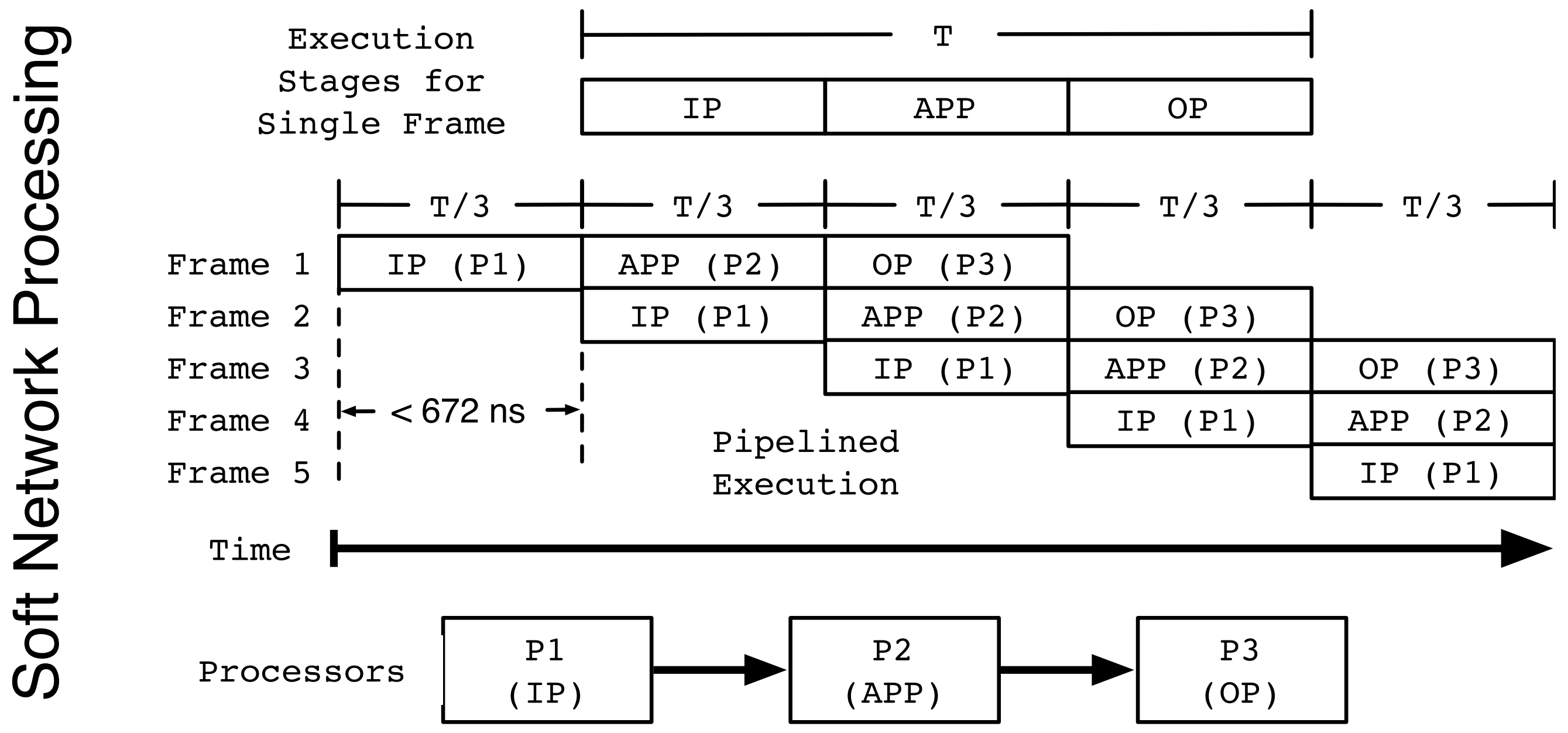
FastForward for Efficient Pipeline Parallelism

John Giacomoni, Tipp Moseley,
and Manish Vachharajani
University of Colorado

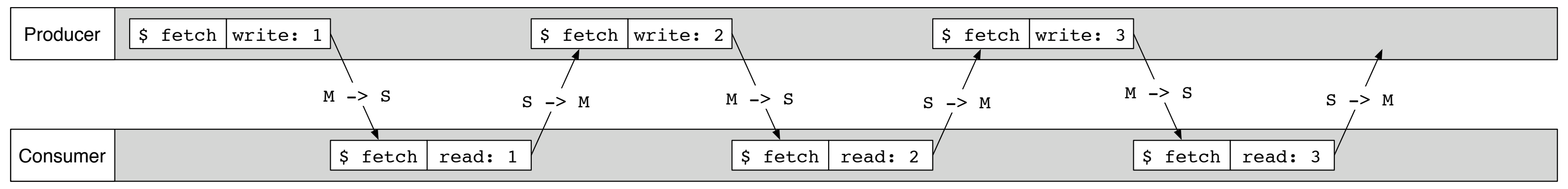
Parallelizing Sequential Tasks:

Need to efficiently support the parallelization of tasks with partially or totally ordered data. Pipeline-parallel organizations avoid contention and stalls caused by intra-thread synchronization other organizations.

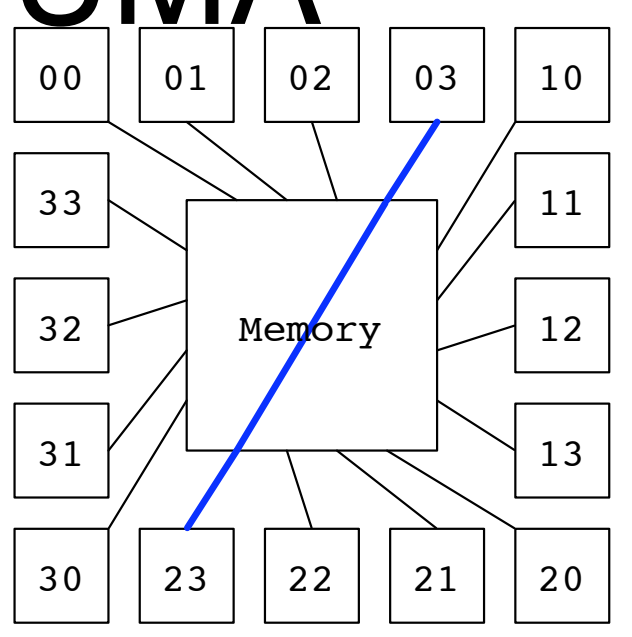


Hardware Performance Issues

Cache Thrashing (Communication Channel)

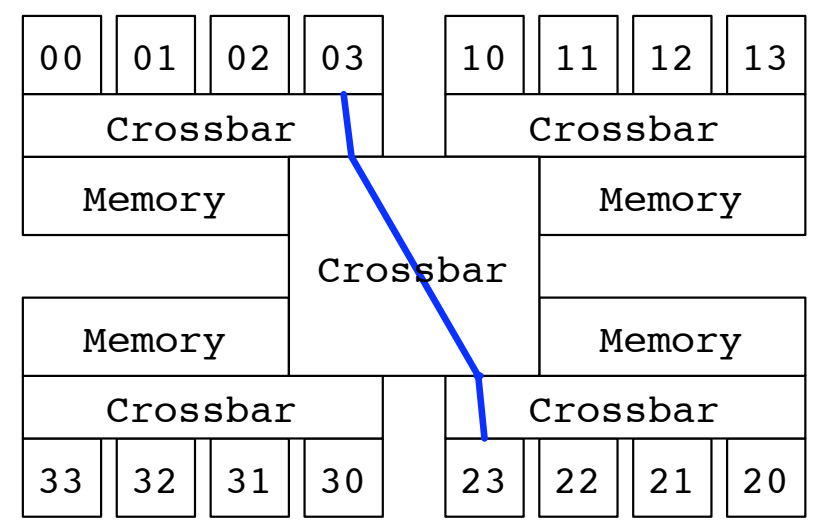


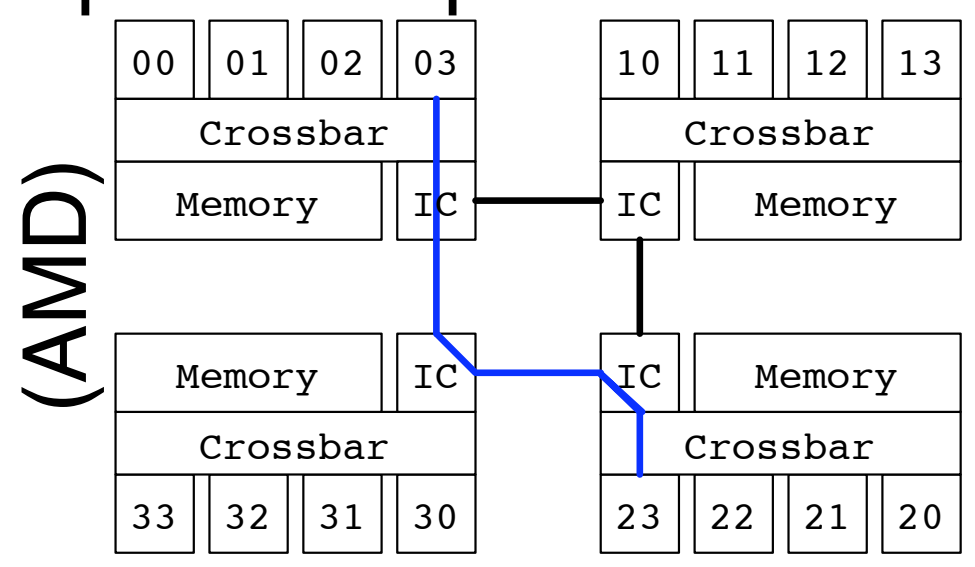
UMA



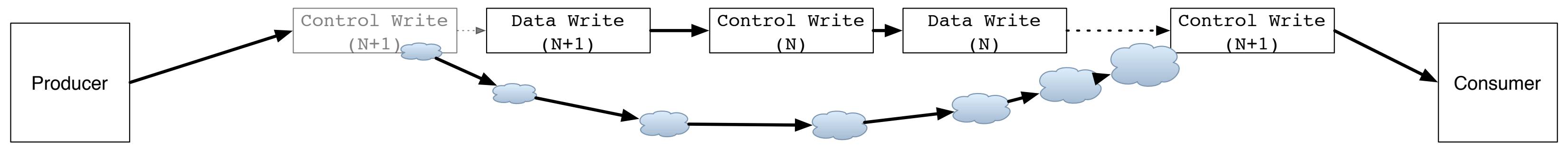
NUMA

Careful resource scheduling may be required for performance.





Weak Memory Models



Weak memory models can exhibit unexpected behavior;
In this example, a write passing a write may result in a read of a stale data value.

The FastForward Solution

```
enqueue_lamport(...) {
  if(NEXT(head) == tail) {
    // Handle full queue.
  }
  buf[head] = data;
  head = NEXT(head);
 }

dequeue_lamport(...) {

  if (head == tail) {
    // Handle empty queue.
  }
  data = buf[tail]
  tail = NEXT(tail)
}
```

```
enqueue_fastforward(...) {
  if(NULL != buf[head]) {
    // Handle full queue.
  }
  buf[head] = data;
  head = NEXT(head);
 }

dequeue_fastforward(...) {
  data = buf[tail];
  if (NULL == data) {
    // Handle empty queue.
  }

  tail = NEXT(tail)
}
```

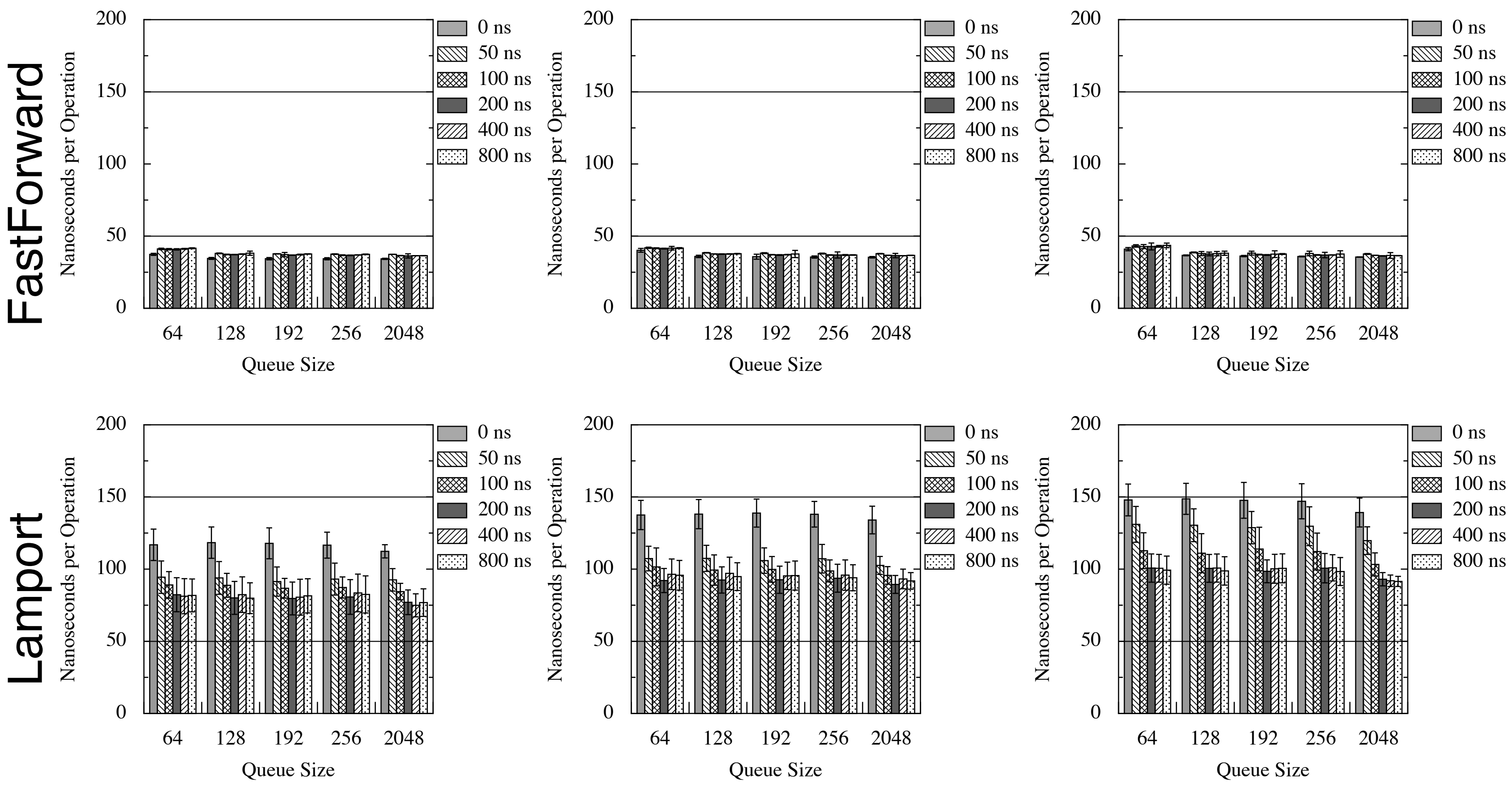
Decoupling at the cache coherence layer can eliminate cache thrashing, hide non-uniform memory access issues, and support weak memory models.

Proof Sketch

The references [1] prove that "in the program order of the consumer, the consumer dequeues values in the same order that they were enqueued in the producer's program order," for strong to weakly ordered consistency models, showing that FastForward works even on relaxed memory models.

[1] J. Giacomoni, T. Moseley, and M. Vachharajani. FastForward for efficient pipeline parallelism. Technical Report CU-CS-1028-07, Univerity of Colorado at Boulder, 2007.

Performance Comparison



2 Threads On-die

2 Threads Cross-die

3 Threads Cross-die

Dual-processor Dual-core 2.0 GHz AMD Opteron

Conclusions

1) Decoupling communicating threads at the cache layer on ccNUMA machines may yield significant performance improvements.
2) FastForward provides an efficient point-to-point communication primitive ideally suited to pipeline-parallel applications.
   a) Consistent performance.
   b) Software only solution.
   c) Correct under strong to weakly ordered consistency models.
   d) May provide performance improvements to other streaming parallel organizations.

UNIVERSITY OF COLORADO computer SCIENCE AT BOULDER

UNIVERSITY OF COLORADO AT BOULDER
ELECTRICAL & COMPUTER ENGINEERING

UNIVERSITY OF COLORADO AT BOULDER
CORE RESEARCH LAB