Challenges to Application Development and Support posed by current and future HPC Architectures

Markus Rampp, Stefan Heinzel (RZG)
(markus.rampp@rzg.mpg.de)

rzg
RECHENZENTRUM GARCHING

Topics

Performance trends:

- **entering the Petaflop era (Top500 list)**
- **Moore's law and new consequences**

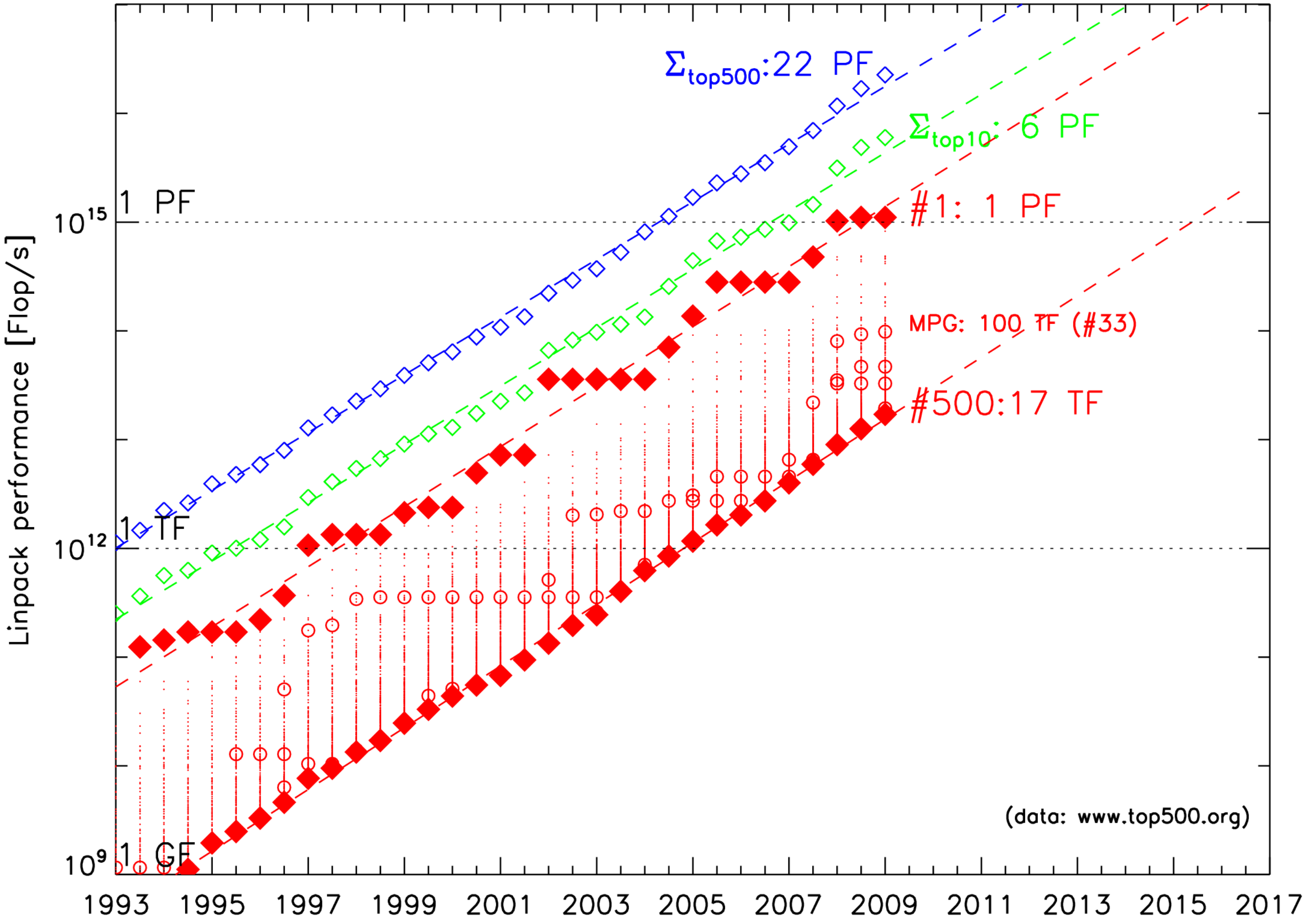
Towards HPC systems with *sustained* Petaflop performance

Implications for applications:

- **towards applications with sustained Petaflop performance**

Notes on visualization & data analysis

Performance trends (Top 500)

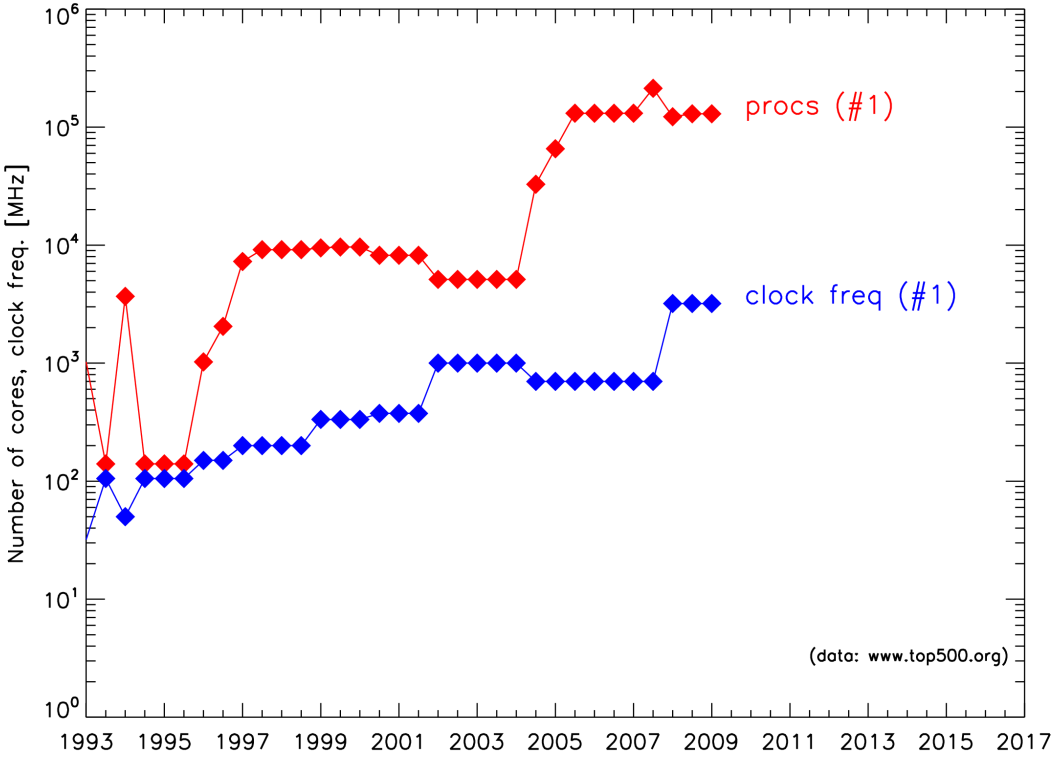


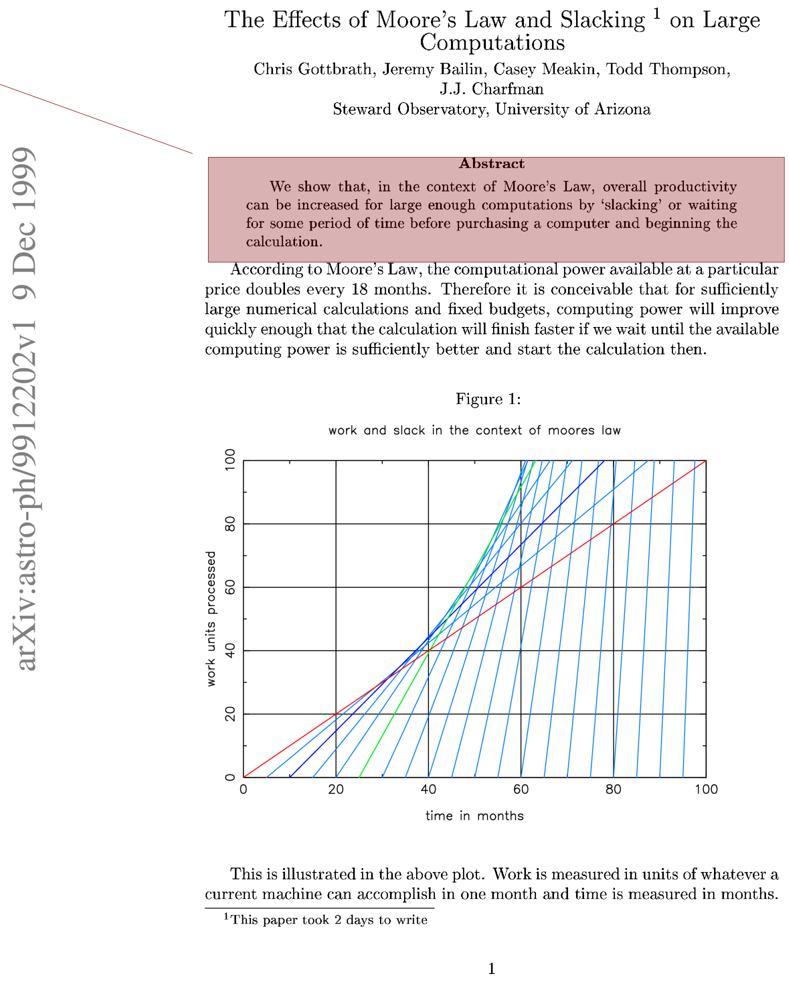
(acknowledgement: W.-X. Merkt)

Consequences (more or less serious?)

> *“We show that, in the context of Moore's Law, overall productivity can be increased for large enough computations by `slacking' or waiting for some period of time before purchasing a computer and beginning the calculation.”*

… this may have been true back in 1999, but now:





arXiv:astro-ph/9912202v1 9 Dec 1999

The Effects of Moore’s Law and Slacking [1] on Large Computations

Chris Gottbrath, Jeremy Bailin, Casey Meakin, Todd Thompson, J.J. Charfman

Steward Observatory, University of Arizona

Abstract

We show that, in the context of Moore’s Law, overall productivity can be increased for large enough computations by ‘slacking’ or waiting for some period of time before purchasing a computer and beginning the calculation.

According to Moore’s Law, the computational power available at a particular price doubles every 18 months. Therefore it is conceivable that for sufficiently large numerical calculations and fixed budgets, computing power will improve quickly enough that the calculation will finish faster if we wait until the available computing power is sufficiently better and start the calculation then.

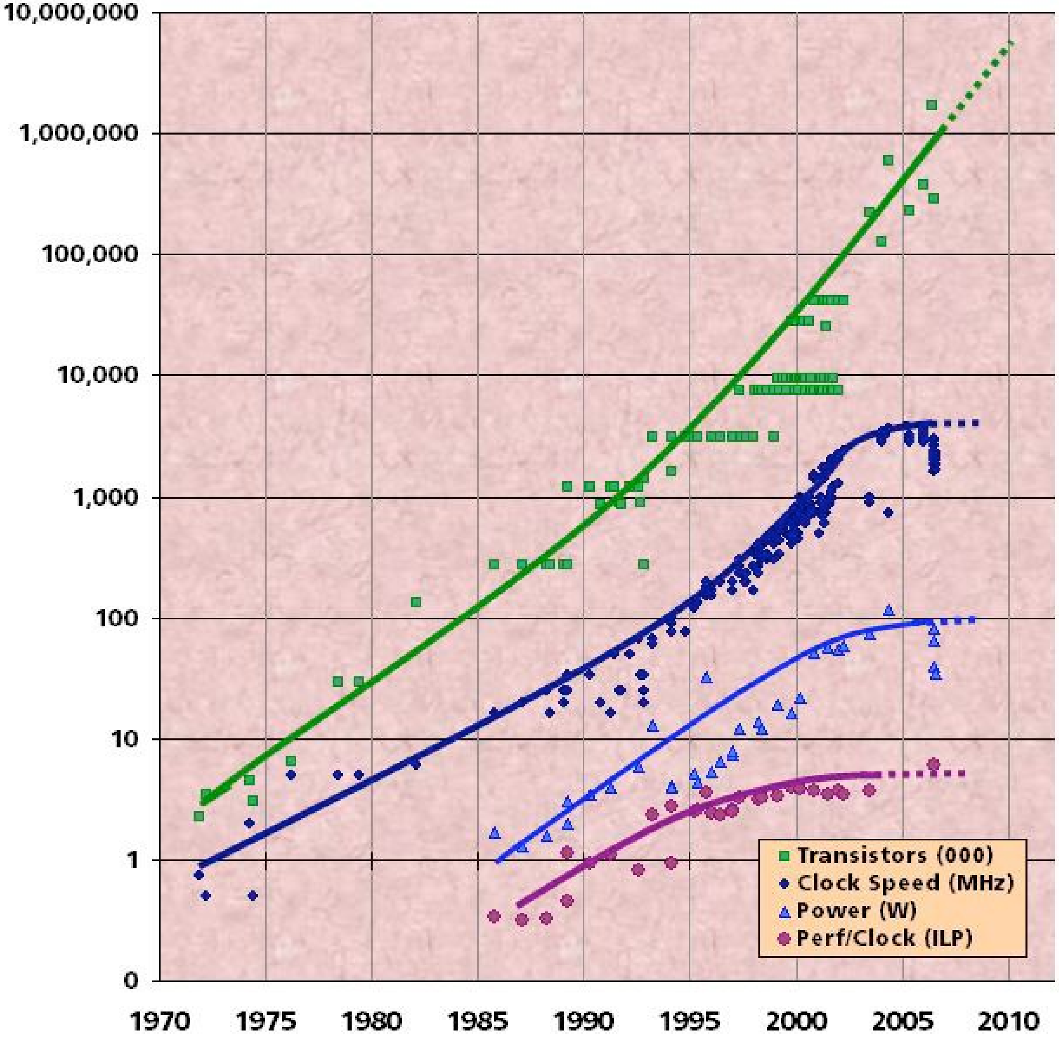
Figure 1:

This is illustrated in the above plot. Work is measured in units of whatever a current machine can accomplish in one month and time is measured in months.

[1] This paper took 2 days to write

1

Moore's law is holding – needs reinterpretation



From Herb Sutter<hsutter@microsoft.com>

Moore’s law is holding, in the original terms of number of transistors

- Transistors on an ASIC still doubling every 18 months at constant cost
- 15 years of *exponential* clock rate growth has ended

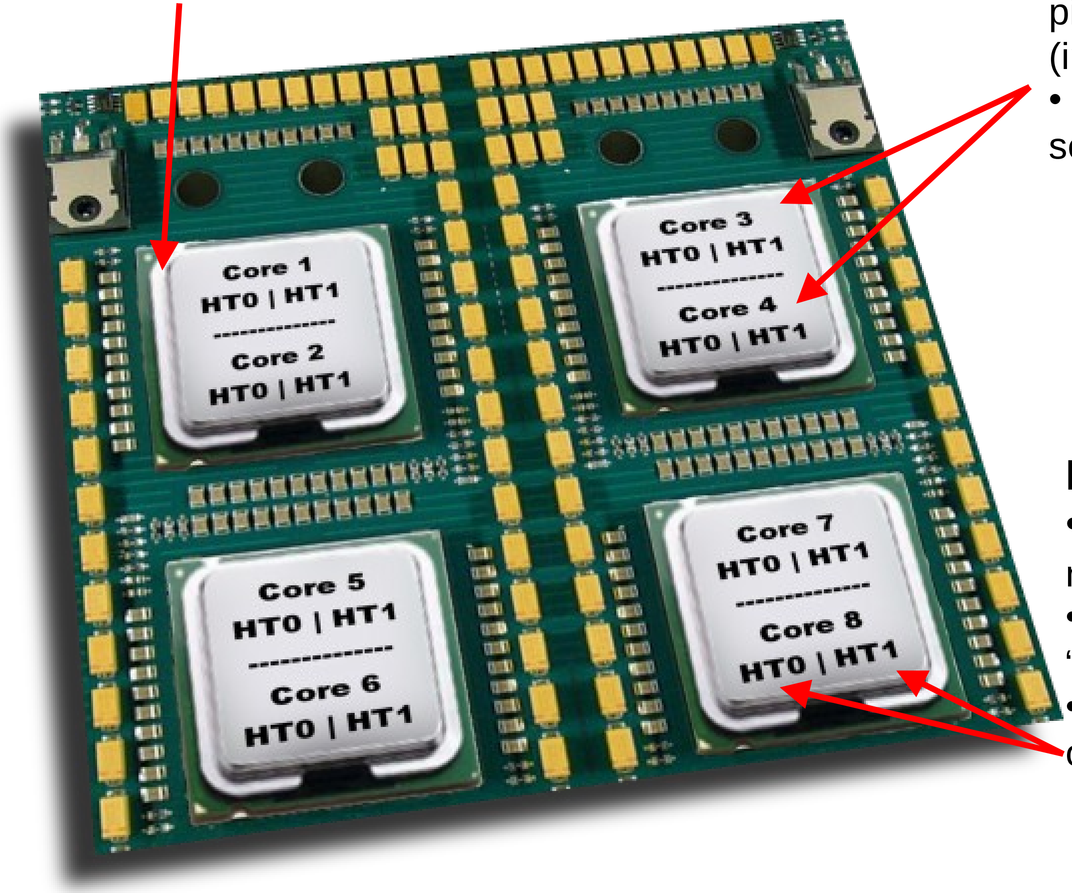
Moore’s Law reinterpreted

- Performance improvements are now coming from the increase in the number of cores on a processor (ASIC)
- #cores per chip doubles every 18 months *instead* of clock
- 64-512 threads per node will become visible soon

Some Terminology

CPU Socket

The connector linking the motherboard to the CPU



Multi-core CPU

- Combining two or more cores into a single die
- Each core implements independent execution, pipelining and multithreading unit (independent registers,…)
- Cores share a single coherent cache or have separate caches

Hyper-Threading

- Duplicates certain sections of the processor, but not the main execution resources
- The processor pretends to be two or more “logical” processors to the host operating system
- The host operating system allows to schedule two or more threads

Million–core systems on the horizon

Current Status (10k-300k cores)

- BGP@Juelich: 295K
- BGL@LLNL: 200K
- BGP@ANL: 160K
- XT5@ORNL: 150K

Preview 2011– 2013: Sustained Petaflop Systems (1M cores)

- Blue Waters – IBM Power7, NCSA – 2011
- Sequoia – IBM BlueGeneQ, LLNL – 2011
- Riken Project – 2012

Implications for applications

Sequential applications O(1 cores)

- no trivial performance progress by faster cores (from desktop to HPC applications)
- memory speed increases only 7% per year
- improvement of the latency of the cache architecture and memory bandwidth

Current HPC applications O(1K cores) → MPI tasks mapped to threads

"Classical" scalability not higher than O(3000); Blue Gene: O(30000)

Higher scalability in the range of O(1M cores) → more complex hardware architectures:

- many nodes with many multicore CPUs
- many cores per CPU
- many threads per core
- possibly boosted by "accelerators": GPGPU, (Cell), ...

Accelerators (GPGPU)

Why bother?

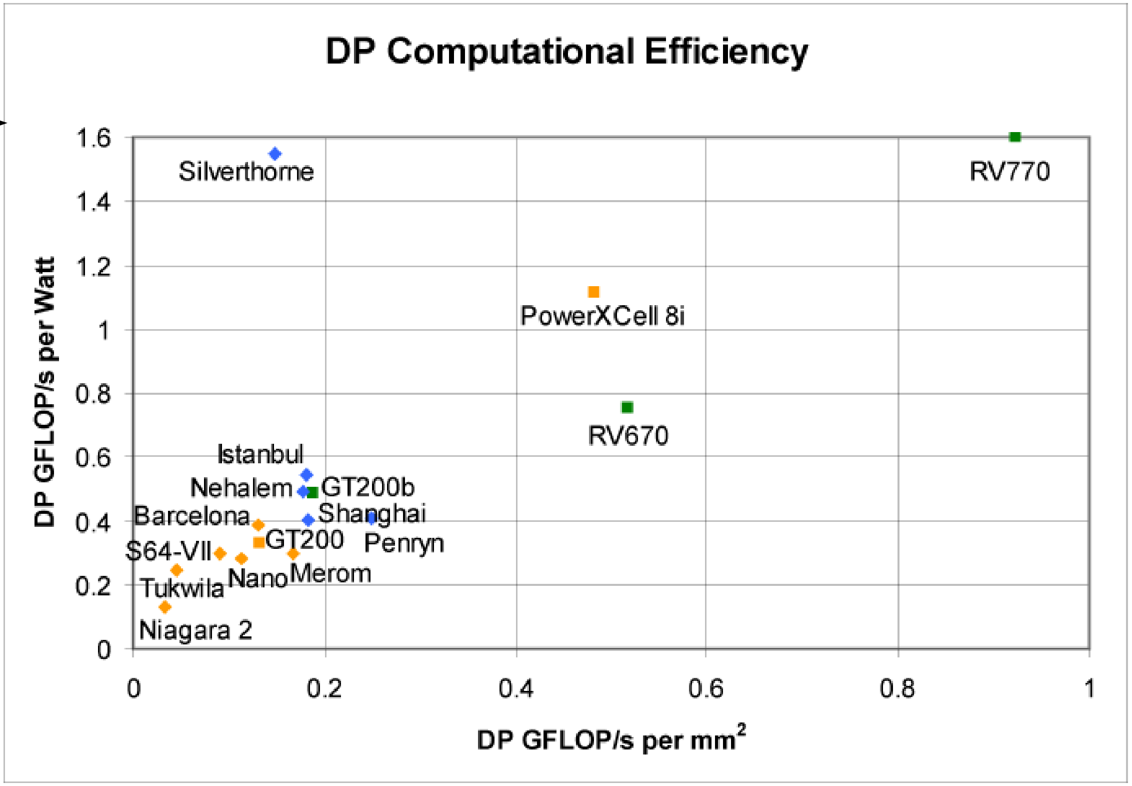
- *efficiency!*
- prospect for HPC: e.g. for boosting shared-memory nodes

Issues:

- “low level” programming model (similar to graphics, OpenGL)
- double precision available early 2010 (currently: 10x performance penalty)
- ECC memory expected only within a couple of years
- PCI bus connection to memory

Hype or maturing HPC technology?

- CUDA (NVidia), Stream (AMD/ATI)
- OpenCL (standardization)
- response of Intel, IBM?



squares: GPU, diamonds: CPU

orange: 65nm, green: 55nm, blue: 45nm

(taken from: D.Kanter 2009, “Computational Efficiency in Modern Processors”, see www.realtech.com)

Towards sustained petaflop applications

HPC := significant fraction of a system is used by a single application

Questions:

- is there such a need for sustained petaflop performance?
- are we able to build/support such applications?

Application Area	2007 in TF/s	2008-2010 in TF/s	2011 in TF/s
Astrophysics and Cosmology	5-40	50-100	>500
Earth Sciences	1-5	10-100	>1000
Plasma Physics	5-30	10-50	>500
Biophysics	1-5	50-80	>1000
Fluid Dynamics		25-100	>1000
Material Science	5-10	50-100	>500
High Energy Physics	30	100	>1000

Challenges for applications

Amdahl's law →challenges in the range of 100K – 1M cores:

recall: speedup < 1/f_sequential => f_sequential < 0.00001 @100000 cores

Major problems/challenges:

- more complex parallelization strategies required (at least hybrid OpenMP/MPI)
- lack of adequate debugging, profiling and development tools
- compiler generated automatic parallelism has not been successful
- implementation of new parallel programming methods in existing large applications has not always a promising perspective
- many numerical libraries are not well adapted for extreme parallelism
- increasing times for code development (exceeding PhD timescales)
- how to overcome the “memory wall” in the nodes?

First steps towards sustained petaflop applications: status and prospects

Development of new tools, algorithms and libraries

(within BMBF initiative "HPC Software für skalierbare Parallelrechner")

- project **ISAR**: Integrierte System- und Anwendungsanalyse für massivparallele Rechner im Petascale-Bereich (with TUM, LRZ, IBM, ...)
- project **ELPA**: Eigenvalue SoLvers for Petaflop Applications (with FHI, MPI-MS, TUM, U. Wuppertal, IBM)

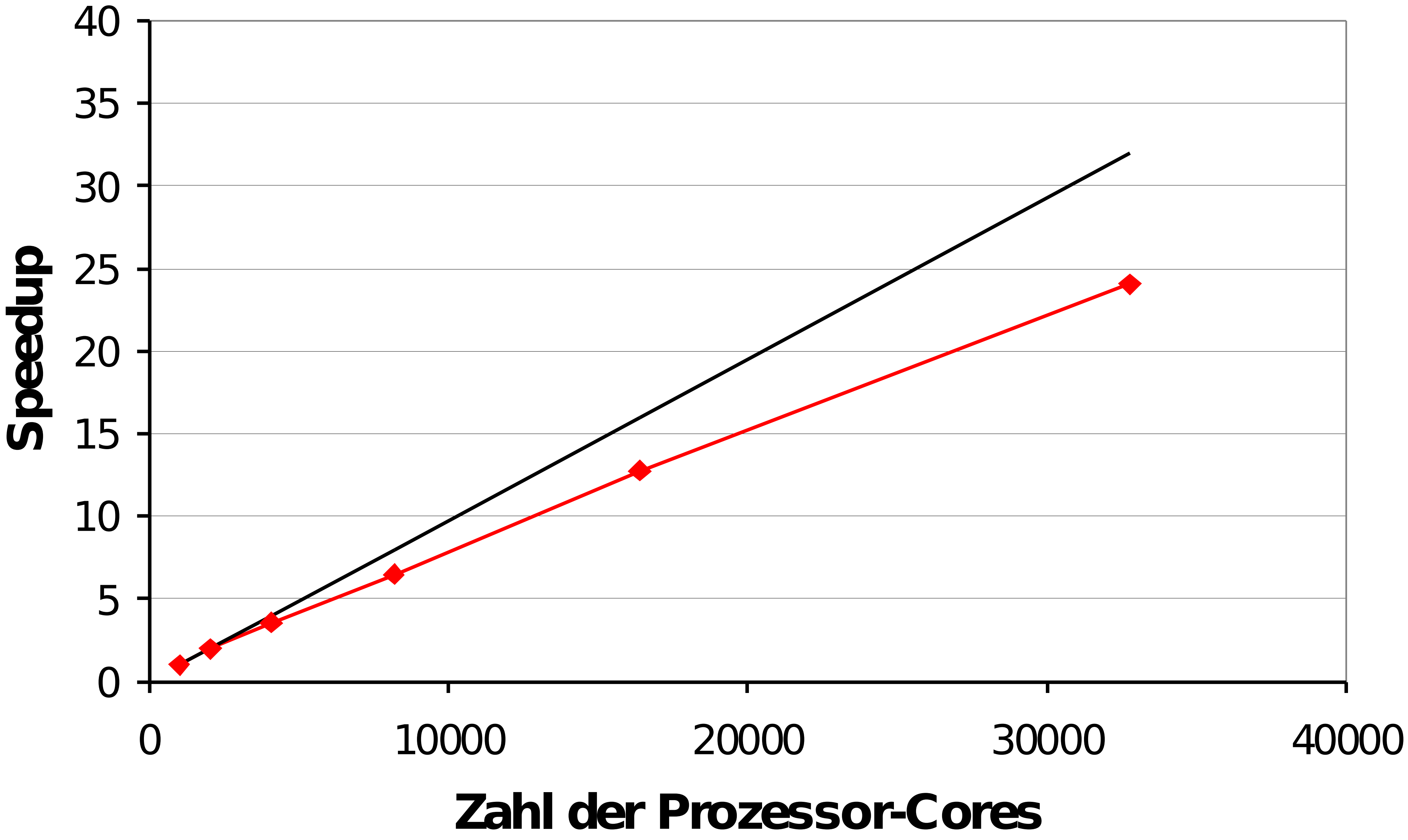
Continuous optimization efforts in close collaboration with MPG scientists and code development teams

Examples: "hyperscaling" of Gene, ORB5 (IPP), …

- *"scale-up" → weak scaling:* increased problem size (e.g. resolution)
- *"speed-up" → strong scaling:* fixed problem size (often harder to achieve used to be "for free")

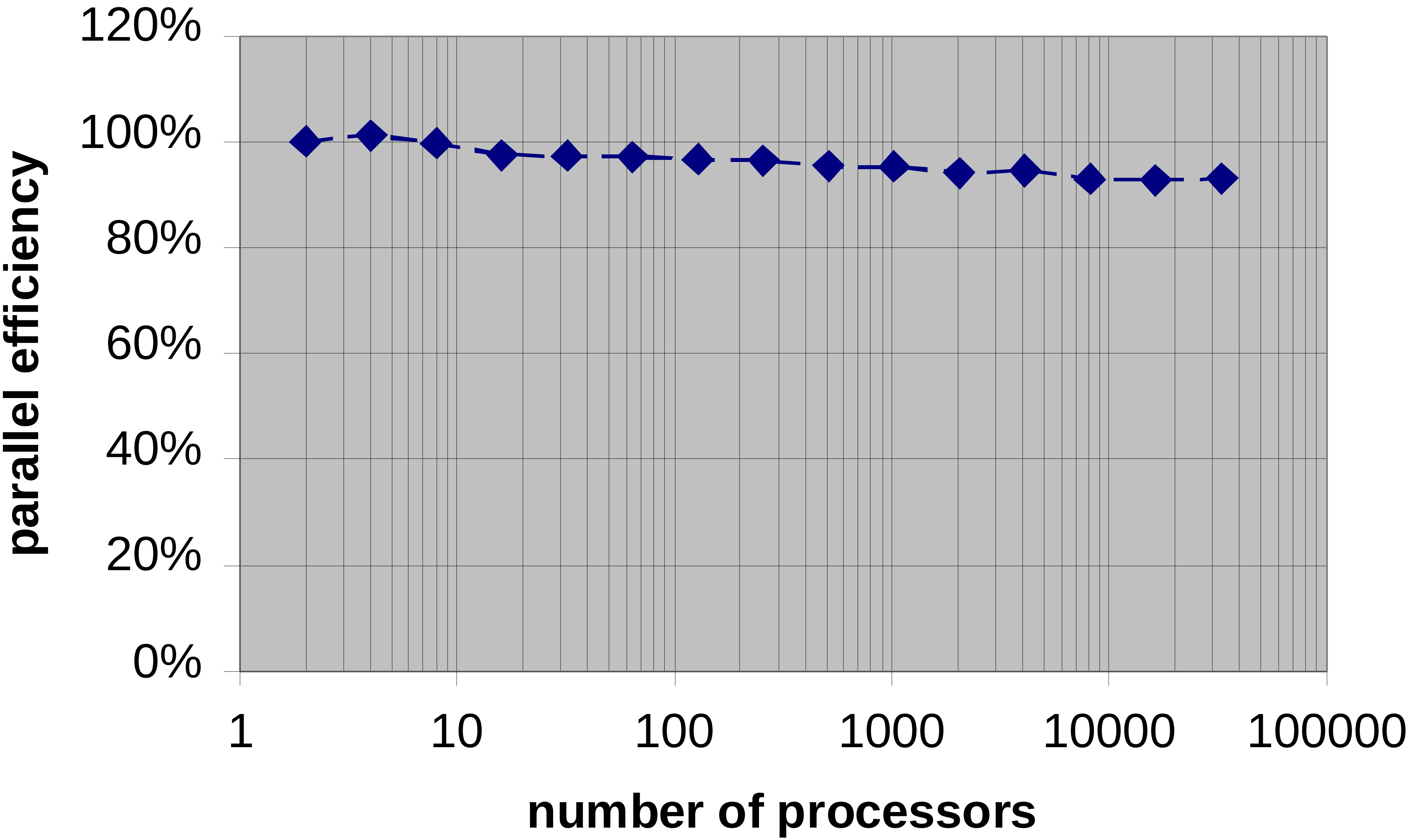
GENE Strong Scaling auf BlueGene/P

(4 Cores pro Knoten; Problemgröße ¼ TB; von 1 k bis 32 k Cores)



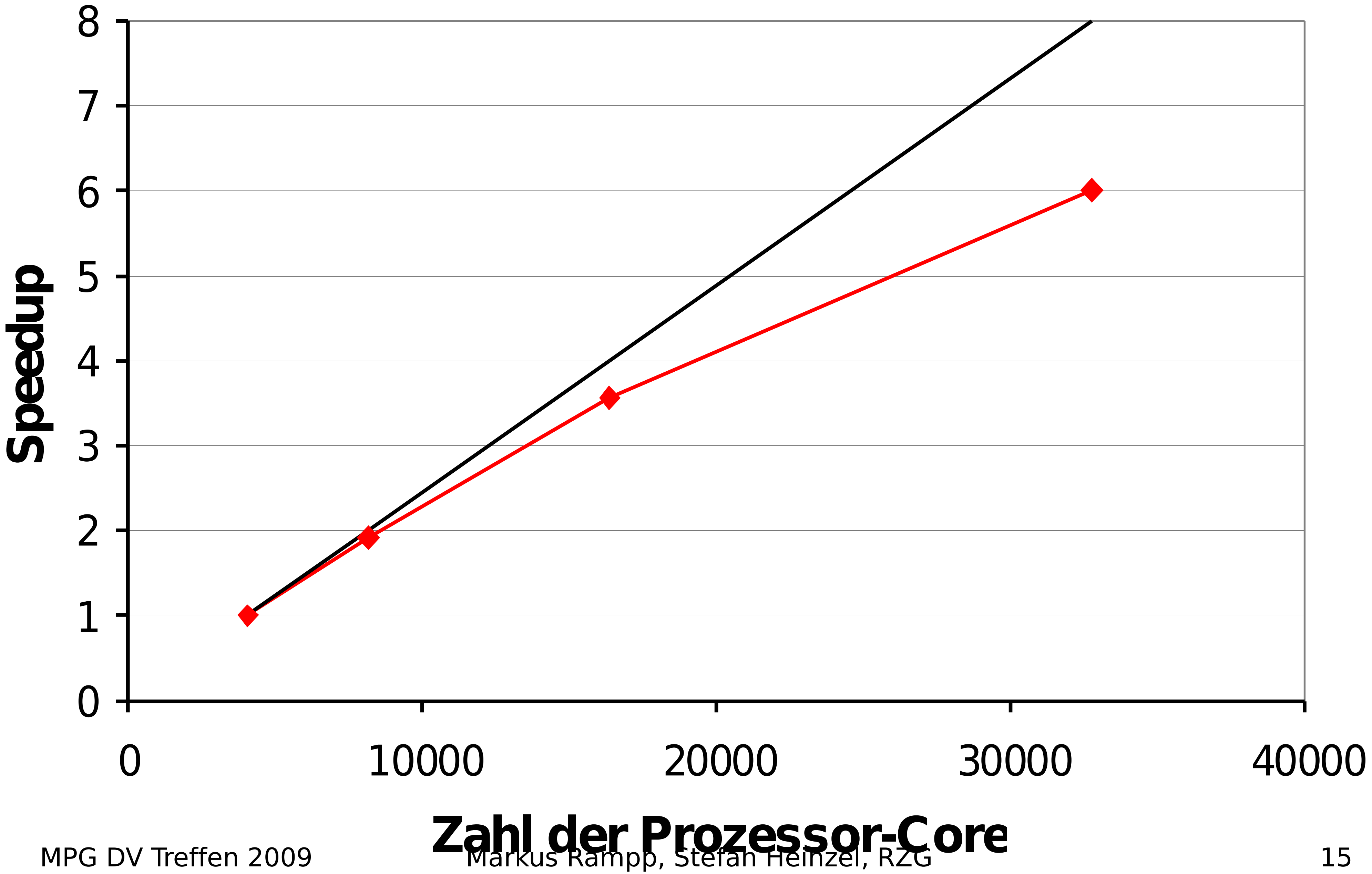
GENE Weak Scaling auf BlueGene/L

(Problemgröße steigend von 2 bis 32 k Cores)



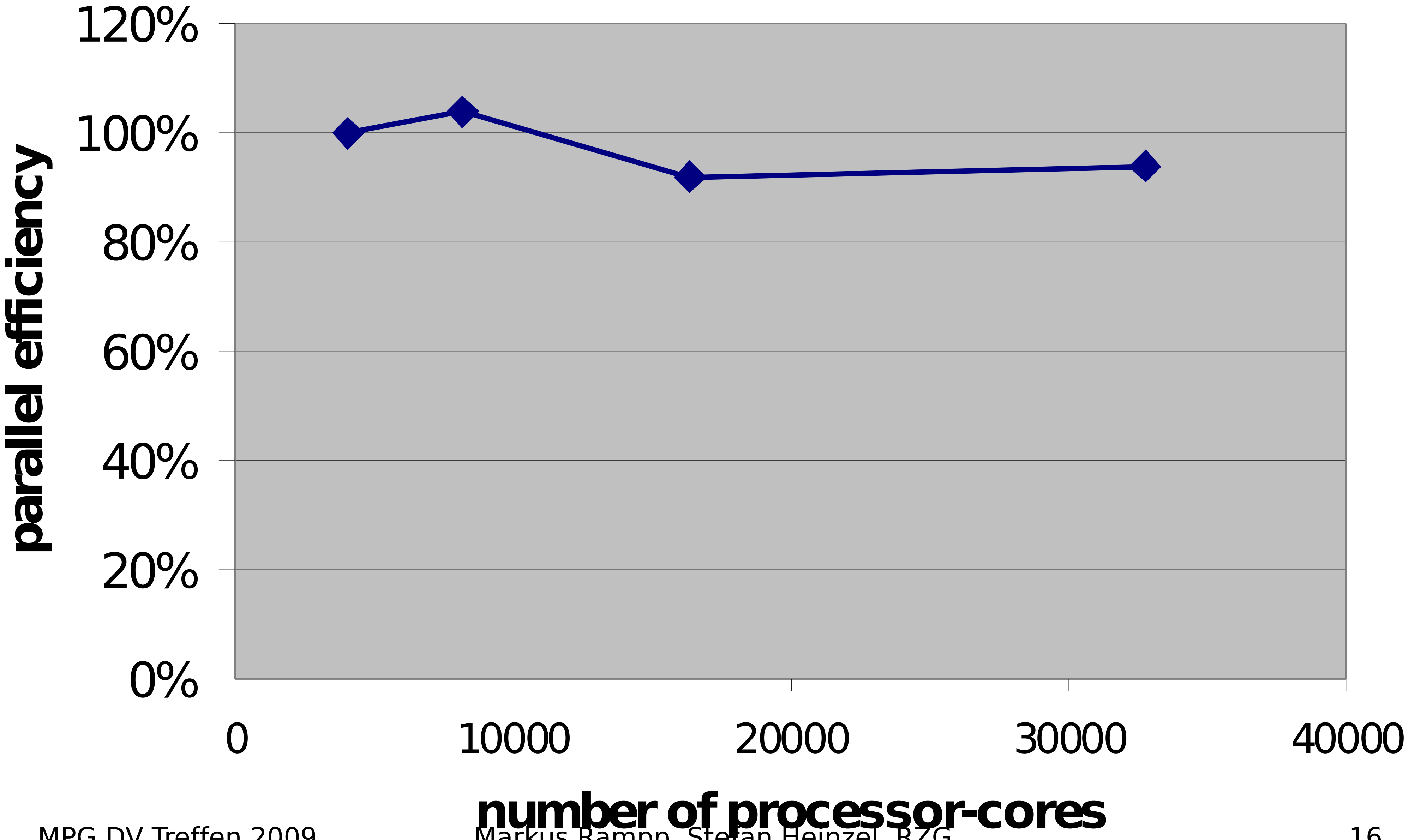
ORB5 Strong Scaling auf BlueGene/P

(4 Cores pro Knoten; Problemgröße ca 2 TB; von 4k bis 32 k Cores)

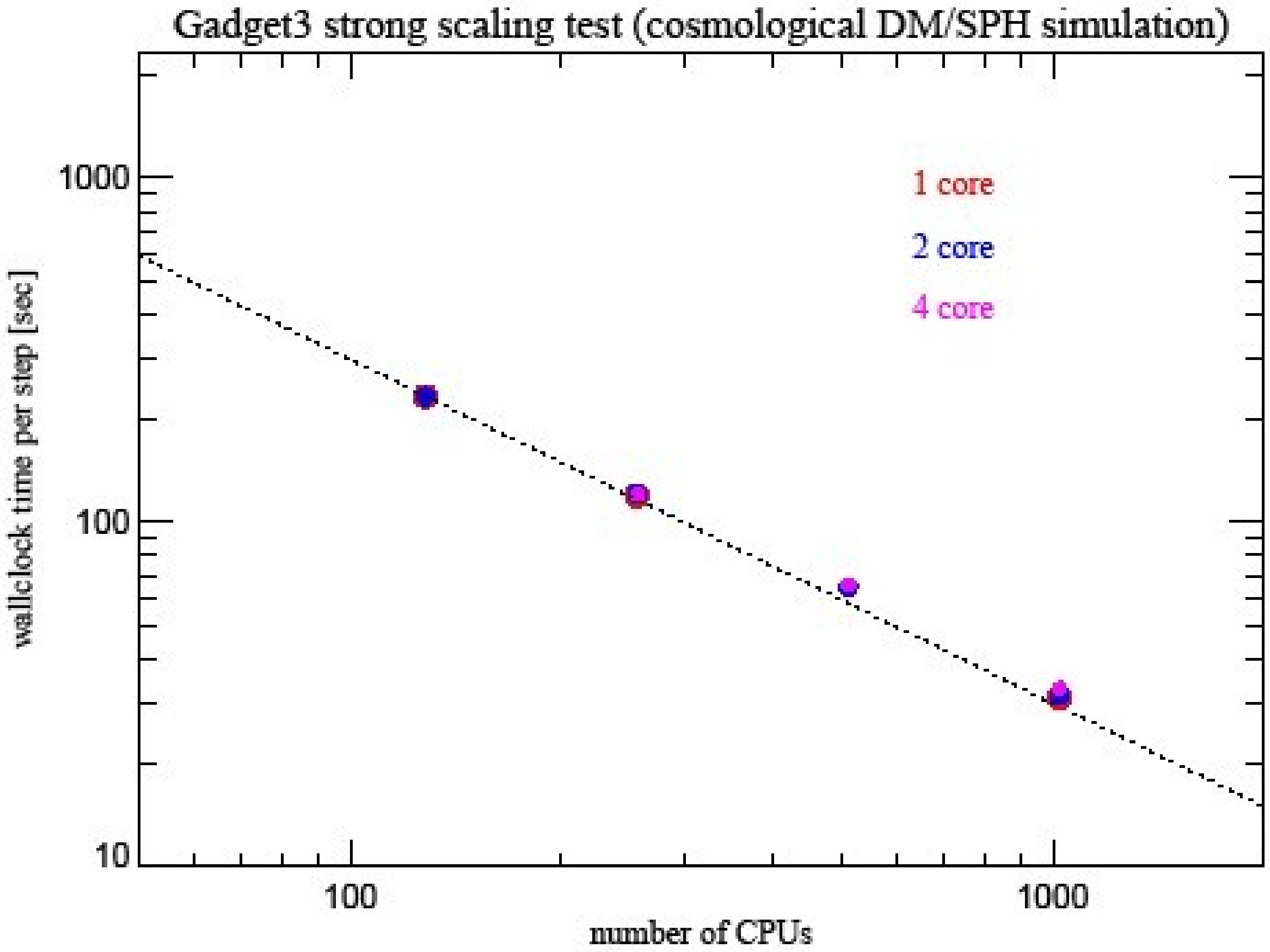


ORB5 Weak Scaling auf BlueGene/P

(4 Cores pro Knoten; Problemgröße steigend von 4k bis 32 k Cores)



Preliminary Gadget-3 Scaling on BlueGene/P



Analysis & Visualization of large-scale data

Central soft and hardware infrastructure for visualization:

→http://www.rzg.mpg.de/visualisation

motivation for centralization (at RZG):

- *Data management*: HPC simulation data (TB … PB) remain at RZG
- Establishment of *persistent „know-how" and services* for MPG
- *Hosting* of specialized hardware: Linux-cluster + GPUs (2010)

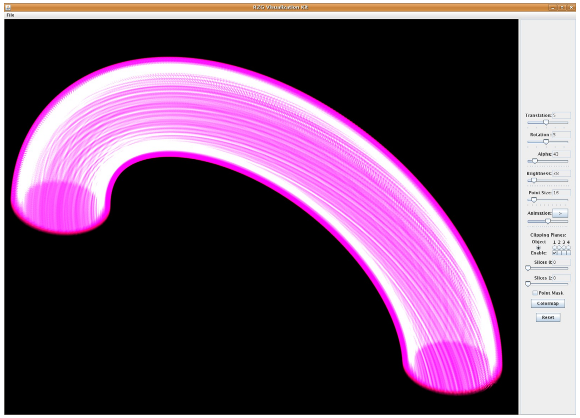
technical prerequisite:

- efficient *"Remote Visualisation"* via LAN and WAN(!)

RZG provides support for:

- adoption, configuration, application of visualization and data analysis tools: e.g. *VisIt*, *ParaView*, *Amira/Avizo*, etc.
- instrumentation of major simulation codes (I/O, data formats, etc.)
- individual (demanding) visualization projects: dedicated support in close collaboration with MPG scientists

Example visualization projects



Translation: 5

Rotation : 5

Alpha: 43

Brightness: 38

Point Size: 16

Animation: >

Clipping Planes:

Object ● 1 2 3 4 ○○○○

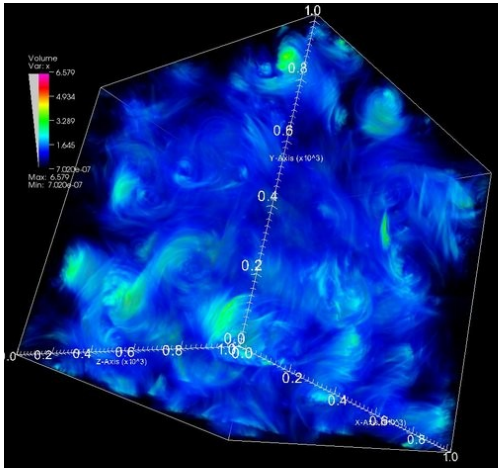
Enable: ☑☐☐☐

Slices 0: 0

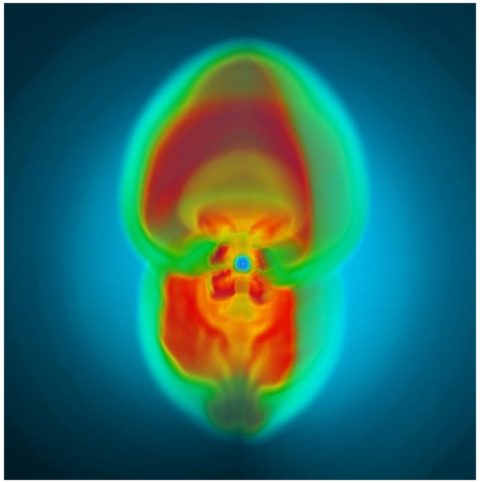
Slices 1: 0

☐ Point Mask

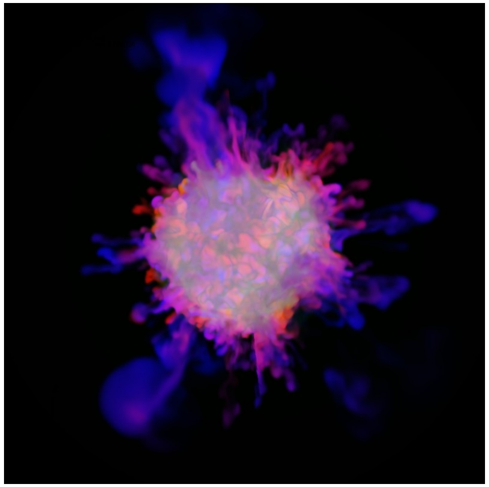
(2007/08) Entwicklung eines portablen Toolkit zur Visualisierung von Plasmafluktuationen in Tokamak Geometrie. Simulationen mit GENE, (M.Püschel, F.Jenko, IPP).



(2009) Unterstützung bei Anwendung von “VisIt” (als Avizo Ersatz) zur Analyse und Visualisierung grosser Datensätze (bis 2048^3). MHD Turbulenz Simulationen (W.C. Müller, IPP).



(2009) Visualisierung von Typ-II Supernova Simulationen (H.Th.Janka, MPA). Paralleles *volume rendering* (Datenanalyse, hochauflösende Bilder, Movie) mit VisIt. Press release *Exzellenzcluster “Universe”.*



(2009) Visualisierung von Mischungsinstabilitäten in 3D Typ-II Supernova Simulationen (E.Müller, MPA). Paralleles *volume rendering* mit verschiedenen Farbkanälen (Datenanalyse, hochauflösende Bilder, Movie) mit VisIt.

Summary

- **HPC systems with sustained Petaflop/s performance expected for 2011**

- **projected Top500 *entry level* exceeds 100 Teraflop/s in 2011**

- **sustained Petaflop/s performance requires 100k – 1M cores**

- **the large number of cores/threads together with the use of multicore architectures implies great challenges for petascale applications:**
  - huge “classical” parallelization/optimization efforts (← Amdahl's law)
  - development of new tools, algorithms and parallel libraries

- **Petabytes of generated output data require new strategies for data analysis and visualization → *“closer to the simulations”*:**
  - central visualization on dedicated hardware (instead of transfer of output data and processing on local graphics workstations)
  - or even “online” visualization (possibly on GPUs)
  - visualization services & hardware procurement at RZG