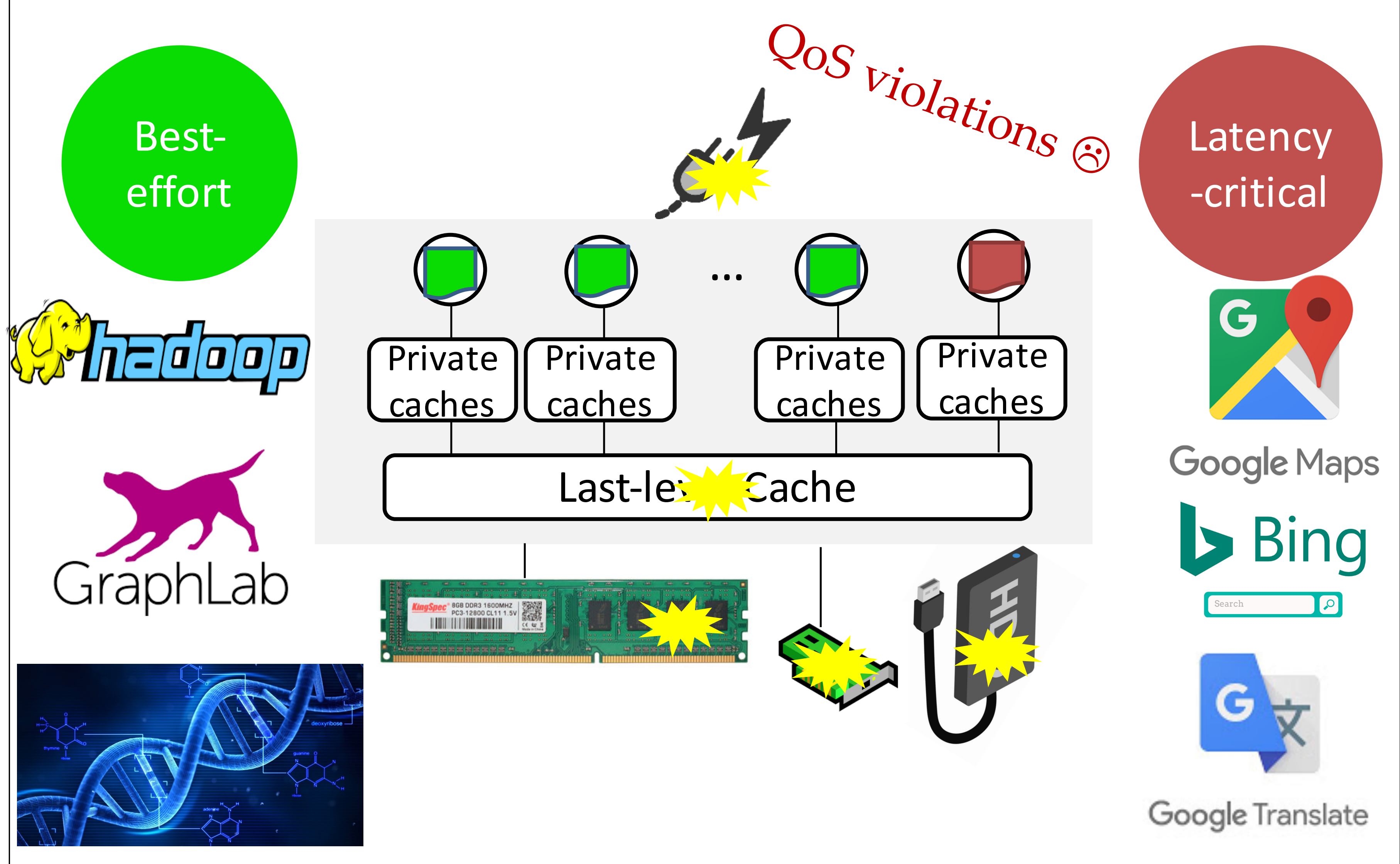
Computer Systems Laboratory
Cornell University

ASPLOS 2019
Shuang Chen
Christina Delimitrou
José F. Martínez

PARTIES: QoS-Aware Resource Partitioning for Multiple Interactive Service

Motivation

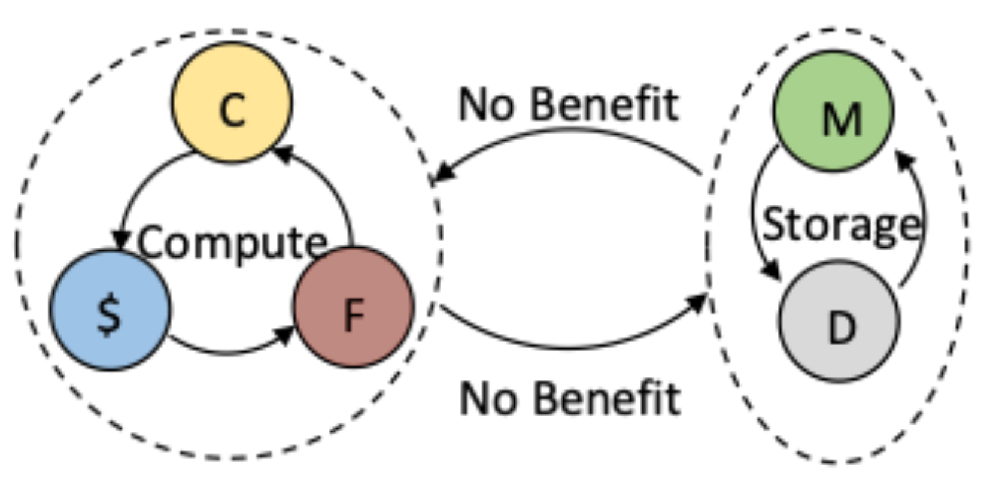


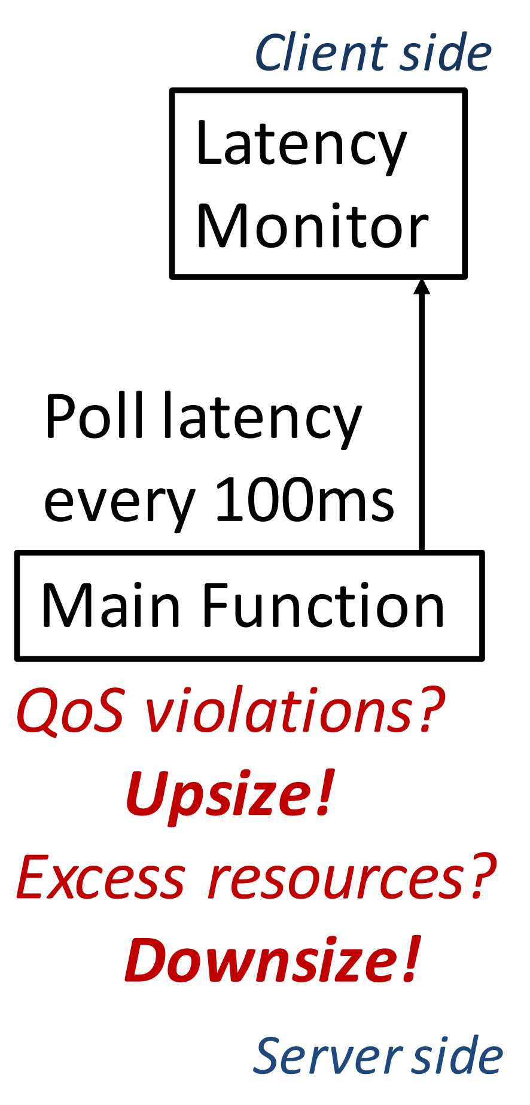
Performance unpredictability from Interference in shared resources leads to QoS violations for LC applications.

PARTIES Design

Design principles:

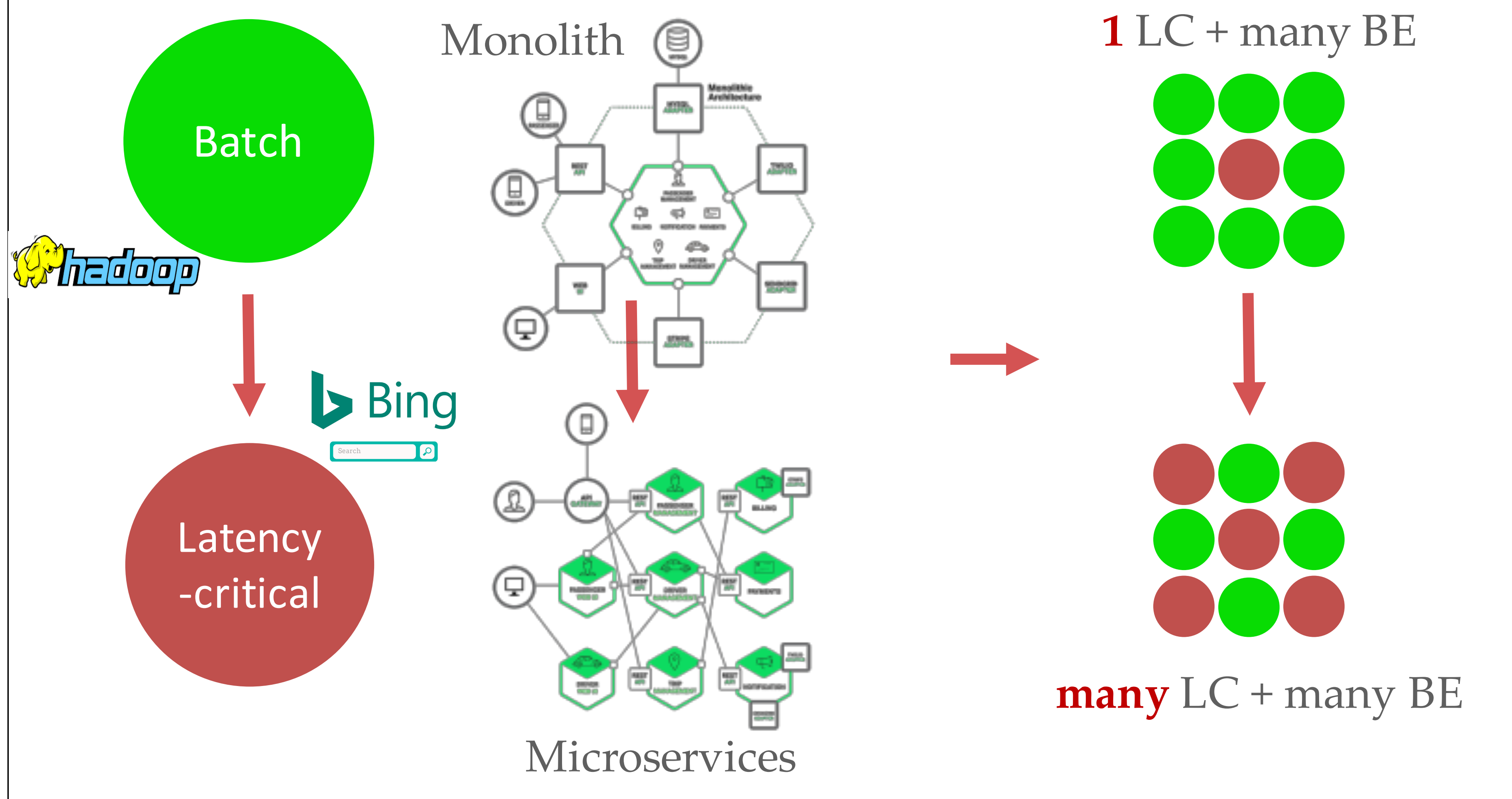
1. All LC applications are equally important.
2. Allocation should be dynamic and fine-grained.
3. No a priori application knowledge or offline profiling is needed.
4. Recover quickly from incorrect decisions.
5. Migration is used as a last resort.





1. 5 knobs organized into two wheels based on functionality.
2. Start from a random resource.
3. Stick to one resource until no benefit.
4. Follow the wheel to visit all resources.

Colocation of Multiple LC Applications

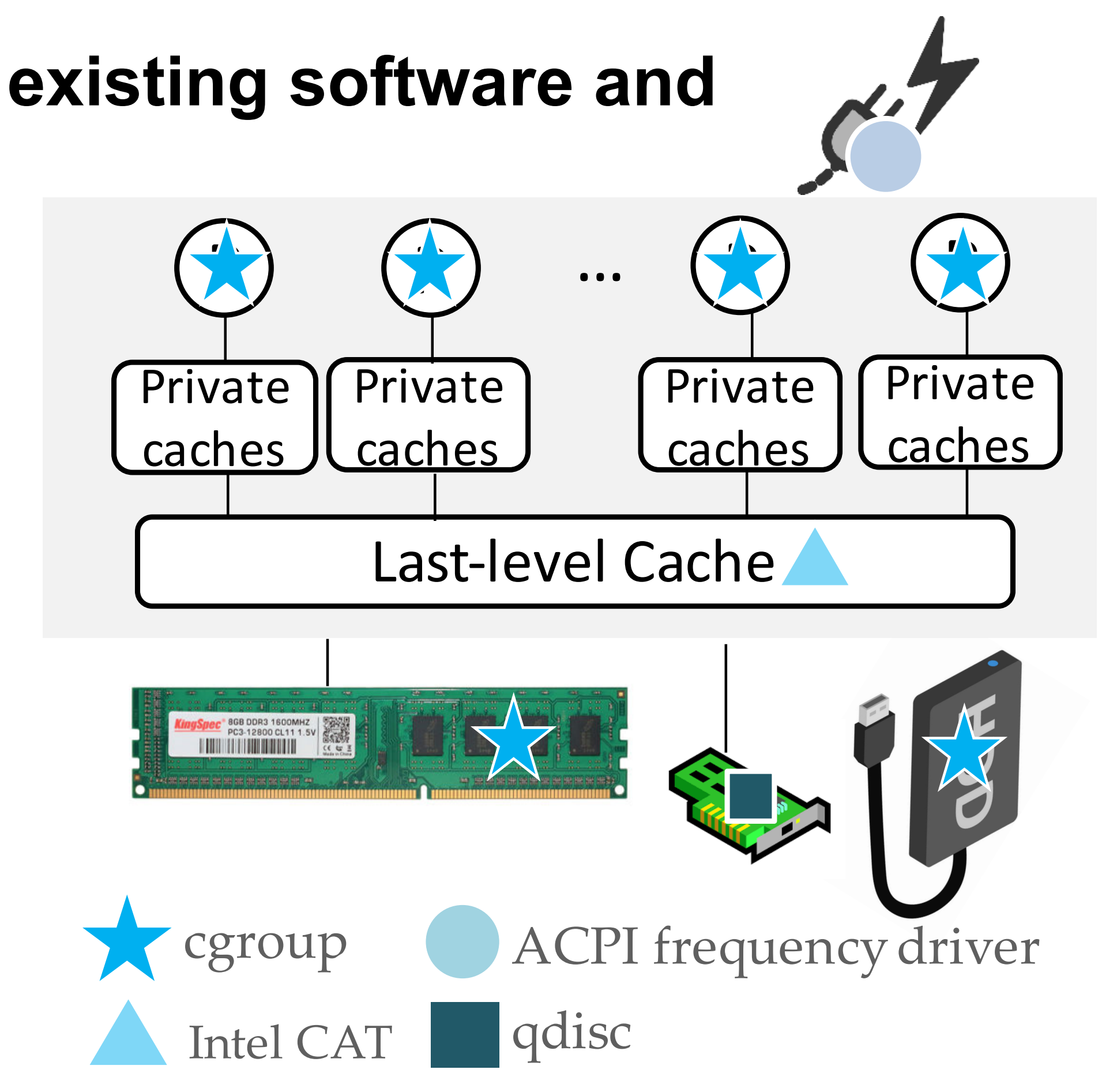


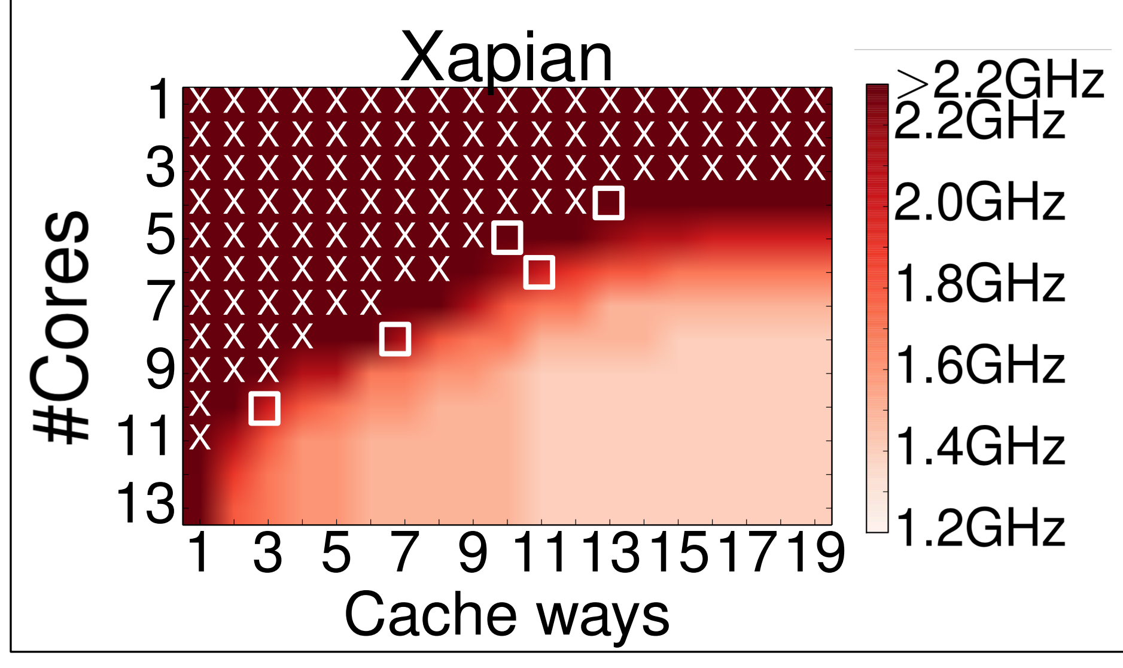
Challenge: all LC services have QoS targets, so none of them can be easily scarified for another.

Isolation Mechanisms

PARTIES leverages all the existing software and hardware isolation mechanisms to partition:

* Cores
  * Hyperthreads
  * Core counts
* Power budget
* Last-level cache capacity
  * LLC bandwidth
  * Memory bandwidth
* Memory capacity
* Disk bandwidth
* Network bandwidth

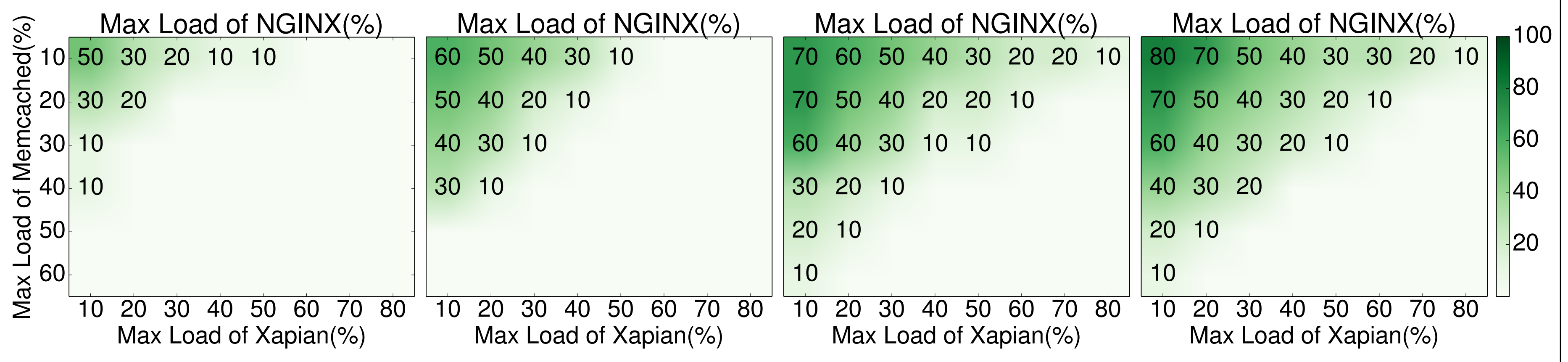




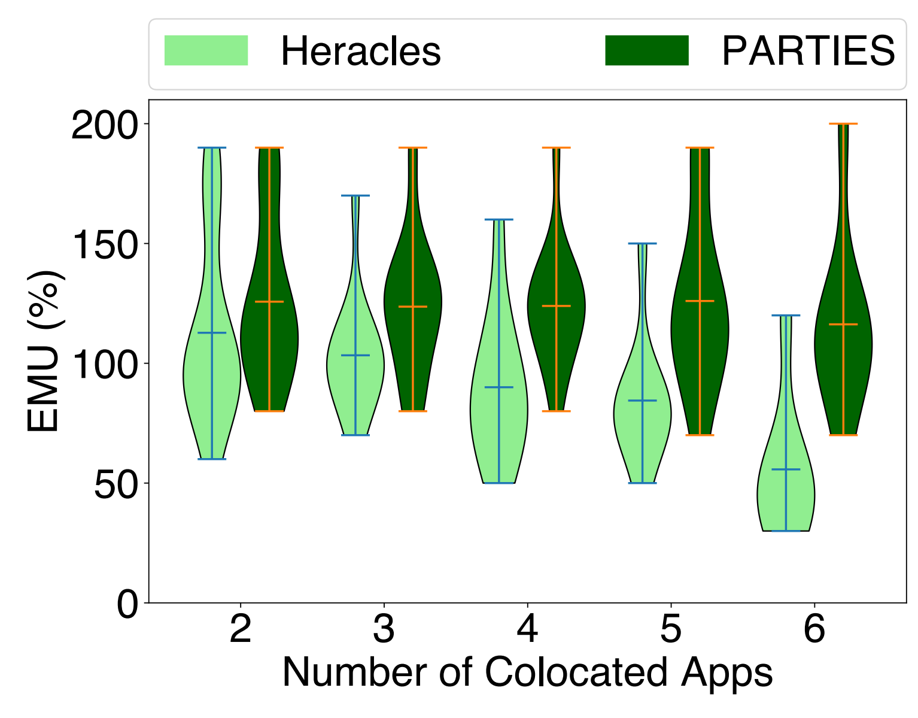
Resource fungibility: resources can be traded with each other.

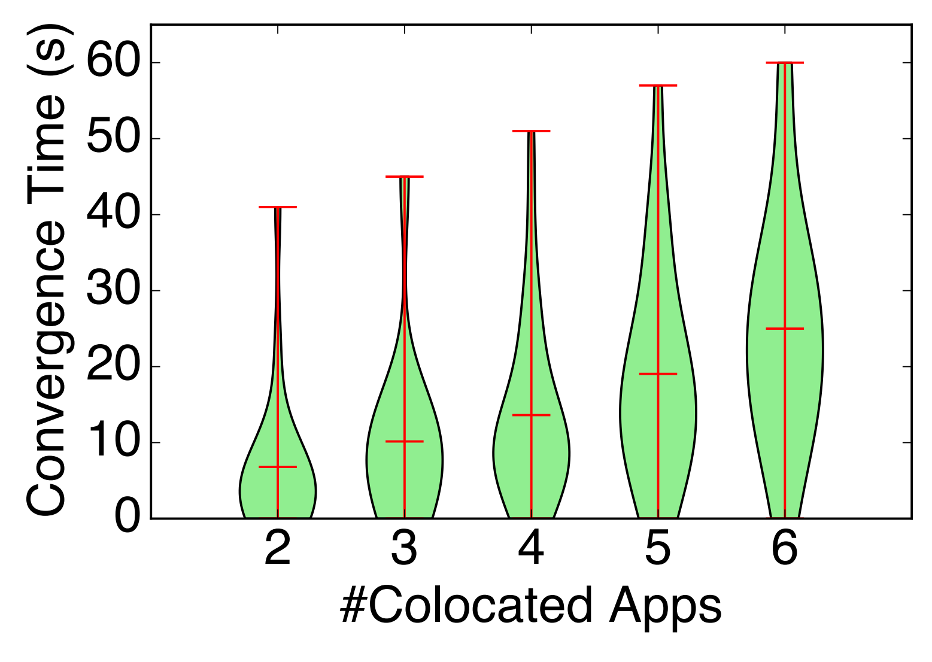
Evaluation

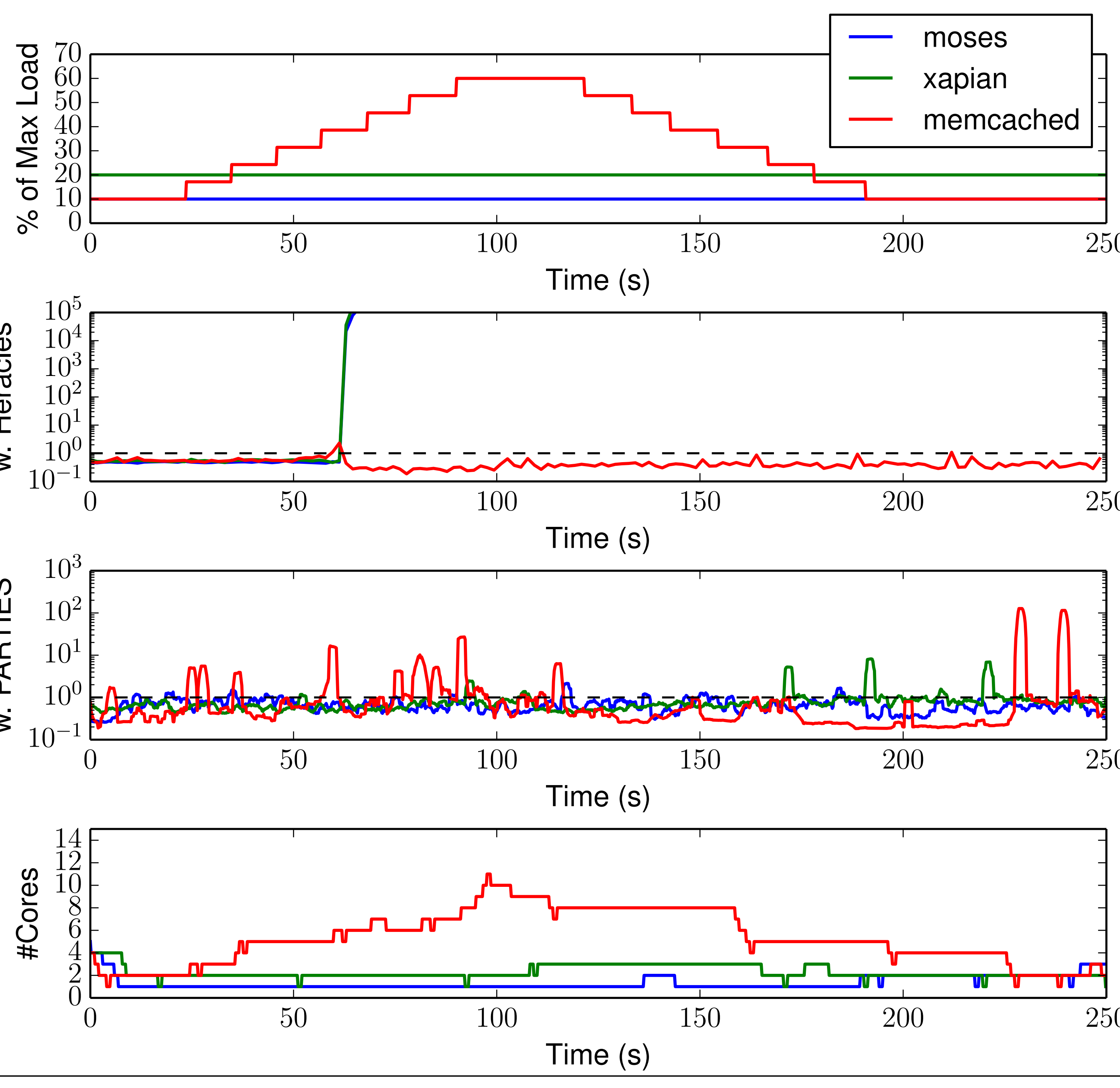
Platform: Intel E5-2699 v4
Benchmarks: Memcached; Xapian; NGINX; Moses; MongoDB; Sphinx



(a) Unmanaged (b) Heracles (c) PARTIES (d) Oracle


Heracles
PARTIES
EMU (%)
Number of Colocated Apps


Convergence Time (s)
#Colocated Apps


% of Max Load
moses
xapian
memcached
Time (s)
Norm. Latency w. Heracles
Norm. Latency w. PARTIES
#Cores