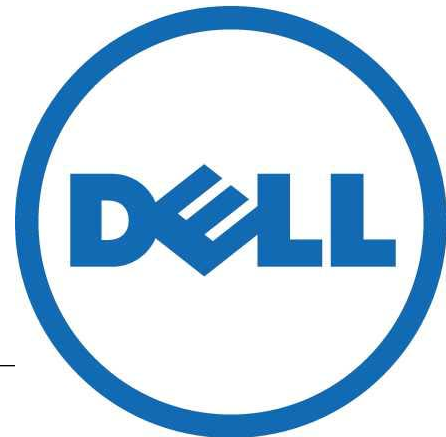


# ScaleMP Applications

at RWTH Aachen University

Christian Terboven, Dirk Schmidl, Dieter an Mey  
{terboven, schmidl, anmey}@rz.rwth-aachen.de



**a HPC case  
study with Dell**

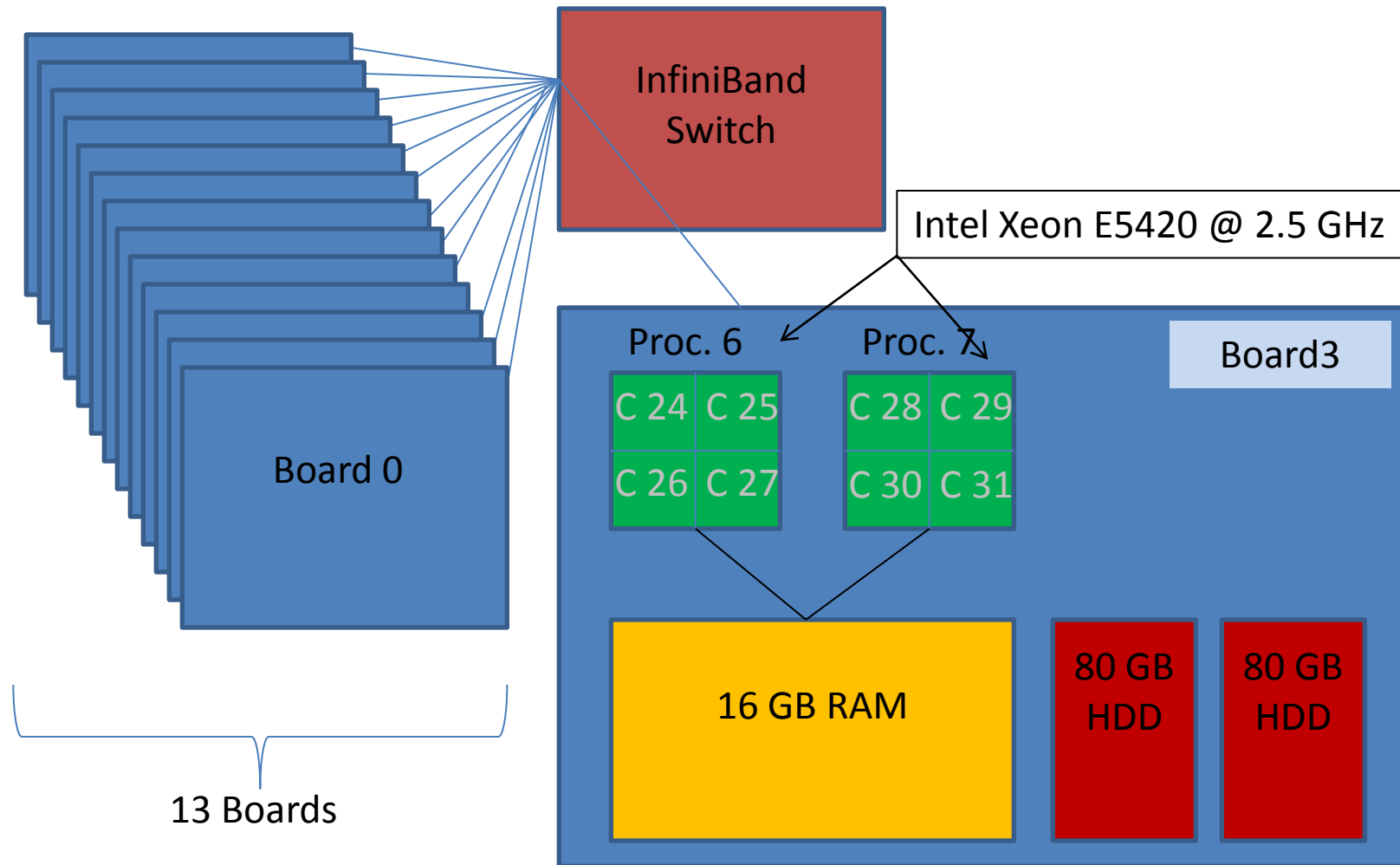
01.06.2010, ISC 2010  
Hamburg, Germany

- ▶ **The ScaleMP vSMP Architecture**
- ▶ **System Examination**
  - ▶ vSMP Utilities
  - ▶ Synthetic Benchmarks
  - ▶ Kernels
- ▶ **Applications on ScaleMP**
  - ▶ FIRE
  - ▶ SHEMAT-Suite
  - ▶ TrajSearch
- ▶ **Conclusion**



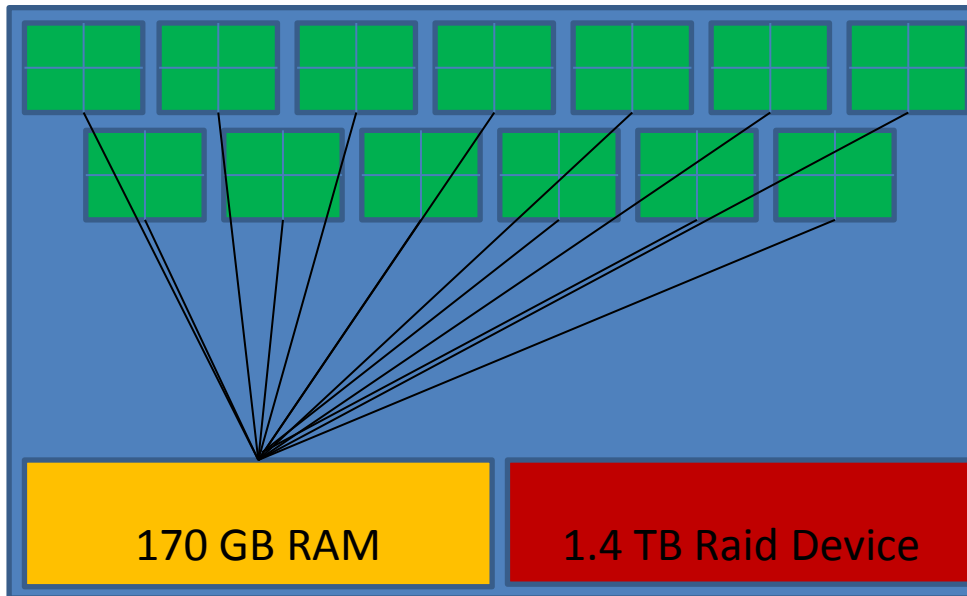
# The ScaleMP vSMP Architecture

- ▶ In total 104 cores, 170 GB of memory, 1.4 TB of disc space



- ▶ **vSMP: Cache-coherent aggregation of physical resources via the InfiniBand network (employing virtualization techniques)**

- ▶ The OS view (Single System Image):



- ▶ **Shared-Memory parallelization on a cluster of x86-based boards**

- ▶ Cheaper and more flexible than hardware-based solutions
- ▶ Strong cc-NUMA characteristics

# System Examination

# vSMP utilities: vsmpstat



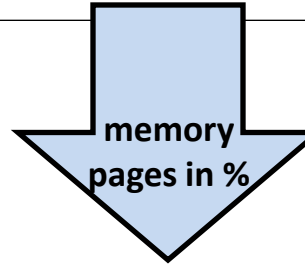
```
$ vsmpstat -bbc
```

```
[...]
```

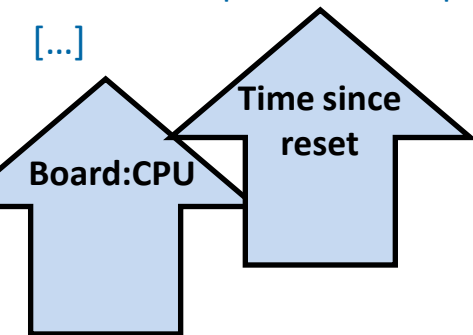
Board Basic Counters:

bbc:Bd:CPU	Time (mS)	%vSMP	TLB Flush	PT Write	PT WrtEm	HPET Resync	Drift (uS)	Frames	%Used
bbc:00:00	899802	13.1	3440358	480597	255081	0.000	0.000	4005935	40.8
bbc:00:01	899801	4.0	301449	435875	330412	0.000	0.000	4005935	40.8
bbc:00:02	899803	4.3	505126	401652	211529	0.000	0.000	4005935	40.8
bbc:00:03	899805	4.5	608419	726302	485302	0.000	0.000	4005935	40.8

```
[...]
```



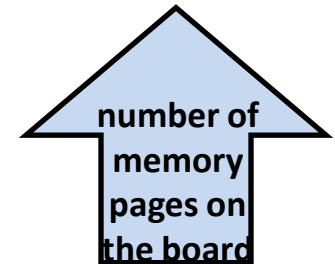
memory  
pages in %



Board:CPU

Time since  
reset

TLB  
flushed by  
vSMP



number of  
memory  
pages on  
the board

## ▶ Innovative architectures require tools to examine them properly

- ▶ ScaleMP: Memory traffic between boards
- ▶ ScaleMP: Process and thread migration + memory usage

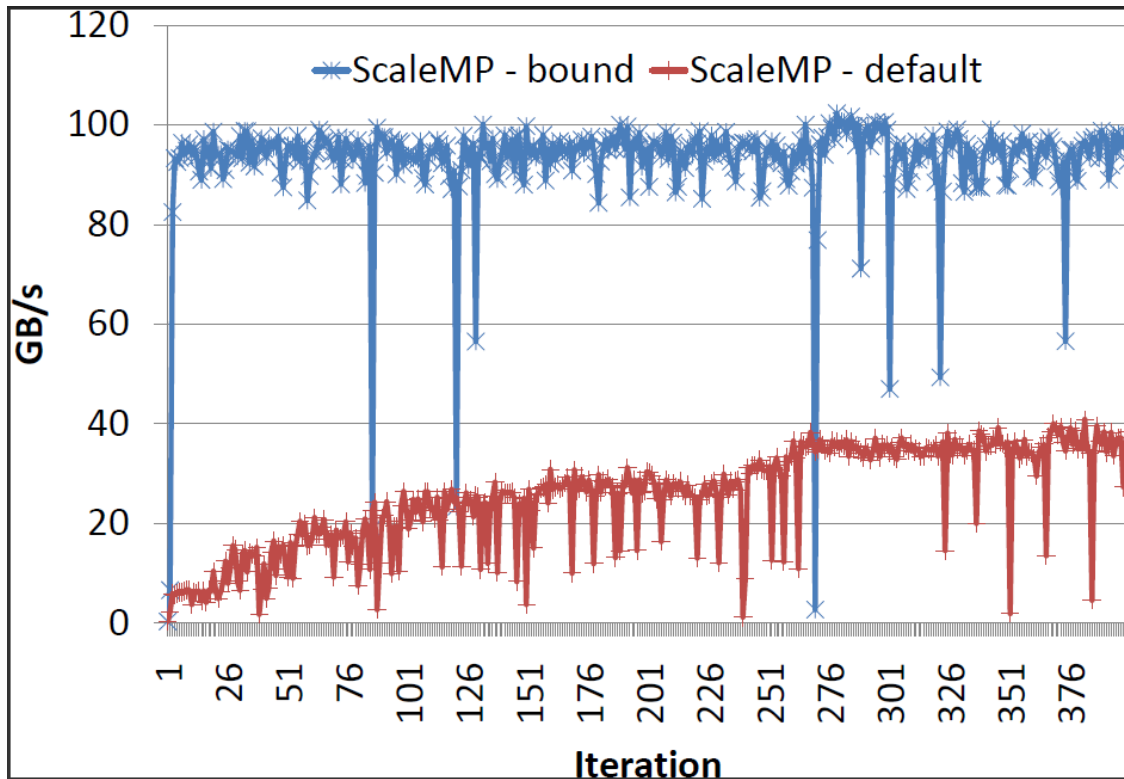
- ▶ **Distributed-Shared-Memory systems can be characterized by their „numaness“ (ration between local and remote memory acces time):**
  - ▶ *read\_from\_other*: Time to read a page that was written by another thread, both threads are on separate boards
  - ▶ *write\_from\_other*: Time to write to a page that was written by another thread before, both threads are on separate boards
  - ▶ *write\_self*: Time to write to self-written page (test for vSMP overhead)

	ScaleMP system in AC	4s Intel Tigerton machine
read_from_other	43.37	1.76
write_from_other	40.44	2.29
write_self	2.34	2.10

- ▶ Page Access Benchmark results in microseconds [usec], two threads

## ▶ OpenMP-parallelized version of STREAM w/ first-touch initialization

- ▶ Red line: 104 threads, scheduling left to the system
- ▶ Blue line: 104 threads, explicit thread binding



400 iterations, time has been measured for each one individually.

ScaleMP scheduler is improving over time, but does not reach optimum.

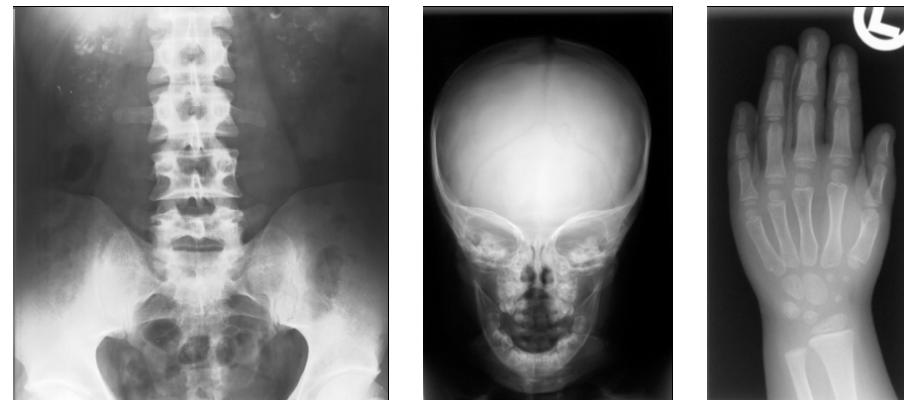
## ▶ Theoretical peak: 8.5 GB/s per boards => 110.5 GB/s

# Applications on ScaleMP



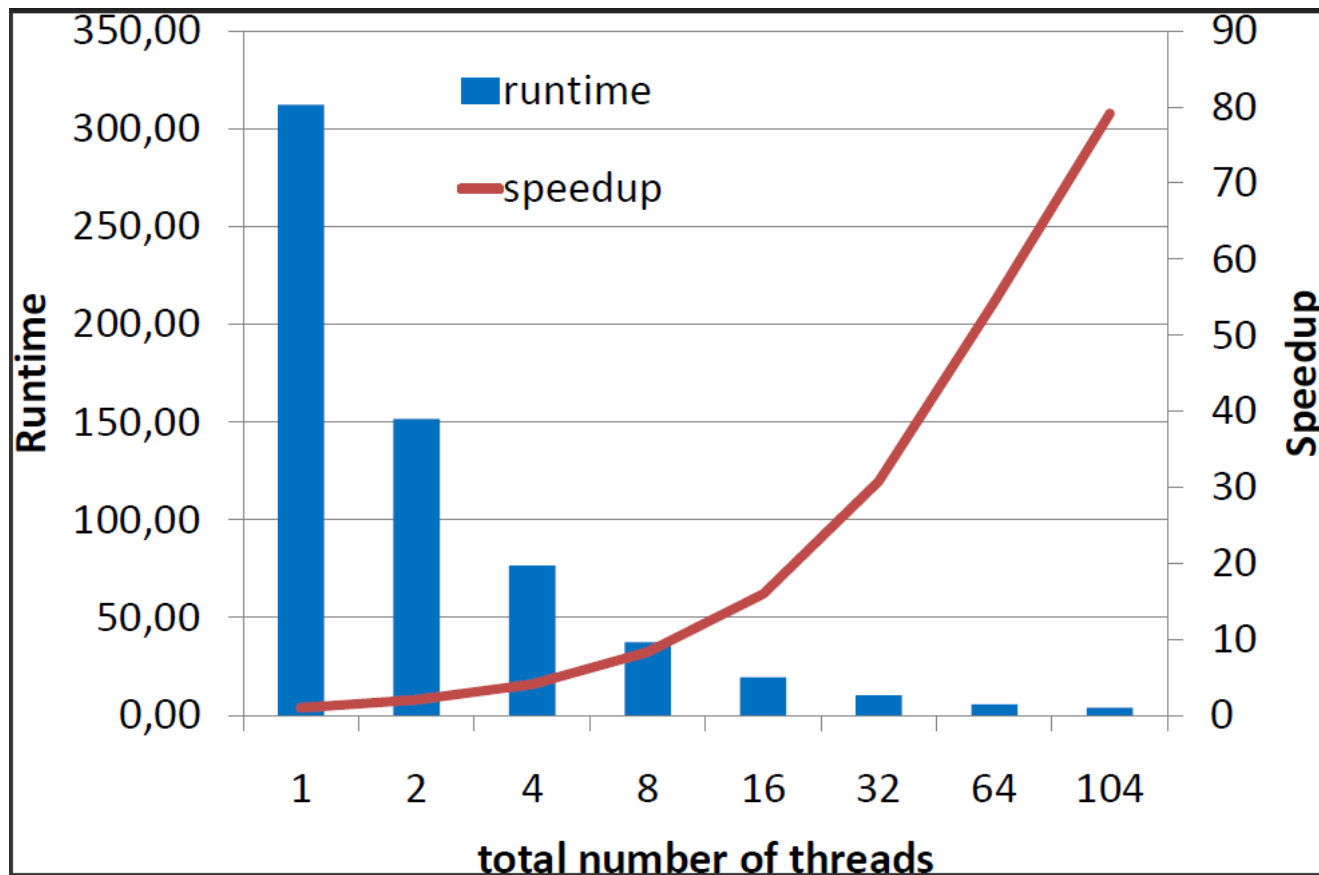
- ▶ **Image Retrieval: A set of query images is compared with all images in a (huge) database and the  $k$  most similar images are returned**
  - ▶ Performance comparison of common features on different databases
  - ▶ Analysis of correlation of different features
  - ▶ Nested Parallelization – Outer level: Queries, Inner level: Comparison
- ▶ **Data Mining is well-suited for Shared-Memory parallelization, but hard with MPI**

*Thomas Deselaers and Daniel Keysers,  
RWTH I6: Chair for Human Language  
Technology and Pattern Recognition*



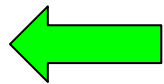
- ▶ **Nested Parallelization improves efficiency by reducing the total overhead.**

- ▶ **Explicit Thread Binding:** `export KMP_AFFINITY=scatter`



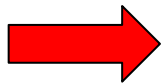
▶ **Geothermal Simulation Package to simulate groundwater flow, heat transport, and the transport of reactive solutes in porous media at high temperatures (3D)**

▶ Forward simulation



▶ 3D finite-differences solver, Coupled transient equations for groundwater flow, Compute state variables from rock properties

▶ Inverse computation

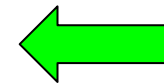
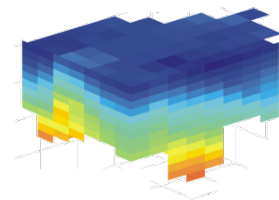
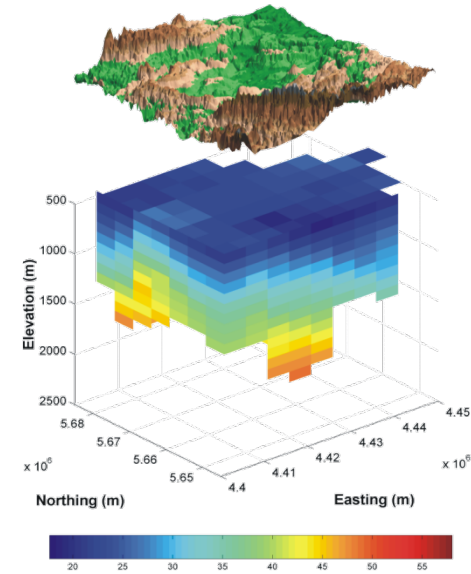


▶ Parameter estimation

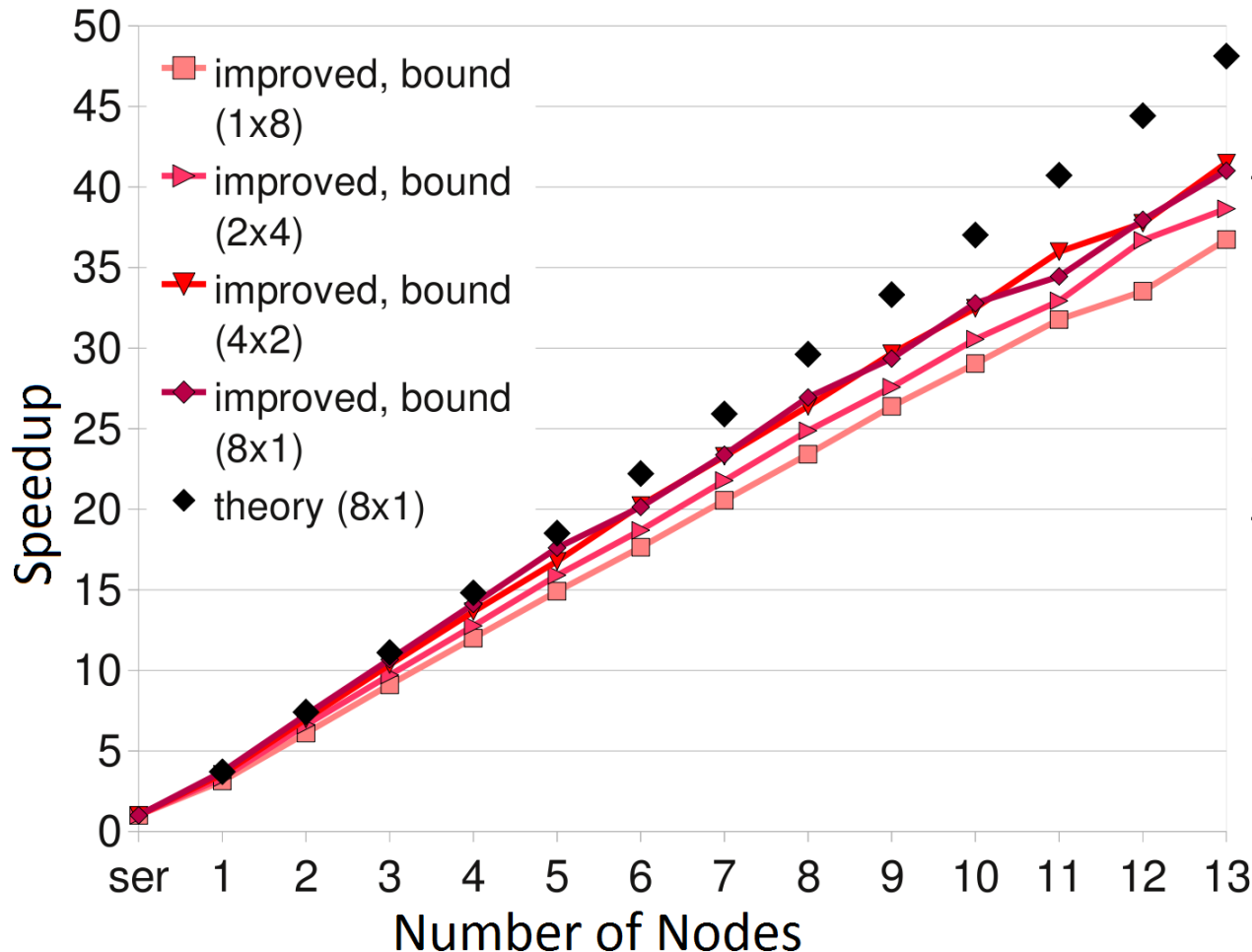
▶ **Written in Fortran, two levels of parallelism**

▶ Independent Computations of the Directional Derivatives

▶ Setup and Solving of linear equation systems



- ▶ **vSMP reduces memory allocation performance, therefore the code has been modified to re-use arrays instead of re-allocation**

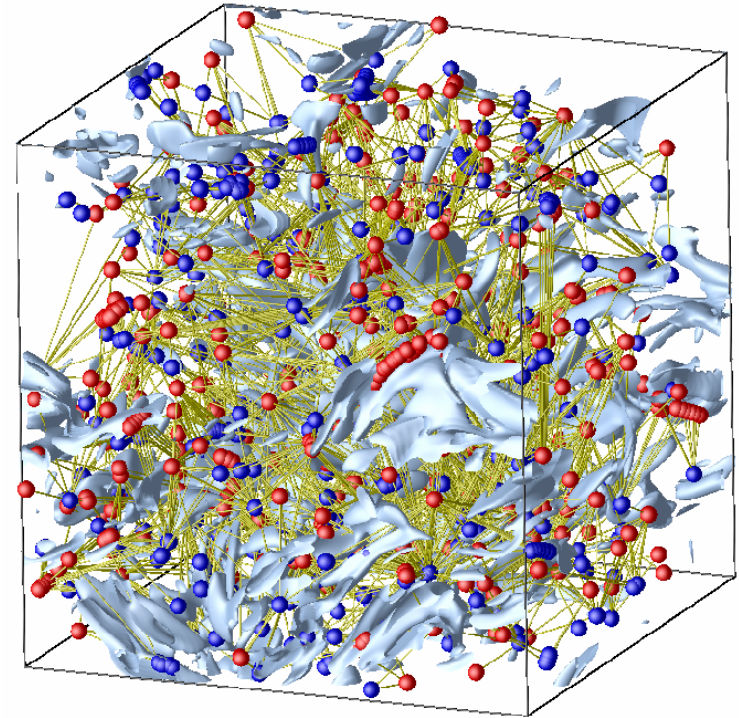


Scalability is near the theoretical peak.

Simulation time has been reduced from one month to less than a day!

No MPI involved 😊

- ▶ **Analysis of DNS (Direct Numerical Simulation) output generated on IBM BlueGene for a  $2048^3$  grid.  $\frac{1}{2}$  year with 16K MPI processes.**
- ▶ **Analysis of geometric properties at micro-scale with the Gradient Trajectory Method by grouping trajectories starting at each grid point which share the same minimum and maximum.**
- ▶ **Not suited for MPI because of extreme load imbalances. Thus parallelized with OpenMP for large Shared Memory Machines.**
- ▶ **Parallelization on ScaleMP with up to  $512^3$  grid points leads to 80X speed-up on 104 cores on the ScaleMP machine.**





# Conclusion

- ▶ **For some applications Shared-Memory is just the right parallelization paradigm and ScaleMP offers a nice solution**
  - ▶ Applications might need multiple levels of parallelization
  
- ▶ **ScaleMP has proven to deliver good performance**
  - ▶ Shared-Memory machine with strong cc-NUMA characteristics
  - ▶ Shared-Memory in software is cheaper than in hardware (but more cc-NUMA)
  
- ▶ **What's still missing and subject to further research:**
  - ▶ Experience with Tuning Methodologies on ScaleMP
    - ▶ Does manual data „prefetching“ help?
    - ▶ Sometimes the optimal thread binding is „surprising“: lack of binding support for nested OpenMP in the Intel compiler.

**Thank you for your attention.**