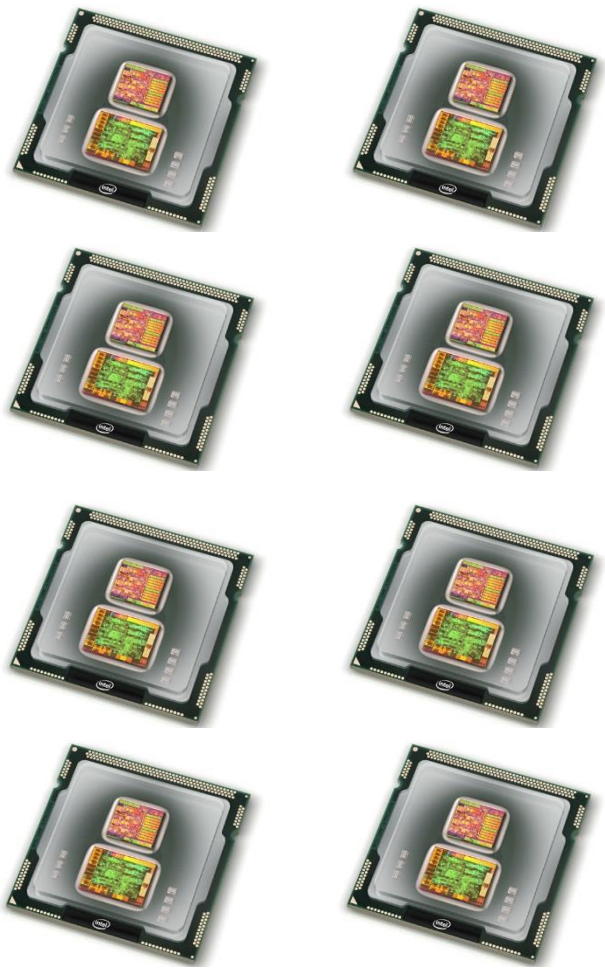
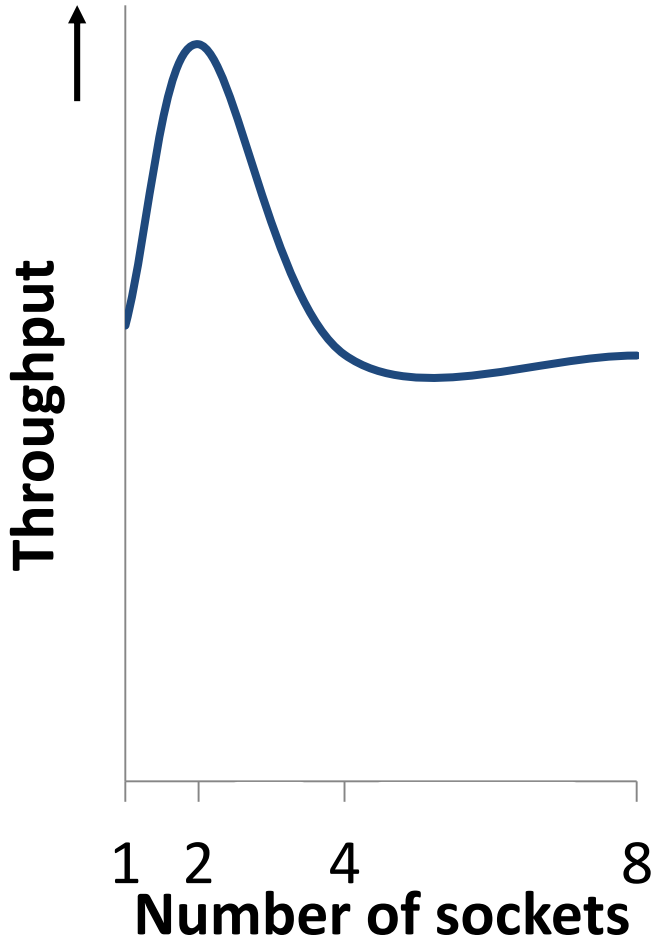
ATraPos: Adaptive Transaction Processing on Hardware Islands

Danica Porobic, Erietta Liarou, Pınar Tözün, Anastasia Ailamaki

Data-Intensive Application and Systems Lab, EPFL

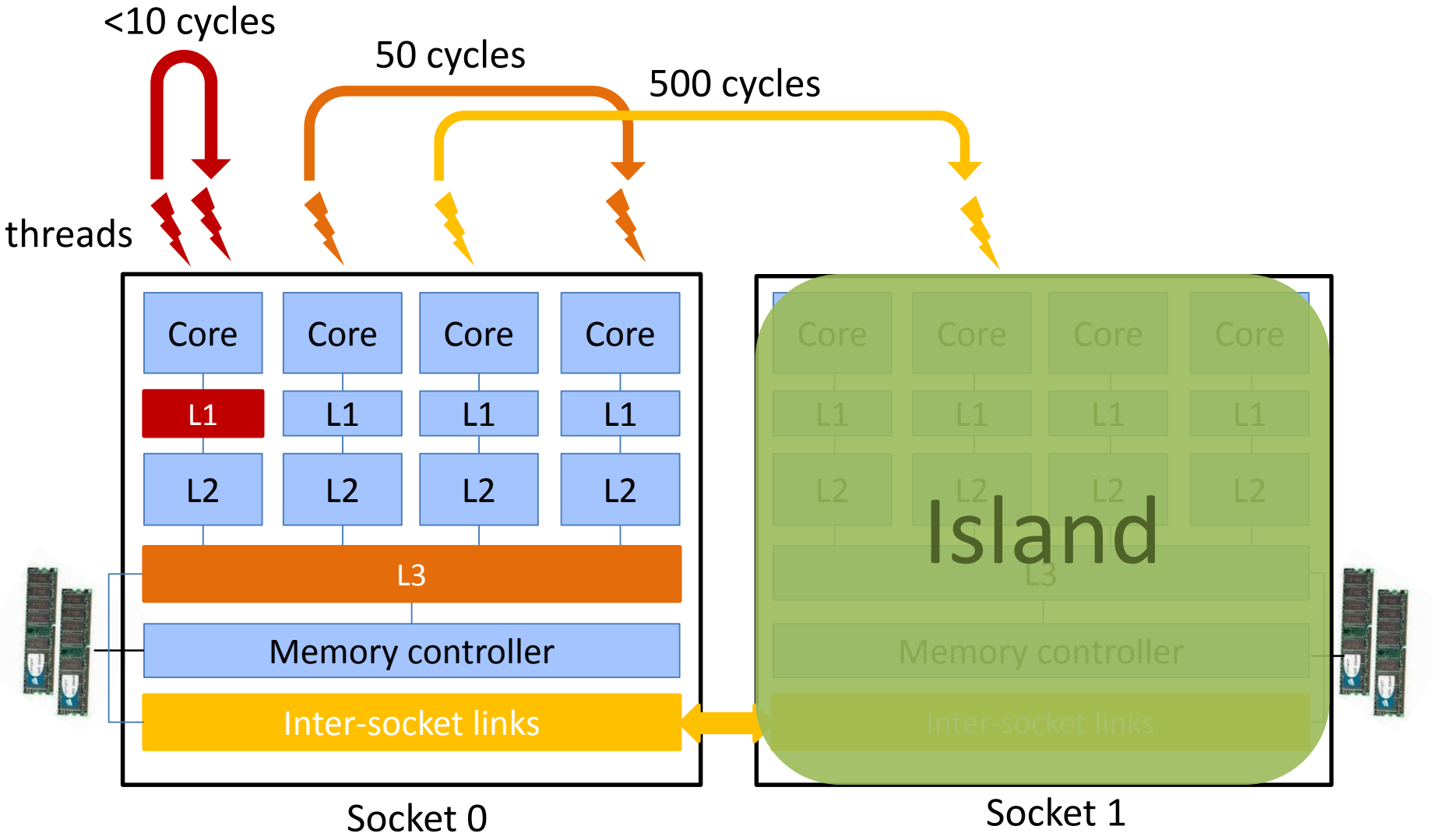
Scaling up OLTP on multisockets





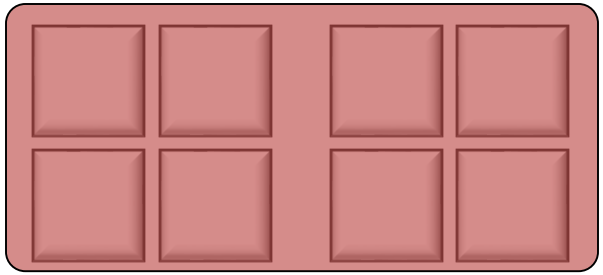
Multisocket servers are severely under-utilized

Multisocket multicores

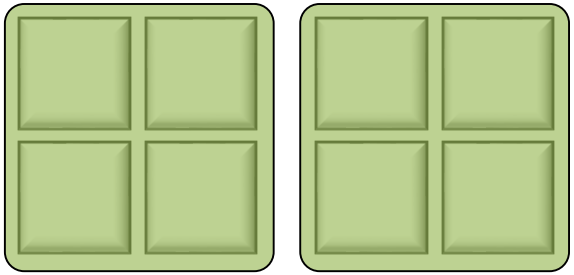


Communication latencies vary by an order-of-magnitude

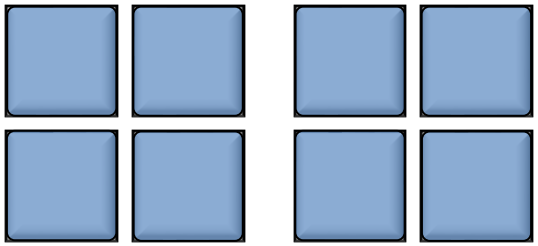
OLTP on Hardware Islands



Shared-everything

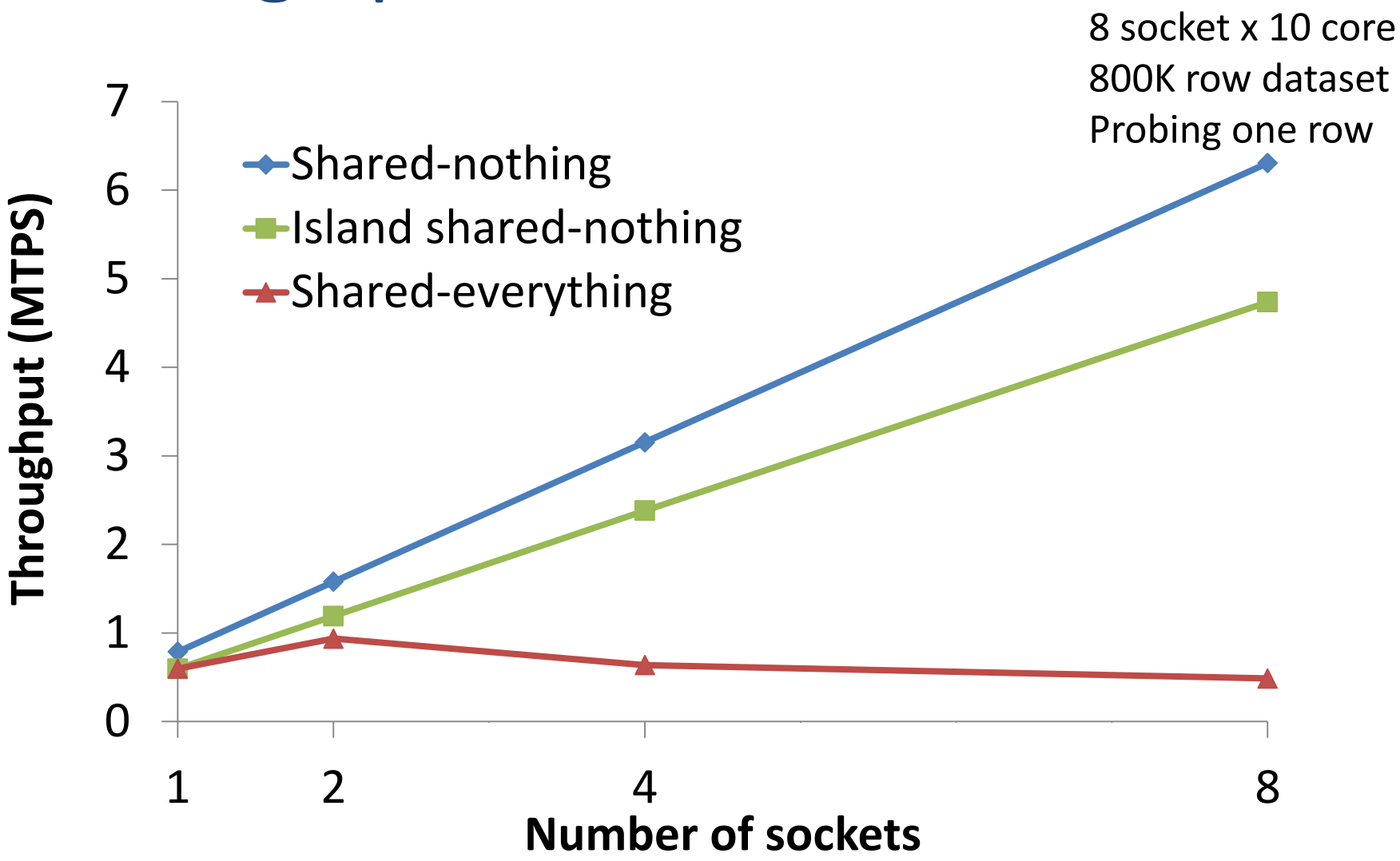


Island shared-nothing



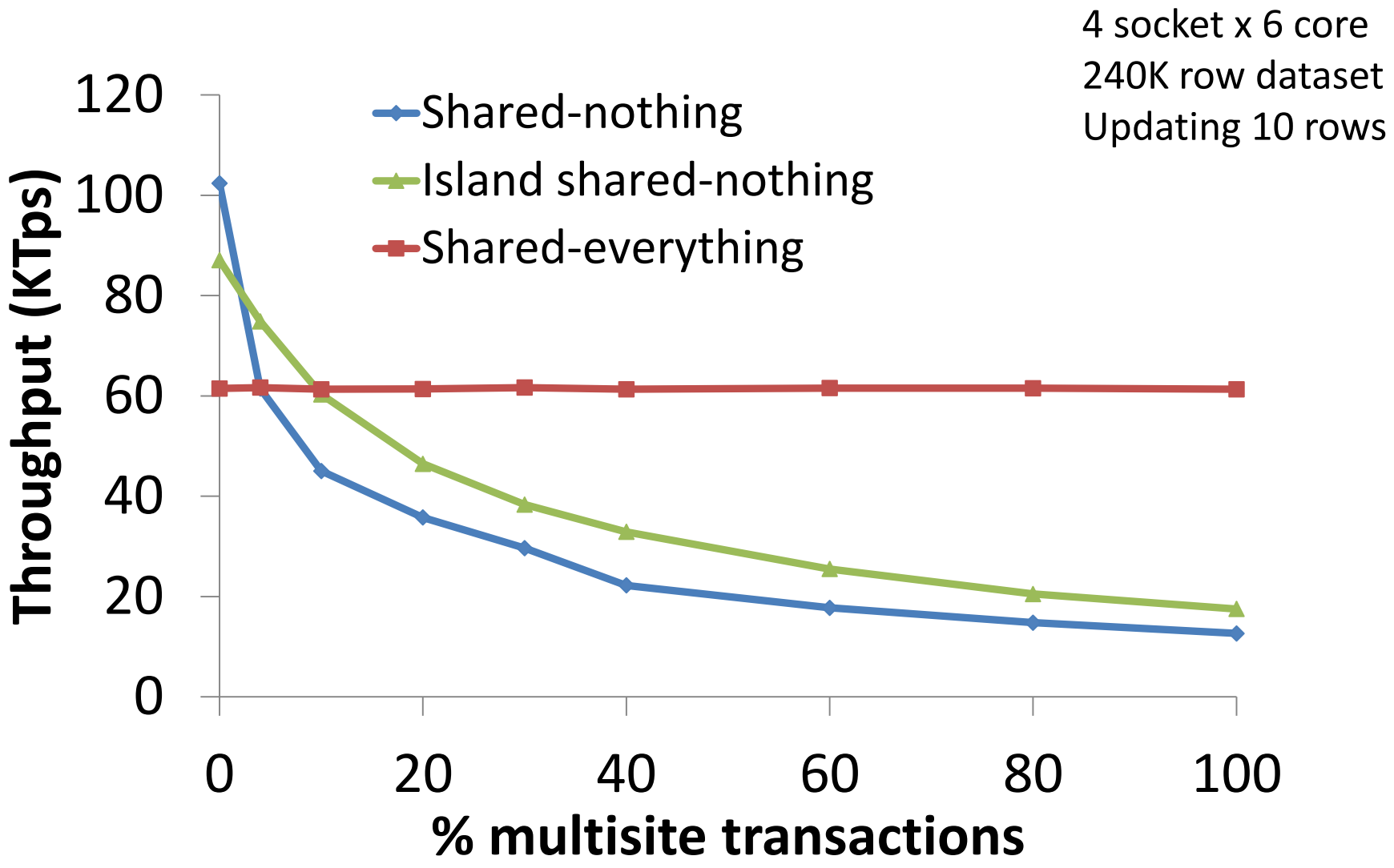
Shared-nothing

Scaling up on a 8-socket machine



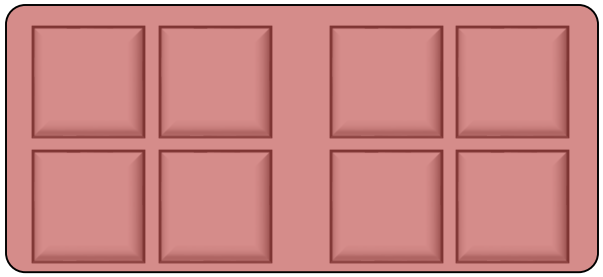
Islands significantly challenge scalability

Physical partitioning for Islands



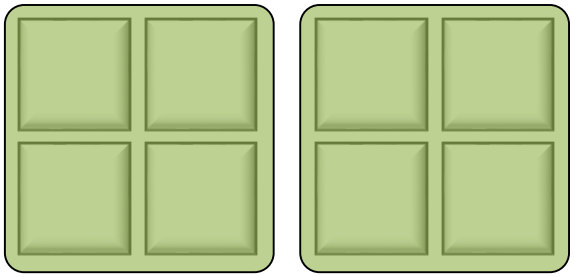
No configuration is optimal for all environments

OLTP on Hardware Islands



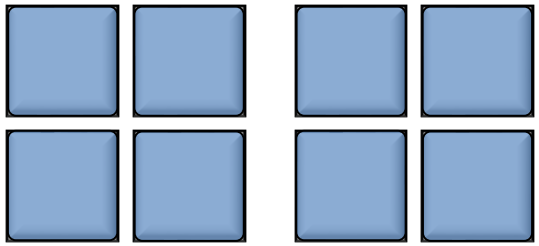
Shared-everything

- ✓ Stable
- ✗ Not optimal



Island shared-nothing

- ✓ Robust middle ground



Shared-nothing

- ✓ Fast
- ✗ Sensitive to workload

- Challenges
  - Optimal configuration depends on workload and hardware
  - Expensive repartitioning due to physical data movement

ATraPos: hardware and workload-aware shared-everything adaptive system

ATraPos: Adaptive Transaction Processing

- No unnecessary inter-socket synchronization
- Workload & hardware-aware partitioning
- Lightweight monitoring and repartitioning

ATraPos: hardware and workload-aware shared-everything adaptive system

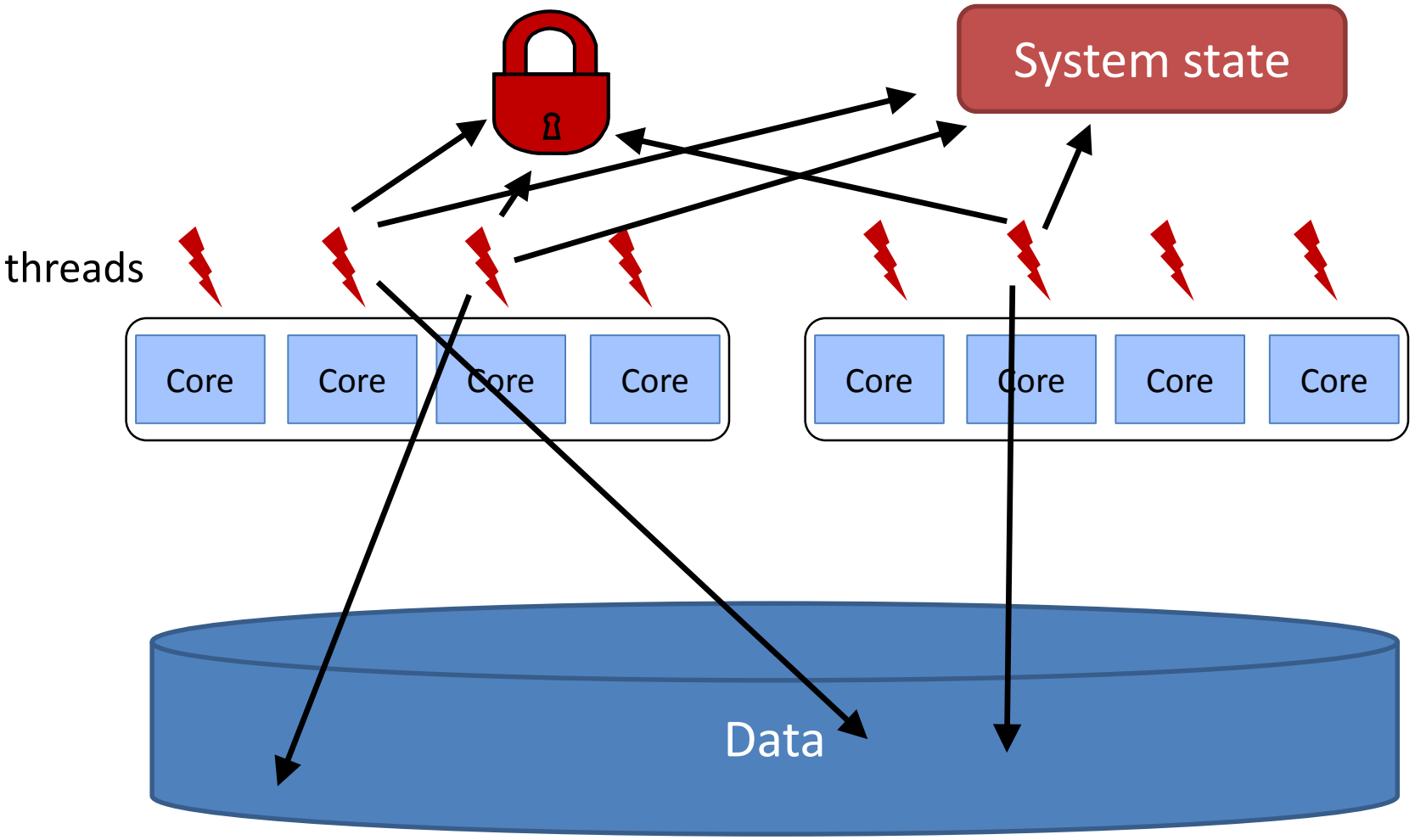
Outline

- Impact of Hardware Islands on OLTP

- **ATraPos**
  - **Avoiding unnecessary synchronization**
  - Workload & hardware-aware partitioning & placement
  - Lightweight monitoring & repartitioning

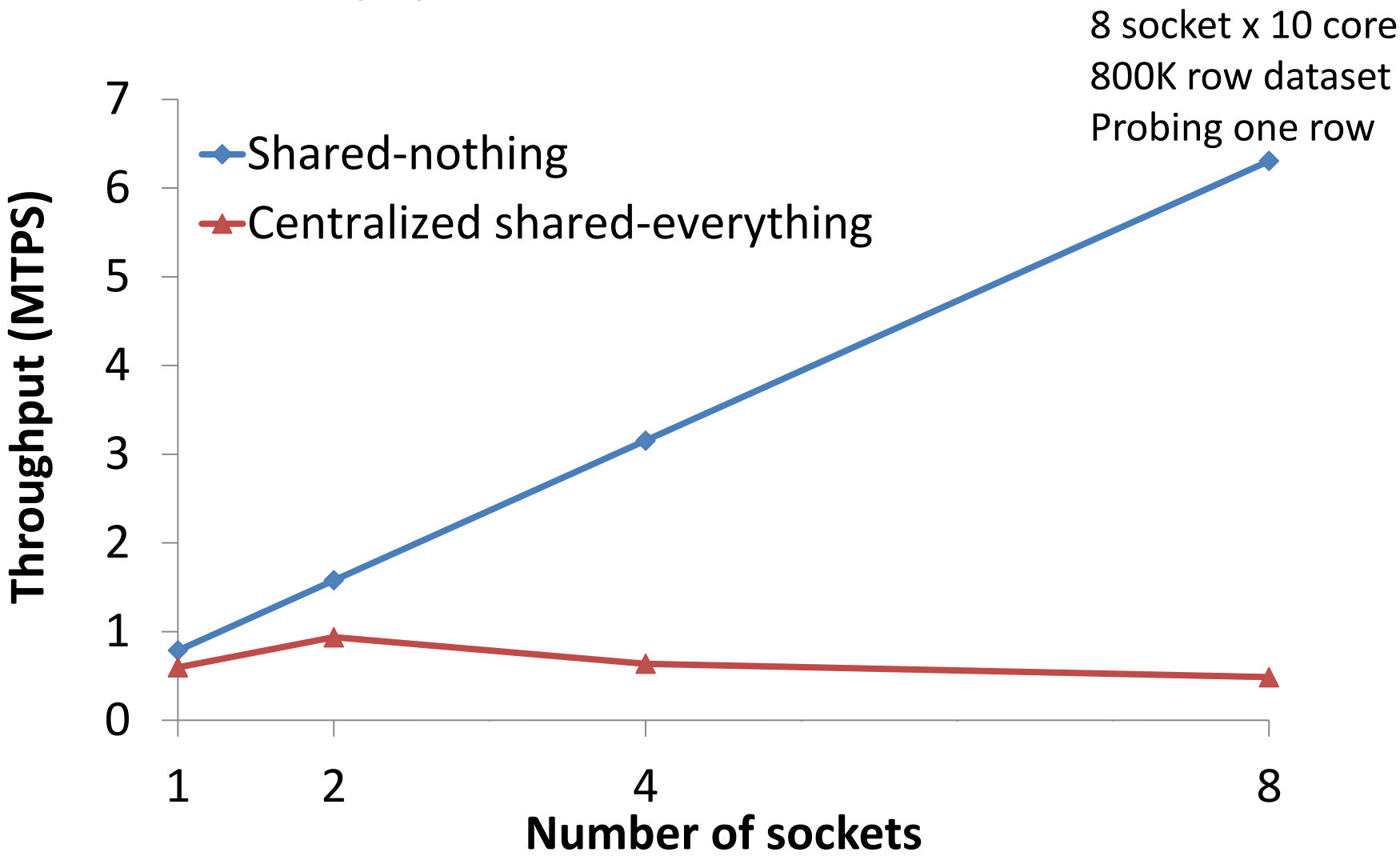
- Summary

Critical path of transaction execution



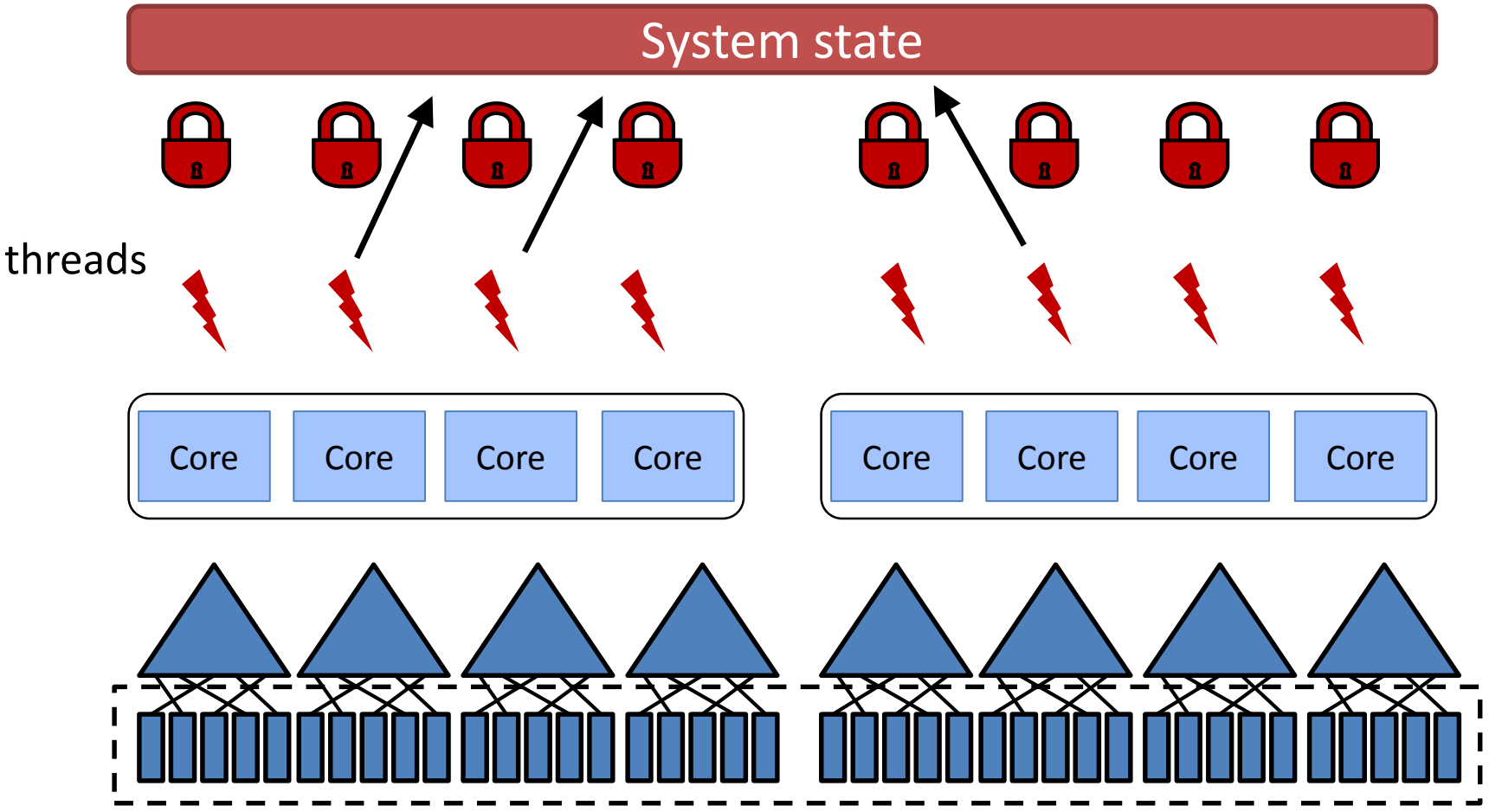
Many accesses to shared data structures

Perfectly partitionable workload



Accessing centralized data structures limits scalability

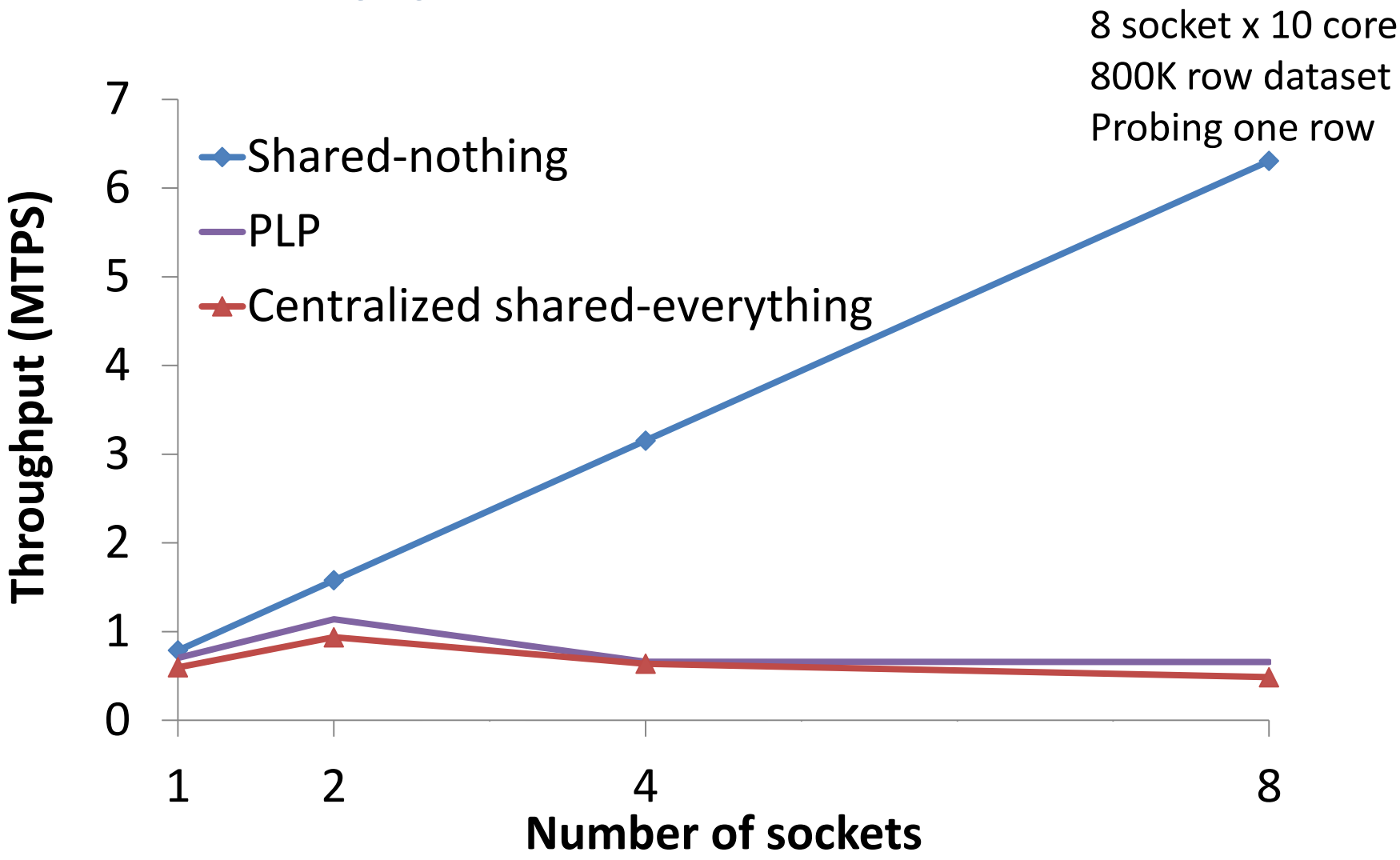
PLP: Physiologically partitioned SE*



System state is still shared

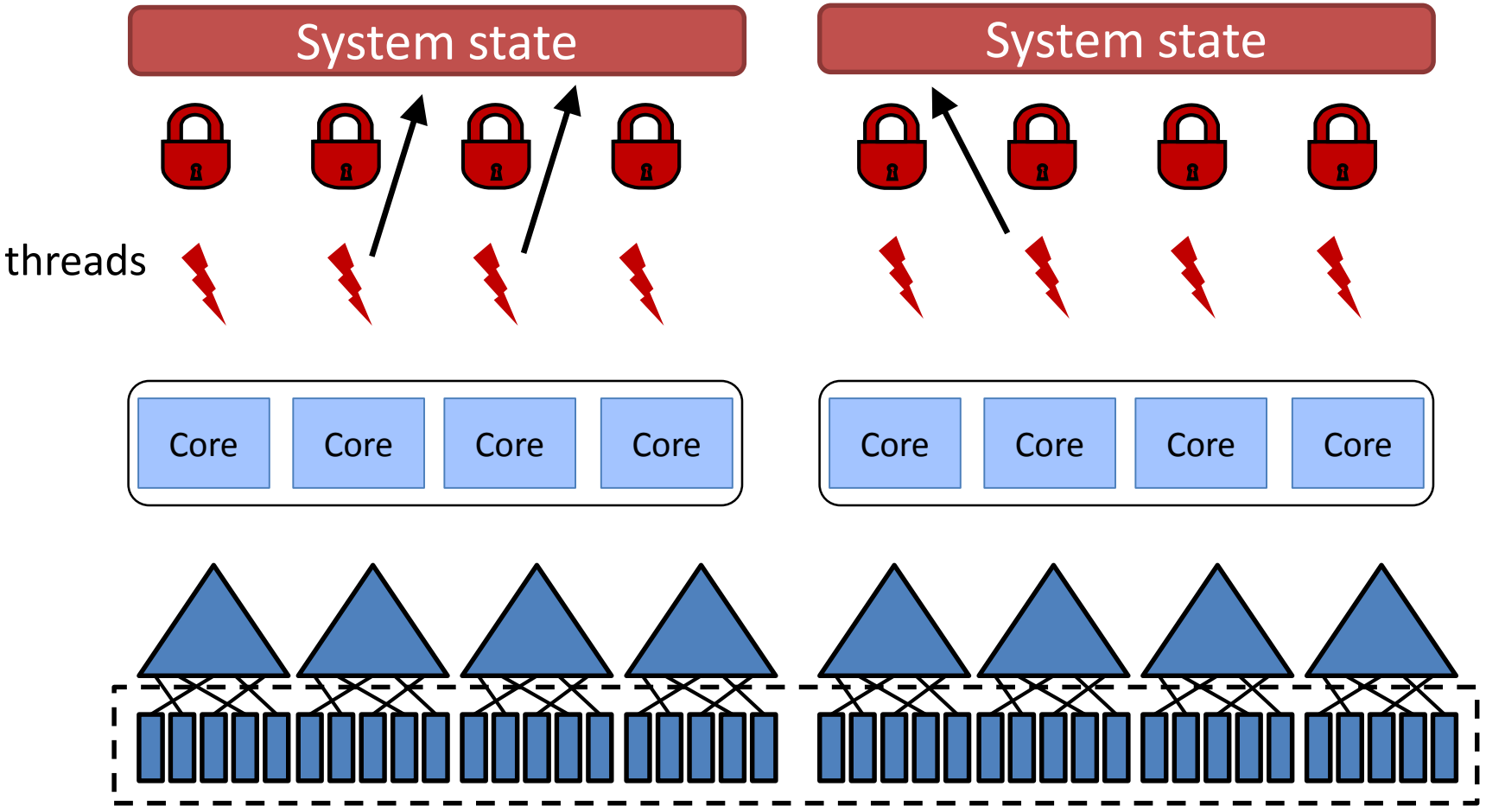
*I. Pandis, et al: PLP: Page Latch-free Shared-everything OLTP, VLDB 2011

Perfectly partitionable workload

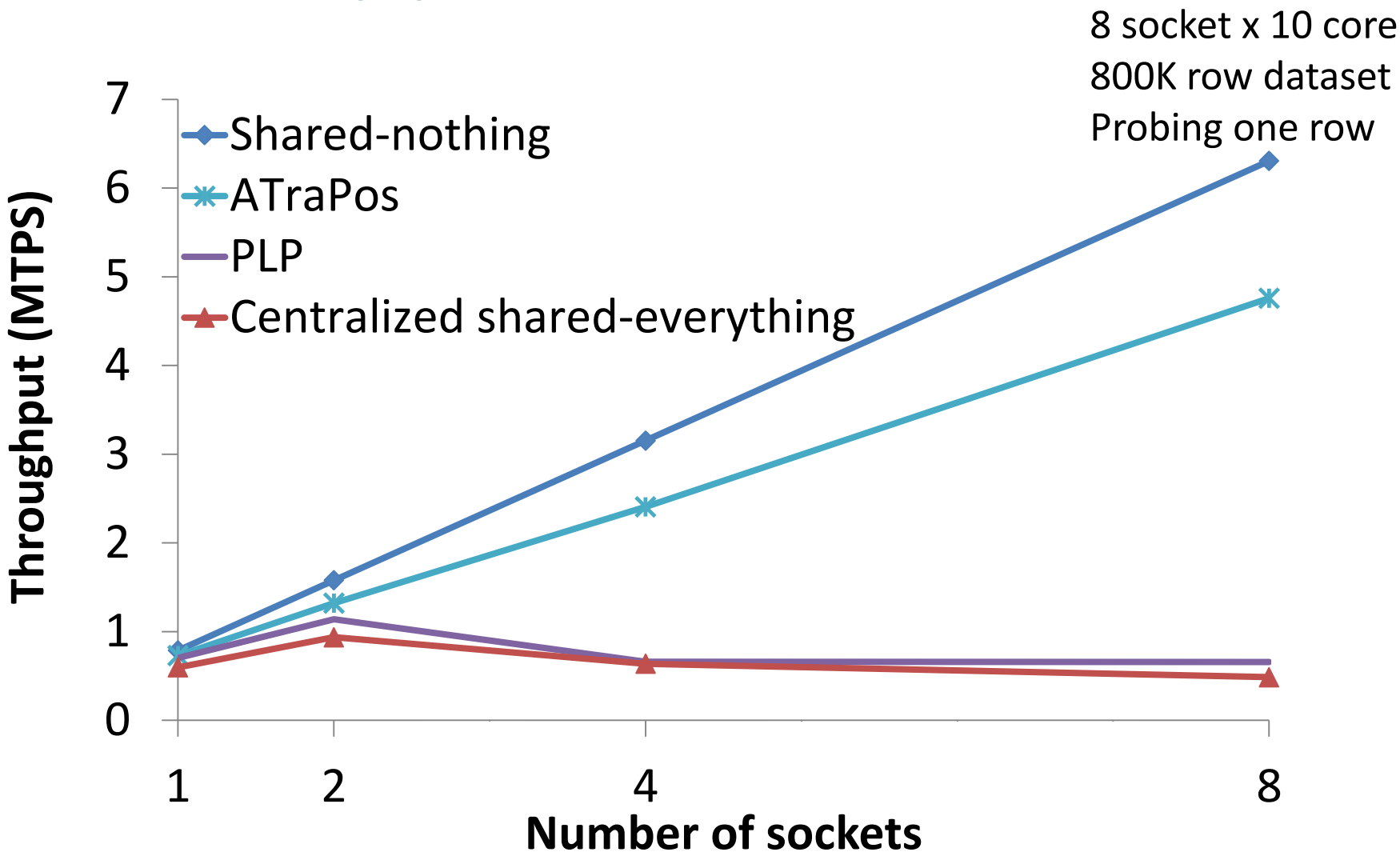


Inter-socket accesses to system state are a bottleneck

ATraPos: Island-aware SE



Perfectly partitionable workload



Island awareness brings scalability

Outline

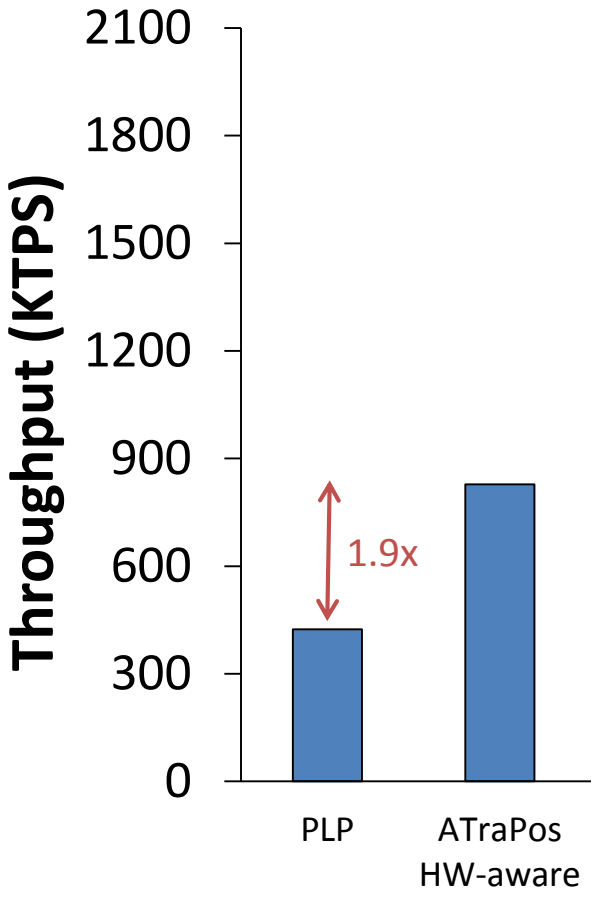
- Impact of Hardware Islands on OLTP

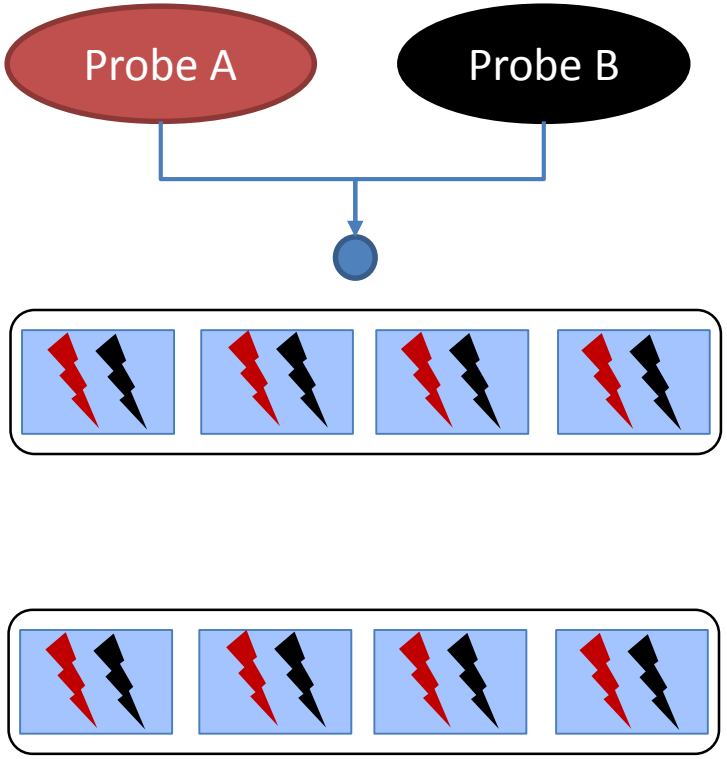
- **ATraPos**
  - Avoiding unnecessary synchronization
  - **Workload & hardware-aware partitioning & placement**
  - Lightweight monitoring & repartitioning

- Summary

Naive partitioning and placement

8 socket x 10 core
800K rows per table
Probing 1 row each from **A** and **B**

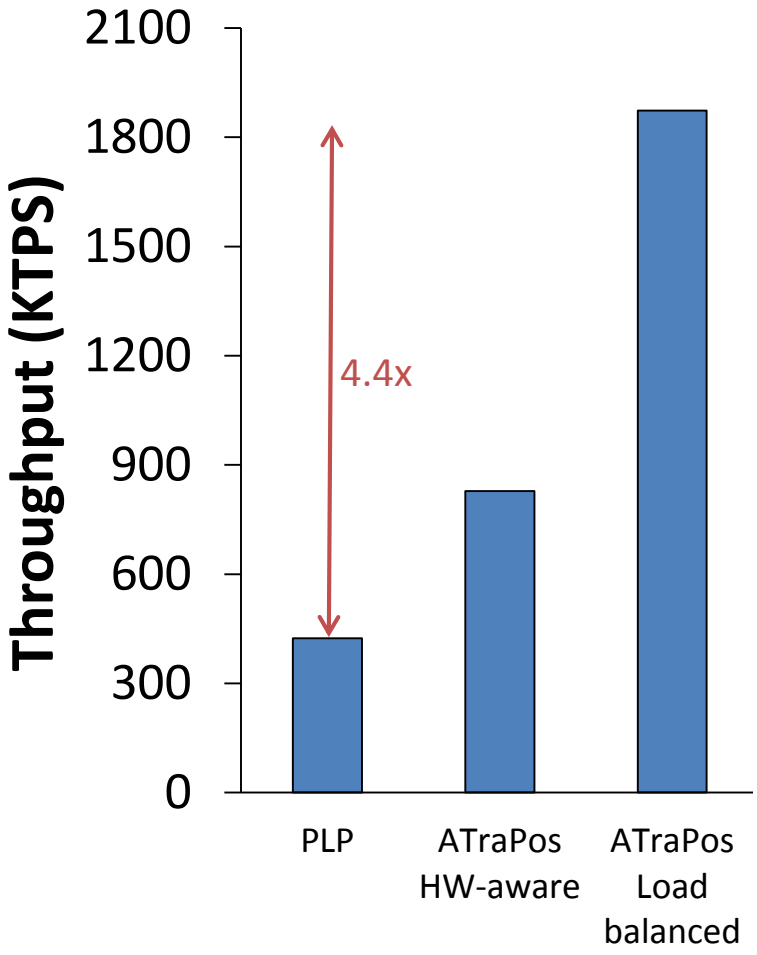


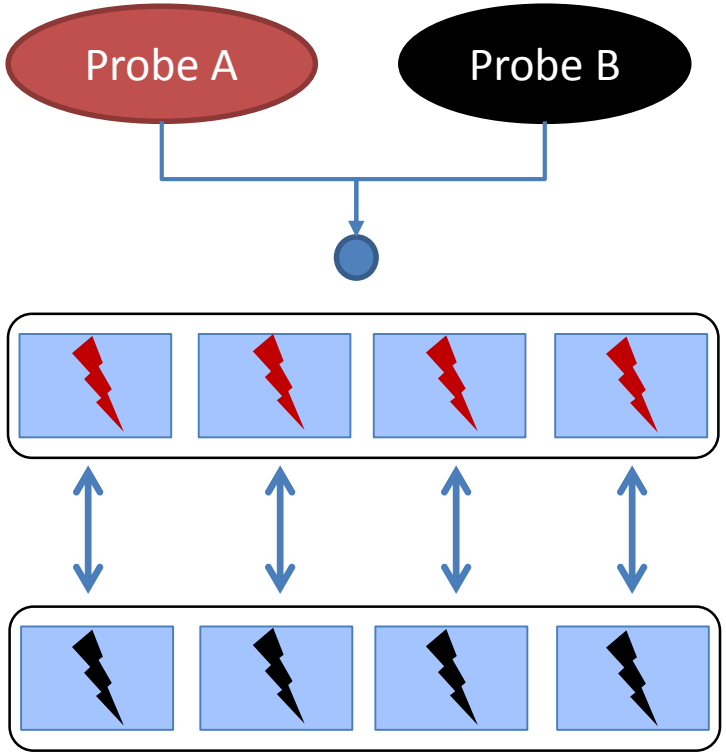


Cores are overloaded with contending threads

ATraPos partitioning and placement

8 socket x 10 core
800K rows per table
Probing 1 row each from **A** and **B**

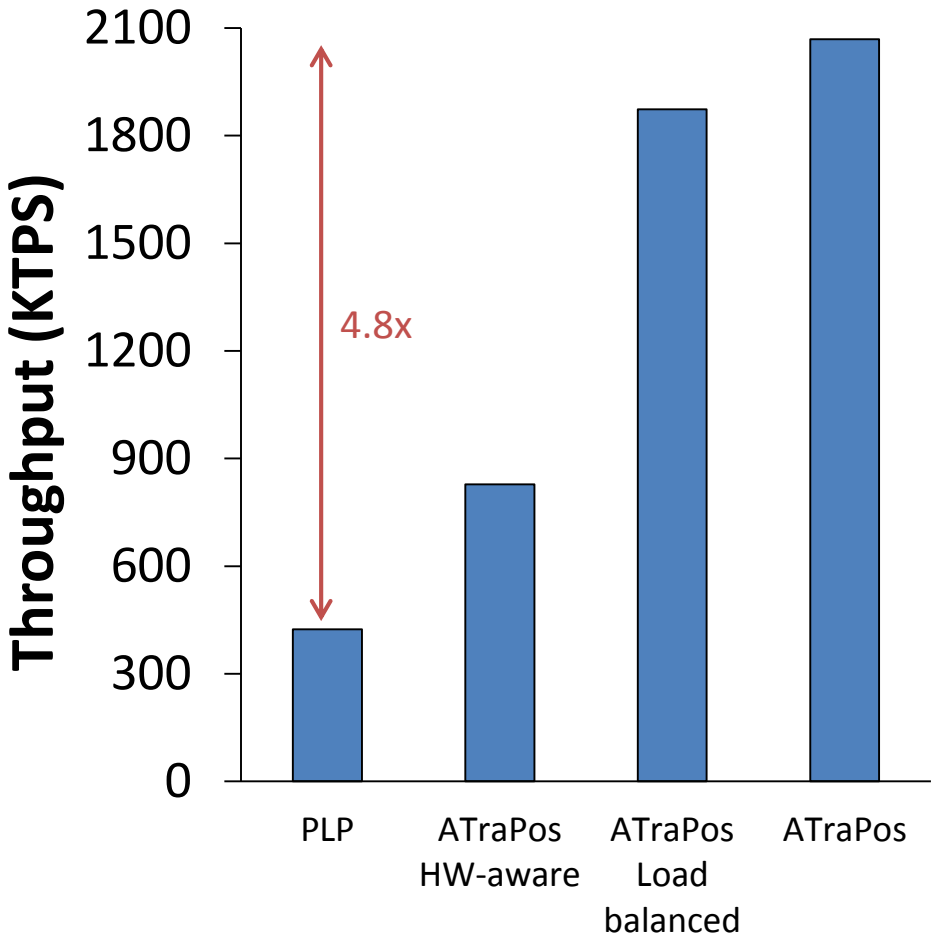


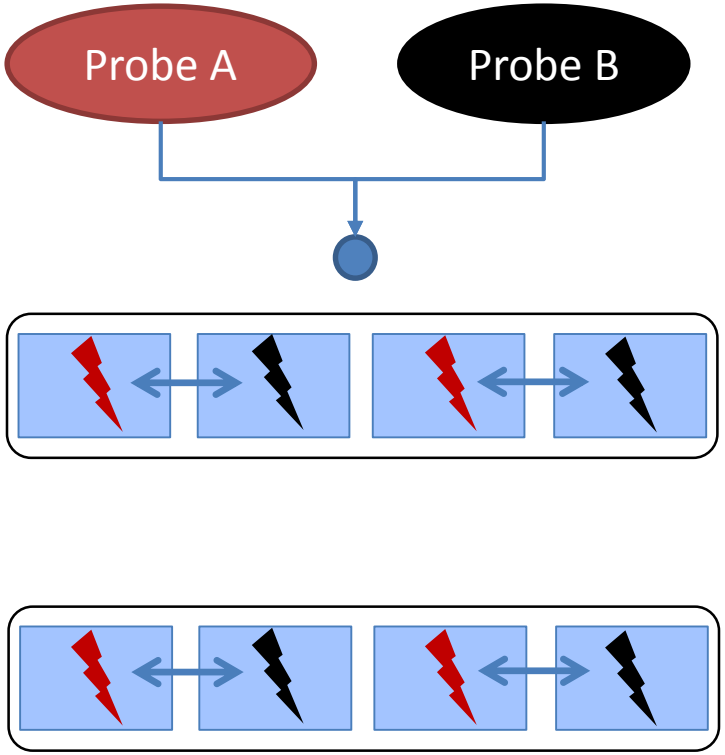


Ignoring Islands -> synchronization overhead

ATraPos partitioning and placement

8 socket x 10 core
800K rows per table
Probing 1 row each from **A** and **B**





ATraPos: balanced load + reduced synchronization

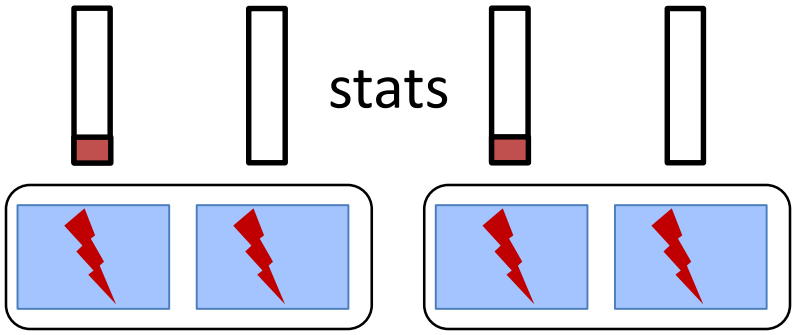
Outline

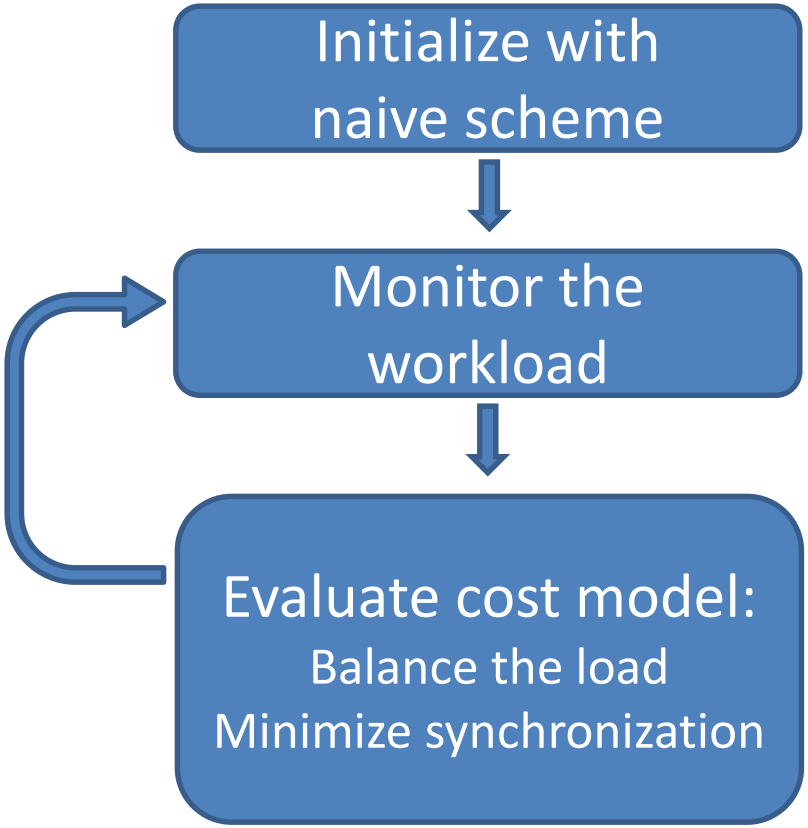
- Impact of Hardware Islands on OLTP

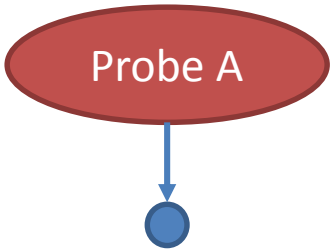
- **ATraPos**
  - Avoiding unnecessary synchronization
  - Workload & hardware-aware partitioning & placement
  - **Lightweight monitoring & repartitioning**

- Summary

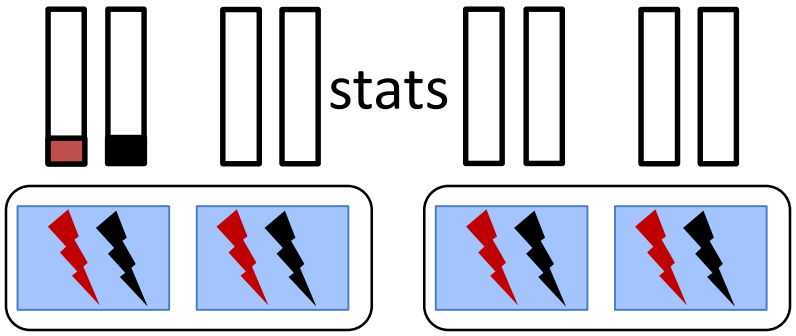
ATraPos monitoring

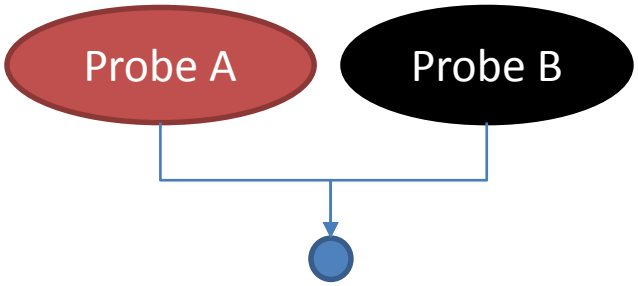


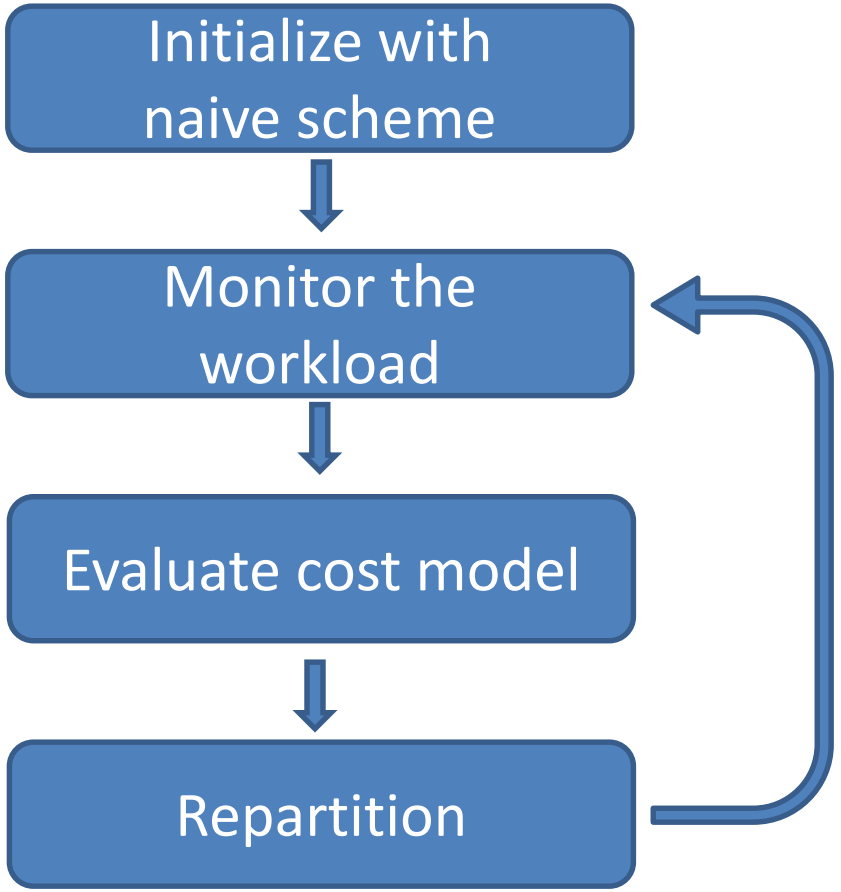




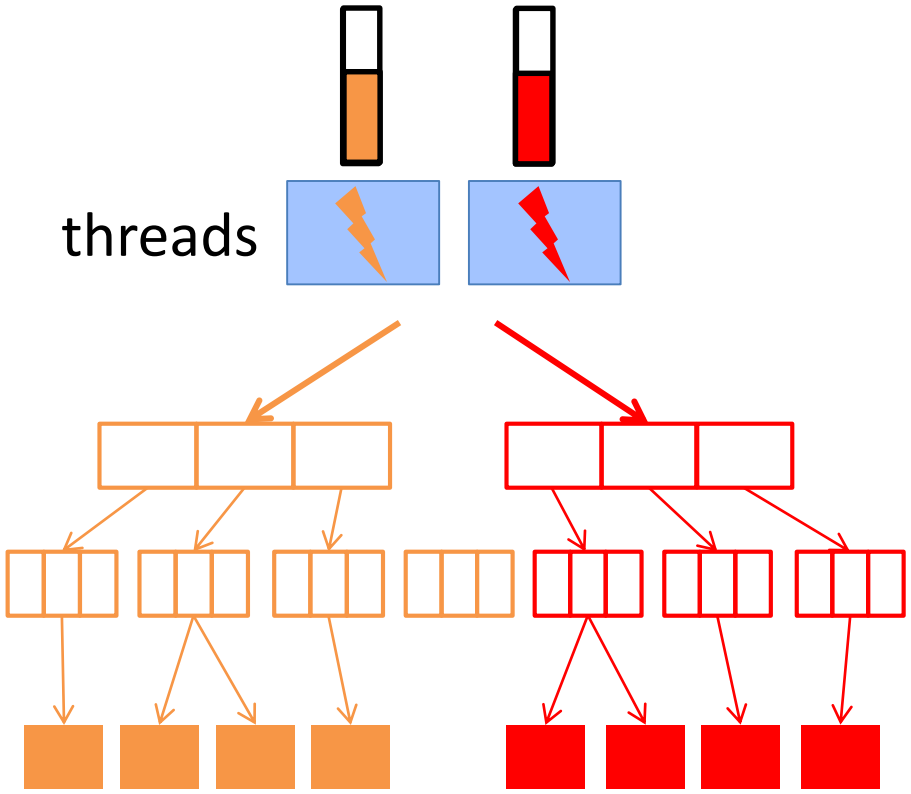
ATraPos monitoring





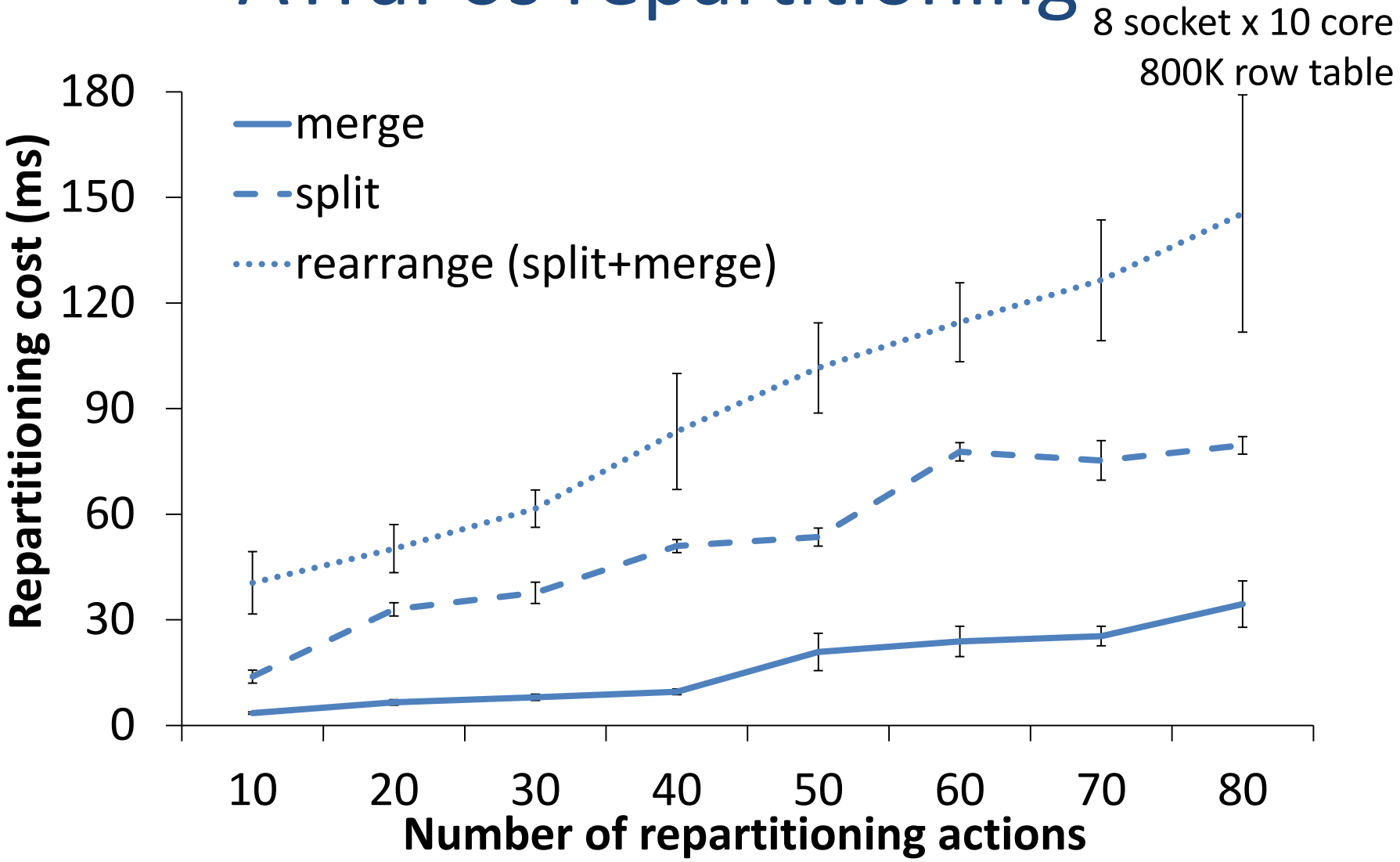


Repartitioning Multi-Rooted B-trees



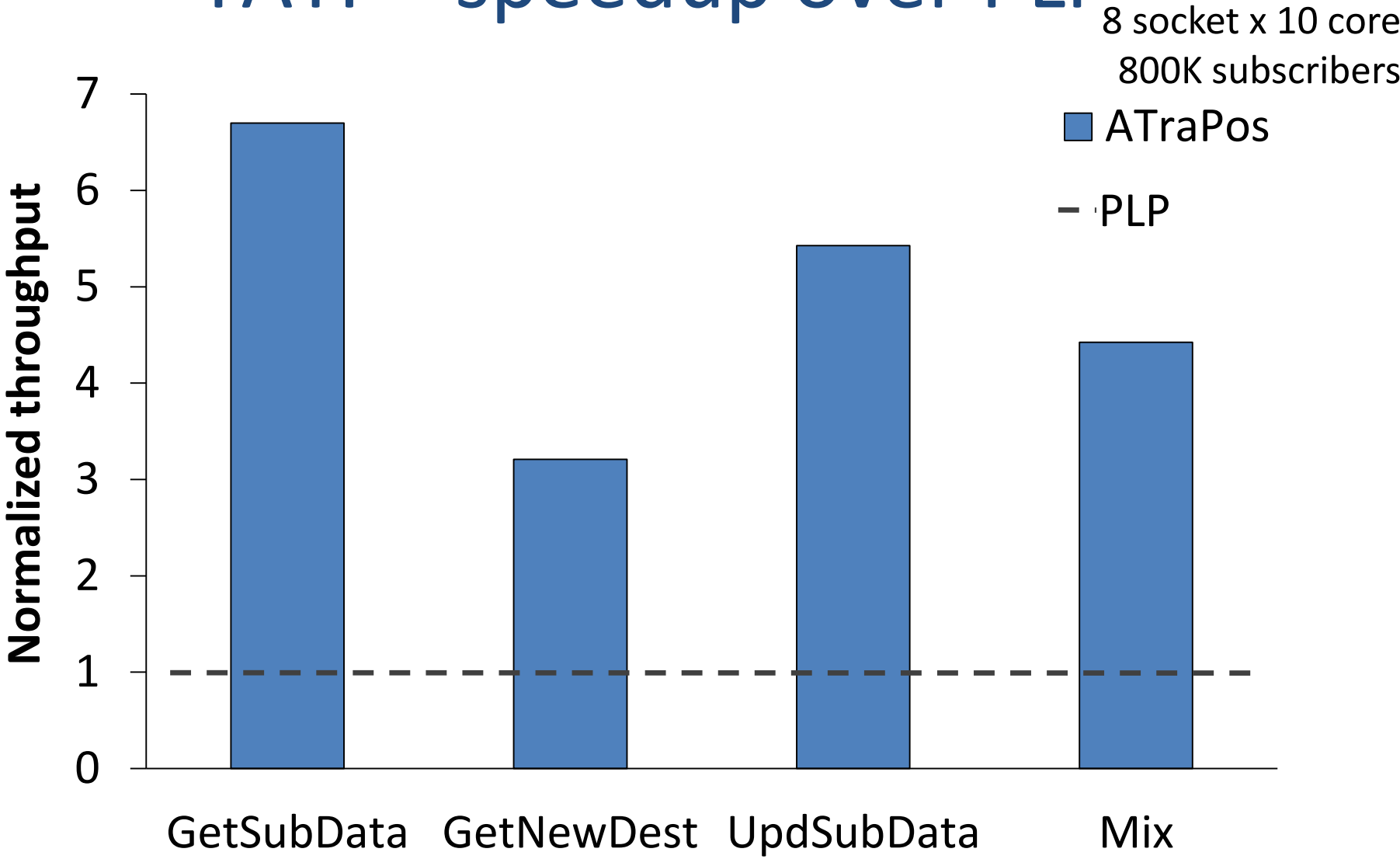
Splitting and merging B-trees accesses few pages

ATraPos repartitioning



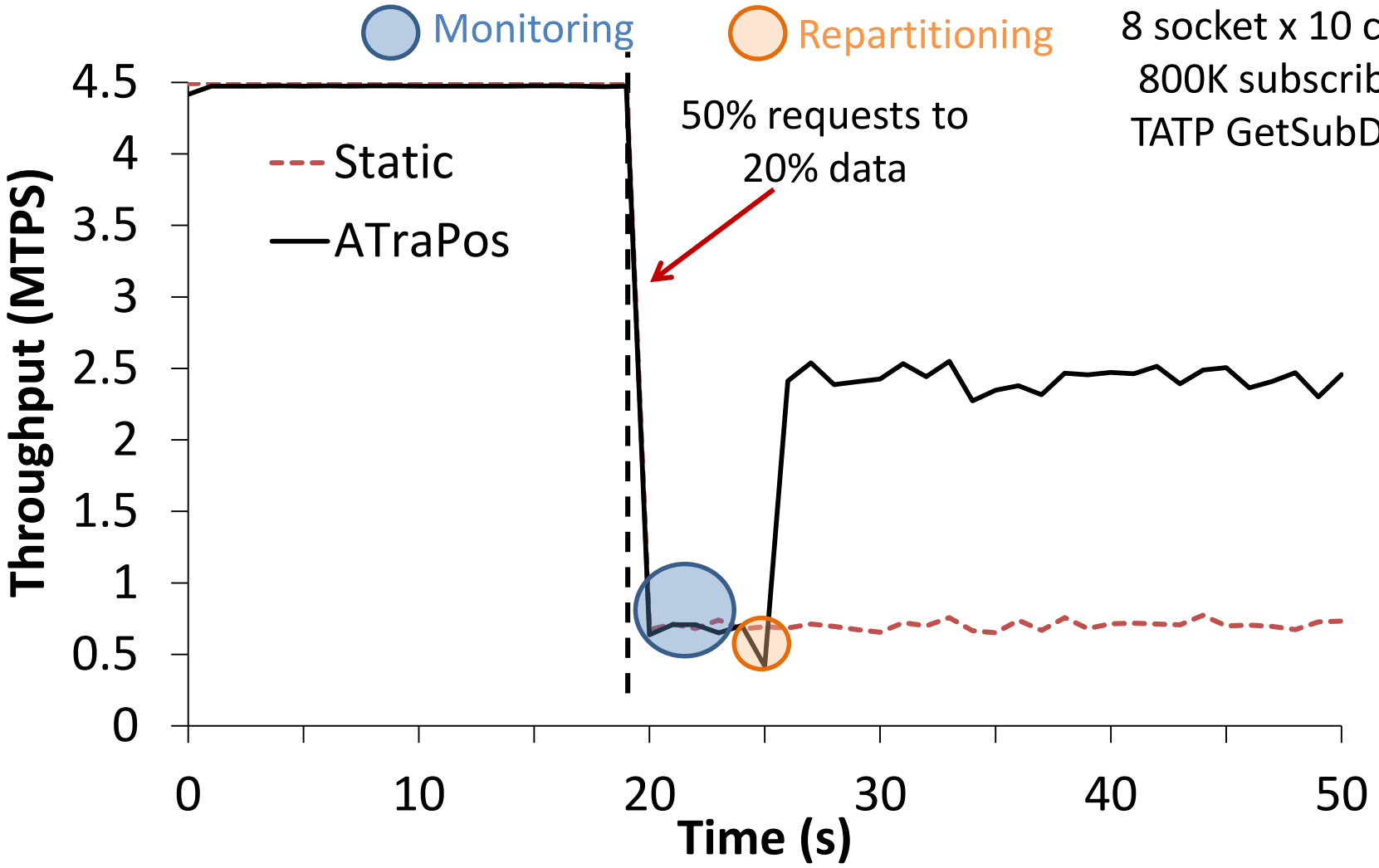
Repartitioning of a table takes < 200ms

TATP - speedup over PLP



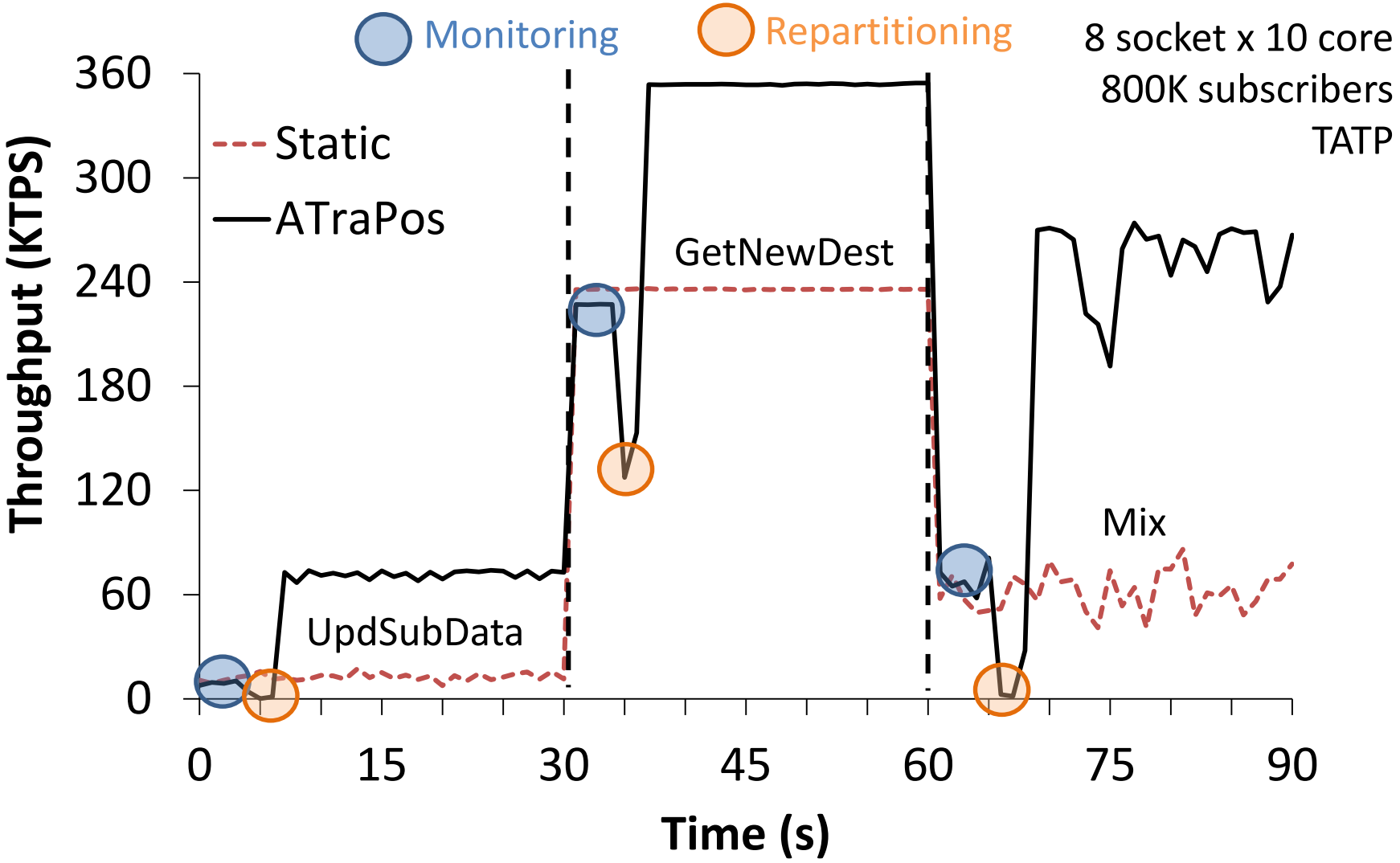
ATraPos improves performance of TATP by 3.1-6.7x

Adapting to workload skew



ATraPos detects skew and quickly adapts

Adapting to changing workload type



ATraPos gracefully adapts to any change

ATraPos: Adaptive OLTP for Islands

- Challenges
  - Optimal configuration depends on workload and hardware
  - Expensive repartitioning due to physical data movement

- ATraPos
  - Minimal inter-socket accesses in the critical path
  - Workload & hardware-aware partitioning & placement
  - Lightweight monitoring and repartitioning

Thank you!