Dynamically Replicating memcached

COS 518: *Advanced Computer Systems*

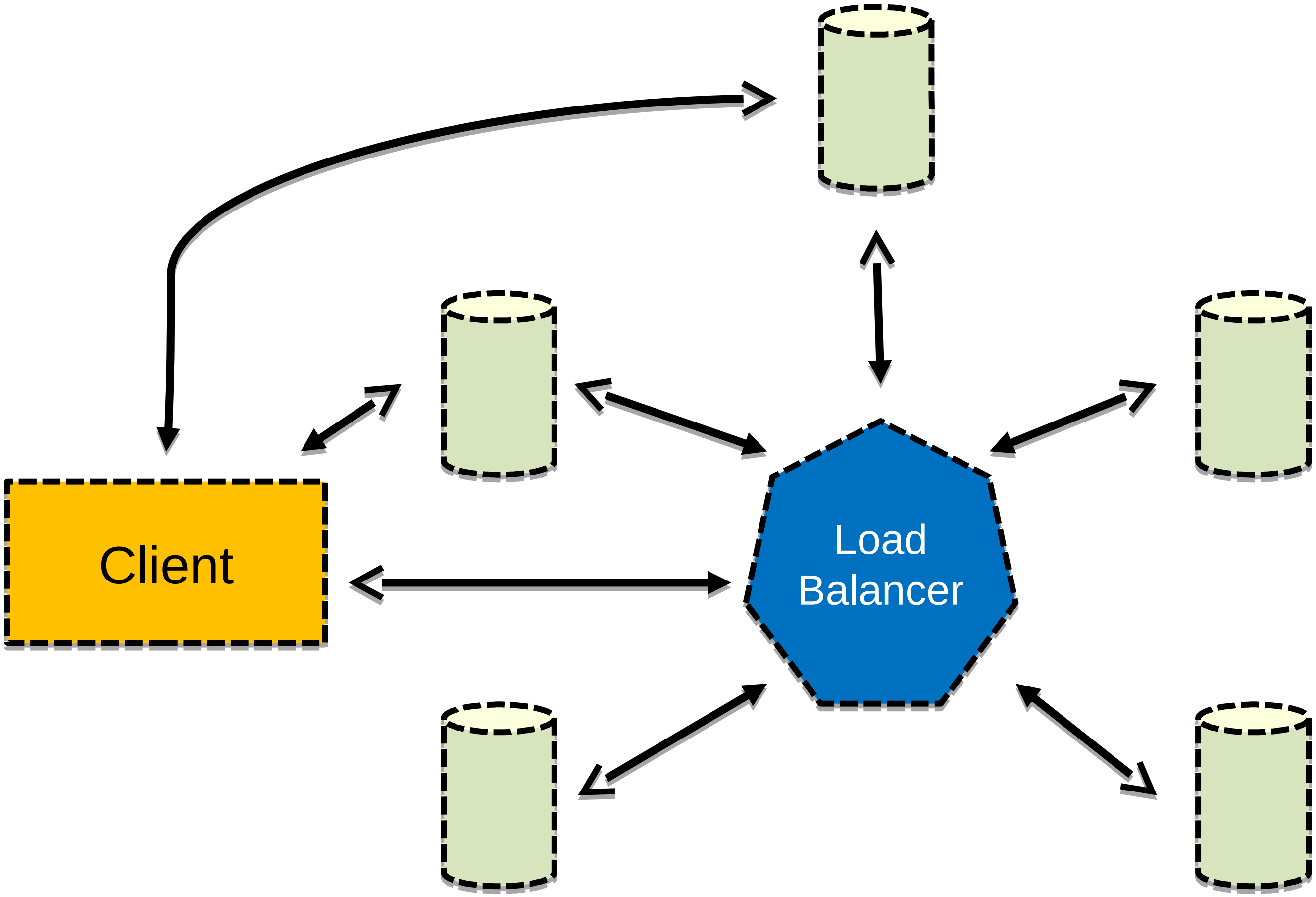
Rushy R. Panchal and Michael J. ~~Freedman~~
Friedman

March 28th, 2018

Problem Overview

- memcached does not support dynamically replicating keys for performance
- Large-scale systems will have **some keys** that are **accessed significantly more frequently** than others (“80/20 rule”)
- **Most popular keys** can be **replicated** to provide additional throughput and distribute load

Technical Solution



Technical Solution

- Central load balancer
- Load balancer will **query servers for most popular keys** and **dynamically redistribute** (and replicate) them across servers
- memcached clients find a key by indexing into a table of servers → load balancer **updates client table** in-place to change where keys are found

Implementation Plan

- Server-side
  - **Create a Load Balancer** that can redistribute keys given load characteristics
  - **Modify memcached server** to provide most popular keys on demand to Load Balancer
- Client-side
  - Modify Load Balancer to reconfigure clients
  - **Modify memcached client** to accept dynamic configuration from Load Balancer
- *Stretch: replicate Load Balancer for fault-tolerance*

Evaluation Plan

- Primary evaluation: performance vs standard memcached version with **20% popular keys**
- Other useful evaluation: performance vs standard memcached) with **no popular keys** → evaluate overhead incurred by load balancer