, but really, it’s more like a moving target.
An incremental, iterative approach will lead to the best outcomes over time

- Always have a clear value-adding goal
- Constantly invest in both the short and long term
- Let Nirvana be your north star, not your destination
Thank you!

@TamarBercovici
@BoxEng