

First Experiences with Intel Cluster OpenMP

Christian Terboven, Dieter an Mey,
Dirk Schmidl, Marcus Wagner

{terboven, anmey, schmidl, wagner}
@rz.rwth-aachen.de

Center for Computing and Communication
RWTH Aachen University

Agenda

- Intel Cluster OpenMP
 - OpenMP Memory Model
 - Consistency
- Micro-Benchmarks
 - EPCC
 - DSM investigations
- Applications
 - Jacobi
 - SMXV
 - Panta
- Conclusion and Future Work

2

Cluster OpenMP

- OpenMP versus MPI
 - Shared-Memory is more intuitive
 - OpenMP allows for incremental parallelization
 - BUT: Shared-Memory is bound to one machine
- OpenMP for clusters, based on ...
 - ... TreadMarks (subset of OpenMP only)
 - ... Omni/SCASH (no support for C++)
 - ... Linux kernel modifications, e.g. Kerrighed (compatibility)
- Intel Cluster OpenMP (CIOMP)
 - Full support of OpenMP 2.5 (no nesting support)
 - C, C++ and FORTRAN
 - First commercial implementation (based on TreadMarks)

3

Center for

Computing and

Communication

Cluster OpenMP

Micro-
Benchmarks

Applications

Conclusion

Intel Cluster OpenMP

- Fork-Join model
 - Worker threads are created at the entrance of a Parallel Region
 - And suspended at the end, ready to be reused at next opportunity
- Cluster OpenMP
 - Distributed-Shared-Memory Model (DSM)
 - Intel: `sharable` (not all variables can be made sharable automatically)
- Weak memory model
 - Temporary View

Master / Initial Thread

Serial Part

Parallel Region

Slave
Threads

4

Center for

Computing and

Communication

Cluster OpenMP

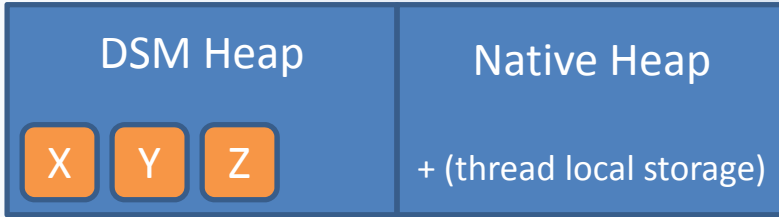
Micro-
Benchmarks

Applications

Conclusion

Consistency model in DSM

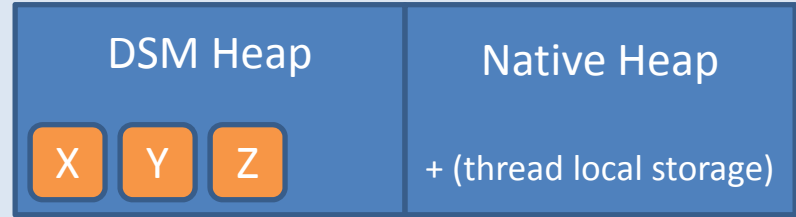
Node 0: ClOMP process A w/ 1 thread



Write X → Twin page creation (X)

Write Z → Twin page creation (Z)

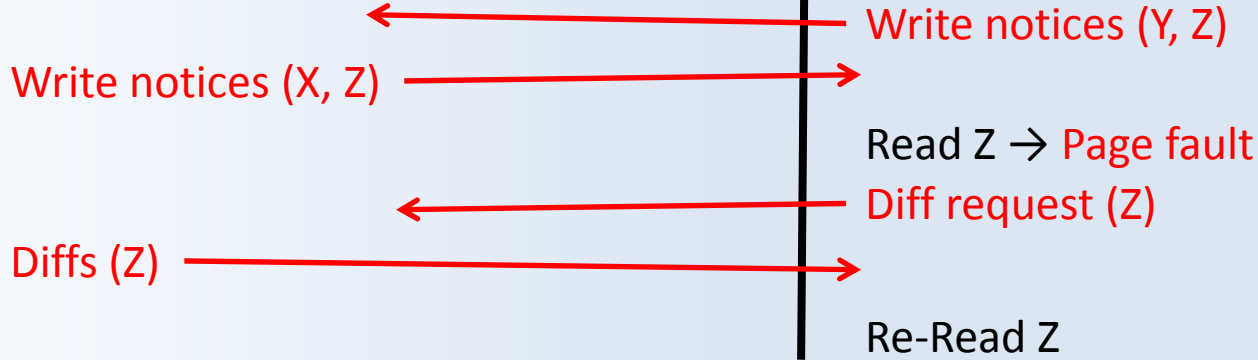
Node 1: ClOMP process B w/ 1 thread



Write Y → Twin page creation (Y)

Write Z → Twin page creation (Z)

OpenMP Barrier



- Initially, pages in the DSM are read + write protected
- Write notices are sent to node 0 and then propagated

5

Agenda

- Intel Cluster OpenMP
 - OpenMP Memory Model
 - Consistency
- Micro-Benchmarks
 - EPCC
 - DSM investigations
- Applications
 - Jacobi
 - SMXV
 - Panta
- Conclusion and Future Work

6

EPCC Micro-Benchmarks on two nodes

J. M. Bull. Measuring Synchronization and Scheduling Overheads in OpenMP. 1999.

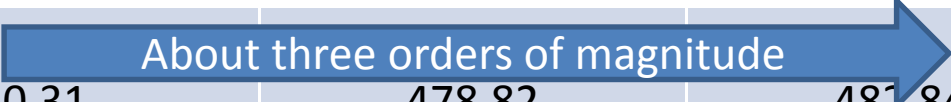
	OpenMP	CIOMP (Eth)	CIOMP (IB)
PARALLEL FOR			
1 thread p. node	0.31	478.82	482.84
2 threads p. node	1.00	1159.53	720.62
4 threads p. node	1.12	1540.97	962.52
BARRIER			
1 thread p. node	0.01	478.24	481.37
2 threads p. node	0.43	738.38	589.95
4 threads p. node	0.60	751.61	634.63
REDUCTION			
1 thread p. node	0.35	479.44	481.34
2 threads p. node	1.54	1888.25	1302.87
4 threads p. node	2.32	3315.19	2660.42

Overhead in microseconds [us]. Intel 10.0.025 compilers for 64bit Linux. Dell PowerEdge 1950 cluster, 2x Intel Xeon 5160 (dual-core, 3.0 GHz). Gigabit Ethernet (Eth) versus 4x SDR InfiniBand (IB) on PCI-Express.

7

EPCC Micro-Benchmarks on two nodes

J. M. Bull. Measuring Synchronization and Scheduling Overheads in OpenMP. 1999.

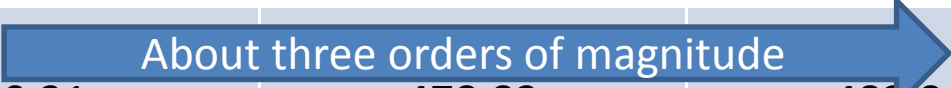

	OpenMP	CIOMP (Eth)	CIOMP (IB)
PARALLEL FOR	About three orders of magnitude 		
1 thread p. node	0.31	478.82	482.84
2 threads p. node	1.00	1159.53	720.62
4 threads p. node	1.12	1540.97	962.52
BARRIER			
1 thread p. node	0.01	478.24	481.37
2 threads p. node	0.43	738.38	589.95
4 threads p. node	0.60	751.61	634.63
REDUCTION			
1 thread p. node	0.35	479.44	481.34
2 threads p. node	1.54	1888.25	1302.87
4 threads p. node	2.32	3315.19	2660.42

Overhead in microseconds [us]. Intel 10.0.025 compilers for 64bit Linux. Dell PowerEdge 1950 cluster, 2x Intel Xeon 5160 (dual-core, 3.0 GHz). Gigabit Ethernet (Eth) versus 4x SDR InfiniBand (IB) on PCI-Express.

8

EPCC Micro-Benchmarks on two nodes

J. M. Bull. Measuring Synchronization and Scheduling Overheads in OpenMP. 1999.

	OpenMP	CIOMP (Eth)	CIOMP (IB)
PARALLEL FOR	About three orders of magnitude 		
1 thread p. node	0.31	478.82	482.84
2 threads p. node	1.00	1159.53	720.62
4 threads p. node	1.12	1540.97	962.52
BARRIER	IB is significantly faster 		
1 thread p. node	0.01	478.24	481.37
2 threads p. node	0.43	738.38	589.95
4 threads p. node	0.60	751.61	634.63
REDUCTION			
1 thread p. node	0.35	479.44	481.34
2 threads p. node	1.54	1888.25	1302.87
4 threads p. node	2.32	3315.19	2660.42

Overhead in microseconds [us]. Intel 10.0.025 compilers for 64bit Linux. Dell PowerEdge 1950 cluster, 2x Intel Xeon 5160 (dual-core, 3.0 GHz). Gigabit Ethernet (Eth) versus 4x SDR InfiniBand (IB) on PCI-Express.

EPCC Micro-Benchmarks on two nodes

J. M. Bull. Measuring Synchronization and Scheduling Overheads in OpenMP. 1999.

	OpenMP	CIOMP (Eth)	CIOMP (IB)
PARALLEL FOR	About three orders of magnitude		
1 thread p. node	0.31	478.82	482.84
2 threads p. node	1.00	1159.53	720.62
4 threads p. node	1.12	1540.97	962.52
BARRIER	IB is significantly faster		
1 thread p. node	0.01	478.24	481.37
2 threads p. node	0.43	738.38	589.95
4 threads p. node	0.60	751.61	634.63
REDUCTION	Intel MPI is faster than Intel Cluster OpenMP: up to 50%!		
1 thread p. node	0.35	479.44	481.34
2 threads p. node	1.54		1302.87
4 threads p. node	2.32		2660.42

Overhead in microseconds [us]. Intel Cluster OpenMP vs. Intel MPI vs. OpenMP. Linux. Dell PowerEdge 1950 cluster, 2x Intel Xeon 5160 (dual-core, 3.0 GHz). Gigabit Ethernet (Eth) versus 4x SDR InfiniBand (IB) on PCI-Express.

10

DSM investigations with one thread per node

	Allocate page in DSM heap	Read page from other CIOMP thread	Write page to other CIOMP thread
OpenMP	0.85	1.8	2.32
CIOMP (Eth)			
1 node	3.81	2.74	2.44
2 nodes	10.75	255.56	251.82
CIOMP (IB)			
1 node	3.80	1.76	4.26
2 nodes	26.33	101.34	104.54

Overhead in microseconds [us].

- All measurements with two threads:
 - OpenMP: two threads on one node
 - Cluster OpenMP: two threads in total on one / two nodes

DSM investigations with one thread per node

	Allocate page in DSM heap	Read page from other CIOMP thread	Write page to other CIOMP thread
OpenMP	0.85	1.8	2.32
CIOMP (Eth)	↑ Eth is faster		
1 node		3.81	2.74
2 nodes	10.75	255.56	251.82
CIOMP (IB)			
1 node	3.80	1.76	4.26
2 nodes	26.33	101.34	104.54

Overhead in microseconds [us].

- All measurements with two threads:
 - OpenMP: two threads on one node
 - Cluster OpenMP: two threads in total on one / two nodes

DSM investigations with one thread per node

	Allocate page in DSM heap	Read page from other CIOMP thread	Write page to other CIOMP thread
OpenMP	0.85	1.8	2.32
CIOMP (Eth)			
1 node	3.81	2.74	2.44
2 nodes	10.75	255.56	251.82
CIOMP (IB)			
1 node	3.80	1.76	4.26
2 nodes	26.33	101.34	104.54

Eth is faster

IB is faster

1.76
101.34

Overhead in microseconds [us].

- All measurements with two threads:
 - OpenMP: two threads on one node
 - Cluster OpenMP : two threads in total

Starting Intel Cluster OpenMP with multiple threads per process (node) can improve performance significantly.

Agenda

- Intel Cluster OpenMP
 - OpenMP Memory Model
 - Consistency
- Micro-Benchmarks
 - EPCC
 - DSM investigations
- Applications
 - Jacobi
 - SMXV
 - Panta
- Conclusion and Future Work

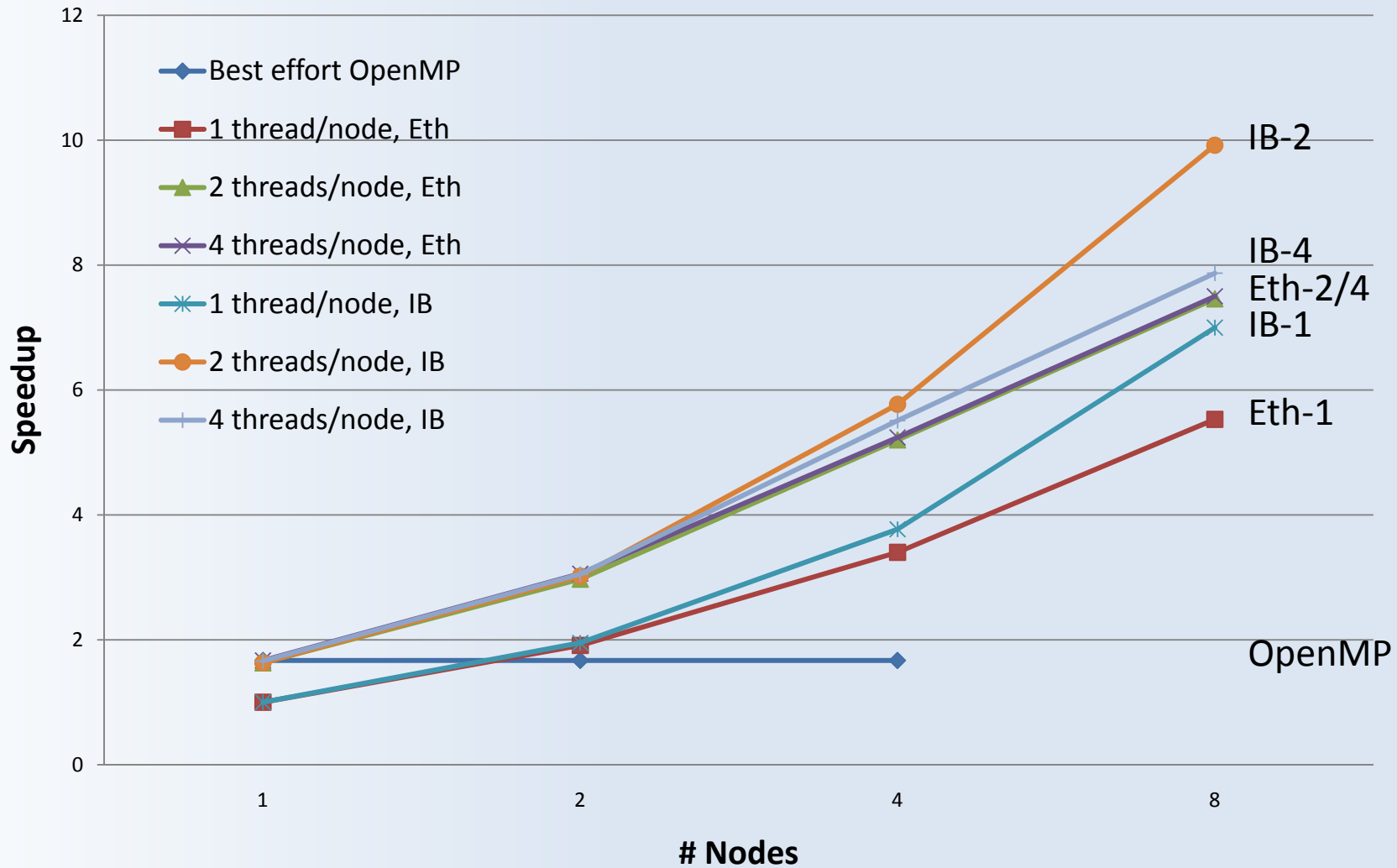
14

Jacobi

- Simple example program: Jacobian solver (www.openmp.org)
 - Matrix size of 6000 x 6000
 - Limited to 100 iterations
 - Parallelization strategy: Domain decomposition
- Performance tuning:
 - Thread binding (`KMP_AFFINITY`)
 - Strategy: *scattered*

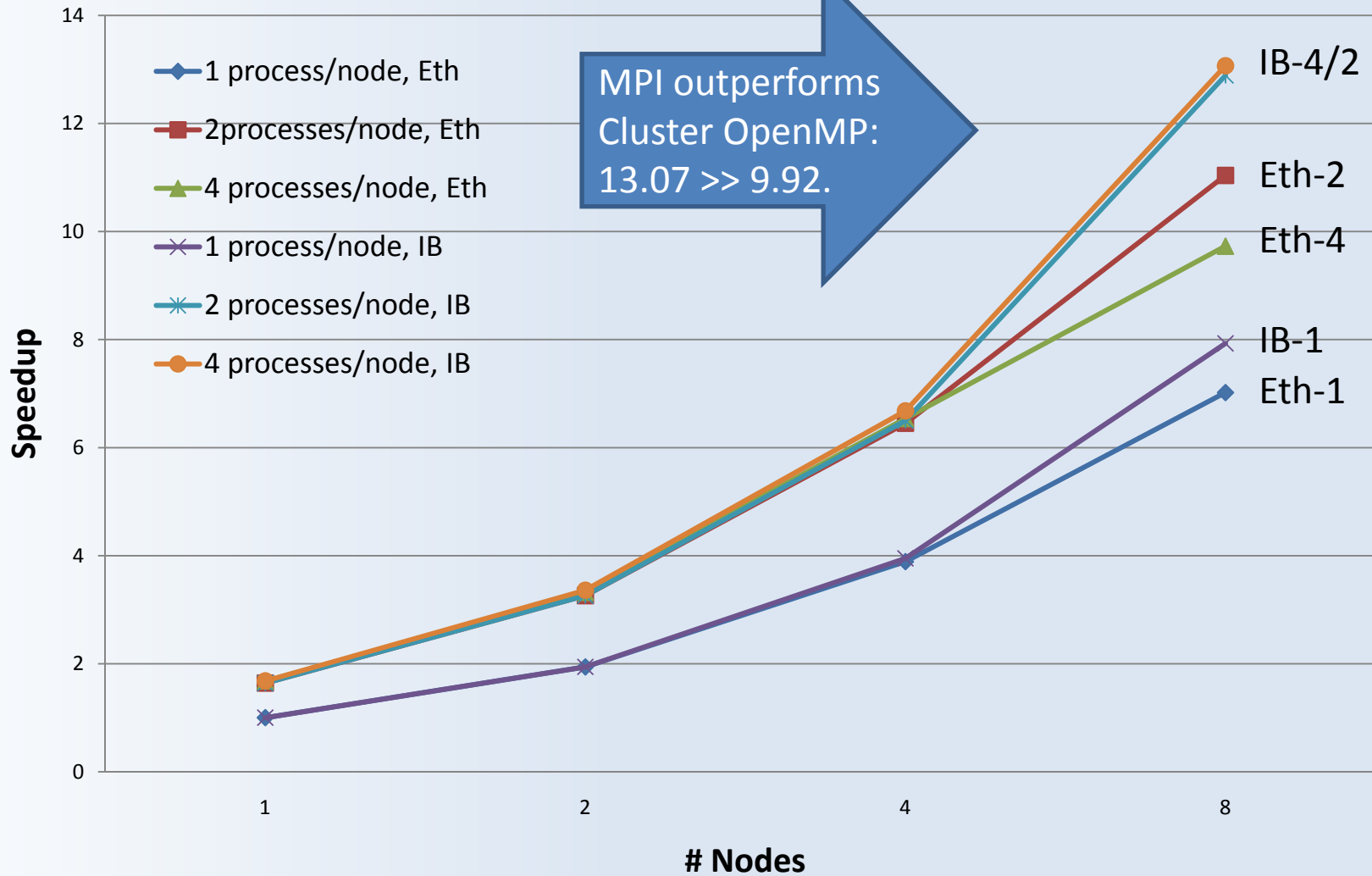
15

Jacobi: Intel Cluster OpenMP

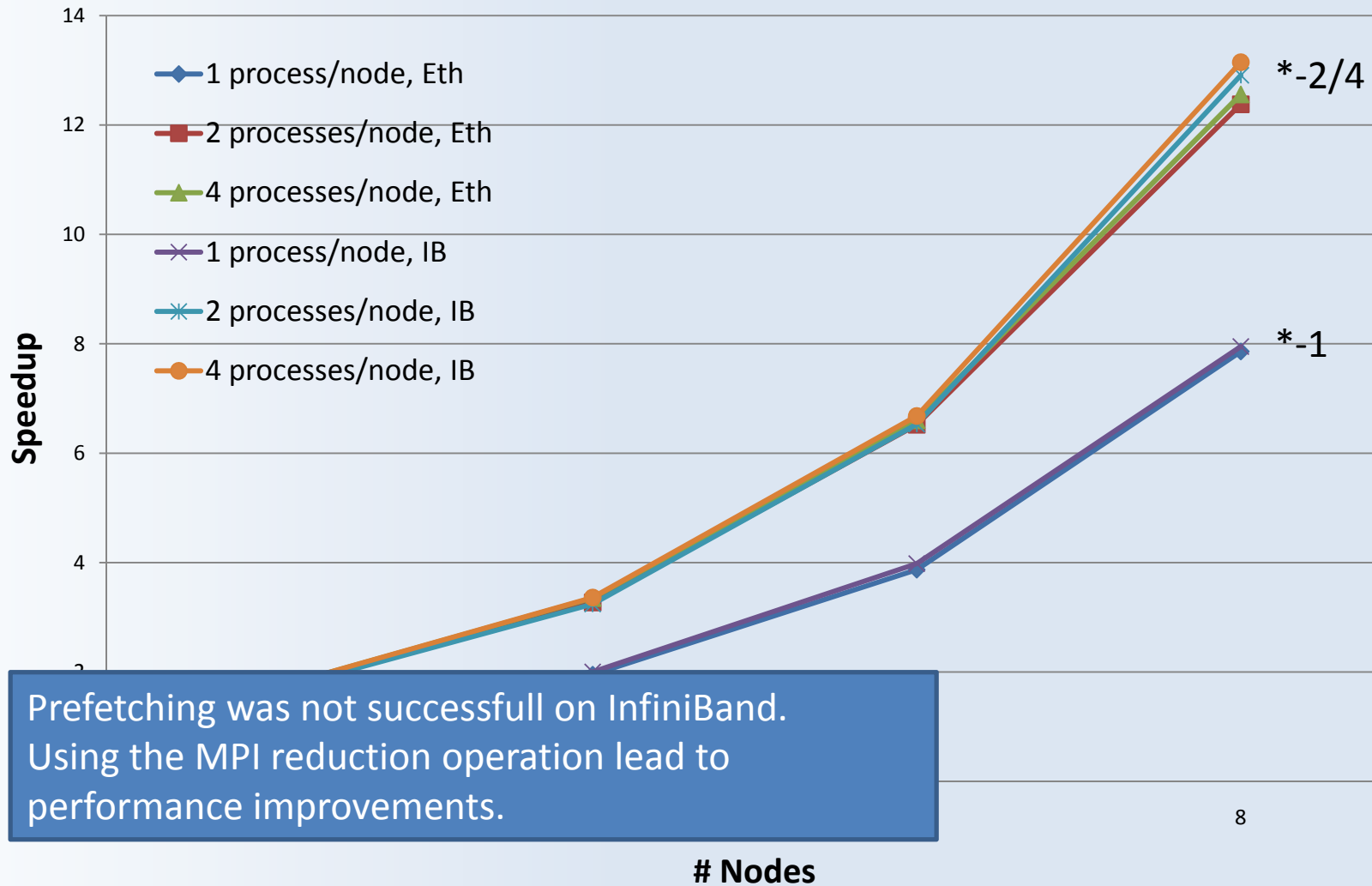


16

Jacobi: sync. MPI



Jacobi: async. MPI



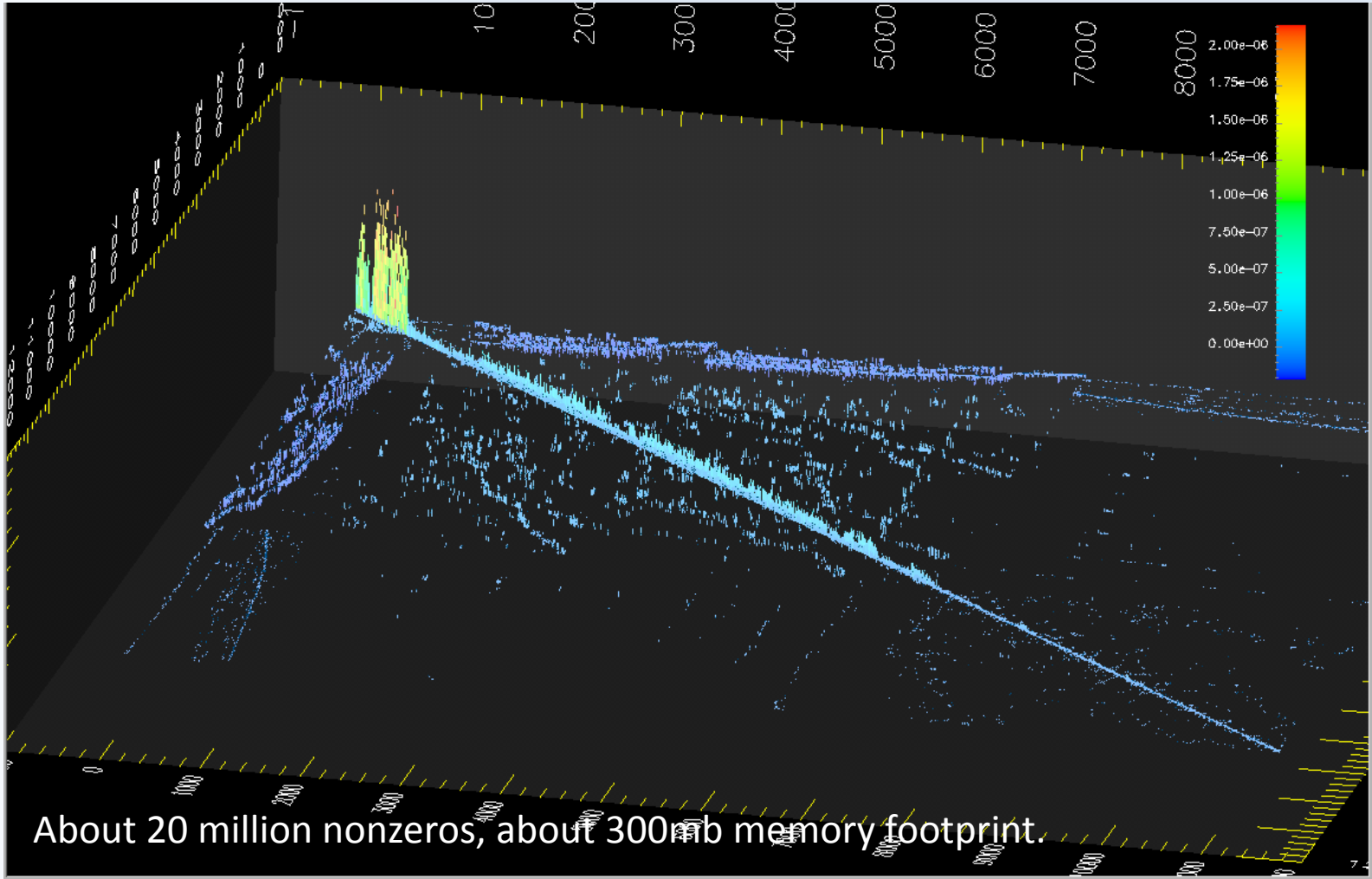
Prefetching was not successful on InfiniBand. Using the MPI reduction operation lead to performance improvements.

18

8

Sparse Matrix-Vector-Multiplication

- DROPS, a C++ Navier-Stokes solver of two-phase flows



19

Center for

Computing and
Communication

Cluster OpenMP

Micro-
Benchmarks

Applications

Conclusion &
Future Work

Sparse Matrix-Vector-Multiplication [Mflop/s]

	<u>rows</u>		<u>nonzeros</u>	
	1 thread p. node	4 threads p. node	1 thread p. node	4 threads p. node
OpenMP, UMA	561.9	906	561.5	978.1
OpenMP, ccNUMA	326.3	793.9	324.5	1147.6
CIOMP, Eth, 1 node	548.0	887.2	551.8	939.4
CIOMP, Eth, 2 nodes	113.0	540.1	1058.7	1382.4
CIOMP, Eth, 4 nodes	14.5	136.8	2037.9	2435.6
CIOMP, IB, 1 node	547.9	817.9	551.9	940.4
CIOMP, IB, 2 nodes	904.4	1208.4	1072.0	1415.3
CIOMP, IB, 4 nodes	1328.3	1845.4	2075.0	2536.6

ccNUMA: Sun Fire V40z server, 4x AMD Opteron 848 (single-core, 2.2 GHz).

- rows-strategy: parallel loop over #rows, dynamic loop sched.
- nonzeros-strategy: #nonzeros statically partitioned

20

Sparse Matrix-Vector-Multiplication [Mflop/s]

	<u>rows</u>		<u>nonzeros</u>	
	1 thread p. node	4 threads p. node	1 thread p. node	4 threads p. node
OpenM	561.9	906	561.5	978.1
OpenM MA	326.3	793.9	324.5	1147.6
CIOMP node	548.0	887.2	551.8	939.4
CIOMP nodes	113.0	540.1	1058.7	1382.4
CIOMP nodes	14.5	136.8	2037.9	2435.6
CIOMP nodes	547.9	817.9	551.9	940.4
CIOMP nodes	904.4	1208.4	1072.0	1415.3
CIOMP, 16 nodes	1328.3	1845.4	2075.0	2536.6

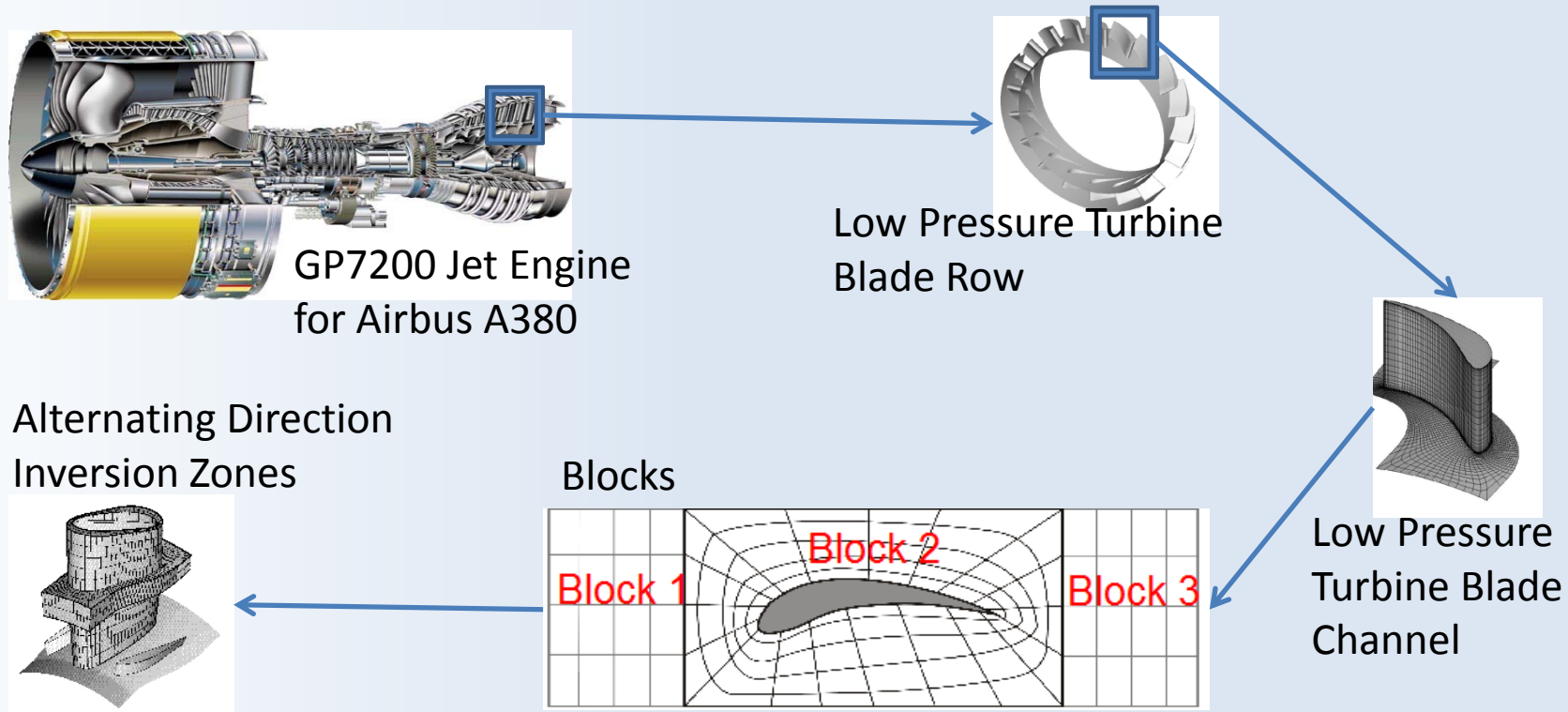
Cluster OpenMP similar to ccNUMA

ccNUMA: Sun Fire V40z server, 4x AMD Opteron 848 (single-core, 2.2 GHz).

- rows-strategy: parallel loop over #rows, dynamic loop sched.
- nonzeros-strategy: #nonzeros statically partitioned

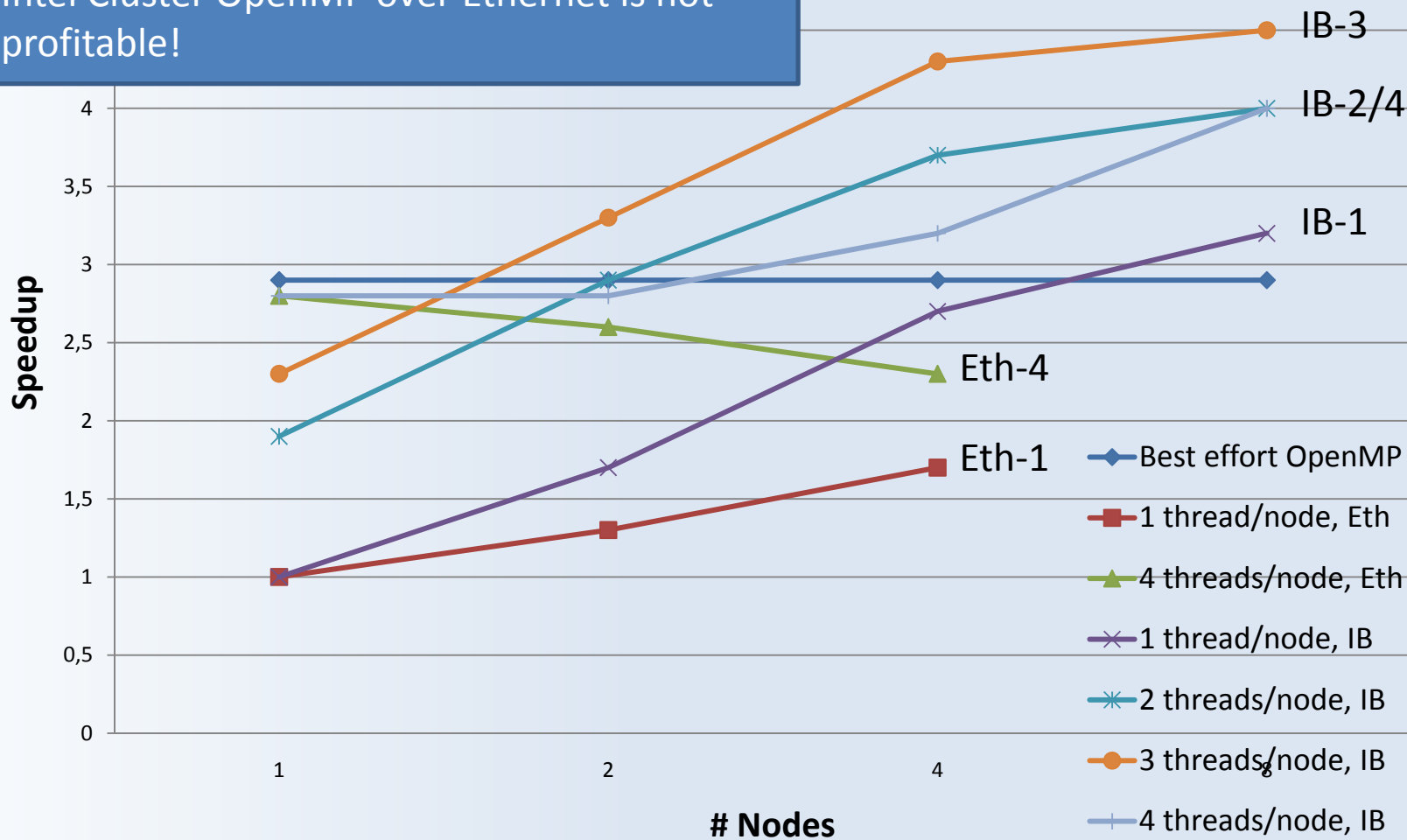
PANTA

- o Developed at IST at RWTH Aachen University
 - Computation of turbomachinery flow → solution of PDEs
 - About 50,000 lines of Fortran90 code
 - Hybrid code. Here: OpenMP-parallel loop over 80 inv. zones

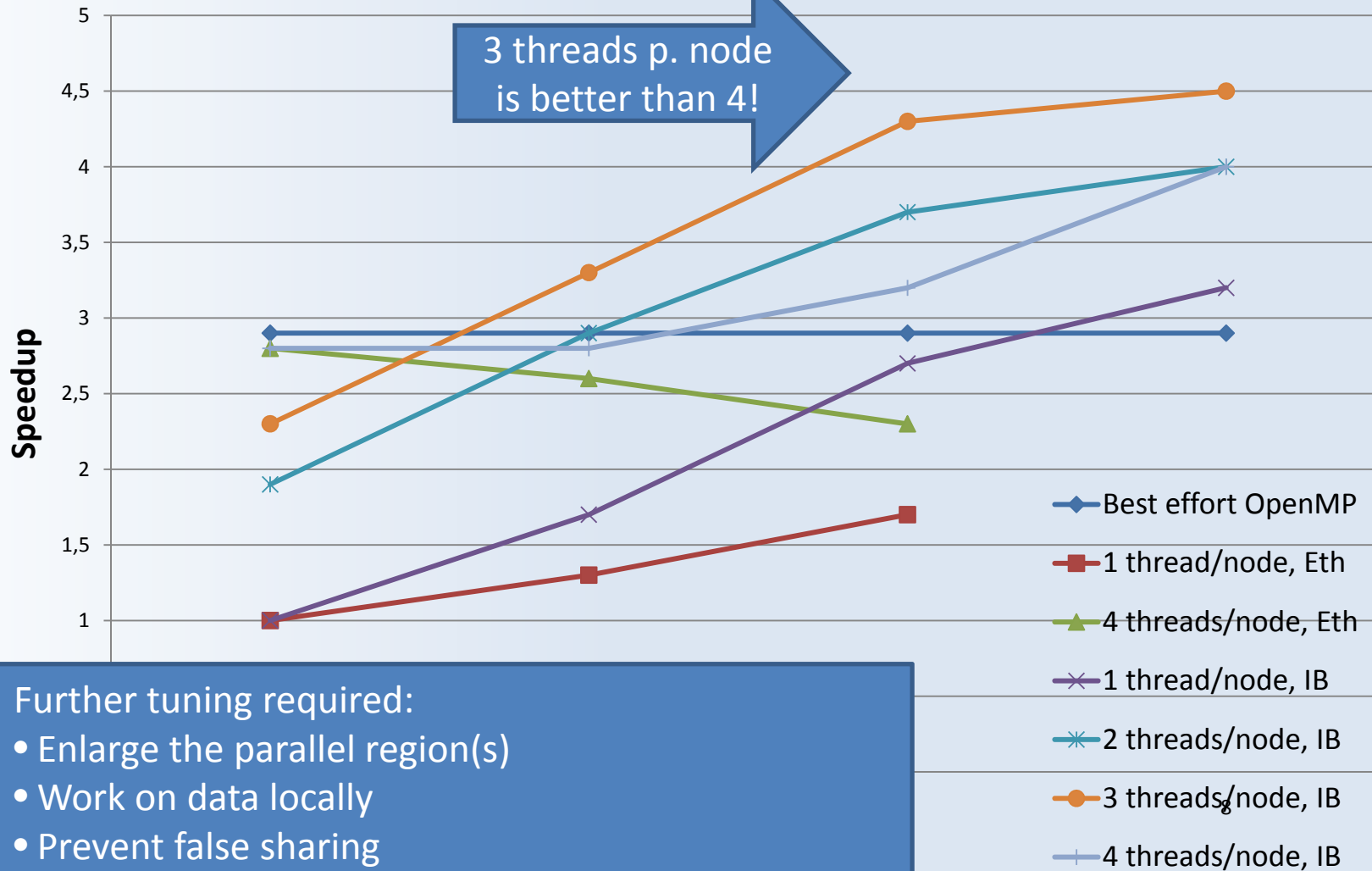


PANTA

OpenMP scalability on one node is limited.
Intel Cluster OpenMP over Ethernet is not profitable!



PANTA



Further tuning required:

- Enlarge the parallel region(s)
- Work on data locally
- Prevent false sharing

Agenda

- Intel Cluster OpenMP
 - OpenMP Memory Model
 - Consistency
- Micro-Benchmarks
 - EPCC
 - DSM investigations
- Applications
 - Jacobi
 - SMXV
 - Panta
- Conclusion and Future Work

25

Conclusion and Future Work

- Cluster OpenMP brings OpenMP onto a cluster
 - Takes advantage of relaxed consistency model
 - OpenMP primitives become significantly more expensive
- Intel Cluster OpenMP
 - Proved to be successful for several small applications
 - Tuning work required for large applications
 - A fast network (IB) is crucial for application performance
 - Problems with C++ programs employing the STL
- Future Work
 - Tuning strategies
 - Multi-level parallelism (PThreads, Threading Building Blocks)
 - Scalability on large cluster (100+ nodes)

26

End

Thank you
for your
attention!

27