

OpenMP Implementation on Advanced Cluster Systems

Jie Cai

Dr. Peter Strazdins

Supervisors: Dr. Alistair Rendell

Dr. Eric McCreath

29th May 2008

Outline

- Introduction
 - Advanced cluster systems
 - Programming models on clusters
- Cluster-enabled OpenMP systems
 - Current state-of-art
 - Performance evaluation and modeling
 - Pros and Cons
- Optimizing Cluster-enabled OpenMP
 - high-performance interconnects with RDMA
 - scheduler for heterogeneous cluster

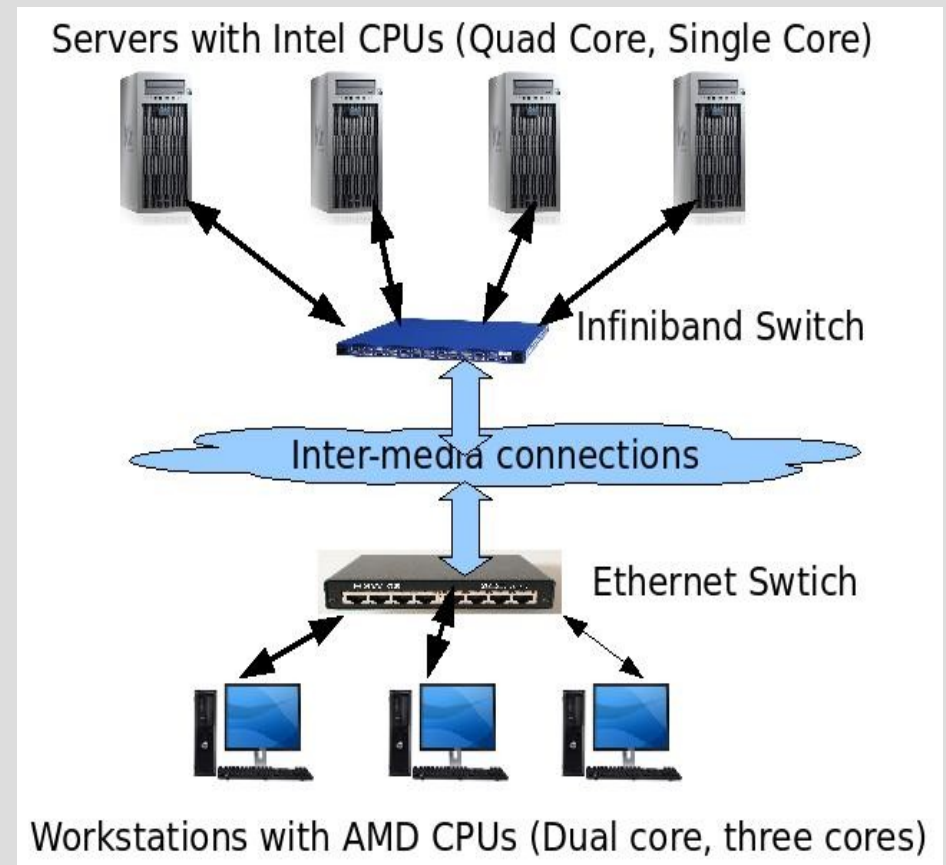
Cluster Systems?

- A cluster is a group of loosely coupled computers that work together to solve a distributed computation.



Advanced Cluster Systems

- High-performance interconnects
 - RDMA support
- Heterogeneous computing nodes
- Heterogeneous interconnects
 - InfiniBand
 - Giga-Ethernet
 - Myrinet
 -

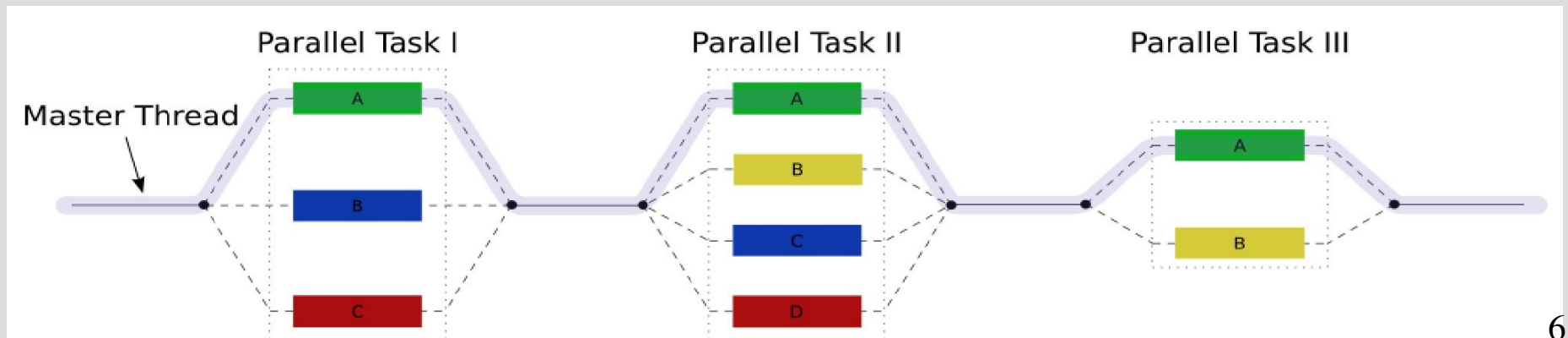


Programming Models on Clusters

- Message Passing Interface (MPI)
 - Dominant programming model on clusters
 - “Assembly language of parallel programming”
- Cluster-enabled OpenMP (OMP) systems

What is OpenMP?

- Standard multi-processing programming model for shared memory architecture
 - consists of a set of compiler directives, library routines, and environment variables
 - is supported by most C/Fortran compilers, e.g. `icc/ifort`, `gcc/g77`..
 - fork-join parallelism approach



Why OpenMP?

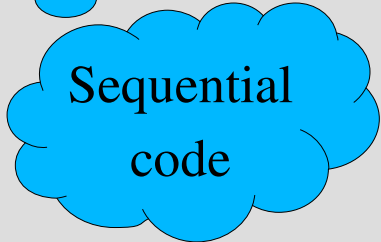
- Easier programming model

OpenMP model “test.c”

```
#include <omp.h>
```

```
#pragma parallel for default(shared) reduction(+:s)
```

```
for ( i = 0; i < N; i++ )  
    s += a[i];
```



Sequential
code

MPI model “test.c”

```
#include <mpi.h>
```

```
MPI_Scatter(sendbuf, (N/nprocs), MPI_INT,  
           rbuf, (N/nprocs), MPI_INT, 0,  
           comm);
```

```
for ( i = 0; i < N/nprocs; i++ )  
    ss += rbuf[i];
```

```
MPI_Reduce(&ss, &s, 1, MPI_INT,  
          MPI_SUM, 0, comm);
```

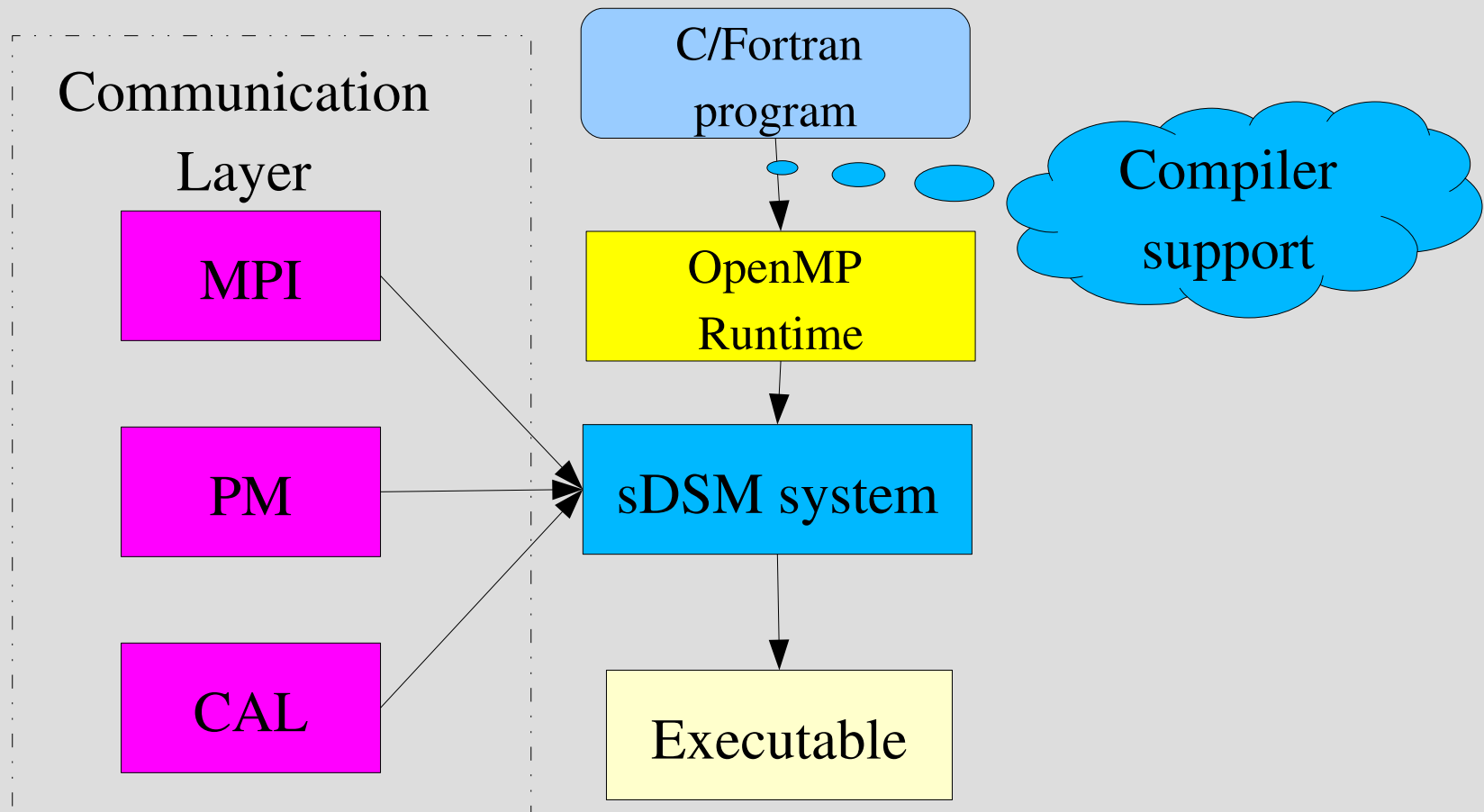
Outline

- Introduction
 - Advanced cluster systems
 - Programming models on clusters
- **Cluster-enabled OpenMP systems**
 - **Current state-of-art**
 - **Performance evaluation and modeling**
 - **Pros and Cons**
- **Optimizing Cluster-enabled OpenMP**
 - high-performance interconnects with RDMA
 - scheduler for heterogeneous cluster

What is a Cluster-enabled OpenMP System?

- A software system to extend OpenMP program on cluster.
- Typically based on software Distributed Shared Memory (sDSM) systems, a virtual shared memory environment.

Implementing Cluster OMP



Current and Past sDSM Systems

- Currently active
 - Intel Cluster OpenMP (CLOMP)
 - Omni/SCLIB, Omni/Danui
- Past
 - TreadMarks
 - Omni/SCASH
 - ParADE

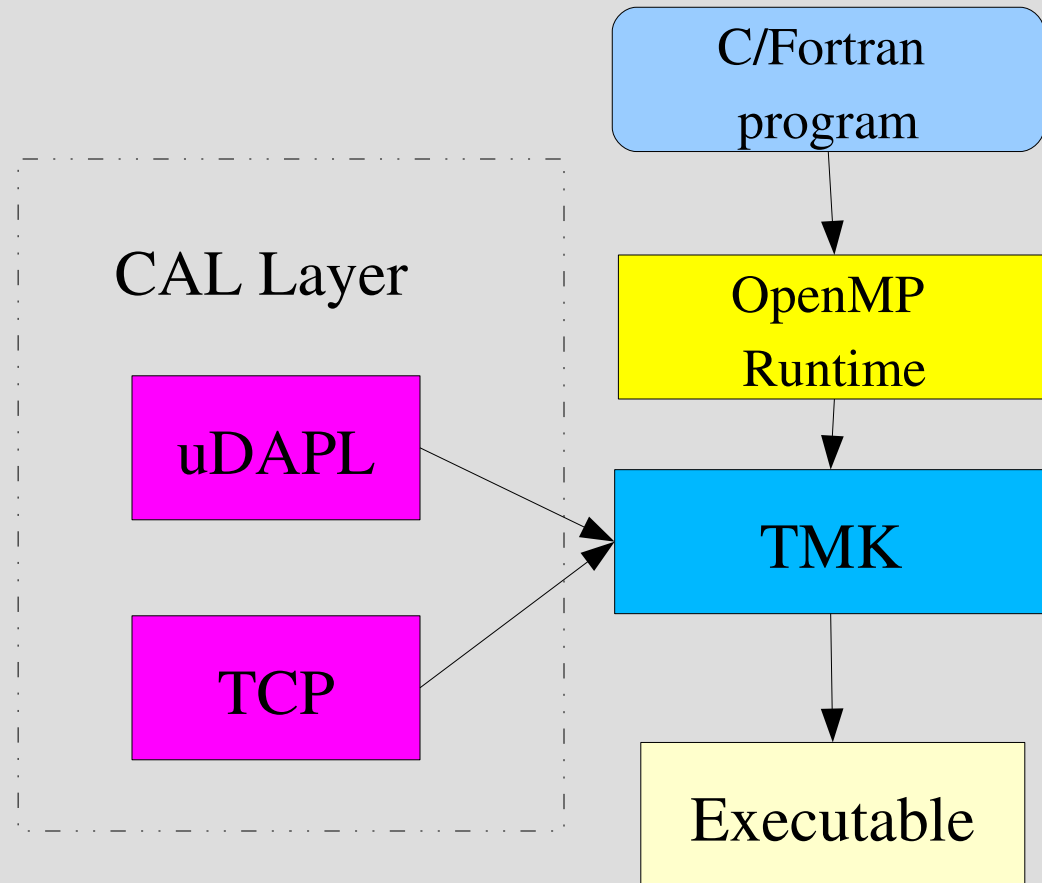
We will concentrate on current active sDSM systems.

Page-based sDSM Systems

- Memory is managed in fixed block of size, called page.
- Memory is kept consistent through detecting and servicing different type of page-faults.
- Memory consistency protocols are different for different sDSM systems.

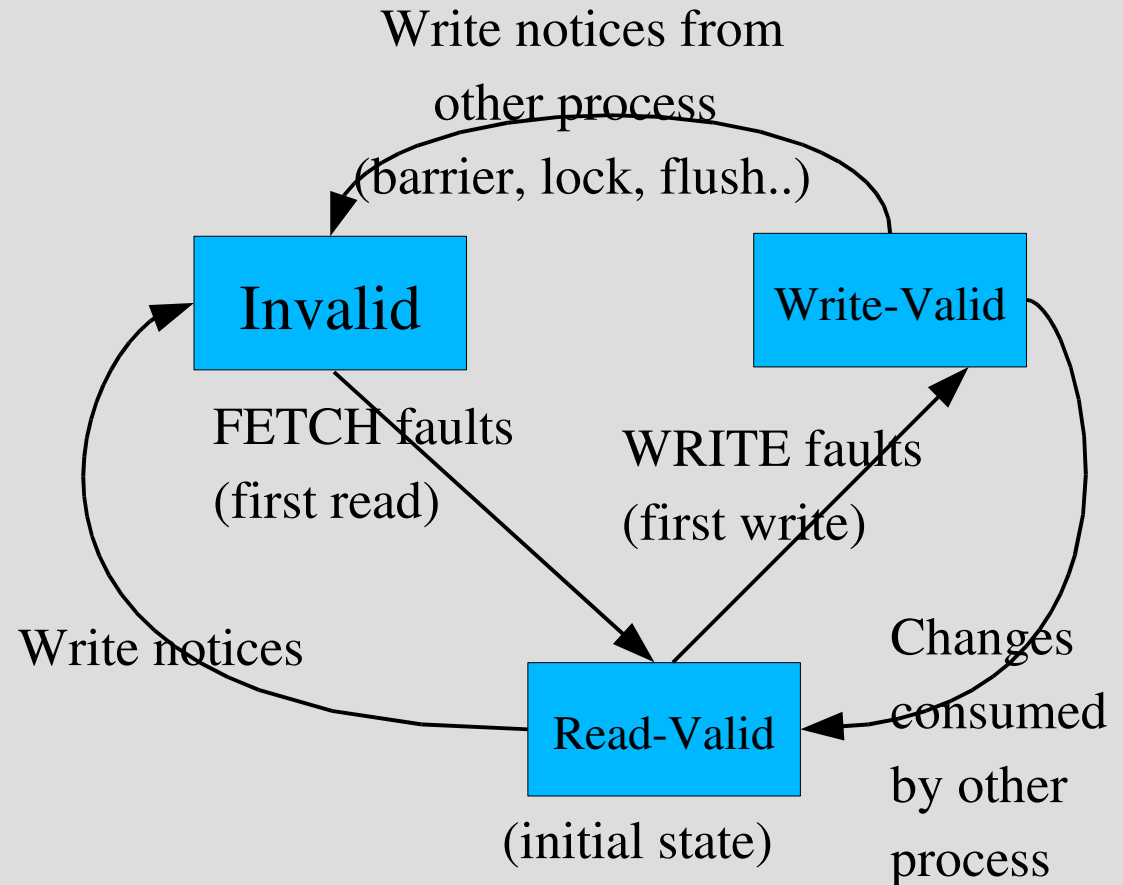
CLOMP

- Derived from TreadMarks, a home-less sDSM system, and released by Intel at 2006.
- Supported by Intel C/C++ and Fortran compiler.
- Lazy Release Consistency model is used.



Memory Consistency -- CLOMP

- **Write fault** includes:
 - make a “twin” page of the page to be written,
 - and calculate “diffs”.
- **Fetch fault** includes:
 - send requests,
 - “diffs” transfer,
 - apply “diffs”,
 - and send response.



CAL Layer of CLOMP

- Employs two communication modules
 - uDAPL: user Direct Access Programming Library provides a network, architecture and operating system independent communication layer.
 - Enables RDMA over high performance interconnects, e.g. InfiniBand, Myrinet.
 - TCP
 - Utilizes sockets to communication via Ethernet or other interconnections.

Experimental Setup

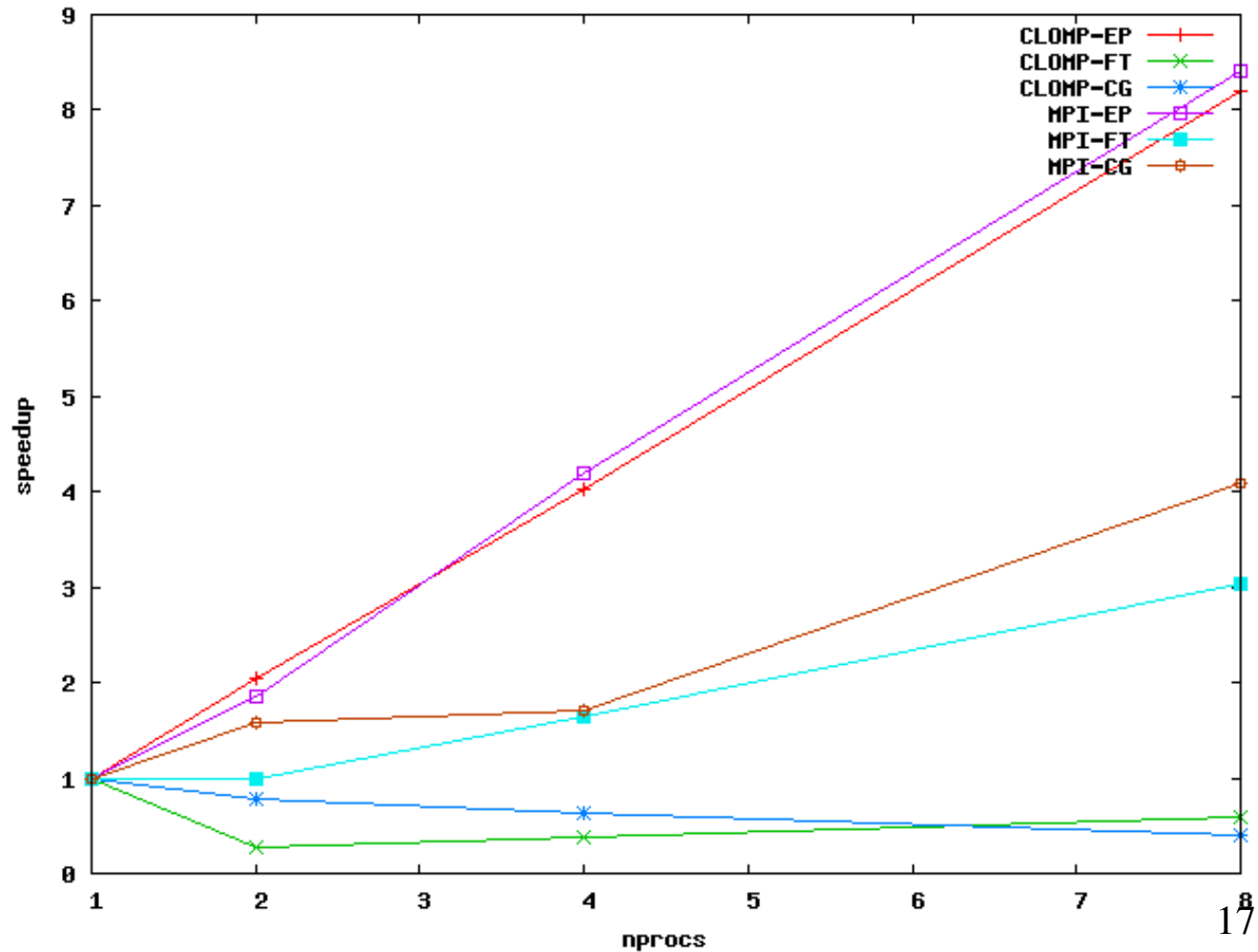
- Hardware
 - 8 nodes AMD Athlon dual core @ 2.2GHz
 - 4 GB memory
 - Giga-Ethernet interconnection
- Software
 - CLOMP 10.0 with icc
 - MPICH2 with icc
- NAS Parallel Benchmark suite

Performance Evaluation

Results (NPB class A)

- MPI speedup
 - EP: 8.4
 - FT: 3.0
 - CG: 4.1
- CLOMP speedup
 - EP: 8.2

CLOMP 1 thread elapsed time is used as sequential time for each case.

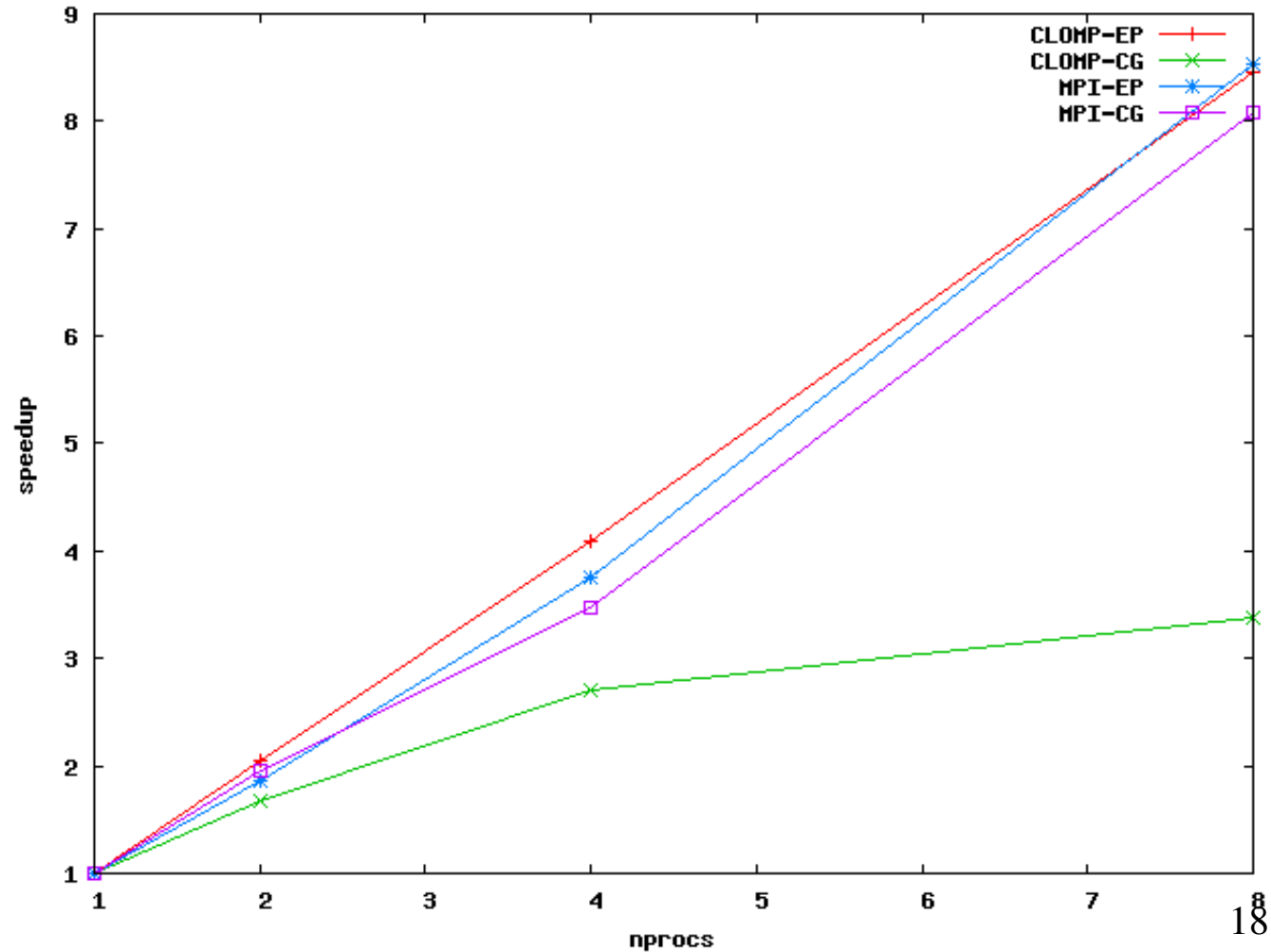


Performance Evaluation

Results (NPB class C)

- MPI speedup
 - EP: 8.5
 - CG: 8.09
- CLOMP speedup
 - EP: 8.5
 - CG: 3.38

CLOMP 1 thread elapsed time is used as sequential time for each case.



Evaluation Results Discussion

- The performance of CLOMP (and SCLIB) is not satisfactory for some benchmarks.
 - may be better for large problem size.
- Major overhead is memory consistency overhead:
 - “diff” fetching through network: FETCH fault
 - “twining” and “diffing” cost: WRITE fault

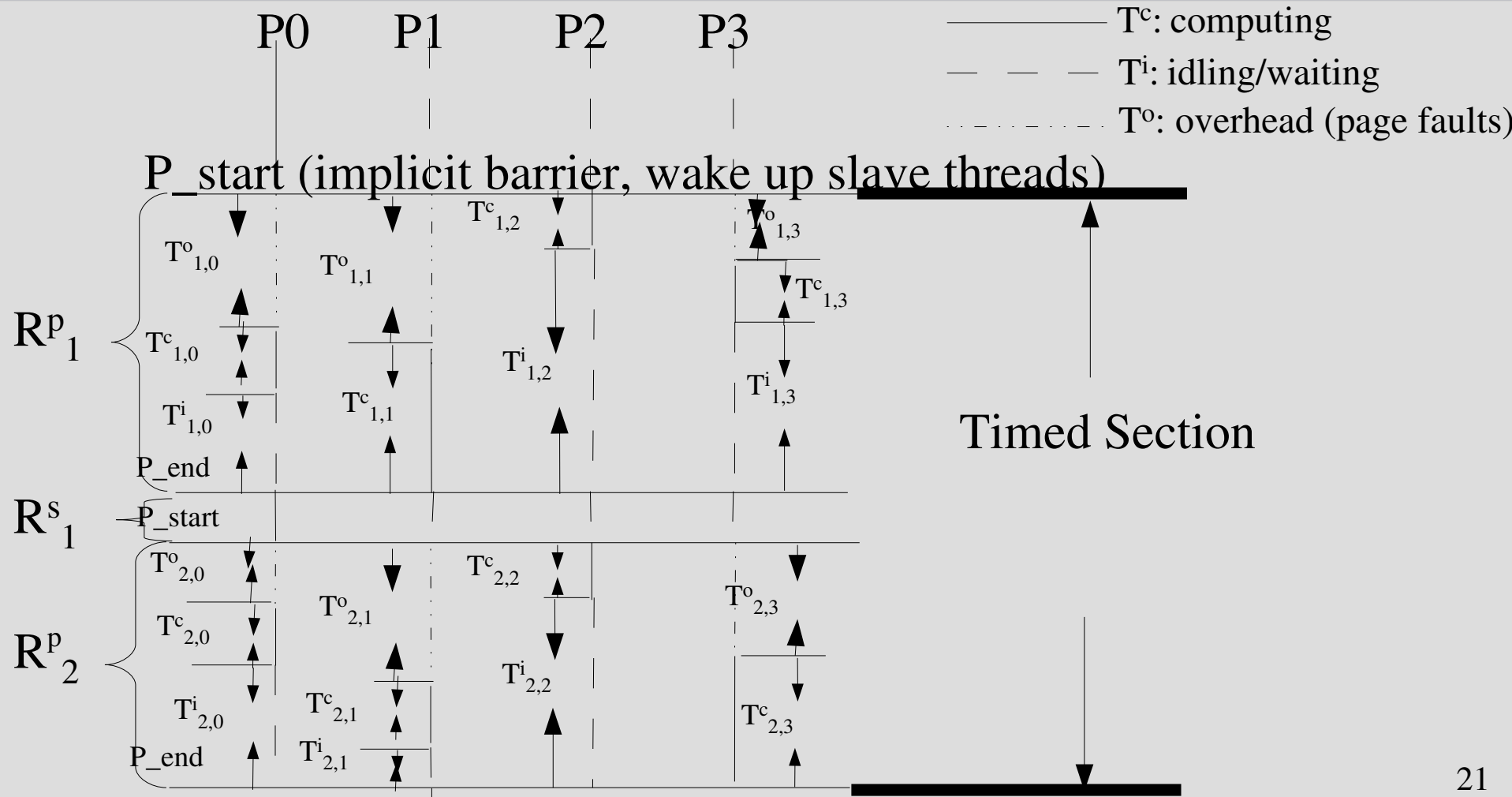
Performance Models

- SIGSEGV Driven Performance (SDP) model:
 - Rationalizing numbers and types of page faults to performance of sDSM systems.
 - For p processes, the model is:

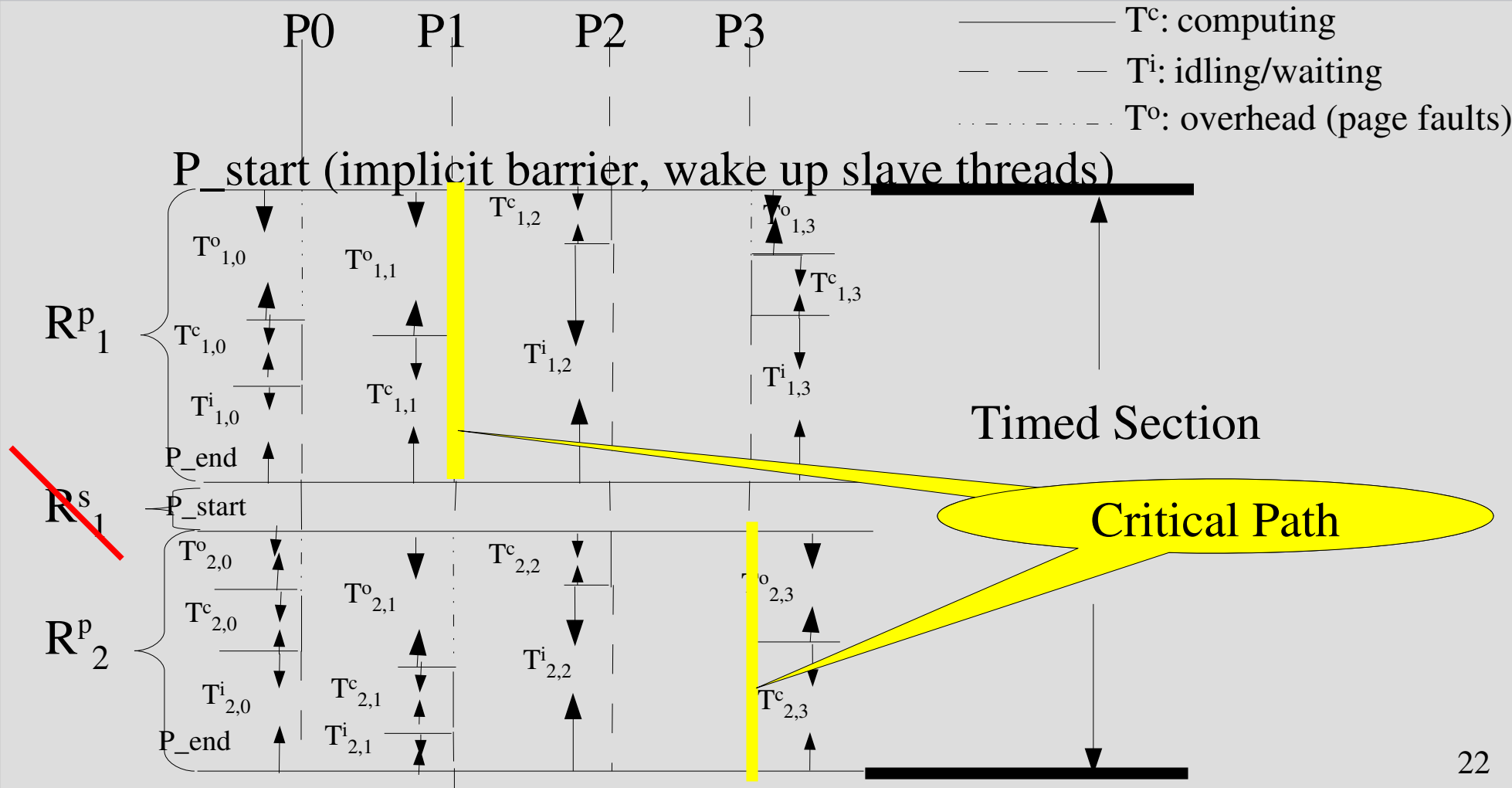
$$T_{est} = (T_{Tot} \div p) + \sum_{R_1^p}^{R_n^p} \text{Max}_{P_0}^{P_p} (N_{Wfaults} \times Cost_{Wfaults} + N_{Ffaults} \times Cost_{Ffaults})$$

- assume that sequential time is negligible,
- cost of fetch fault is constant,
- page faults can be overlapped,
- benchmarks are load-balanced.

SDP Model Illustration

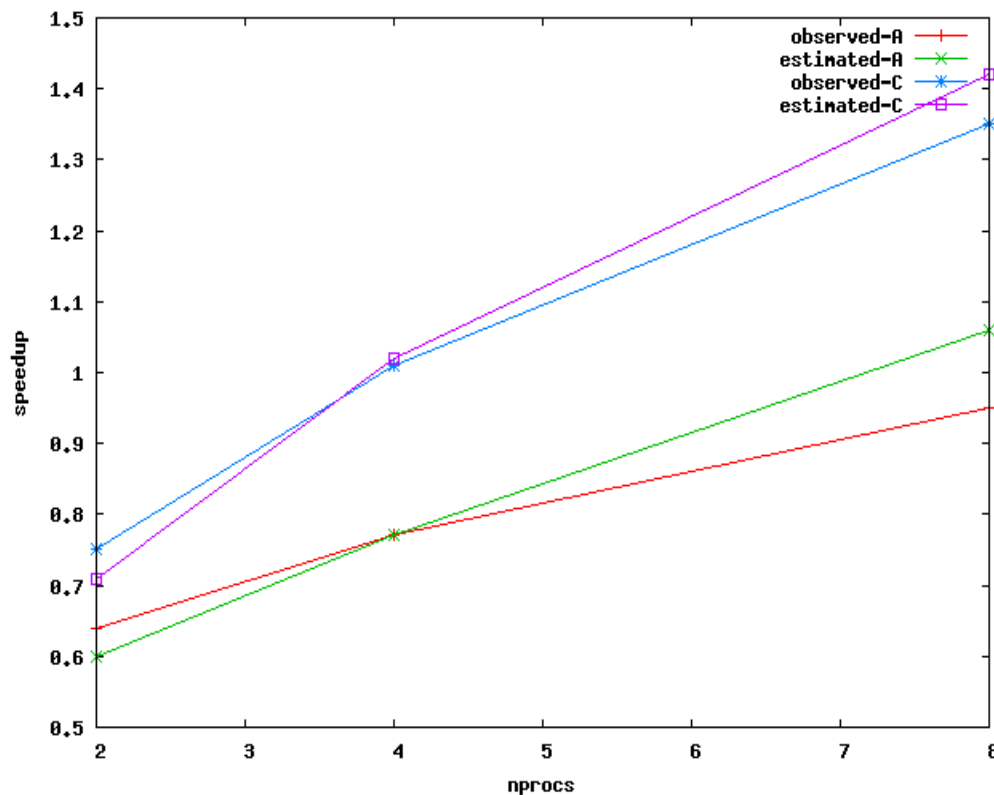


SDP Model Illustration (cont.)

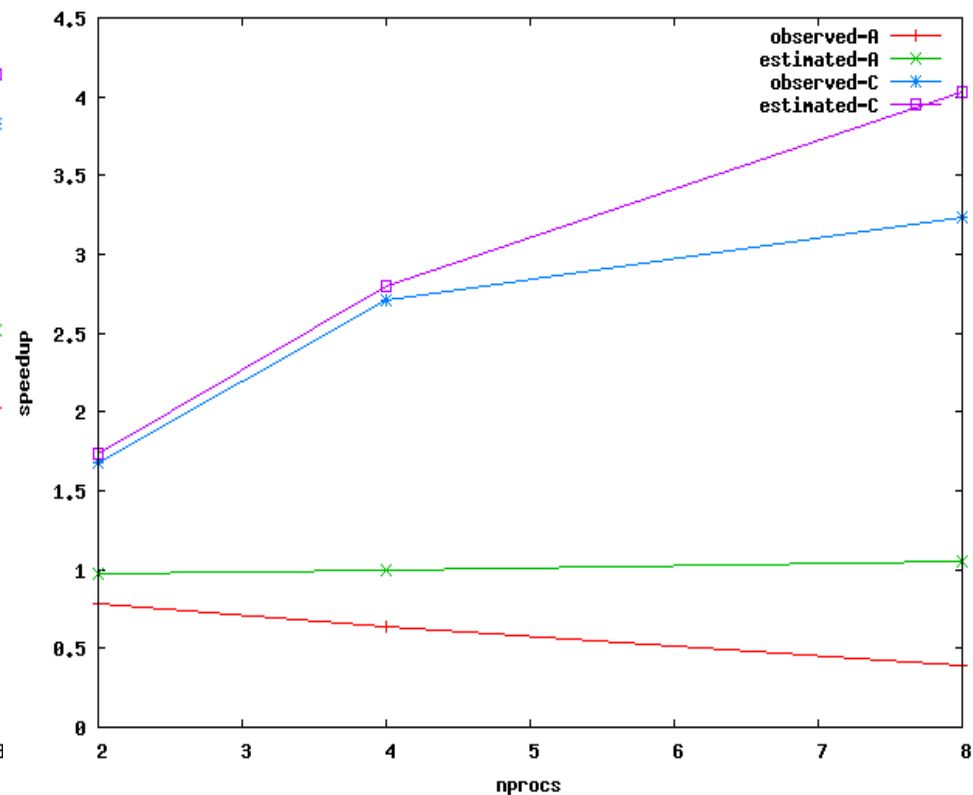


SDP Model Evaluation Results for CLOMP

BT estimation



CG estimation



average absolute fractional error: BT-A 6% BT-C 4%, CG-A 83% CG-C 11%

Heterogeneity Issues

- NPB-OMP with CLOMP has been run on a heterogeneous cluster
 - 4 AMD Athlon dual core @ 2.2Ghz nodes
 - 4 Intel Core2 dual core @2.13Ghz nodes
 - From SPEC2006 CPU published results, Intel node is around 26% faster than AMD node
 - Giga-Ethernet connection
- Estimation of best and worst elapsed times for NPB benchmarks

Heterogeneity Issue (cont.)

- T_{slow} is the elapsed time for running the benchmark on a homogeneous cluster consisting of N slow nodes.
- N is number of nodes of the heterogeneous cluster.
- T_{best} : the best estimate (load-balanced).
- T_{worst} : the worst estimate (T_{slow}).
- $r_i (1 \leq i \leq N)$ denotes the normalized execution power of i th node against the corresponding power of slow node ($r_i \geq 1$).
 - measured using 1-thread execution time fraction.

$$T_{best} = T_{slow} N / (r_1 + \dots + r_N) \quad \text{and} \quad T_{worst} = T_{slow}$$

Benchmarking Results on Heterogeneous Cluster

		nodes x threads					
		2x2		4x2		8x2	
		T_best (s)	observed (%)	T_best (s)	observed (%)	T_best (s)	observed (%)
Class A	BT	134.0	100%	120.2	96%	98.0	93%
	CG	7.9	101%	9.6	102%	13.3	102%
Class C	BT	1840.4	106%	1579.2	100%	1478.5	104%
	CG	428.6	183%	387.1	152%	330.5	163%

- Benchmarks performs close to best estimate for class A (small size) and BT class C.
 - ➔ Memory consistency (data fetching) dominates the performance.
- CG class C performs far from the ideal (T_{best}).
 - ➔ Computation contributes close to memory consistency (data fetching).

Pros and Cons of sDSM Systems

- Pros of sDSM systems:
 - provide easy programming model.
- Cons of sDSM systems:
 - the performance for some benchmarks is not satisfactory.
 - Heterogeneity of clusters some time makes things worse.

Outline

- Introduction
 - Advanced cluster systems
 - Programming models on clusters
- Cluster-enabled OpenMP systems
 - Current state-of-art
 - Performance evaluation and modeling
 - Pros and Cons
- Optimizing Cluster-enabled OpenMP
 - high-performance interconnects with RDMA
 - scheduler for heterogeneous cluster

RDMA

- Remote Direct Memory Access (RDMA or RMA)
 - enable direct access to the address space of a remote process without causing software overhead on the remote process.
 - RDMA communication also is known as one-sided communication.
- Supported by high-performance interconnects .
 - e.g InfiniBand, Myrinet, and Quadrics
 - Native verbs (hardware APIs), uDAPL
 - uDAPL is defined by DAT Collaborative to provide hardware independent interface.

Why RDMA?

- Highly efficient on point-to-point communications mechanism.
 - up to 17% decrement on latency been observed.
 - a further 50% decrement on latency been observed for multi-rail configured InfiniBand cluster.
 - ✓ Multi-ports per HCAs (Host Channel Adapters).
 - ✓ Multi HCAs per nodes.
- Enhancing communication-computation overlapping.

Optimize sDSMs for High-performance Interconnects

- Utilizing RDMA communication.
- Utilizing remote atomic operation for synchronizing locking on sDSMs.
- Tuning sDSMs for multi-rail configured clusters.
- Optimizing sDSM to enhance overlapping RDMA communication and computation.

Heterogeneity Scheduler

- Heterogeneity of a cluster highly affects the performance.
- None of the sDSM systems has addressed this issue yet.
 - OpenMP dynamic scheduling may not be efficient for sDSM systems

Optimize sDSMs for Heterogeneous Cluster

- Developing a runtime to override current simple scheduling of OpenMP runtime.
- Both computational power and network speed will be taken into account.

Conclusion

- OpenMP + sDSM systems is an easier programming model on clusters.
- Performance of sDSMs is not satisfactory.
- Utilizing promising performance advantage of RDMA on sDSMs is a potential solution
- Heterogeneity scheduler will be developed to optimize sDSMs on heterogeneous clusters.

Questions?

Thank you!

