Preparing the COSMO-Model for Future HPC Architectures

Ulrich Schättler

Deutscher Wetterdienst

Research and Development

Oliver Fuhrer

Meteo Swiss

Numerical Models

Contents

- Evolving Computer Architectures
- Is the COSMO-Model suitable for Many-Core Supercomputers?
- The Swiss National HP2C Initiative and the COSMO Priority Project POMPA

Evolving Computer Architectures

Evolving Computer Architectures

Moore‘s law:
The number of transistors per square inch on integrated circuits doubles every
12 – 18 – 24 (?) months.

NEC SX-9

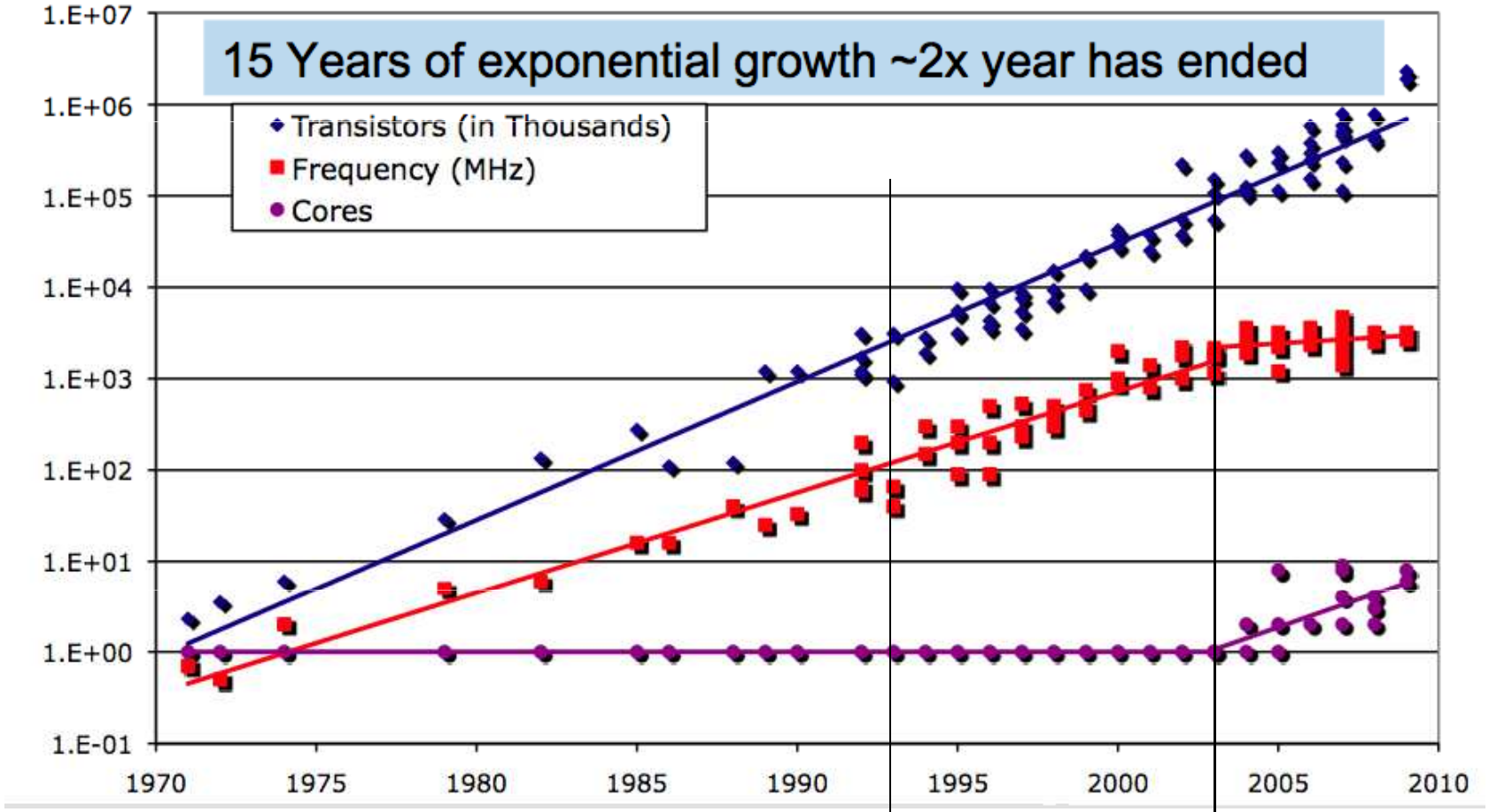
IBM pwr5

Cray T3E

Cray YMP

But this does not mean, that scientists can run the same code once and forever on all available computer platforms and get an ever increasing performance!

Moore‘s law re-visited



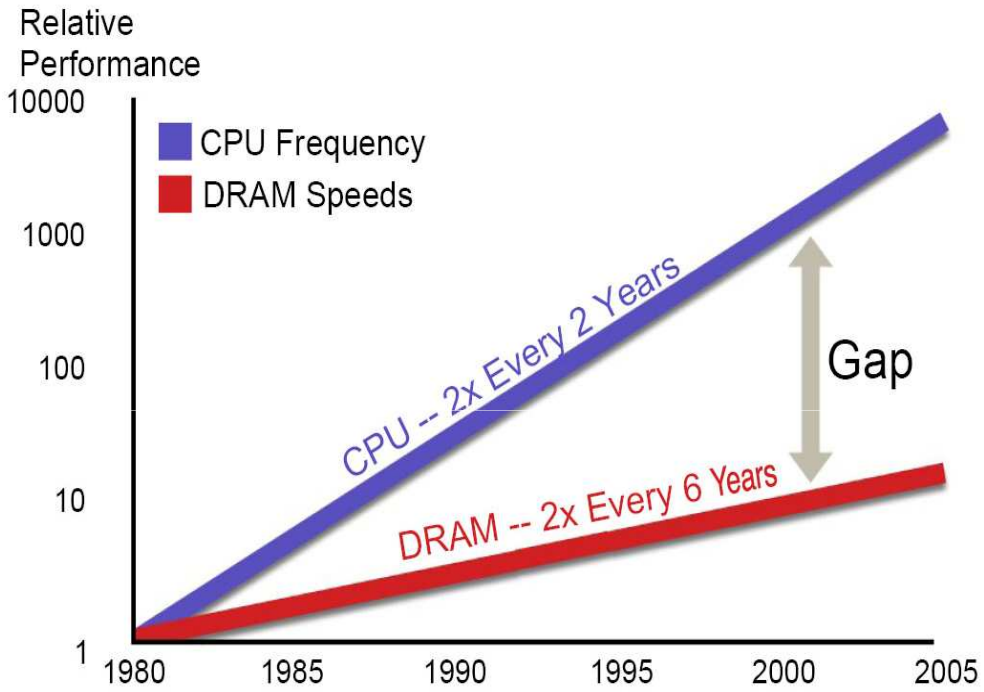
Introduced distributed memory parallel computers for further performance increase

More or less unnoticed due to distributed memory parallelization

The Memory Gap

➔ Memory speed only doubles every 6 years!

Source: Hennessy and Patterson, 2006



A practical example:

Precomputations

➔ Cray YMP: Because of lack of memory, fields could not be stored and had to be computed several times

➔ Cray T3E (and similar): Because of lack of computing power, pre-compute and store as many fields as possible

➔ Today: Because of lack of memory bandwidth, avoid precomputation and storing of fields but compute them several times!

Some HPC Facts

- **Massive concurrency** – increase in number of cores, stagnant or decreasing clock frequency
- **Less and "slower" memory per thread** – memory bandwidth per instruction/second and thread will decrease, more complex memory hierarchies
- **Only slow improvements of inter-processor and inter-thread communication** – interconnect bandwidth will improve only slowly
- **Stagnant I/O sub-systems** – technology for long-term data storage will stagnate compared to compute performance
- **Resilience and fault tolerance** – mean time to failure of massively parallel system may be short as compared to time to solution of simulation, need fault tolerant software layers

We will have to adapt our codes to exploit the power of future HPC architectures!

Is the COSMO-Model suitable for many-core supercomputers?

Problems on existing Computers

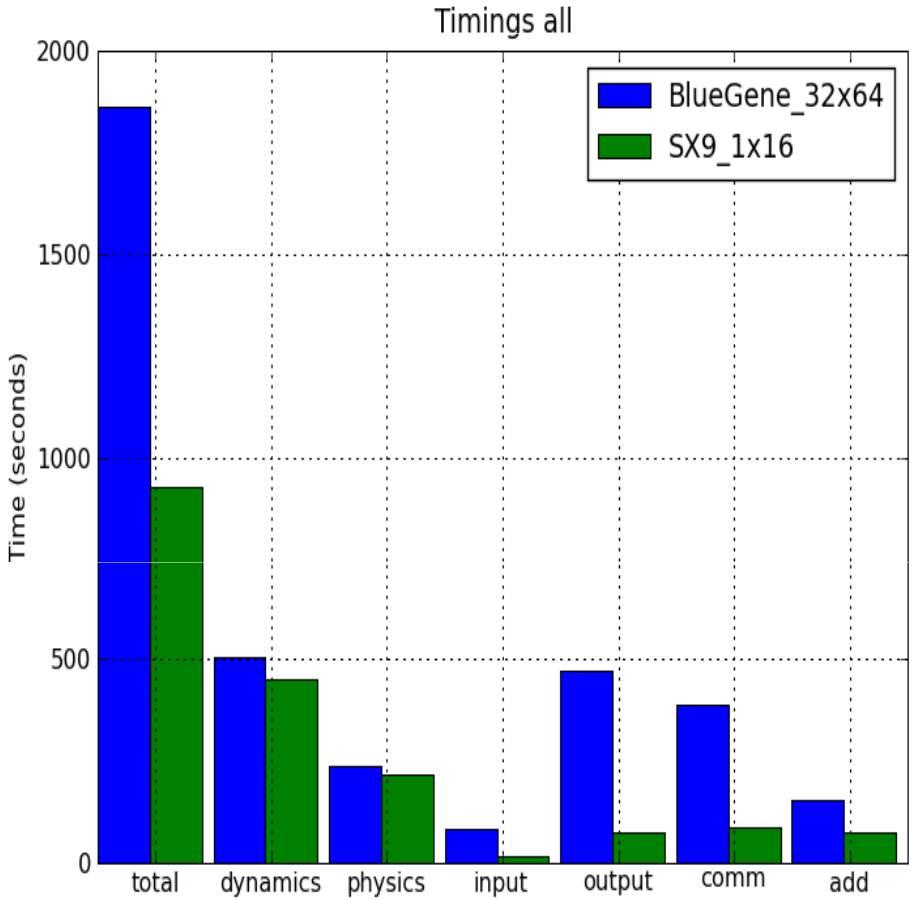
- 21 hours COSMO-DE forecast

	NEC SX-9 8 Procs	IBM pwr6 256 procs
Computations Dynamics	729.59	570.44
Computations Physics	506.18	220.45
Communications	115.61	207.69
I/O	124.43	108.40
% of I/O and Comm.	15	25

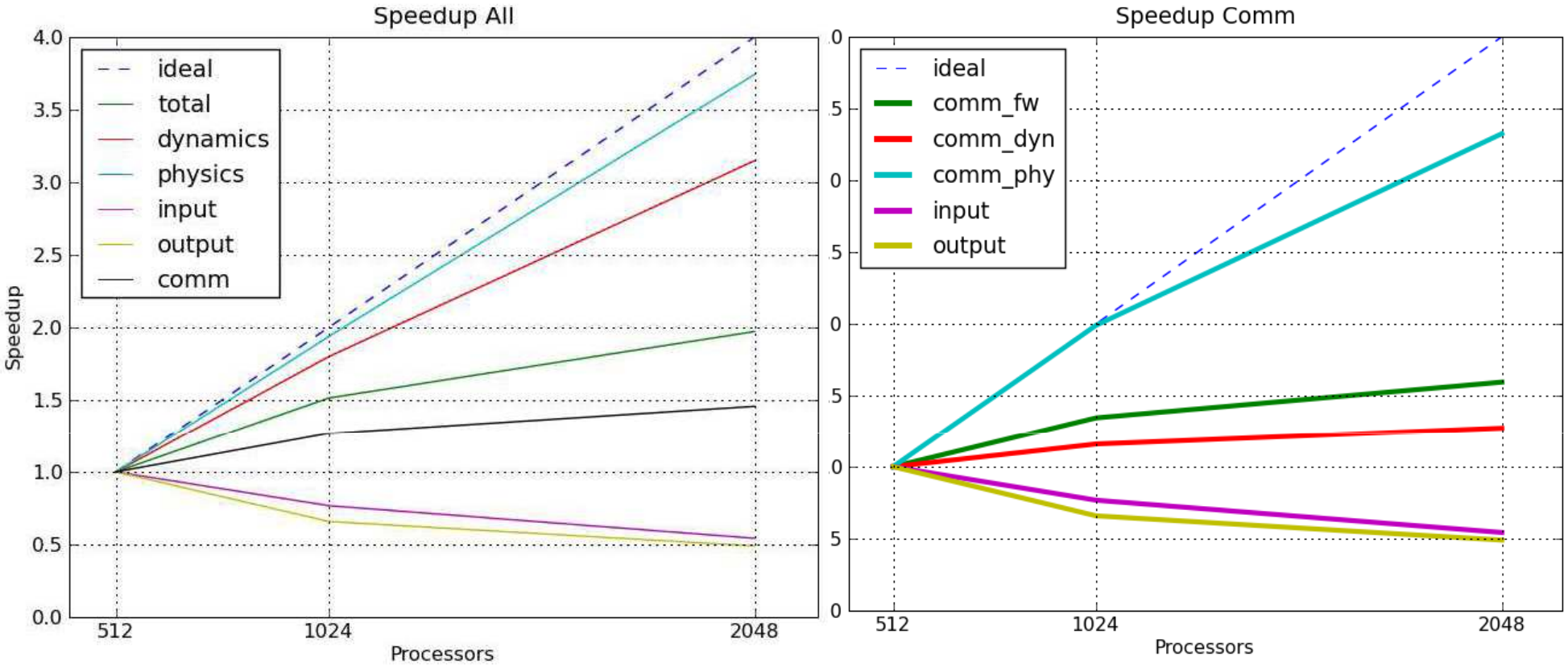
- Code efficiency
  - NEC SX-9: 13 % of peak
  - IBM pwr6: about 5-6 % of peak
  - Cray XT4: about 2-3 % of peak

Investigations from Fraunhofer / SCAI on IBM BlueGene with COSMO RAPS 4.1

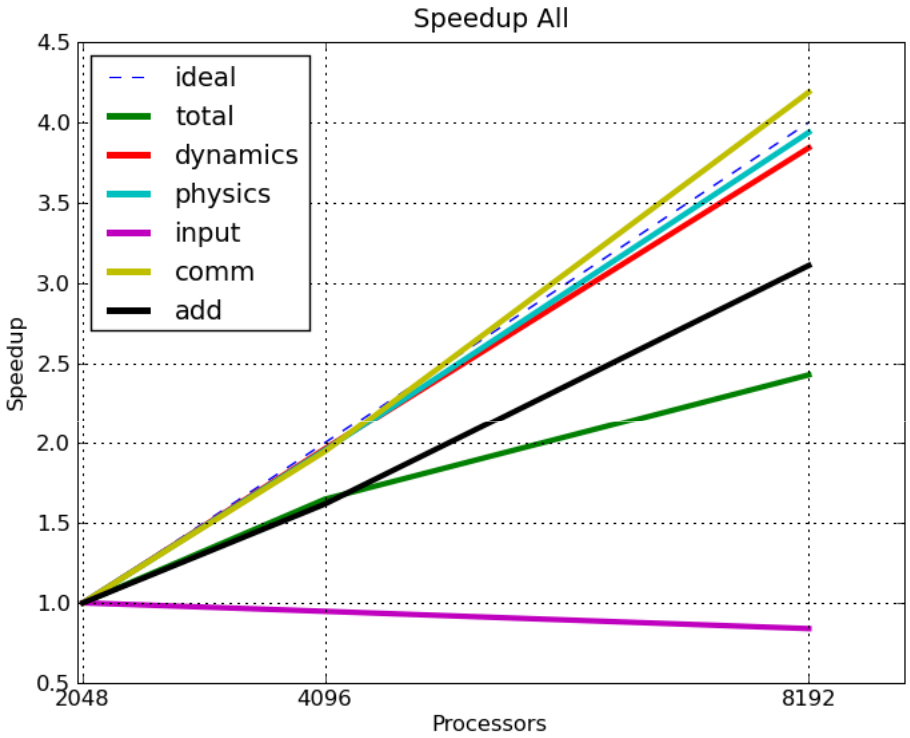
- Fraunhofer / SCAI investigated LM_RAPS_4.1 on IBM BlueGene
- The RAPS code is not as optimized on NEC SX-9 as the current operational model is
- Timings for COSMO-DE 421 × 461 × 50 grid points, dt=25 s
- Modern architectures surely are more similar to BlueGene than they are to SX-9
- There are problems in the I/O and the overall scalability!

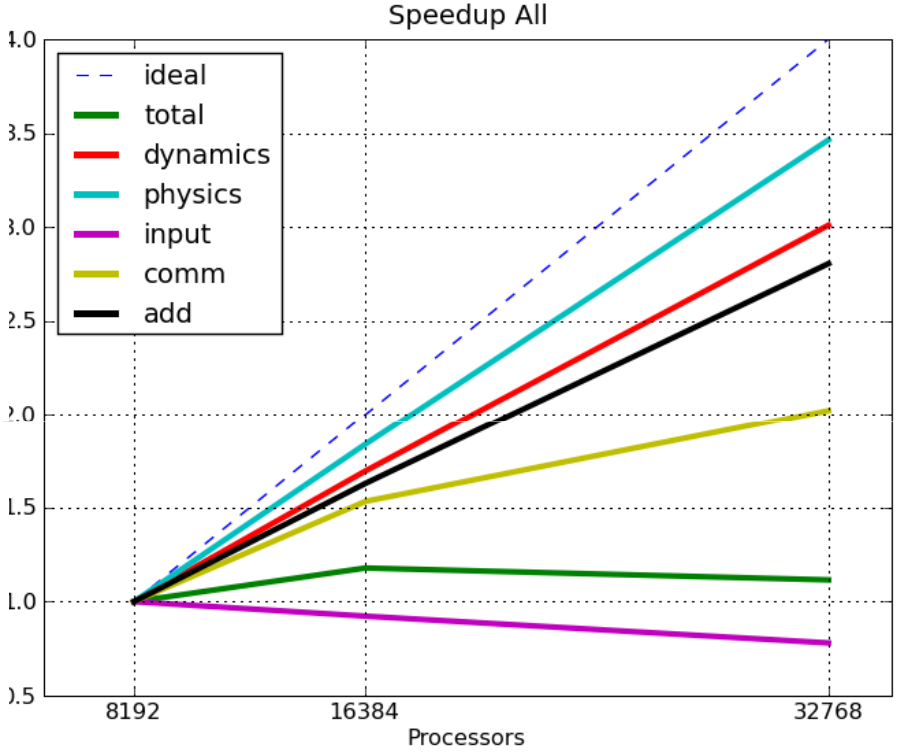


Scalability of COSMO-DE: 421 × 461 × 50, 21h



Scalability of COSMO-Europe: 1500 ×1500 × 50, 2.8 km, 3 h, no output





The Problems of the COSMO-Code …

- I/O: Accessing the disks and the global communication involved disturbes scalability heavily.
- Although the communications besides I/O are almost all local, the speedup degrades when using many processors.
- What cannot be seen on the pictures before: Although the speedup of the computations is not bad, the efficiency of the code is not satisfying:
  - NEC SX-9: 13 % of peak
  - IBM pwr6: about 5-6 % of peak
  - Cray XT4: about 2-3 % of peak
- This is because of the memory boundedness of the code

CONSORTIUM FOR SMALL SCALE MODELING

COSMO

Deutscher Wetterdienst

Wetter und Klima aus einer Hand

DWD

The Swiss national HP2C Initiative

and the

COSMO Priority Project POMPA

HP2C Initiative

- Part of national HPCN strategy
  - Hardware: mid-petaflops system by 2012
  - Infrastructure: new building in Lugano for CSCS
  - Software: HP2C initiative shall prepare software for emerging massively parallel architectures
- 10 projects
  - Cardiovascular simulation (EPFL)
  - Stellar explosions (University of Basel)
  - Quantum dynamics (University of Geneva)
  - …
- Computer scientists (the core group) at CSCS closely work with people in the projects
- Prototype hardware at CSCS

HP2C project COSMO-CLM

- “Regional Climate and Weather Modeling on the Next Generations High-Performance Computers: Towards Cloud-Resolving Simulations”
- Joint proposal C2SM / MeteoSwiss, strong involvement of CSCS / DWD / SCS
- **Tasks**
  1) Cloud resolving climate simulations (IPCC AR5)
  2) Adapt existing code (hybrid, I/O)
  3) Rewrite of dynamical core
- **Funding** ~ 900 kCHF, 3 years, 6 FTEs + core group

COSMO PP-POMPA (Lead: Oli Fuhrer)

- **P**erformance **O**n **M**assively **P**arallel **A**rchitectures
- **Timeframe** 3 years (Sep. 2010 – Sep. 2013)
- **Goal** Prepare COSMO code for emerging massively parallel architectures, especially help in implementing HP2C work into the official COSMO code
- We **need to adapt our codes** to efficiently run on current / future massively parallel architectures!
- **Great opportunity** to profit from the momentum and knowhow generated by the HP2C project and use synergies
- Consistent with goals of the **COSMO Science Plan** and similar activities in other consortia

Tasks Overview

- Performance Analysis: has already been done several times in the past, but try to bring some consistency to the results
- Investigate the memory layout (for memory bandwidth, cache efficiency, vectorization)
- Hybrid Parallelization: Can that improve scaling?
- Tackle the I/O bottleneck: check existing asynchronous I/O; investigate parallel I/O
- Redesign of the dynamical core: design a modern implementation of the dynamical core that maps more optimally onto emerging architectures
  - reduce number of memory accesses
  - investigate changing the index order
  - use iterators instead of array position computations (NOT in Fortran!)
  - merge loops to reduce the number of sweeps over the full domain
  - vectorize as much as possible (!)
- Explore GPU acceleration and possibilities of simple porting of parts of the code to GPUs

Conclusions

- HPC architectures are changing
- There is a great risk that our software will not run efficient any more
- The main problems are identified: memory boundedness, I/O, (global) communications, …
- Together with our „strong  partner“ we hope to be able to tackle the deficiencies in the next 3 years
- Accelerators (e.g. GPUs) may substantially accelerate the COSMO-Model due to a higher memory bandwidth


Thank you
very much
for your
attention