Frame Shared Memory: Line-Rate Networking on Commodity Hardware

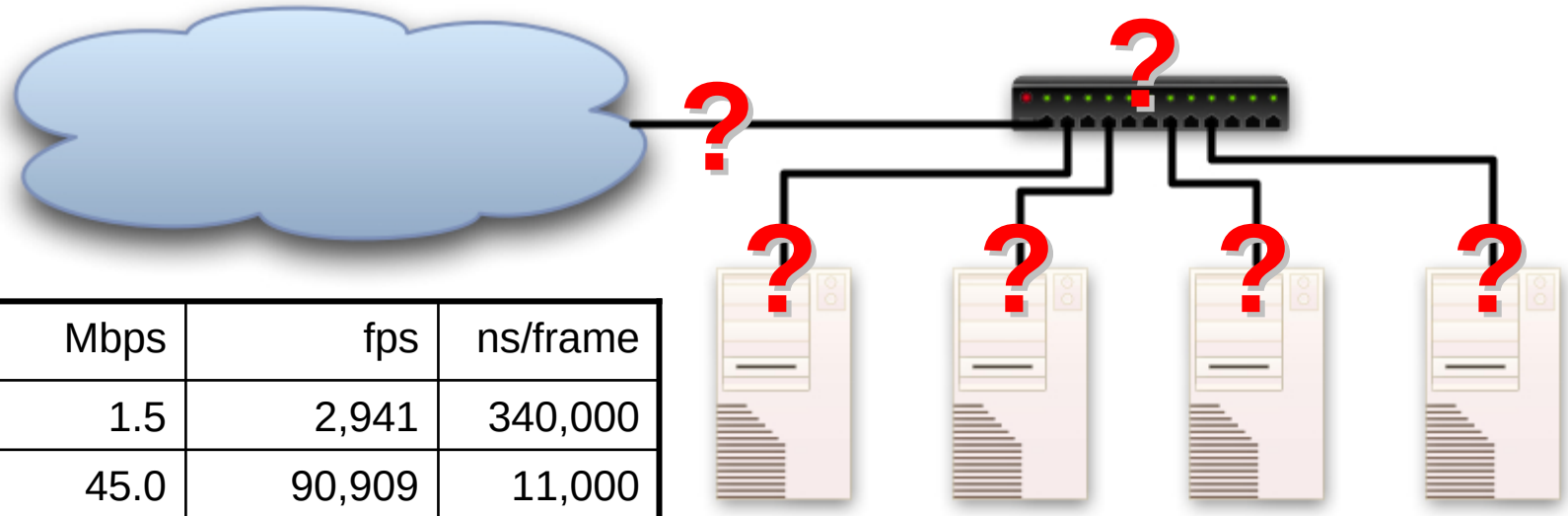
John Giacomoni

John K. Bennett, Douglas C. Sicker, and Manish Vachharajani
University of Colorado at Boulder

Alexander L. Wolf - Imperial College London
Antonio Carzaniga - University of Lugano

2007.12.03

Problem Description



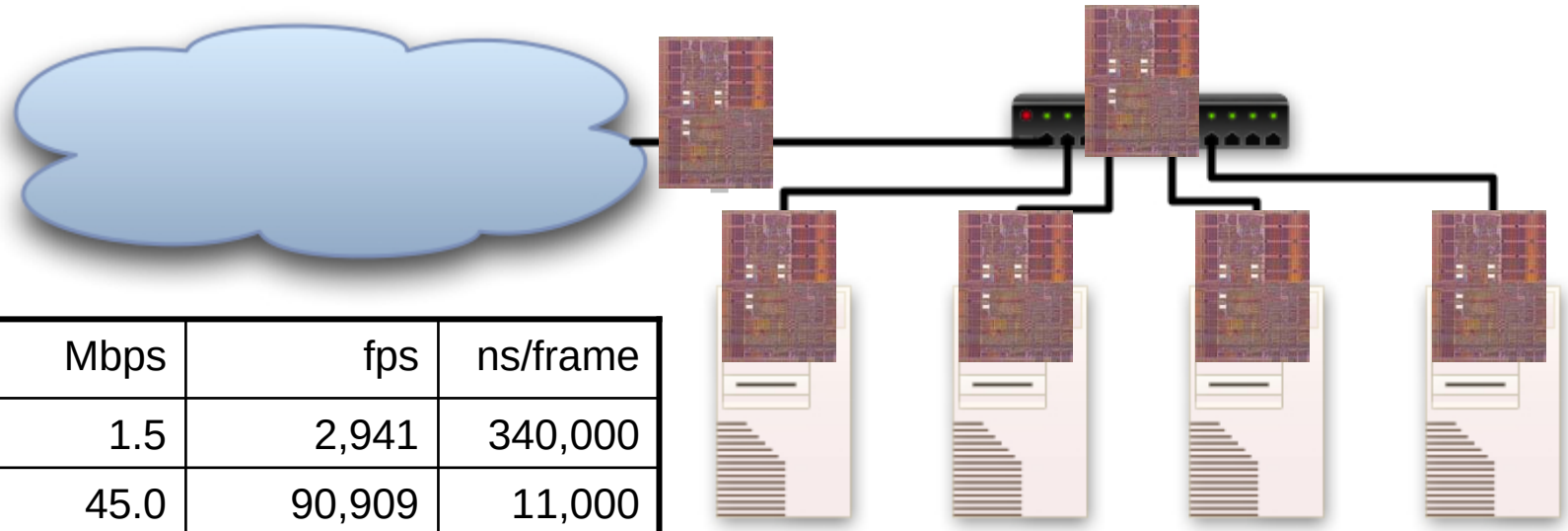
Link	Mbps	fps	ns/frame
T-1	1.5	2,941	340,000
T-3	45.0	90,909	11,000
OC-3	155.0	333,333	3,000
OC-12	622.0	1,219,512	820
GigE	*1,000.0*	*1,488,095*	*672*
OC-48	2,500.0	5,000,000	200
10 GigE	10,000.0	14,925,373	67
OC-192	9,500.0	19,697,843	51

How do we route?

How do we protect?

How do we correlate?

ASIC Solutions



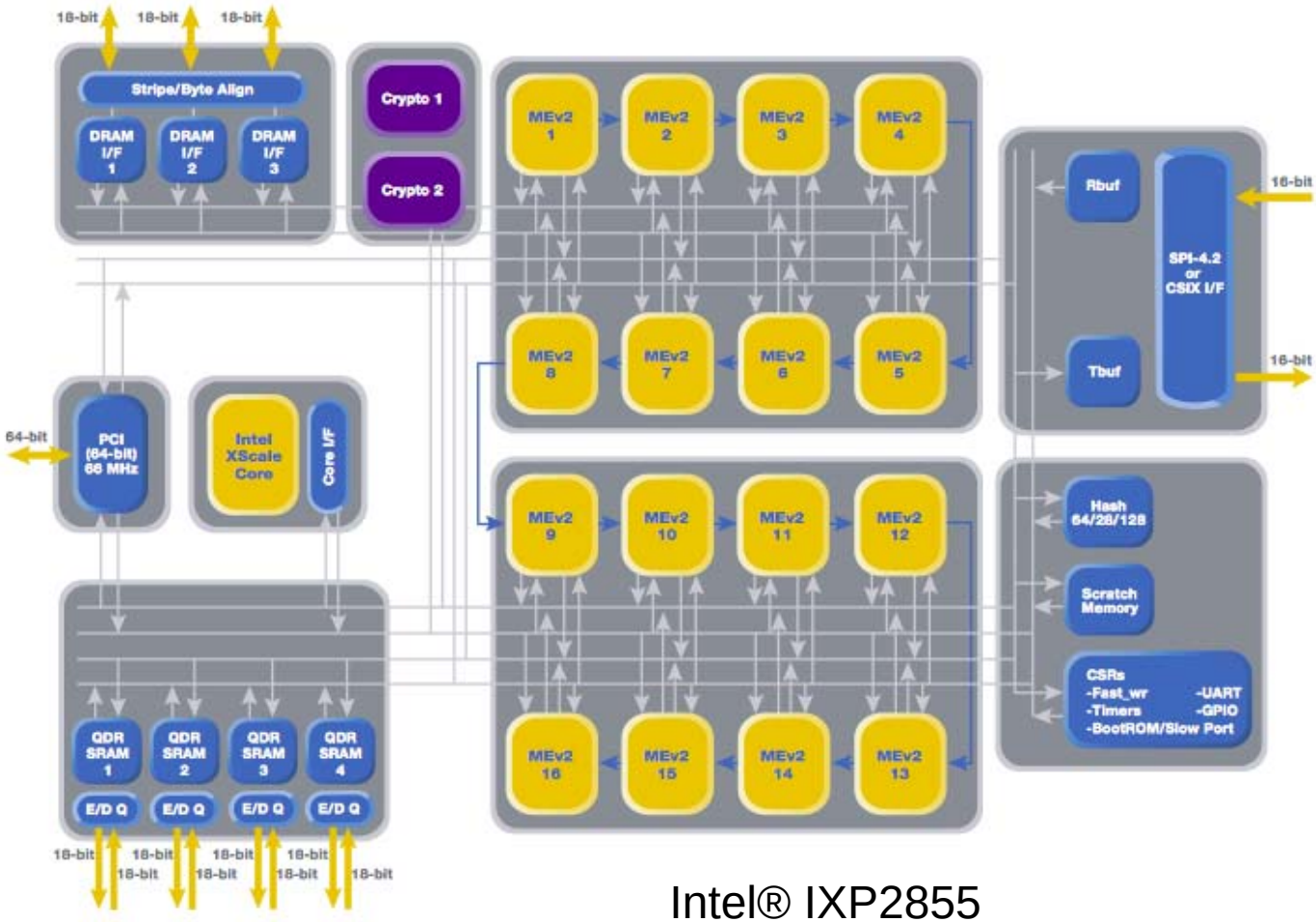
Link	Mbps	fps	ns/frame
T-1	1.5	2,941	340,000
T-3	45.0	90,909	11,000
OC-3	155.0	333,333	3,000
OC-12	622.0	1,219,512	820
GigE	*1,000.0*	*1,488,095*	*672*
OC-48	2,500.0	5,000,000	200
10 GigE	10,000.0	14,925,373	67
OC-192	9,500.0	19,697,843	51

How do we route?

How do we protect?

How do we correlate?

Programmable Network Processors

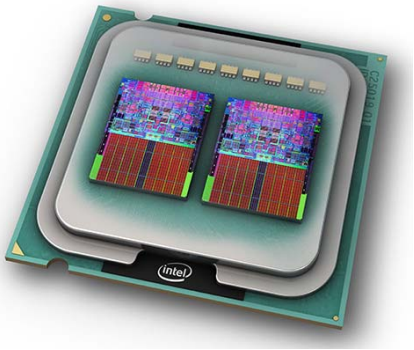


Intel® IXP2855

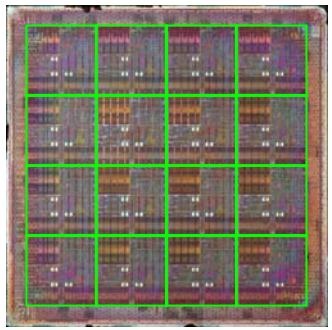
:(

Multicore Systems

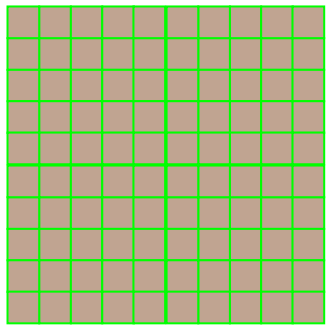
- GPP Multicore systems
  - Individual cores less powerful than UP
  - 10s-100s-1000s of cores
  - Full OS & Library Support
  - Asymmetric (Alpha)
  - Heterogeneous (AMD, Intel)



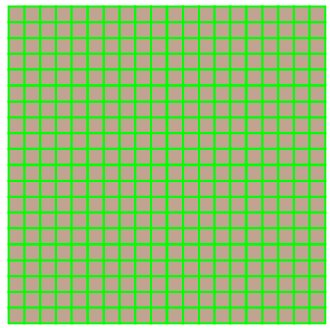
Intel (2x2-core)



MIT RAW (16-core)

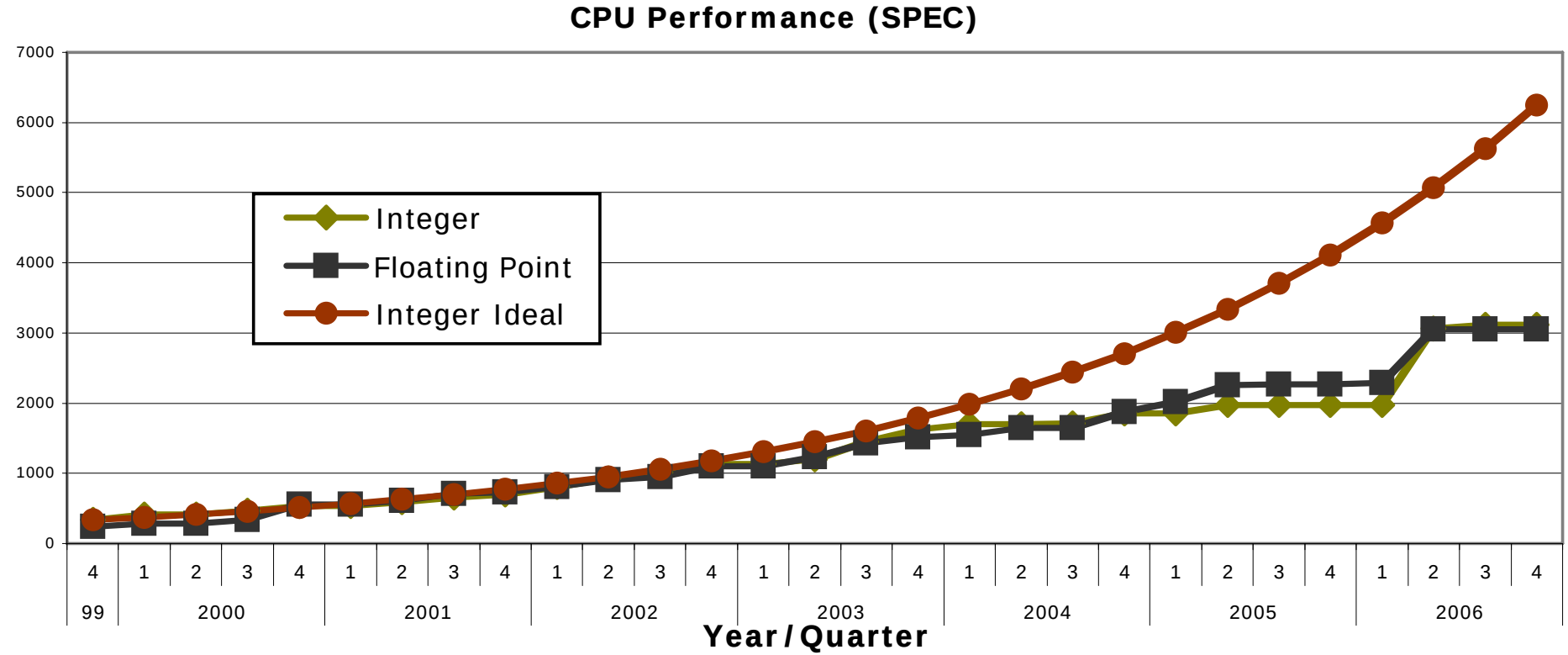


100-core



400-core

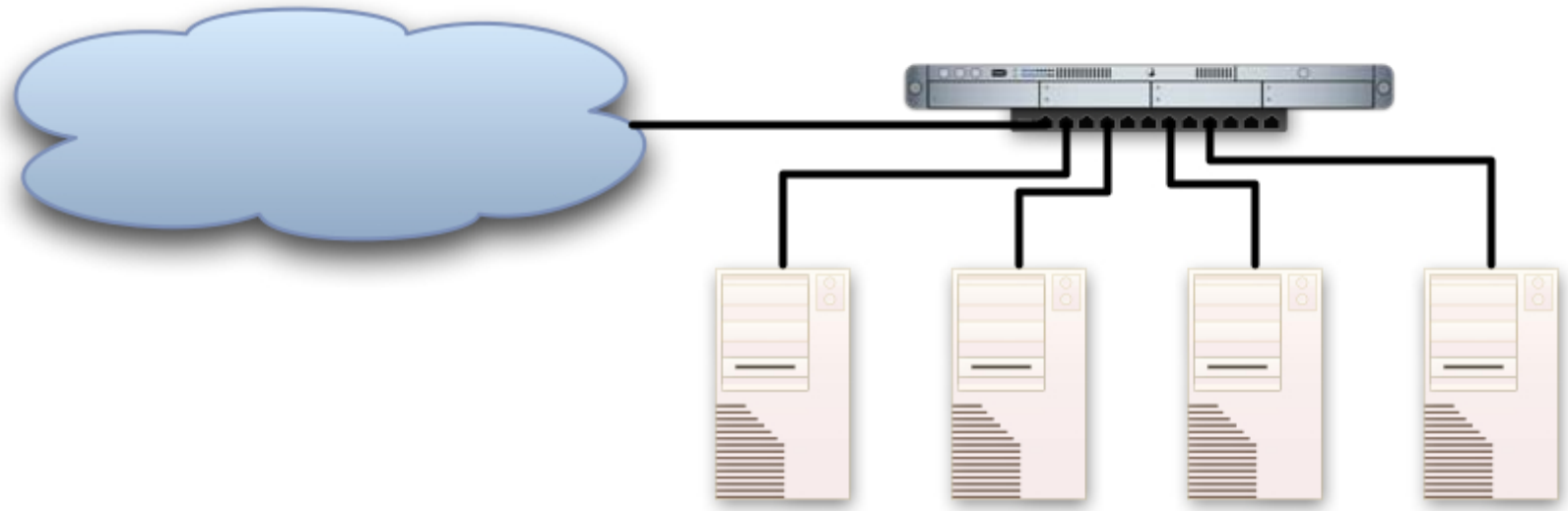
Moore's Corollary vs. Moore's Law



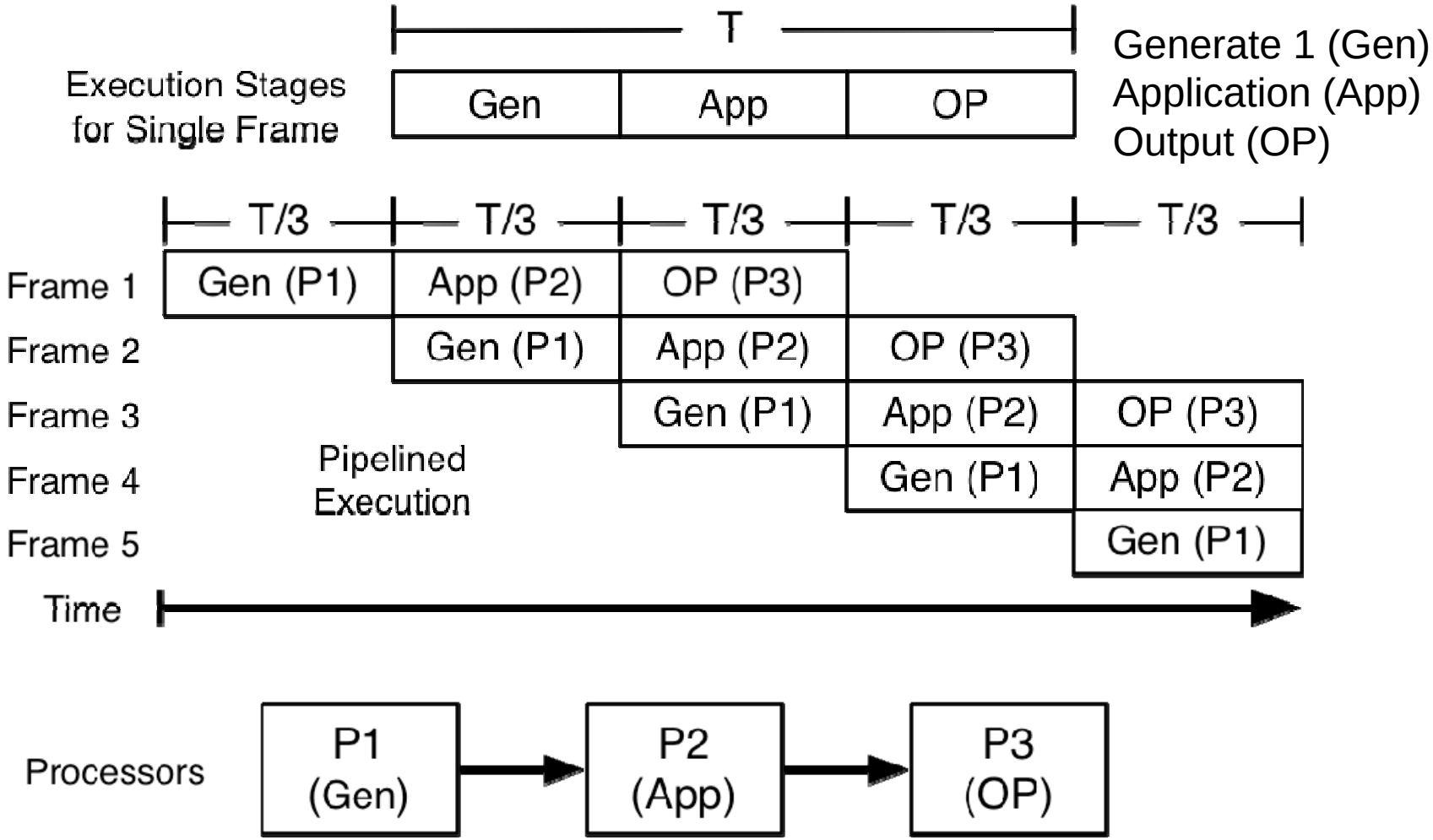
- SPEC Benchmark Suite Performance
  - Predicted vs. actual

Graph Courtesy Tipp Moseley

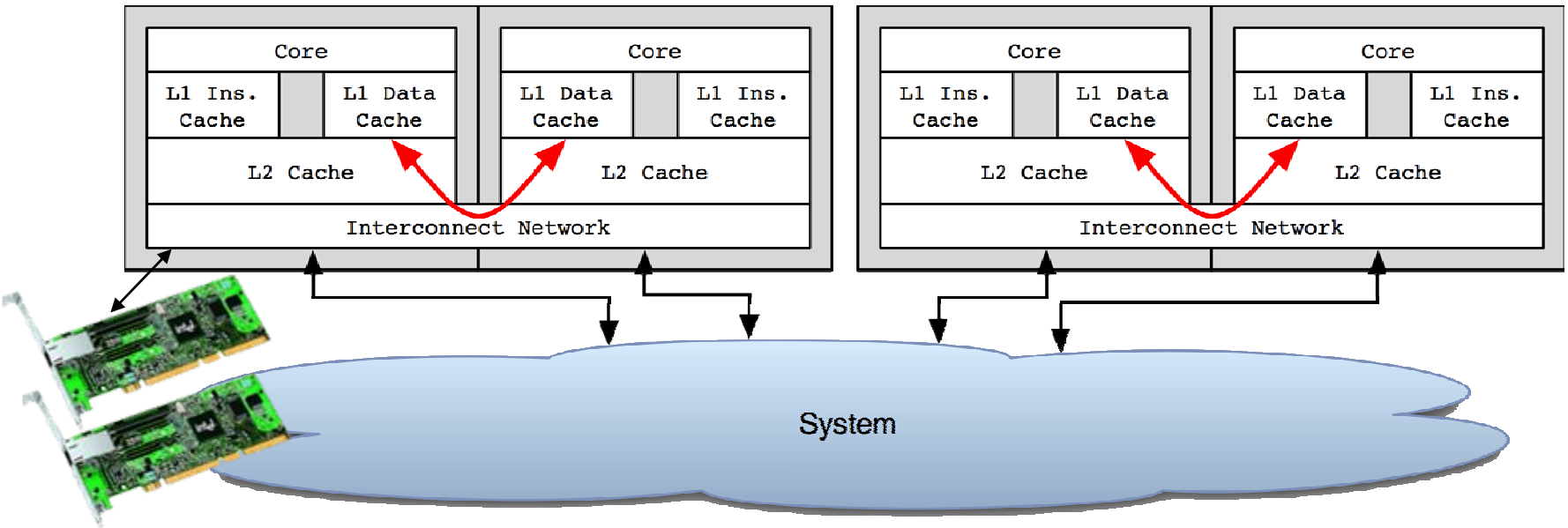
Soft Network Processing (Soft-NP)



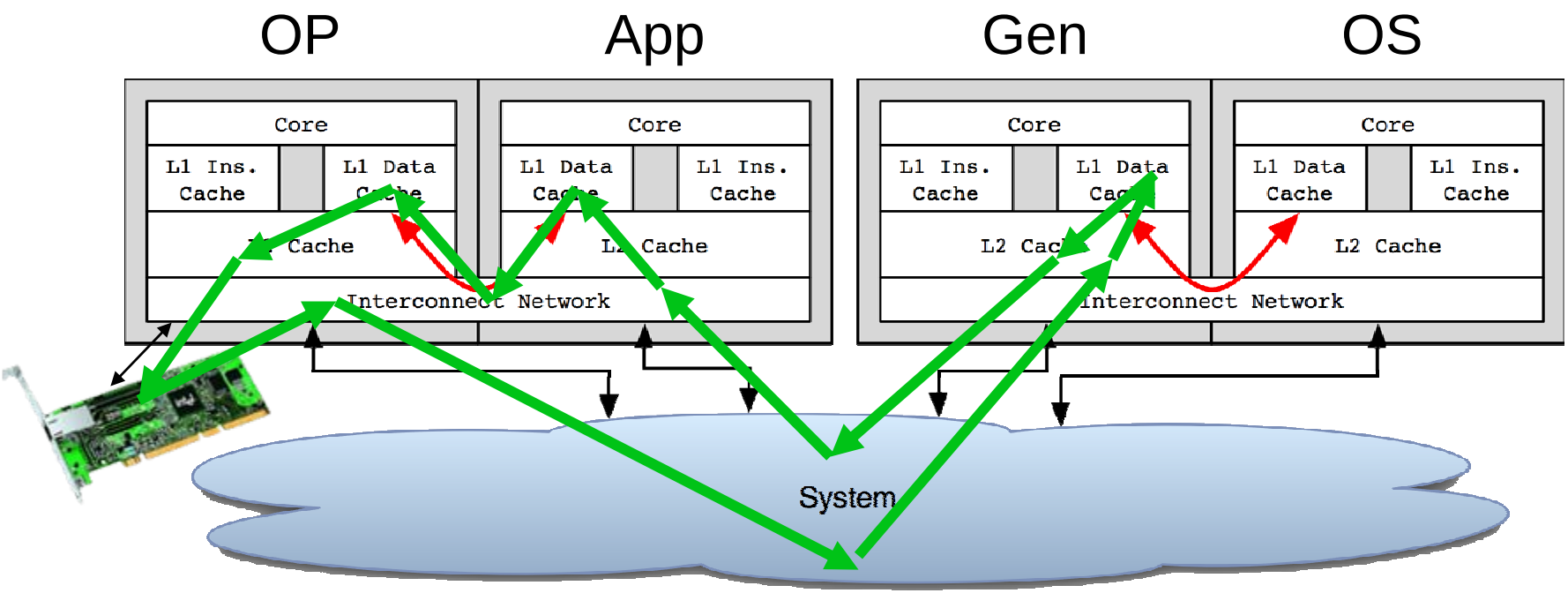
Soft-NP Technique Frame Generation



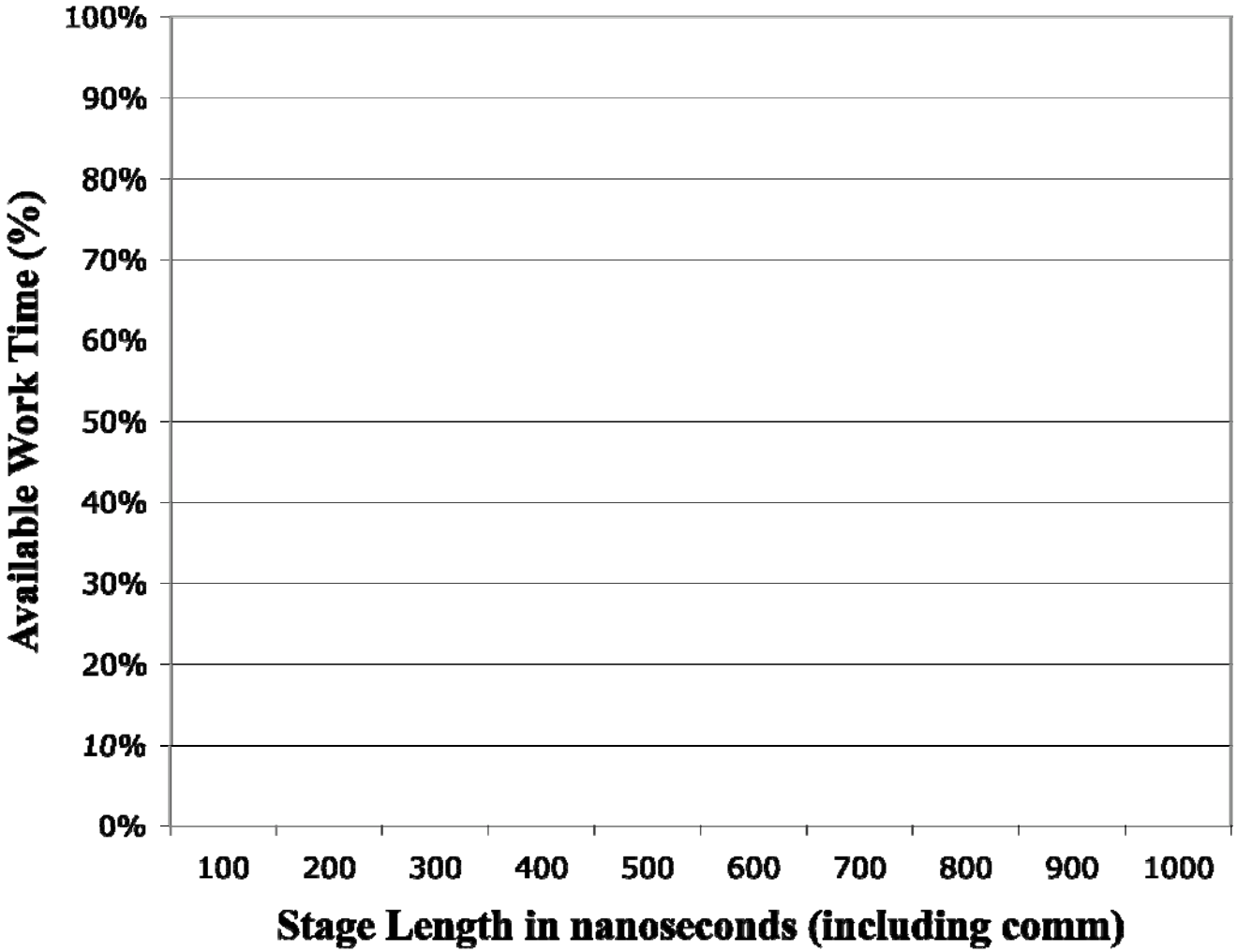
AMD Opteron System Overview



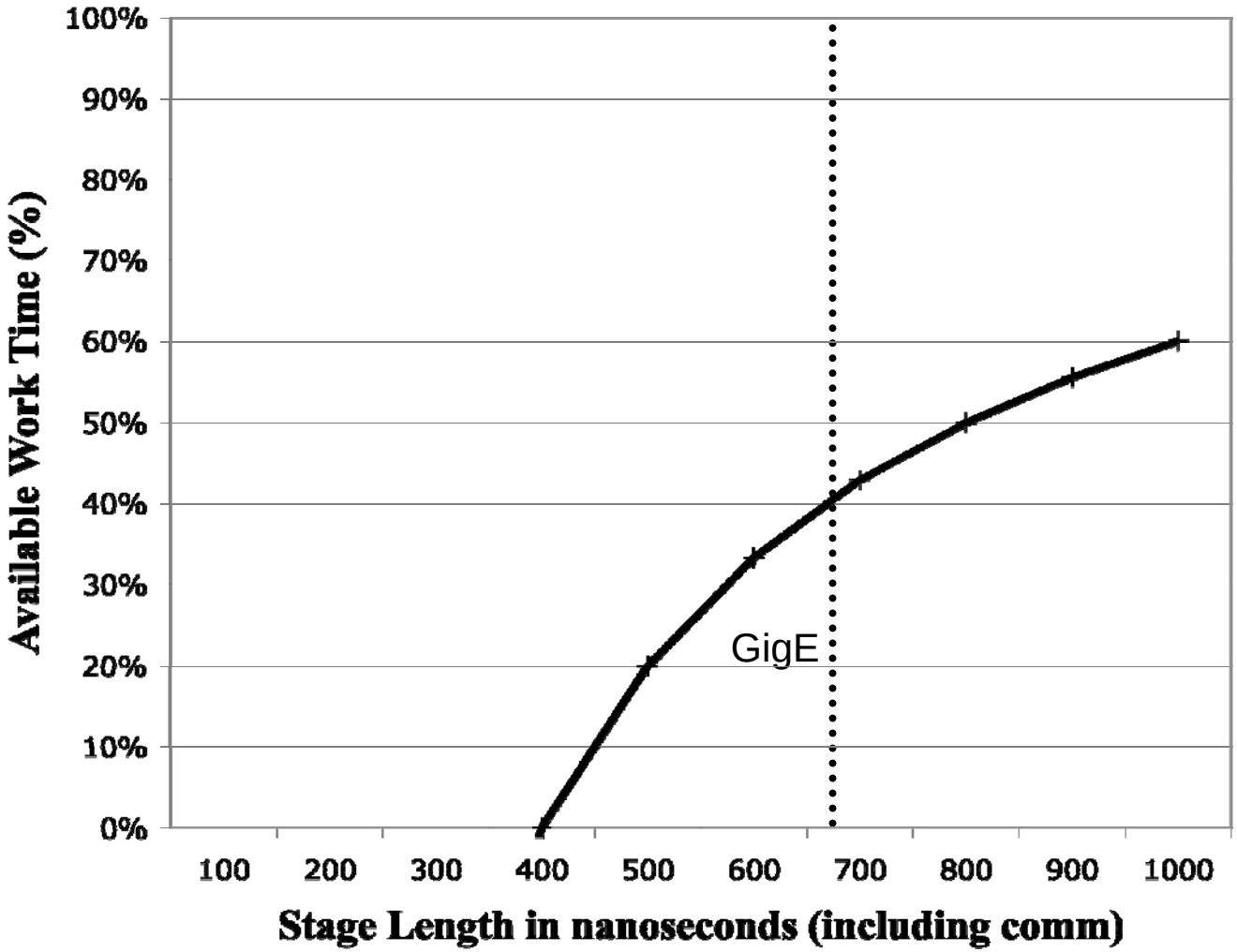
Data Flow
Frame Generation



Communication Overhead

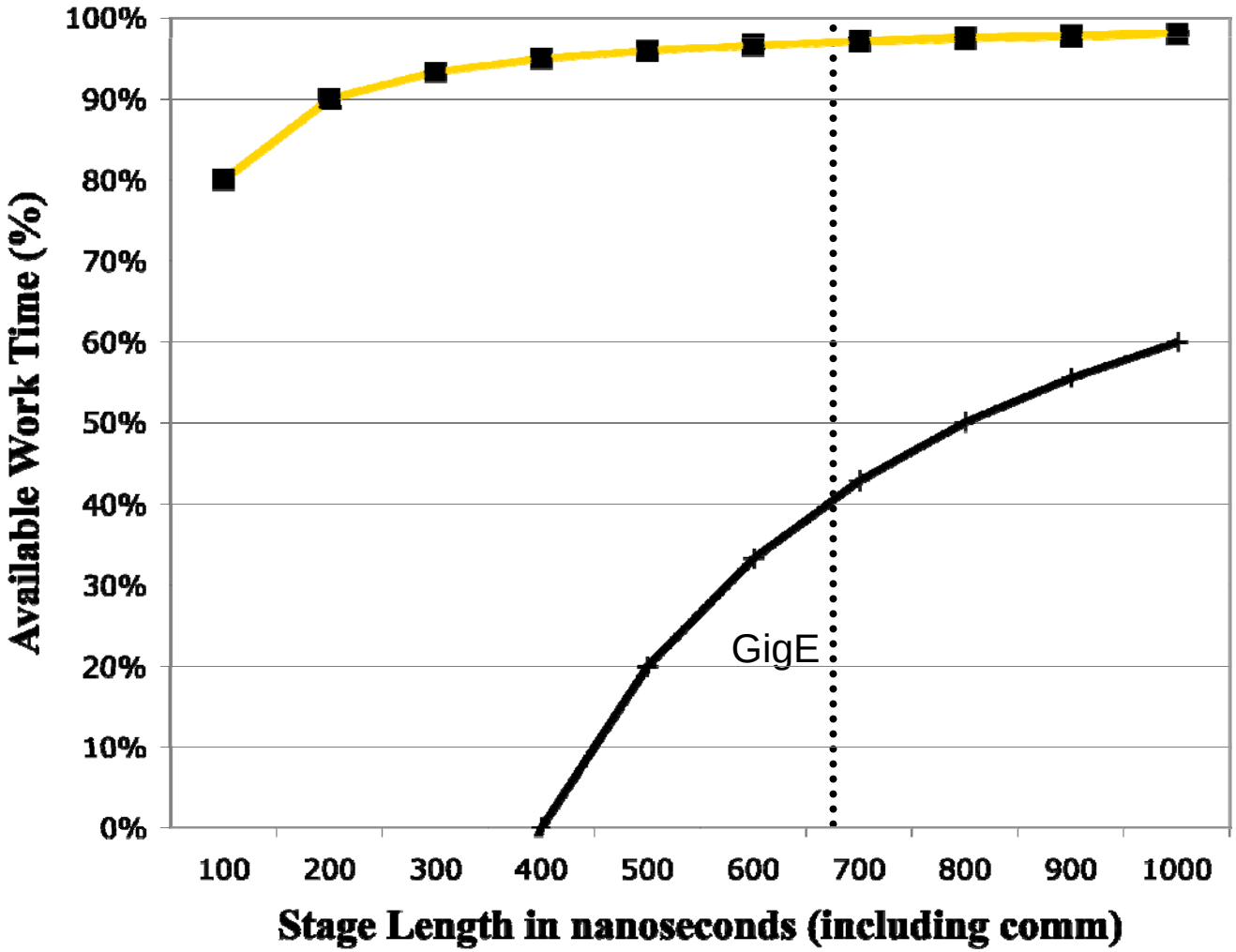


Communication Overhead



Locks ≈ 200ns

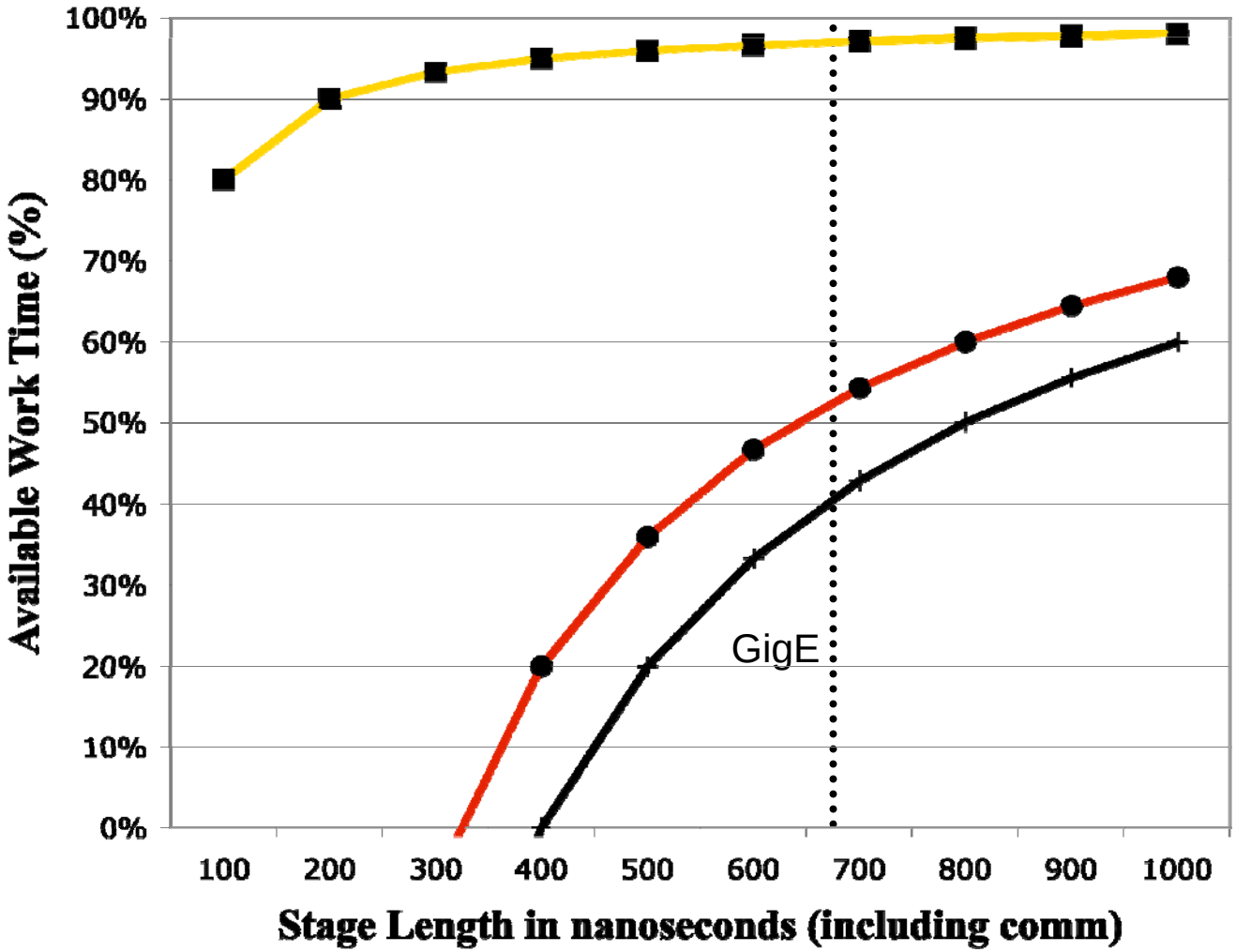
Communication Overhead



Hardware ≈ 10ns

Locks ≈ 200ns

Communication Overhead

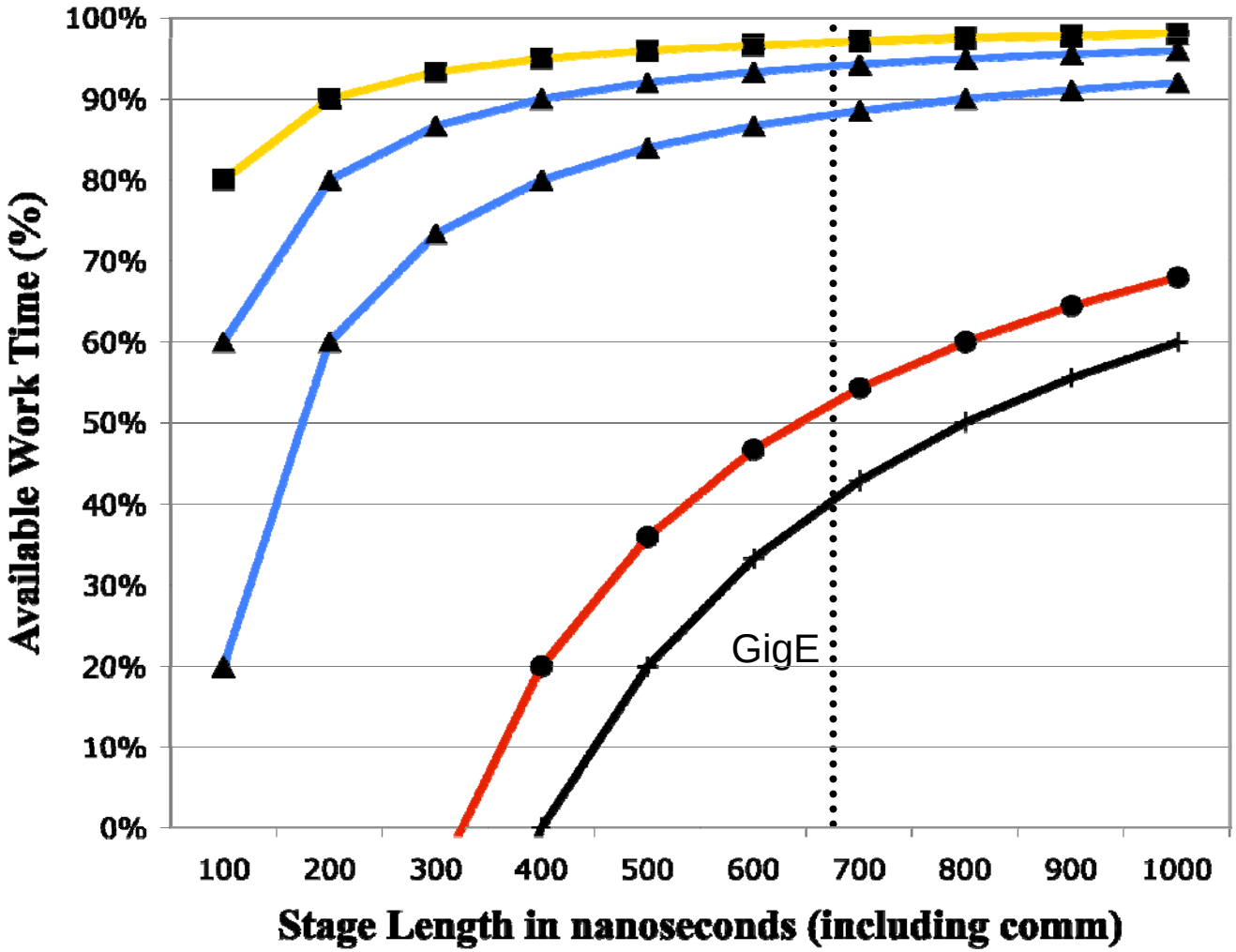


Hardware ≈ 10ns

Lamport ≈ 160ns

Locks ≈ 200ns

Communication Overhead



Hardware ≈ 10ns
FastForward ≈ 28ns

Lamport ≈ 160ns
Locks ≈ 200ns

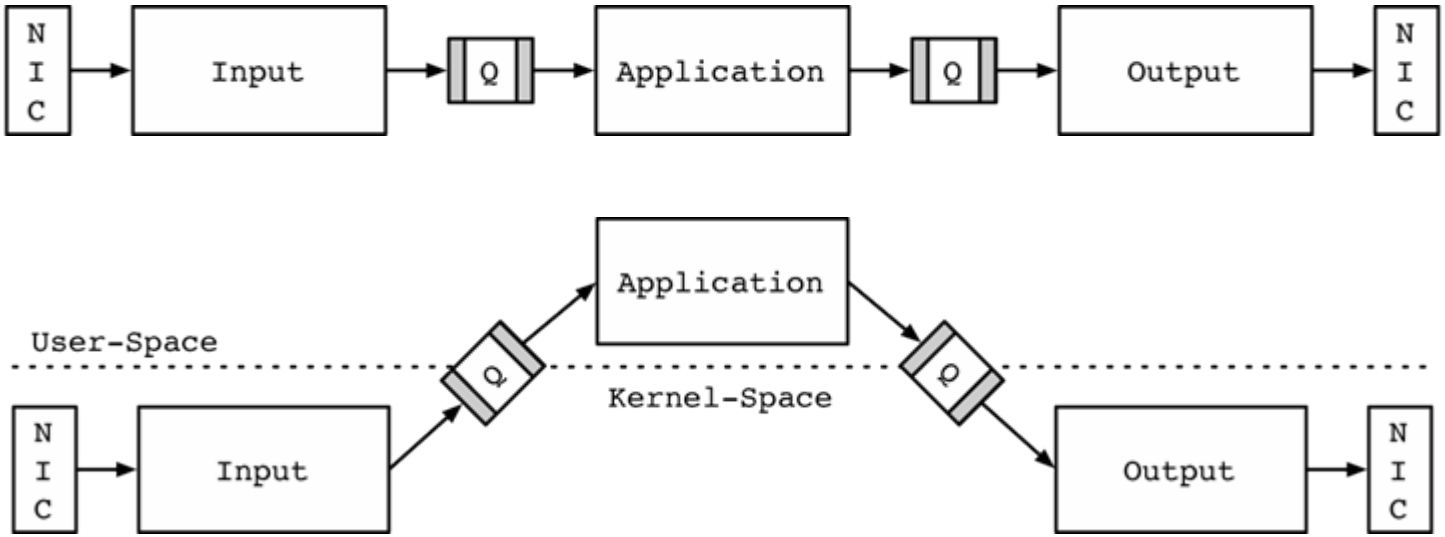
FastForward

```
enqueue(data) {
  lock(queue);
  if (NEXT(head) == tail) {
    unlock(queue);
    return EWOULDBLOCK;
  }
  buffer[head] = data;
  head = NEXT(head);
  unlock(queue);
  return 0;
}
```

```
enqueue_fastforward(data) {

  if (NULL != buffer[head]) {

    return EWOULDBLOCK;
  }
  buffer[head] = data;
  head = NEXT(head);

  return 0;
}
```

- Cache-optimized CLF queues
- Works with strong to weak consistency models
- Hides die-die communication
- Giacomoni, Moseley, and Vachharajani. “FastForward for Efficient Pipeline Parallelism: A Cache-Optimized Concurrent Lock-Free Queue.” To appear: Proceedings of the 13th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (**PPoPP**), February 2008

Frame Shared Memory (FShm)



- Pure software stack communicating via shared memory
  - Abstracted at the driver/NIC boundary
  - Cross-Domain modules (Kernel/Process, T/T, P/P, K/K)
  - Compatible with existing OS/library/language services
  - Can communicate with any device on the memory interconnect

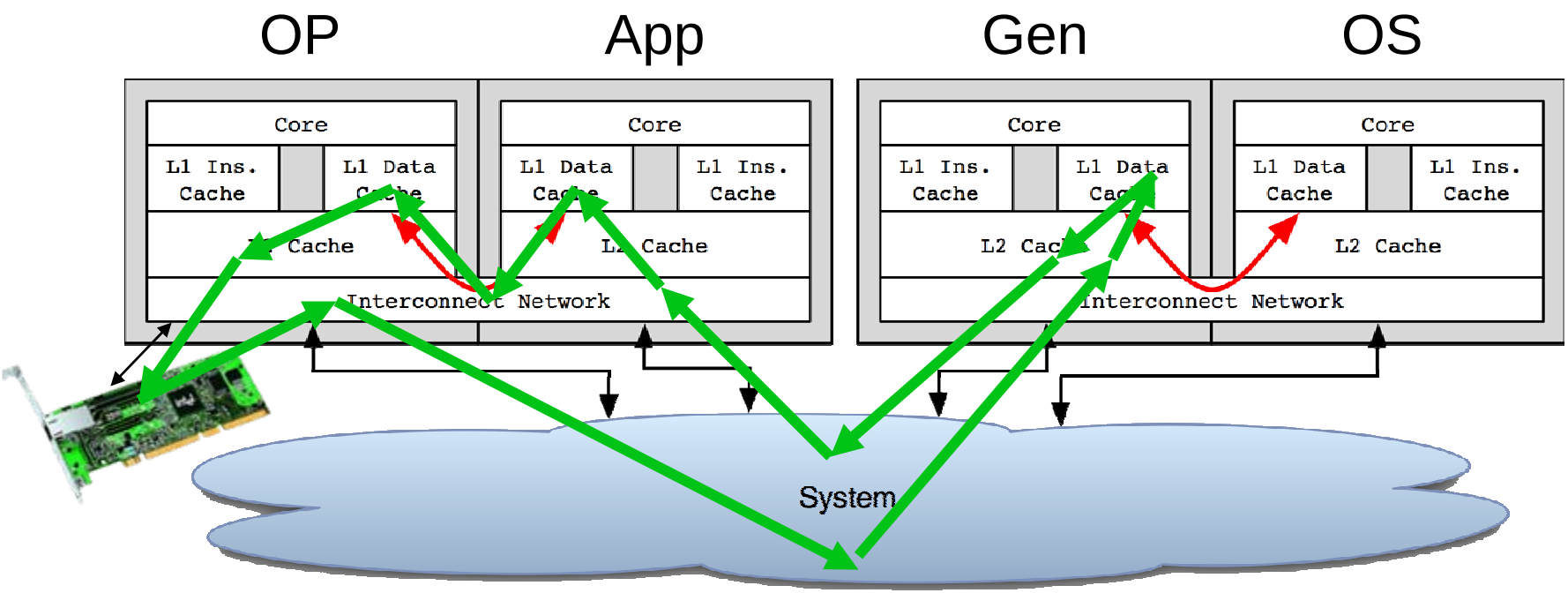
FShm Driver API

```
struct ifdirect {
  void            (*if_direct_tick) (void *softc);

  void            (*if_direct_attach) (struct ifnet *, void *);
  void            (*if_direct_detach) (struct ifnet *, void *);

  int             (*if_direct_tx) (void *softc, struct mbuf *txbuf);
  void            (*if_direct_tx_post) (void *softc);

  void            (*if_direct_tx_clean_pre) (void *softc);
  struct mbuf*    (*if_direct_tx_clean) (void *softc);
  void            (*if_direct_tx_clean_post) (void *softc);

  void            (*if_direct_rx_pre) (void *softc);
  struct mbuf*    (*if_direct_rx) (void *, struct mbuf *new_rxbuf);
  void            (*if_direct_rx_post) (void *softc);
};
```

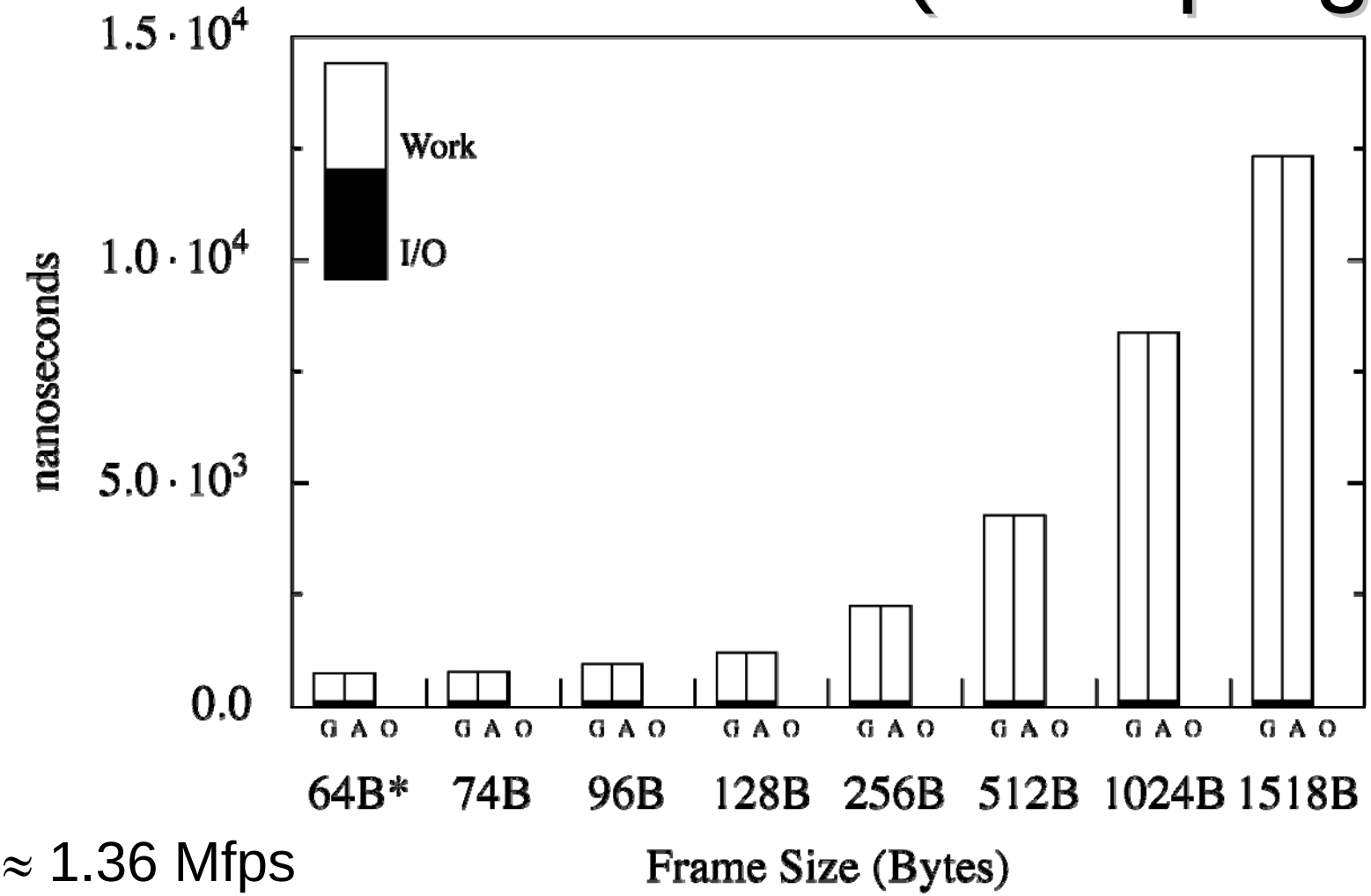
FShm Evaluation Methodology

- AMD Opteron 2.0 GHz
  - Dual-Processor & Dual-Core
- Compute average time per call
  - TSC

Frame Generation Data Flow

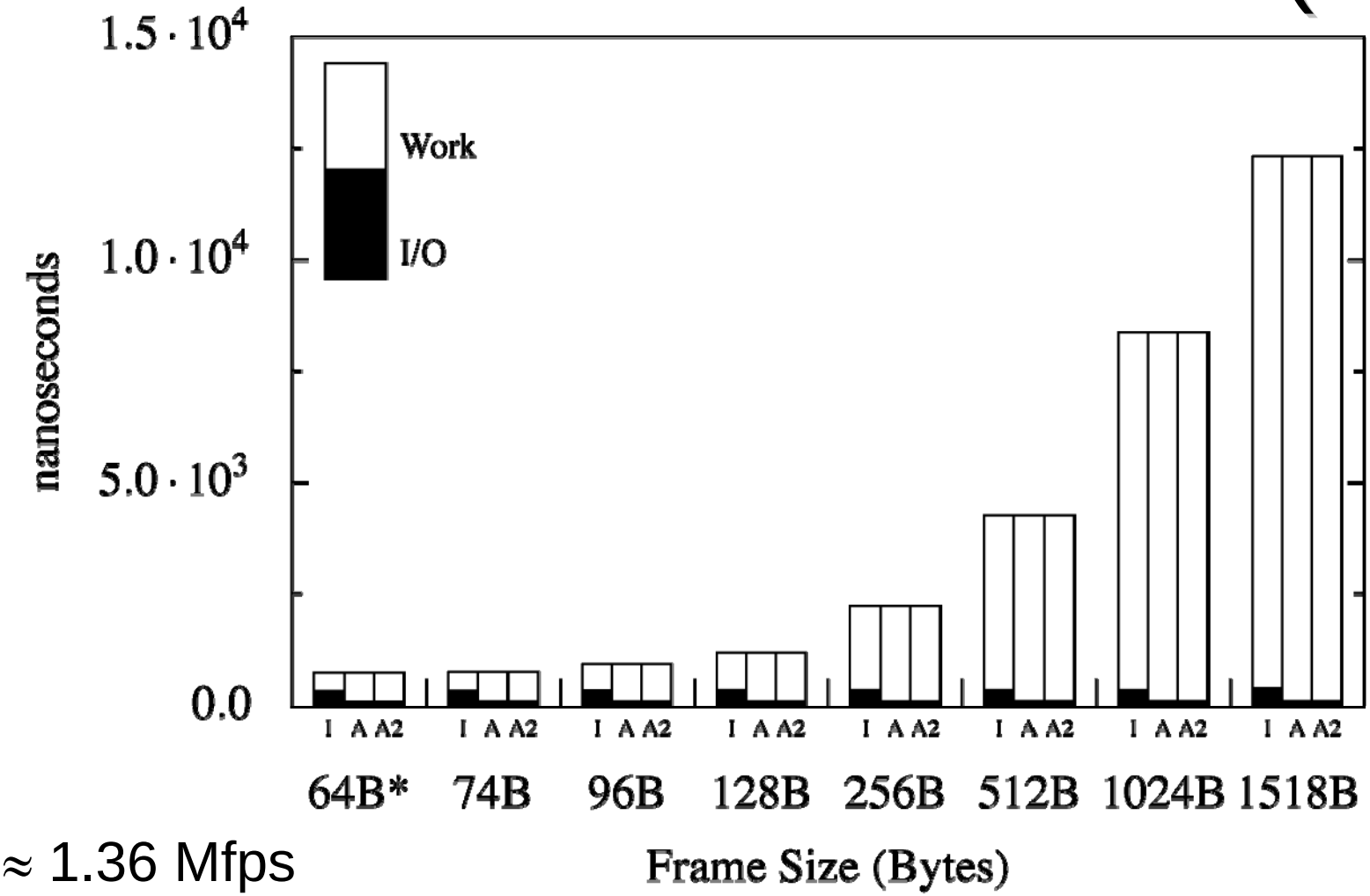


FShm Generate (linux pktgen)



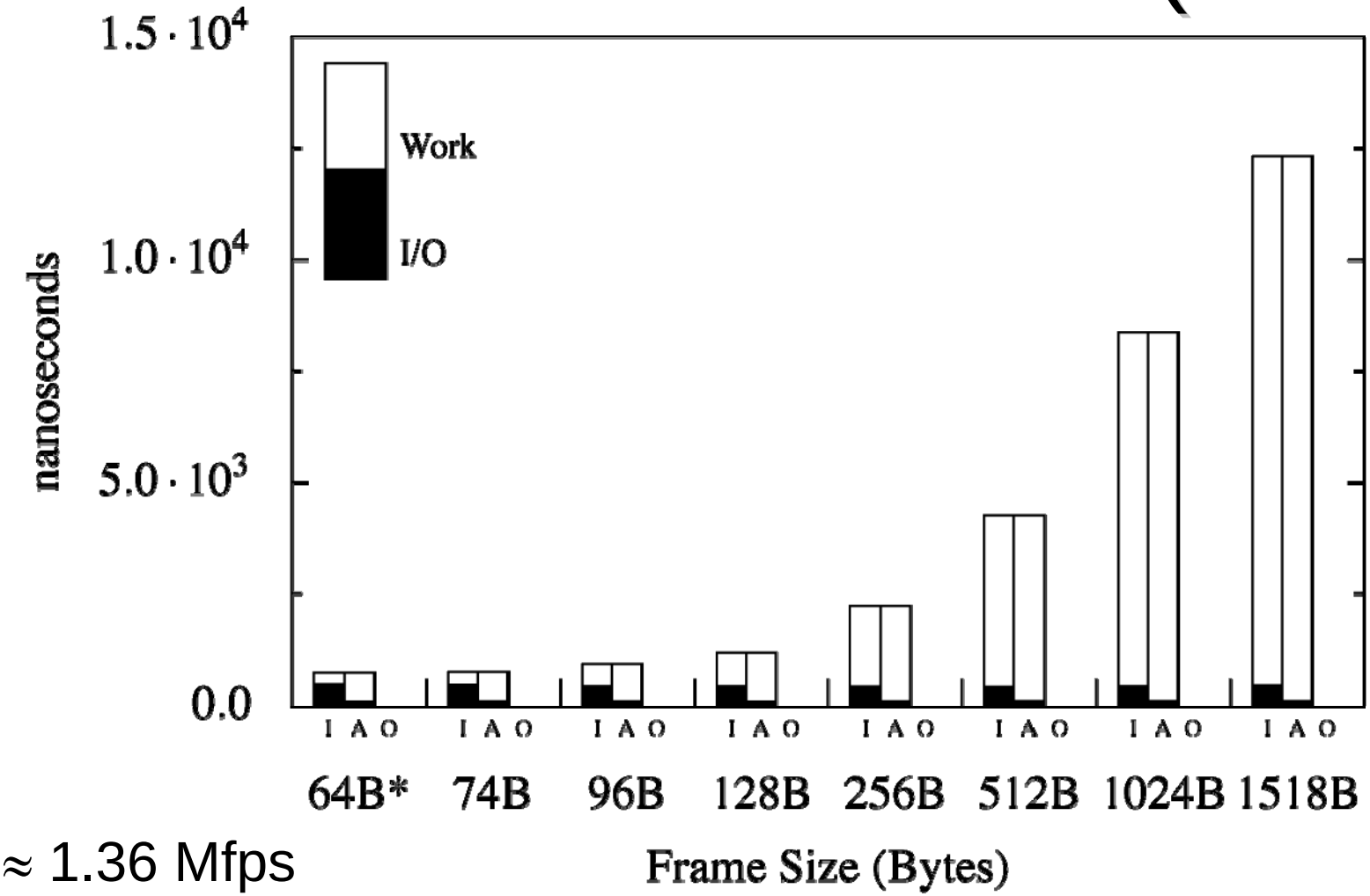
64B* ≈ 1.36 Mfps

FShm Capture (IDS)



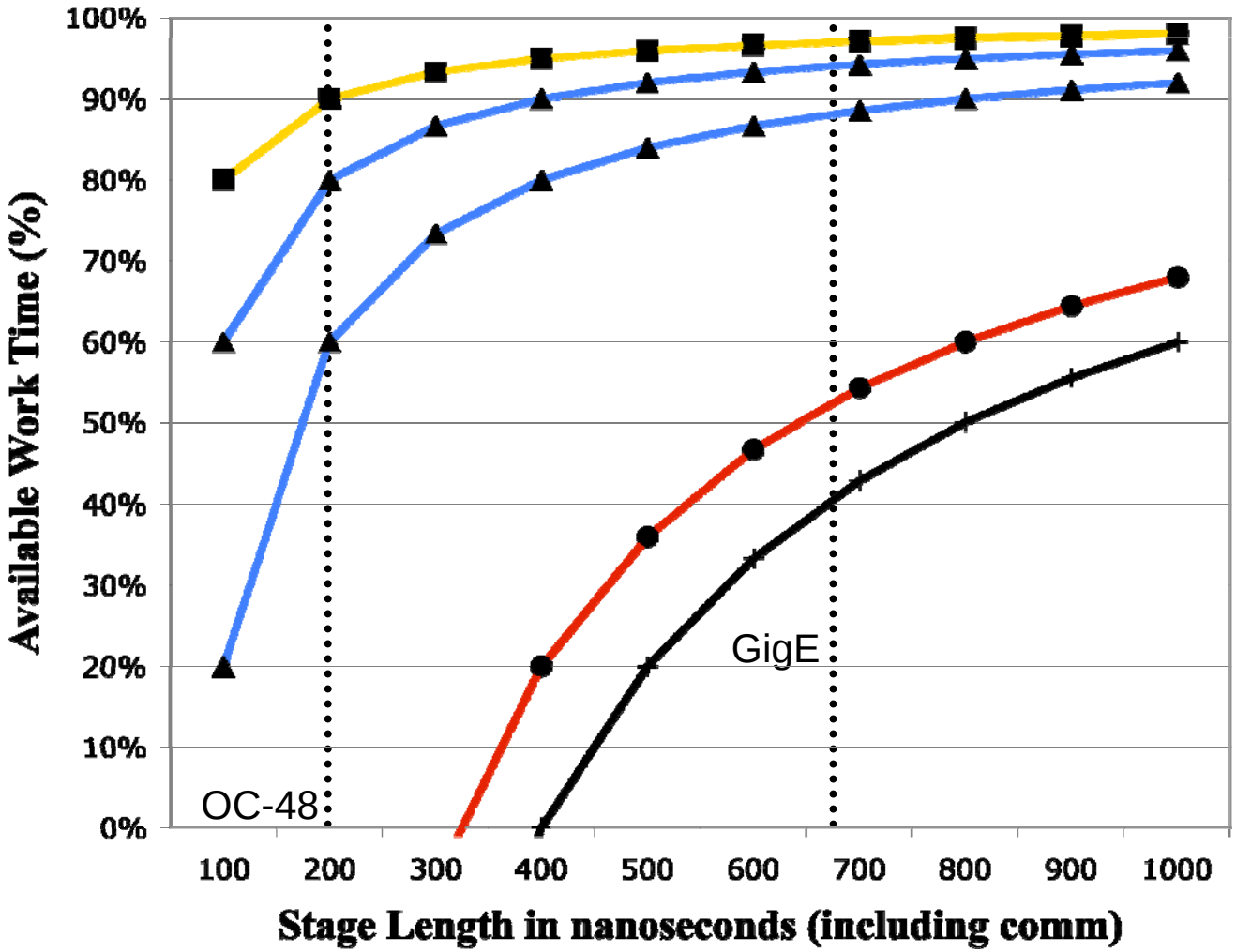
64B* ≈ 1.36 Mfps

FShm Forward (Bridge)



64B* ≈ 1.36 Mfps

FShm's Future



Hardware ≈ 10ns
FastForward ≈ 28ns

Lamport ≈ 160ns
Locks ≈ 200ns

Questions?

john.giacomoni@colorado.edu