PARTIES: QoS-Aware Resource Partitioning for Multiple Interactive Service

Motivation

Challenges:
1. Performance unpredictability from interference in shared resources leads to QoS violations for LC applications.

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.

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.

Evaluation

Platform: Intel E5-2699 v4

Benchmarks: Memcached; Xapian; NGINX; Moses; MongoDB; Sphinx

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

Performance comparision:
- Classification time
- Convergence time
- Resource utilisation
- QoS violations

Conclusion:
- PARTIES: QoS-Aware Resource Partitioning
- Evaluation
- Migration is used as a last resort.