Experiences with HPC on Windows

Christian Terboven
terboven@rz.rwth-aachen.de

Center for Computing and Communication
RWTH Aachen University

Server Computing Summit 2008
April 7-11, HPI/Potsdam
Agenda

- High Performance Computing (HPC)
  - OpenMP & MPI & Hybrid Programming
- Windows-Cluster @ Aachen
  - Hardware & Software
  - Deployment & Configuration
- Top500 submission
- Case Studies
  - Dynamic Optimization: AVT
  - Bevel Gears: WZL
  - Application & Benchmark Codes
- Summary
High Performance Computing (HPC)

- HPC begins ... when performance (runtime) matters!
  - HPC is about reducing latency - the „Grid“ increases latency
  - Dominating programming languages: Fortran (77 + 95), C++, C

- Focus in Aachen is on Computational Engineering Science!

- We do HPC on Unix for many years - why try Windows?
  - Attract new HPC users:
    - Third party cooperations sometimes depend on Windows
    - Some users look out for the Desktop-like HPC experience
    - Windows is a great development platform
  - We rely on tools: Combine the best of both worlds!
  - Top500 on Windows: Why not 😊?
Parallel Computer Architectures: Shared-Memory

- Dual-Core processors are common now
  - A laptop/desktop is a Shared-Memory parallel computer!
- Multiple processors have access to the same main memory.
- Different types:
  - Uniform memory access (UMA)
  - Non-uniform memory access (ccNUMA), still cache-coherent
- Trend towards ccNUMA

*Intel Woodcrest-based system*

*I am ignoring instruction caches, address caches (TLB), write buffers, prefetch buffers, ... as data caches are most important for HPC applications.*
const int N = 100000;
double a[N], b[N], c[N];
[...]
#pragma omp parallel for
for (int i = 0; i < N; i++) {
    c[i] = a[i] + b[i];
}
[...]

- ... and other threading-based paradigms ...
Parallel Computer Architectures: Distributed-Memory

- Distributed-Memory: Each processor has only access to its own main memory
- Programs have to use external network for communication

Cooperation is done via message exchange → Cluster
Experiences with HPC on Windows

7

MPI Process 1

MPI Process 2

www.mpi-forum.org

int rank, size, value = 23;
MPI_Init(…);
MPI_Comm_size(MPI_COMM_WORLD, &size);
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
if (rank == 0) {
    MPI_Send(&value, 1, MPI_INT, rank+1, …)
} else {
    MPI_Recv(&value, 1, MPI_INT, rank-1, …)
} MPI_Finalize(…);
Multiple levels of parallelism can improve scalability!
- One or more MPI tasks per node
- One or more OpenMP threads per MPI task

Many of our applications are hybrid: Best fit for clusters of SMP nodes.

The SMP node will grow in the future!
Experiences with HPC on Windows

07.04.2008 – C. Terboven

Speedup and Efficiency

- The *Speedup* denotes how much a parallel program is faster than the serial program:

\[ S_p = \frac{T_1}{T_p} \]

- The *Efficiency* of a parallel program is defined as:

\[ E_p = \frac{S_p}{p} \]

- According to Amdahl’s law the Speedup is limited to \( p \).

*Note: Other (similar) definitions for other needs available in the literature.*
Experiences with HPC on Windows

07.04.2008 – C. Terboven

Agenda

- High Performance Computing (HPC)
  - OpenMP & MPI & Hybrid Programming
- Windows-Cluster @ Aachen
  - Hardware & Software
  - Deployment & Configuration
- Top500 submission
- Case Studies
  - Dynamic Optimization: AVT
  - Bevel Gears: WZL
  - Application & Benchmark Codes
- Summary
Recently installed Cluster:
- Fujitsu-Siemens Primergy RS 200 S4 servers
  - 2x Intel Xeon 5450 (quad-core, 3.0 GHz)
  - 16 / 32 GB memory per node
  - 4x DDR InfiniBand (latency: 2.7 us, bandwidth: 1250 MB/s)
- In total: About 25 TFLOP/s peak performance (260 nodes)

Cluster Frontend
- Windows Server 2008
- Better performance when machine gets loaded
- Faster file access: Lower access time, higher bandwidth
- Automatic load balancing: Three machines are clustered to form „one“ interactive machine: 
  cluster-win.rz.rwth-aachen.de
Experiences with HPC on Windows 07.04.2008 – C. Terboven

Harpertown-based InfiniBand Cluster

- Quite a lot of Pizza boxes...
- 4 cables per box:
  - InfiniBand: MPI
  - Gigabit Ethernet: I/O
  - Management
  - Power

Pictures taken during Cluster installation
Software Environment

- **Complete Development Environment**
  - Visual Studio 2005 Pro + Microsoft Compute Cluster Pack
  - Subversion Client, X-Win32, Microsoft SDKs, ...
  - Intel Tool Suite:
    - C/C++ Compiler, Fortran Compiler
    - VTune Performance Analyzer, Threading Tools
    - MKL library, Threading Building Blocks
    - Intel MPI + Intel Tracing Tools
  - Java, Eclipse, ...

- **Growing list of ISV software**
  - ANSYS, HyperWorks, Fluent, Maple, Mathematica, Matlab, MS Office 2003, MSC.Marc, MSC.Adams, ...
  - User-licensed software hosting, e.g. GTM-X
ISV codes in the batch system

- Several requirements
  - No usage of graphical elements allowed
  - No user input allowed (except redirected standard input)
  - Execution of compiled set of commands, Termination

- Exemplary usage instructions for Matlab

Command line:
```
\\cifs\\cluster\\Software\\MATLAB\\bin\\win64\\matlab.exe
/minimize /nosplash
/logfile log.txt
/r "cd('\\cifs\\cluster\\home\\YOUR_USERID\\YOUR_PATH'),
YOUR_M_FILE")
```

- The .M file should contain "quit;" as last statement
Deploying a 256 node cluster

- Setting up the Head Node: approx. 2 hours
  - Installing and configuring Microsoft SQL Server 2005
  - Installing and configuring Microsoft HPC Pack

- Installing the compute nodes:
  - Installing 1 compute node: 50 minutes
  - Installing n compute nodes: 50 minutes? (Multicasting!)
    - We installed 42 nodes at once in 50 minutes
Deploying a 256 node cluster

- The deployment performance is only limited by Head Node:

42 multicasting nodes served by WDS on Head Node

Network utilization at 100%

CPU utilization at 100%
Configuring a 256 node cluster

- Updates? Deploy an updated image to the compute nodes!

<table>
<thead>
<tr>
<th>Node Management</th>
<th>Nodes (243)</th>
<th>Metrics</th>
<th>Cpu Usage (%)</th>
<th>Running Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>WCF Broker Nodes</td>
<td>1.00</td>
<td>Memory Paging (Fetcher/Seconds)</td>
<td>100</td>
<td>Remove</td>
</tr>
<tr>
<td>LINUXAX</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCCS20-250</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WEHBC</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEA</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCCR1-85</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCCR1-99</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LINSAXA</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>By Status</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Online (256)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Offline (4)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unknown (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provisioning (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Starting (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Draining (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Removing (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rejected (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>By Node Template</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multicast Compute</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compute Node JB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Compute</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>empty</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HeadNodeTemplate</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>By Health</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OK (256)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unreachable (1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ongoing Operation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diagnostic Failed</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provisioning Failed</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operations</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Configurations</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Job Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diagnostics</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Charts and Reports</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

HPC Overview | Windows Cluster | Top500 | Case Studies | Summary
Experiences with HPC on Windows

07.04.2008 – C. Terboven

Agenda

- High Performance Computing (HPC)
  - OpenMP & MPI & Hybrid Programming
- Windows-Cluster @ Aachen
  - Hardware & Software
  - Deployment & Configuration
- Top500 submission
- Case Studies
  - Dynamic Optimization: AVT
  - Bevel Gears: WZL
  - Application & Benchmark Codes
- Summary
Experiences with HPC on Windows 07.04.2008 – C. Terboven

The Top500 List (www.top500.org)
Experiences with HPC on Windows

07.04.2008 – C. Terboven

The Top500 List: Windows clusters

Operating system Family share for 11/2007

In addition to the table below, you can view the visual charts using the TOP500 charts page. A direct link to the charts is also available.

<table>
<thead>
<tr>
<th>Operating system Family</th>
<th>Count</th>
<th>Share %</th>
<th>Rmax Sum (GF)</th>
<th>Rpeak Sum (GF)</th>
<th>Processor Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>426</td>
<td>85.20 %</td>
<td>4897046</td>
<td>7956758</td>
<td>970790</td>
</tr>
<tr>
<td>Windows</td>
<td>6</td>
<td>1.20 %</td>
<td>47495</td>
<td>86797</td>
<td>12112</td>
</tr>
<tr>
<td>Unix</td>
<td>30</td>
<td>6.00 %</td>
<td>408378</td>
<td>519178</td>
<td>73532</td>
</tr>
<tr>
<td>BSD Based</td>
<td>2</td>
<td>0.40 %</td>
<td>44783</td>
<td>50176</td>
<td>5696</td>
</tr>
<tr>
<td>Mixed</td>
<td>34</td>
<td>6.80 %</td>
<td>1540037</td>
<td>1900361</td>
<td>580693</td>
</tr>
<tr>
<td>Mac OS</td>
<td>2</td>
<td>0.40 %</td>
<td>28430</td>
<td>44816</td>
<td>5272</td>
</tr>
<tr>
<td>Totals</td>
<td>500</td>
<td>100%</td>
<td>6966169.82</td>
<td>10558086.75</td>
<td>1648095</td>
</tr>
</tbody>
</table>

- Why add just another Linux cluster? → Top500 on Windows!

- Watch out for the Windows/Unix ratio in the upcoming Top500 list in June...
The LINPACK benchmark

- Ranking is determined by running a single benchmark: HPL
  - Solve a dense system of linear equations
  - Gaussian elimination-type of algorithm: $\frac{2}{3} n^3 + O(n^2)$
  - Very regular problem $\rightarrow$ high performance / efficiency
  - [www.netlib.org/hpl](http://www.netlib.org/hpl)

- Result is not of high significance for our applications ...
- ... but it’s a pretty good sanity check for the system!
  - Low performance $\rightarrow$ Something is wrong!

- Steps to success:
  - Cluster setup & sanity check
  - Linpack parameter tuning
Experiences with HPC on Windows 07.04.2008 – C. Terboven

**Results of initial Cluster Sanity Check**

- The cluster was in normal operation since end of January
- Bandwidth test: Variations from 180 MB/s to 1180 MB/s

<table>
<thead>
<tr>
<th>Nodes 1 to 9</th>
<th>714</th>
<th>719.2</th>
<th>719.3</th>
<th>719.7</th>
<th>719.4</th>
<th>649.2</th>
<th>719.3</th>
<th>719.1</th>
<th>649</th>
</tr>
</thead>
<tbody>
<tr>
<td>184.6</td>
<td>185.1</td>
<td>185</td>
<td>185.1</td>
<td>185.1</td>
<td>185</td>
<td>185</td>
<td>185</td>
<td>185.1</td>
<td>185</td>
</tr>
<tr>
<td>1162.6</td>
<td>1177.3</td>
<td>1180.8</td>
<td>1175.4</td>
<td>1173.4</td>
<td>722.4</td>
<td>1176.3</td>
<td>1179.5</td>
<td>723</td>
<td></td>
</tr>
<tr>
<td>1159.6</td>
<td>1178.4</td>
<td>1180.6</td>
<td>1180</td>
<td>1176.5</td>
<td>722</td>
<td>1174.7</td>
<td>1181.5</td>
<td>722.5</td>
<td></td>
</tr>
<tr>
<td>714.5</td>
<td>718.5</td>
<td>720.1</td>
<td>719.8</td>
<td>719.7</td>
<td>649</td>
<td>719.8</td>
<td>719.8</td>
<td>648.6</td>
<td></td>
</tr>
<tr>
<td>1163</td>
<td>1178.2</td>
<td>1176.1</td>
<td>1178.4</td>
<td>1173.7</td>
<td>722.8</td>
<td>1177.7</td>
<td>1180.1</td>
<td>721.8</td>
<td></td>
</tr>
<tr>
<td>1162.5</td>
<td>1175.4</td>
<td>1031.5</td>
<td>1125.4</td>
<td>1174.6</td>
<td>722.4</td>
<td>1179.3</td>
<td>1179.5</td>
<td>723.2</td>
<td></td>
</tr>
<tr>
<td>1164.3</td>
<td>1177.9</td>
<td>1181.9</td>
<td>1178.2</td>
<td>1175.6</td>
<td>721.7</td>
<td>1176.4</td>
<td>1180.4</td>
<td>722.8</td>
<td></td>
</tr>
<tr>
<td>1160.2</td>
<td>1178.2</td>
<td>1180.9</td>
<td>1179.4</td>
<td>1177.2</td>
<td>722.7</td>
<td>1177.8</td>
<td>1179.2</td>
<td>723.3</td>
<td></td>
</tr>
<tr>
<td>710.8</td>
<td>716.2</td>
<td>717.1</td>
<td>716.4</td>
<td>717.8</td>
<td>649</td>
<td>717.5</td>
<td>718</td>
<td>649</td>
<td></td>
</tr>
<tr>
<td>186.9</td>
<td>187.3</td>
<td>187.3</td>
<td>187.3</td>
<td>187.3</td>
<td>187.3</td>
<td>187.3</td>
<td>187.3</td>
<td>187.3</td>
<td></td>
</tr>
<tr>
<td>184.6</td>
<td>185</td>
<td>185</td>
<td>185</td>
<td>185</td>
<td>185.1</td>
<td>185</td>
<td>185</td>
<td>185</td>
<td></td>
</tr>
<tr>
<td>1160.1</td>
<td>1177.5</td>
<td>1179.1</td>
<td>1177.8</td>
<td>1175</td>
<td>722.4</td>
<td>1174.4</td>
<td>1181.5</td>
<td>721.8</td>
<td></td>
</tr>
<tr>
<td>1161.5</td>
<td>1174.7</td>
<td>1179.6</td>
<td>1177.7</td>
<td>1175.5</td>
<td>721.9</td>
<td>1176.5</td>
<td>1179.5</td>
<td>723</td>
<td></td>
</tr>
<tr>
<td>1165.8</td>
<td>1179</td>
<td>1176.1</td>
<td>1180.1</td>
<td>1176.7</td>
<td>722.4</td>
<td>1175</td>
<td>1180.8</td>
<td>722.7</td>
<td></td>
</tr>
<tr>
<td>1164.7</td>
<td>1178.5</td>
<td>1178.3</td>
<td>1174.7</td>
<td>1175.7</td>
<td>722.3</td>
<td>1177.3</td>
<td>1179.5</td>
<td>722.7</td>
<td></td>
</tr>
<tr>
<td>1165.7</td>
<td>1177</td>
<td>1179</td>
<td>1178.6</td>
<td>1177.8</td>
<td>721.2</td>
<td>1172.9</td>
<td>1179.2</td>
<td>721.9</td>
<td></td>
</tr>
<tr>
<td>1162.9</td>
<td>1178.9</td>
<td>1177.1</td>
<td>1180.6</td>
<td>1177.7</td>
<td>722</td>
<td>1176.2</td>
<td>1178.3</td>
<td>721.2</td>
<td></td>
</tr>
<tr>
<td>713.5</td>
<td>720.3</td>
<td>718.2</td>
<td>720.2</td>
<td>718.4</td>
<td>649.4</td>
<td>719.6</td>
<td>719.8</td>
<td>648.8</td>
<td></td>
</tr>
<tr>
<td>1161.5</td>
<td>1178.8</td>
<td>1180.3</td>
<td>1173.9</td>
<td>1176</td>
<td>722.4</td>
<td>1176.8</td>
<td>1179.8</td>
<td>722.4</td>
<td></td>
</tr>
<tr>
<td>1165</td>
<td>1177.1</td>
<td>1178.7</td>
<td>1179.4</td>
<td>1173.5</td>
<td>723.2</td>
<td>1178.5</td>
<td>1177.9</td>
<td>722.2</td>
<td></td>
</tr>
</tbody>
</table>

---

**Nodes 1 to 21**

**HPC Overview**

**Windows Cluster**

**Top500**

**Case Studies**

**Summary**
Parameter Tuning

- HPC approach: Performance Analysis → Tuning → Analysis → Tuning → Analysis → Tuning → ...

Performance got better over time: System and Parameter Tuning
Terboven

HPC Overview

Windows Cluster

Top500 Case Studies

Parameter Tuning – Going the Windows way

Excel application on a laptop controlled the whole cluster!

We easily reached 76.69 GFlop/s on one node. Peak performance is: 8 cores * 3 GHz * 4 results per cycle = 96 Gflop/s → That is 80% efficiency!
Windows HPC Server 2008 in Action

- Job startup comparison – 2048 MPI processes:
  - Our Linux configuration: Order of Minutes
  - Our Windows configuration: Order of Seconds

Video: Job startup on Windows 2008
Experiences with HPC on Windows

07.04.2008 – C. Terboven

Agenda

- High Performance Computing (HPC)
  - OpenMP & MPI & Hybrid Programming
- Windows-Cluster @ Aachen
  - Hardware & Software
  - Deployment & Configuration
- Top500 submission
- Case Studies
  - Dynamic Optimization: AVT
  - Bevel Gears: WZL
  - Application & Benchmark Codes
- Summary
Case Study: Dynamic Optimization (AVT)

- Dynamic optimization in chemical industry

Composition of A und B => unfeasible material

Task: Changing the product specification (from A to B) of a plastic manufactory in ongoing business
- Minimize the junk (composition of A and B)
- Search for economic and ecologic optimal operational mode!

This task is solved by the DyOS (Dynamic Optimization Software) tool, developed at the chair for process systems engineering (AVT: Aachener Verfahrenstechnik) at RWTH Aachen University.
Case Study: Dynamic Optimization (AVT)

What is dynamic optimization?

- Drive 250 m in minimal time!
- Start with \( v = 0 \), stop exactly at \( x = 250 \) m!
- Maximal speed is \( v = 10 \) m/s!
- Maximal acceleration is \( a = 1 \) m/s²!

\[ v(t) \leq v_{\text{max}} \]

\[ x(0), v(0) = 0 \]

\[ v(t_f) = 0, \ x(t_f) \geq x_{i_f} \]
Case Study: Dynamic Optimization (AVT)

- One simulation typically takes up to two weeks and requires a significant amount of memory (>4GB) → MPI parallelization
- Challenge: Commercial software component depending on VS6, only one instance per machine allowed → outsourced into DLL

Tasks of MPI communicator (ROOT):
- storing data
- sending and receiving data
- control over entire software
Case Study: Dynamic Optimization (AVT)

- A scenario consists of several simulations
- Projected compute time on desktop:
  - 5 scenarios à 2 months = 10 months
- Compute time on our cluster (faster machines, multiple jobs):
  - 5 scenarios à 1.5 months = 3 months
- MPI parallelization lead to factor 4.5 on 8 cores:
  - 5 scenarios à 0.3 months = 0.7 months
- Arndt Hartwich: Stability of RZ’s compute nodes is significantly higher than stability of our desktops.
Case Study: KegelToleranzen (WZL)

Simulation of Bevel Gears
- Written in Fortran, using Intel Fortran 10.1 compiler
- Very cache-friendly \( \rightarrow \) runs at high Mflop/s rates
Experiences with HPC on Windows

07.04.2008 – C. Terboven

Case Study: KegelToleranzen (WZL)

- **Target**
  - Pentium/Windows/Intel → Xeon/Windows/Intel
  - Serial Tuning + Parallelization with OpenMP

- **Procedure**
  - Get the tools: Porting to UltraSparc IV/Solaris/Sun Studio
  - Simulog Foresys: Convert to Fortran 90
    - 77000 Fortran77 lines → 91000 Fortran 90 lines
  - Sun Analyzer: Runtime Analysis with different datasets
    - Deduce targets for Serial Tuning and OpenMP Parallelization
  - OpenMP Parallelization: 5 Parallel Regions, 70 Directives
  - Get the tools: Porting new code to Xeon/Windows/Intel
  - Intel Thread Checker: Verification of OpenMP Parallelization

- **Put new code in production on Xeon/Windows/Intel**
Case Study: KegelToleranzen (WZL)

Comparing Linux and Windows Server 2003:

Performance of KegelToleranzen

- Linux 2.6: 4x Opteron dual-core, 2.2 GHz
- Windows 2003: 4x Opteron dual-core, 2.2 GHz

Comparing Linux and Windows Server 2003:

- Linux 2.6: 4x Opteron dual-core, 2.2 GHz
- Windows 2003: 4x Opteron dual-core, 2.2 GHz

better

24%
Case Study: KegelToleranzen (WZL)

- Comparing Linux and Windows Server 2008:

  ![Graph showing performance comparison](image)

  - Linux 2.6: 4x Opteron dual-core, 2.2 GHz
  - Windows 2003: 4x Opteron dual-core, 2.2 GHz
  - Linux 2.6: 2x Harpertown quad-core, 3.0 GHz
  - Windows 2008: 2x Harpertown quad-core, 3.0 GHz

  Performance of KegelToleranzen

  - 7% better

Experiences with HPC on Windows

07.04.2008 – C. Terboven
Case Study: KegelToleranzen (WZL)

- Comparing Linux and Windows Server 2008:

**Performance of KegelToleranzen**

- What about 96 Gflop/s per node?!?

- In fact, this application is rather fast.
- HPC applications are waiting for memory!

- Performance gain for the user: Speedup of 5.6 on one node.
- Even better from starting point (desktop: 220 MFlop/s).
- MPI parallelization is work in progress.

![Graph showing performance comparison between Linux and Windows Server 2008](image-url)
Experiences with HPC on Windows 07.04.2008 – C. Terboven

Agenda

- High Performance Computing (HPC)
  - OpenMP & MPI & Hybrid Programming
- Windows-Cluster @ Aachen
  - Hardware & Software
  - Deployment & Configuration
- Top500 submission
- Case Studies
  - Dynamic Optimization: AVT
  - Bevel Gears: WZL
  - Application & Benchmark Codes
- Summary
Einladung zum 1. Treffen der Windows High Performance Computing Nutzergruppe im deutschsprachigen Raum

21./22. April 2008
Rechen- und Kommunikationszentrum, RWTH Aachen

http://www.rz.rwth-aachen.de/winhpcug
Invitation: Windows-HPC User Group Meeting

- Ziele
  - Informationsaustausch zwischen Nutzern untereinander
  - Informationsaustausch zwischen Nutzern und Microsoft

- Montag, 21. April
  - 17:00h: Domführung
  - 18:30h: Gemeinsames Abendessen

- Dienstag, 22. April
  - Präsentationen der Firmen Allinea, Cisco, Intel, The Mathworks und Microsoft
  - Präsentationen von Einrichtungen aus Forschung und Lehre
  - Frage-und-Antwort-Runde mit HPC-Experten von Microsoft
Summary & Outlook

- The Windows-HPC environment has been well accepted
  - Growing interest and need of compute power on Windows.

- Windows HPC Server 2008 (beta) is pretty impressive!!!
  - Deploying and Configuring 256 nodes.
  - Job Startup and Job Management.
  - Performance improvements & Linpack efficiency.

- Interoperability in heterogeneous environments got easier.
  - E.g. WDS can interoperate with Linux DHCP and PXE-Boot.

- Performance: Need for Windows-specific tuning.
The End

Thank you for your attention!