Development Support for Concurrent Threaded Pipelining

John Giacomoni

Core Research Laboratory

Dr. Manish Vachharajani

Brian Bushnell, Graham Price, Rohan Ambli

University of Colorado at Boulder

2007.02.10

Outline

- Problem
- Concurrent Threaded Pipelining
  - High-rate Networking
- Communication
  - FastForward
  - Results
- Additional Development Support

Problem

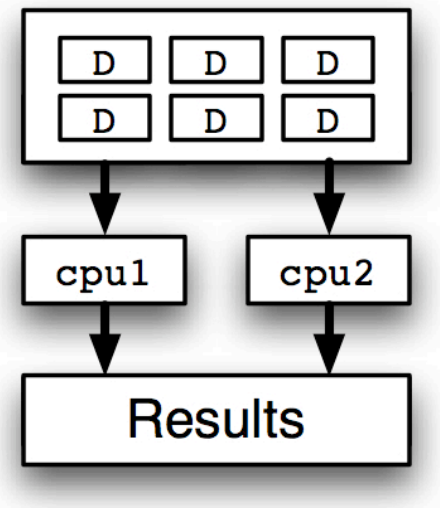
- UP performance at “end of life”
- Chip-Multiprocessor systems
  - Individual cores less powerful than UP
  - Asymmetric and Heterogeneous
  - 10-100s of cores
- How to program?

PL Support

- Programmers are:
  - Bad at explicitly parallel programming
    - FShm architectures make this easier
  - Better at sequential programming
- Hide parallelism
  - Compilers
  - Sequential libraries?
    - Math, iteration, searching, and ??? routines

Using Multi-Core

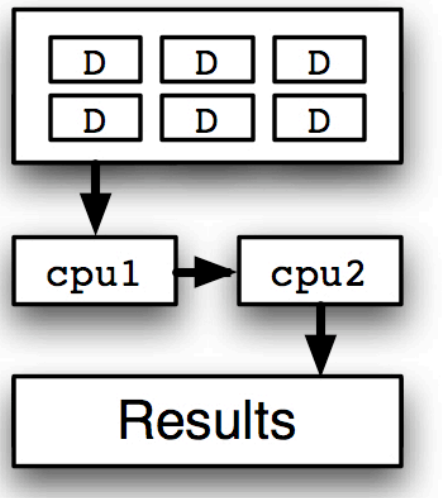
- Task Parallelism
  - Desktop - easy
- Data Parallelism
  - Web serving - “easy”



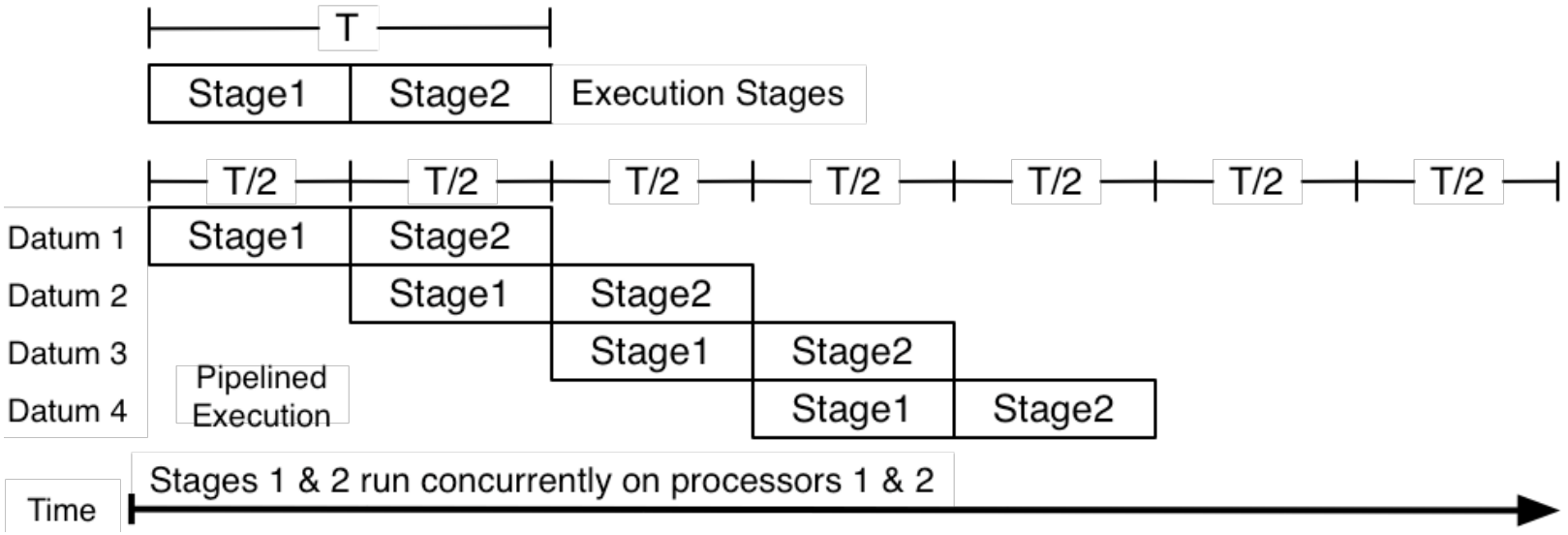
- Sequential applications
  - HARD (data dependencies)
    - Ex: Video Decoding
    - Ex: Network Processing

A Solution: Concurrent Threaded Pipelining

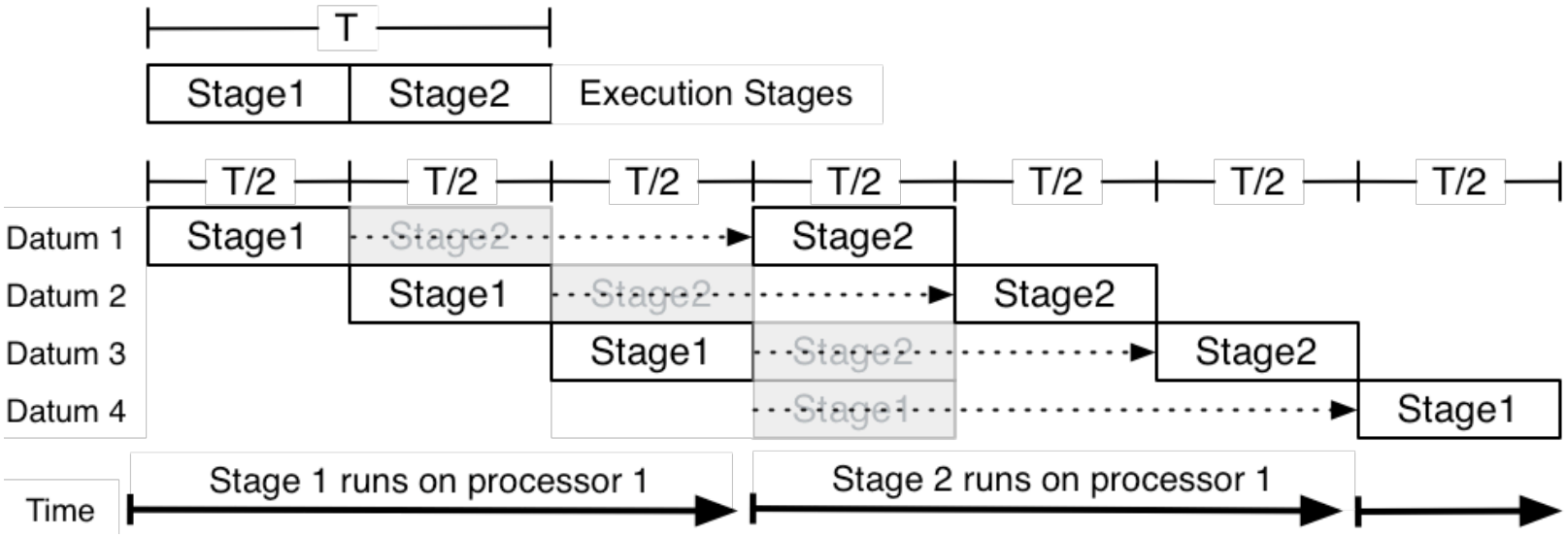
- Arrange applications as pipelines
  - (Pipeline-parallel)
  - Each stage bound to a processor
  - Sequential data flow
    - Data Hazards are a problem
- Software solution
  - Frame Shared Memory (FShm)
  - FastForward



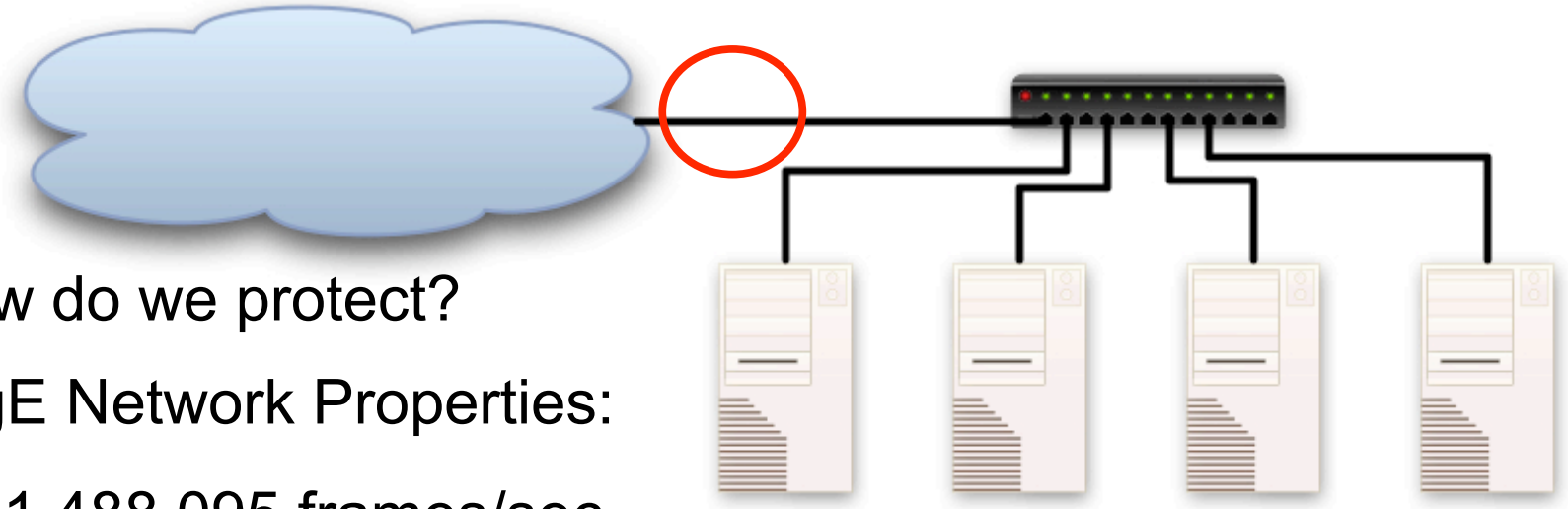
Concurrent Threaded Pipelining



Sequential Threaded Pipelining



Network Scenario

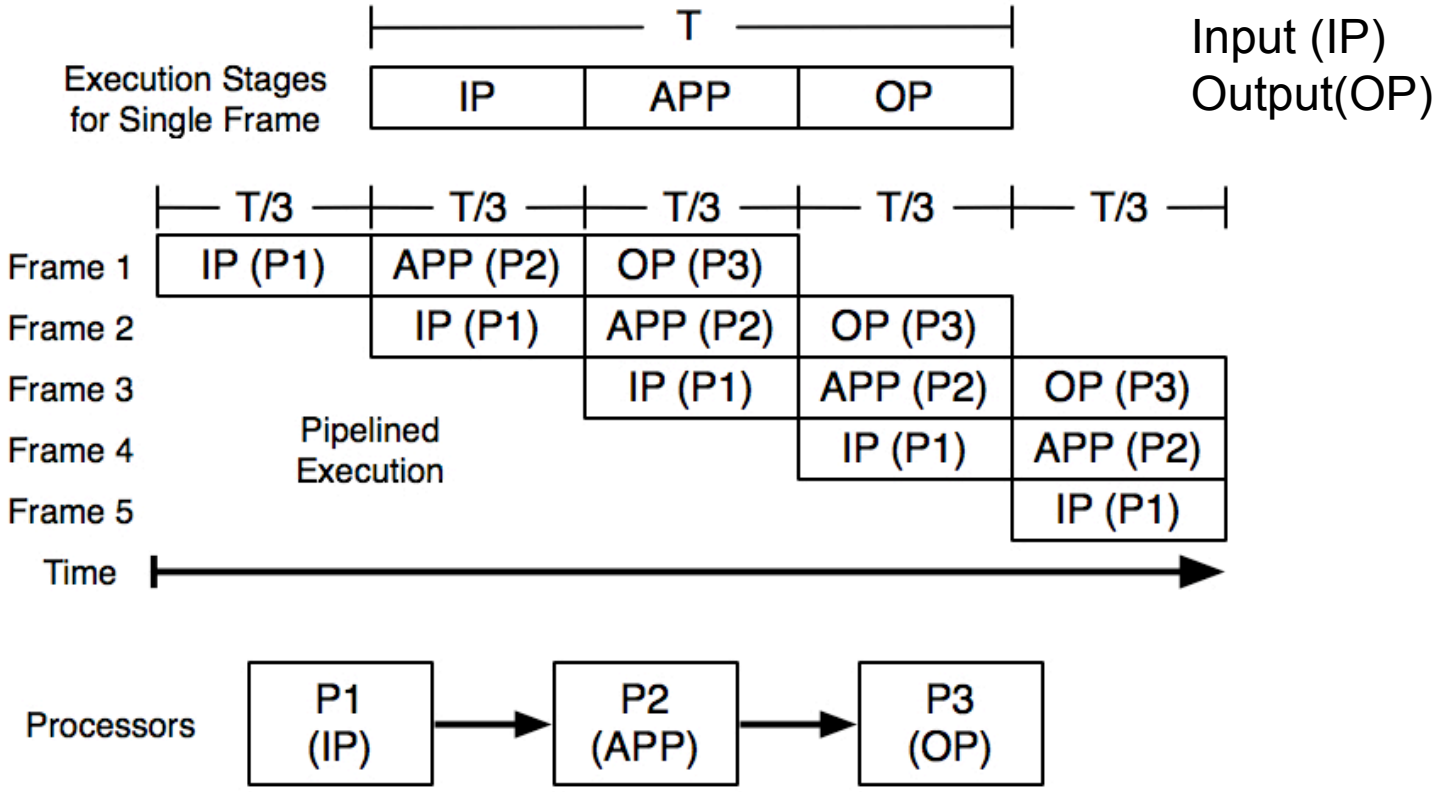


- How do we protect?
- GigE Network Properties:
  - 1,488,095 frames/sec
  - 672 ns/frame
  - Frame dependencies

FShm

- Frame Shared Memory
  - User programming environment
  - Linked Kernel and User-Space Threads
    - Input (Kernel)
    - Processing (User-Space)
    - Output (Kernel)
  - GigE frame forwarding (672ns/frame)

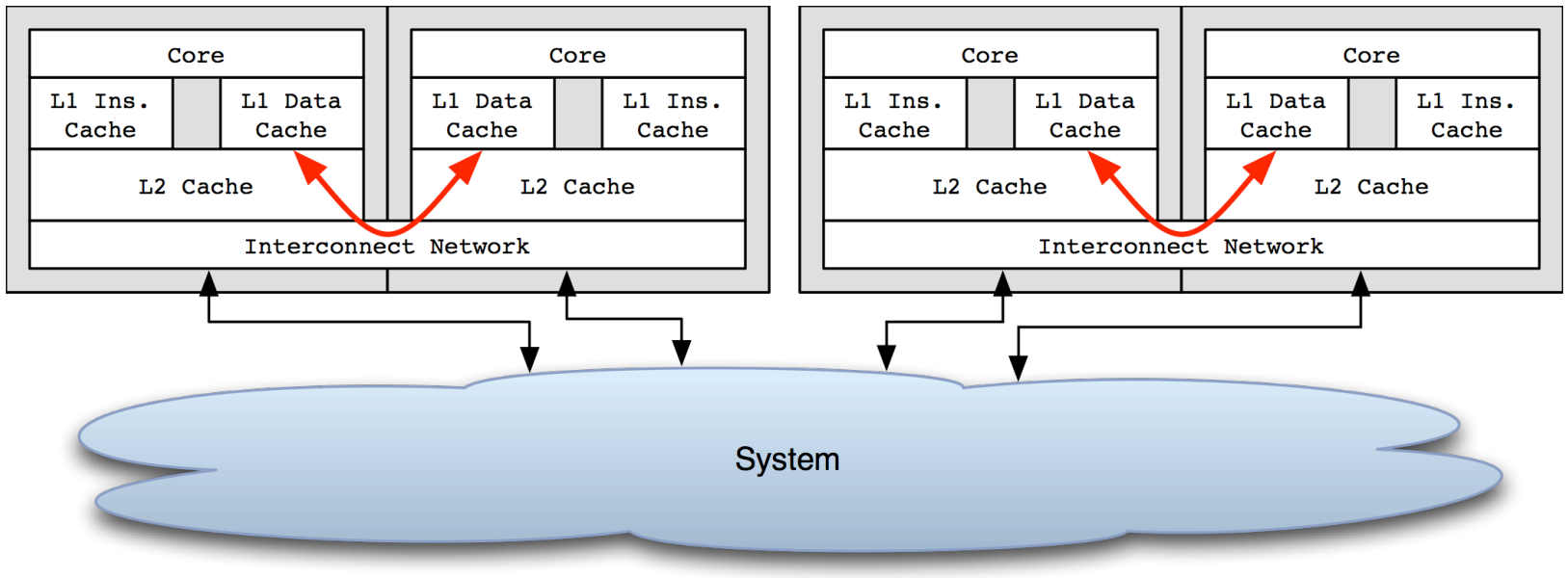
FShm
Network Pipelining



Some Other Existing Work

- Decoupled Software Pipelining
  - Automatic pipeline extraction
  - Modified IMPACT compiler
  - Assumes hardware queues
- Stream-oriented languages
  - StreamIt, Cg, etc…

AMD Opteron Structure



Communication is Critical for CTPs

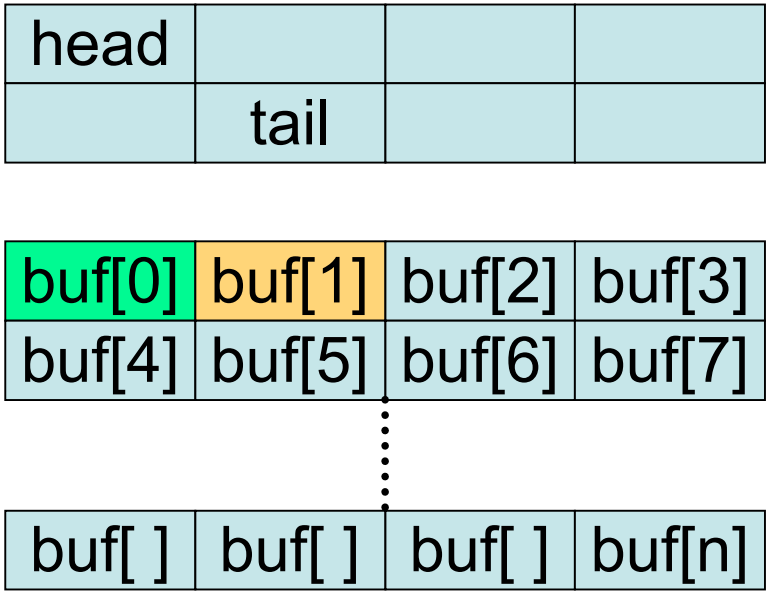
- Hardware modifications, expensive ($$$)
  - DSWP (<= 100 cycles)
- Software communication
  - Serializing queues (locks)
    - >= ~ 600 ns (per operation)
- How to forward faster ?
  - Concurrent Lock-Free Queues
    - Point-to-Point CLF queues (Lamport ‘83)
    - ~ 200 ns per operation
    - Good… Can we do better?

FastForward

- Portable software only framework
  - Architecturally tuned CLF queues
    - Works with all consistency models
    - Temporal slipping & prefetching to hide die-die communication

    ~35-40ns/queue operation
    - Core-core & die-die
    - Fits within DSWP's performance envelope
  - Cross-domain communication
    - Kernel/Process/Thread

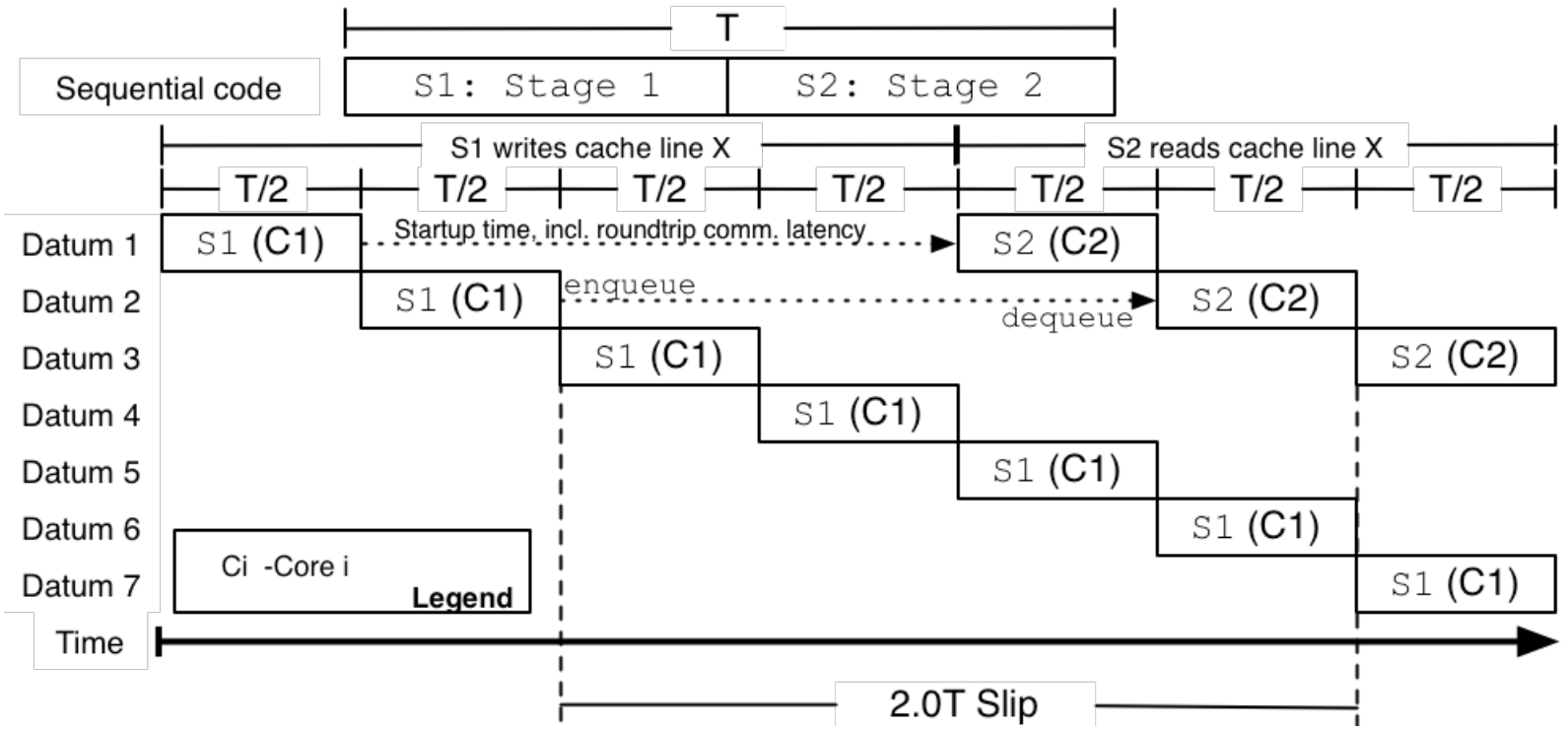
Optimized CLF Queues



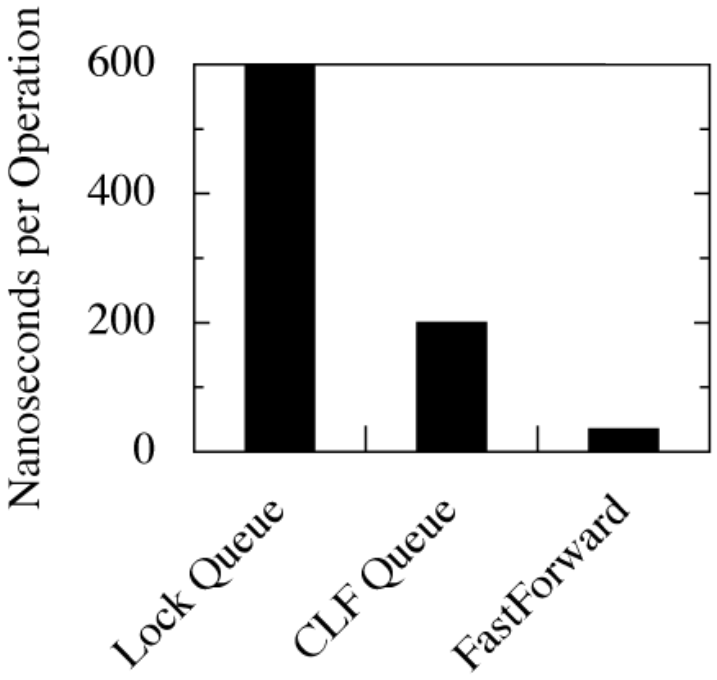

```
ff_enqueue(data) {
    while(0 != buf[head]);

    buf[head] = data;
    head = NEXT(head);
}
```

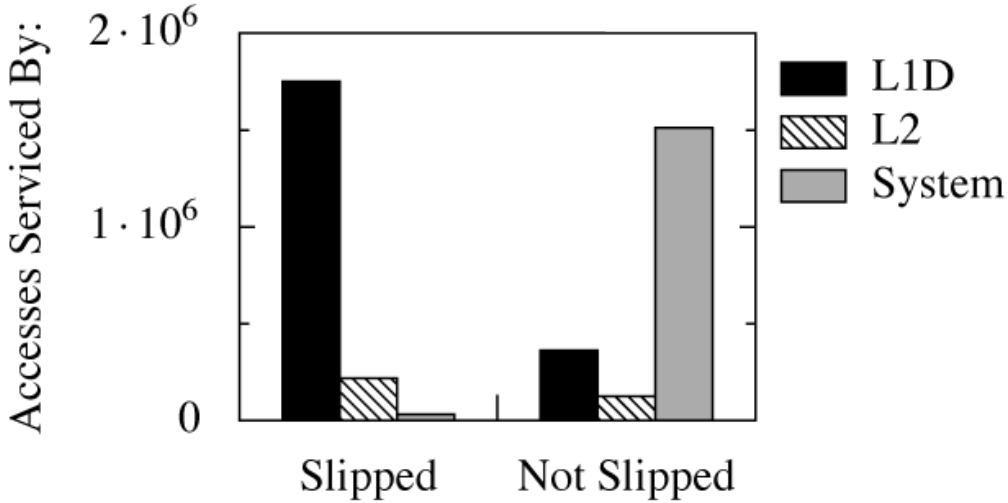
Observe how head/tail cachelines will NOT ping-pong.
BUT, “buf” will still cause the cachelines to ping-pong.

Slip Timing


T
Sequential code
S1: Stage 1
S2: Stage 2
S1 writes cache line X
S2 reads cache line X
T/2
T/2
T/2
T/2
T/2
T/2
T/2
Datum 1
Datum 2
Datum 3
Datum 4
Datum 5
Datum 6
Datum 7
Time
S1 (C1)
Startup time, incl. roundtrip comm. latency
S2 (C2)
enqueue
dequeue
Ci -Core i
Legend
2.0T Slip

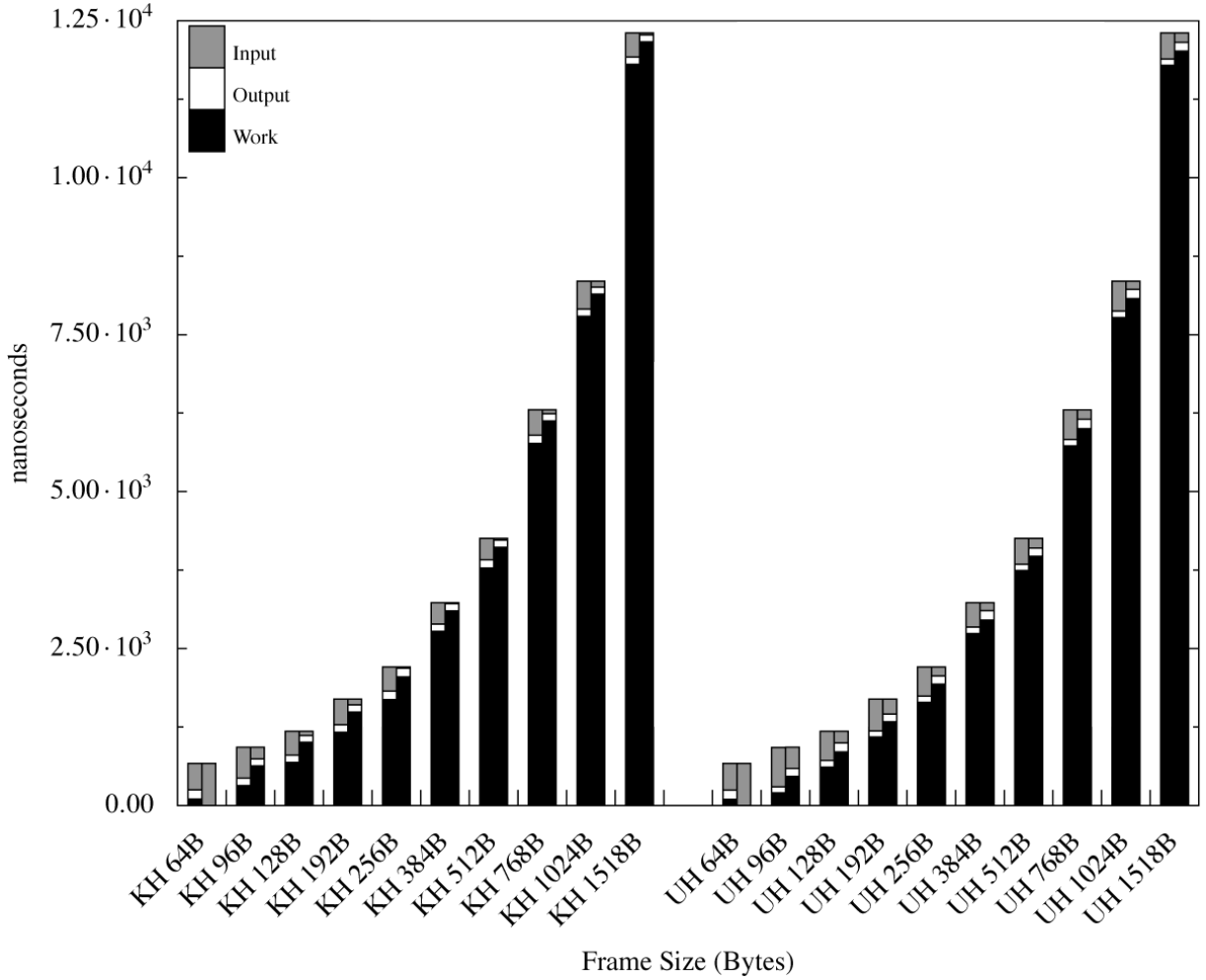
Performance





Performance FShm

Forwarding



Additional Development Support

- OS support
- Identification and Visualization tools

OS Support

- Hardware virtualization
  - Asymmetric and heterogeneous cores
  - Cores may not share main memory (GPU)
- Pipelined OS services
- Pipelines may cross process domains
  - FShm
  - Each domain should keep its private memory
    - Protection
- Need label for each pipeline
  - Co/gang-scheduling of pipelines

Reported Results

- http://www.cs.colorado.edu/~jgiacomo/publications.html
  - John Giacomoni and Manish Vachharajani, “Harnessing Chip-Multiprocessors with Concurrent Threaded Pipelines,” Technical Report CU-CS-1024-07, University of Colorado at Boulder, January 2007.
  - John Giacomoni, Manish Vachharajani and Tipp Moseley, “FastForward for Concurrent Threaded Pipelines,” Technical Report CU-CS-1023-07, University of Colorado at Boulder, January 2007.
  - John Giacomoni, John K. Bennett, Antonio Carzaniga, Manish Vachharajani and Alexander L. Wolf, “FShm: High-Rate Frame Manipulation in Kernel and User-Space,” Technical Report CU-CS-1015-06, University of Colorado at Boulder, October 2006.

Questions?