science + computing

A Bull Group Company

Performance Analysis and Optimizations of CAE Applications

Case Study: STAR-CCM+

Dr. Fisnik Kraja
HPC Services and Software

science + computing ag
IT Service and Software Solutions for Complex Computing Environments
Tuebingen | Muenchen | Berlin | Duesseldorf

02 April 2014

science + computing at a glance

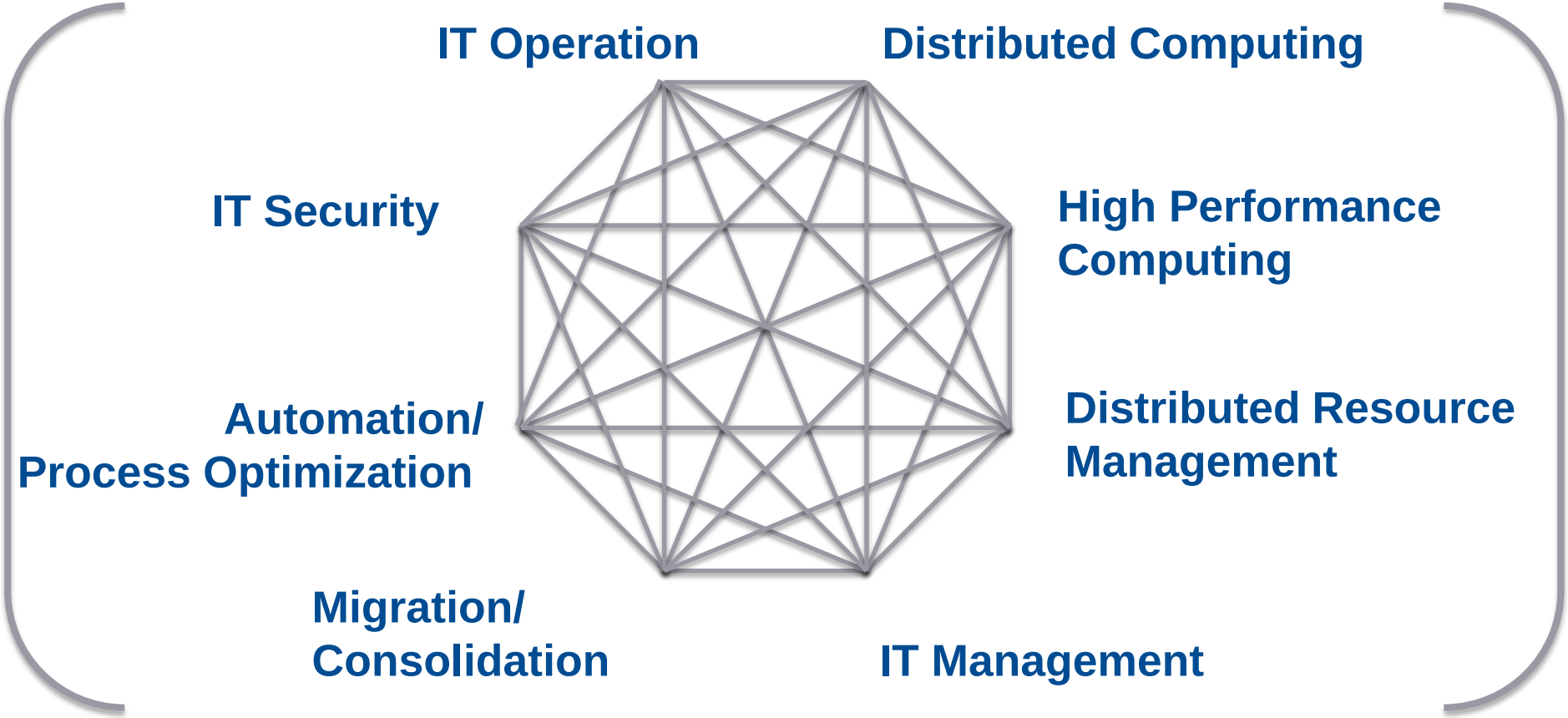
science + computing
A Bull Group Company

- Founded in 1989
- ~300 Employees in Tübingen, München, Berlin, Düsseldorf, Ingolstadt
- Focus on technical computing (CAD, CAE, CAT)
- We count the following among our customers:
  - Automobile manufacturers
  - Suppliers of the automobile industry
  - Manufacturers of microelectronic components
  - Aerospace companies
  - Manufacturing
  - Chemical and pharmaceutical companies
  - Public Sector

Dr. Fisnik Kraja | f.kraja@science-computing.de

s+c core competencies:
IT Services | Consulting | Software

science + computing
A Bull Group Company



Dr. Fisnik Kraja | f.kraja@science-computing.de

Main Contributors

science + computing
A Bull Group Company

- Applications and Performance Team
  - Madeleine Richards
  - Damien Declat (TL)

BULL
Architect of an Open World™

- HPC Services and Software Team
  - Josef Hellauer
  - Oliver Schröder
  - Jan Wender (TL)

science + computing
A Bull Group Company

- CD-adapco

CD-adapco™

Dr. Fisnik Kraja | f.kraja@science-computing.de

Overview

science + computing
A Bull Group Company

- Introduction
- Benchmarking Environment
- Initial Performance Optimizations
- Performance Analysis and Comparison
  - CPU Frequency Dependency
  - Memory Hierarchie Dependency
  - CPU Comparison
- Hyperthreading Impact
- Intel(R) Turbo Boost Analysis
- MPI Profiling
- Conclusions

Dr. Fisnik Kraja | f.kraja@science-computing.de

Introduction

science + computing
A Bull Group Company

- ISV software is in general pre-compiled
- A lot of optimization possibilities exist in the HPC environment
  - Node selection and scheduling (scheduler)
  - System- and node-level task placement and binding (runtimes)
  - Operating system optimizations
- Purpose of this study is to:
  - Analyse the behavior of an ISV application
  - Apply optimizations that improve resource utilization
- The test case
  - STAR-CCM+ 8.02.008
  - Platform Computing MPI-08.02
  - Aerodynamics Simulation (60M)

Dr. Fisnik Kraja | f.kraja@science-computing.de

Benchmarking Environment

Architect of an Open World

CD-adapco

science + computing
A Bull Group Company

Compute Node	Sid B710	Sid B71010c	Sid B71012c	Robin ivy27-12c-hton	Robin ivy27-12c-E3-htoff
Processor	Intel E5-2697 V2 IvyBridge	Intel E5-2680 V2 IvyBridge	Intel E5-2697 V2 IvyBridge	Intel E5-2697 V2 IvyBridge	Intel E5-2697 V2 IvyBridge
Frequency	2.70 GHz	2.80 GHz	2.70 GHz	2.70 GHz	2.70 GHz
Cores per processor	12	10	12	12	12
Sockets per node	2	2	2	2	2
Cores per nodes	24	20	24	24	24
Memory	32 GB	64 GB	64 GB	64 GB	64 GB
Frequency of memory	1866 MHz	1866 MHz	1866 MHz	1866 MHz	1866 MHz
IO – FS	NFS over IB	NFS over IB	NFS over IB	NFS over IB	NFS over IB
Interconnect	IB FDR	IB FDR	IB FDR	IB FDR	IB FDR
STAR-CCM+	8.02.008	8.02.008	8.02.008	8.02.008	8.02.008
Platform MPI	8.2.0.0	8.2.0.0	8.2.0.0	8.2.0.0	8.2.0.0
SLURM	2.6.0	2.6.0	2.6.0	2.5.0	2.5.0
OFED	1.5.4.1	1.5.4.1	1.5.4.1	1.5.4.1	1.5.4.1

Dr. Fisnik Kraja | f.kraja@science-computing.de

Initial Optimizations

science + computing
A Bull Group Company

1. CPU Binding (cb)
   - Bind tasks to specific physical/logical cores. This eliminates the overhead coming from thread migrations and improves data locality in combination with the first touch policy.

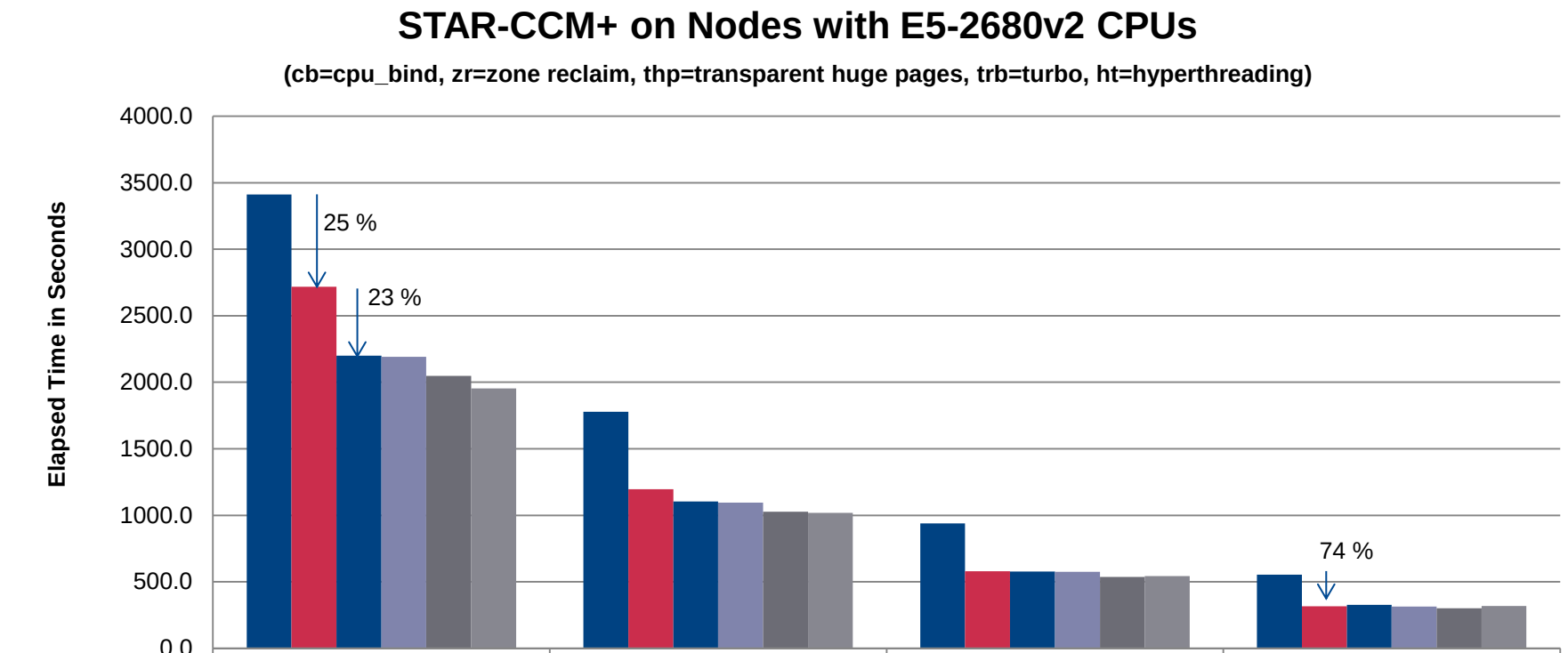
2. Zone Reclaim (zr)
   - Linux measures at startup the transfer rate between NUMA nodes and decides whether to enable or not Zone Reclaim in order to optimize memory performance on NUMA systems. We enabled it by force.

3. Transparent Huge Pages (thp)
   - Latest Linux kernels support different page sizes. In some cases, to improve the performance huge pages are used since memory management is simplified. THP is an abstraction layer in RHEL 6 that makes it easy to use huge pages.

4. Optimizations for Intel (R) CPUs
   - Turbo Boost (trb) enables the processor to run above its base operating frequency via dynamic control of the CPU's clock rate
   - Hyperthreading (ht) allows the operating system to address two (or more) virtual or logical cores per physical core, and share the workload between them when possible.

Dr. Fisnik Kraja | f.kraja@science-computing.de

Obtained Improvements



	120 Tasks 6 Nodes	240 Tasks 12 Nodes	480 Tasks 24 Nodes	960 Tasks 48 Nodes
Opt: none	3412.6	1779.3	938.1	553.0
Opt: cb	2719.0	1196.5	578.7	316.7
Opt: cb,zr	2199.0	1103.5	577.3	325.8
Opt: cb,zr,thp	2191.5	1094.4	575.2	313.6
Opt: cb,zr,thp,trb	2048.6	1027.7	537.2	300.9
Opt: cb,rz,thp,ht	1953.8	1017.3	543.8	317.2

Dr. Fisnik Kraja | f.kraja@science-computing.de

Performance Analysis and Comparison

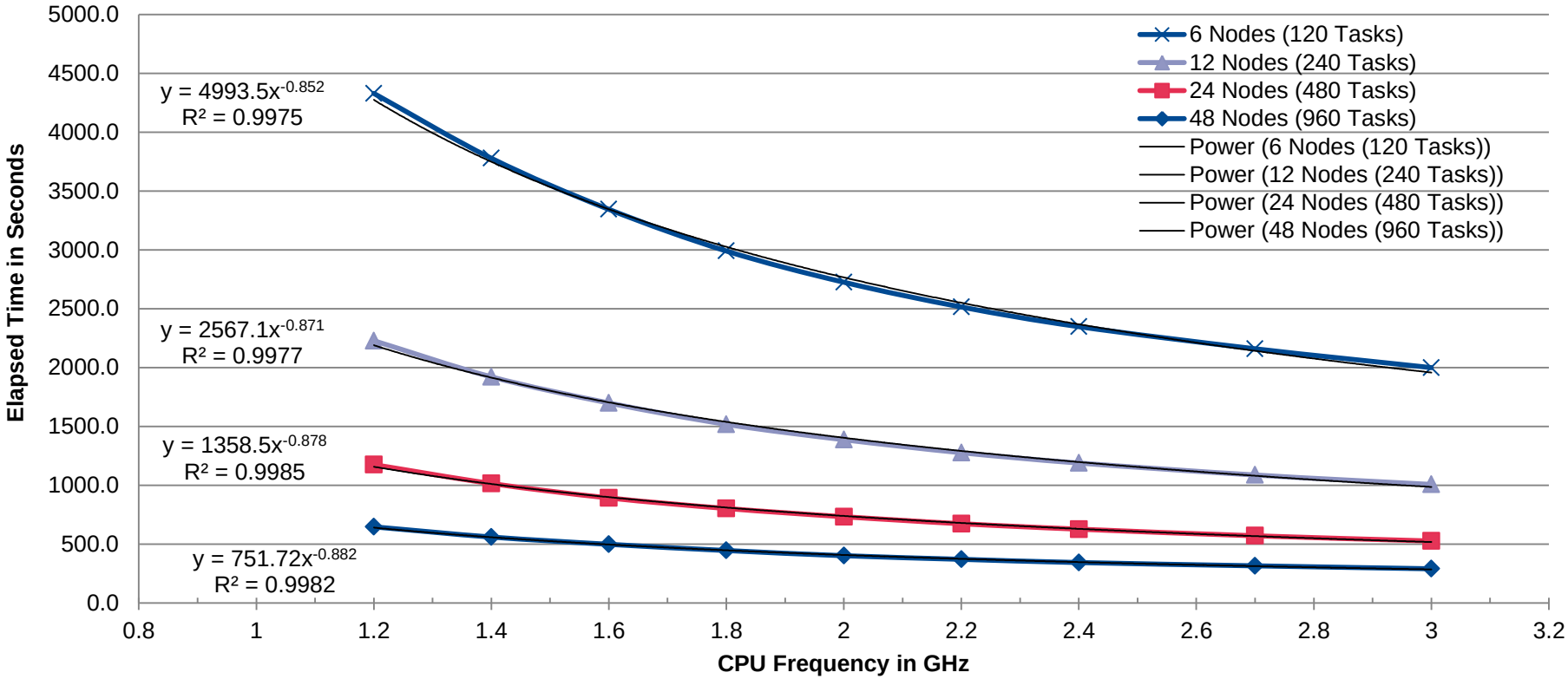
science + computing
A Bull Group Company

1. What can we do?
   - Analyze the dependency on:
     - CPU Frequency
     - Memory Hierarchy
   - Compare the performance of different CPU types
   - Analyze the impact of Hyperthreading and Turbo Boost
   - Profile MPI communications

2. Why should we do that?
   - To better utilize the resources in a heterogeneous environment
     - by selecting the appropriate compute nodes
     - by giving each job exactly the resources needed (neither more nor less)
   - To find an informed compromise between performance and costs
     - power consumption and licenses
   - To predict the behavior of the application on upcoming systems

Dr. Fisnik Kraja | f.kraja@science-computing.de

Dependency on the CPU Frequency

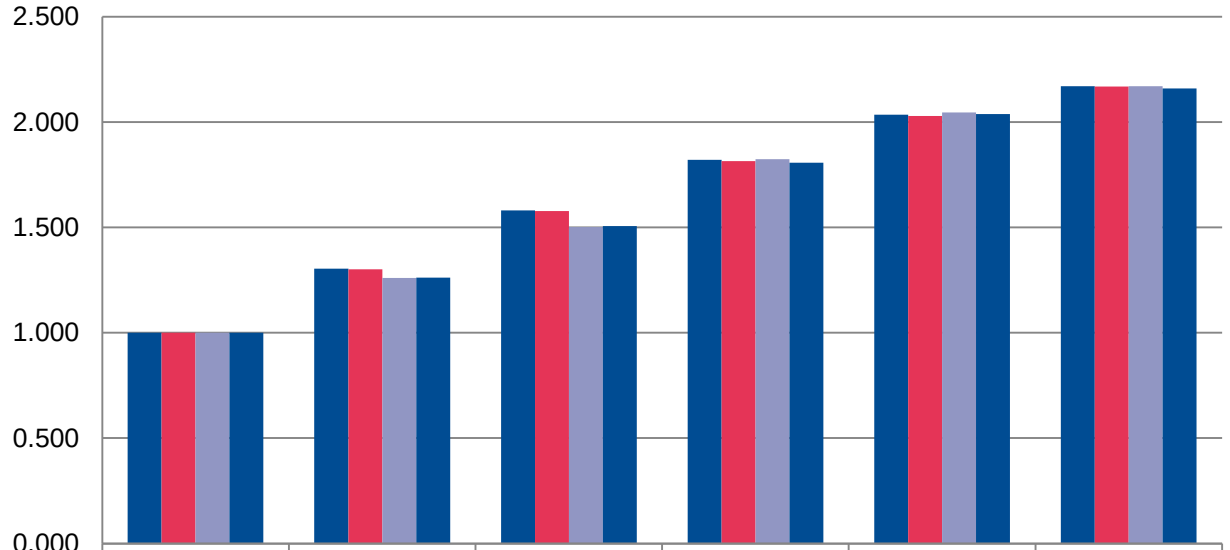


1. This test case shows a 85-88% dependency on the CPU frequency
   - 85% on 6 Nodes
   - 87% on 12 Nodes
   - 88% on 24 and also on 48 Nodes

Dr. Fisnik Kraja | f.kraja@science-computing.de

CPU Frequency Impact on Memory Throughput

science + computing
A Bull Group Company



	1.2 GHz	1.6 GHz	2.0 GHz	2.4 GHz	2.8 GHz	3.1 GHz - TB
Speedup(6 Nodes)	1.000	1.304	1.581	1.821	2.034	2.170
Memory Throughput Increase (6 Nodes)	1.000	1.301	1.577	1.815	2.029	2.168
Speedup(48 Nodes)	1.000	1.259	1.501	1.823	2.046	2.171
Memory Throughput Increase (48 Nodes)	1.000	1.262	1.505	1.807	2.037	2.160

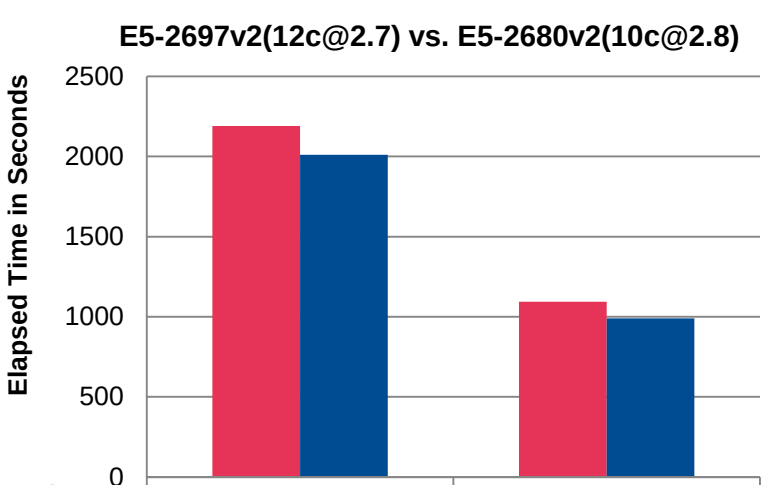
1. Memory throughput increases with almost the same ratio as the speedup:
   - The integrated memory controller and the caches are faster
   - We can see that memory is not a bottleneck

2. Almost the same behavior is observed with different tests on 6 and 48 nodes

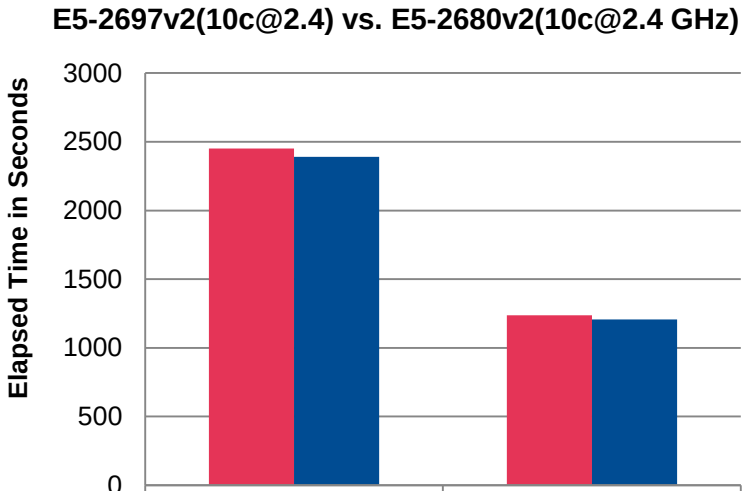
Dr. Fisnik Kraja | f.kraja@science-computing.de

CPU Comparison
E5-2697v2 vs. E5-2680v2

science + computing
A Bull Group Company



Nodes	6	12
Sid B71010c - 2.8 GHz	2191.5	1094.4
Sid B71012c - 2.7 GHz	2011.9	991.0



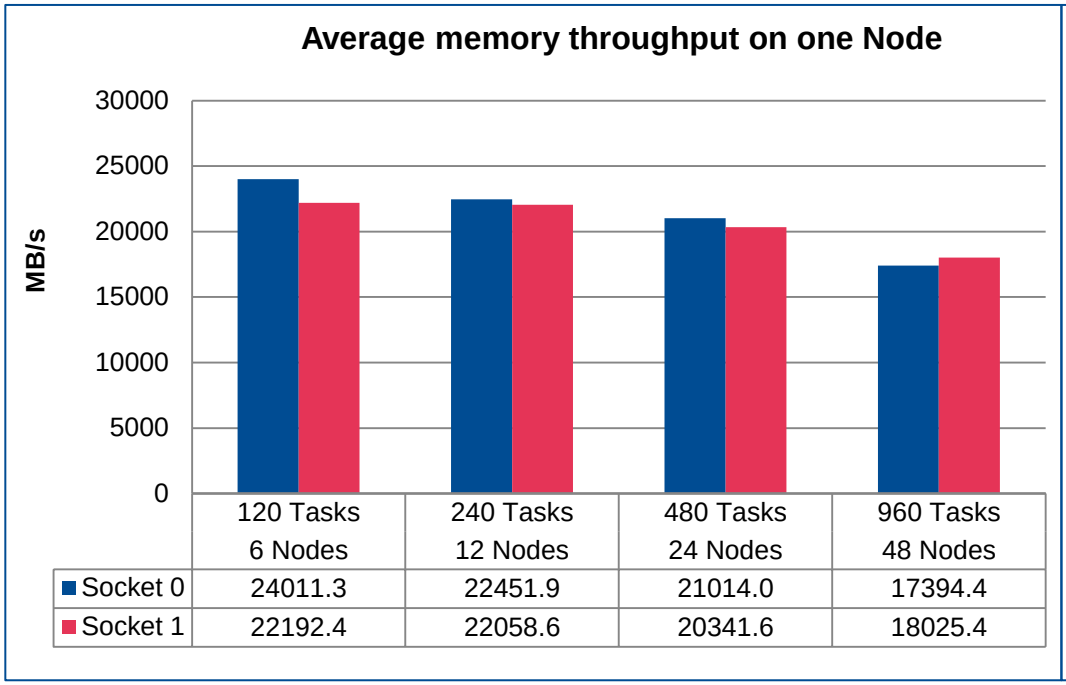
Nodes (Tasks)	6 (120)	12 (240)
Sid B71010c - E5-2680v2	2450.4	1236.20
Sid B71012c - E5-2697v2	2389.2	1205.6

- The 12 core CPU is faster by 8-9 %
  - However, there are also drawbacks:
    - Increased power consumption
    - Increased license costs

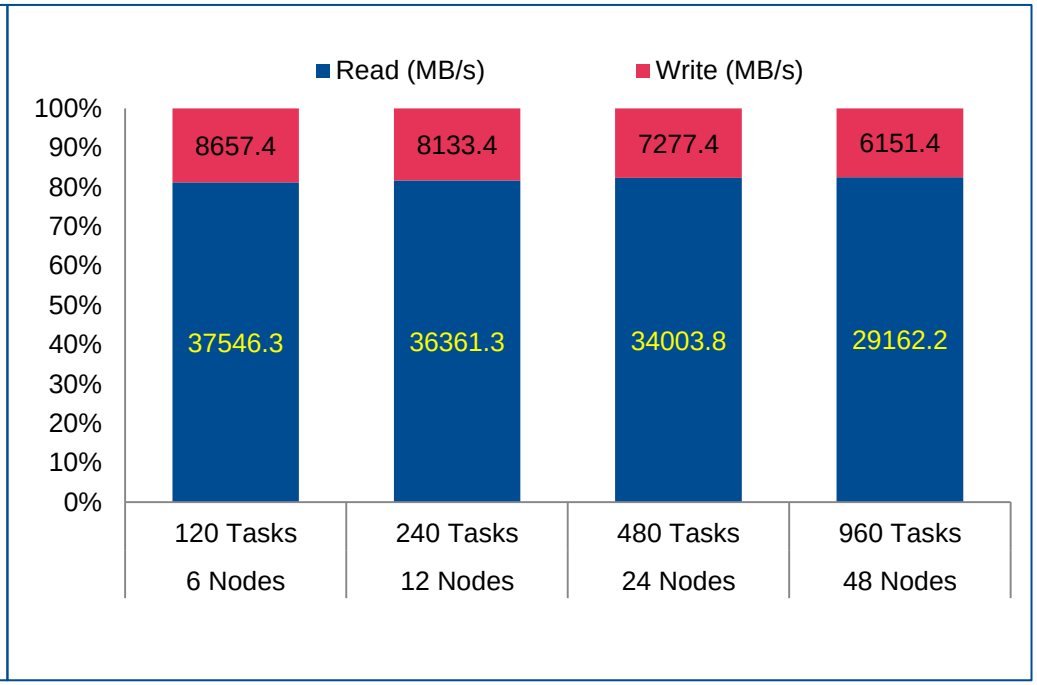
- In these tests we use 10 cores @ 2.4 GHz on both CPU Types
- The E5-2697v2 is still faster
- Why?

Dr. Fisnik Kraja | f.kraja@science-computing.de

Memory Throughput Analysis



	120 Tasks 6 Nodes	240 Tasks 12 Nodes	480 Tasks 24 Nodes	960 Tasks 48 Nodes
Socket 0	24011.3	22451.9	21014.0	17394.4
Socket 1	22192.4	22058.6	20341.6	18025.4

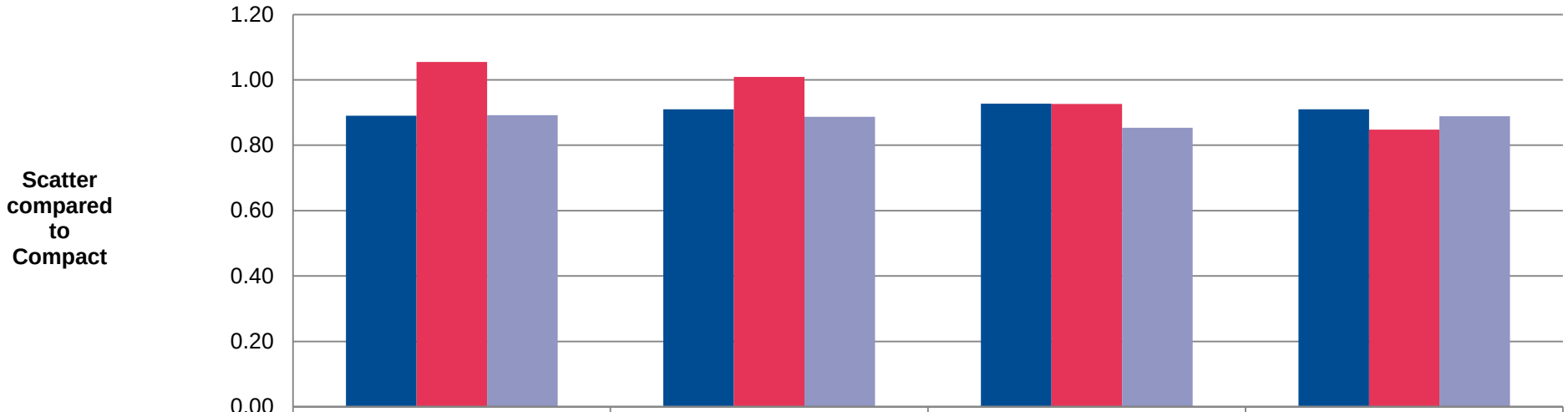


1. Memory is stressed a bit more when running on 6 nodes
2. The more nodes are used, the less memory bandwidth is needed:
   - More time is spent on MPI
   - More caches become available
3. Sockets are well balanced, considering that these measurements are done on the first node where Rank 0 is running.
4. The ratio between Write and Read to memory is always around 20 / 80 %.

Dr. Fisnik Kraja | f.kraja@science-computing.de

Memory Hierarchy Dependency
Scatter vs. Compact Task Placement

science + computing
A Bull Group Company



	120 Tasks 12 Nodes	240 Tasks 24 Nodes	480 Tasks 48 Nodes	960 Tasks 96 Nodes
Impact on TSET	0.89	0.91	0.93	0.91
Impact on Memory Throughput	1.06	1.01	0.93	0.85
Impact on L3 Misses	0.89	0.89	0.85	0.89

Performance improves with *Scatter Mode*

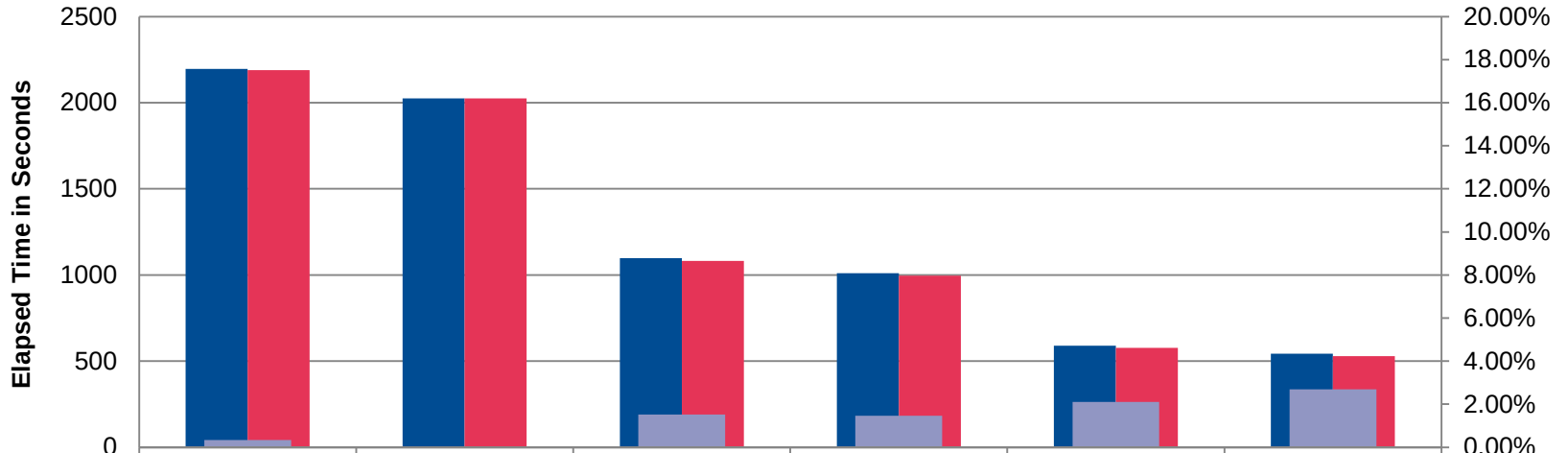
- Even in cases when less memory bandwidth is used (24 and 48 Nodes)
- The reason for this is that L3 cache on the second socket becomes available
- By doubling the L3 cache per task/core we have reduced the cache misses by at least 10 %

Dr. Fisnik Kraja | f.kraja@science-computing.de

Hyperthreading Impact on Performance

HT-ON vs. HT-OFF - 24 Tasks per Node - E5 2697v2

science + computing
A Bull Group Company

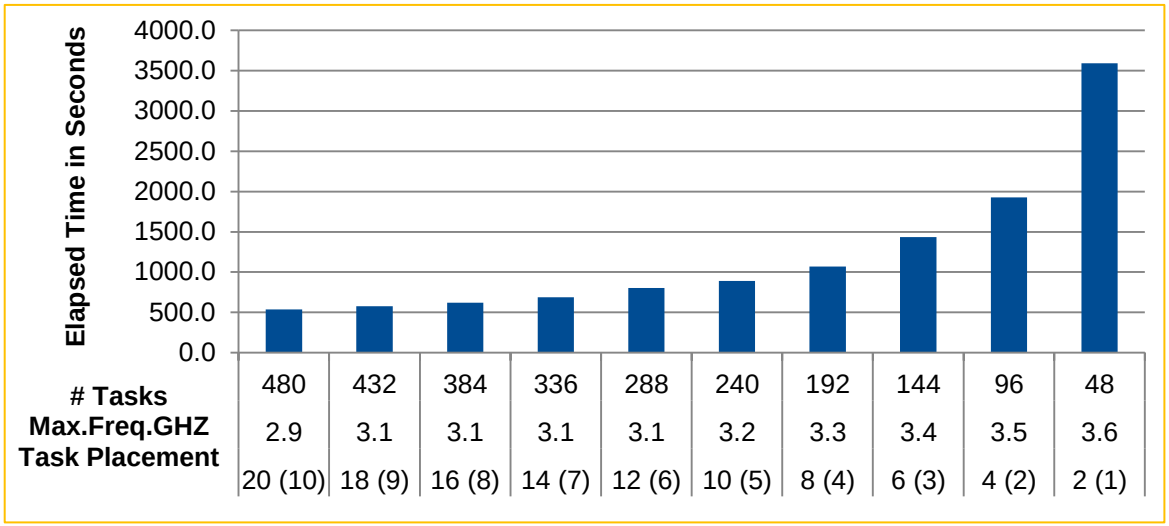


	2,4 ghz	2,7 ghz	2,4 ghz	2,7 ghz	2,4 ghz	2,7 ghz
	6 nodes x 24cores		12 nodes x 24 cores		24 nodes x 24 cores	
Robin_ivy27-12c-hton_sbatch	2197	2026	1098	1011	589	544
Robin_ivy27-12c-E3-htoff_sbatch	2189	2025	1081	996	577	529
Reduction of TSET	0.33%	0.02%	1.52%	1.46%	2.11%	2.69%

1. Here we analyze the impact of having HT=ON/OFF (in BIOS)
   - Even when not overpopulating the nodes
2. As shown the impact on performance is minimal
3. The real reasons for this behavior are not clear
   - Could be OS Jitter
   - What do you think?

Dr. Fisnik Kraja | f.kraja@science-computing.de

Turbo Boost Analysis



During these tests we use always 24 nodes and reduce by 2 the number of tasks per node. This reduces the number of CPU cores being used:

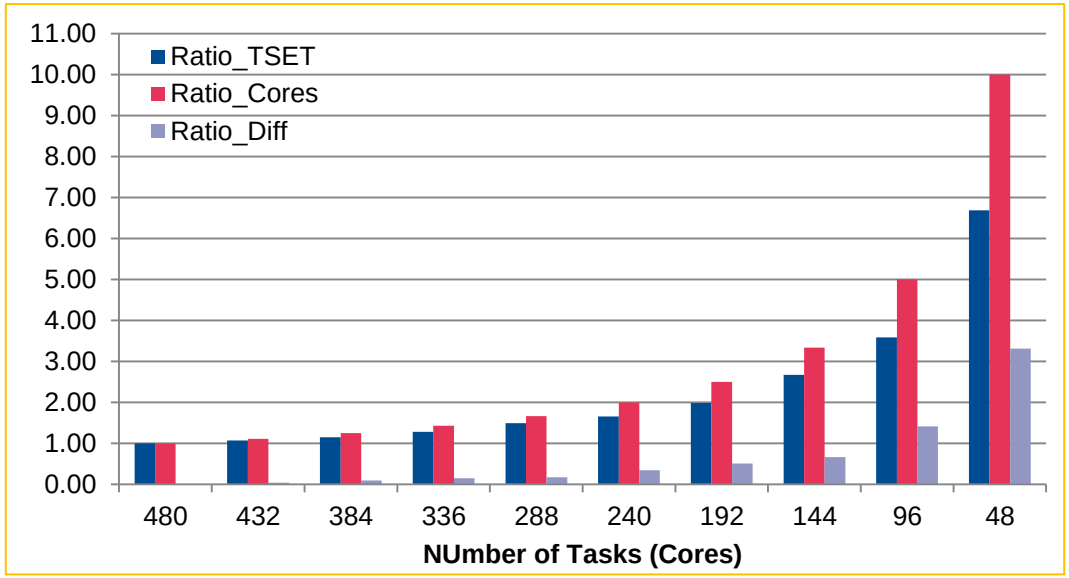
1. Allowing the Turbo Boost Technology to clock the cores up to 3.6 GHz (including here the integrated memory controller)
2. Giving each core a higher memory bandwidth

As expected, on fewer cores the elapsed time increases.

However the impact becomes more pronounced as one reduces the number of cores.

By reducing the number of cores by a factor of 10, the elapsed time increases only by a factor of 6.7.

This could be an interesting use case to reduce license costs.

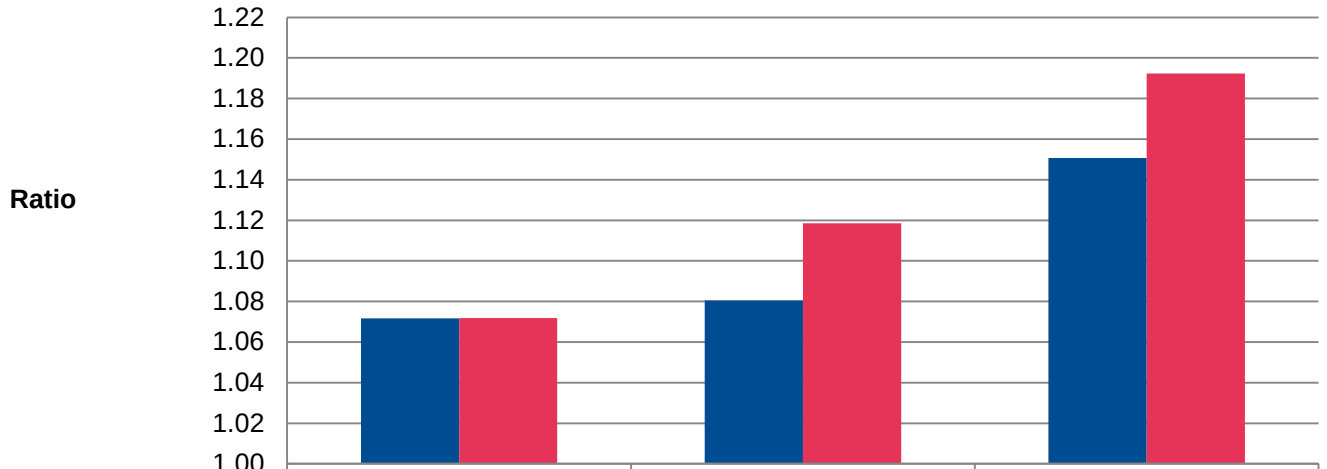


Dr. Fisnik Kraja | f.kraja@science-computing.de

Turbo Boost and Hyperthreading impact on Memory Throughput

Tests with 240 Tasks on 12 Fully/Over Populated Nodes

science + computing
A Bull Group Company

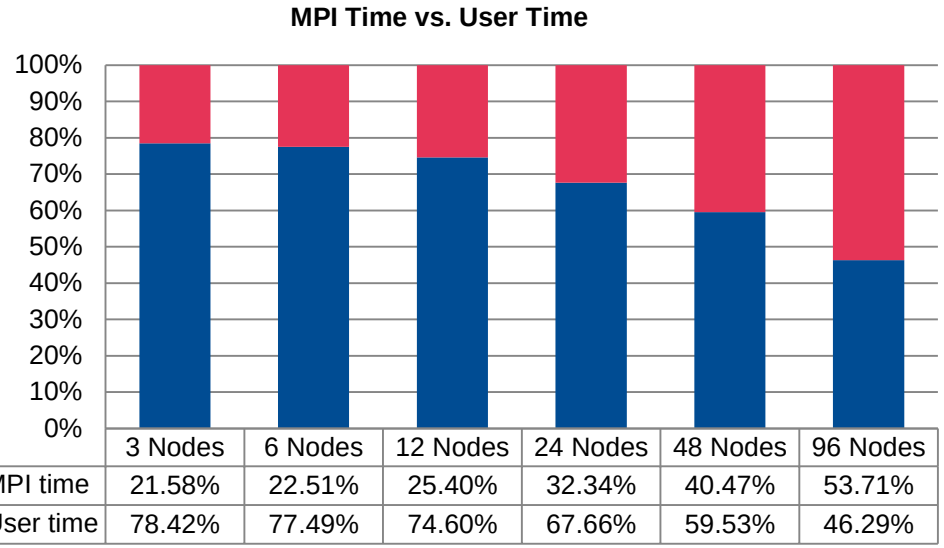


	Fully-Turbo vs Fully	HT (over populated) vs. Fully	TB+HT (overpopulated) vs. Fully
Speedup in Time	1.07	1.08	1.15
Memory Throughput Increase	1.07	1.12	1.19

1. The Turbo Boost impact on memory throughput and on speedup is at the same ratio
2. The HT impact is not.
   - The reason for this might be the eviction of cache data since 2 threads are running on the same core.

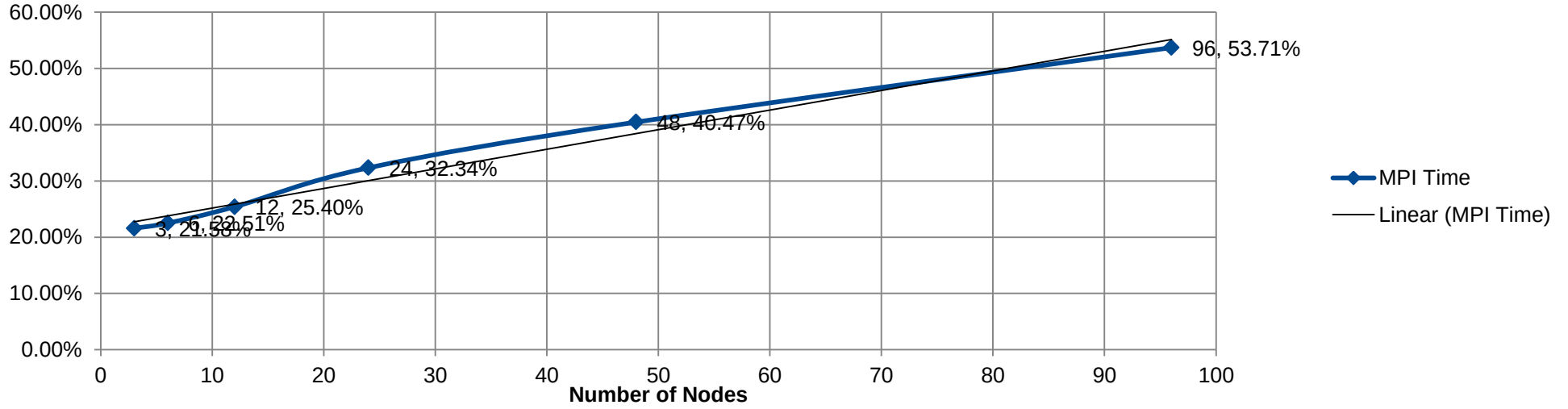
Dr. Fisnik Kraja | f.kraja@science-computing.de

MPI Profiling (1)



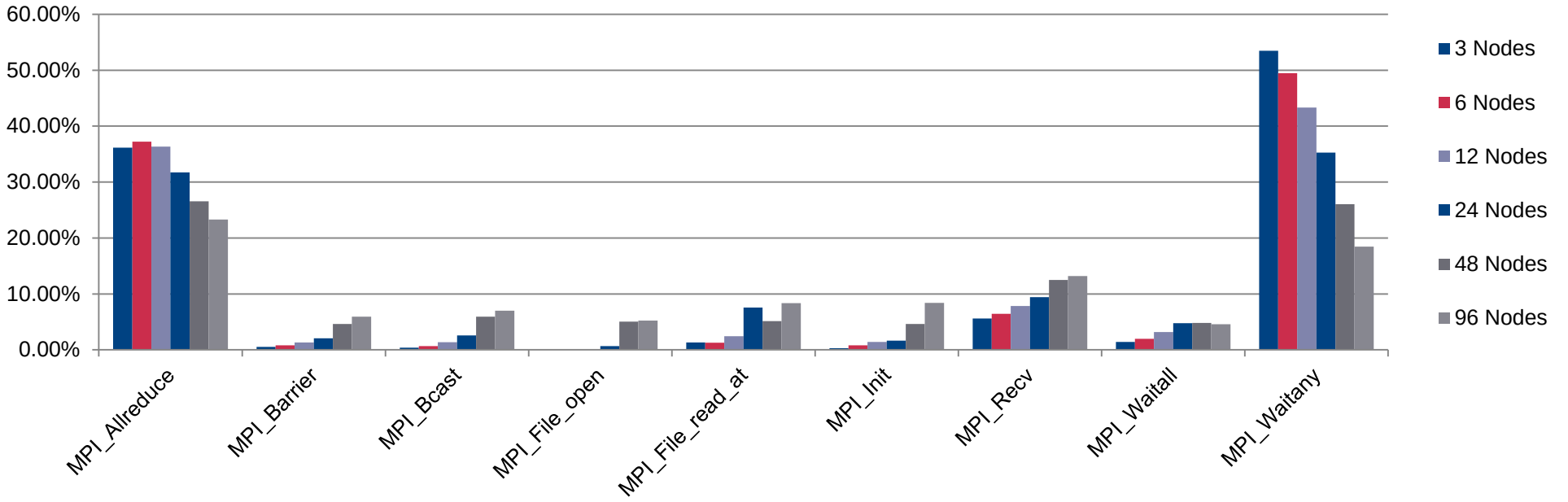
	3 Nodes	6 Nodes	12 Nodes	24 Nodes	48 Nodes	96 Nodes
MPI time	21.58%	22.51%	25.40%	32.34%	40.47%	53.71%
User time	78.42%	77.49%	74.60%	67.66%	59.53%	46.29%

As shown in these charts the part of time spent in MPI communications increases almost linearly with the increase in the number of nodes.



Dr. Fisnik Kraja | f.kraja@science-computing.de

MPI Profiling (2)



1. Most of the MPI Time is spent in MPI_Allreduce and MPI_Waitany
   - Benchmarks on Platform MPI selection of collective algorithms
   - 5-10% performance improvement of MPI collective time is expected

2. While increasing the number of Nodes, the part of time spent on MPI_Allreduce and MPI_Waitany gets distributed over the other MPI calls like MPI_Recv, MPI_Waitall, etc …

3. Interesting is that up to 9% of the MPI Time is spent on MPI_File_read_at
   - A parallel File system might help in reducing this part

Dr. Fisnik Kraja | f.kraja@science-computing.de

Conclusions

science + computing
A Bull Group Company

1. CPU Frequency
   - STAR-CCM+ showed dependency on the CPU Frequency
2. Cache Size
   - STAR-CCM+ is dependent on the Cache Size per Core
3. Turbo Boost
   - Is worth only in cases when not all the cores are used
4. Hyperthreading
   - When used, HT has no big positive impact on performance
   - When not used, HT has a small negative impact on performance
5. Memory Bandwidth and Latency
   - STAR-CCM+ is more latency than bandwidth dependent
6. MPI Profiling
   - MPI Time increases linearly with the number of nodes
7. File System
   - Parallel File systems should be considered for MPI-IO

science + computing

A Bull Group Company

Thank you for your attention.

Fisnik Kraja
science + computing ag
www.science-computing.de

Email: f.kraja@science-computing.de

Theory behind
CPU Frequency Dependency

science + computing
A Bull Group Company

- Lets start with the statement
  - To Solve ***m*** Problems on ***x*** resourses we need ***t=T*** time
- Then we can make the following assumptions:
  - To solve ***n*m*** problems on ***n*x*** resourses we still need ***t=T*** time
  - To solve ***m*** problems on ***n*x*** resourses we need ***t=T/n*** time (hopefully)

- From the last assumption: $\mathrm{t} \cong T \times n^{-1}$
- We can expand it: $\mathrm{t} \cong T \times n^{-(a+b)}$, where $a + b = 1$
- We use a to represent the dependency on CPU frequency
- We use b to represent the dependency on other factors
- To find the value of a we keep $b = 0$