Fast and Accurate Load Balancing for Geo-Distributed Storage Systems

Kirill L. Bogdanov\textsuperscript{1} \quad Waleed Reda\textsuperscript{1,2} \quad Gerald Q. Maguire Jr.\textsuperscript{1}

Dejan Kostic\textsuperscript{1} \quad Marco Canini\textsuperscript{3}

\textsuperscript{1}KTH Royal Institute of Technology
\textsuperscript{2}Université Catholique de Louvain
\textsuperscript{3}KAUST
Geo-Distributed Services

Service Level Objective (SLO):
Request completion time at the target percentile (e.g., 30 ms at 95th percentile)
Web-based services demonstrate temporal and spatial variability in load.

**Problem:** it is difficult to meet strict SLOs, while maintaining high resource utilization and low cost.
Approach 1 - Datacenter Elasticity

![Graph showing the arrival rate over time for Datacenter-1 load.](image-url)
Approach 1 - Datacenter Elasticity

![Graph showing datacenter load and elasticity threshold over time](graph.png)
Approach 1 - Datacenter Elasticity
Approach 1 - Datacenter Elasticity

- Provisioning delay (minutes) due to time needed to spawn and warm up a VM
- Hard to predict workload far into the future
- Load spikes can be short lived

Lead to overprovisioning!
Approach 2 - Geo-Distributed Load Balancing

Redirection delays
Inaccurate response time estimation
Excessive or insufficient redirection

Redirection delay
SLO violations

How much to redirect?
Our Approach: Kurma

- Reacts to changes in load within seconds
- Avoids unnecessary scaling out
- Accurately estimates remote rate of SLO violations
- Tames SLO violation at the target level
Request Completion Time

**Base Propagation:** Stable component associated with packet propagation along a network path

**Delay Variance:** Variable component associated with competing traffic and queuing

**Service Time:** Variable component associated with load on the server

Kurma solves global optimization model while considering: Base Propagation + Delay Variance + Service Time at all datacenters
Understanding Service Time

Datacenter Frankfurt
5 Server Cassandra cluster

![Graph showing the relationship between service violations and offered load. The graph indicates that as the offered load increases, the number of violations also increases significantly beyond a certain point.]
Understanding Service Time

Challenge: How to accurately estimate remote fraction of SLO violations at runtime under variable network conditions?
Understanding Service Time

![Graphs showing service time and latency distribution.](image)
Understanding Service Time

Insight: the farther away a remote datacenter is, the less loaded it should be to serve remote requests within a given SLO target.
Understanding WAN Latency

- Base propagation delay
- Service time distribution recorded locally at a specific load
- Monte Carlo Simulations
Understanding WAN Latency

Base propagation delay

Monte Carlo Simulations

Service time distribution recorded locally at a specific load
Understanding WAN Latency

- Base propagation delay
- Service time distribution recorded locally at a specific load
- Monte Carlo Simulations
- Gives SLO violation rate given a specific load and WAN conditions

CDF

WAN residual latency

RTT [ms]

Latency [ms]

Monte Carlo Simulations

Residual WAN latency

Local service time

Base propagation latency

Estimated remote service time, Kurma (base + residual)

RTT Frankfurt to Ireland

Gives SLO violation rate given a specific load and WAN conditions
Understanding WAN Latency

- **Base propagation delay**
- **WAN residual latency**

![Graph showing CDF and RTT](image)

RTT Frankfurt to Ireland

- Residual WAN latency
- Local service time
- Estimated remote service time, Kurma (base + residual)
- Estimated remote service time (raw samples)

![Graph showing CDF and Latency](image)

- CDF
- Latency [ms]

Estimation Error
Incorporating WAN and Load

5 VM Cassandra cluster

SLO violations at 30 ms [%]

Offered load [1000x]
Incorporating WAN and Load

5 VM Cassandra cluster

- Local datacenter
- Remote 12 ms RTT
Incorporating WAN and Load

5 VM Cassandra cluster

- Local datacenter
- Remote 16 ms RTT
- Remote 12 ms RTT

SLO violations at 30 ms [%]

Offered load [1000x]
Incorporating WAN and Load
Optimisation Model

Runtime load in each datacenter \( \{\lambda_1, \lambda_2, \lambda_3\} \)

Optimisation Problem

- Minimize global SLO violations (KurmaPerf)
- Minimize the cost of running a service (KurmaCost)
Implementation

Each Epoch
2.5 sec $\rightarrow$ 0.4Hz

Perform run-time WAN latency measurements

Aggregate load information (rates of requests)

Exchange metrics to obtain global view

Solve decentralized performance model

Global View: latencies + loads
Implementation

Each Epoch
2.5 sec → 0.4Hz

- Perform run-time WAN latency measurements
- Aggregate load information (rates of requests)
- Exchange metrics to obtain global view
- Solve decentralized performance model
- Enforce computed rates of requests redirection
Evaluation Setup

Geo-distributed Cassandra cluster

- 3 Amazon EC2 datacenter (Ireland, Frankfurt, London)
- 5 x r5.large VMs per datacenter
- SLO: 30 ms at the 95\textsuperscript{th} percentile
- Modified YCSB to replay workload traces
  (World Cup http://ita.ee.lbl.gov/html/contrib/WorldCup.html)

Experiments:

- Minimizing SLO violations for reads
- Maintaining Target SLO (accuracy)
- Cost Savings for 1 min billing intervals (simulations)
- Reads and writes, scalability, etc. [link here](#).
Workload Trace

No elastic scaling

Load threshold for 5% SLO violations
Cumulative Normalized SLO Violations

Kurma’s SLO violations are at 2.4%

The numbers shown above the bars indicate the amount of inter-datacentre traffic transferred, whiskers → 75\textsuperscript{th} percentile
Kurma’s SLO violations are at 2.4%.

The numbers shown above the bars indicate the amount of inter-datacentre traffic transferred, whiskers $\rightarrow$ 75th percentile.
Average Provisioning Cost Over 30 Consecutive Days

- Reactive threshold based elastic controller
- Minimum billing period of 1 minute
- Results obtained using simulations
Average Provisioning Cost Over 30 Consecutive Days

- **All Shared**: WAN latency = 0ms, Bandwidth cost = 0$
  - 21% savings

- **KurmaCost**
  - 14% savings

- **KurmaPerf**
  - 8% savings

- **All local**
  - 0% savings

**Keeps SLO violations under 5%** (minimize redirections while avoiding scaling out)

**Minimize SLO violations** (no consideration for traffic usage)
Taming SLO Violations Under Elastic Threshold

No elastic scaling

5% SLO Bound
Conclusion

Kurma – **fast and accurate** load balancer for geo-distributed systems that takes advantage of spatial variability in load

**Decouples** end-to-end response time into components of base propagation latency, network congestion, and service time distribution

By operating at the granularity of a few seconds, Kurma **reduces SLO violations or lowers the costs** of running services by avoiding excessive global service overprovisioning

Contact: KIRILLB@kth.se