Kurma: Geo-Distributed Load Balancer for Back-End Storage Systems

Kirill L. Bogdanov
KTH Royal Institute of Technology

Waleed Reda
Université catholique de Louvain
KTH Royal Institute of Technology

Gerald Q. Maguire Jr.
KTH Royal Institute of Technology

Dejan Kostić
KTH Royal Institute of Technology

Marco Canini
KAUST

Abstract

Geo-distributed backend deployments are becoming increasingly common for large-scale web services. Data is typically replicated across multiple dataceneters in order to reduce user response times, improve data locality, and increase tolerance to failures. The increased density of datacenters reduces the average distance (and thus network latency) between neighboring datacenters. This allows services deployed in neighboring locations to share workload, when necessary, without violating strict Service Level Objectives (SLOs).

In this work, we present Kurma, a practical implementation of a geo-distributed load balancer for back-end storage systems. Kurma integrates network latency and system's service time distribution to solve a decentralized rate-based performance model allowing global coordination among datacenters. Using this model, Kurma proactively redirects requests away from loaded datacenters *before* SLO violations occur.

We integrated Kurma with Cassandra, a popular storage system. Using real world traces, and geo-distributed deployment across Amazon EC2 we demonstrated the ability of Kurma to reduce SLO violations up to a factor of 4.

Overview

Modern web-based services demonstrate temporal and spatial variability in load. Kurma exploits the decorrelated nature of geo-distributed workloads, enabling it to effectively share spare capacity among datacenters *without* adding more resources (i.e., redirect load from overloaded to underloaded datacenters). This leads to reduction in both: SLO violations and the cost of running a service.

Kurma instances are logically-centralized at each datacenter. At regular intervals, in the order of a few seconds, each instance of Kurma combines (i) WAN latencies among datacenters, (ii) current load, and (iii) the operating curve to solve a decentralized performance model that determines how requests should be split among datacenters.

The operating curve relates the arrival rate of requests to resulting latency and throughput and hence identifies the load that each datacenter can sustain before violating its SLO at the target percentile. When redirecting requests, Kurma uses the operating curve to estimate the rate of SLO violations given the load and the WAN latency to the remote datacenter. The operating curve can be obtained via offline profiling or estimated using common queue modeling techniques.

Results. To evaluate Kurma, we deployed Cassandra clusters across three geo-distributed datacenters of Amazon's EC2 located in London, Frankfurt, and Ireland, each running 5 VMs. Using the YCSB benchmark and real world traces, we subject each cluster to varying levels of load and evaluated the ability of Kurma to reduce SLO violations. Fig. 1 shows comparison of Kurma with three baselines: (i) Cassandra's default load balancer - Dynamic Snitch, (ii) C3 state of the art load balancer, and (iii) baseline performing no geo-distributed load balancing. Kurma is able to achieve a factor of 4 reduction in SLO violations in comparison to the evaluated baselines.

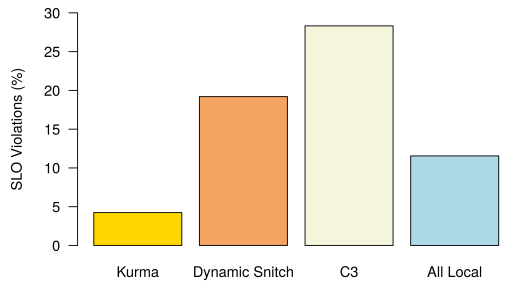

Figure 1: SLO violations for different techniques.