When to Hedge in Interactive Services

Mia Primorac (Oracle*),
Prof. Katerina Argyraki (EPFL),
Prof. Edouard Bugnion (EPFL)

NSDI ‘21, April 13th 2021

* Work done while at EPFL
Our focus: OLDI internet services

• On-Line Data-Intensive services
  • e.g., web search, advertising, social networking, recommendation services
"On-Line" in OLDI services

Network latency

Datacenter latency

Internet

Service

Datacenter
“Data-Intensive” in OLDI services

Request completion time > \( \max(\text{query completion times}) \)
Sources of latency variability

Query completion time

Queuing

Service time

Predictable component (P)

IQ-jitter (J)
Intra-query service time variability (IQ-jitter)
IQ-jitter is real


Scheduling for tail latency despite IQ-jitter?

- **Load balancing** reduces queuing delays
  - *Push-based* (distributed queuing) mitigates network latency
  - *Pull-based* (centralized per-shard queuing - PSQ) → optimal queuing delays within a shard if network latency is negligible

- **Hedging** reduces service time, but creates extra load
A Naïve Hedging Policy
Let’s model an OLDI application

• Discrete-event simulation with:
  • 50 shards × 2 replicas
  • Exponential $P$
  • IQ-jitter due to random performance “hiccups” : $1/1000$ queries, $15 \times \text{AVG}(P)$

Similarly to Mirhosseini et al. "The Queueing-First Approach for Tail Management of Interactive Services", IEEE Micro
Naïve hedging vs the best load-balancing
If Idealized Hedge does not yield benefits on a certain workload, no realistic hedging policy will do so either.
When is hedging a good idea?

The potential of hedging = the best LB – Idealized Hedge
How close to Idealized Hedge can we get?
State of the art

Can we get even closer to Idealized Hedge?

Load-Aware Hedge (LÆEDGE) – properties

- A queue per shard (pull-based)
- Work conservation with redundancy (best-effort)
- Prioritizes new over duplicated queries (best-effort)
- 3 variants depending on the type of cancellations (no cancellations, CC, CC+PC)
- No parameter tuning
LÆDGE with and without cancellations

![Graph showing 99th percentile latency vs. Utilization]

- PSQ LB
- Best d-Hedge w/ CC
- LÆDGE
- LÆDGE w/ CC
- Idealized Hedge

Best existing
System evaluation in AWS

• Scheduling policies implemented in Google’s OLDIsim framework

• **Workload**: Lucene search engine trained on English Wikipedia
  • *Nightly* benchmark (10k queries)

• **Deployment**:
  10 leaves (5 shards × 2) in 10 AWS VMs (16 vCPUs) running vanilla Ubuntu 16.04
System evaluation in AWS

System results closely match our simulation results despite realistic network latencies.
More information in the paper:

• Benefits of Idealized Hedged and LÆDGE increase with the number of replicas
• Scalable architecture with multiple root nodes and multiple per-shard load balancers
• Sensitivity analysis of the potential benefits of hedging wrt cluster sizes and hiccup distributions
• Performance of LÆDGE with CC and PC → very sensitive to pre-emption accuracy
Takeaways

• There are many realistic scenarios where hedging does not make sense

• Adding cancellations does not significantly reduce latency (LÆEDGE vs LÆEDGE w/ CC)

• Pull-based policies can outperform push-based ones
Thank you!

Contact:
mia.primorac(at)oracle.com / mia.primorac(at)gmail.com