An Analysis of Load Imbalance in Scale-out Data Serving

Stanko Novakovic  
EPFL  
stanko.novakovic@epfl.ch  

Alexandros Daglis  
EPFL  
alexandros.daglis@epfl.ch  

Edouard Bugnion  
EPFL  
edouard.bugnion@epfl.ch  

Babak Falsafi  
EPFL  
babak.falsafi@epfl.ch  

Boris Grot  
University of Edinburgh  
boris.grot@ed.ac.uk

ABSTRACT
Despite the natural parallelism across lookups, performance of distributed key-value stores is often limited due to load imbalance induced by heavy skew in the popularity distribution of the dataset. To avoid violating service level objectives expressed in terms of tail latency, systems tend to keep server utilization low and organize the data in micro-shards, which in turn provides units of migration and replication for the purpose of load balancing. These techniques reduce the skew, but incur additional monitoring, data replication and consistency maintenance overheads. This work shows that the trend towards extreme scale-out will further exacerbate the skew-induced load imbalance, and hence the overhead of migration and replication.

CCS Concepts
•Computer systems organization → Distributed architectures;

Keywords
Load imbalance; Replication

1. INTRODUCTION
A significant portion of contemporary web-scale applications is latency-sensitive. However, designing a datacenter-scale system that guarantees low latency for the majority of user requests is notoriously challenging. Such latency-critical systems are often powered by in-memory distributed key-value stores (KVS), which span hundreds of servers and provide the means for locating and retrieving data fast by using consistent hashing. The flexibility and scalability of using KVS in a scale-out environment has led to its broad use as a state-of-the-art approach for low-latency data serving applications. However, a substantial pain point of the approach is that it leads to severe inter-server load imbalance whenever the data popularity distribution is skewed.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
SIGMETRICS ’16 June 14-18, 2016, Antibes Juan-Les-Pins, France
© 2016 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-4266-7/16/06.
DOI: http://dx.doi.org/10.1145/2896377.2901501

To ensure good quality of service under skewed workloads, providers set strict service-level objectives (SLO), requiring the service deployment to respond to user requests within a short and bounded delay. What makes the challenge even harder is that designing for the average latency is not enough; a good service guarantees that the vast majority of the requests will meet the SLO, and thus targets a 99th or even 99.9th percentile latency of just a few milliseconds [3]. To satisfy such strict latency requirements and at the same time sufficiently utilize the available resources and deliver high throughput, providers rely on various migration and replication schemes.

In this work we first characterize the skew in data serving workloads and show that the trend towards extreme scale-out [4] will further exacerbate the load imbalance problem. We then argue that such increase in load imbalance will raise the frequency of migration and replication operations to keep the system at the desired level of utilization [6].

2. SCALE-OUT-INDUCED IMBALANCE
KVS typically handle very large collections of data items and millions or billions of user requests per second. In such a setting, skewed distributions emerge naturally, as the popularity of the data items varies greatly. Previous work has shown that popularity distributions in real-world KVS workloads follow a power-law distribution, resulting in an access frequency imbalance, commonly referred to as skew [1]. This skewed distribution is often modeled by the power-law Zipfian distribution [2]. A classic example of such skewed popularity distribution is a social network, where a very small subset of users is extremely popular as compared to the average user. These two distinct user categories (popular versus the rest) result in a popularity distribution with a skyrocketing peak and a long tail.

Dataset distribution across the deployment’s collection of servers is typically done by grouping data items into “micro-shards”, each server being responsible for hosting and serving hundreds or thousands of them from its local memory [9]. This data distribution is done by applying a hash function on the key of the data items, which maps each of them to a micro-shard and each micro-shard to a server. In practice, a collection of micro-shards mapping to a single server represents a single data shard that is served by its corresponding server. Thus, even after grouping data items together into shards, the presence of the original popularity skew is still observable as a popularity skew across shards.

We define the shard skew as the access ratio between the hottest server and the average. Shard skew arises in a data-


Facebook. Introducing "Yosemite": the first open source modular chassis for high-powered microservers, 2015.

