Fast and Accurate Load Balancing for Geo-Distributed Storage Systems

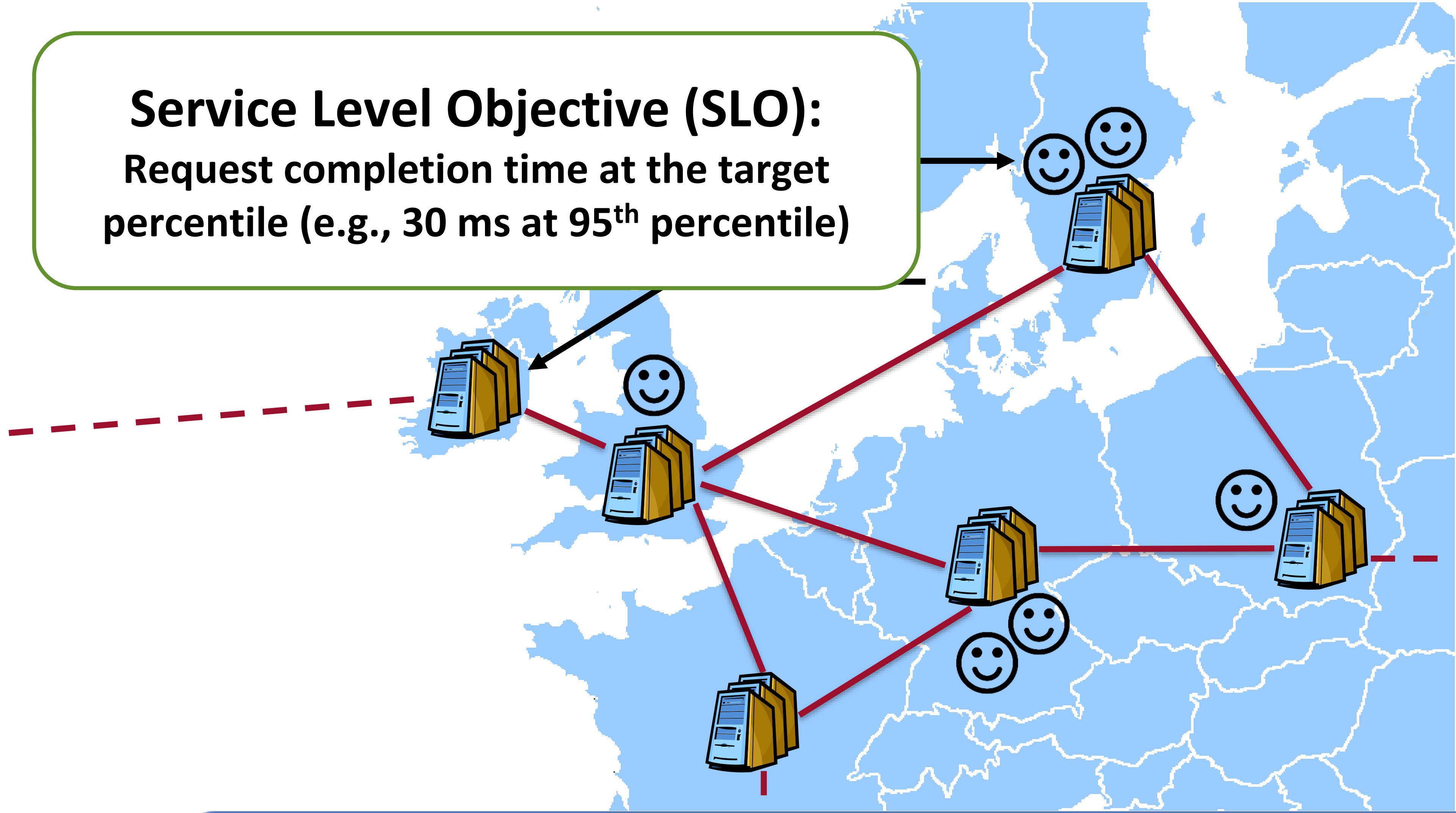
Kirill L. Bogdanov[1] Waleed Reda[1,2] Gerald Q. Maguire Jr.[1]
Dejan Kostic[1] Marco Canini[3]

[1]KTH Royal Institute of Technology
[2]Université Catholique de Louvain
[3]KAUST

Geo-Distributed Services

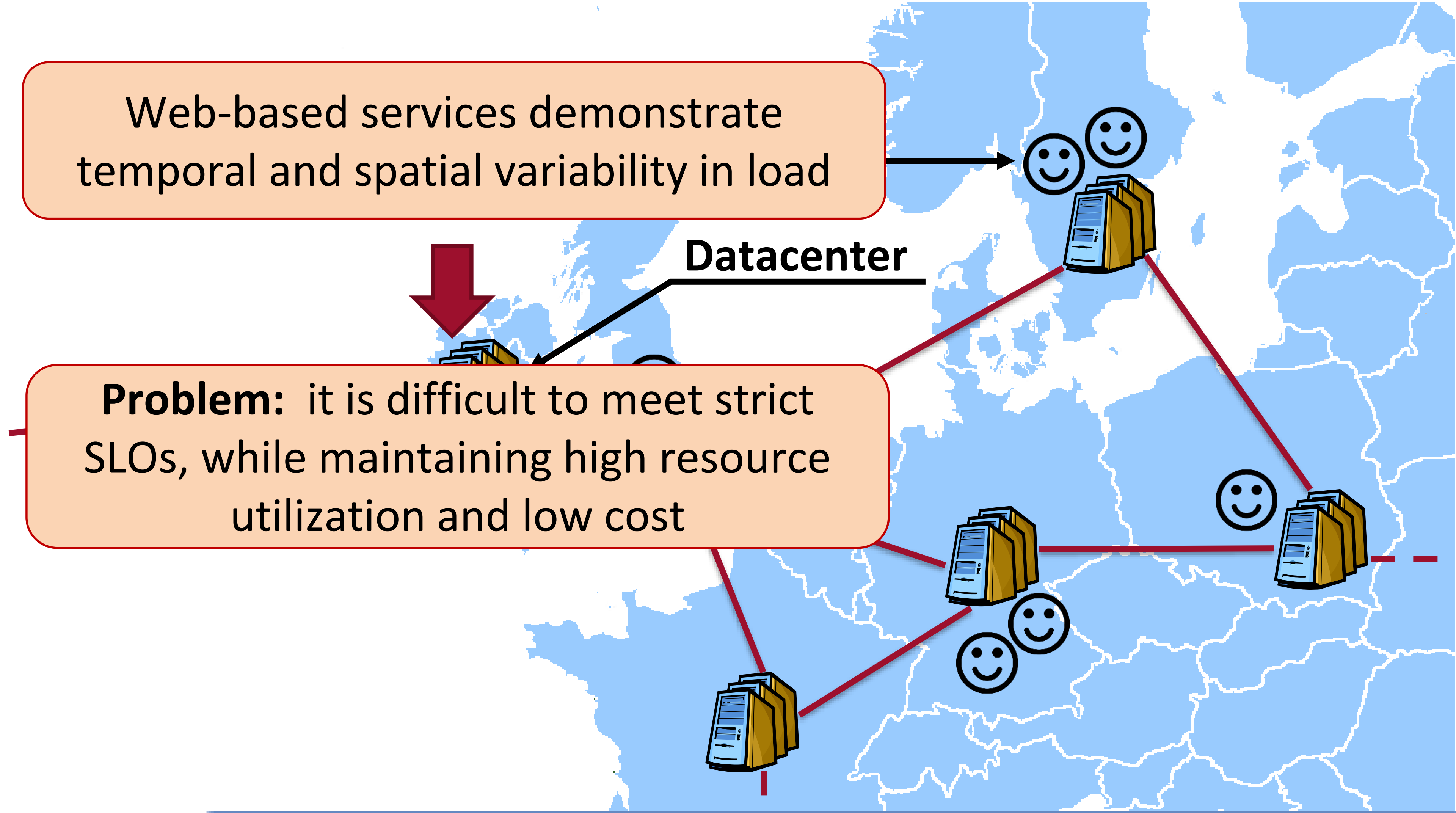
Service Level Objective (SLO):

Request completion time at the target percentile (e.g., 30 ms at 95th percentile)



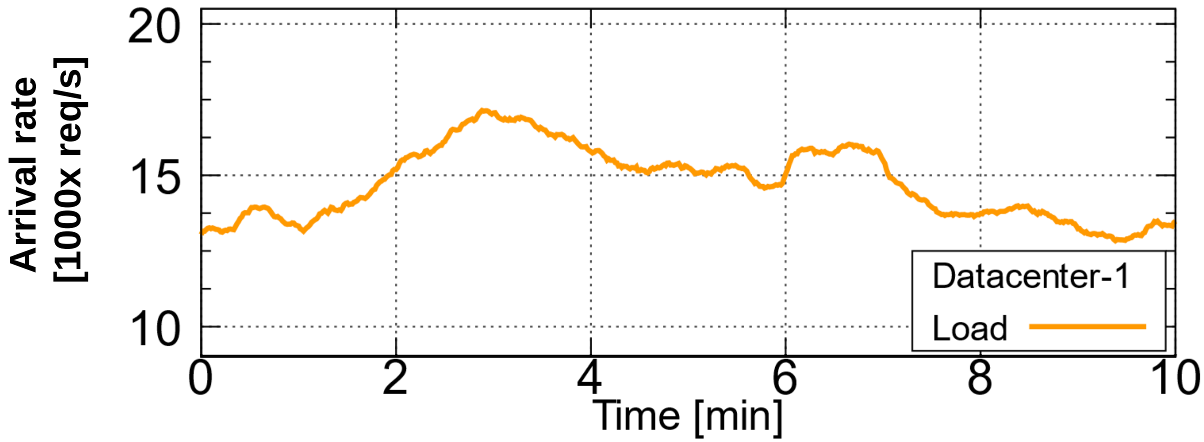
Geo-Distributed Services

Web-based services demonstrate temporal and spatial variability in load

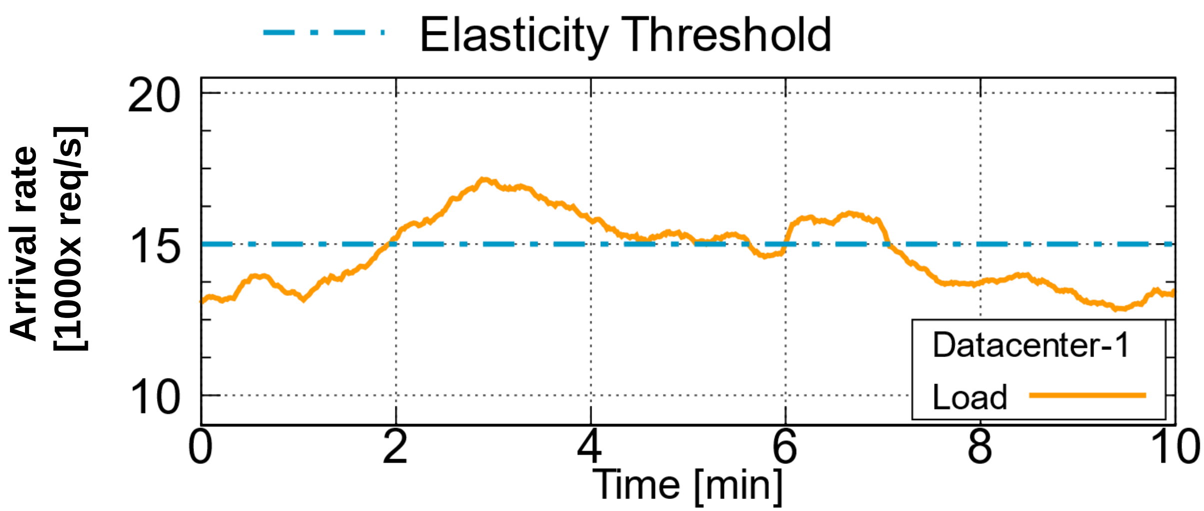


Problem: it is difficult to meet strict SLOs, while maintaining high resource utilization and low cost

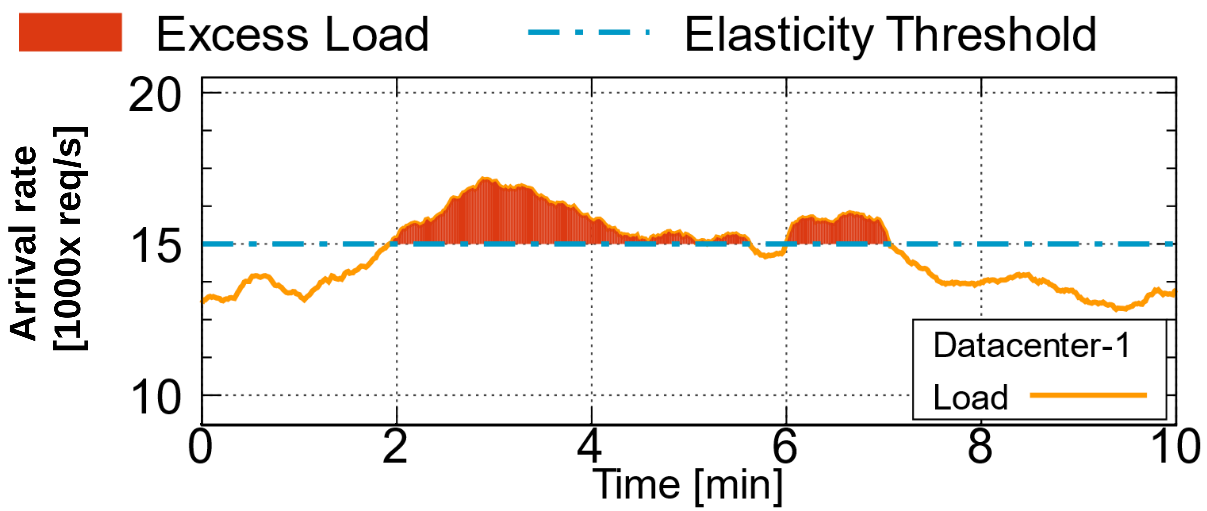
Approach 1 - Datacenter Elasticity



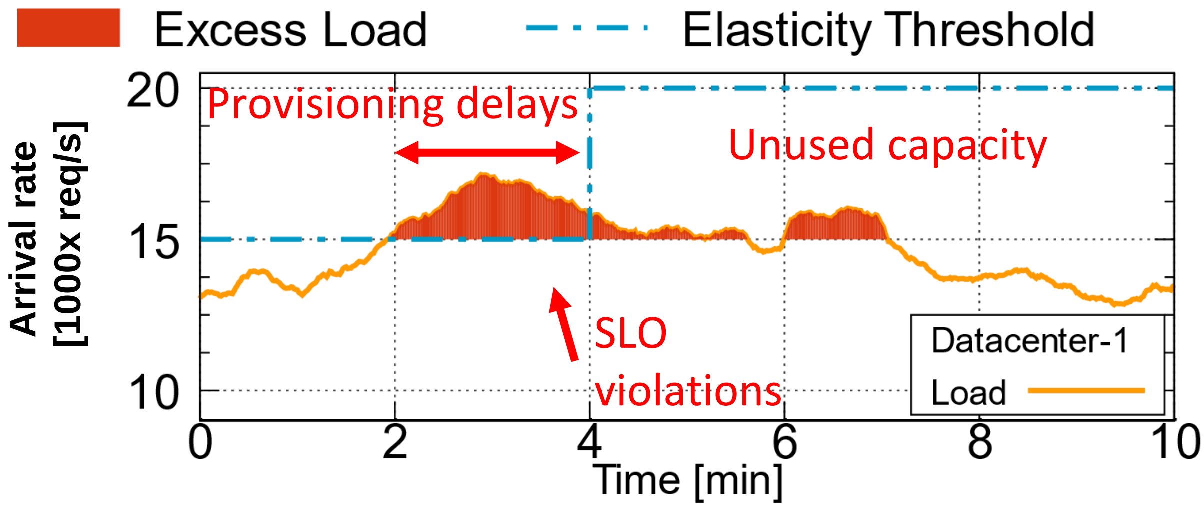
Approach 1 - Datacenter Elasticity



Approach 1 - Datacenter Elasticity



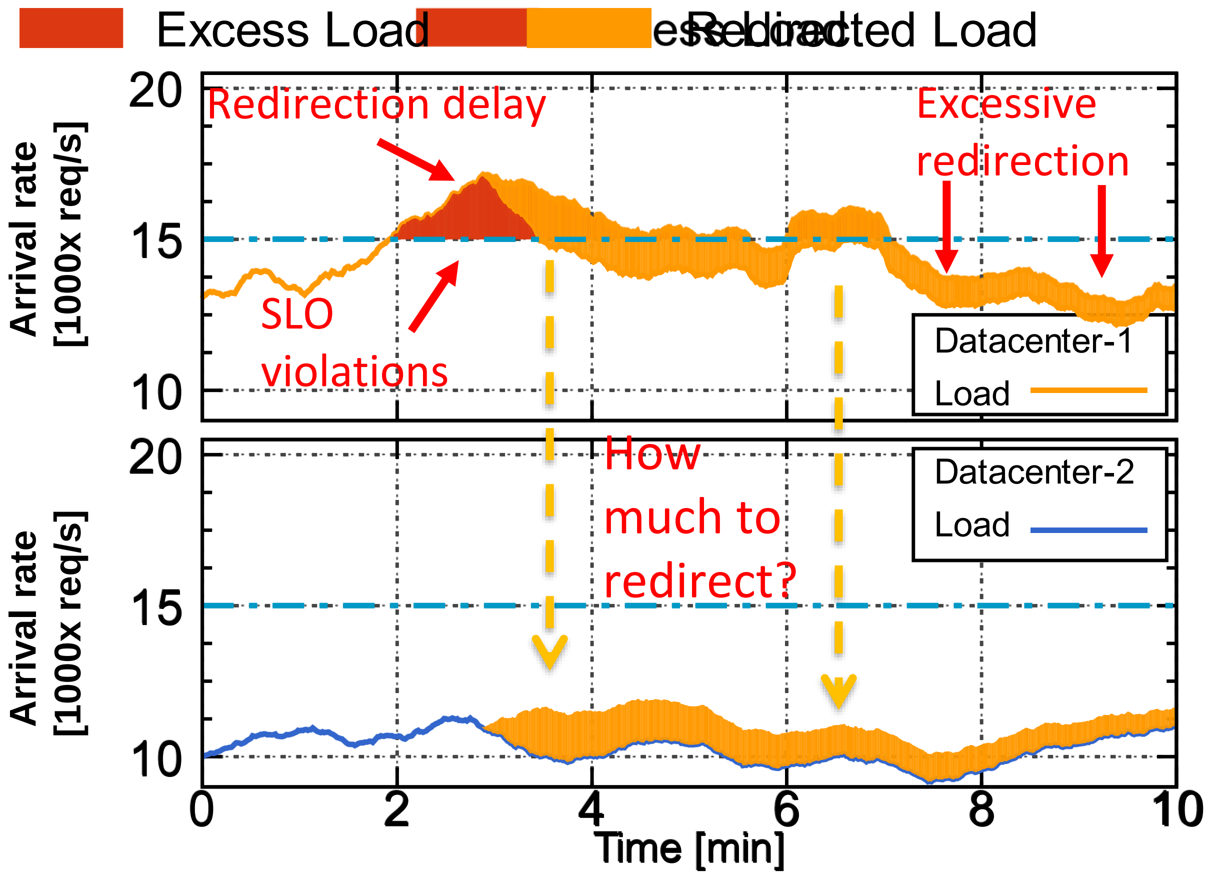
Approach 1 - Datacenter Elasticity



Lead to overprovisioning!

- 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

Approach 2 - Geo-Distributed Load Balancing

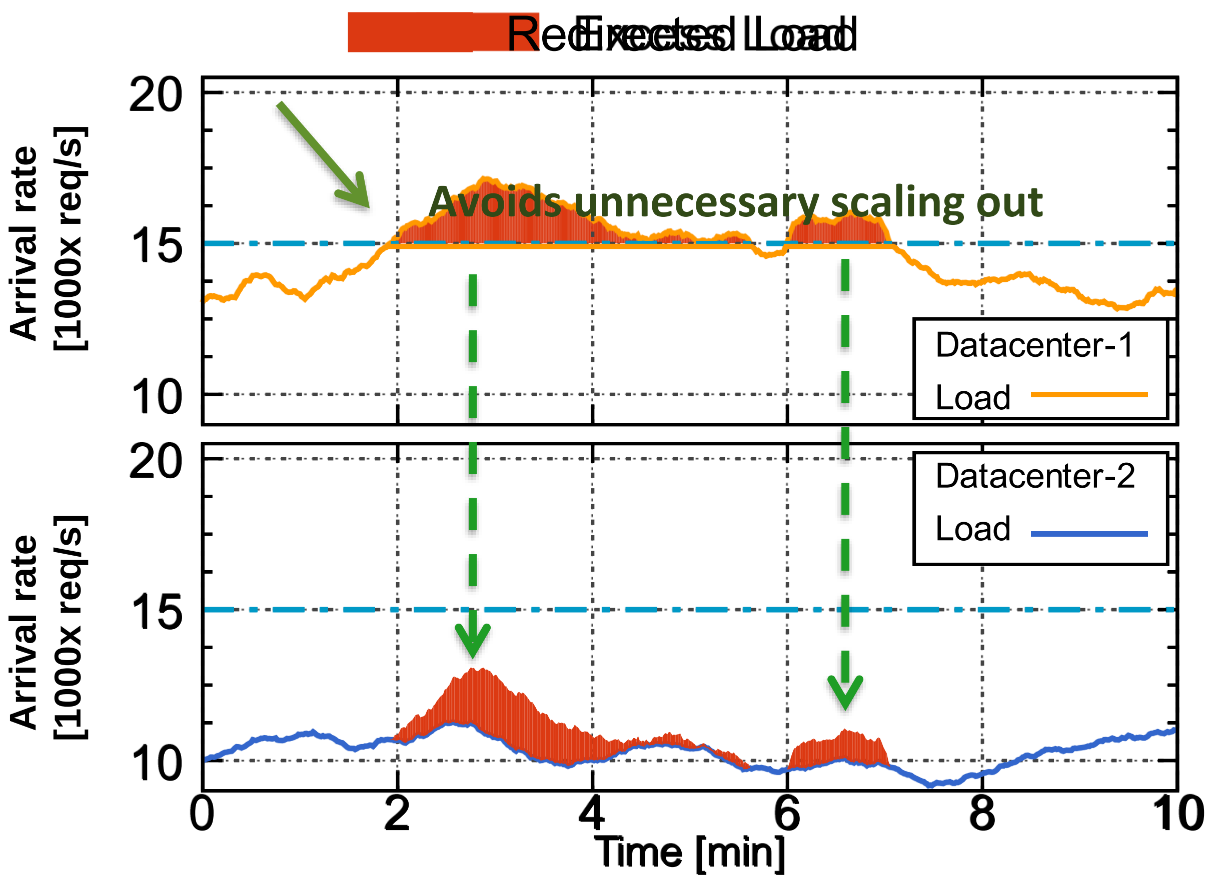


Redirection delays

Inaccurate response time estimation

Excessive or insufficient redirection

Our Approach: Kurma

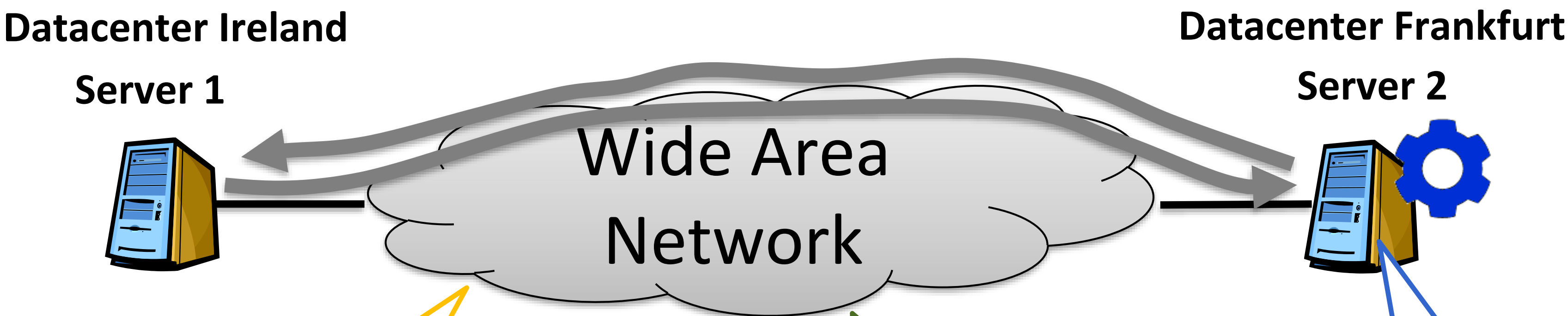


Reacts to changes in load within seconds

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

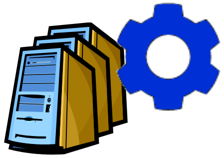
Service Time: Variable component associated with load on the server

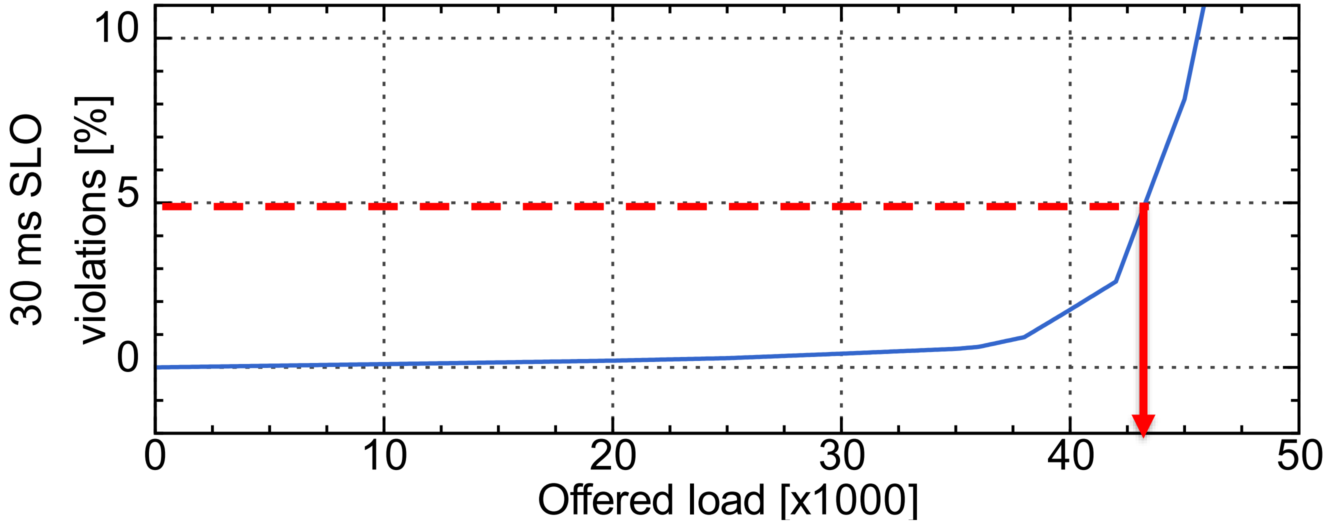
Kurma solves global optimization model while considering:
Base Propagation + Delay Variance + Service Time
at *all* datacenters

Understanding Service Time

Datacenter Frankfurt

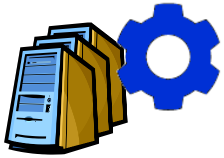
5 Server Cassandra cluster

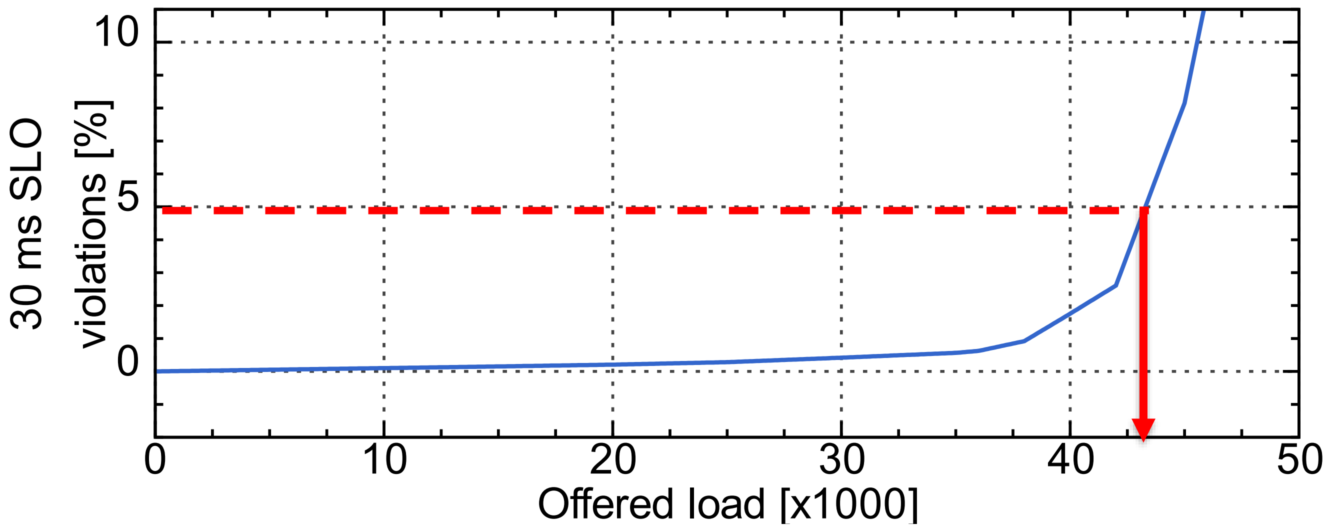




Understanding Service Time

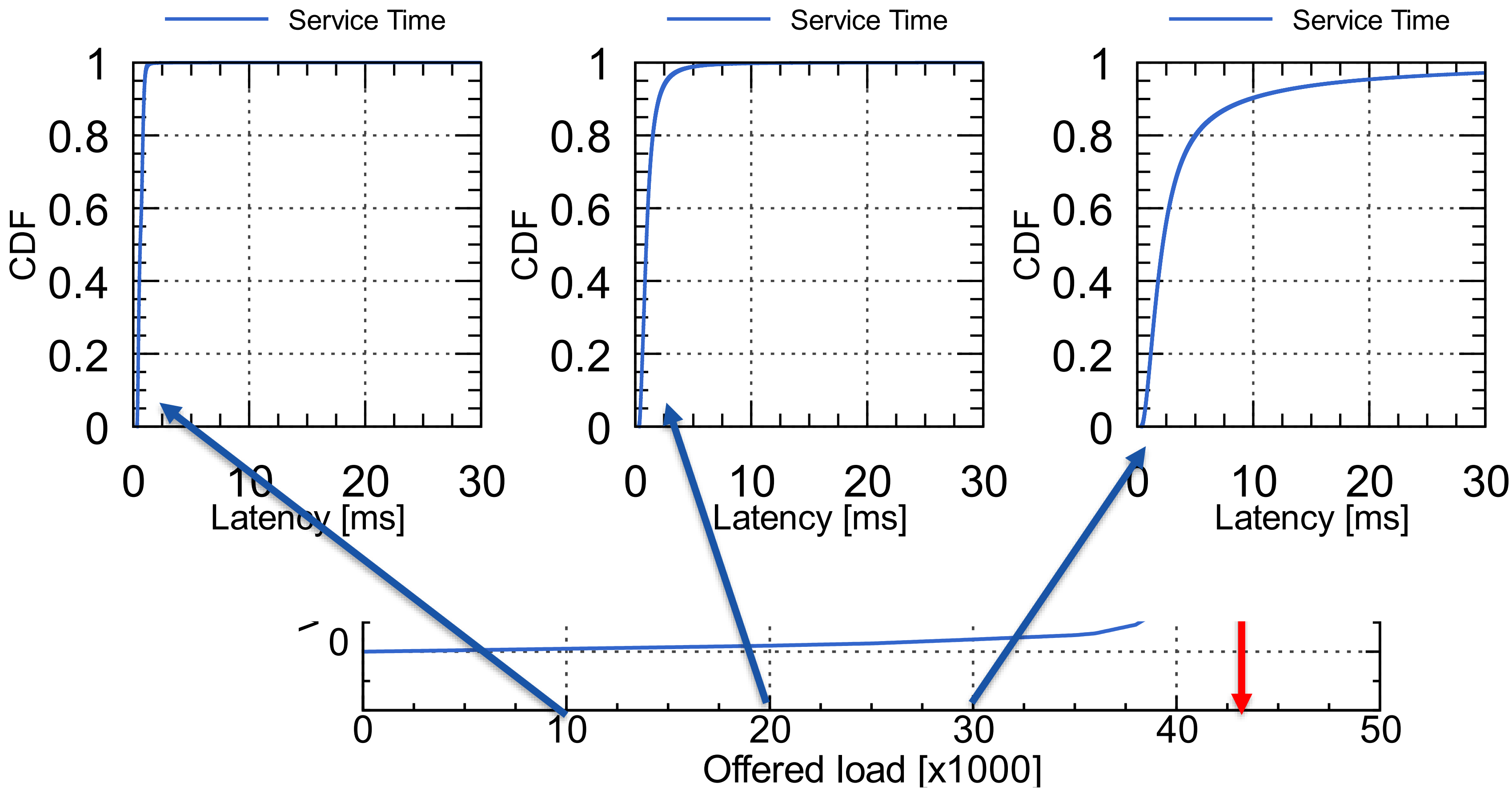
Datacenter Frankfurt
5 Server Cassandra cluster



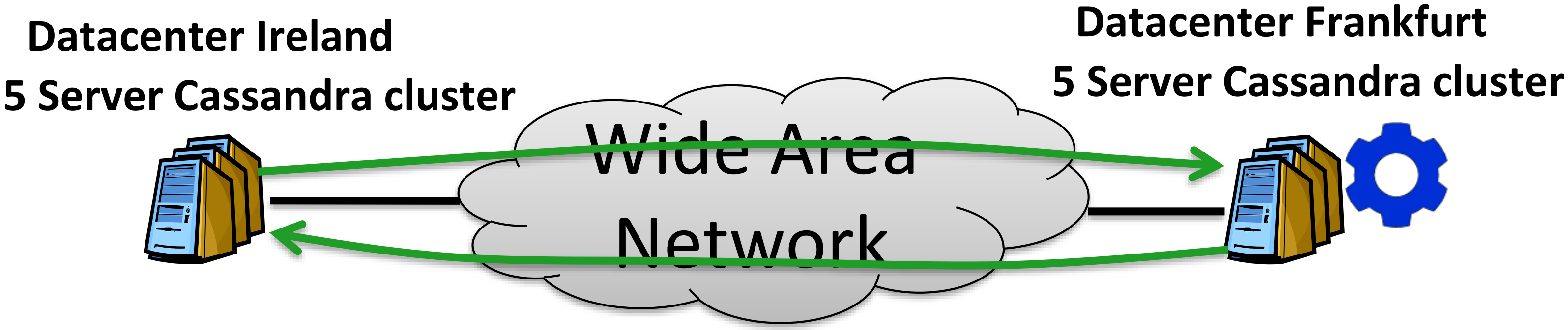


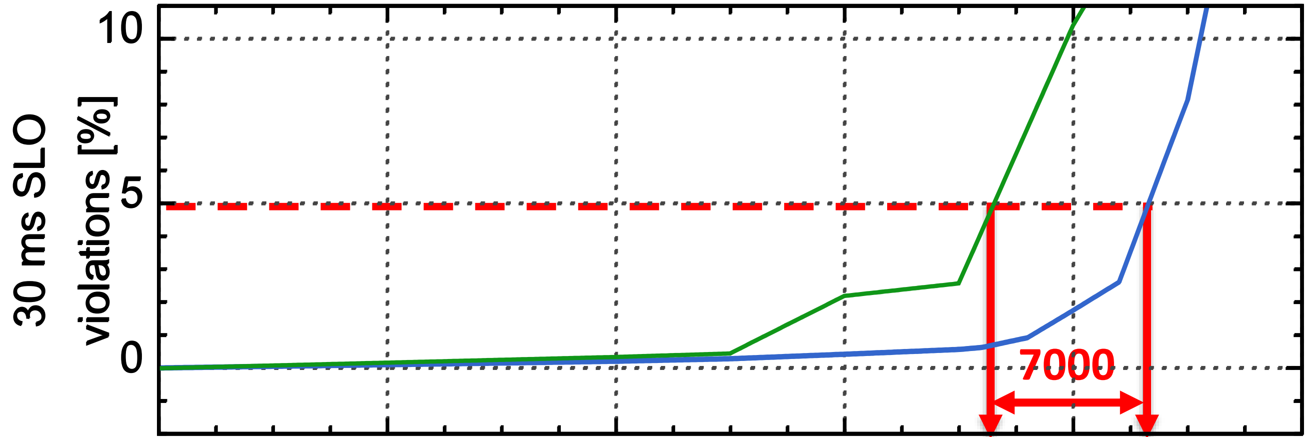
Challenge: How to *accurately* estimate remote fraction of SLO violations at runtime under variable network conditions?

Understanding Service Time



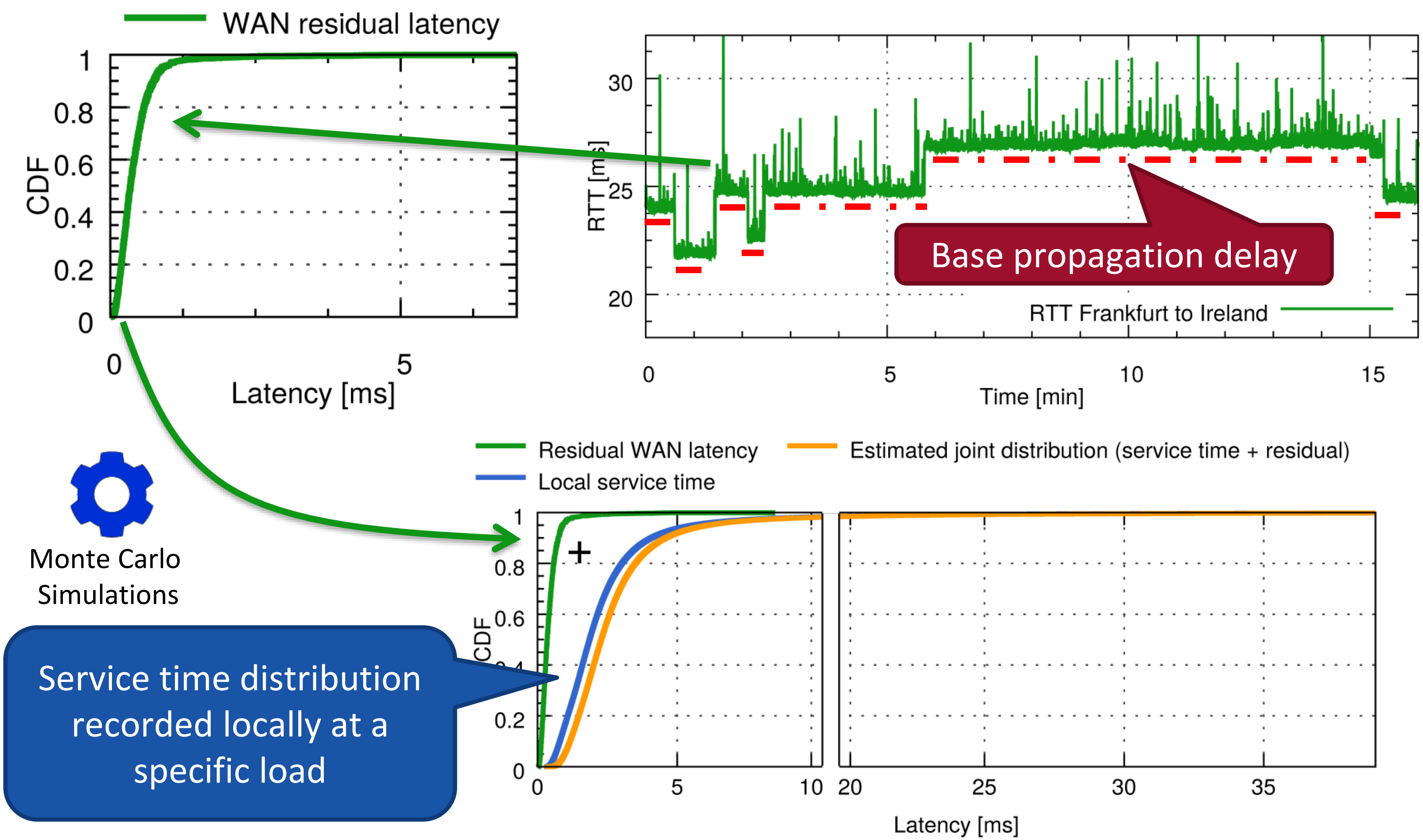
Understanding Service Time



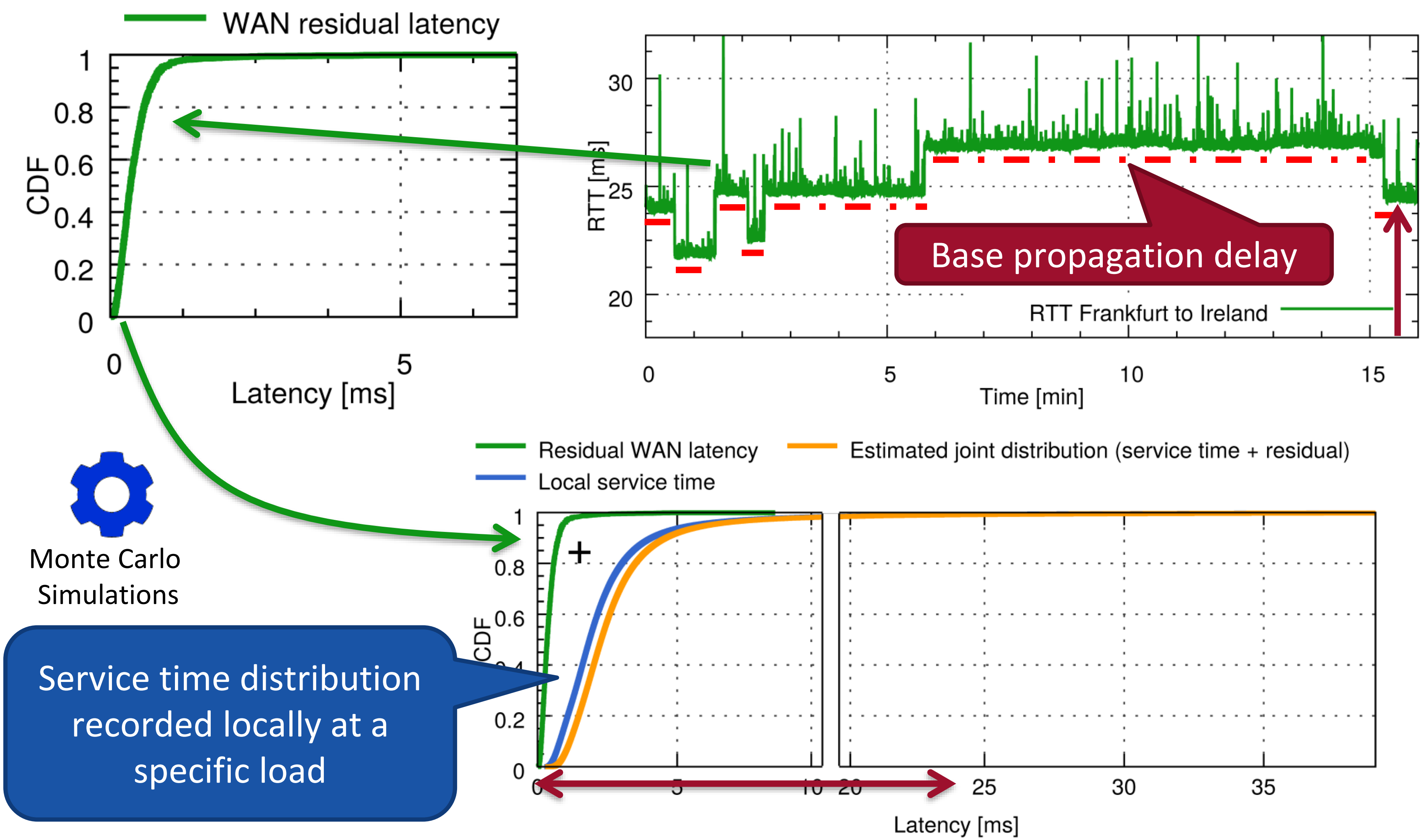


Insight: the farther away a remote datacenter is, the less loaded it should be to serve remote requests within a given SLO target

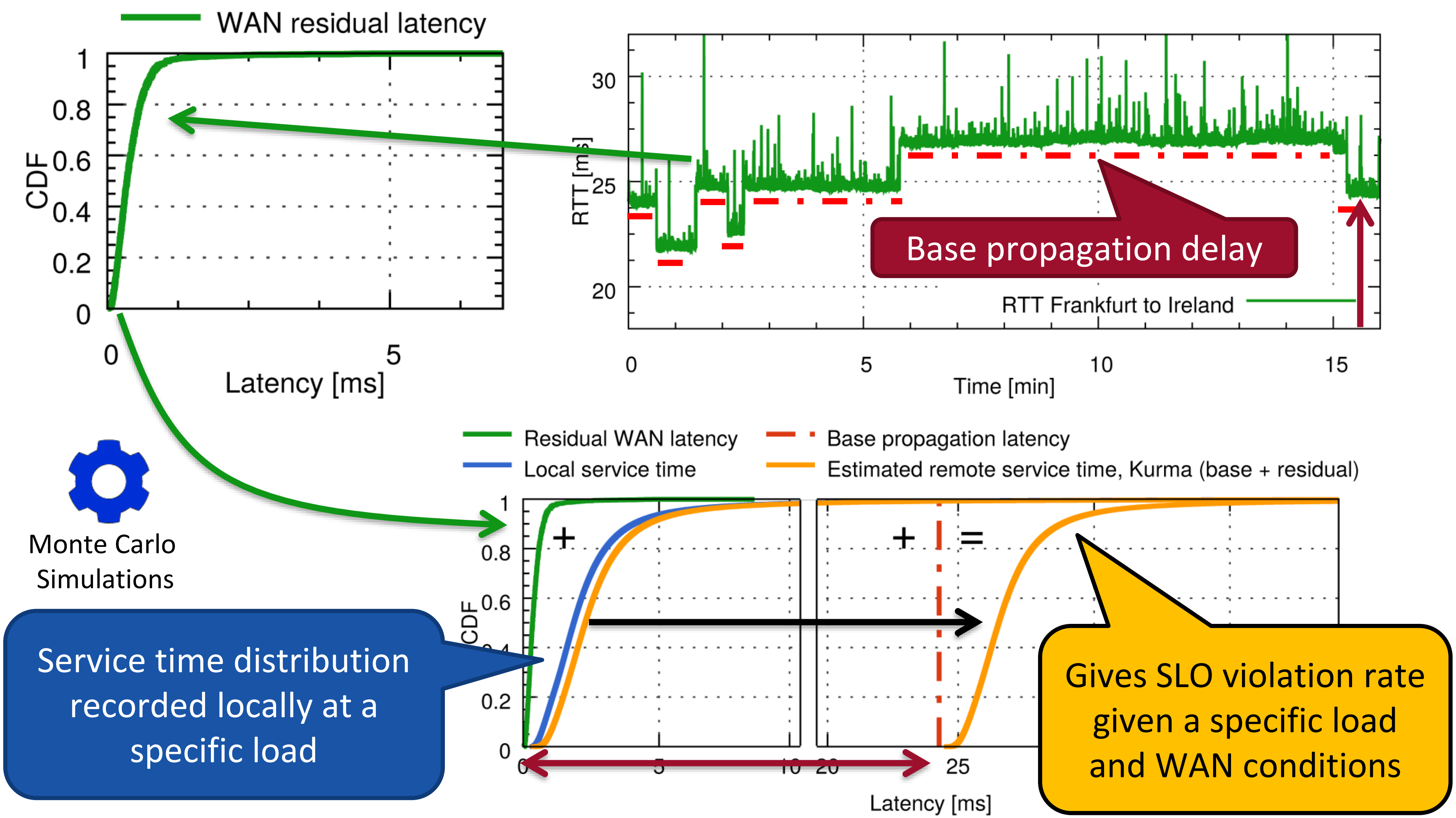
Understanding WAN Latency



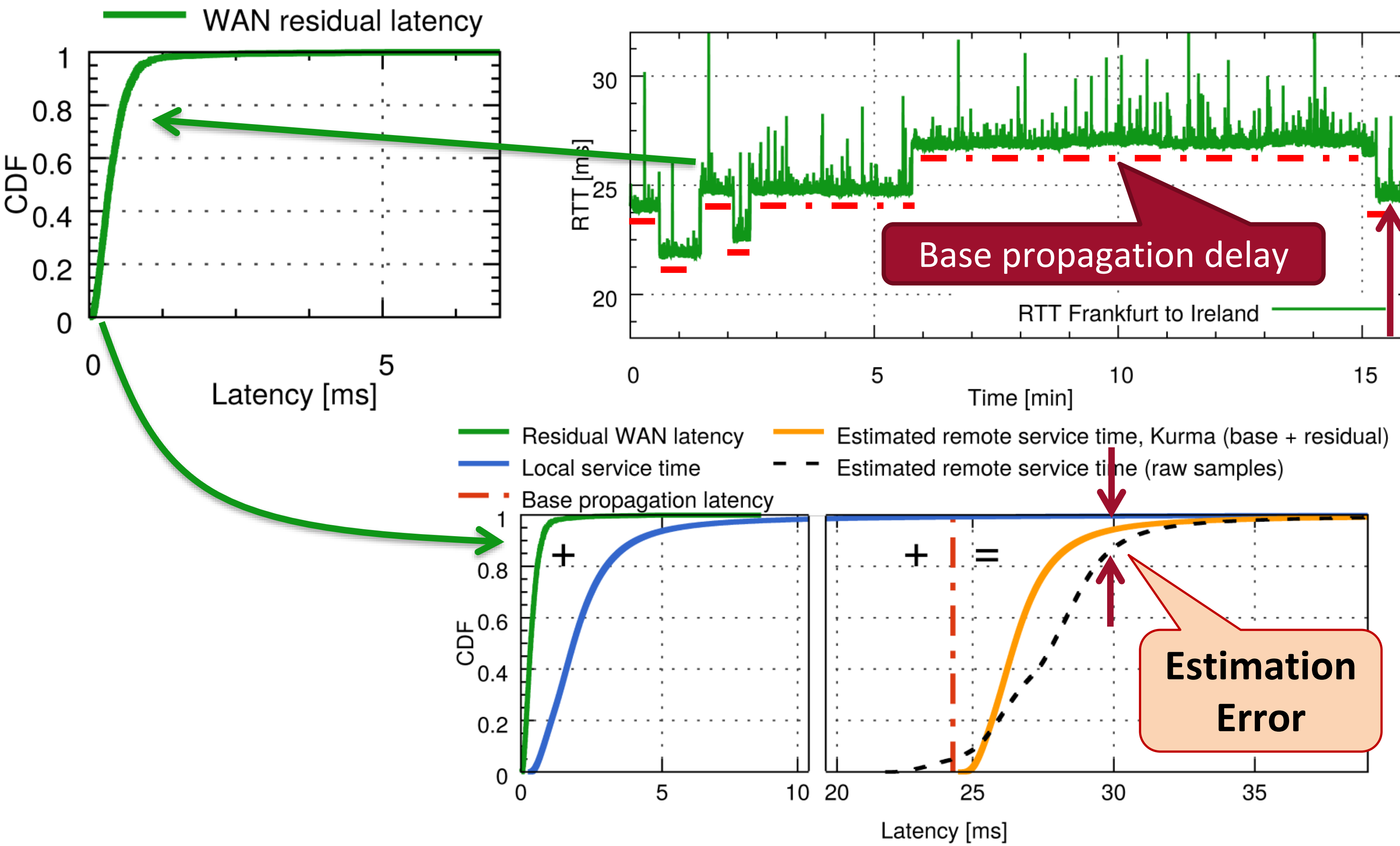
Understanding WAN Latency



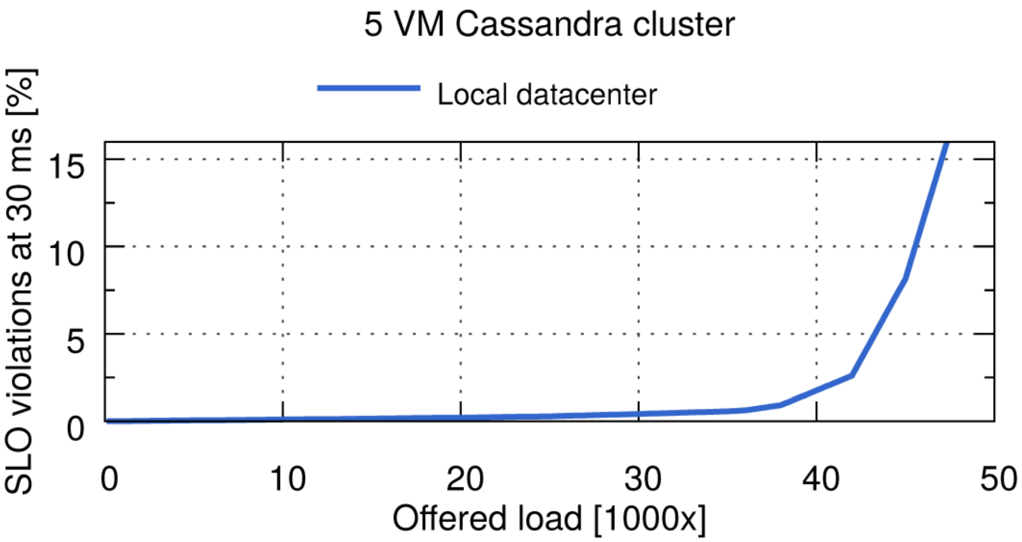
Understanding WAN Latency



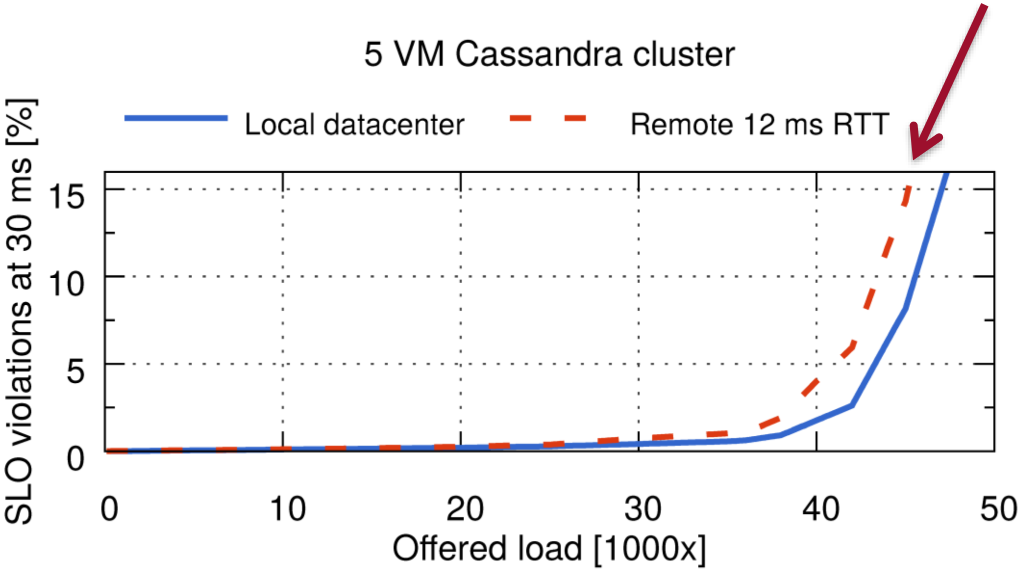
Understanding WAN Latency



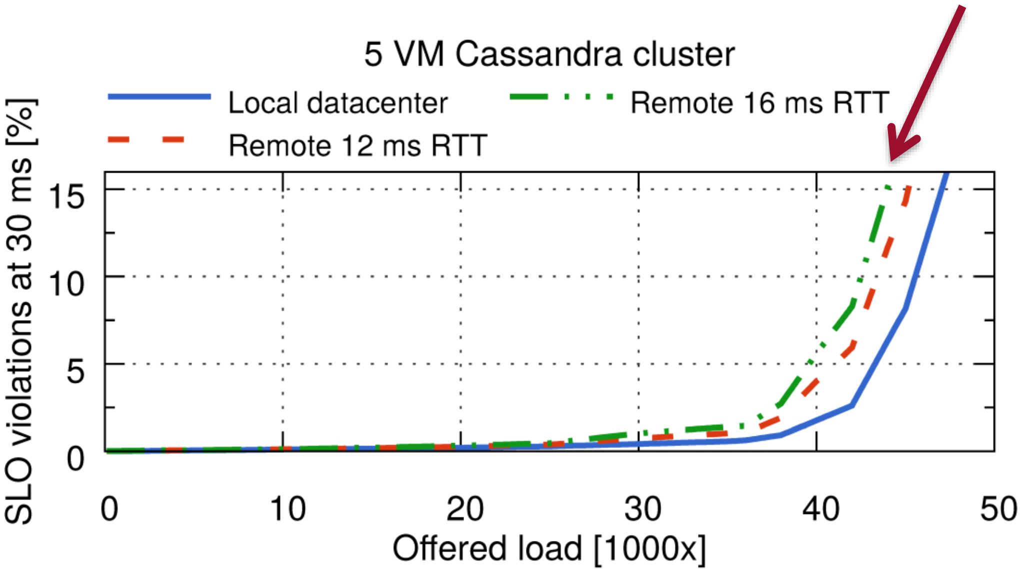
Incorporating WAN and Load



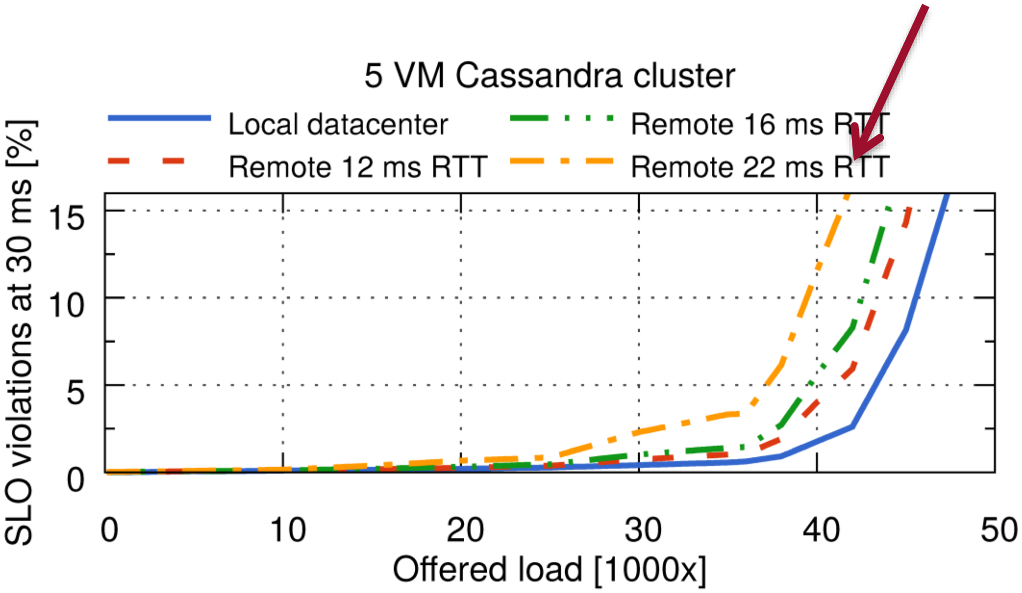
Incorporating WAN and Load



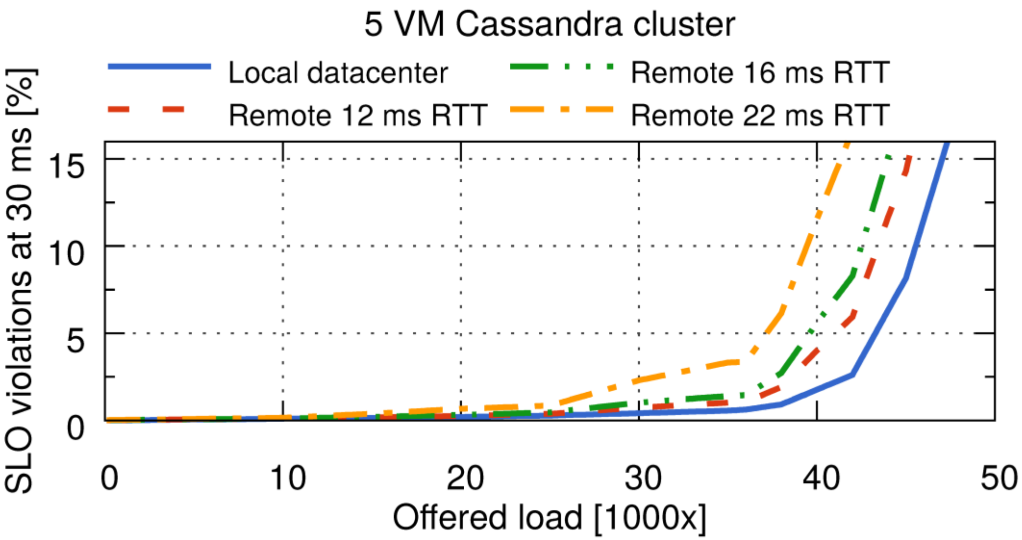
Incorporating WAN and Load



Incorporating WAN and Load



Optimisation Model



+

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

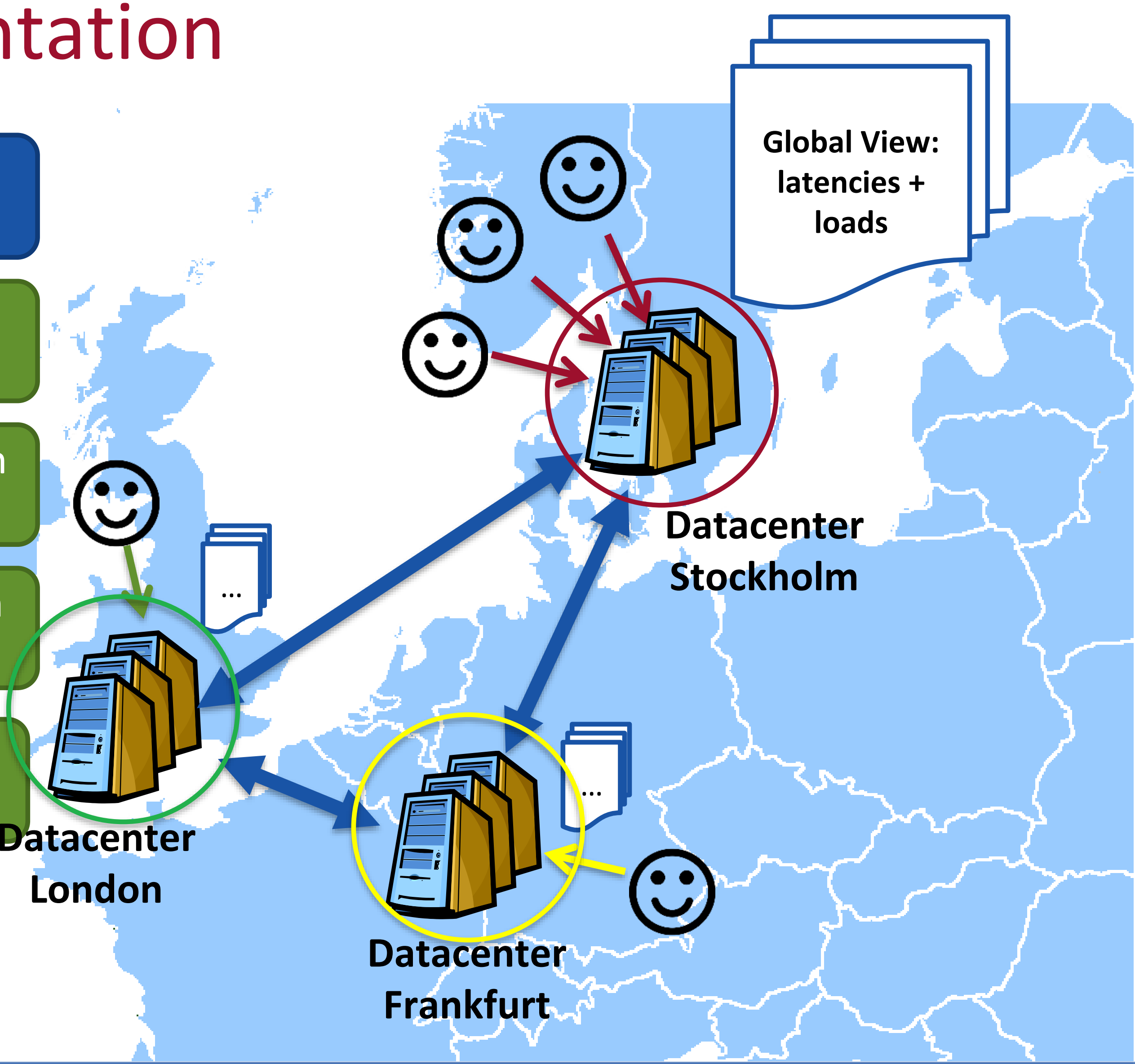
Optimisation Problem

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

Implementation

Each Epoch
2.5 sec → 0.4Hz

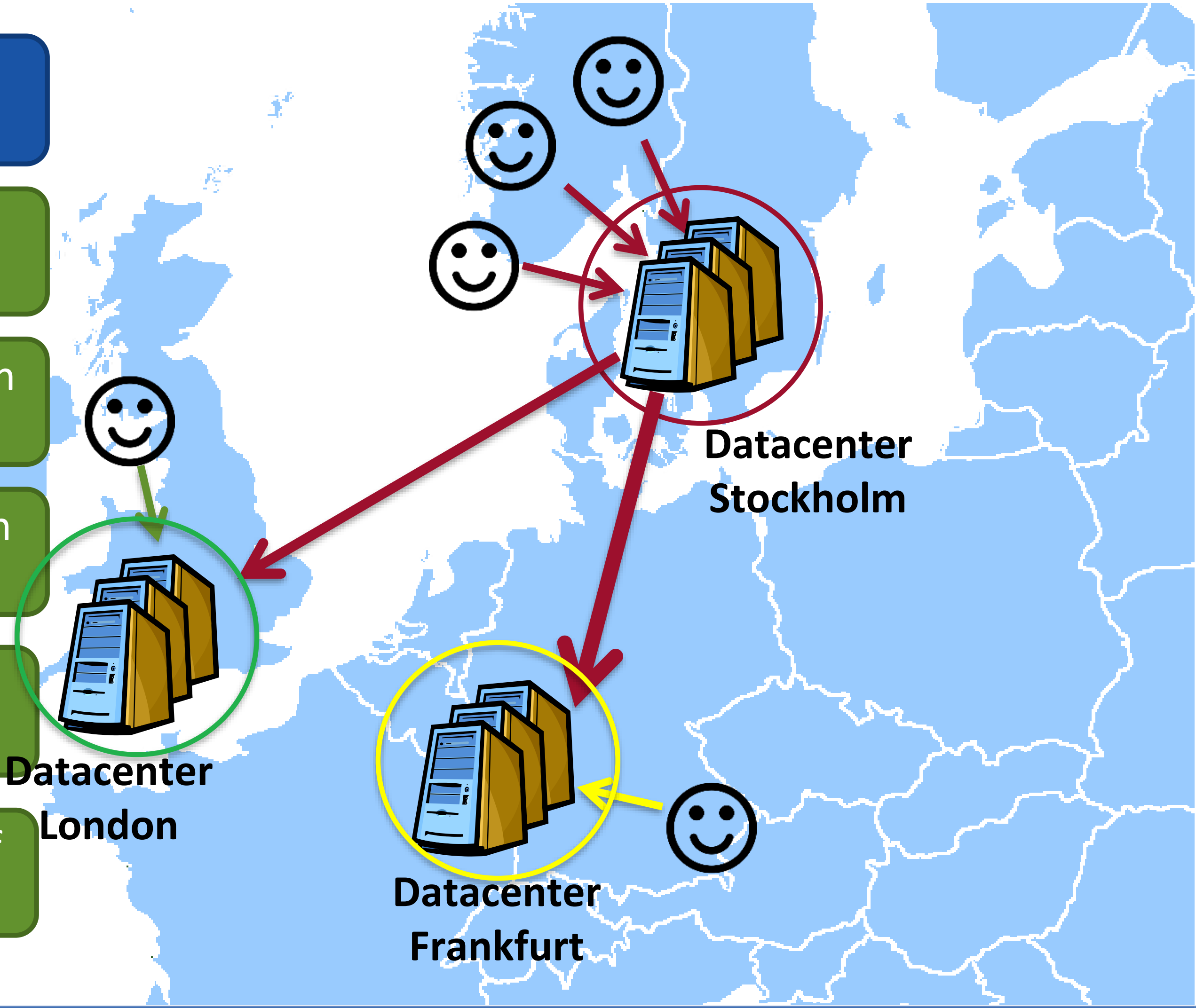
- Perform run-time WAN latency measurements
- Aggregate load information (rates of requests)
- Exchange metrics to obtain global view
- Solve decentralized performance model



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

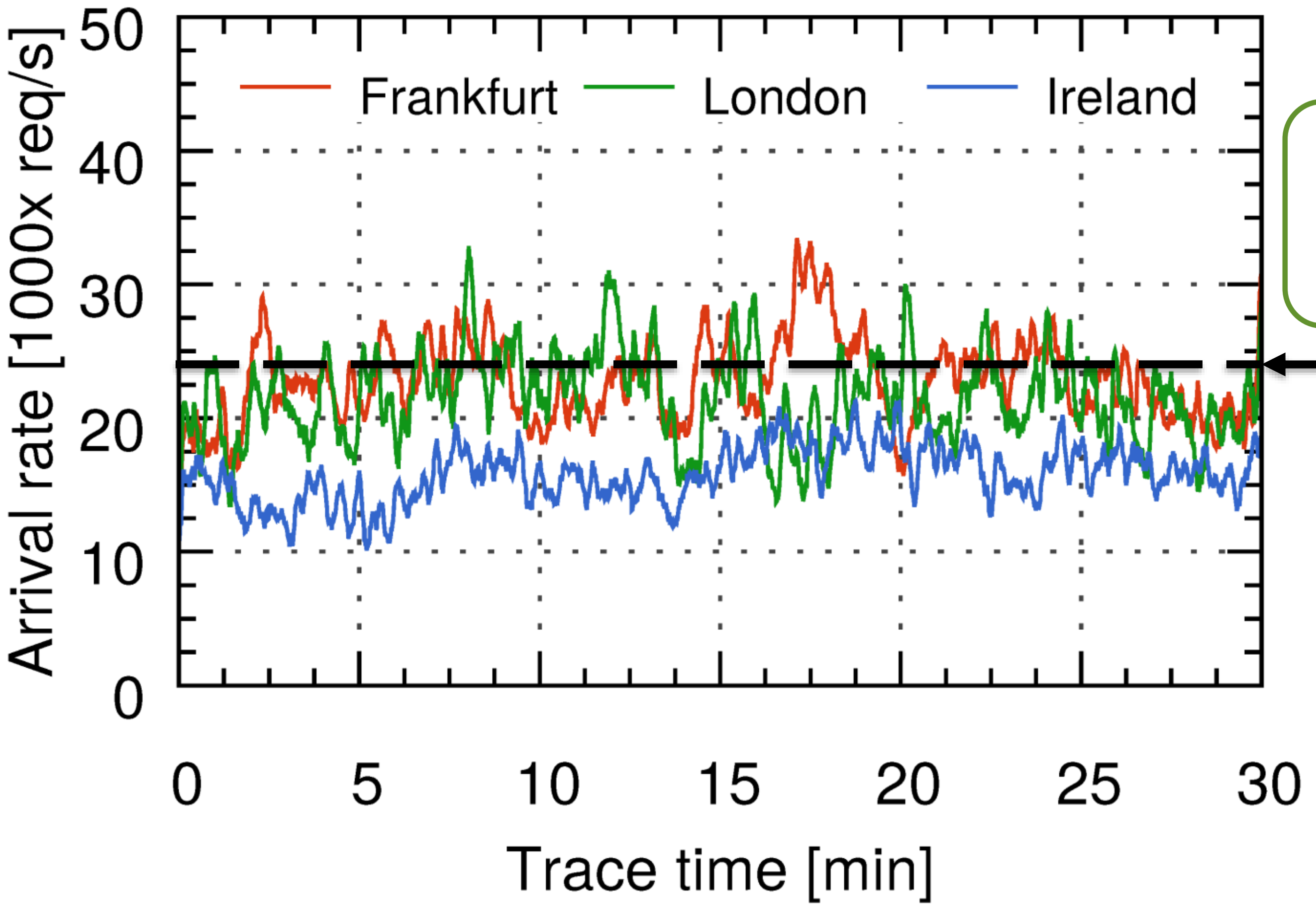
- 3 Amazon EC2 datacenter (Ireland, Frankfurt, London)
- 5 x r5.large VMs per datacenter
- SLO: 30 ms at the 95th percentile
- Modified YCSB to replay workload traces
  (World Cup http://ita.ee.lbl.gov/html/contrib/WorldCup.html)

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.

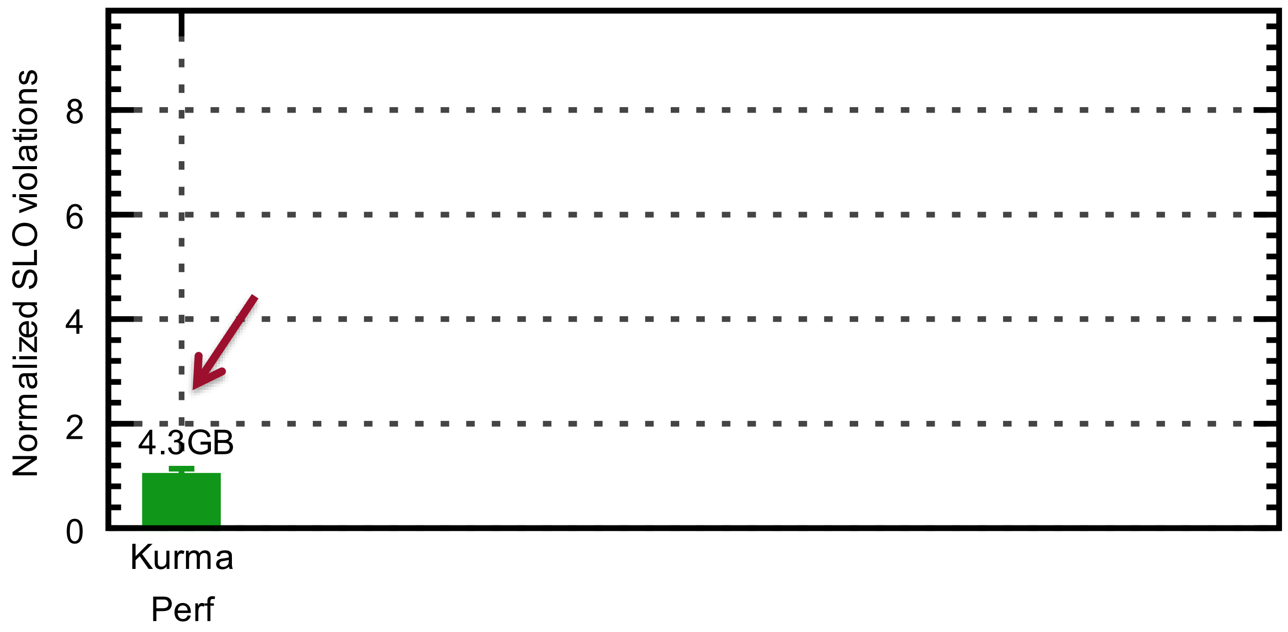
Workload Trace

No elastic scaling



Load threshold for 5% SLO violations

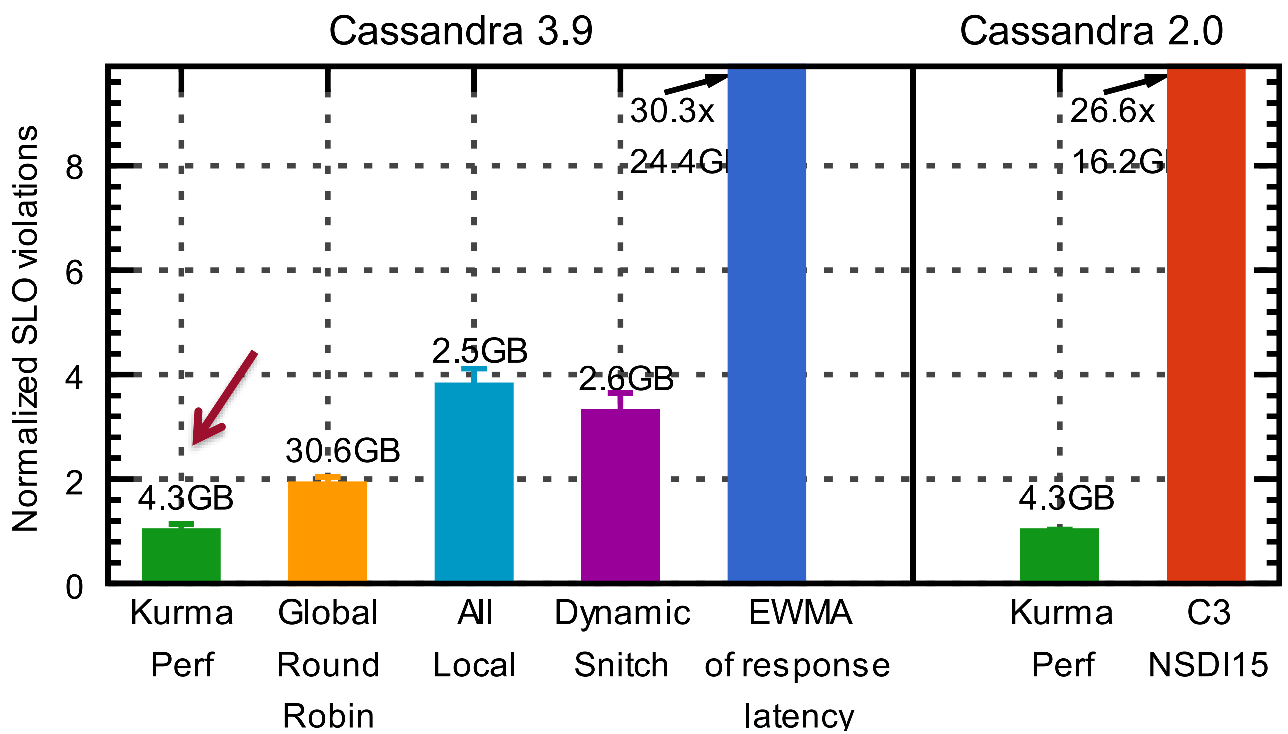
Cumulative Normalized SLO Violations



Kurma's SLO violations are at 2.4%

The numbers shown above the bars indicate the amount of inter-datacentre traffic transferred, whiskers → 75th percentile

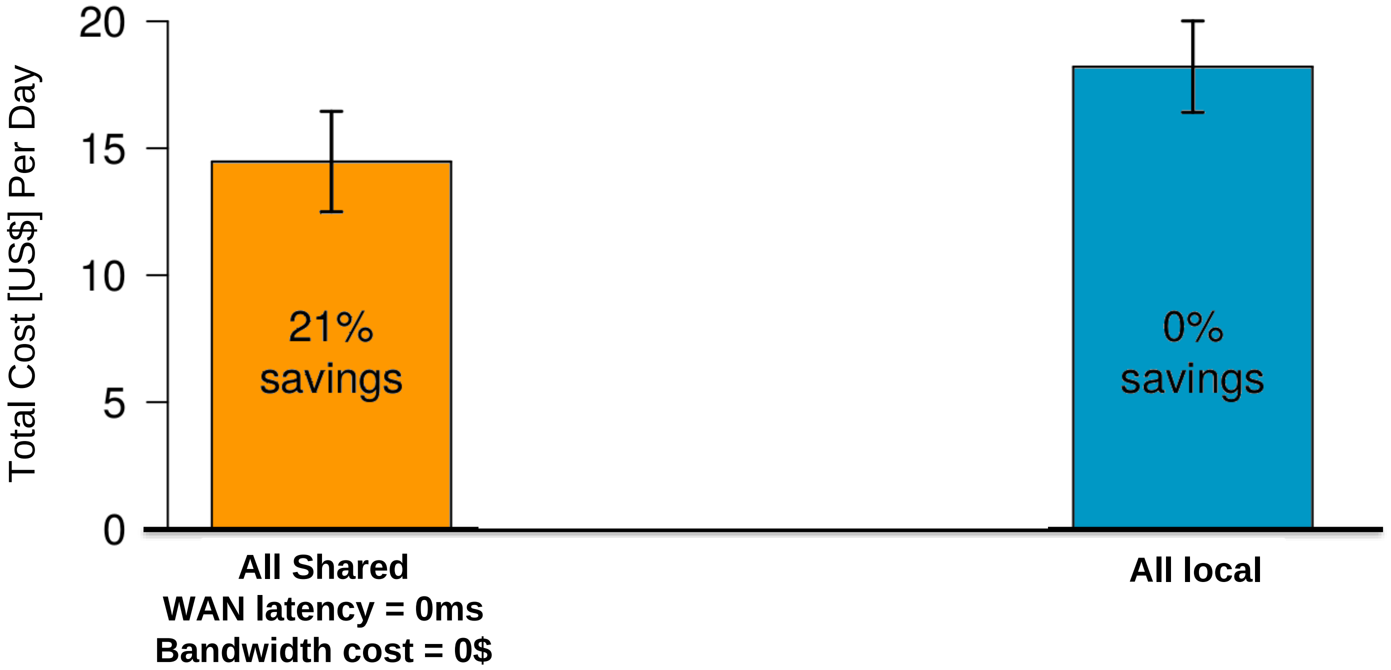
Cumulative Normalized SLO Violations



Kurma's SLO violations are at 2.4%

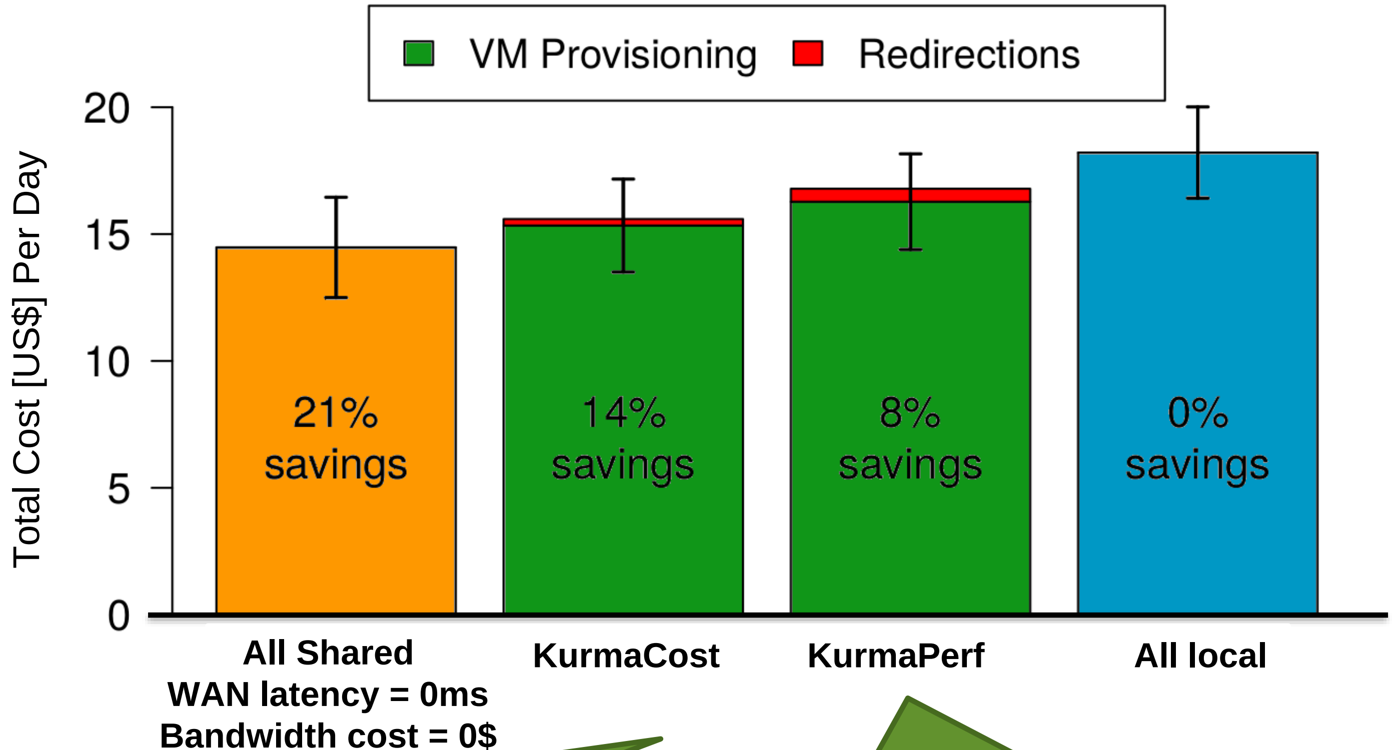
The numbers shown above the bars indicate the amount of inter-datacentre traffic transferred, whiskers → 75th percentile

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

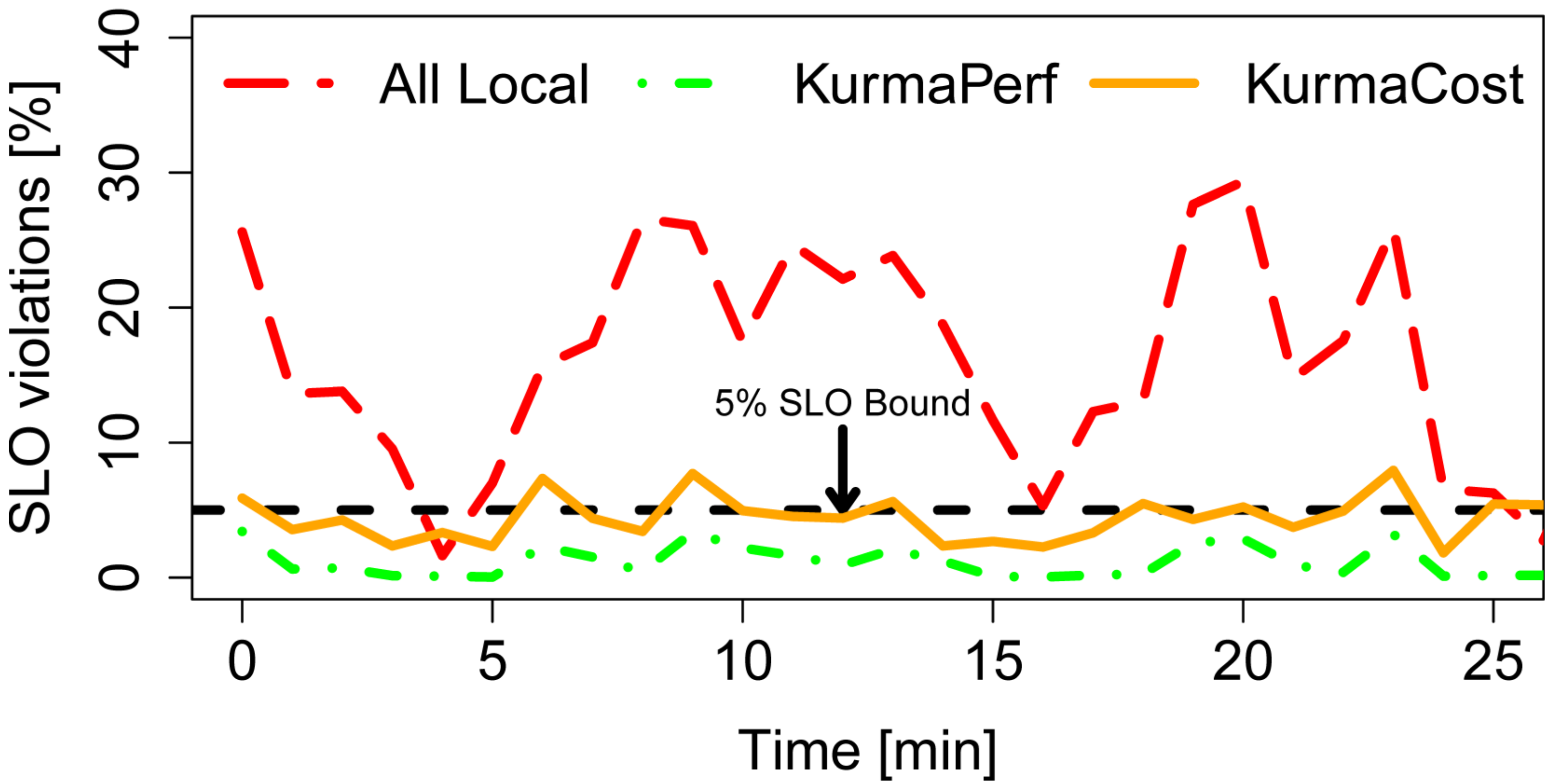


Keeps SLO violations under 5%
(minimize redirections while avoiding scaling out)

Minimize SLO violations
(no consideration for traffic usage)

Taming SLO Violations Under Elastic Threshold

No elastic scaling



Conclusion

Q&A

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