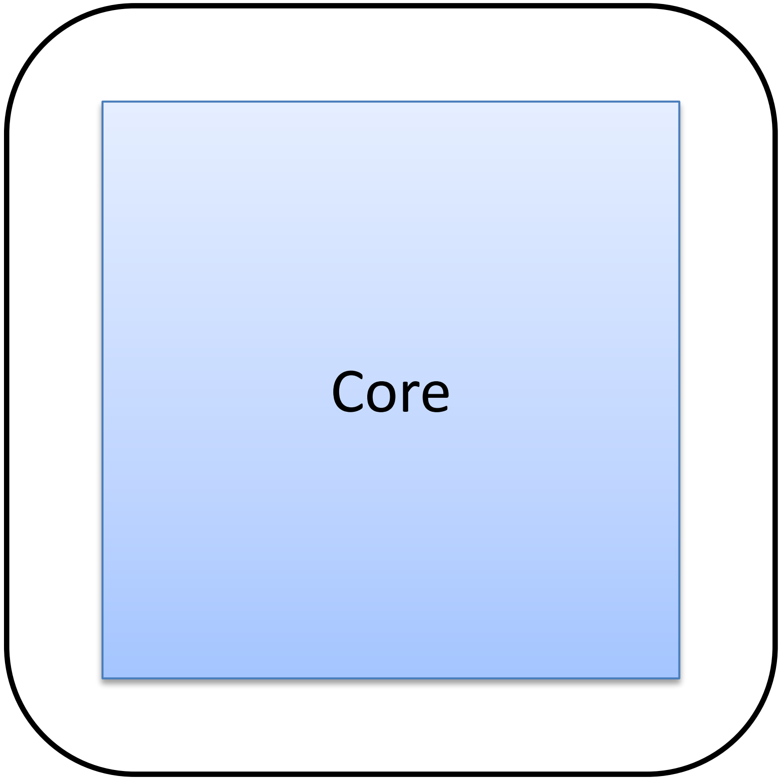
OLTP on Hardware Islands

Danica Porobic, Ippokratis Pandis*, Miguel Branco, Pınar Tözün, Anastasia Ailamaki

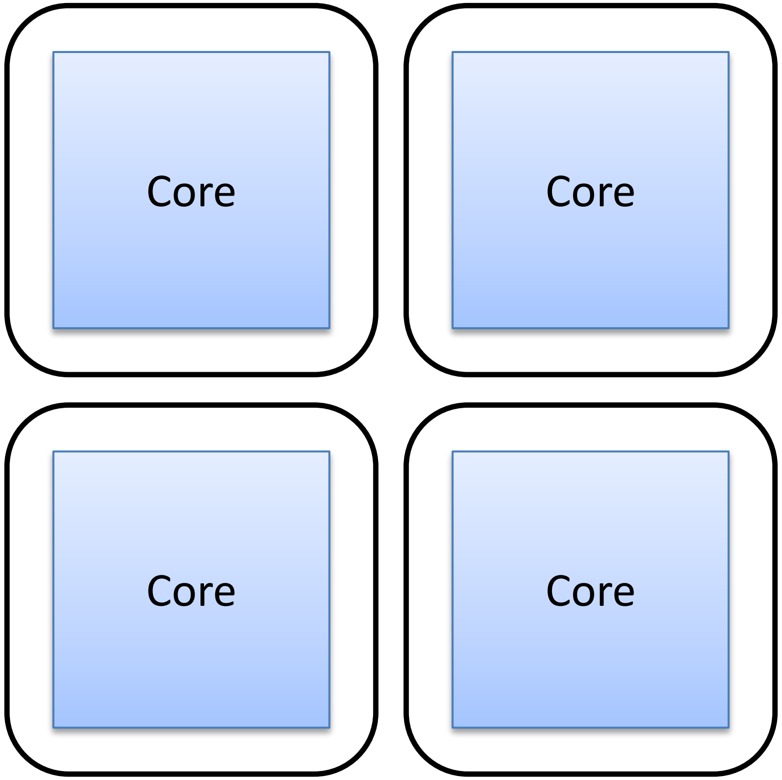
Data-Intensive Application and Systems Lab, EPFL

*IBM Research - Almaden

Hardware topologies have changed

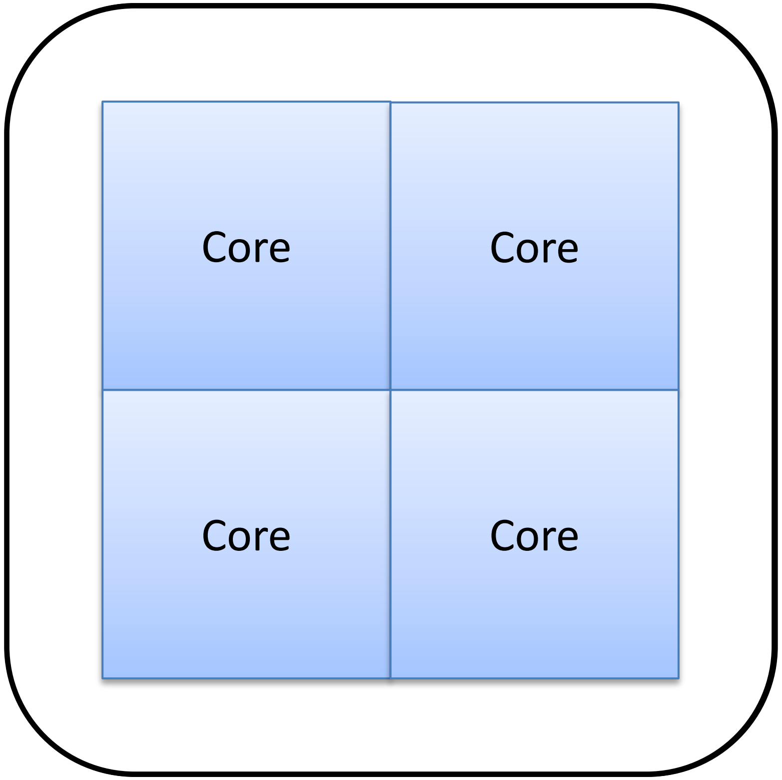


Hardware topologies have changed



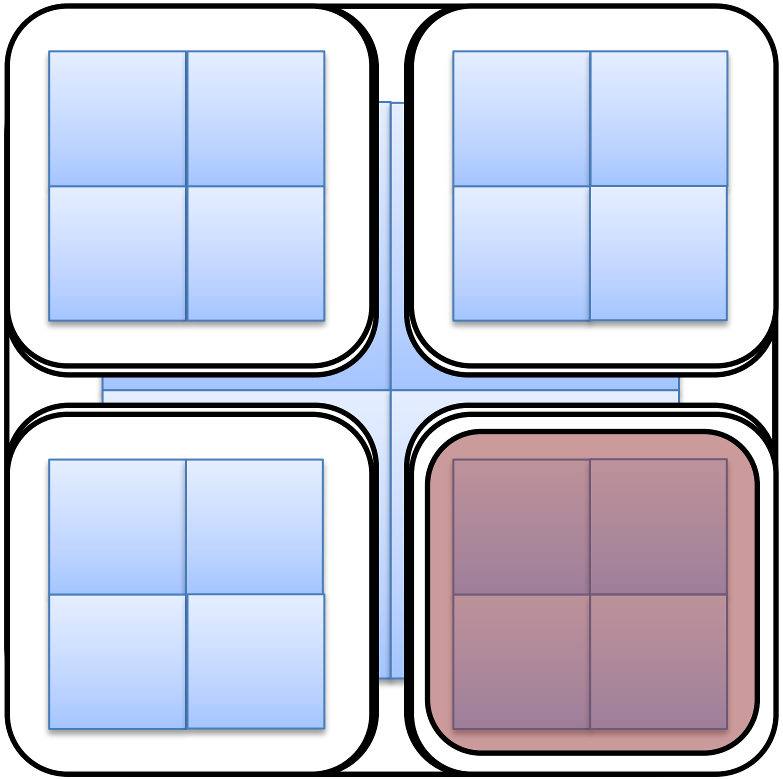
Topology	Core to core latency
SMP	high (~100ns)

Hardware topologies have changed



Topology	Core to core latency
SMP	high (~100ns)
CMP	low (~10ns)

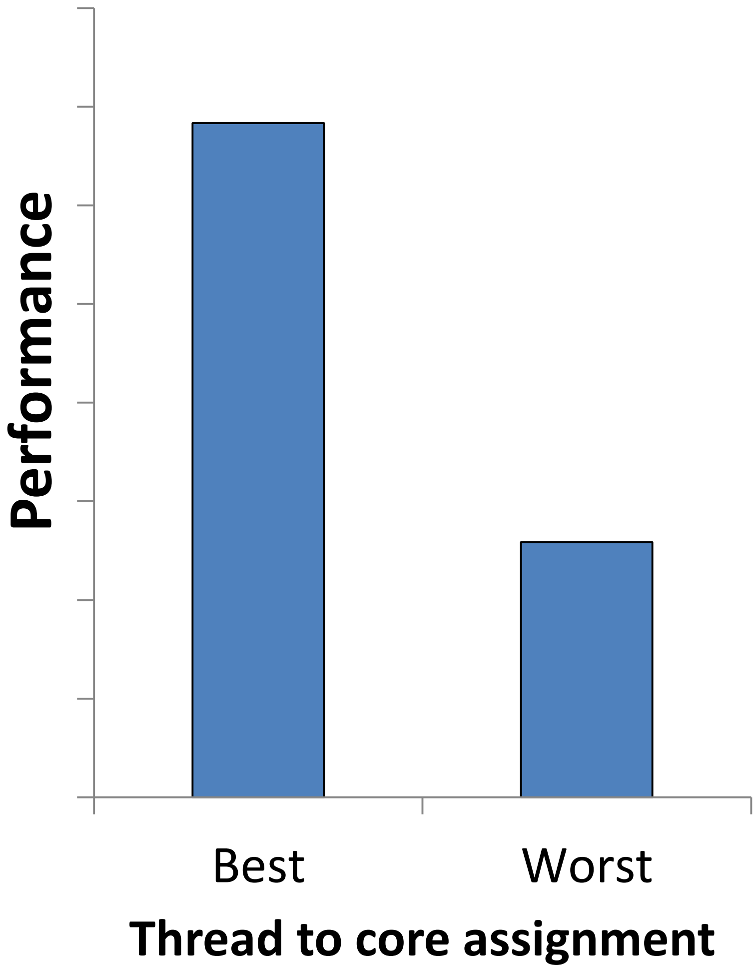
Hardware topologies have changed

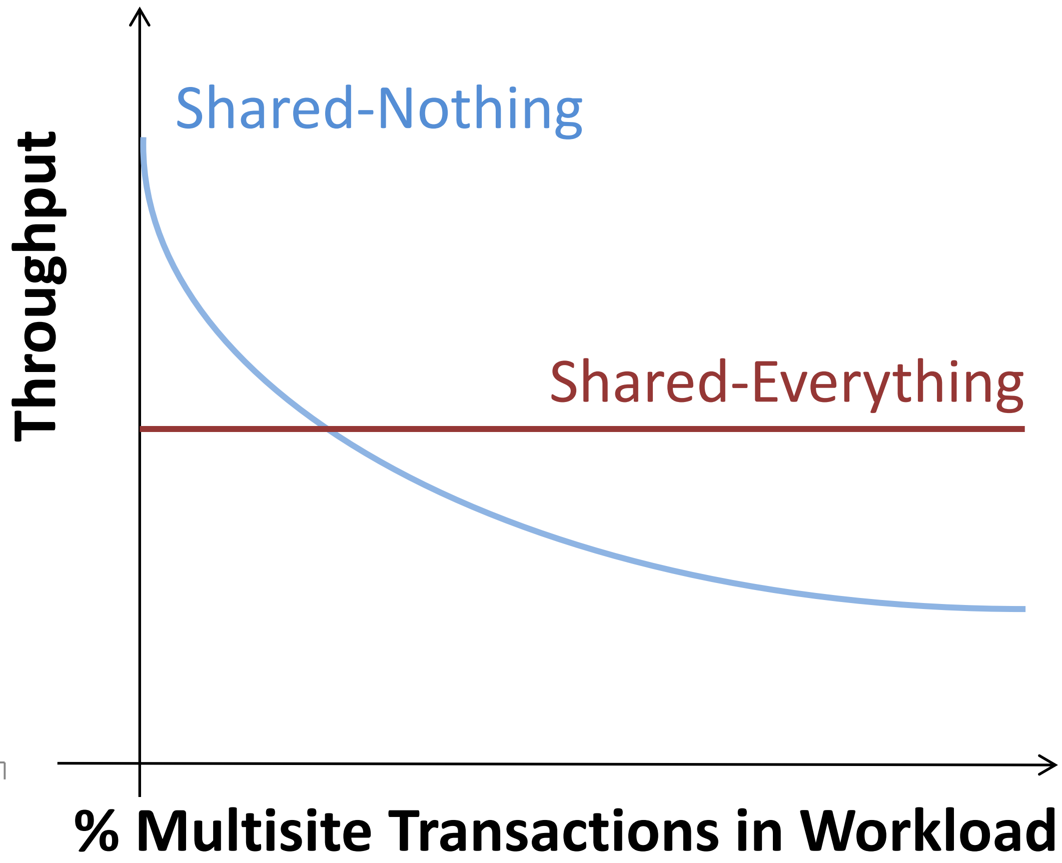


Topology	Core to core latency
SMP	high (~100ns)
CMP	low (~10ns)
SMP of CMP	*variable* (10-100ns)

Variable latencies affect performance & predictability

Deploying OLTP on Hardware Islands





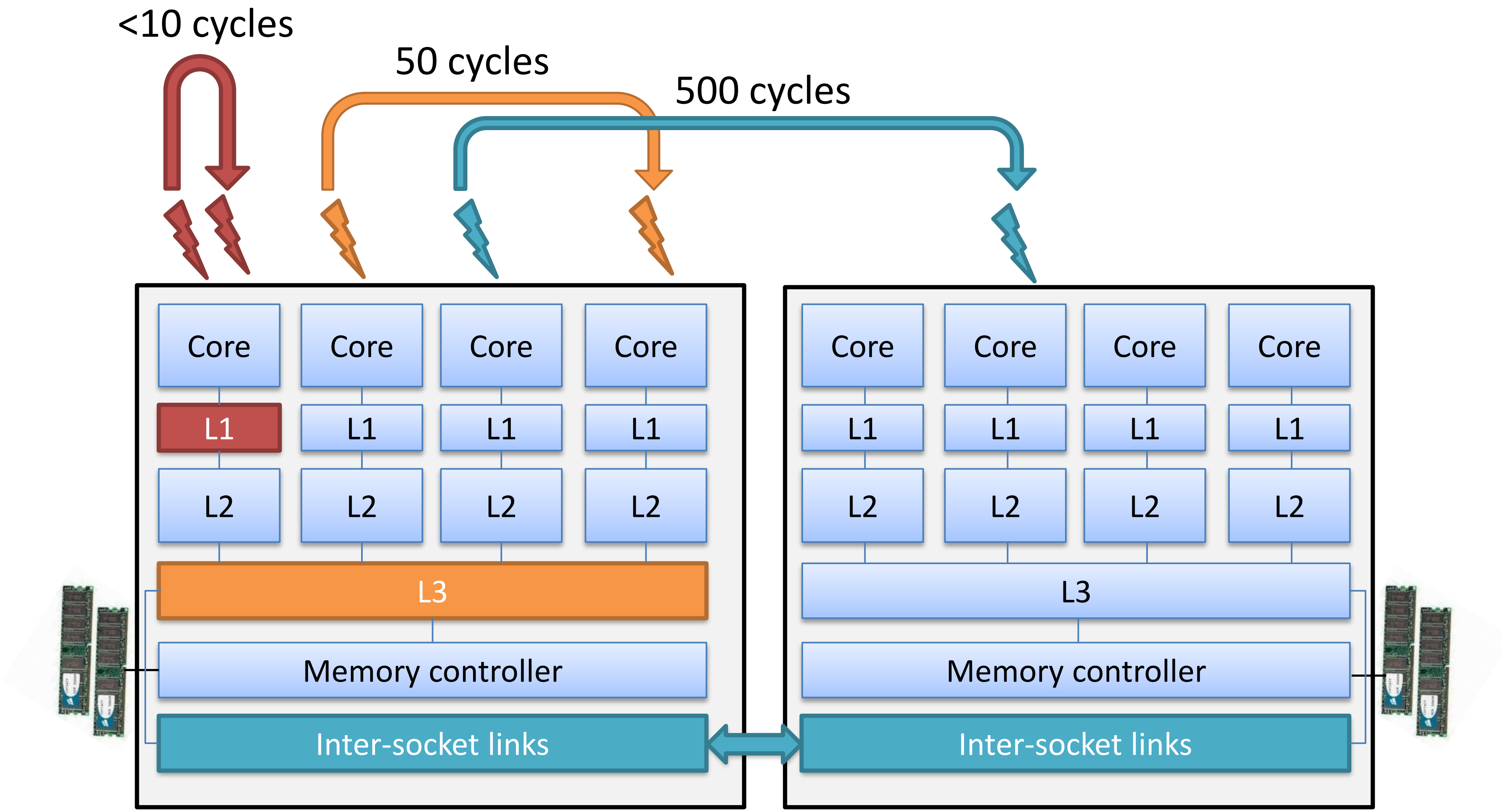
HW + Workload -> Optimal OLTP configuration

Outline

- Introduction
- **Hardware Islands**
- OLTP on Hardware Islands
- Conclusions and future work

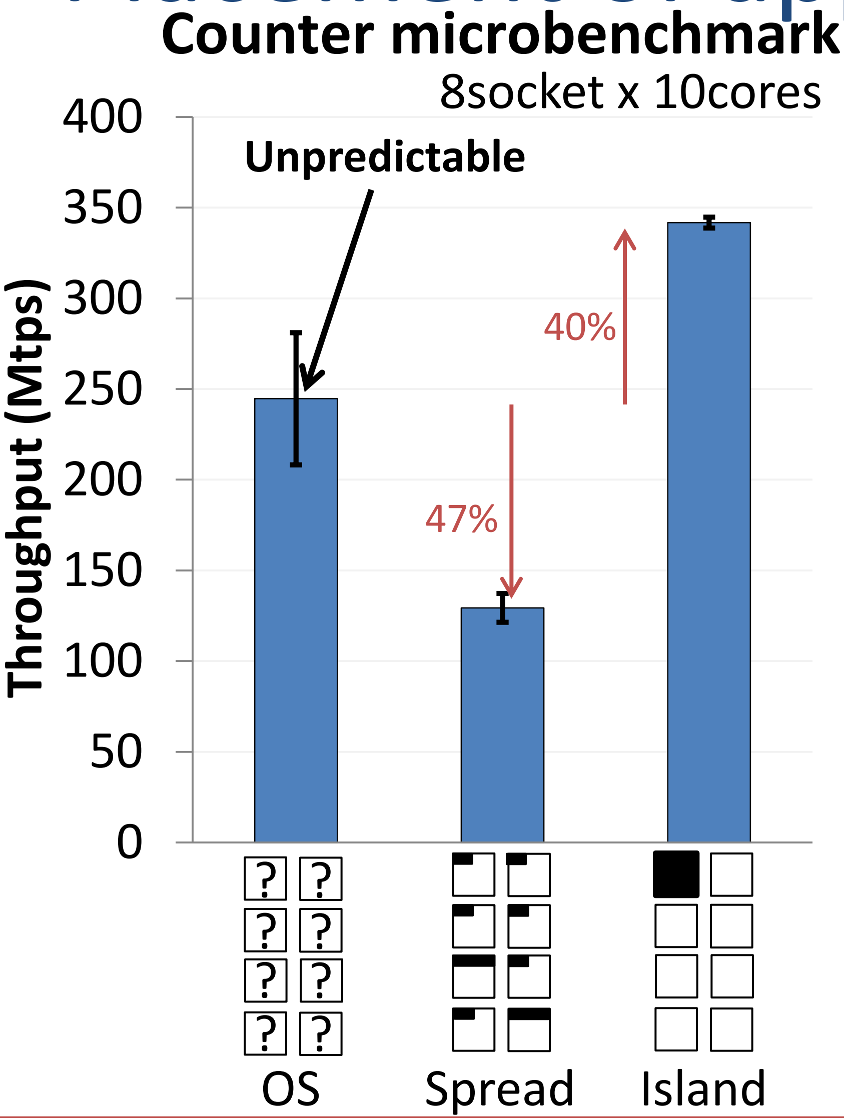
Data-Intensive Applications and Systems

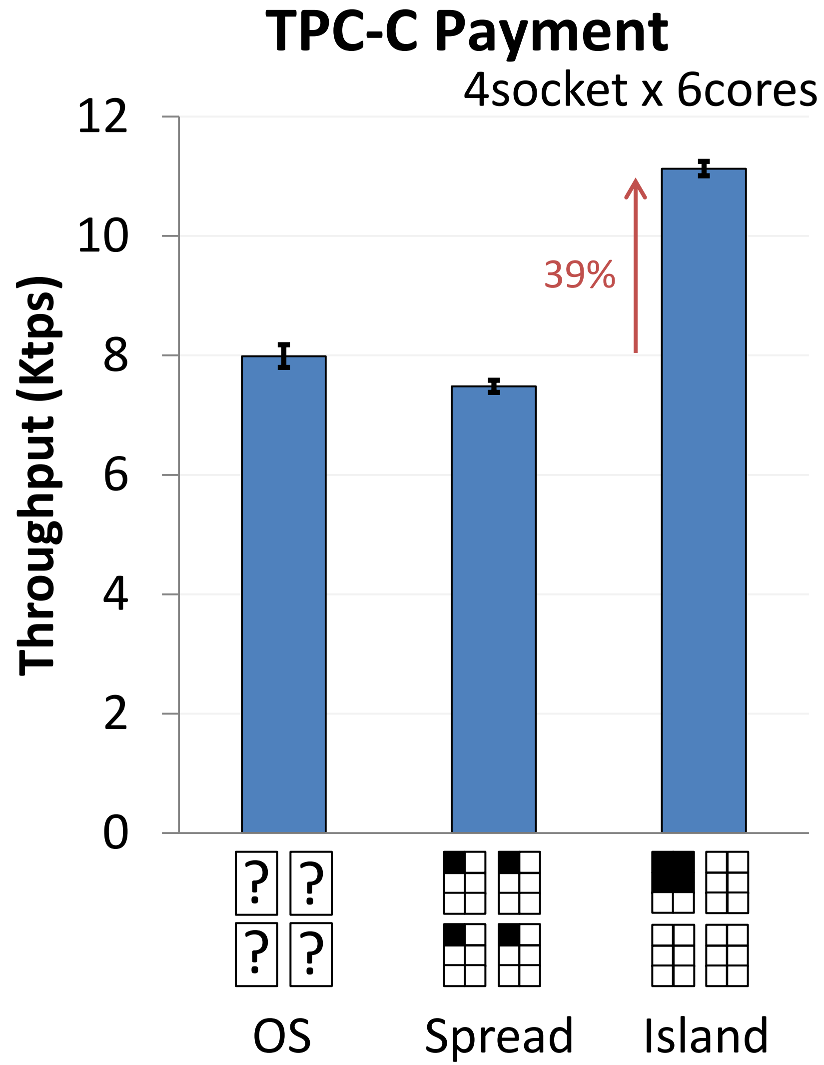
Multisocket multicores



Communication latency varies significantly

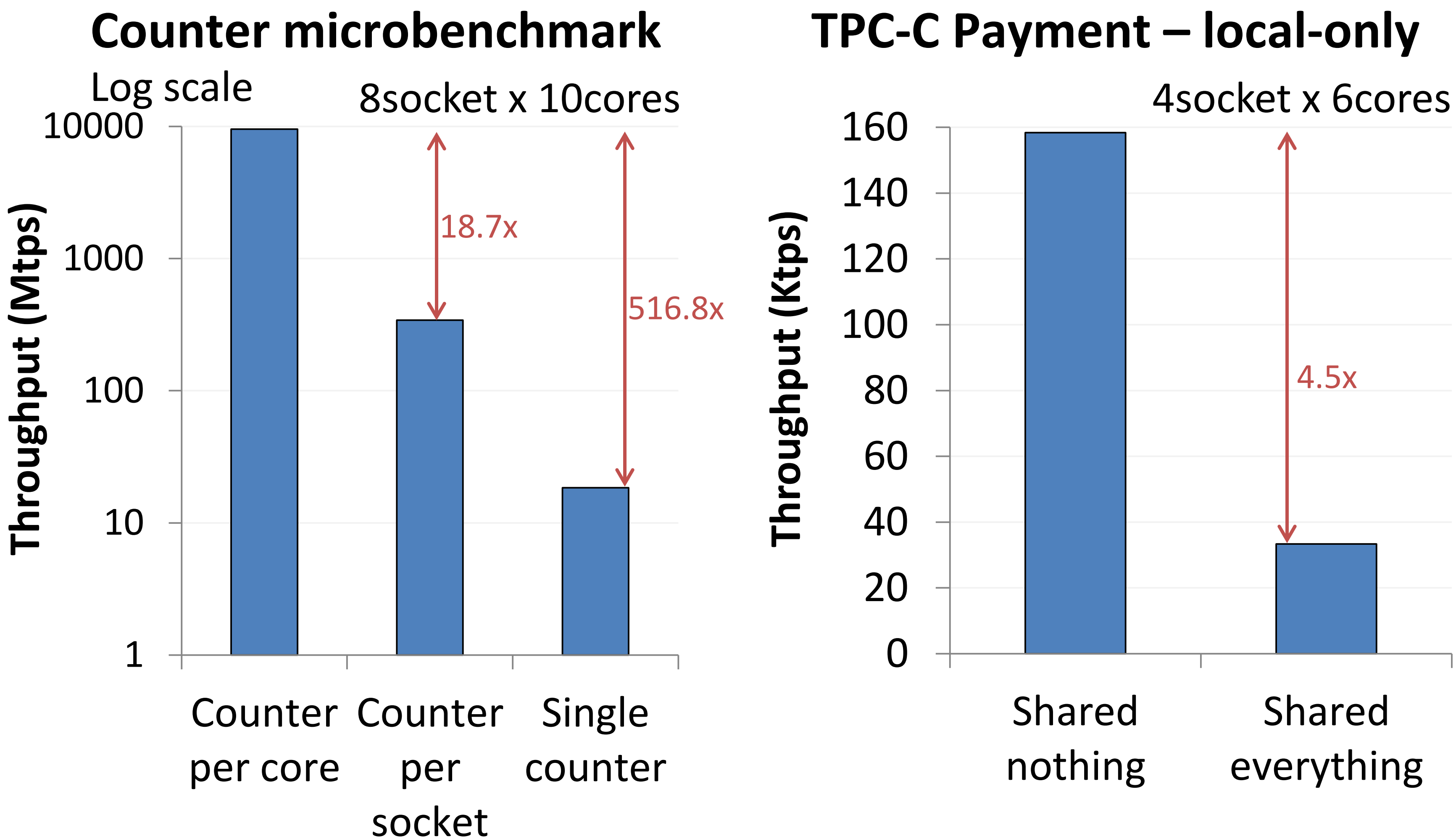
Placement of application threads





Islands-aware placement matters

Impact of sharing data among threads



Fewer sharers lead to higher performance

Outline

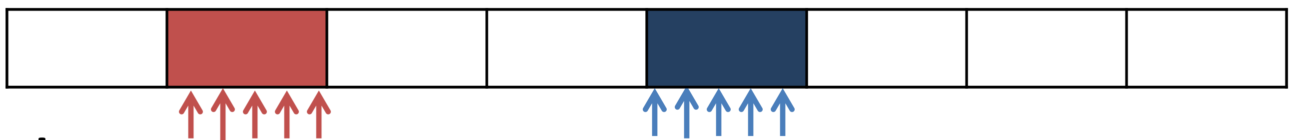
- Introduction
- Hardware Islands
- **OLTP on Hardware Islands**
  - Experimental setup
  - Read-only workloads
  - Update workloads
  - Impact of skew
- Conclusions and future work

Experimental setup

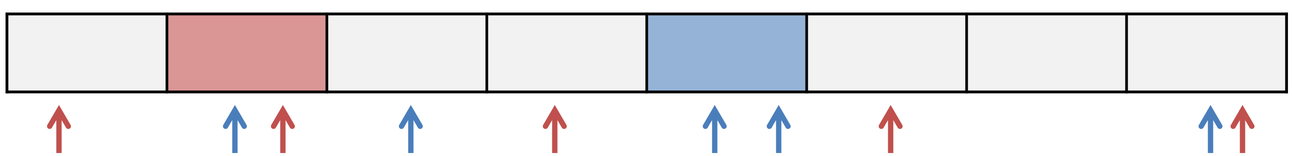
- Shore-MT
  - Top-of-the-line open source storage manager
  - Enabled shared-nothing capability
- Multisocket servers
  - 4-socket, 6-core Intel Xeon E7530, 64GB RAM
  - 8-socket, 10-core Intel Xeon E7-L8867, 192GB RAM
- Disabled hyper-threading
- OS: Red Hat Enterprise Linux 6.2, kernel 2.6.32
- Profiler: Intel VTune Amplifier XE 2011
- Workloads: TPC-C, microbenchmarks

Microbenchmark workload

- Singlesite version
  - Probe/update N rows from the local partition

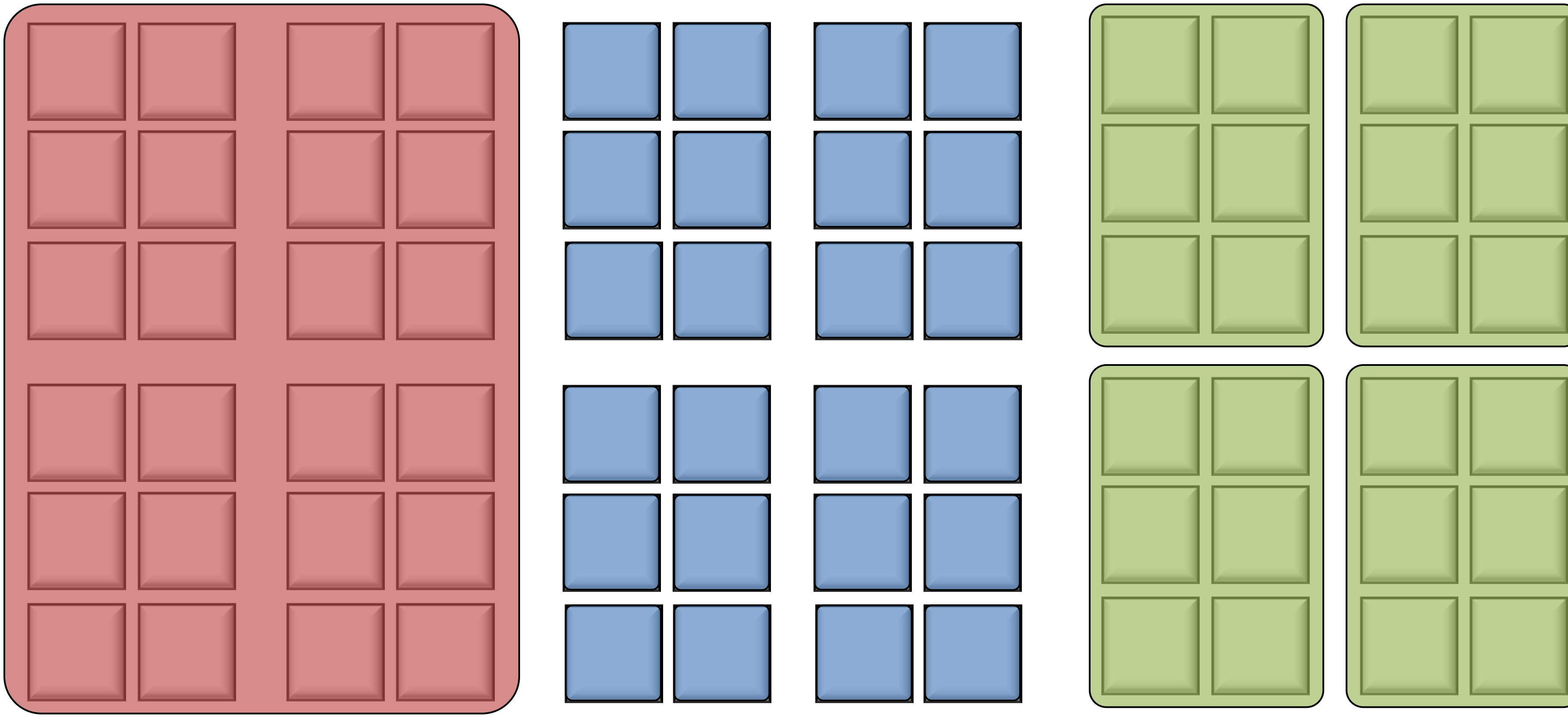


- Multisite version
  - Probe/update 1 row from the local partition
  - Probe/update N-1 rows uniformly from any partition
  - Partitions may reside on the same instance



- Input size: 10 000 rows/core

Software System Configurations

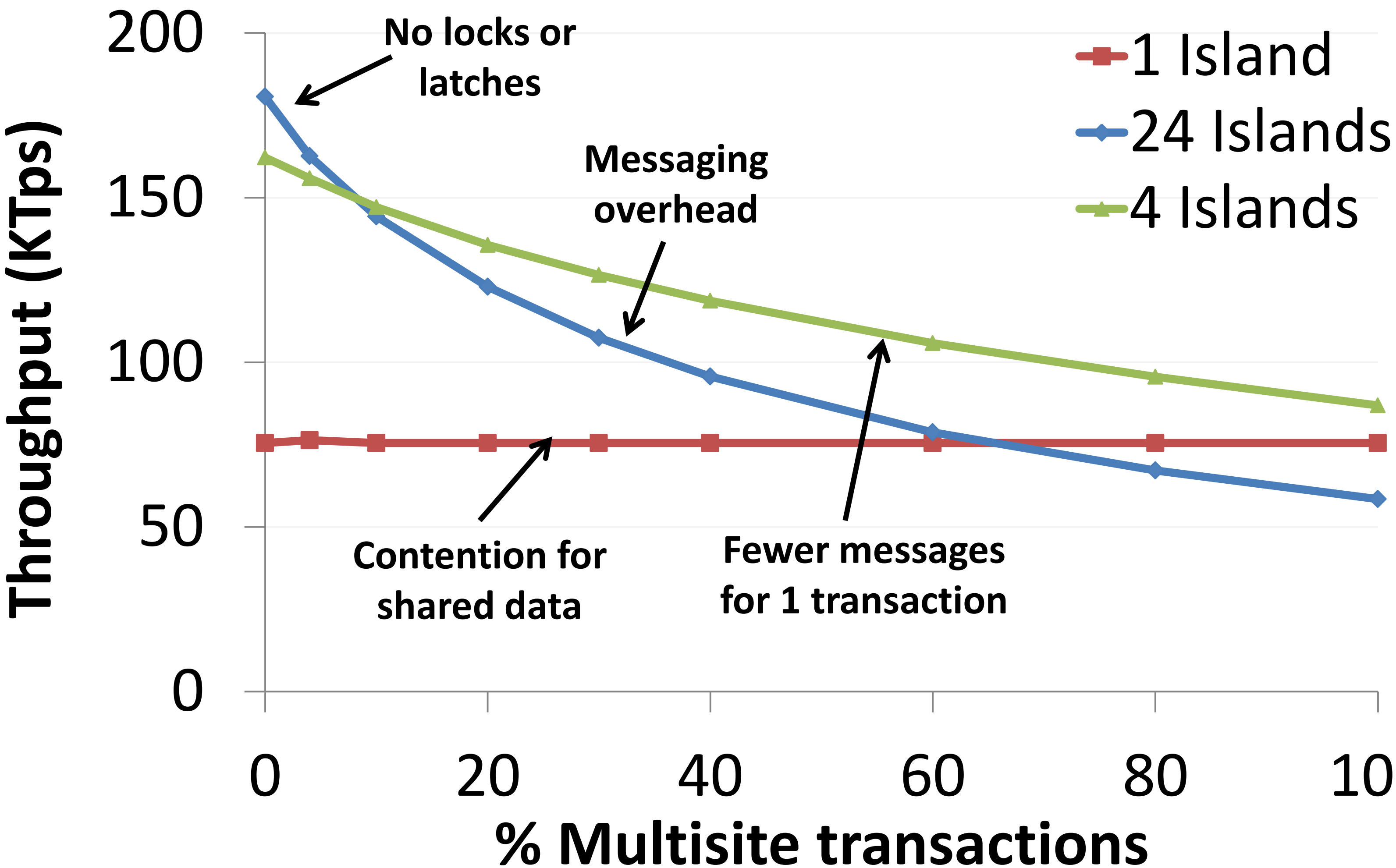


1 Island
Shared-everything

24 Islands
Shared-nothing

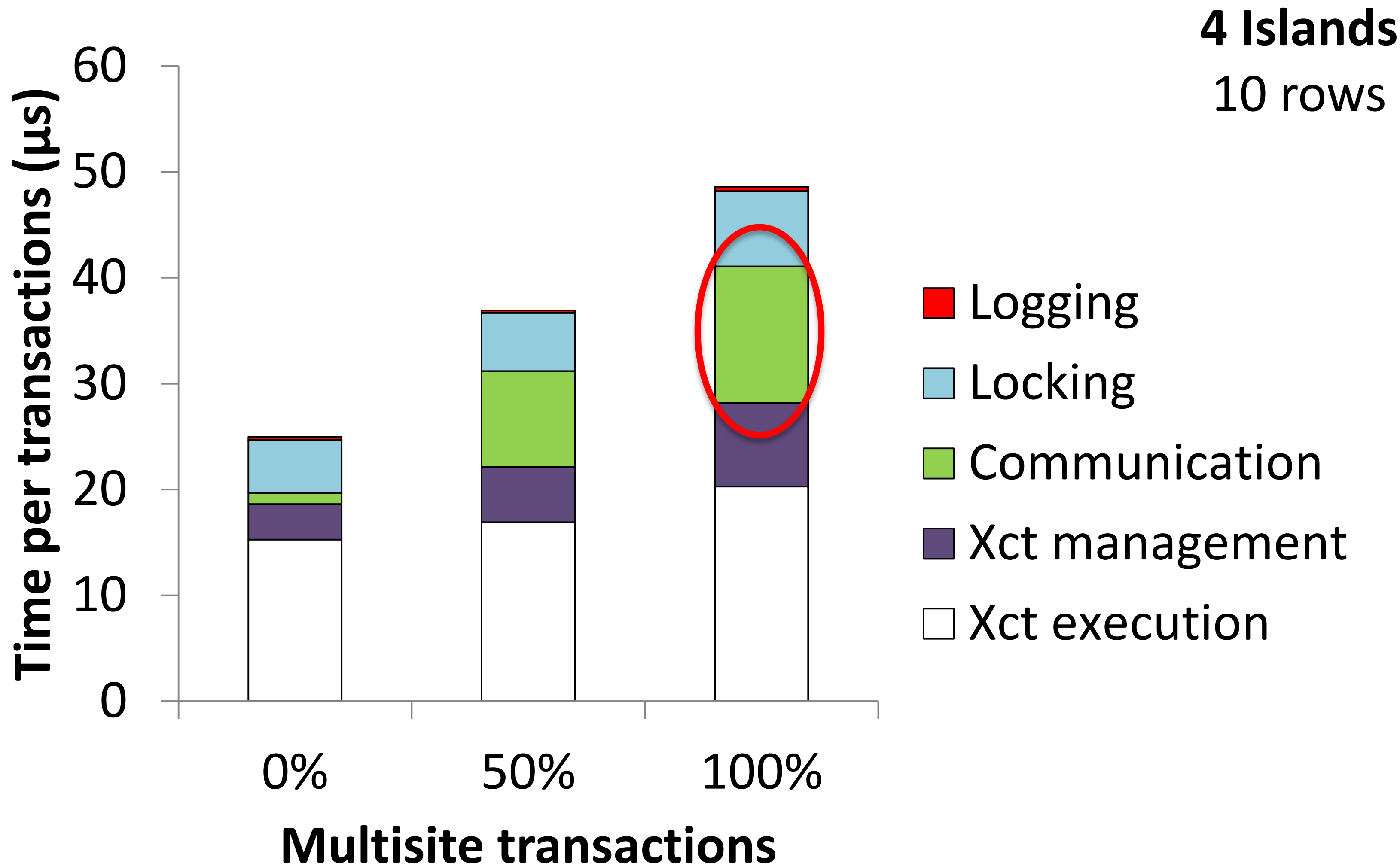
4 Islands

Increasing % of multisite xcts: reads



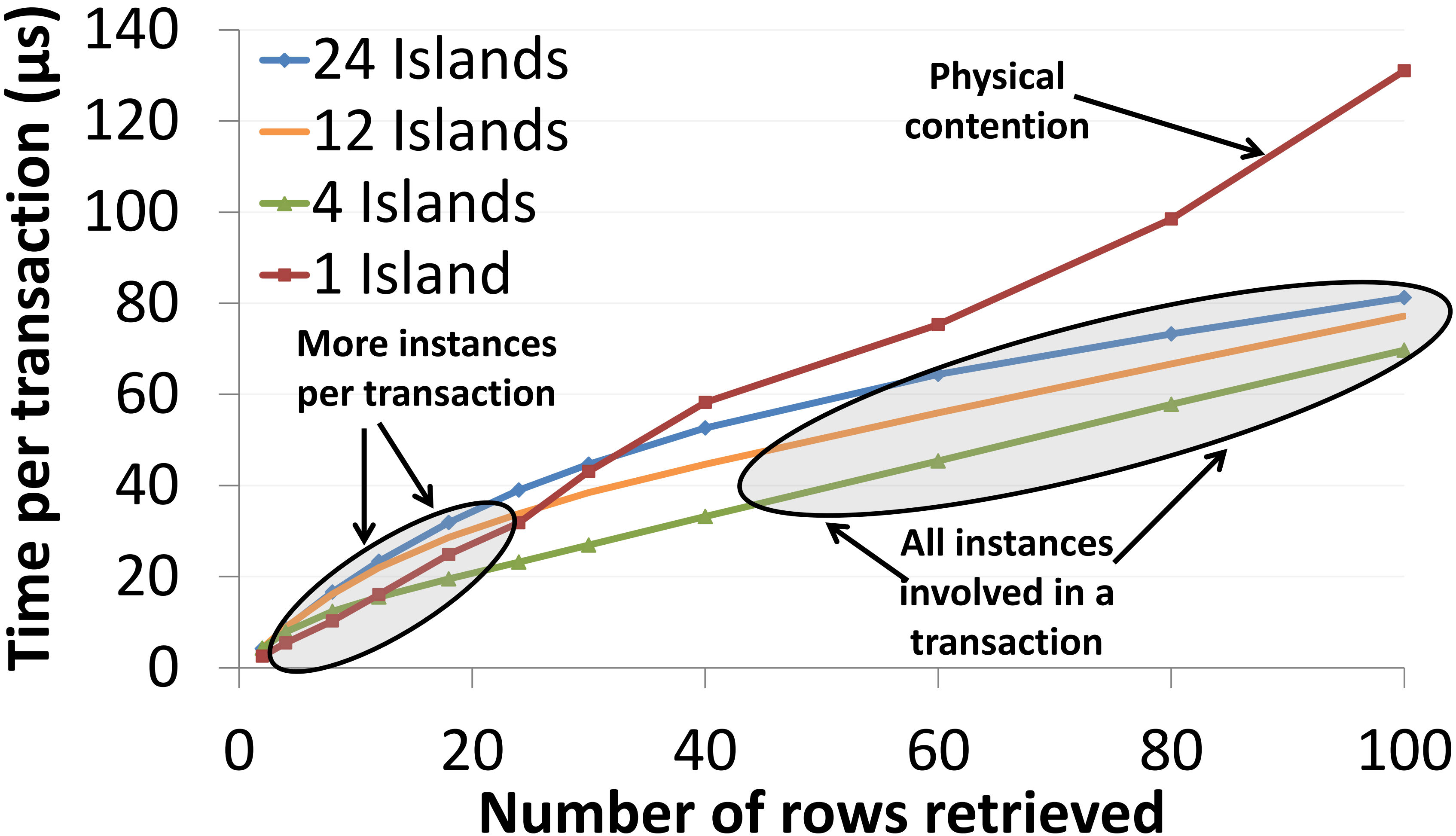
Finer grained configurations are more sensitive to distributed transactions

Where are the bottlenecks? Read case



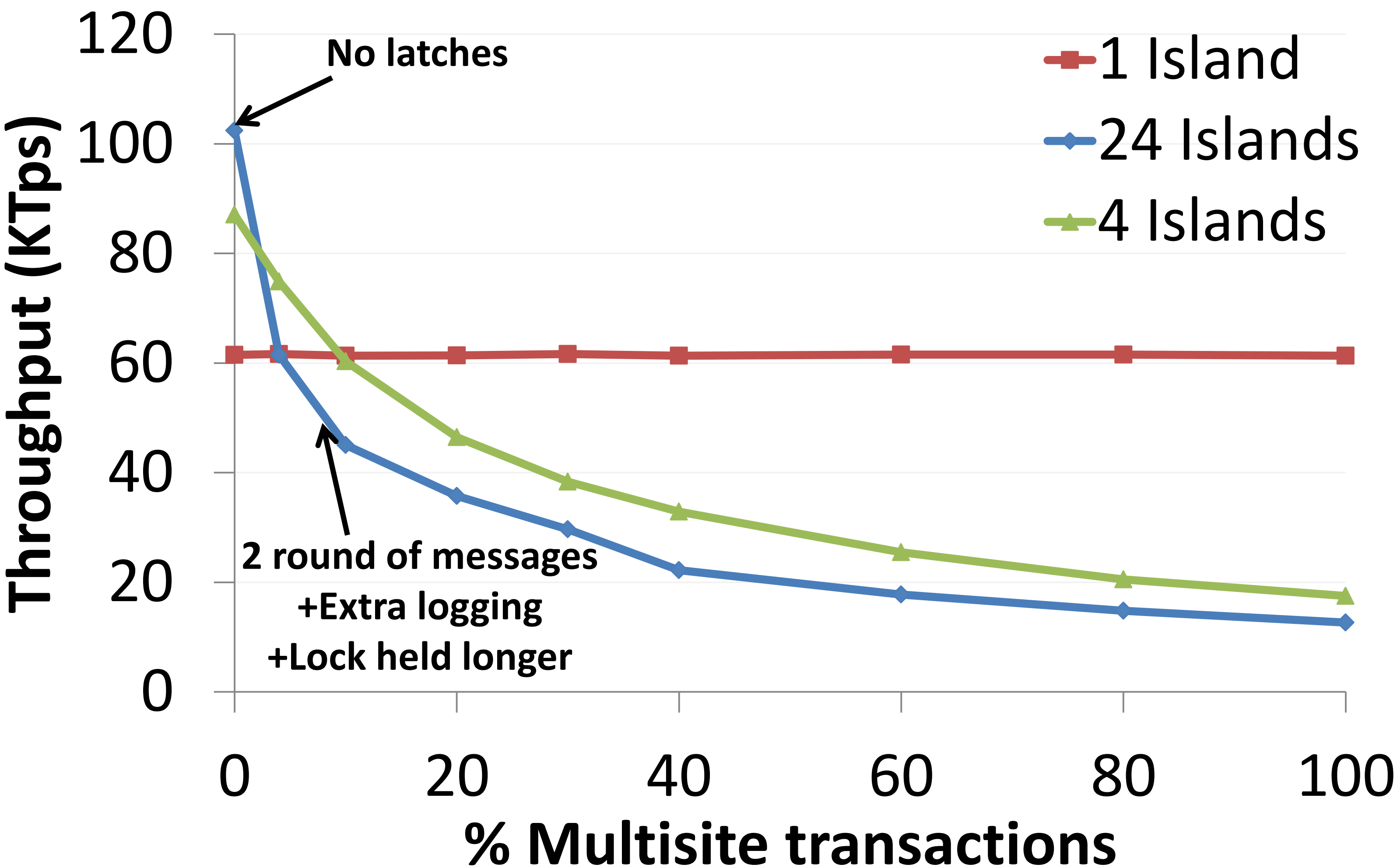
Communication overhead dominates

Increasing size of multisite xct: read case



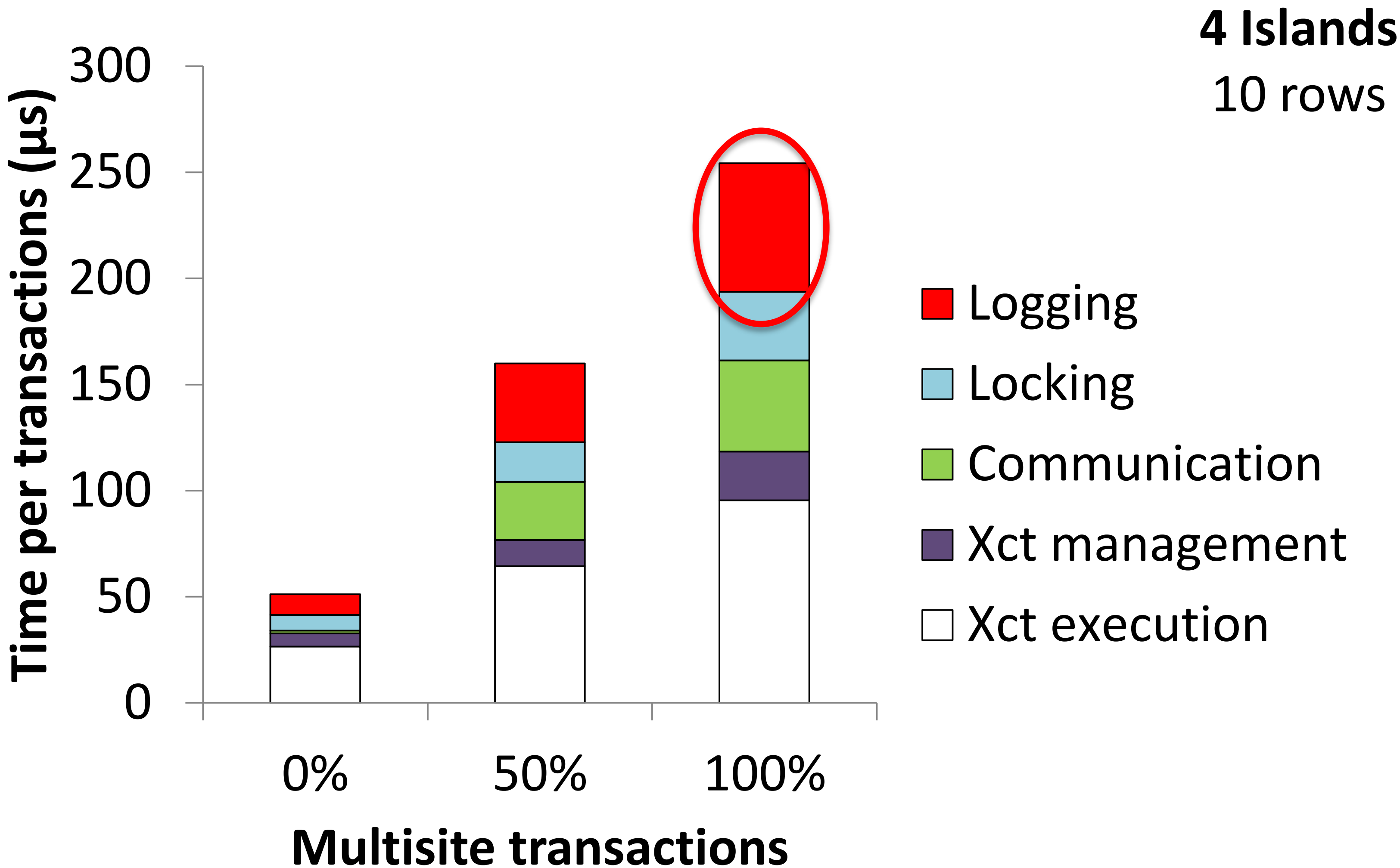
Communication costs rise until all instances are involved in every transaction

Increasing % of multisite xcts: updates



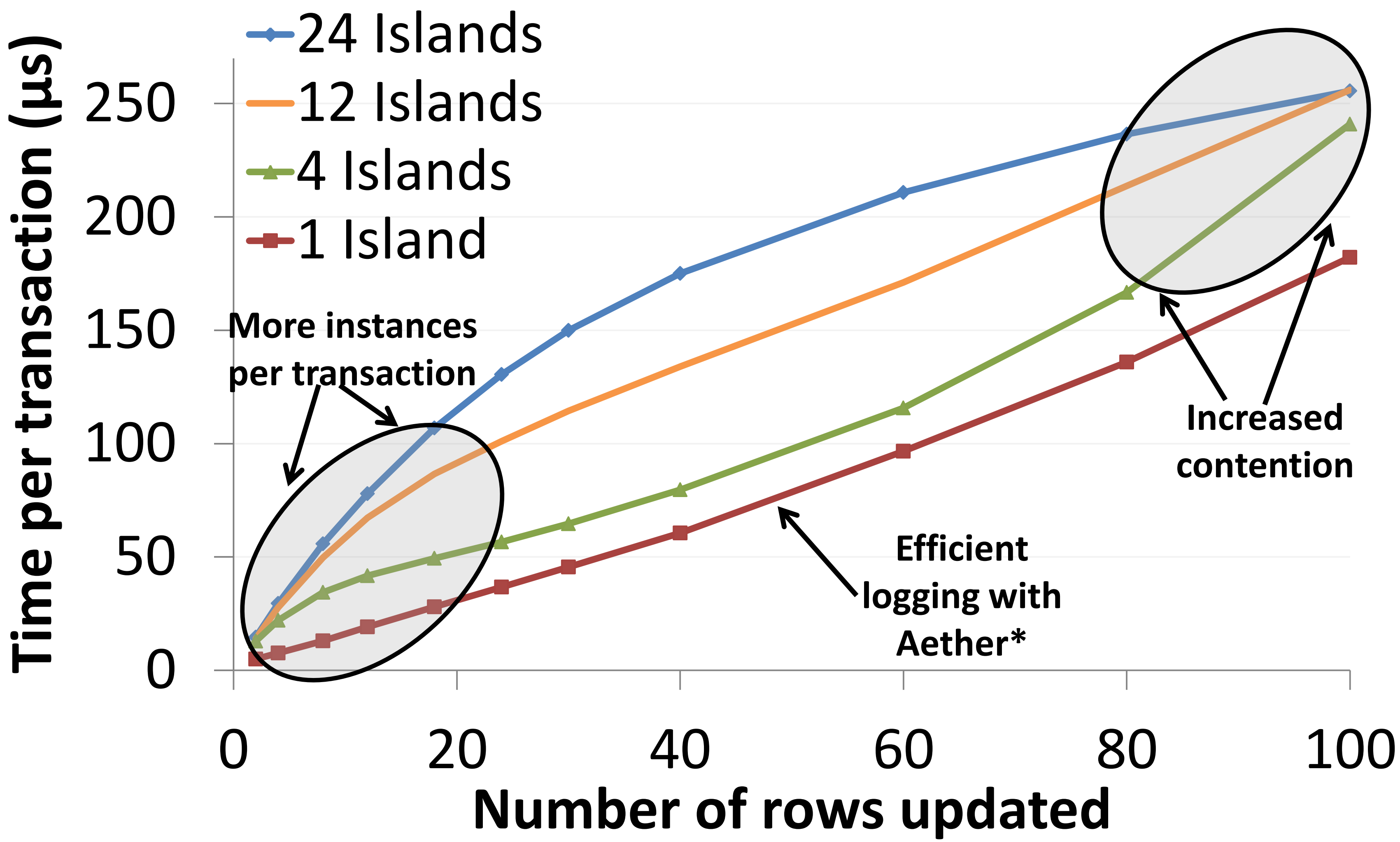
Distributed update transactions are more expensive

Where are the bottlenecks? Update case



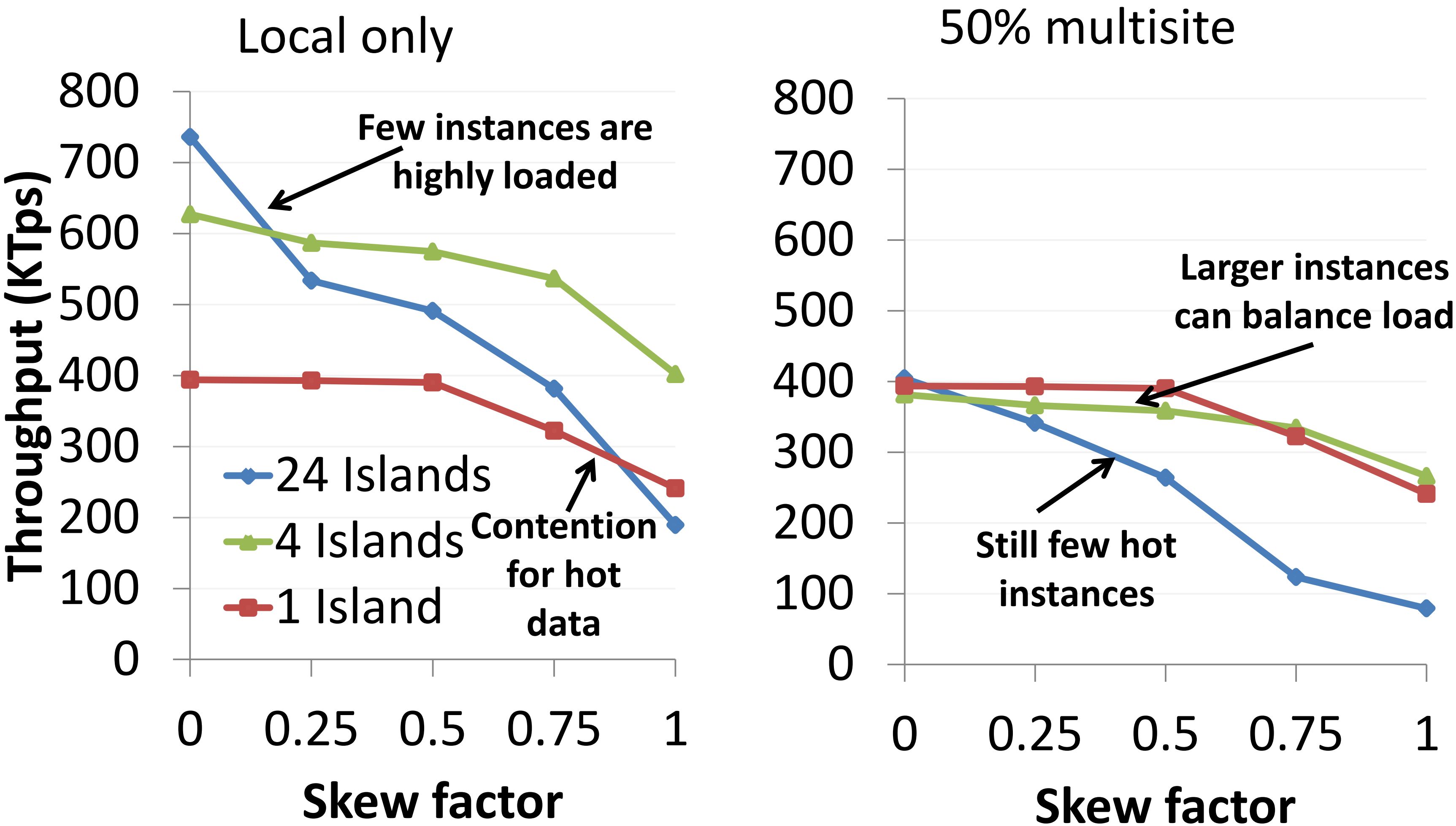
Communication overhead dominates

Increasing size of multisite xct: update case



Shared everything exposes constructive interference

Effects of skewed input



4 Islands effectively balance skew and contention

OLTP systems on Hardware Islands

- Shared-everything: stable, but non-optimal
- Shared-nothing: fast, but sensitive to workload
- OLTP Islands: a robust, middle-ground
  - Runs on close cores
  - Small instances limits contention between threads
  - Few instances simplify partitioning
- Future work:
  - Automatically choose and setup optimal configuration
  - Dynamically adjust to workload changes

Thank you!