Implementing an OpenMP Execution Environment on InfiniBand Clusters

Jie Tao [1], Wolfgang Karl [1], and Carsten Trinitis [2]

[1] Institut für Technische Informatik
Universität Karlsruhe

[2] Lehrstuhl für Rechnertechnik und Rechnerorganisation
Technische Universität München

Outline

- Motivation
- ViSMI: Virtual Shared Memory for InfiniBand clusters
- Omni/Infini: towards OpenMP execution on cluster systems
- Initial experimental results
- Conclusions

Motivation

- InfiniBand
  - Point-to-point, switched I/O interconnect architecture
  - Low latency, high bandwidth
    - Basic connection: serial at data rates of 2.5 Gbps
    - Bundled: e.g. 4X connection -- 10Gbps; 12X -- 30Gbps
  - Special feature: Remote Direct Memory Access (RDMA)
- The InfiniBand Cluster at TUM
  - Configuration: 6 Xeon (2-way), 4 Itanium 2 (4-way), 36 Opteron
  - MPI available

Software Distributed Shared Memory

- Virtually global address space on cluster architectures
- Memory consistency models
  - Sequential consistency
    - Any execution has the same result as if operations were executed in a sequential order
  - Relaxed consistency
    - Consistency through explicit synchronization: acquire & release
    - Lazy Release Consistency (LRC)
      - Home-based Lazy Release Consistency (HLRC)
        - Multiple writable copies of the same page
        - A home for a page

ViSMI: Software DSM for InfiniBand Clusters

- HLRC implementation
  - Home node: first-touch
  - *Diff*-based mechanism
    - Clean copy before first write
    - *Diff*s propagated by invalidations
  - InfiniBand hardware-based multicast
- Programming interface
  - A set of annotations
    - HLRC_Malloc
    - HLRC_Myself
    - HLRC_InitParallel
    - HLRC_Barrier
    - HLRC_Aquire
    - HLRC_Release

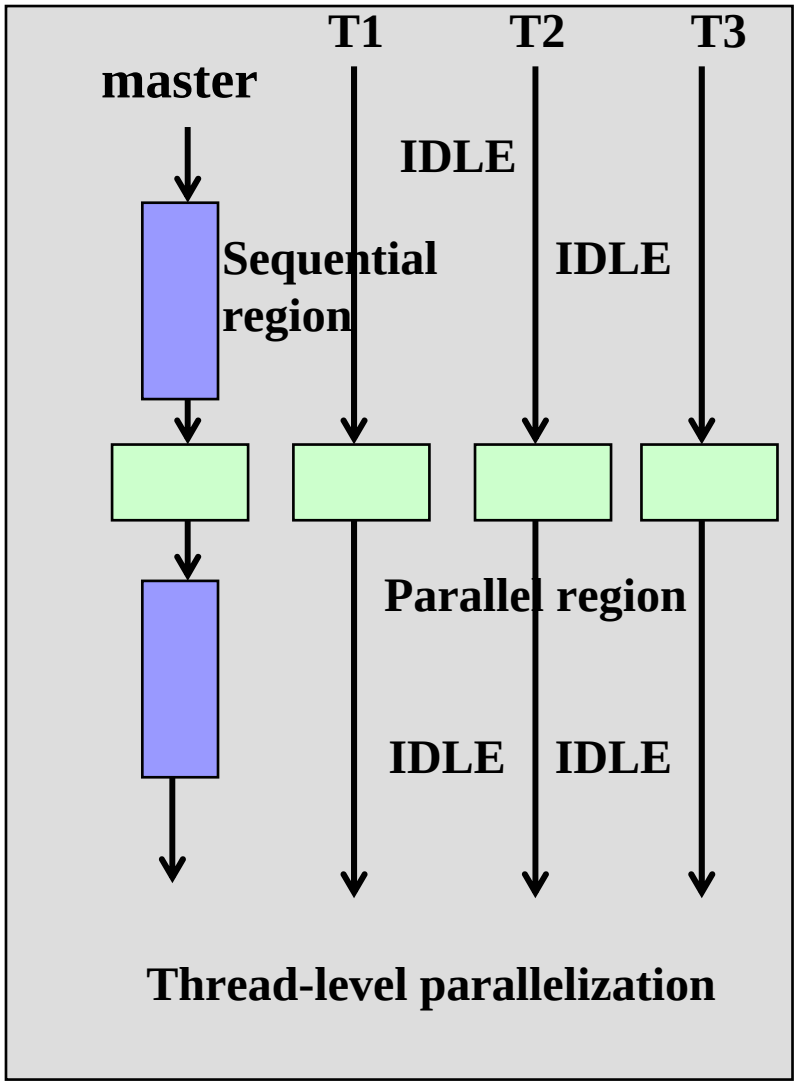
Omni/Infini: Compiler & Runtime for OpenMP Execution on InfiniBand

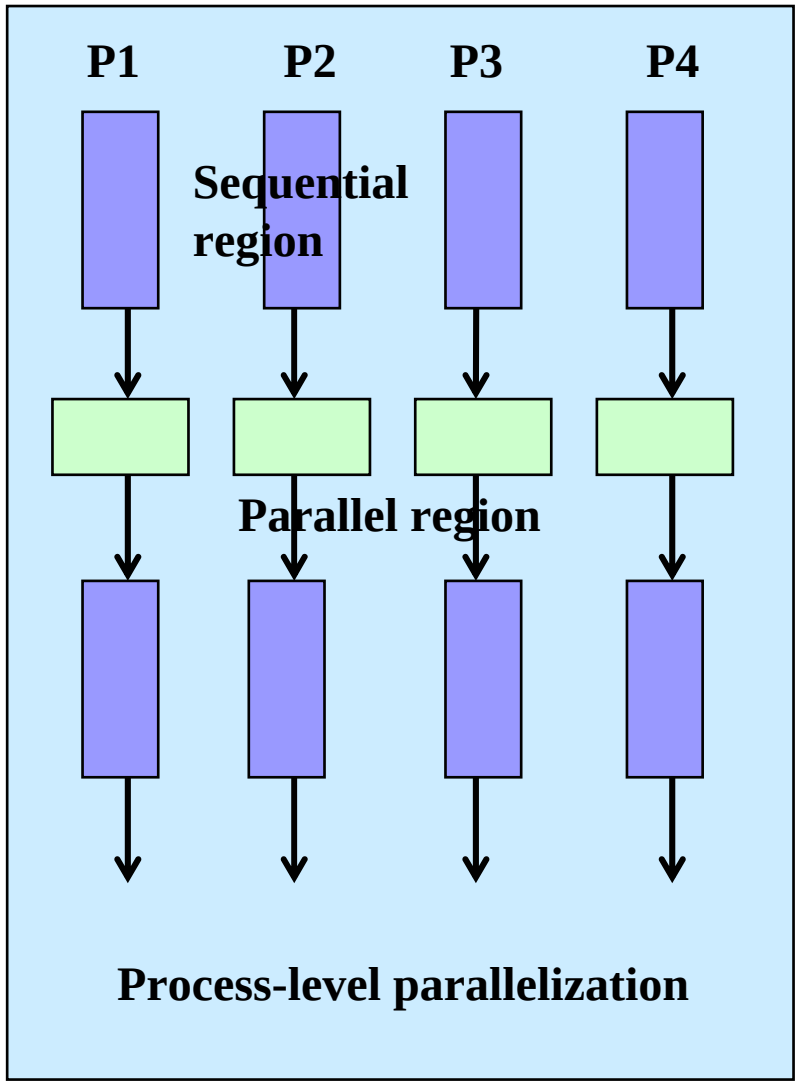
- Omni for SMP as the basis
- A new runtime library: adapting to ViSMI interface
  - Scheduling
    - ompc_static_bschd ()
    - ompc_get_num_threads ()
  - Parallelization
    - ompc_do_parallel ()
  - Synchronization
    - ompc_lock ()
    - ompc_unlock ()
    - ompc_barrier ()
  - Reduction operation
    - ompc_reduction ()
  - Specific code region
    - ompc_do_single ()
    - ompc_is_master ()
    - ompc_critical ()

Omni/Infini (cont.)

- Shared data allocation
  - Static data → pointer
  - HLRC_Malloc into shared region
- Adapting process structure to thread structure
  - ViSMI: process structure
  - OpenMP: thread structure

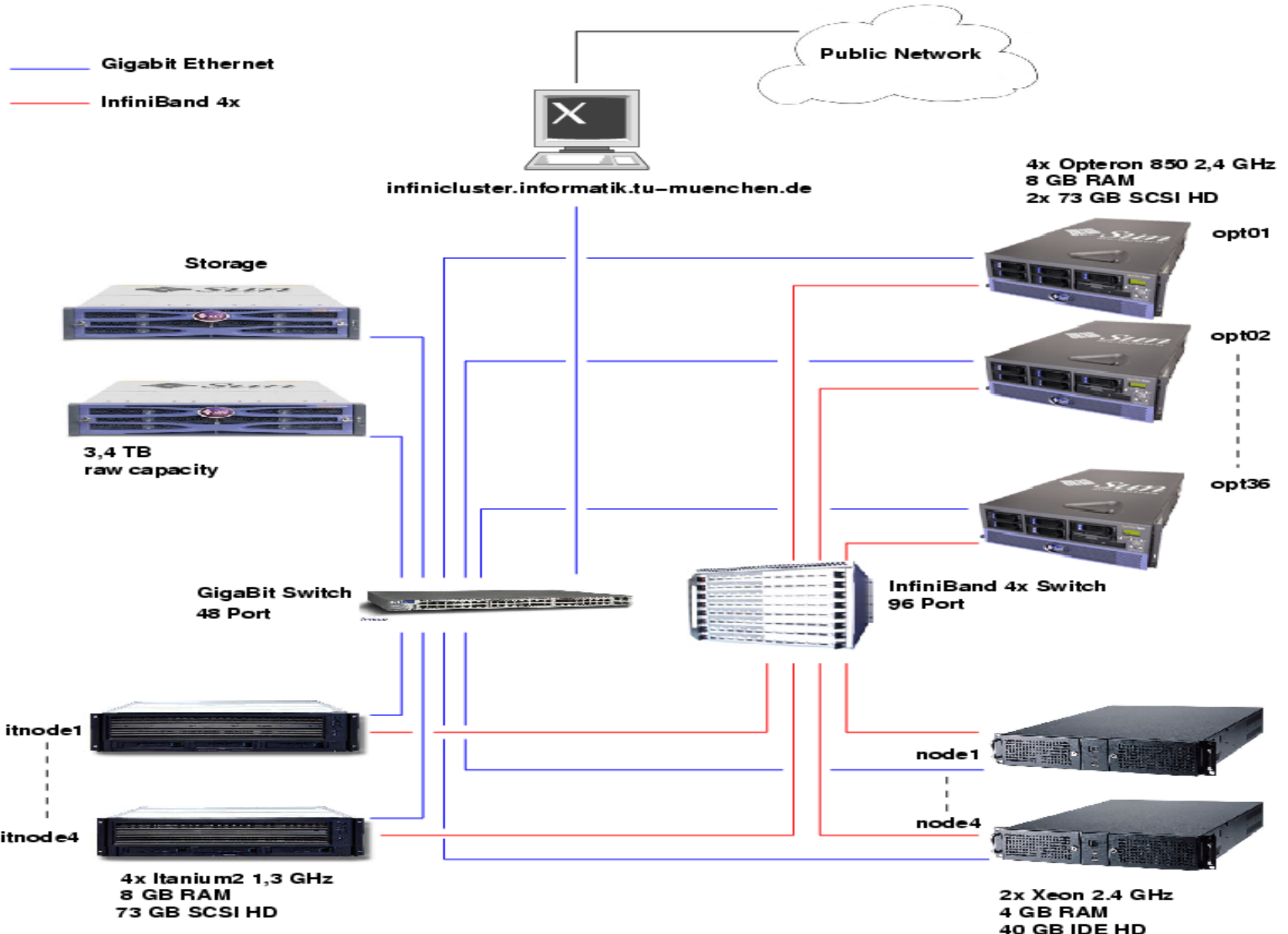
Thread-level & Process-level Parallelization



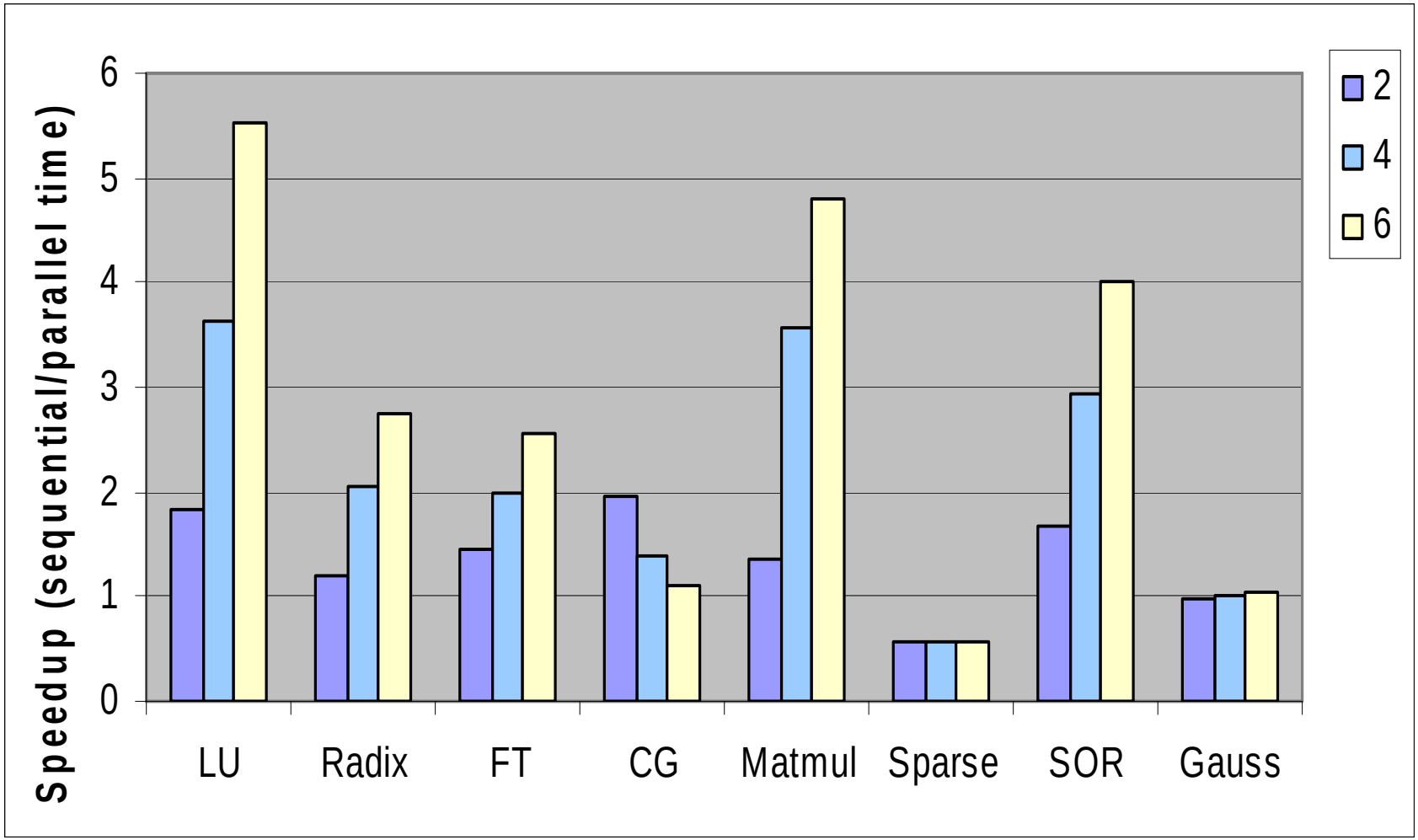


Experimental Results

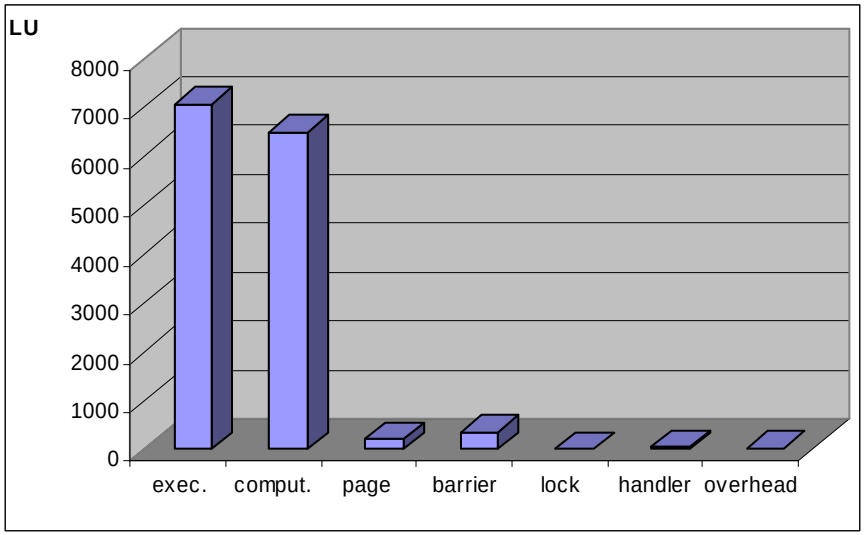
- Platform and setup
  - Initial: 6 Xeon nodes
  - Most recently: Opteron nodes
- Application:
  - NAS parallel benchmark suite
  - OpenMP version of SPLASH-2
  - Small codes
    - SMP programming course
    - Self-coded

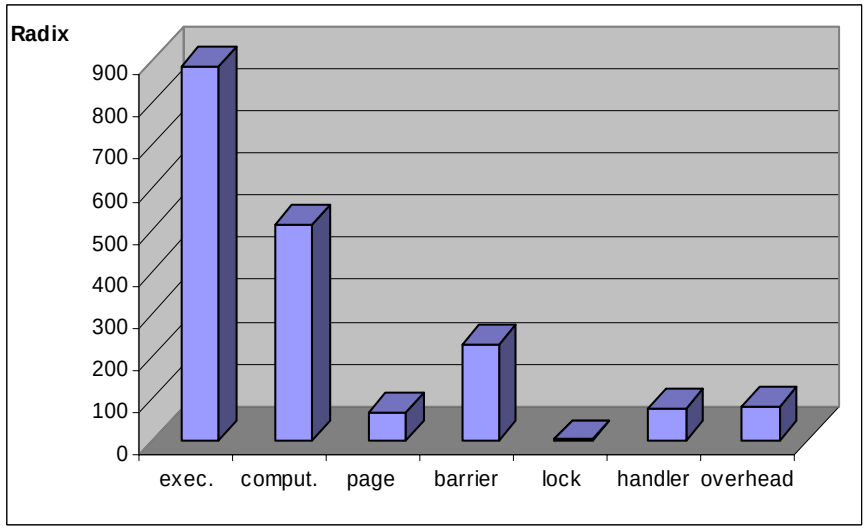

Gigabit Ethernet
InfiniBand 4x
Public Network
infinicluster.informatik.tu-muenchen.de
4x Opteron 850 2,4 GHz
8 GB RAM
2x 73 GB SCSI HD
opt01
opt02
opt36
Storage
3,4 TB
raw capacity
GigaBit Switch
48 Port
InfiniBand 4x Switch
96 Port
itnode1
itnode4
4x Itanium2 1,3 GHz
8 GB RAM
73 GB SCSI HD
node1
node4
2x Xeon 2.4 GHz
4 GB RAM
40 GB IDE HD

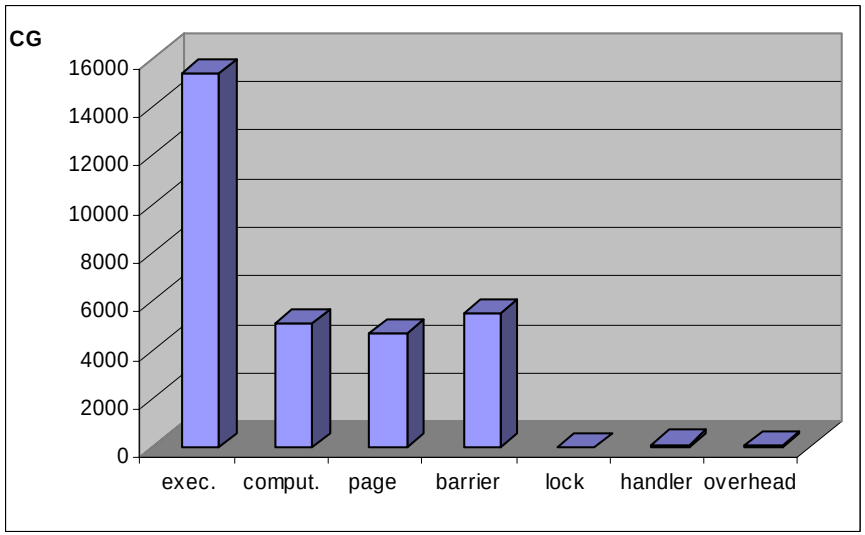
Speedup

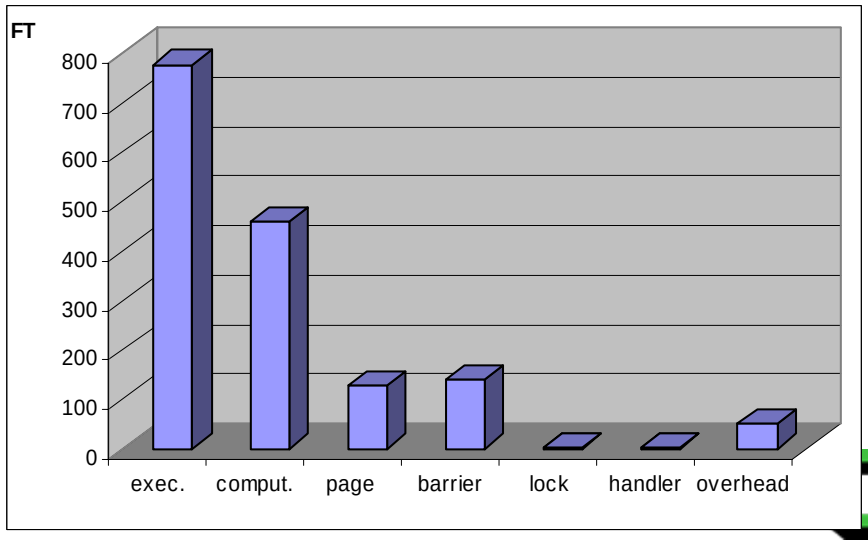


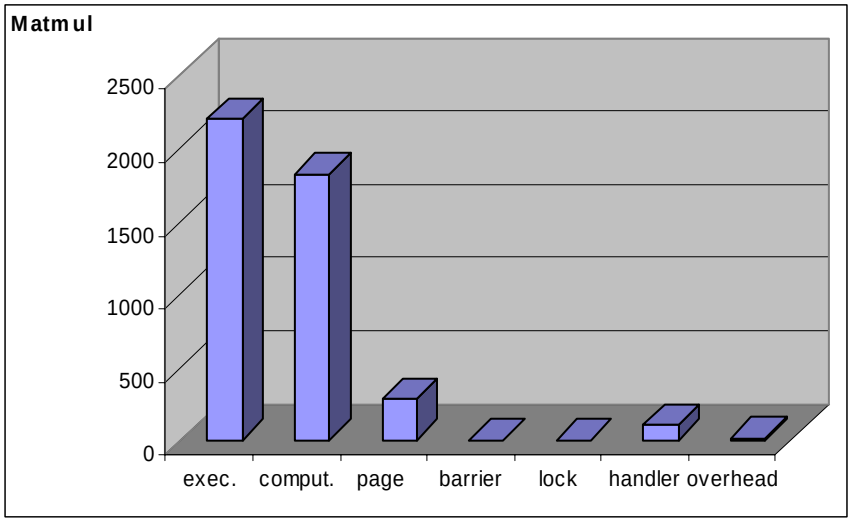
Normalized Execution Time Breakdown

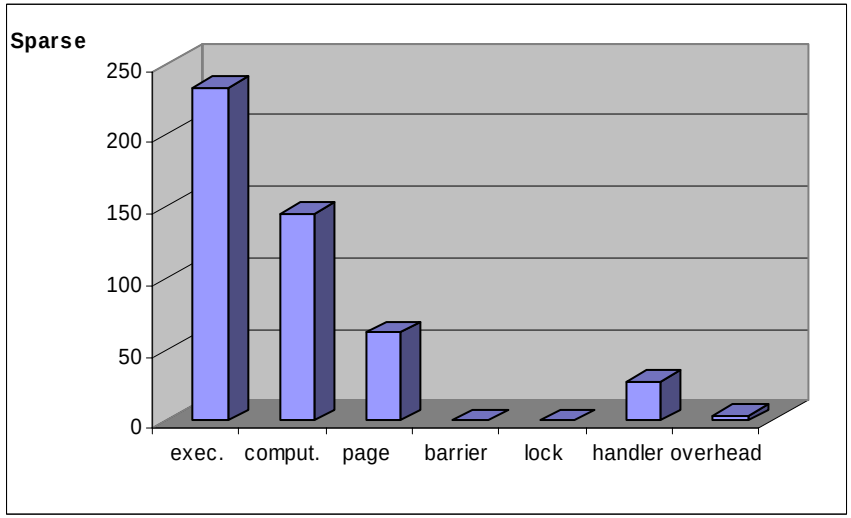


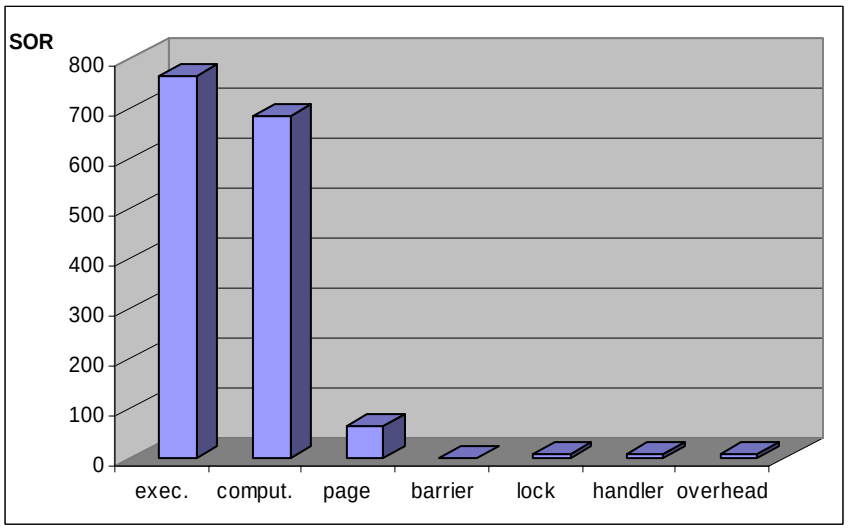


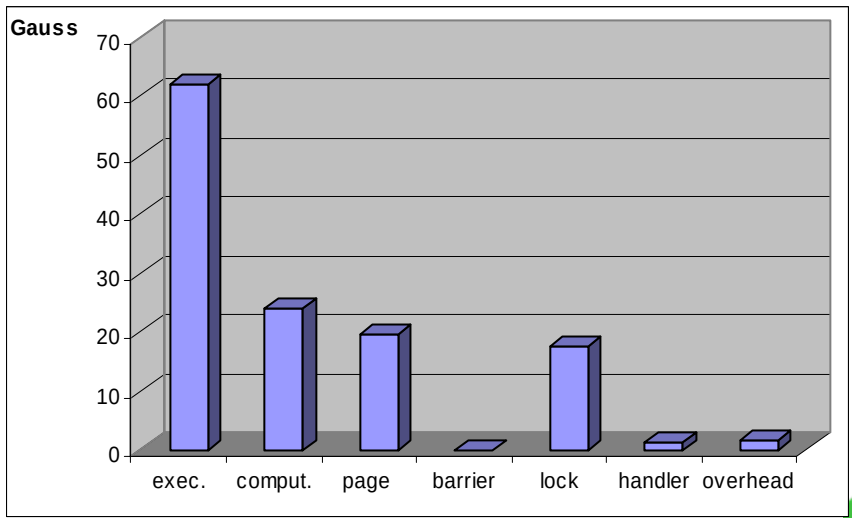




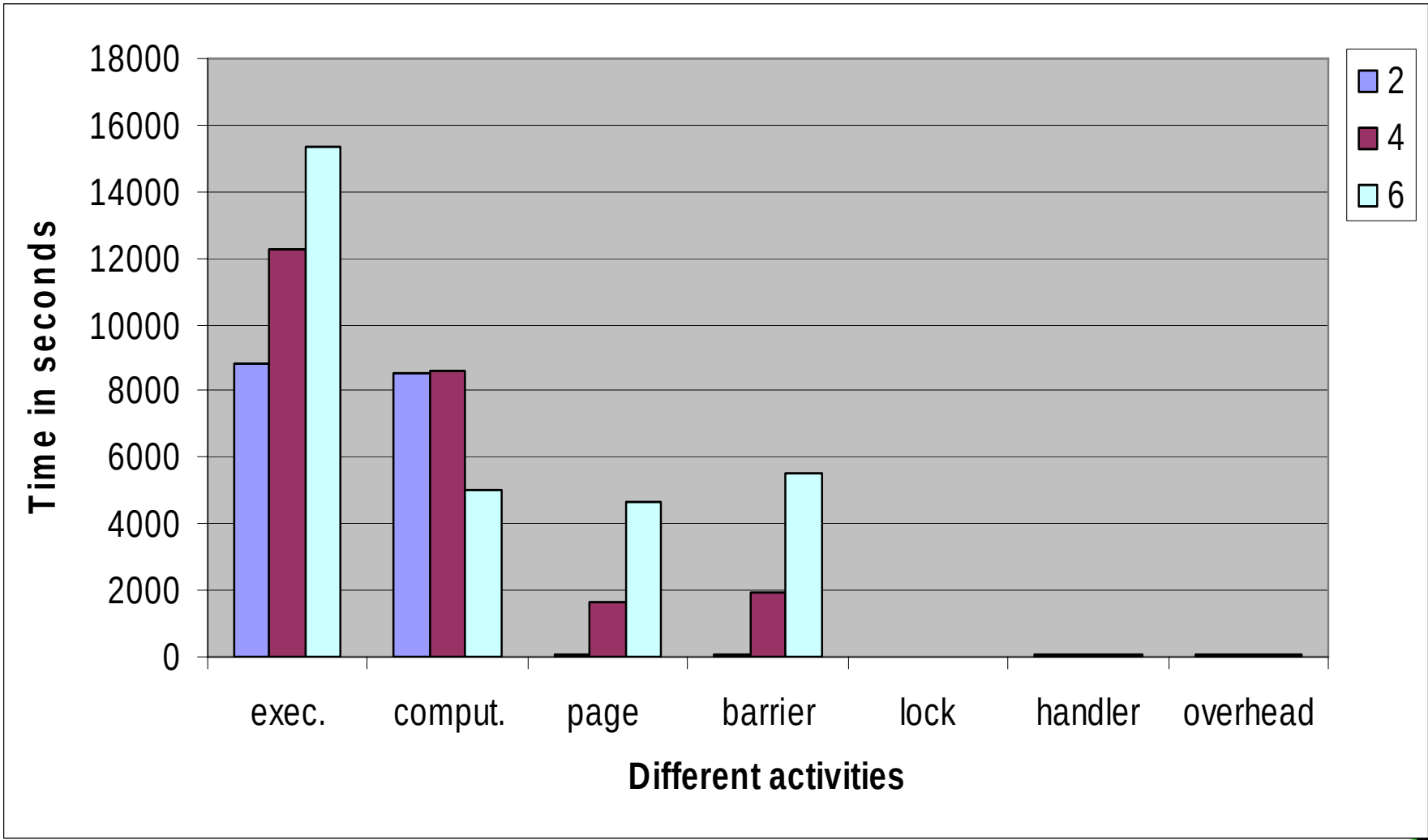




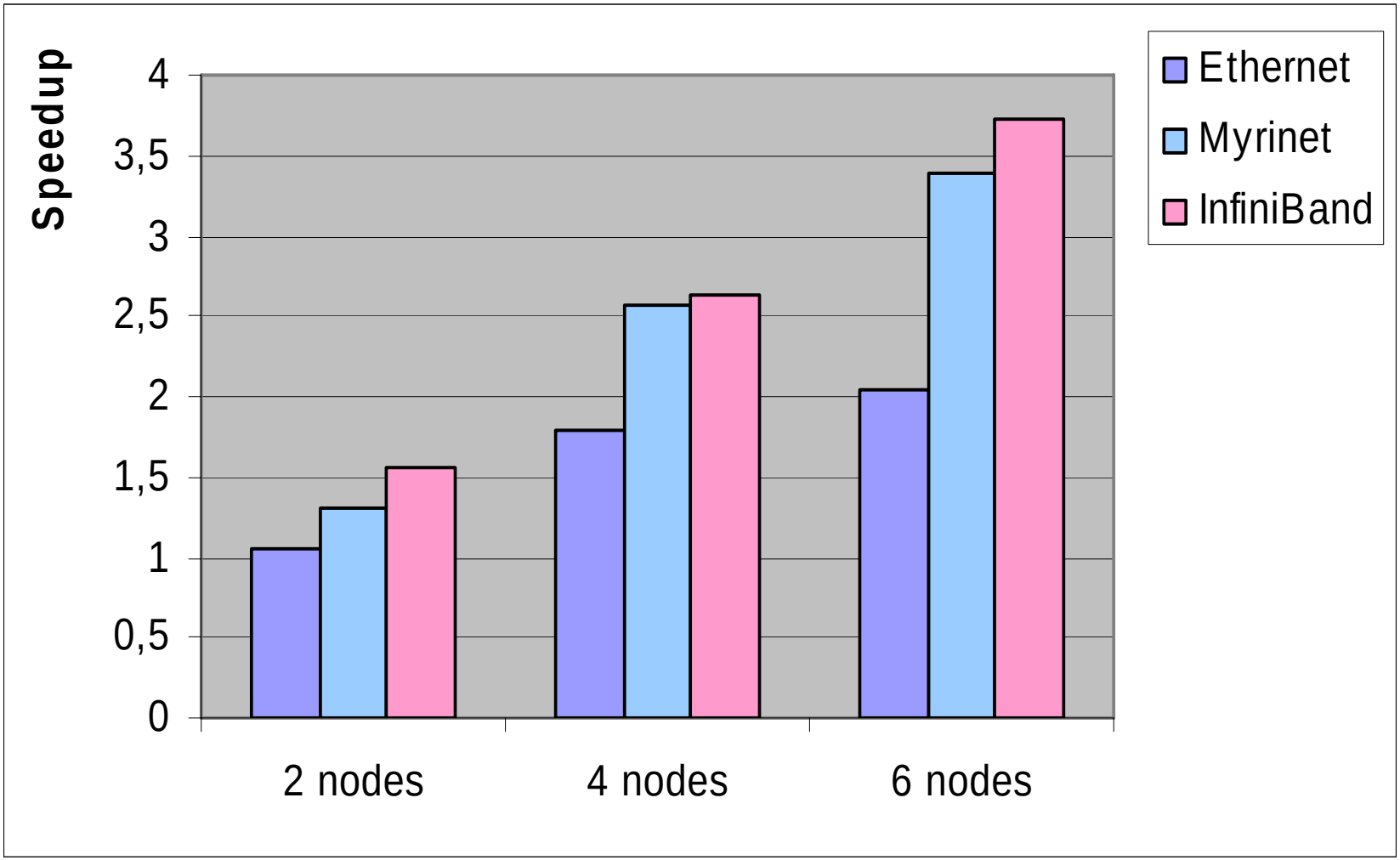




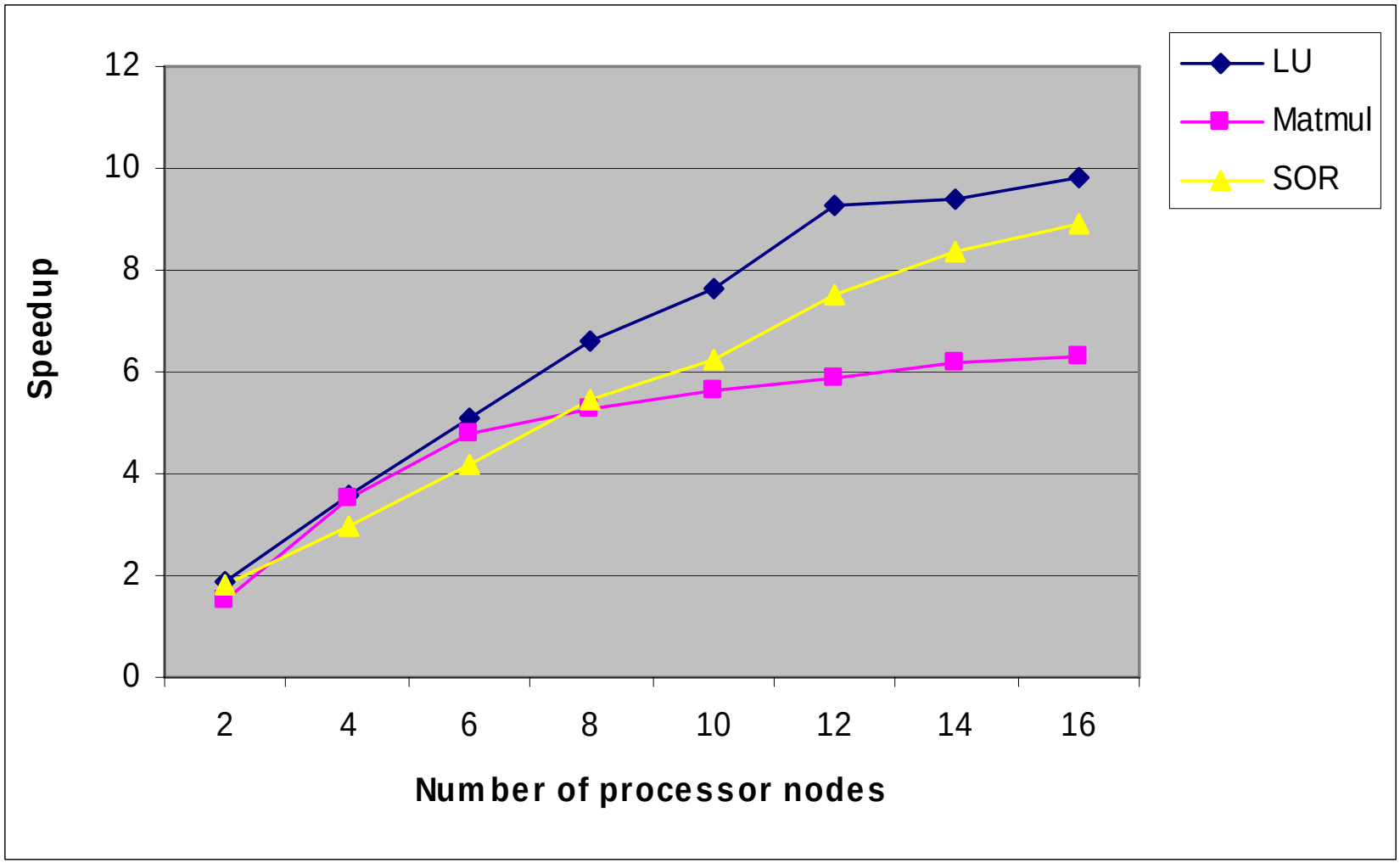
CG Time Breakdown



Comparison with Omni/SCASH



Scalability



Conclusions

- Summary
  - An OpenMP execution environment for InfiniBand
    - Built on top of a software DSM
    - Omni compiler as basis
  - Speedup on 6 nodes: up to 5.22
- Future work
  - Optimization: barrier, page operation
  - Test with realistic applications