

Message-Passing and Hybrid Parallelization on Clusters of Multi-Core SMP Nodes

Future Directions of the MPI-3 Standardization

Rolf Rabenseifner

Rabenseifner@hlrs.de

High Performance Computing Center (HLRS),
University of Stuttgart, Germany
www.hlrs.de

Séminaires de l'IDRIS
Paris, April 2, 2009



Hybrid Parallelization on Clusters of Multi-Core
Slide 1 Höchstleistungsrechenzentrum Stuttgart

H L R I S

Aspects & Outline

- **Future High Performance Computing (HPC)**
 - always hierarchical hardware design
- **Mismatches and opportunities with current MPI based programming models**
 - Some new features are needed
 - Some optimizations can be done best by the application itself
- **Optimization always requires knowledge on the hardware:**
 - Qualitative and quantitative information is needed
 - through a standardized interface
- **Future Directions of the MPI-3 Standardization**
 - MPI-2.1 / MPI-1 + MPI-2 in one book —
 - MPI-2.2 / small enhancements
 - MPI-3.0 / new challenges



Hybrid Parallelization on Clusters of Multi-Core
Slide 2 / 61 Rolf Rabenseifner

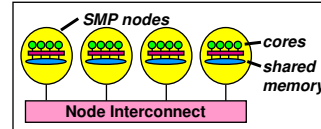
H L R I S

Future High Performance Computing (HPC) → always hierarchical hardware design

- Efficient programming of clusters of SMP nodes

SMP nodes:

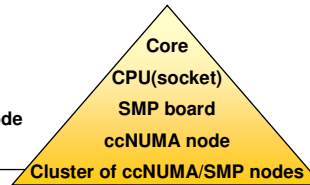
- Dual/multi core CPUs
- Multi CPU shared memory
- Multi CPU ccNUMA
- Any mixture with shared memory programming model



- Hardware range

- mini-cluster with dual-core CPUs
- ...
- large constellations with large SMP nodes
- ... with several sockets (CPUs) per SMP node
- ... with several cores per socket

→ Hierarchical system layout



- Hybrid MPI/OpenMP programming seems natural

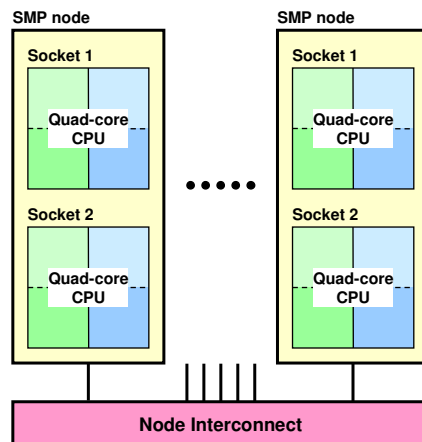
- MPI between the nodes
- OpenMP inside of each SMP node



Hybrid Parallelization on Clusters of Multi-Core
Slide 3 / 61
Rolf Rabenseifner

HLRS

Which is best programming model?



- Which programming model is fastest?

- MPI everywhere?



- Fully hybrid MPI & OpenMP?



- Something between? (Mixed model)



- Often hybrid programming **slower** than pure MPI
- Examples, Reasons, ...

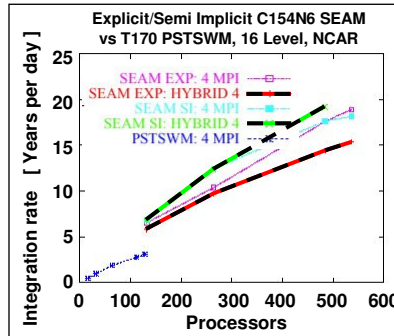
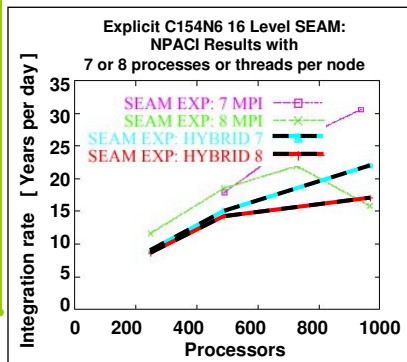


Hybrid Parallelization on Clusters of Multi-Core
Slide 4 / 61
Rolf Rabenseifner

HLRS

Example from SC

- Pure MPI versus Hybrid MPI+OpenMP (Masteronly)
- What's better?
→ it depends on?



Figures: Richard D. Loft, Stephen J. Thomas, John M. Dennis:
Terascale Spectral Element Dynamical Core for Atmospheric General Circulation Models.
Proceedings of SC2001, Denver, USA, Nov. 2001.
<http://www.sc2001.org/papers/pap.pap189.pdf>
Fig. 9 and 10.



Hybrid Parallelization on Clusters of Multi-Core
Slide 5 / 61
Rolf Rabenseifner

H L R I S

Goals

Minimizing

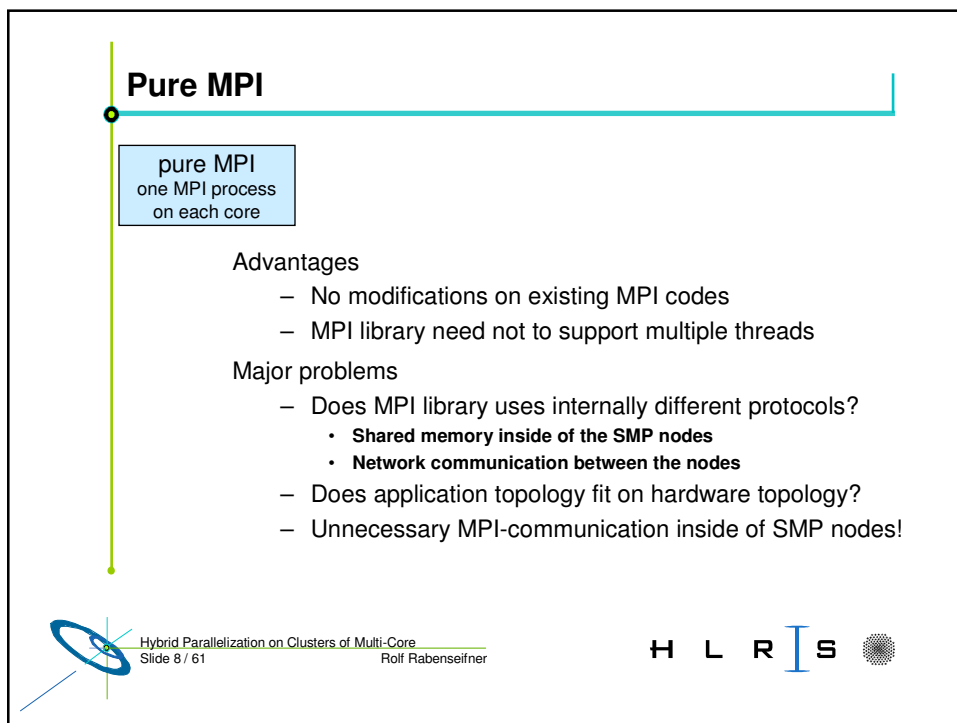
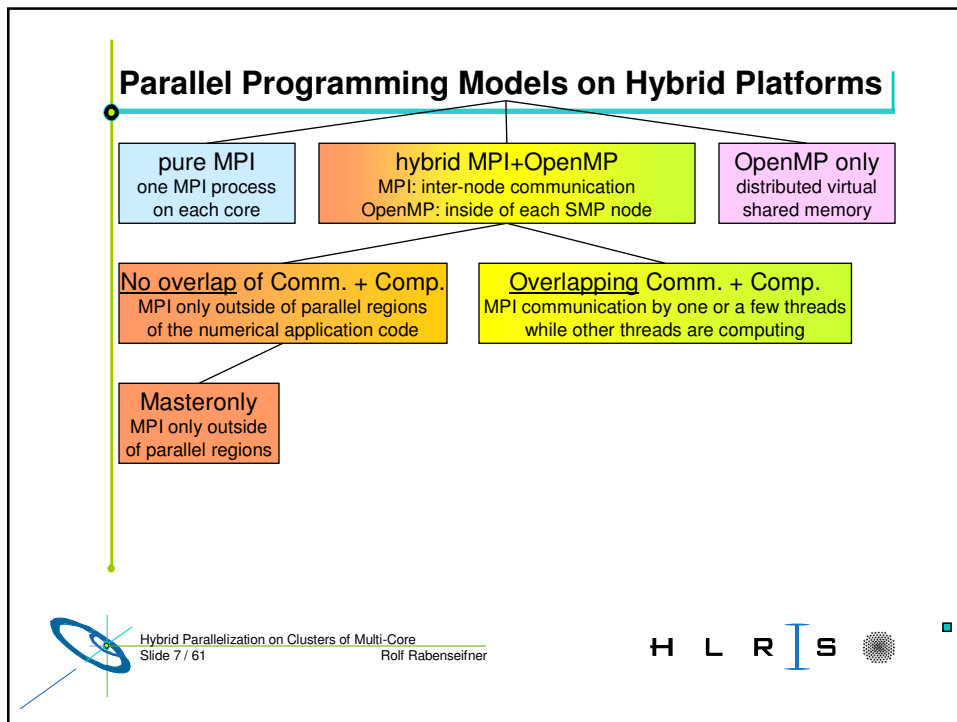
- Communication overhead,
 - e.g., messages inside of one SMP node
- Synchronization overhead
 - e.g., OpenMP fork/join
- Load imbalance
 - e.g., using OpenMP *guided* worksharing schedule
- Memory consumption
 - e.g., replicated data in MPI parallelization
- Computation overhead
 - e.g., duplicated calculations in MPI parallelization

Optimal
parallel
scaling



Hybrid Parallelization on Clusters of Multi-Core
Slide 6 / 61
Rolf Rabenseifner

H L R I S



Hybrid Masteronly

Masteronly
MPI only outside
of parallel regions

Advantages

- No message passing inside of the SMP nodes
- No topology problem

```
for (iteration ....)
{
    #pragma omp parallel
    numerical code
    /*end omp parallel */

    /* on master thread only */
    MPI_Send (original data
             to halo areas
             in other SMP nodes)
    MPI_Recv (halo data
             from the neighbors)
} /*end for loop
```

Major Problems

- All other threads are sleeping while master thread communicates!
- Which inter-node bandwidth?
- MPI-lib must support at least MPI_THREAD_FUNNELED



Hybrid Parallelization on Clusters of Multi-Core
Slide 9 / 61
Rolf Rabenseifner

H L R I S

Overlapping Communication and Computation

MPI communication by one or a few threads while other threads are computing

```
if (my_thread_rank < ...) {
    MPI_Send/Recv....
    i.e., communicate all halo data
} else {
    Execute those parts of the application
    that do not need halo data
    (on non-communicating threads)
}

Execute those parts of the application
that need halo data
(on all threads)
```



Hybrid Parallelization on Clusters of Multi-Core
Slide 10 / 61
Rolf Rabenseifner

H L R I S

Pure OpenMP (on the cluster)

OpenMP only
distributed virtual
shared memory

- Distributed shared virtual memory system needed
- Must support clusters of SMP nodes
- e.g., Intel® Cluster OpenMP
 - Shared memory parallel inside of SMP nodes
 - Communication of modified parts of pages at OpenMP flush (part of each OpenMP barrier)

by rule of thumb:
**Communication
may be
10 times slower
than with MPI**
→ Appendix

i.e., the OpenMP memory and parallelization model
is prepared for clusters!

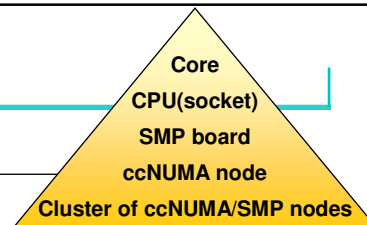


Hybrid Parallelization on Clusters of Multi-Core
Slide 11 / 61
Rolf Rabenseifner

HLRS

Mismatch Problems

- None of the programming models fits to the hierarchical hardware (cluster of SMP nodes)
- Several mismatch problems
 - following slides
- Benefit through hybrid programming
 - opportunities, see next section
- Quantitative implications
 - depends on you application



• **Less frustration and**
• **More success**
with your parallel program
on clusters of SMP
nodes

Examples:	No.1	No.2
Benefit through hybrid (see next section)	30%	10%
Loss by mismatch problems	-10%	-25%
Total	+20%	-15%

In most cases:
**Both
categories!**



Hybrid Parallelization on Clusters of Multi-Core
Slide 12 / 61
Rolf Rabenseifner

HLRS

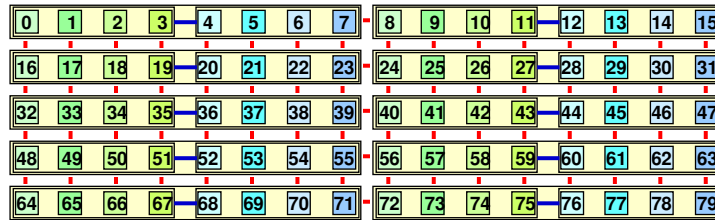
The Topology Problem with

pure MPI

one MPI process
on each core

Application example on 80 cores:

- Cartesian application with $5 \times 16 = 80$ sub-domains
- On system with 10 x dual socket x quad-core



- + 17 x inter-node connections per node
- 1 x inter-socket connection per node

Sequential ranking of
MPI_COMM_WORLD

Does it matter?



Hybrid Parallelization on Clusters of Multi-Core
Slide 13 / 61

Rolf Rabenseifner

HLRS

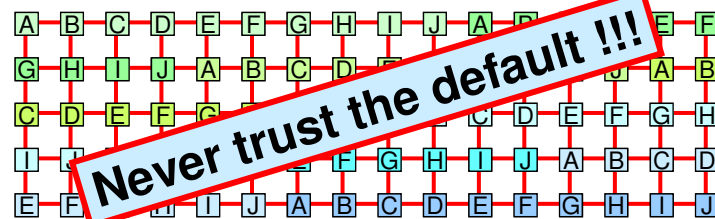
The Topology Problem with

pure MPI

one MPI process
on each core

Application example on 80 cores:

- Cartesian application with $5 \times 16 = 80$ sub-domains
- On system with 10 x dual socket x quad-core



- + 32 x inter-node connections per node
- 0 x inter-socket connection per node

Round robin ranking of
MPI_COMM_WORLD



Hybrid Parallelization on Clusters of Multi-Core
Slide 14 / 61

Rolf Rabenseifner

HLRS

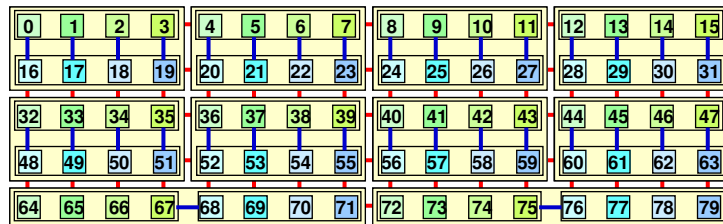
The Topology Problem with

pure MPI

one MPI process
on each core

Application example on 80 cores:

- Cartesian application with $5 \times 16 = 80$ sub-domains
- On system with 10 x dual socket x quad-core



+ 10 x inter-node connections per node

+ 4 x inter-socket connection per node

Two levels of
domain decomposition

Bad affinity of cores to thread ranks



Hybrid Parallelization on Clusters of Multi-Core
Slide 15 / 61

Rolf Rabenseifner

H L R I S

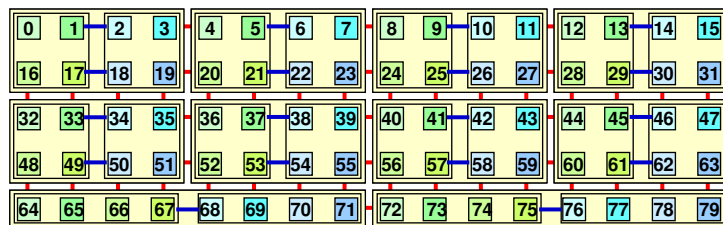
The Topology Problem with

pure MPI

one MPI process
on each core

Application example on 80 cores:

- Cartesian application with $5 \times 16 = 80$ sub-domains
- On system with 10 x dual socket x quad-core



+ 10 x inter-node connections per node

+ 2 x inter-socket connection per node

Two levels of
domain decomposition

Good affinity of cores to thread ranks



Hybrid Parallelization on Clusters of Multi-Core
Slide 16 / 61

Rolf Rabenseifner

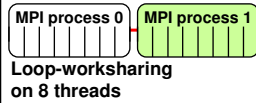
H L R I S

The Topology Problem with hybrid MPI+OpenMP

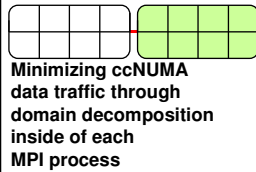
MPI: inter-node communication
OpenMP: inside of each SMP node

Exa.: 2 SMP nodes, 8 cores/node

Optimal ?



Optimal ?



Problem

- Does application topology inside of SMP parallelization fit on inner hardware topology of each SMP node?

Solutions:

- Domain decomposition inside of each thread-parallel MPI process, and
- first touch strategy with OpenMP

Successful examples:

- Multi-Zone NAS Parallel Benchmarks (MZ-NPB)



Hybrid Parallelization on Clusters of Multi-Core
Slide 17 / 61
Rolf Rabenseifner

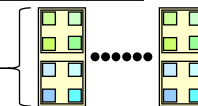
HLRS

The Topology Problem with hybrid MPI+OpenMP

MPI: inter-node communication
OpenMP: inside of each SMP node

Application example:

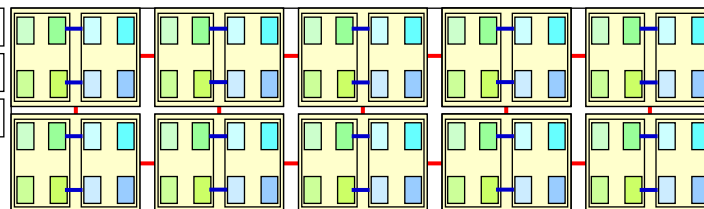
- Same Cartesian application aspect ratio: 5 x 16
- On system with 10 x dual socket x quad-core
- 2 x 5 domain decomposition



Application

MPI Level

OpenMP

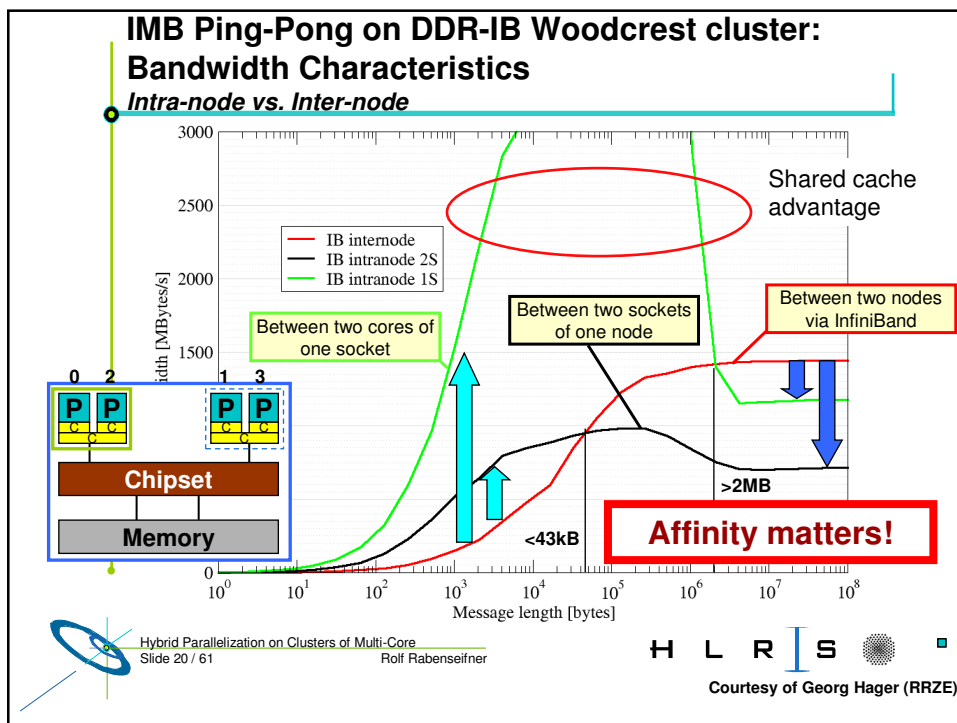
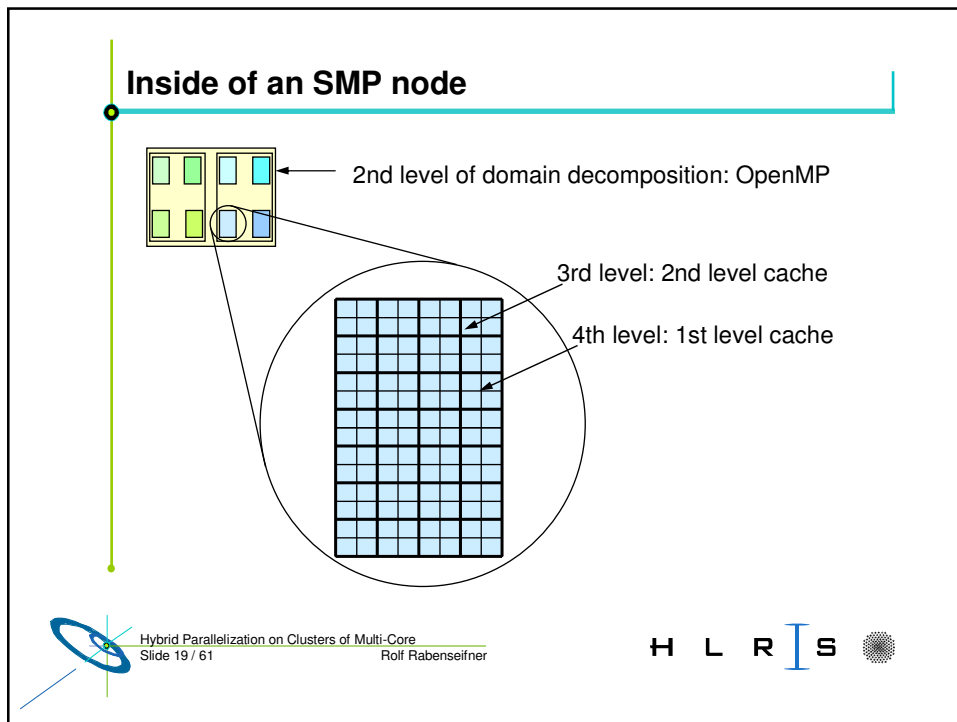


- + 3 x inter-node connections per node, but ~ 4 x more traffic
- + 2 x inter-socket connection per node



Hybrid Parallelization on
Slide 18 / 61

Affinity of cores to thread ranks !!!



The Mapping Problem with mixed model

pure MPI
 &
 hybrid MPI+OpenMP

Do we have this?

... or that?

Several multi-threaded MPI process per SMP node:

Problem

- Where are your processes and threads really located?

Solutions:

- Depends on your platform,
- e.g., lbrun **numactl** option on Sun

→ case-study on Sun Constellation Cluster Ranger with BT-MZ and SP-MZ

Hybrid Parallelization on Clusters of Multi-Core

Slide 21 / 61

Rolf Rabenseifner

HLRS

Unnecessary intra-node communication

pure MPI
 Mixed model
 (several multi-threaded MPI processes per SMP node)

Problem:

- If several MPI process on each SMP node
→ unnecessary intra-node communication

Solution:

- Only one MPI process per SMP node

Remarks:

- MPI library must use appropriate fabrics / protocol for intra-node communication
- Intra-node bandwidth higher than inter-node bandwidth
→ problem may be small
- MPI implementation may cause unnecessary data copying
→ waste of memory bandwidth

}

Quality aspects of the MPI library

Hybrid Parallelization on Clusters of Multi-Core

Slide 22 / 61

Rolf Rabenseifner

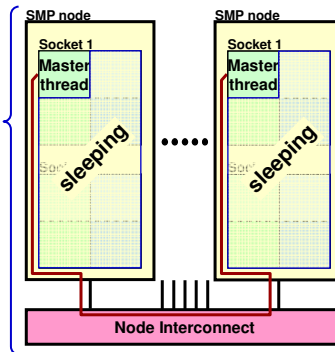
HLRS

Sleeping threads and network saturation with Masteronly

MPI only outside of
parallel regions

```
for (iteration ....)
{
    #pragma omp parallel
    numerical code
    /*end omp parallel */

    /* on master thread only */
    MPI_Send (original data
to halo areas
in other SMP nodes)
    MPI_Recv (halo data
from the neighbors)
} /*end for loop
```



Problem 1:

- Can the master thread saturate the network?

Solution:

- If not, use mixed model
- i.e., several MPI processes per SMP node

Problem 2:

- Sleeping threads are wasting CPU time

Solution:

- Overlapping of computation and communication

Problem 1&2 together:

- Producing more idle time through lousy bandwidth of master thread



Hybrid Parallelization on Clusters of Multi-Core
Slide 23 / 61

Rolf Rabenseifner

HLRS

OpenMP: Additional Overhead & Pitfalls

- Using OpenMP
 - may prohibit compiler optimization
 - may cause significant loss of computational performance
- Thread fork / join
- On ccNUMA SMP nodes:
 - E.g. in the masteronly scheme:
 - One thread produces data
 - Master thread sends the data with MPI
 - data may be internally communicated from one memory to the other one
- Amdahl's law for each level of parallelism
- Using MPI-parallel application libraries?
 - Are they prepared for hybrid?



Hybrid Parallelization on Clusters of Multi-Core
Slide 24 / 61

Rolf Rabenseifner

HLRS

Overlapping Communication and Computation

MPI communication by one or a few threads while other threads are computing

Three problems:

- the application problem:
 - one must separate application into:
 - code that can run before the halo data is received
 - code that needs halo data

→ very hard to do !!!

- the thread-rank problem:
 - comm. / comp. via thread-rank
 - cannot use work-sharing directives

→ loss of major OpenMP support (see next slide)

- the load balancing problem

```
if (my_thread_rank < 1) {  
    MPI_Send/Recv....  
} else {  
    my_range = (high-low-1) / (num_threads-1) + 1;  
    my_low = low + (my_thread_rank+1)*my_range;  
    my_high=high+ (my_thread_rank+1)*my_range;  
    my_high = max(high, my_high)  
    for (i=my_low; i<my_high; i++) {  
        ....  
    }  
}
```



Hybrid Parallelization on Clusters of Multi-Core
Slide 25 / 61
Rolf Rabenseifner

H L R I S

Overlapping Communication and Computation

MPI communication by one or a few threads while other threads are computing

Subteams

- Important proposal for OpenMP 3.x or OpenMP 4.x

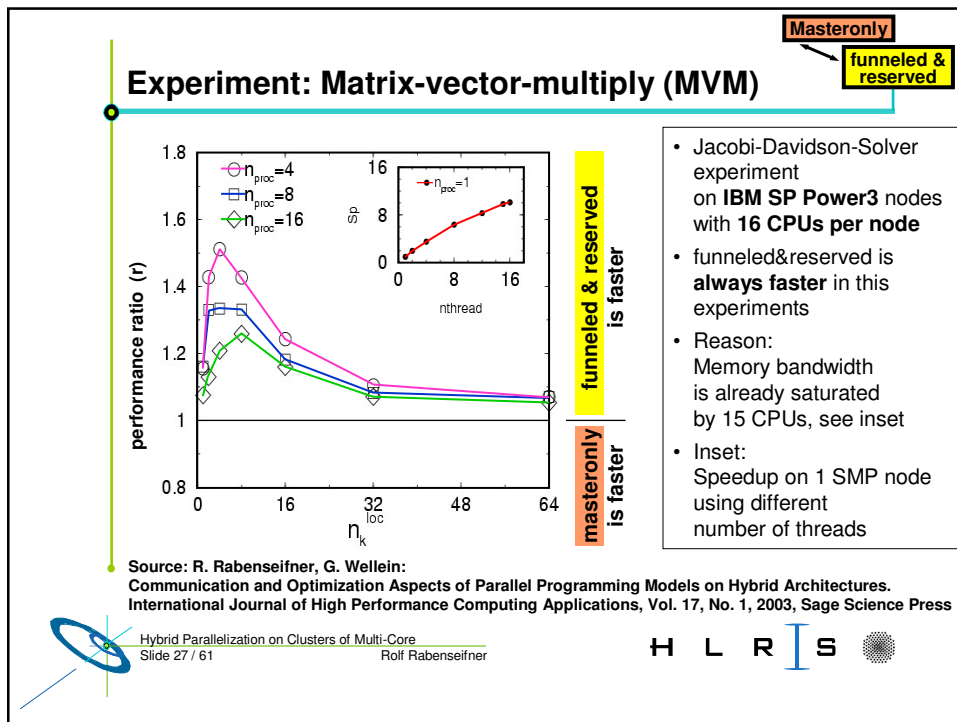
Barbara Chapman et al.:
Toward Enhancing OpenMP's
Work-Sharing Directives.
In proceedings, W.E. Nagel et
al. (Eds.): Euro-Par 2006,
LNCS 4128, pp. 645-654,
2006.

```
#pragma omp parallel  
{  
    #pragma omp single onthreads( 0 )  
    {  
        MPI_Send/Recv....  
    }  
    #pragma omp for onthreads( 1 : omp_get_numthreads()-1 )  
    for (.....)  
    { /* work without halo information */  
    } /* barrier at the end is only inside of the subteam */  
    ...  
    #pragma omp barrier  
    #pragma omp for  
    for (.....)  
    { /* work based on halo information */  
    }  
} /*end omp parallel */
```



Hybrid Parallelization on Clusters of Multi-Core
Slide 26 / 61
Rolf Rabenseifner

H L R I S



No silver bullet

- The analyzed programming models do **not** fit on hybrid architectures
 - whether drawbacks are minor or major
 - **depends on applications' needs**
 - But there are major opportunities → next section
- In the NPB-MZ case-studies
 - We tried to use optimal parallel environment
 - for pure MPI
 - for hybrid MPI+OpenMP
 - i.e., the developers of the MZ codes and we tried to minimize the mismatch problems
 - the opportunities in next section dominated the comparisons

Hybrid Parallelization on Clusters of Multi-Core
Slide 28 / 61
Rolf Rabenseifner

H L R I S

Aspects & Outline

- Future High Performance Computing (HPC)
 - always hierarchical hardware design
- **Mismatches and opportunities with current MPI based programming models**
 - **Some new features are needed** → e.g., OpenMP subteams
 - **Some optimizations can be done best by the application itself**
- Optimization always requires knowledge on the hardware:
 - Qualitative and quantitative information is needed
 - through a standardized interface
- Future Directions of the MPI-3 Standardization
 - MPI-2.1 / MPI-1 + MPI-2 in one book — 
 - MPI-2.2 / small enhancements
 - MPI-3.0 / new challenges



Opportunities of hybrid parallelization (MPI & OpenMP)

Overview

- Nested Parallelism
 - Outer loop with MPI / inner loop with OpenMP
- Load-Balancing
 - Using OpenMP **dynamic** and **guided** worksharing
- Memory consumption
 - Significantly reduction of replicated data on MPI level
- Opportunities, if MPI speedup is limited due to “algorithmic” problems
 - Significantly reduced number of MPI processes
- ... (→ slide on “Further Opportunities”)



Hybrid Parallelization on Clusters of Multi-Core
Slide 30 / 61
Rolf Rabenseifner

HLRS 

Nested Parallelism

- Example NPB: BT-MZ (Block tridiagonal simulated CFD application)
 - Outer loop:
 - **limited number of zones** → **limited parallelism**
 - **zones with different workload** → **speedup** < $\frac{\text{Sum of workload of all zones}}{\text{Max workload of one zone}}$
 - Inner loop:
 - **OpenMP parallelized (static schedule)**
 - **Not suitable for distributed memory parallelization**
- Principles:
 - Limited parallelism on outer level
 - Additional inner level of parallelism
 - Inner level not suitable for MPI
 - Inner level may be suitable for static OpenMP worksharing



Hybrid Parallelization on Clusters of Multi-Core
Slide 31 / 61
Rolf Rabenseifner

HLRS

Benchmark Characteristics

- Aggregate sizes and zones:
 - Class B: 304 x 208 x 17 grid points, 64 zones
 - Class C: 480 x 320 x 28 grid points, 256 zones
 - Class D: 1632 x 1216 x 34 grid points, 1024 zones
 - Class E: 4224 x 3456 x 92 grid points, 4096 zones
- **BT-MZ:**
Block tridiagonal simulated CFD application
 - Size of the zones varies widely:
 - large/small about 20
 - requires multi-level parallelism to achieve a good load-balance
- **SP-MZ:**
Scalar Pentadiagonal simulated CFD application
 - Size of zones identical
 - no load-balancing required

Expectations:

Pure MPI:
Load-balancing
problems!

Good candidate
for
MPI+OpenMP

Load-balanced on
MPI level:
Pure MPI should
perform best



Hybrid Parallelization on Clusters of Multi-Core
Slide 32 / 61
Rolf Rabenseifner

Courtesy of Gabriele Jost (TACC/NPS)

HLRS

Sun Constellation Cluster Ranger (1)

- Located at the Texas Advanced Computing Center (TACC), University of Texas at Austin (<http://www.tacc.utexas.edu>)
- 3936 Sun Blades, 4 AMD Quad-core 64bit 2.3GHz processors per node (blade), 62976 cores total
- 123TB aggregate memory
- Peak Performance 579 Tflops
- InfiniBand Switch interconnect
- Sun Blade x6420 Compute Node:
 - 4 Sockets per node
 - 4 cores per socket
 - HyperTransport System Bus
 - 32GB memory

Courtesy of Gabriele Jost (TACC/NPS)

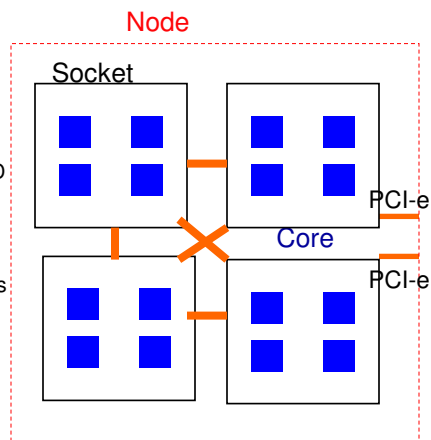
HLRS



Hybrid Parallelization on Clusters of Multi-Core
Slide 33 / 61
Rolf Rabenseifner

Sun Constellation Cluster Ranger (2)

- **Compilation:**
 - PGI pgf90 7.1
 - mpif90 -tp barcelona-64 -r8
- **Cache optimized benchmarks**
- **Execution:**
 - MPI MVAPICH
 - setenv OMP_NUM_THREADS NTHREAD
 - lbrun numactl bt-mz.exe
- **numactl controls**
 - Socket affinity: select sockets to run
 - Core affinity: select cores within socket
 - Memory policy: where to allocate memory
 - <http://www.halobates.de/numaapi3.pdf>



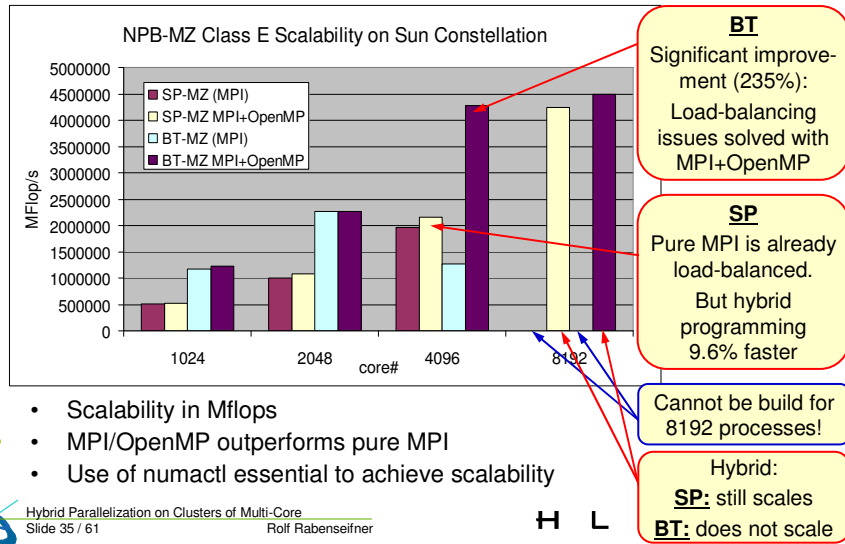
Courtesy of Gabriele Jost (TACC/NPS)

HLRS



Hybrid Parallelization on Clusters of Multi-Core
Slide 34 / 61
Rolf Rabenseifner

NPB-MZ Class E Scalability on Ranger



Next opportunity: Load-Balancing (on same or different level of parallelism)

- OpenMP enables
 - Cheap **dynamic** and **guided** load-balancing
 - Just a parallelization option (clause on omp for / do directive)
 - Without additional software effort
 - Without explicit data movement
- On MPI level
 - Dynamic load balancing** requires moving of parts of the data structure through the network
 - Significant runtime overhead
 - Complicated software / therefore not implemented
- MPI & OpenMP**
 - Simple static load-balancing on MPI level, } **medium quality**
dynamic or guided on OpenMP level } **cheap implementation**

Hybrid Parallelization on Clusters of Multi-Core
Slide 36 / 61
Rolf Rabenseifner

H L R I S

Memory consumption

- Shared nothing
 - Heroic theory
 - In practice: Some data is duplicated
- **MPI & OpenMP**
With n threads per MPI process:
 - Duplicated data is reduced by factor n



Hybrid Parallelization on Clusters of Multi-Core
Slide 37 / 61
Rolf Rabenseifner

H L R I S

Memory consumption (continued)

- Future:
With 100+ cores per chip the memory per core is limited.
 - Data reduction through usage of shared memory may be a key issue
 - Domain decomposition on each hardware level
 - **Maximizes**
 - Data locality
 - Cache reuse
 - **Minimizes**
 - CCnuma accesses
 - Message passing
 - No halos between domains inside of SMP node
 - **Minimizes**
 - Memory consumption



Hybrid Parallelization on Clusters of Multi-Core
Slide 38 / 61
Rolf Rabenseifner

H L R I S

How many multi-threaded MPI processes per SMP node

- SMP node = 1 Socket
 - 1 MPI process per SMP node
- SMP node is n -Socket ccNUMA node
 - With x NICs (network interface cards) per node
- How many MPI processes per SMP node are optimal?
 - somewhere between 1 and n

In other words:

- How many threads (i.e., cores) per MPI process?
 - Many threads
 - overlapping of MPI and computation may be necessary,
 - some NICs unused?
 - Too few threads
 - too much memory consumption (see previous slides)



Hybrid Parallelization on Clusters of Multi-Core
Slide 39 / 61
Rolf Rabenseifner

H L R I S

Opportunities, if MPI speedup is limited due to “algorithmic” problems

- Algorithmic opportunities due to larger physical domains inside of each MPI process
 - If multigrid algorithm only inside of MPI processes
 - If separate preconditioning inside of MPI nodes and between MPI nodes
 - If MPI domain decomposition is based on physical zones



Hybrid Parallelization on Clusters of Multi-Core
Slide 40 / 61
Rolf Rabenseifner

H L R I S

Further Opportunities


- Reduced number of MPI messages, reduced aggregated message size } compared to pure MPI
- Functional parallelism
→ e.g., I/O in an other thread
- MPI shared memory fabrics not loaded if whole SMP node is parallelized with OpenMP



Hybrid Parallelization on Clusters of Multi-Core
Slide 41 / 61
Rolf Rabenseifner

H L R I S

Aspects & Outline

- Future High Performance Computing (HPC)
→ always hierarchical hardware design
- Mismatches and opportunities with current MPI based programming models
→ Some new features are needed → e.g., OpenMP subteams
→ Some optimizations can be done best by the application itself
- **Optimization always requires knowledge on the hardware:**
→ **Qualitative and quantitative information is needed**
→ **through a standardized interface**
- Future Directions of the MPI-3 Standardization
→ MPI-2.1 / MPI-1 + MPI-2 in one book — 
→ MPI-2.2 / small enhancements
→ MPI-3.0 / new challenges



Which hardware topology information

- Structure of the cluster and memory hierarchy
- Data exchange „speed“
(e.g., transmission time for a given data size)



Hybrid Parallelization on Clusters of Multi-Core
Slide 43 / 61
Rolf Rabenseifner

H L R I S

Where to get this information

- Currently, this information is accessible through different interfaces
 - E.g., numalib / numctl
 - Linux processor information
 - ...
- Most information must be measured by the application



Hybrid Parallelization on Clusters of Multi-Core
Slide 44 / 61
Rolf Rabenseifner

H L R I S

What is needed

- A standardized interface
 - Independent of the operating system

Similar to the beginning of the MPI standardization:

Where to get wall-clock-time with

- high accuracy,
- little overhead

What about quantitative information?

- The affinity slide has clearly shown, that this is needed
- Can “benchmark data” be returned by a standardized library?

Yes, it can!

MPI_Wtick is such an information.
It is returned by MPI since days of MPI-1!

Let's do it in MPI-3

- Contribution by the MPI community are welcome!

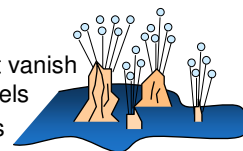


Hybrid Parallelization on Clusters of Multi-Core
Slide 45 / 61
Rolf Rabenseifner

H L R I S

I didn't mention ...


- Other parallelization models:
 - Partitioned Global Address Space (PGAS) languages (Unified Parallel C (UPC), Co-array Fortran (CAF), Chapel, Fortress, Titanium, and X10).
 - High Performance Fortran (HPF)
- Many rocks in the cluster-of-SMP-sea do not vanish into thin air by using new parallelization models
- Area of interesting research in the next years



Hybrid Parallelization on Clusters of Multi-Core
Slide 46 / 61
Rolf Rabenseifner

H L R I S

Aspects & Outline

- Future High Performance Computing (HPC)
 - always hierarchical hardware design
- Mismatches and opportunities with current MPI based programming models
 - Some new features are needed
 - Some optimizations can be done best by the application itself
- Optimization always requires knowledge on the hardware:
 - Qualitative and quantitative information is needed
 - through a standardized interface
- **Future Directions of the MPI-3 Standardization**
 - MPI-2.1 / MPI-1 + MPI-2 in one book — 
 - MPI-2.2 / small enhancements
 - MPI-3.0 / new challenges



Hybrid Parallelization on Clusters of Multi-Core
Slide 47 / 61
Rolf Rabenseifner

H L R I S 

MPI-3 Forum

- MPI-2.1 (done): Merging MPI-1.1 and MPI-2.0 to one book
 - MPI-2.2: Small additions (2009)
 - MPI-3.0: Major new features (2010/2011), e.g.,
 - **Non-blocking collectives** (→ overlap of computation and communication)
 - **Fault-tolerant MPI**
 - **New efficient remote memory access interface**
 - **Fortran interface with argument checking**
 - **Tools support**
 - **Hybrid MPI&OpenMP programming**
 - If you have interest / ideas / ...
 - **please contact one of the members of the MPI Forum**
 - They represent
 - **Industry**
 - **Academics**
 - **Labs**
- MPI users and developers
from USA, Europe, and Asia



Hybrid Parallelization on Clusters of Multi-Core
Slide 48 / 61
Rolf Rabenseifner

H L R I S 

Active Members

Argonne NL, Bull, Cisco,
Cray, Fujitsu,
HDF Group, HLRS,
HP, IBM, INRIA,
Indiana U., Intel,
Lawrence Berkeley NL,
Livermore NL, Los Alamos NL,
Mathworks, Microsoft,
NCSA/UIUC, NEC,
Oak Ridge NL, Ohio State U.,

Pacific NW NL,
Qlogic, Sandia NL,
SiCortex, Sun Microsystems,
Tokyo Institute of Technology,
U. Alabama Birmingham,
U. Houston,
U. Tennessee Knoxville,
U. Tokyo

- Travel support provided by the National Science Foundation

Courtesy of Richard Graham (ORNL)
Chairman and Convener of the MPI-3 Forum



Hybrid Parallelization on Clusters of Multi-Core
Slide 49 / 61
Rolf Rabenseifner

Structure

- Chairman and Convener: Rich Graham (ORNL)
- Secretary: Jeff Squyres (CISCO)
- Steering committee:
 - Jack Dongarra
 - Al Geist
 - Rich Graham
 - Bill Gropp
 - Andrew Lumsdaine
 - Rusty Lusk
 - Rolf Rabenseifner

Courtesy of Richard Graham (ORNL)
Chairman and Convener of the MPI-3 Forum



Hybrid Parallelization on Clusters of Multi-Core
Slide 50 / 61
Rolf Rabenseifner

Implementation of MPI-2.1

Library	Yes mm/yy	asked	Library	Yes mm/yy	asked
MPICH2	YES	Rajeev Thakur	Bull	Yes	Yann Kalemkarian
Open MPI (in V1.3)	YES	Jeff Squyres	HP	12/ 2008	Kannan Narasimhan
IBM	12/ 2008	Dick Treumann	Cray	2009	Mark Pagel
Sun Microsystems	YES (OpenMPI)	Rolf Vandevoort	NEC	YES	Hubert Ritzdorf, Jesper Traeff
Microsoft	12/ 2008	Erez Haba	Intel	2009	Alexander Supalov, Keith Underwood
Fujitsu	6/ 2009	Sueyasu, Naoki	SiCortex	Q1/ 2009	David Gingold



Hybrid Parallelization on Clusters of Multi-Core
Slide 51 / 61
Rolf Rabenseifner

H L R I S

MPI-2.1 – a Statistical Summary

- MPI 1.3 + 2.0: merging 335 distinguished parts
- Ballot 1+2: 40 items implemented
- Ballot 3: 14 items / 12 accepted + implemented
- Ballot 4: 22 items / 20 accepted + implemented
- Reviews: 814 items
- Total summary: **1225 items**
- 14 chapter authors / ~ 55 MPI-Forum members involved



Hybrid Parallelization on Clusters of Multi-Core
Slide 52 / 61
Rolf Rabenseifner

H L R I S

MPI 2.2 – Scope

Scope:

- Small changes to the standard.
- A small change is defined as one
 - that does not break existing user code –
 - either by interface changes or semantic changes
 - and does not require large implementation changes.



Hybrid Parallelization on Clusters of Multi-Core
Slide 53 / 61
Rolf Rabenseifner

H L R I S

MPI-2.2 – Current State (Apr. 1, 2009)

- <https://svn.mpi-forum.org/trac/mpi-forum-web/wiki/mpi22/TicketWorkflow>
 - <https://svn.mpi-forum.org/trac/mpi-forum-web/query>
 - Priorities:
 - 23 Withdrawn/closed
 - 18 Not ready / author rework
 - 13 Forum feedback requested
 - 27 Waiting for reviews
 - 10 Reviewed
 - 41 Had 1st reading
 - 4 Had 1st vote
 - 8 Passed
 - Total: 144 tickets of 24 owners
 - Average: 6 tickets/owner / Maximum: 25 tickets/owner
- } ~ 30 tickets: Candidates for MPI-3.0
- } ~ 90 tickets: May go into MPI-2.2



Hybrid Parallelization on Clusters of Multi-Core
Slide 54 / 61
Rolf Rabenseifner

H L R I S

MPI-2.2 - Some Examples

- New scaling interfaces for
 - Sub-communicator creation
 - Graph topologies
- Non-blocking send-buffer restriction erased
- New collective routines
 - Regular reduce-scatter
 - Additional predefined datatypes for reduction operations
- Add const keyword to C bindings
- Convenience function: MPI_CART_SHIFT_VECTOR
- More items of type
 - corrections / clarifications of the current text



Hybrid Parallelization on Clusters of Multi-Core
Slide 55 / 61
Rolf Rabenseifner

H L R I S

MPI 3.0 - Scope

- **Additions to the standard that are needed for better platform and application support.**
 - These are to be consistent with MPI being a library providing of parallel process management and data exchange.
 - This includes, but is not limited to, issues associated with
 - scalability (performance and robustness),
 - multi-core support,
 - cluster support, and
 - application support.
- **Backwards compatibility is maintained –**
Routines may be deprecated



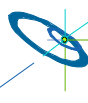
Hybrid Parallelization on Clusters of Multi-Core
Slide 56 / 61
Rolf Rabenseifner

H L R I S

Current Working Groups

- **Application Binary Interface** : Jeff Brown – LANL (Los Alamos)
– Industry consortium formed around this “associated” with the forum
- **Collective Operations** : Torsten Hoeffler, Andrew Lumsdaine - Indiana University
- **Fault Tolerance** : Richard Graham – ORNL (Oak Ridge)
- **Fortran Bindings** : Craig Rasmussen – LANL (Los Alamos)
- **Generalized Requests** : George Bosilca - The University of Tennessee
- **MPI Sub-Setting** : Alexander Supalov - Intel Corporation – on hold
- **Point-To-Point Communications** : Ron Brightwell – SNL (Sandia)
- **Remote Memory Access** : Bill Gropp, University of Illinois Champaign/Urbana - Rajeev Thakur, ANL (Argonne)
- **Tools support for MPI** – Martin Schulz and Bronis de Supinski, LLNL
- **Misc** – Adam Moody, LLNL (Livermore)
- **Hybrid Programming** – Pavan Balaji, ANL (Argonne)

Courtesy of Richard Graham (ORNL)
Chairman and Convener of the MPI-3 Forum

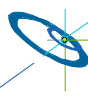


Hybrid Parallelization on Clusters of Multi-Core
Slide 57 / 61
Rolf Rabenseifner

Early Targets – ready for vote by end of 2009

- Non-blocking collectives
– Status: Waiting for reviews
- Remote Memory Access
- Consistent error notification within MPI
- Client/Server Fault Tolerance (accept/connect, spawn)
- Process Fault Tolerance

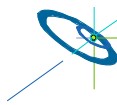
Courtesy of Richard Graham (ORNL)
Chairman and Convener of the MPI-3 Forum



Hybrid Parallelization on Clusters of Multi-Core
Slide 58 / 61
Rolf Rabenseifner

Face-To-Face Meetings 2009

- Feb 9-11, 2009 California Bay Area
- Apr 6-8, 2009 Chicago
- Jun 8-10, 2009 California Bay Area
- Jul 27-29, 2009 Chicago
- Sep 2-5, 2009 Helsinki, Finland @ EuroPVM/MPI'09
- Nov 11-13, 2009 Portland, OR @ SC09
- Telco in between
- Short travel:
 - Meeting Monday 1 pm – Wednesday noon
 - Flights on Monday morning from Europe and everywhere in USA due to time-zone win
 - Flights back on Wednesday afternoon (or through night)
 - California → convenient for Japanese forum members



Hybrid Parallelization on Clusters of Multi-Core
Slide 59 / 61 Rolf Rabenseifner

H L R I S

Further information



- **SC08 Tutorial S02, Sunday, Nov. 16, 2008, Austin Texas.**
Alice Koniges, David Eder, Bill Gropp, Ewing (Rusty) Lusk, and Rolf Rabenseifner:
Application Supercomputing and the Many-Core Paradigm Shift.
- **SC08 Tutorial M09, Monday, Nov. 17, 2008, Austin Texas.**
Rolf Rabenseifner, Georg Hager, Gabriele Jost, and Rainer Keller:
Hybrid MPI and OpenMP Parallel Programming.
- **MPI-2.1** (June 23, 2008 – finally voted at MPI Forum meeting, Sep. 4, 2008)
 - Electronically via www.mpi-forum.org
 - As hardcover book (608 pages)
 - The book was printed by HLRS
 - As a service for the MPI community.
 - High-quality sewn binding.
 - Sold at costs – 17€ or 25\$
 - Available via www.mpi-forum.org/docs
→ www.hlrs.de/mpi/mpi21



Hybrid Parallelization on Clusters of Multi-Core
Slide 60 / 61 Rolf Rabenseifner

H L R I S

Conclusions

- Future High Performance Computing (HPC)
 - always hierarchical hardware design
- Mismatches and opportunities with current MPI based programming models
 - Some new features are needed
 - Some optimizations can be done best by the application itself

MPI + OpenMP:

- Often hard to solve the mismatch problems
- May be a significant opportunity for performance
 - (huge) amount of work

- Optimization always requires knowledge on the hardware:
 - Qualitative and quantitative information is needed
 - through a standardized interface

A new standard may assist the research community, and vice versa.

- The MPI-3 Forum tries to address those aspects
 - MPI-2.1 is only a starting point: combination of MPI-1.1 and 2.0 in one book
 - www.hlrs.de/mpi/mpi21

You may join in or you may share your ideas with the MPI Forum



Hybrid Parallelization on Clusters of Multi-Core
Slide 61 / 61
Rolf Rabenseifner

HLRS



This slides – via my publications list at www.hlrs.de/people/rabenseifner/