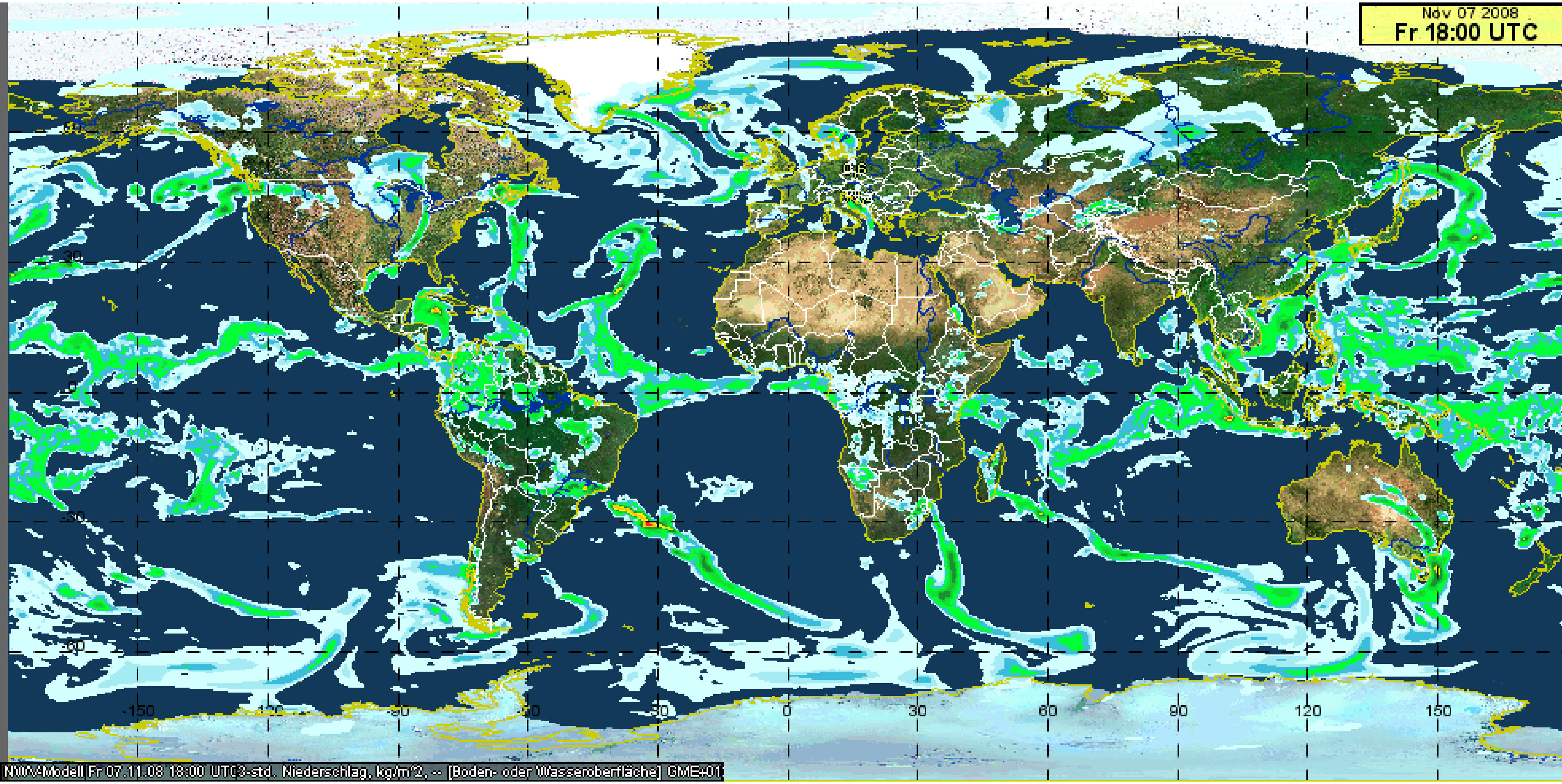
Deutscher Wetterdienst
Wetter und Klima aus einer Hand



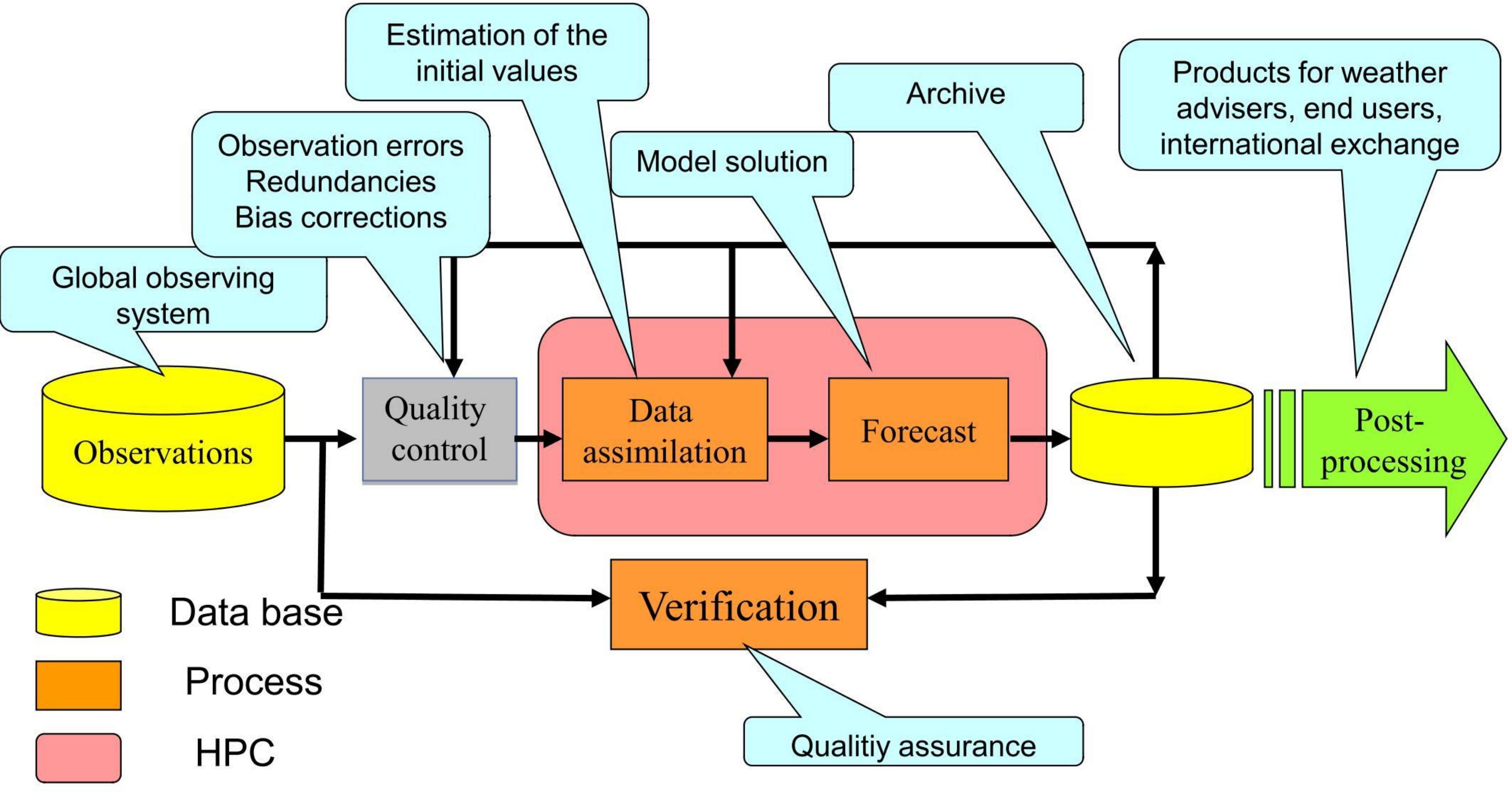
Towards probabilistic weather forecasts – new developments of the Numerical Waether Prediction System at DWD

Gerhard Adrian

Contents

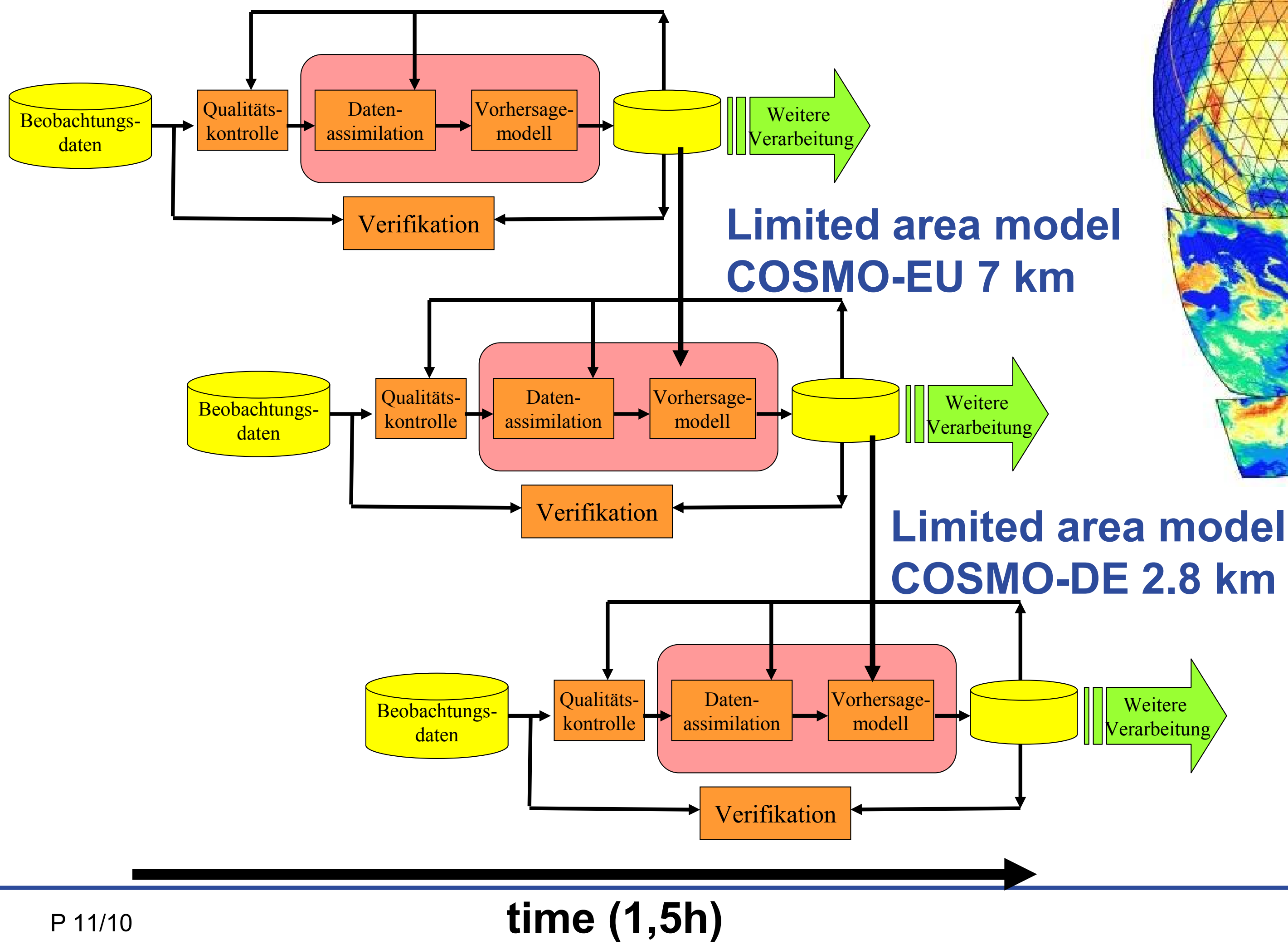
- Components of a Numerical Weather Prediction (NWP) system
- Probabilistic forecasts beyond the limit of deterministic predictability
- Need for probabilistic short range weather forecasts
- Actual developments in NWP
  - Increasing resolution - ICON
  - Data assimilation on small scales

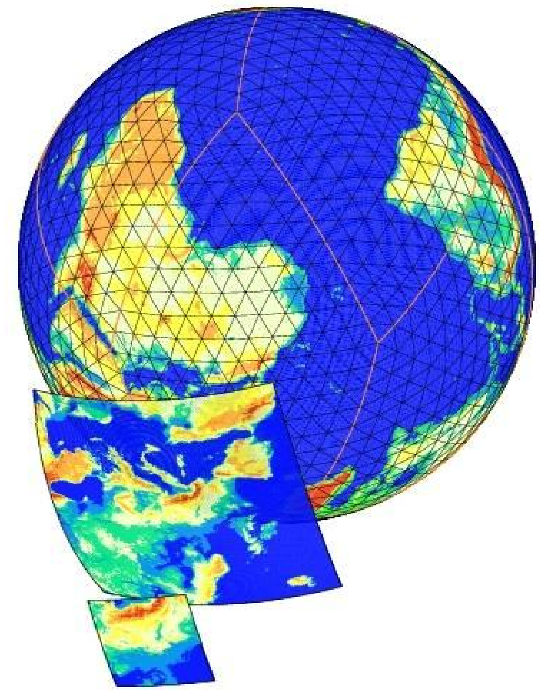
Numerical Weather Prediction as a process



Model chain

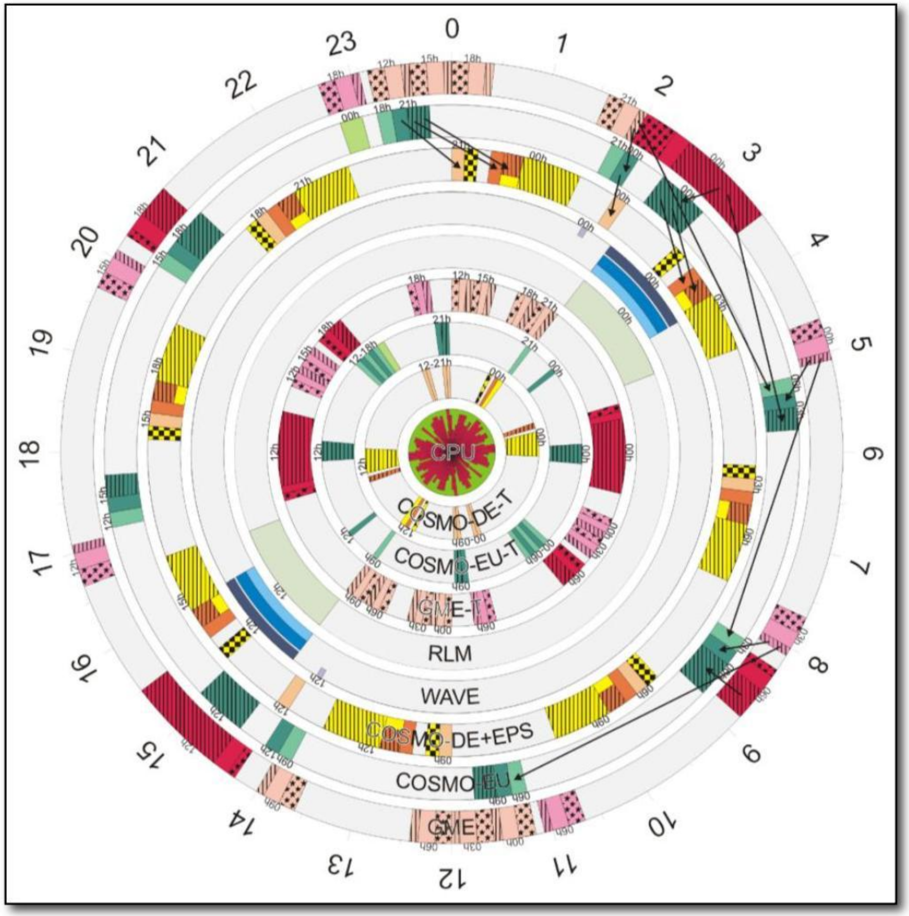
Global model GME 20 km





Numerical Weather Prediction

- ➔ Fixed production schedule
- ➔ Time critical
- ➔ Dependencies between different applications



„NWP Clock“

Probabilistic predictions beyond the limit of deterministic predictability

Scale dependent forecast systems

medium and long range	+1 d - +2W	deterministic, probabilistic	ECMWF
	+1 W - +12 W	probabilistic	
under development	+1 M - +12 M	probabilistic	
Short range	+5 h - + 72 h	deterministic, probabilistic	European NMHS
Very short range	+1h - + 24 h	deterministic	
under development	+2h - + 24 h	probabilistic	
challenge	+10min - +120min	deterministic	

Edward N. Lorenz 1917-2008

130 JOURNAL OF THE ATMOSPHERIC SCIENCES

Deterministic Nonperiodic Flow[1]

EDWARD N. LORENZ

Massachusetts Institute of Technology

(Manuscript received 18 November 1962, in revised form 7 January 1963)

ABSTRACT

Finite systems of deterministic ordinary nonlinear differential equations may be designed to represent forced dissipative hydrodynamic flow. Solutions of these equations can be identified with trajectories in phase space. For those systems with bounded solutions, it is found that nonperiodic solutions are ordinarily unstable with respect to small modifications, so that slightly differing initial states can evolve into considerably different states. Systems with bounded solutions are shown to possess bounded numerical solutions.

A simple system representing cellular convection is solved numerically. All of the solutions are found to be unstable, and almost all of them are nonperiodic.

The feasibility of very-long-range weather prediction is examined in the light of these results.

Edward N. Lorenz, J. Atmos. Sci 1963

- ➔ Finite systems of deterministic ordinary nonlinear differential equations may be designed to represent forced dissipative hydrodynamic flow...
- ➔ For those systems with bounded solutions it is found that nonperiodic solutions are ordinary unstable with respect to small modifications, so that slightly differing initial states can evolve into considerably different states.

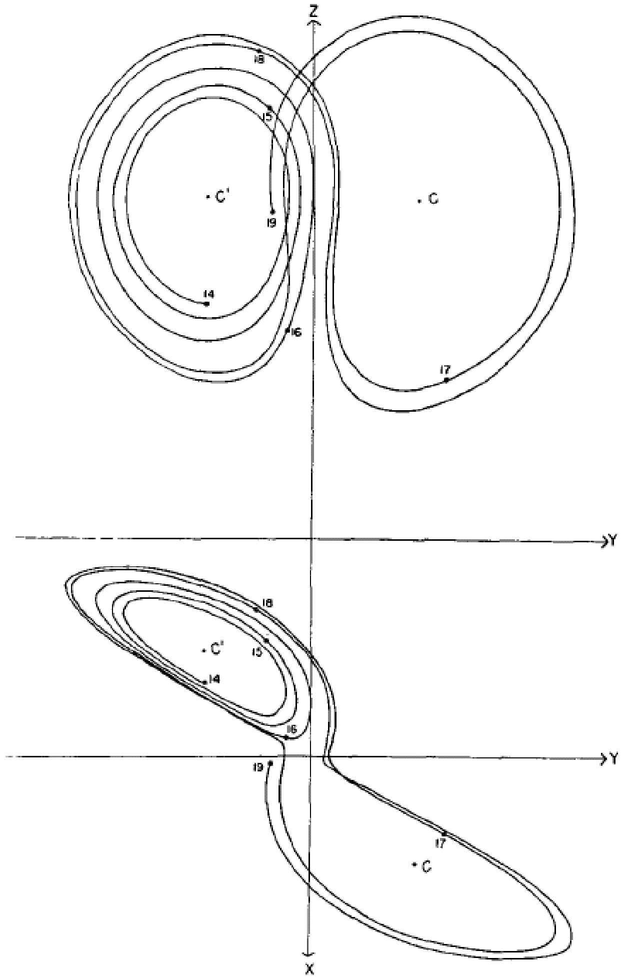


FIG. 2. Numerical solution of the convection equations. Projections on the X-Y-plane and the Y-Z-plane in phase space of the segment of the trajectory extending from iteration 1400 to iteration 1900. Numerals "14," "15," etc., denote positions at iterations 1400, 1500, etc. States of steady convection are denoted by C and C'.

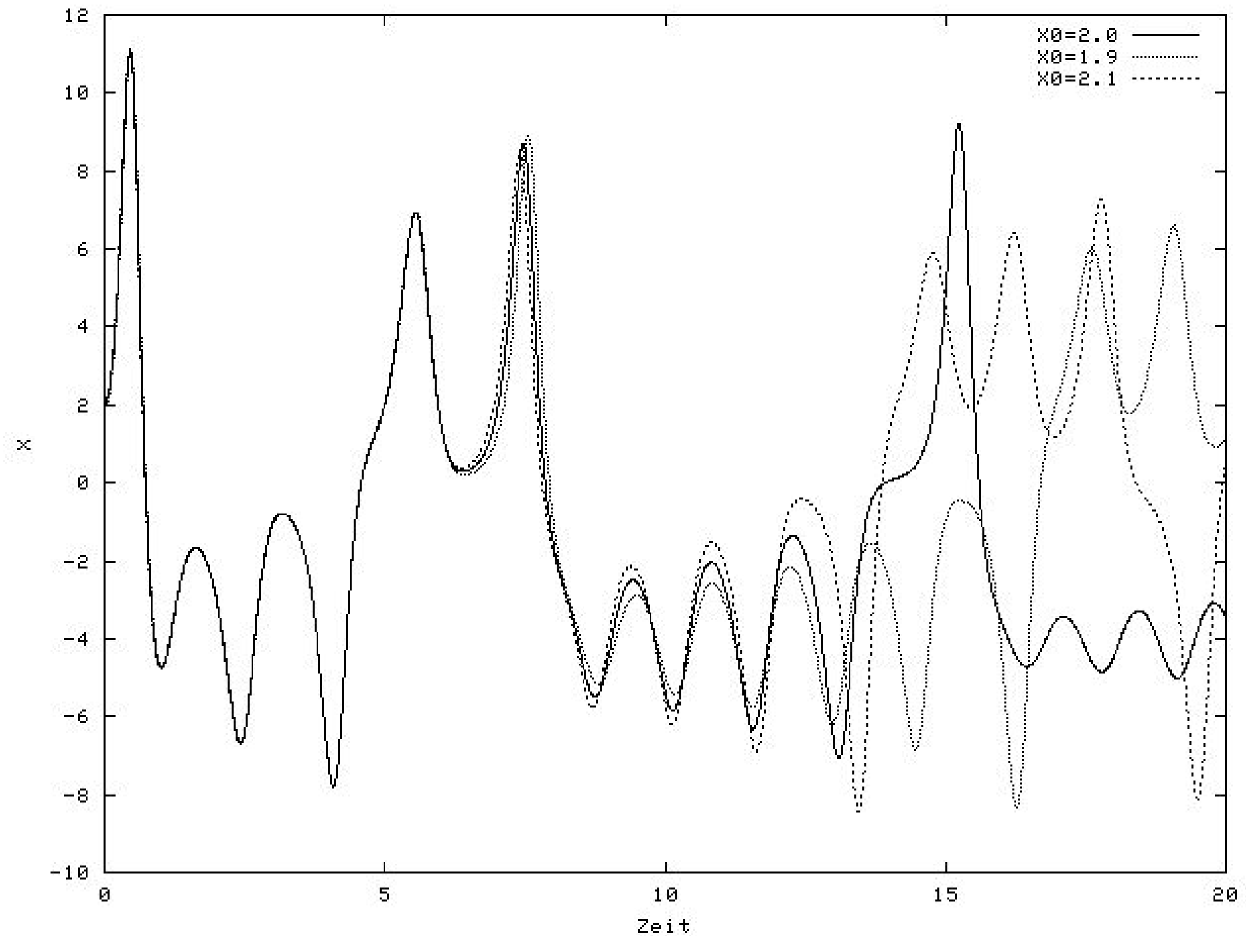
TABLE 2. Numerical solution of the convection equations Values of X, Y, Z are given at every iteration N for which Z possesses a relative maximum, for the first 6000 iterations.

N	X	Y	Z	N	X	Y	Z
0045	0174	0055	0483	3029	0117	0075	0352
0107	−0091	−0083	0287	3098	0123	0076	0365
0168	−0092	−0084	0288	3171	0134	0082	0383
0230	−0092	−0084	0289	3268	0155	0069	0435
0292	−0092	−0083	0290	3333	−0114	−0079	0342
0354	−0093	−0083	0292	3400	−0117	−0077	0350
0416	−0093	−0083	0293	3468	−0125	−0083	0361
0478	−0094	−0082	0295	3541	−0129	−0073	0378
0540	−0094	−0082	0296	3625	−0146	−0074	0413
0602	−0095	−0082	0298	3695	0127	0079	0370
0664	−0096	−0083	0300	3772	0136	0072	0394
0726	−0097	−0083	0302	3853	−0144	−0077	0407
0789	−0097	−0081	0304	3926	0129	0072	0380
0851	−0099	−0083	0307	4014	0148	0068	0421
0914	−0100	−0081	0309	4082	−0120	−0074	0359
0977	−0100	−0080	0312	4153	−0129	−0078	0375
1040	−0102	−0080	0315	4233	−0144	−0082	0404
1103	−0104	−0081	0319	4307	0135	0081	0385
1167	−0105	−0079	0323	4417	−0162	−0069	0450
1231	−0107	−0079	0328	4480	0106	0081	0324
1295	−0111	−0082	0333	4544	0109	0082	0329
1361	−0111	−0077	0339	4609	0110	0080	0334
1427	−0116	−0079	0347	4675	0112	0076	0341
1495	−0120	−0077	0357	4741	0118	0081	0349
1566	−0125	−0072	0371	4810	0120	0074	0360
1643	−0139	−0077	0396	4881	0130	0081	0376
1722	0140	0075	0401	4963	0141	0068	0406
1798	−0135	−0072	0391	5035	−0133	−0081	0381
1882	0146	0074	0413	5124	−0151	−0076	0422
1952	−0127	−0078	0370	5192	0119	0075	0358
2029	−0135	−0070	0393	5262	0129	0083	0372
2110	0146	0083	0408	5340	0140	0079	0397
2183	−0128	−0070	0379	5419	−0137	−0067	0399
2268	−0144	−0066	0415	5495	0140	0081	0394
2337	0126	0079	0368	5576	−0141	−0072	0405
2412	0137	0081	0389	5649	0135	0082	0384
2501	−0153	−0080	0423	5752	0160	0074	0443
2569	0119	0076	0357	5816	−0110	−0081	0332
2639	0129	0082	0371	5881	−0113	−0082	0339
2717	0136	0070	0395	5948	−0114	−0075	0346
2796	−0143	−0079	0402				
2871	0134	0076	0388				
2962	−0152	−0072	0426				

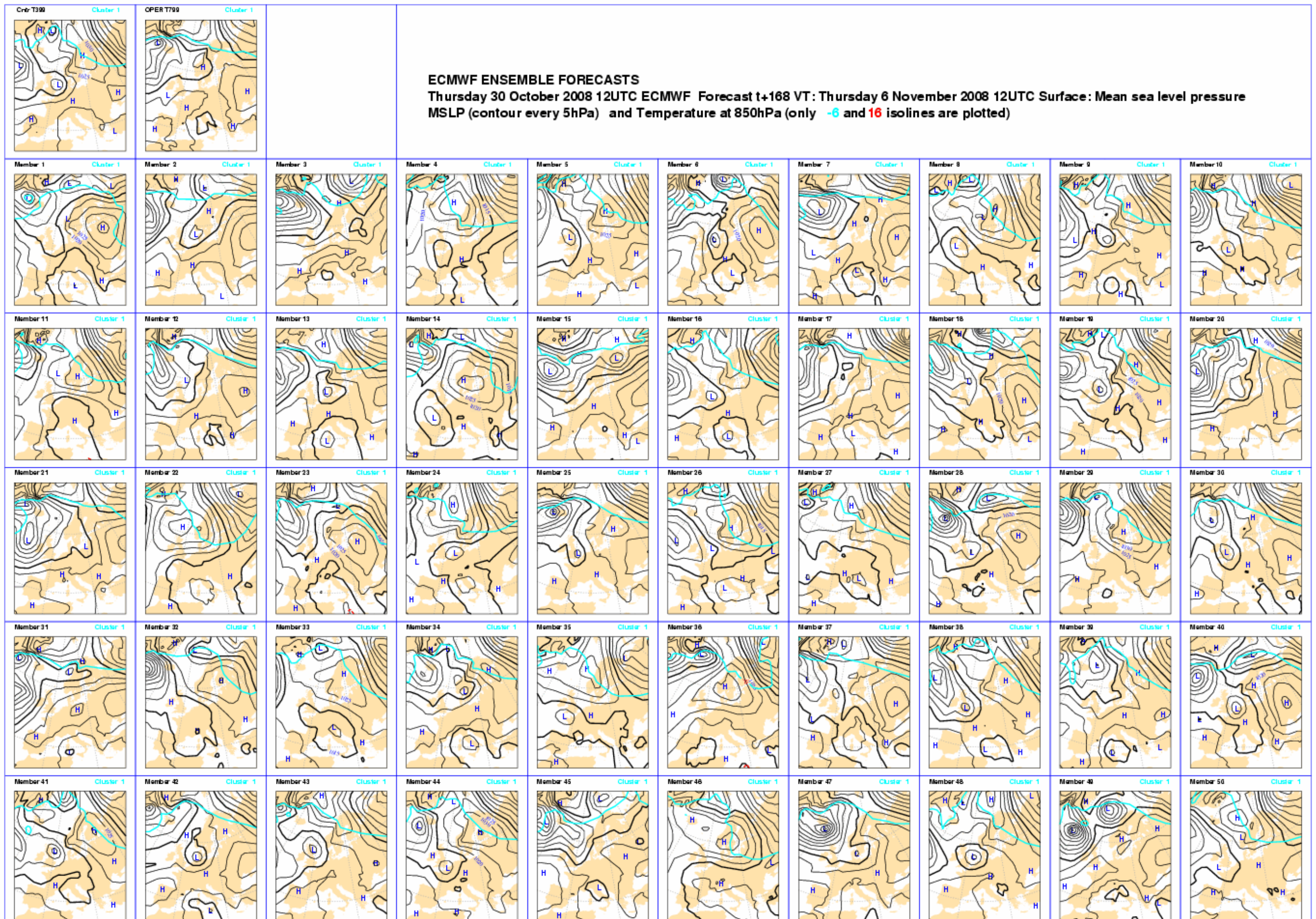
E. N. Lorenz (1963):

„The computations have been performed on a Royal McBee LGP-30 electronic computing machine. Approximately one second per iteration, aside frome output time, is required“

Limit of deterministic predictability (E. N. Lorenz)

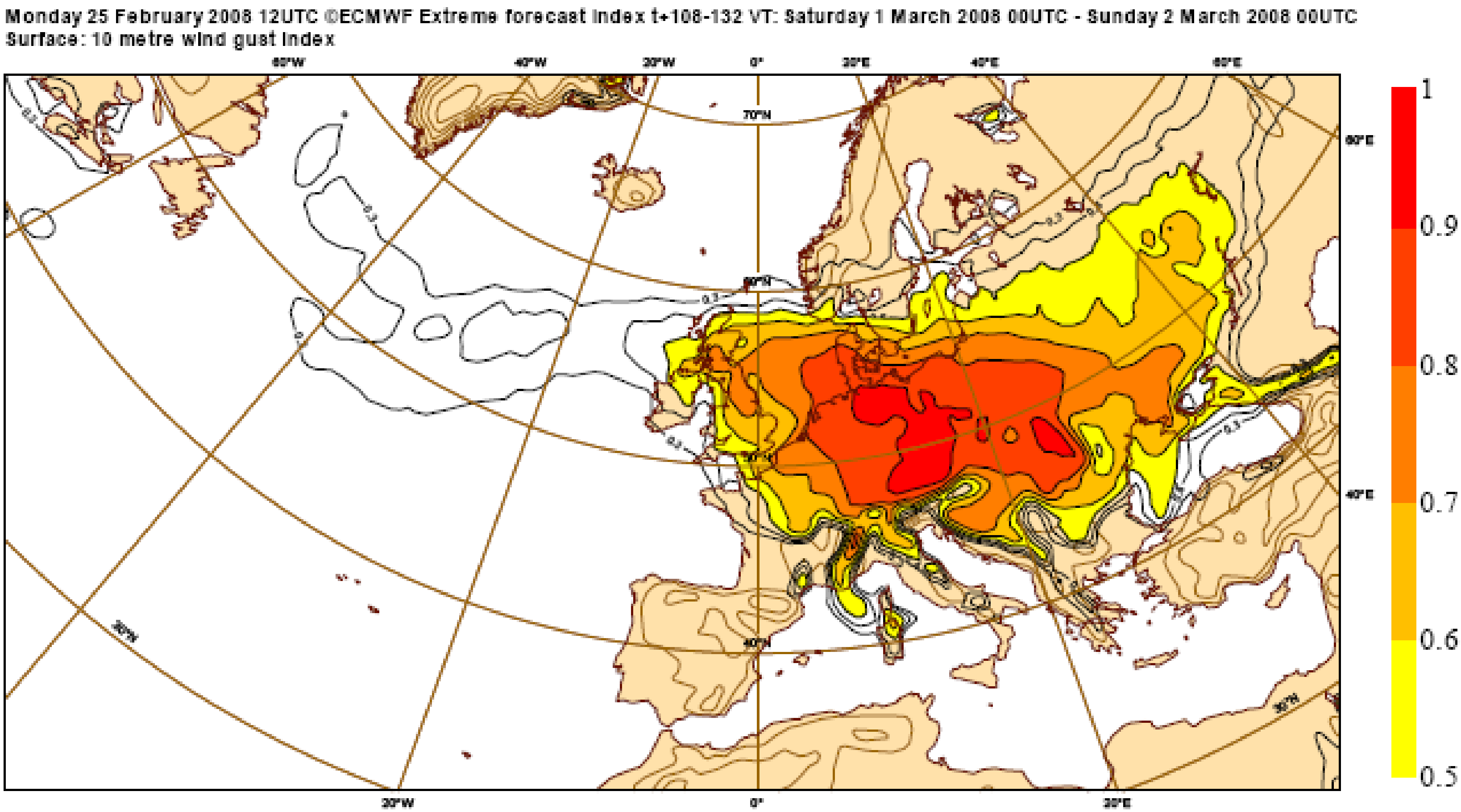


Probabilistic predictions beyond the limit of deterministic predictability by Ensembles of forecasts



Routine product of ECMWF
2 times per day
52 forecasts

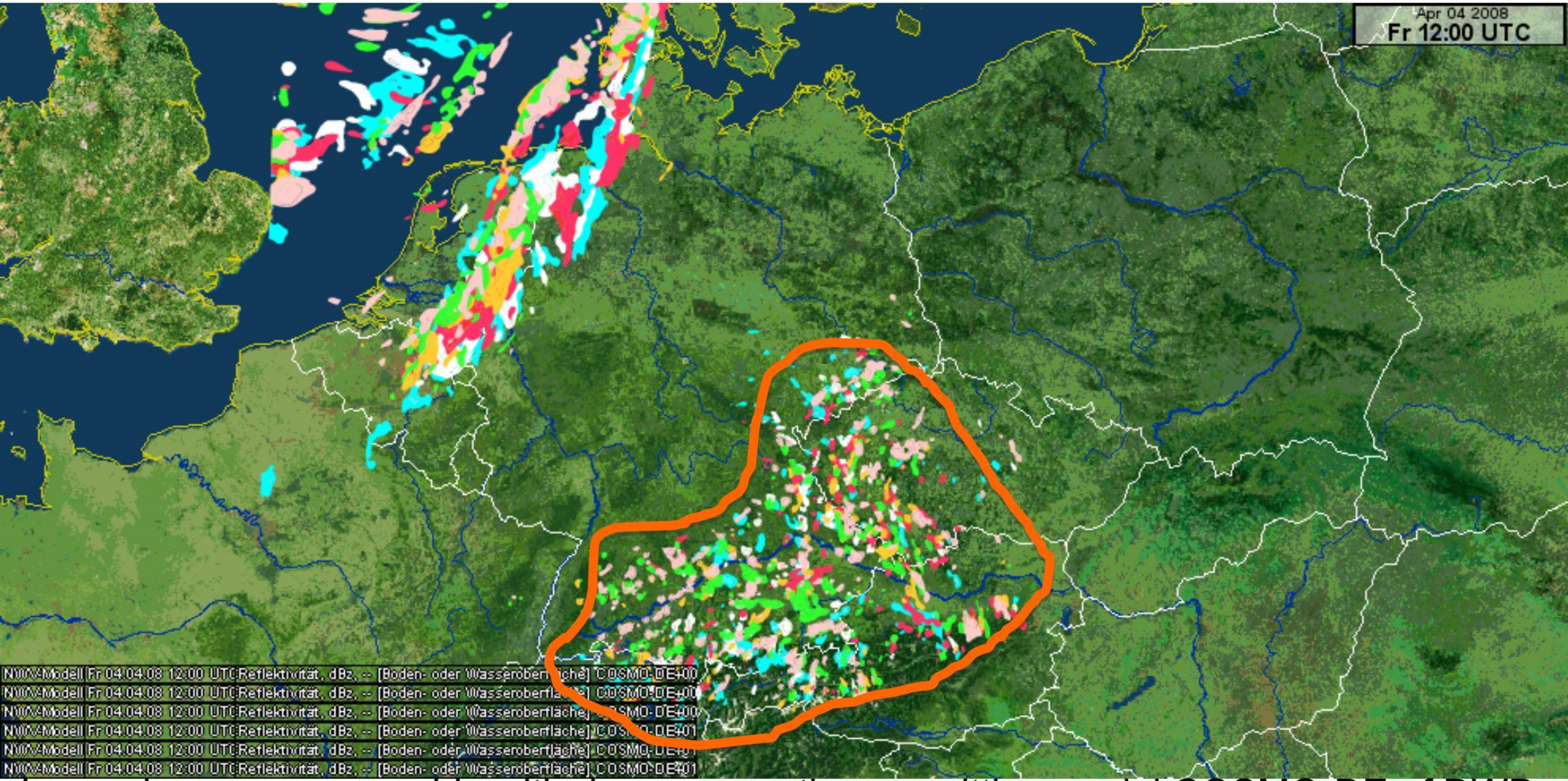
Extreme Forecast Index (EFI) 1.3.2008 (Orkan EMMA)



Routine product of ECMWF

Need for probabilistic short range numerical weather predictions

Simulated Radar reflectivity, 6 forecasts, same lead time



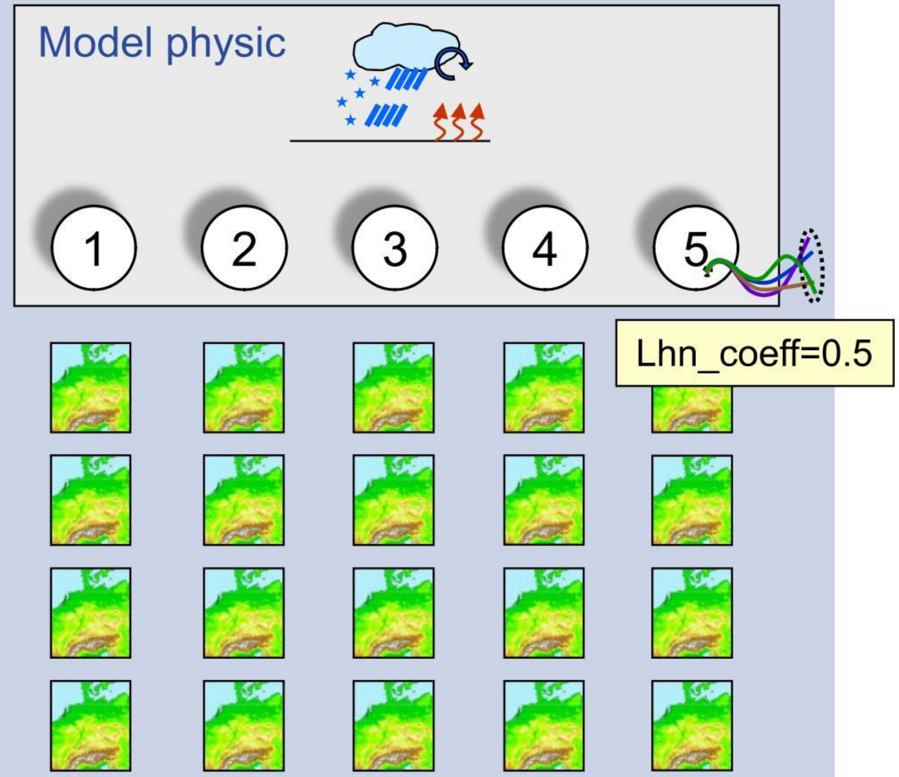
Lagged average ensemble with deep convection permitting model COSMO-DE of DWD

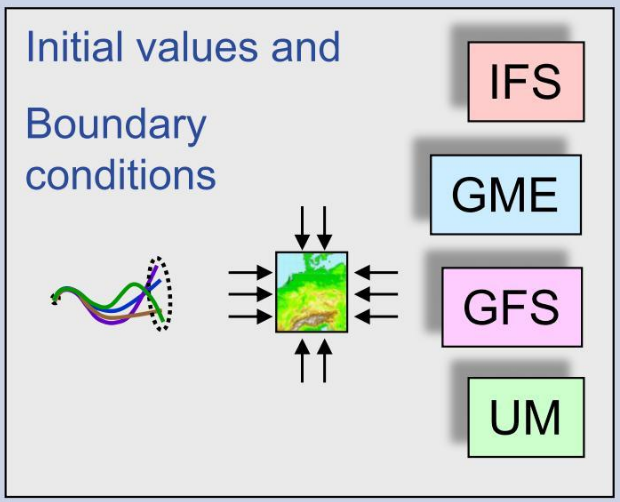
Ensemble members in the experimental COSMO-EPS

Variation in the forecast system:

20 Ensemble Member

(plan to run a global EPS for short range predictions)



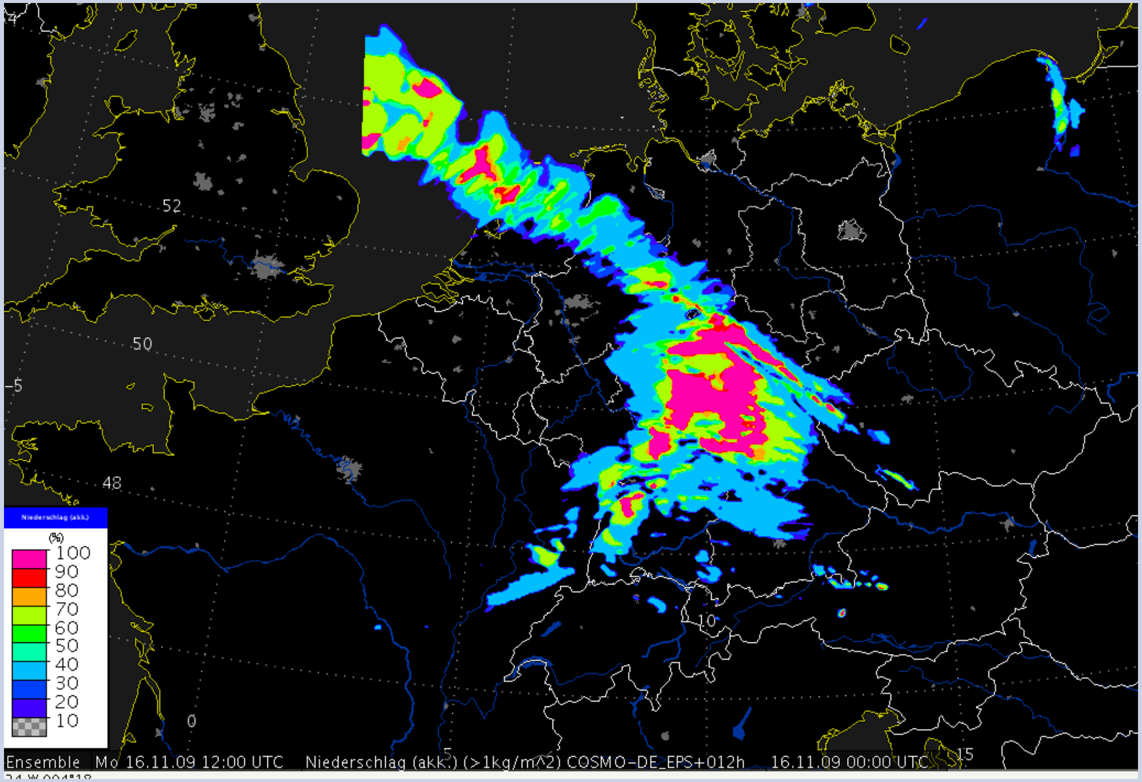


Visualisiation

example:

Probability

precipitation > 1 mm



Actual developments in components of NWP systems

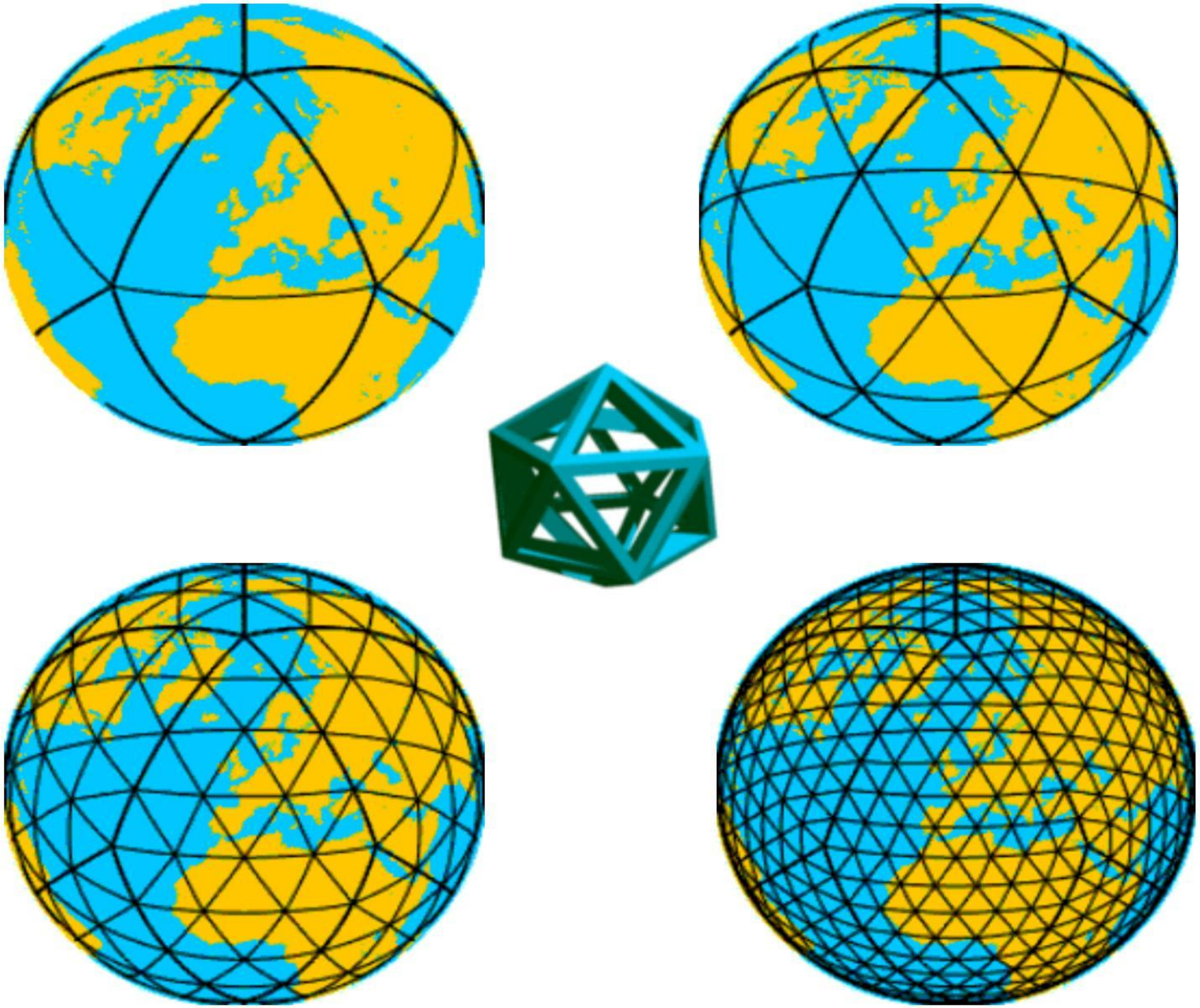
- Increasing resolution
  - Accuracy of the representation of processes in the atmosphere relevant for weather
  - Direct simulation of all relevant processes
- Need for new global nonhydrostatic models
  - compressible, nonhydrostatic model formulations successfully tested in limited area models
- ICON project – collaboration DWD with MPI for Meteorology
- Data assimilation

Requirements for next generation global models

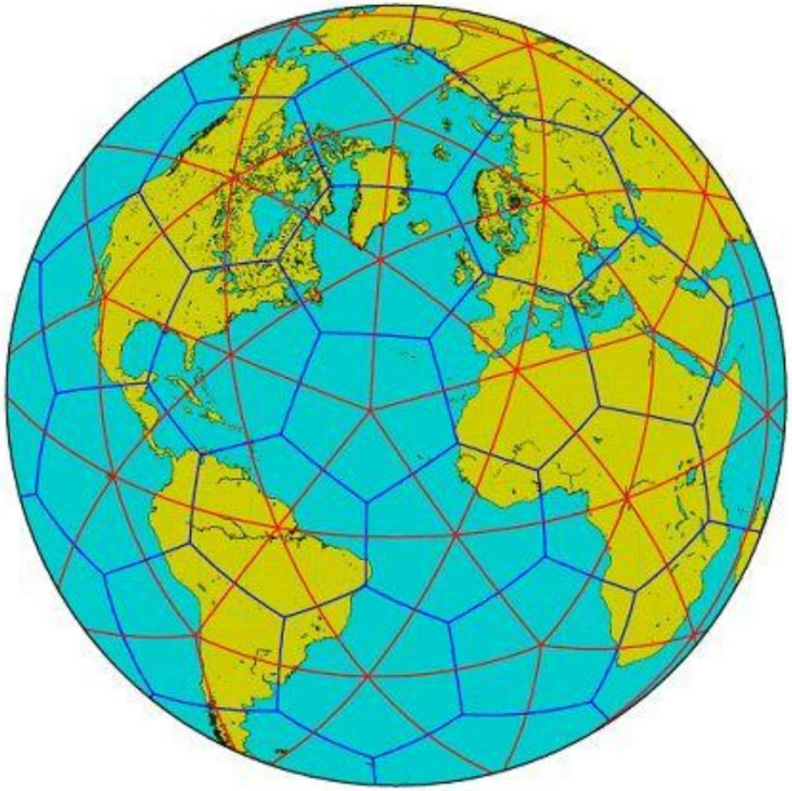
- Applicability on a wide range of scales in space and time in a modular way → „seamless prediction“
- Integration of the fully compressible (non-hydrostatic) equations of motion
- Deep atmosphere option
- (Static) mesh refinement and limited area model (LAM) option
- Scale adaptive physical parameterizations
- Conservation of at least mass and scalar quantities; what else: energy?
- Scalar transport consistent with the discrete mass conservation equation
- Positive definite transport for scalars; monotonicity?
- Scalability and efficiency on massively parallel computer systems with more than 10.000 to 100.000 cores
- Operators of at least 2nd order accuracy

The ICON-Project: Main Goals

- Centralize Know-how in the field of *global modelling* at DWD and the Max-Planck-Institute (MPI-M) in Hamburg.
- Develop a *non-hydrostatic global model with static local zooming option* (ICON: ICOsahedral Non-hydrostatic; http://www.icon.enes.org/).
- At DWD: Replace global model GME and regional model COSMO-EU by ICON with a high-resolution window over Europe. Establish a library of scale-adaptive physical parameterization schemes (to be used in ICON and COSMO-DE).
- At MPI-M: Use ICON as dynamical core of an Earth System Model (COSMOS); replace regional climate model REMO. Develop an ocean model based on ICON grid structures and operators.
- DWD and MPI-M: Contribute to operational seasonal prediction in the framework of the Multi-Model Seasonal Prediction System EURO-SIP at ECMWF).

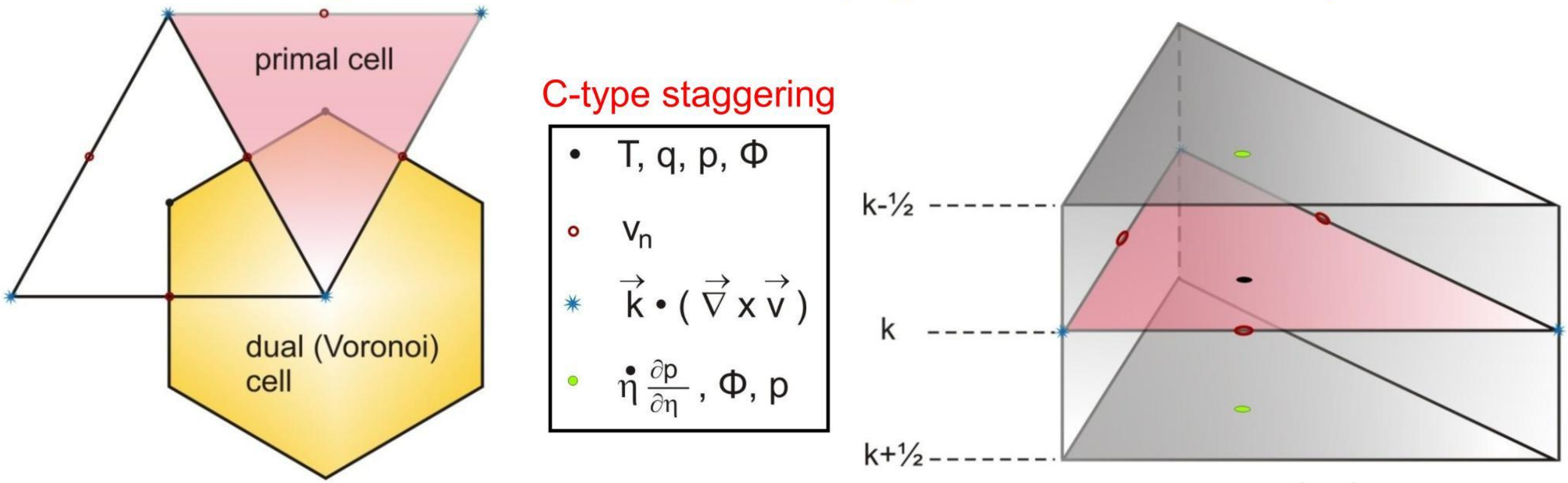


Horizontal grid



Primary (**Delaunay,** triangles) and dual grid (**Voronoi,** hexagons/pentagons)

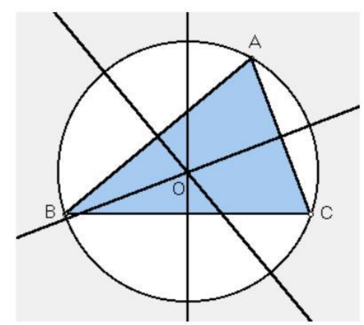
3D-staggering of prognostic and diagnostic variables (hydrostatic core)



horizontal **vertical**

➔ Cell center: center of triangle circumcircle

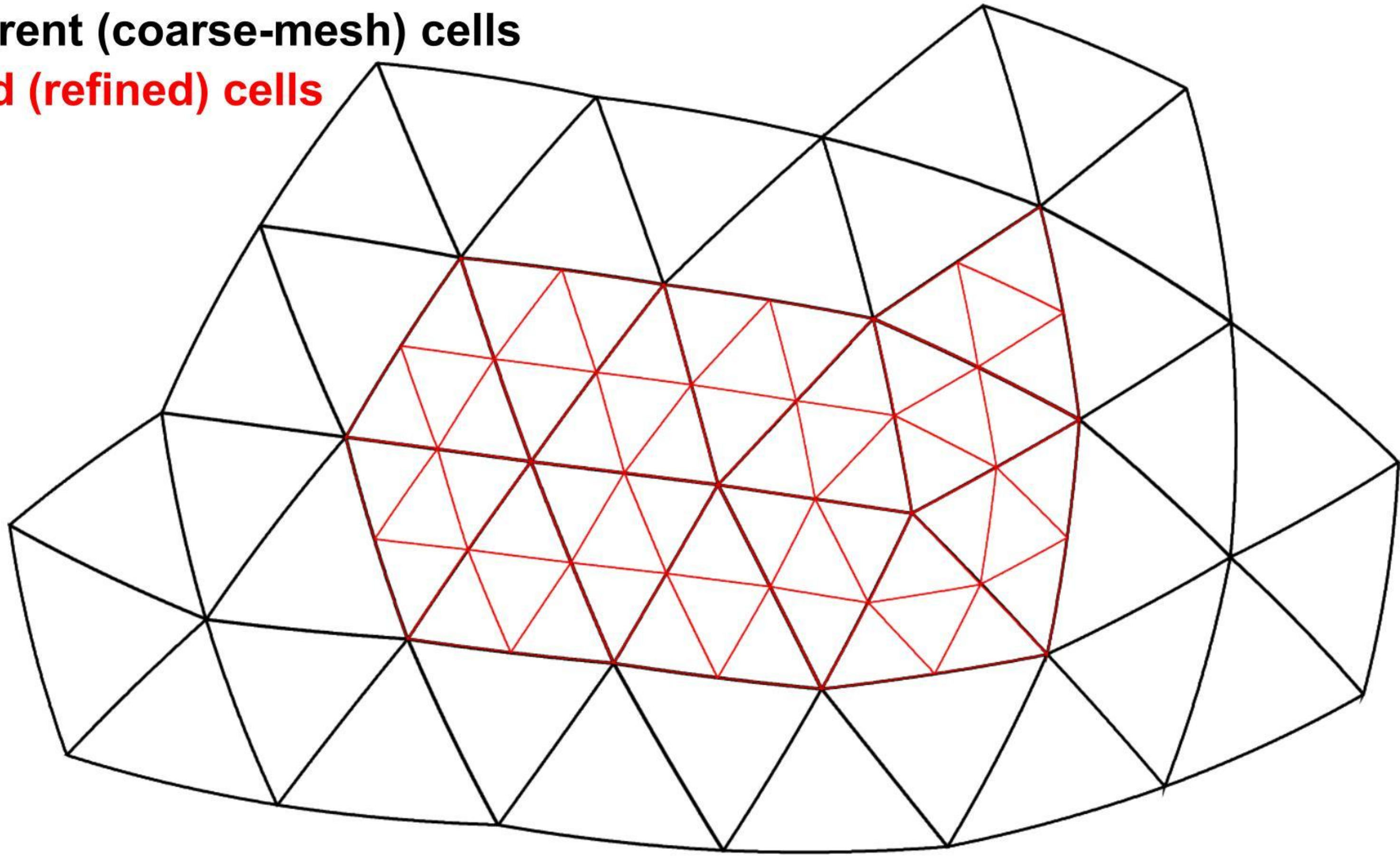
⇒ Arc connecting two mass points is orthogonal to and bisects triangle edge



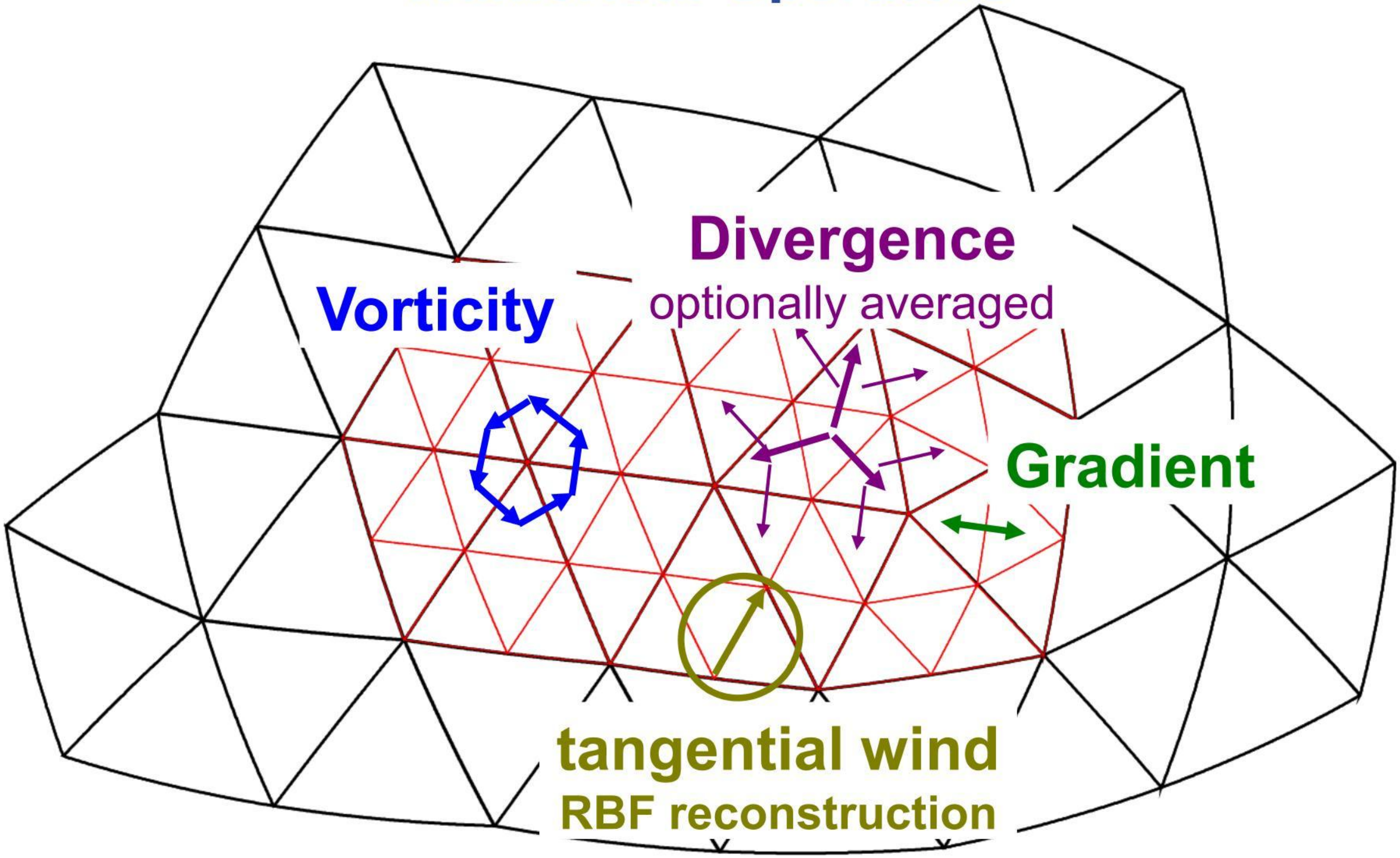
Grid structure in the presence of (static) mesh refinement

Black triangles: parent (coarse-mesh) cells

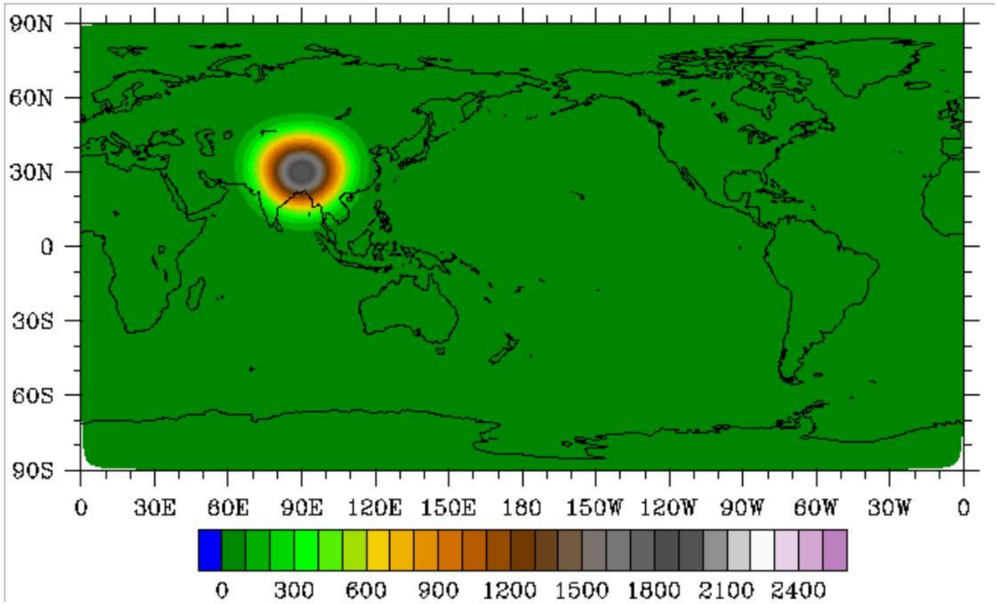
Red triangles: child (refined) cells



Numerical Operators



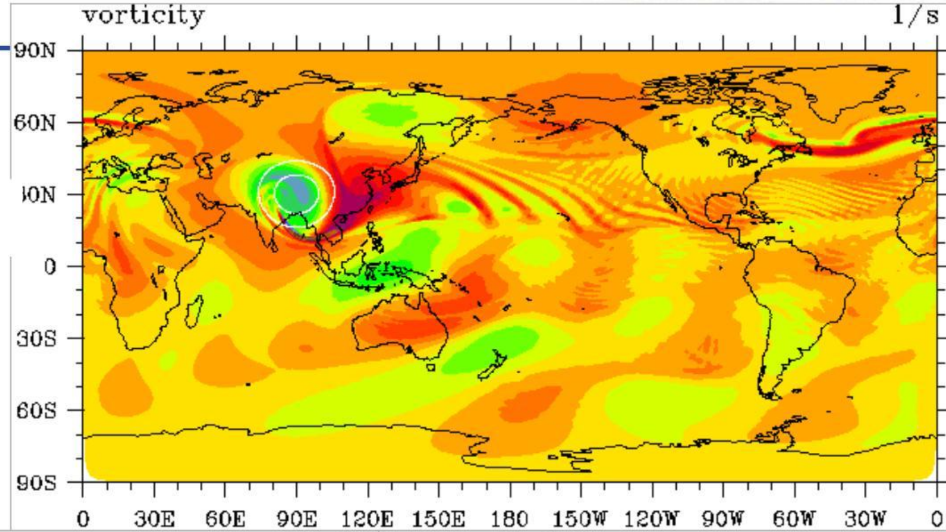
Demonstration example for grid nesting

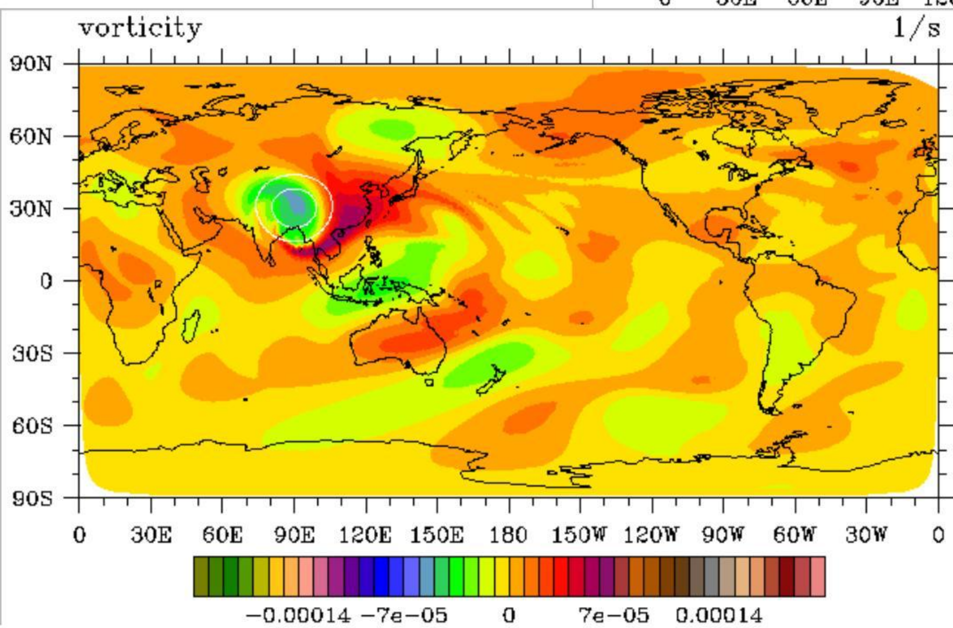


Vorticity at lowest model level on day 20

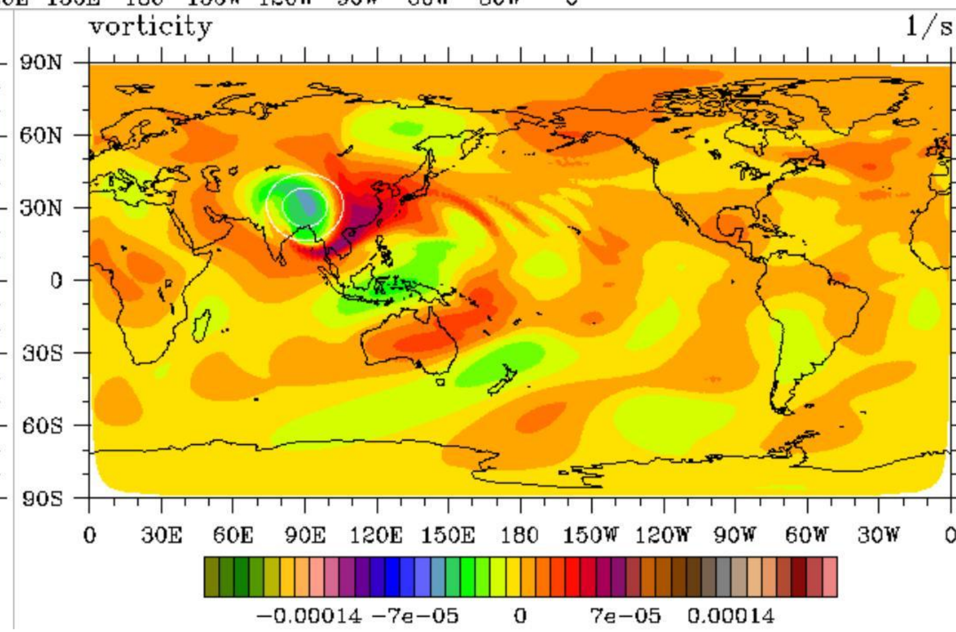
Deutscher Wetterdienst

high-resolution (35 km)





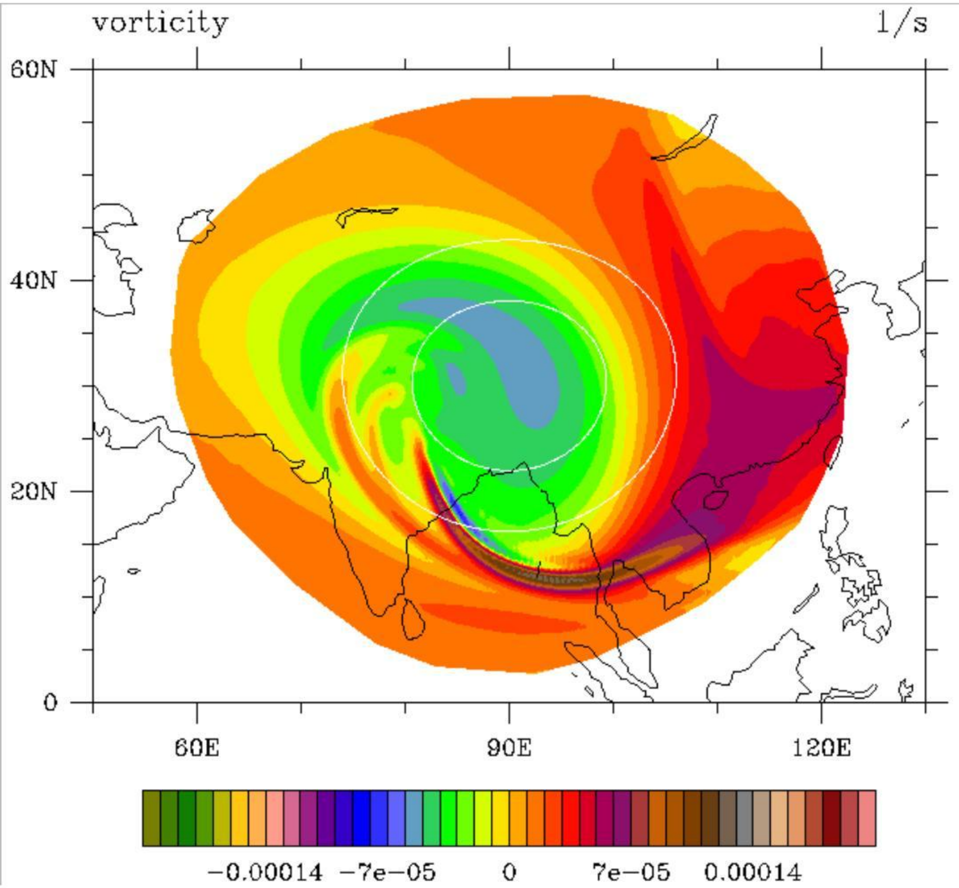
coarse-resolution (140 km)

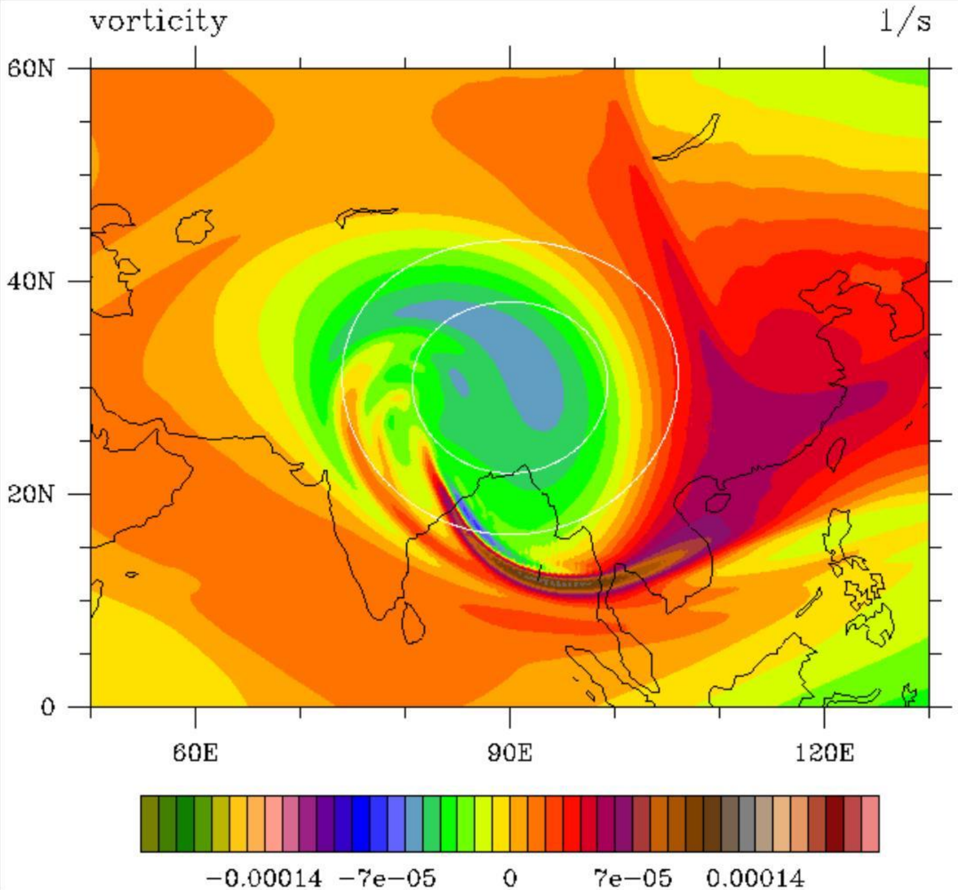


Nested (140 / 35 km)

Vorticity at lowest model level on day 20 (zoom)

Deutscher Wetterdienst
Wetter und Klima aus einer Hand

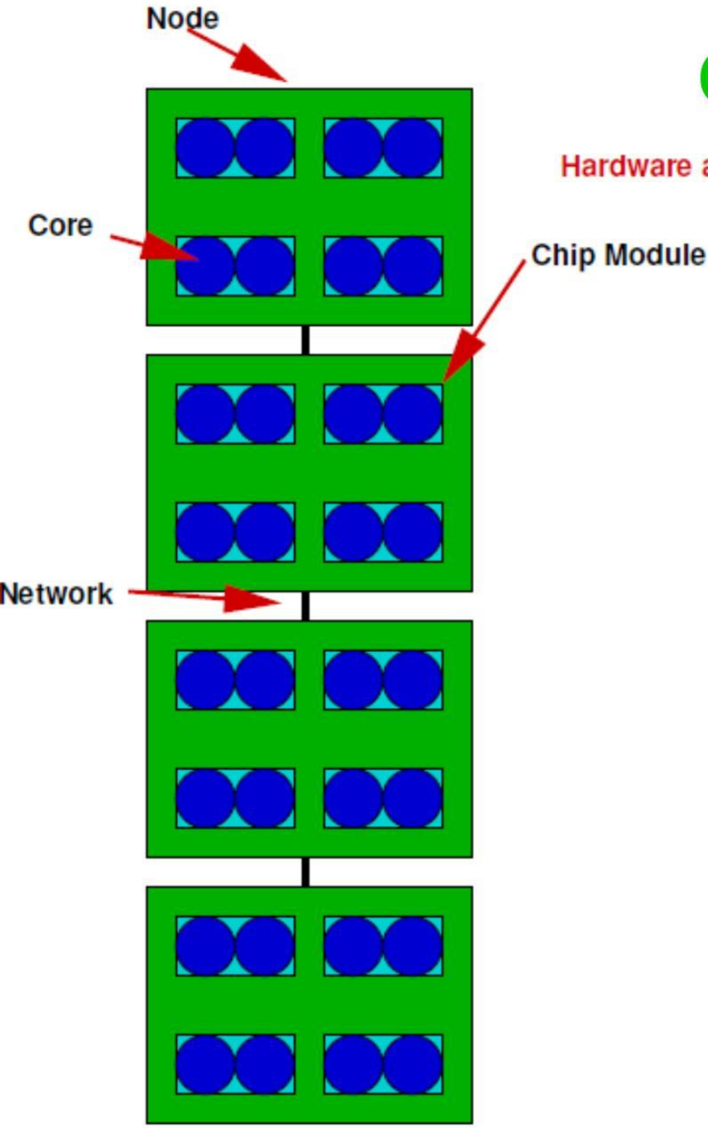


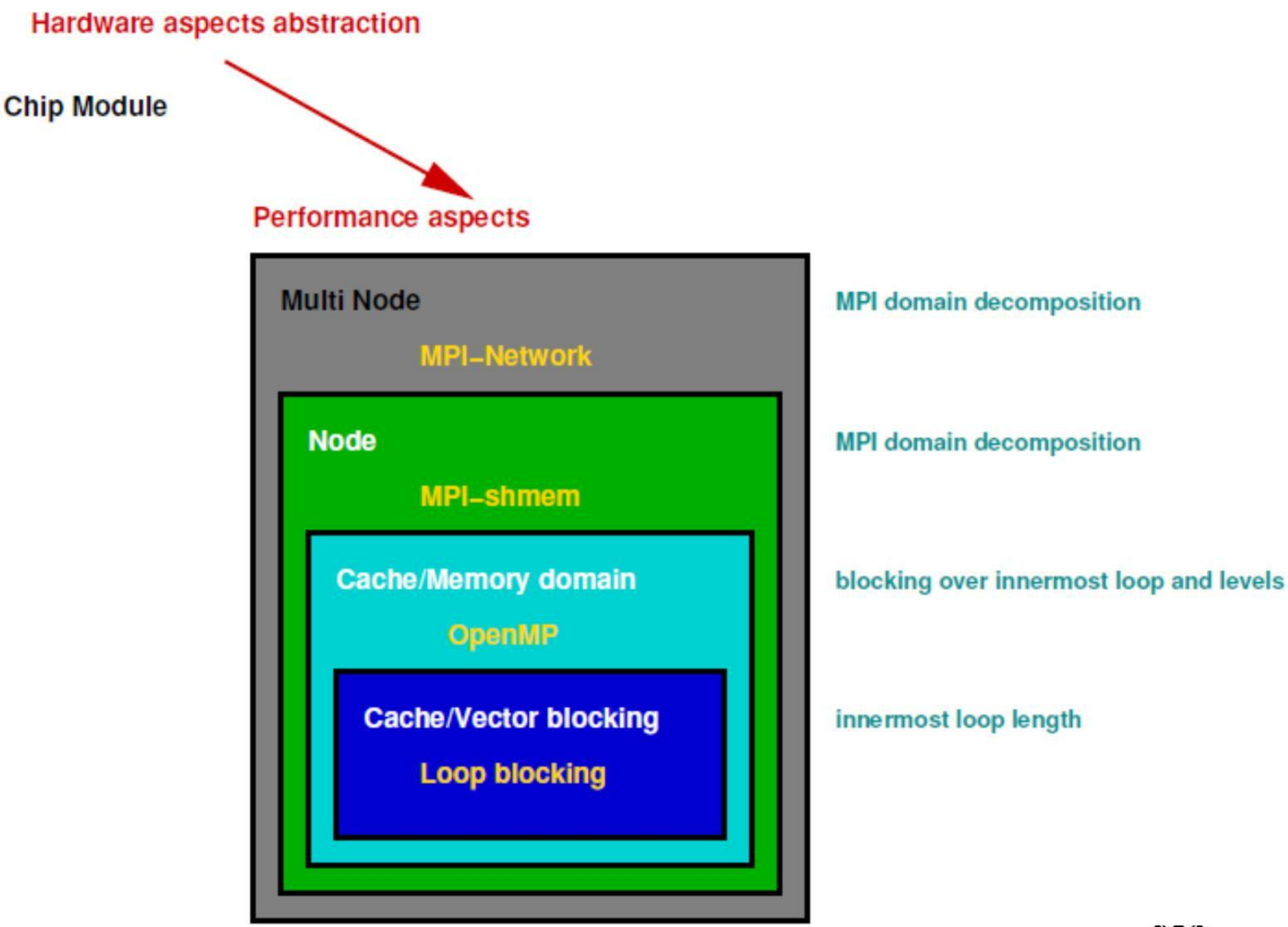


nested (innermost domain; 35 km)

high-resolution (35 km)

Current parallelization levels





Deutscher Wetterdienst
Wetter und Klima aus einer Hand

Challenges

- Grid imprinting ("wave number 5 problem") for icosahedral grids at lower horizontal resolutions.
- Proper balance between conservation, (local) accuracy and efficiency.
- Proper balance between portability (e.g. vector and scalar CPUs), efficiency and code maintenance.
- Scalability of I/O, esp. simulation results, on 10,000 or 100,000 cores.
- Use of GPUs (graphics processing units) to speed up calculations.

Components of a data assimilation system

- Observation operator H:
  - Projection of the model state x on the observation y (simulator of the observation process)
- Observation error covariance **O**:
  - Covariances between the errors of all observations
- Model error covariance **B**:
  - Covariance between the errors of all degrees of freedom of the prdiction system
  - 10^8 degrees of freedom
- Variational problem with 10^8 variables
  - Most probable state of the atmosphere is defined by the minimum of

$$J(x) = (x - x_B)^T \mathbf{B}^{-1} (x - x_B) + (y_O - H(x))^T \mathbf{O}^{-1} (y_O - H(x))$$

Recent Developments in data assimilation

- ➔ Prediction of flow dependent model error covariance matrix **B**
  - ➔ by using the Ensemble of forecasts to approximate a Kalman filter process
- ➔ This requires an ensemble of data assimilations
  - ➔ which provides also the necessary initial values for the ensemble forecast system

Summary

- The development of NWP systems still demands increasing computational ressources
  - Increasing resolution
  - Comprehensive physics
  - Ensemble prediction systems (EPS) on all scales
  - High resolution Ensemble data assimilation


with Acknowledgement to
D. Majewski
S. Theis

Mitglied der Helmholtz-Gemeinschaft

JÜLICH FORSCHUNGSZENTRUM

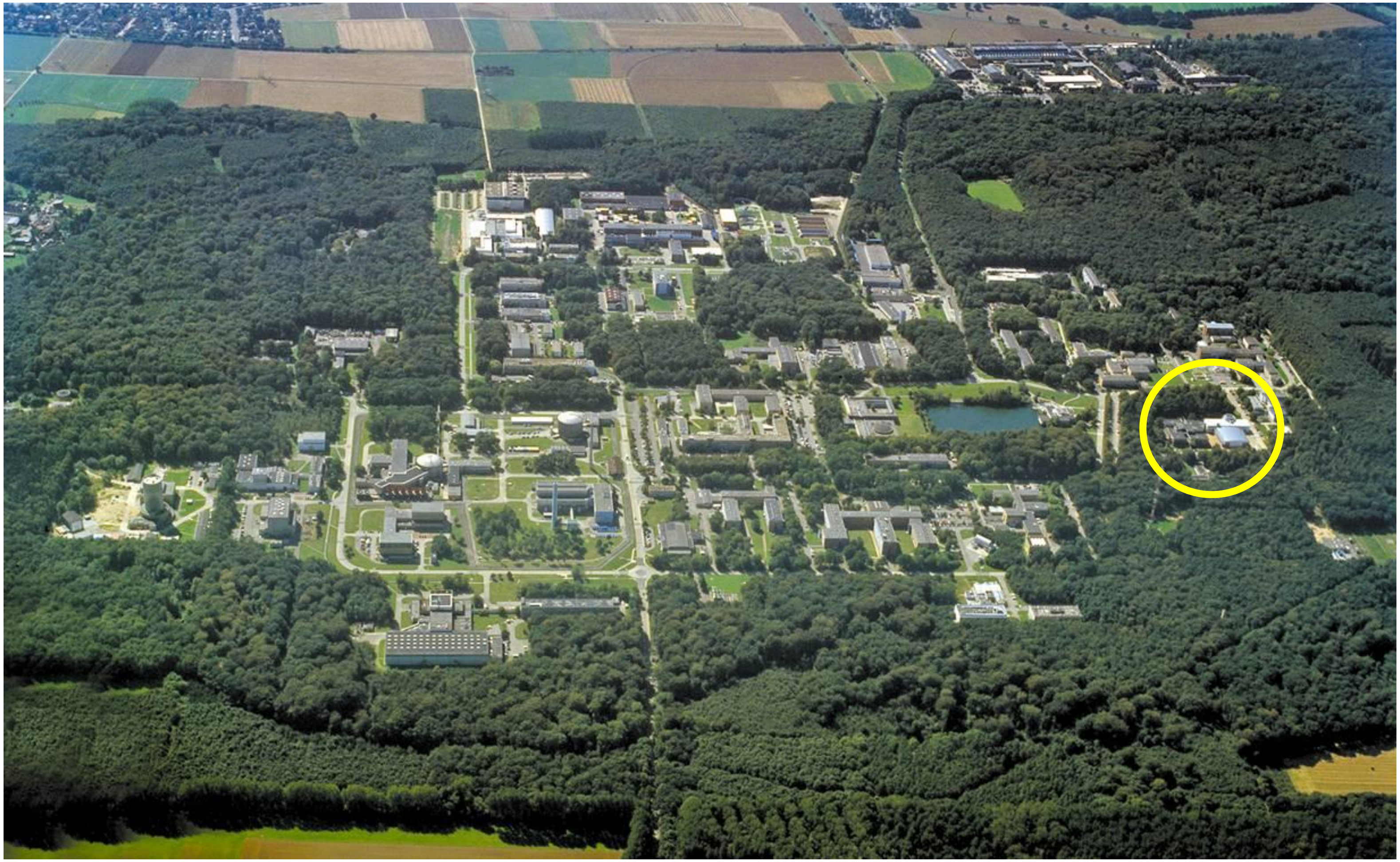
Simulation Laboratories at the Jülich Supercomputing Centre

Paul Gibbon

JSC, Forschungszentrum Jülich

SimLab@KIT Workshop, 29 November 2010

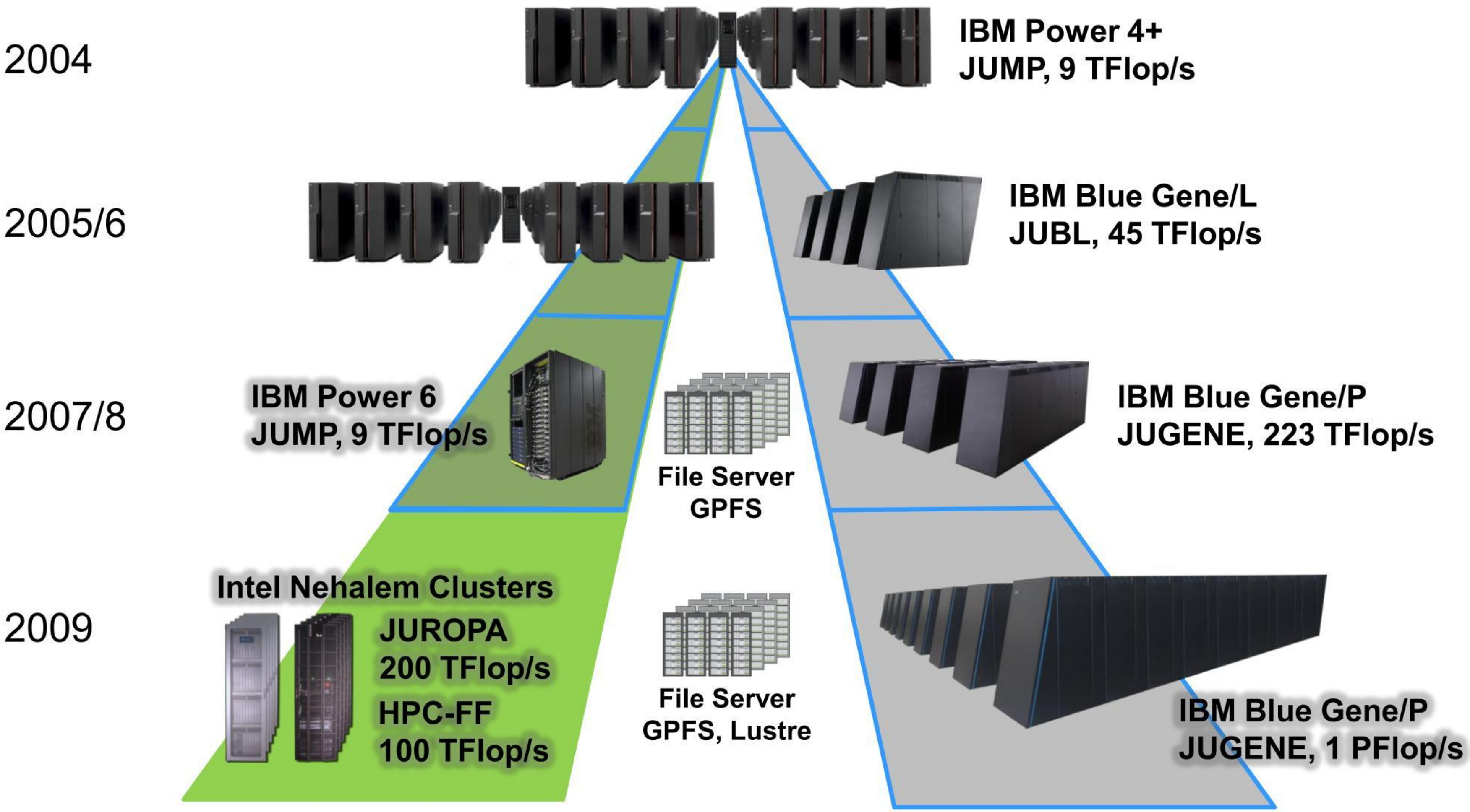
Forschungszentrum Jülich (FZJ)



Main Tasks of the Jülich Supercomputing Centre

- Operation of the supercomputers for local, national and European scientists.
- User support: application tuning; domain-specific support through simulation laboratories
- R&D: architectures, algorithms, performance analysis and tools, GRID computing
- Education and training of users, (bachelor and master courses, PhD programmes)

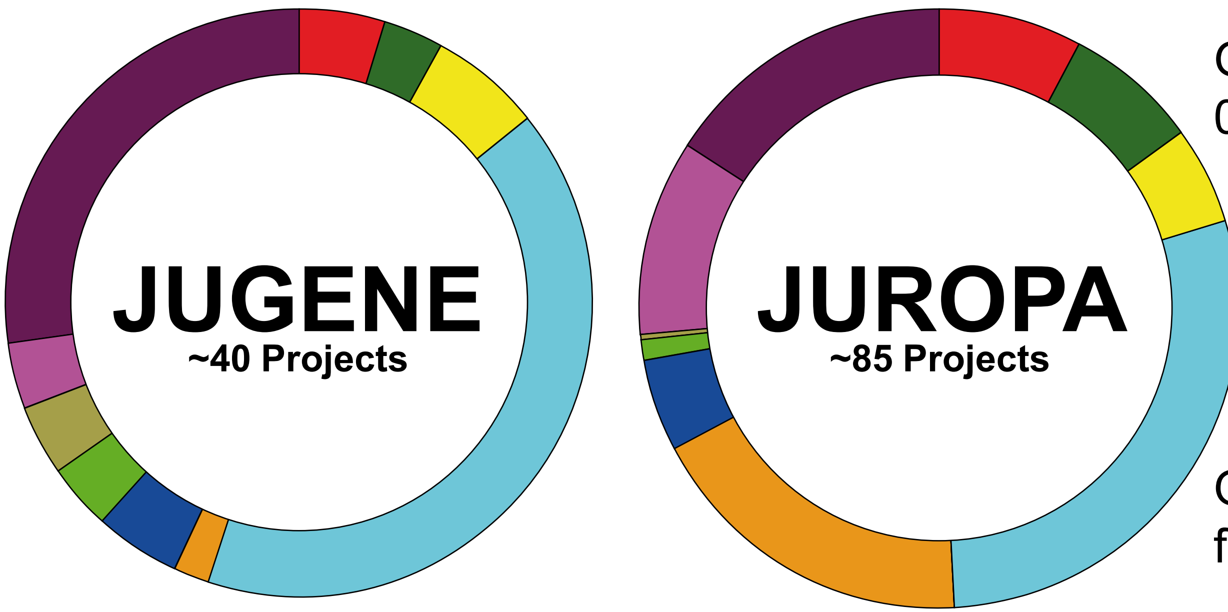
The Jülich Dual Supercomputer Concept



Use by Scientific Discipline

Leadership-Class System

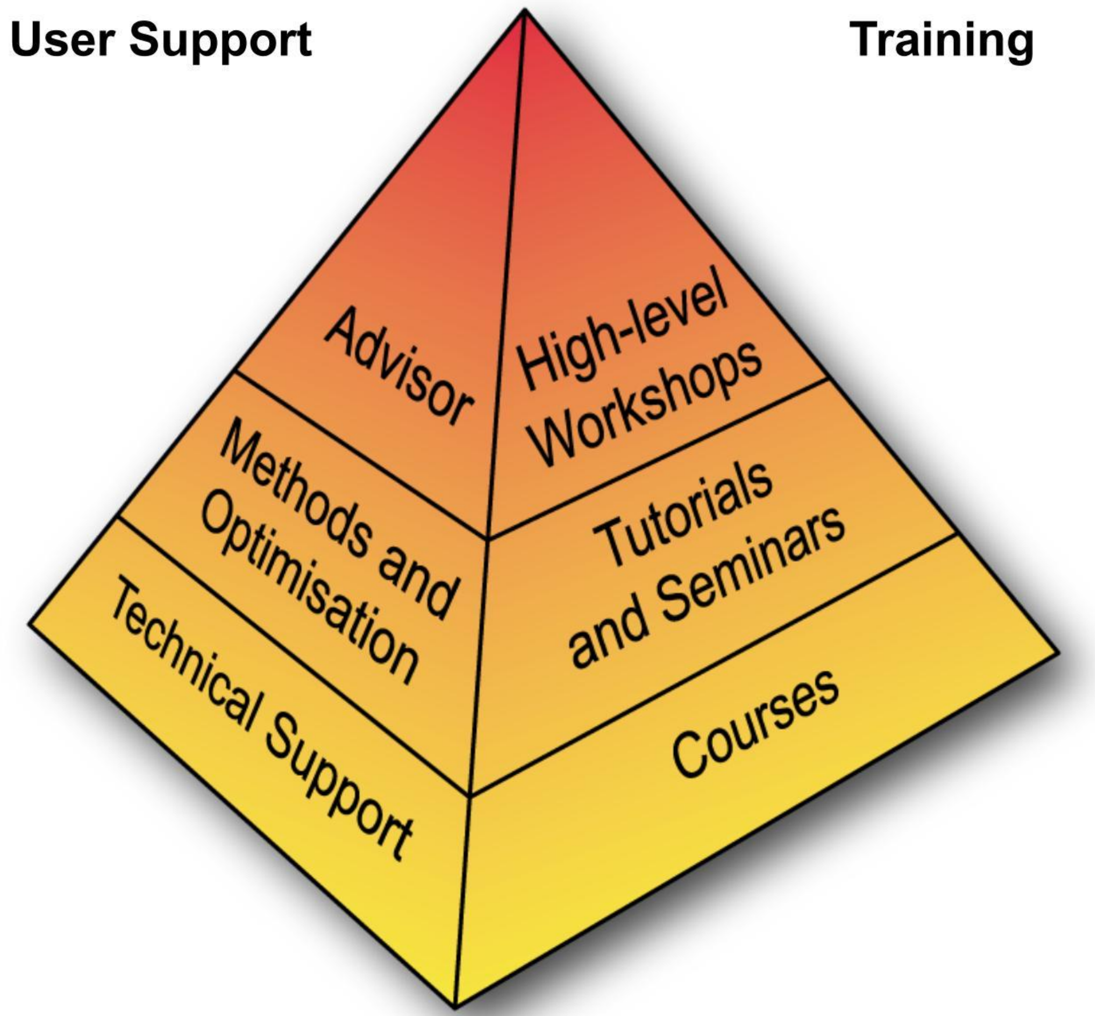
General-Purpose Supercomputer



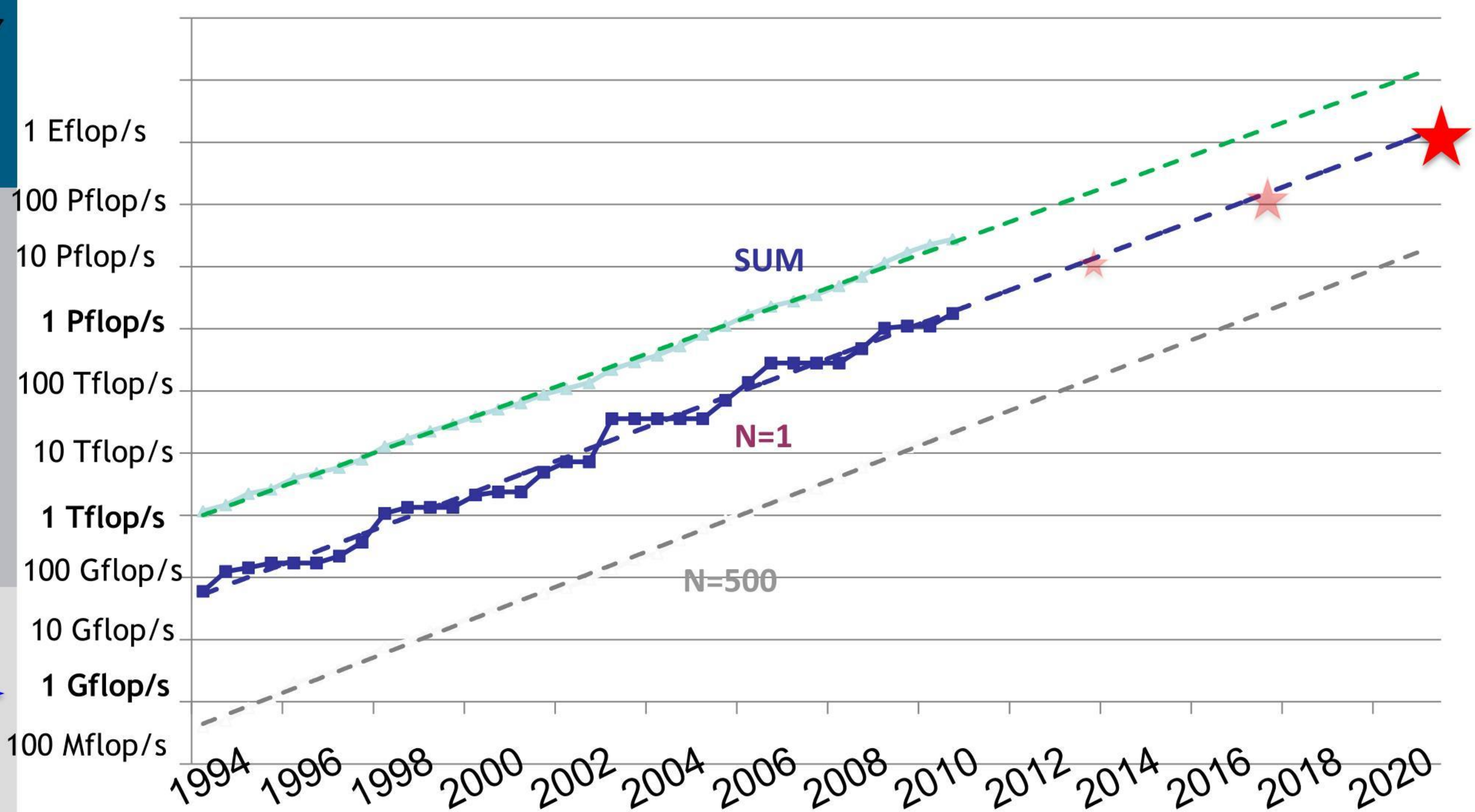
Granting period 07/2009 – 04/2010

Oversubscription factor > 5

Primary Application Support at JSC



Performance Development in Top500

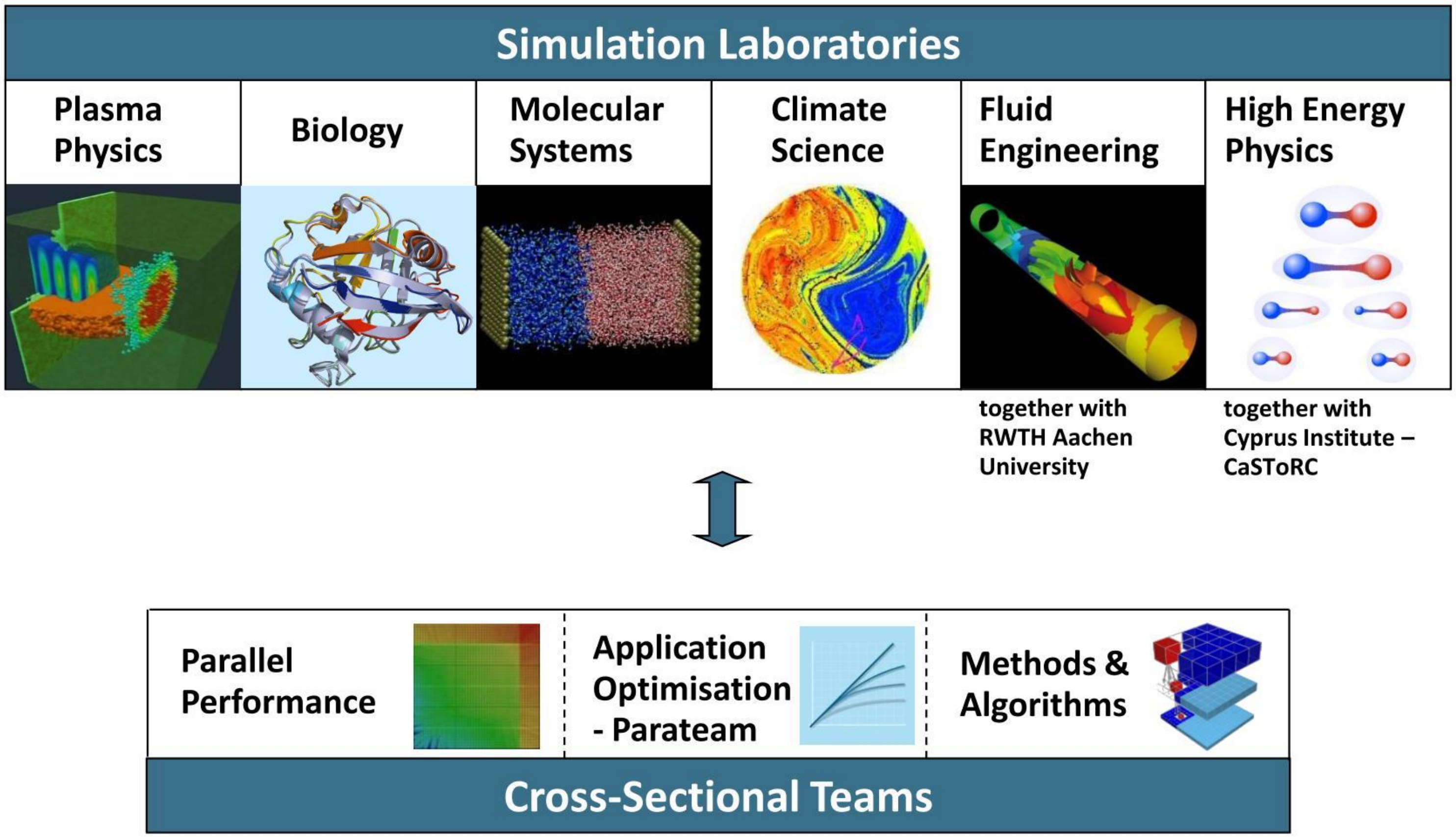


Peta/Exascale Software Challenges

- Are users able to follow the rapid development?
- What can HPC centres do to assist user communities?
- Can we cope with the data tsunami?

> - New approaches to map theories and models onto Exascale systems $\rightarrow$ scalable algorithm design
> - More sophisticated user-support structures: **Simulation Labs**

Domain-specific Research and Support

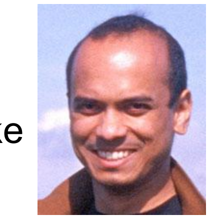


SimLab Teams

- Biology


Olav Zimmermann


Jan Meinke


Sandipan Mohanty

- Molecular Systems


Godehard Sutmann


Thomas Müller


Annika Schiller

- Plasma Physics


Lukas Arnold


Paul Gibbon

- Climate Research

Lars Hoffmann (ICG-1)

Simulation Labs: Structure

Staff

- 1 senior scientist
- 1-2 postdocs
- 1-2 technical staff (informatics/technomath)
- Jointly supervised PhD & MSc students

Research

- Common/generic simulation methods
- Scalable algorithms
- Joint projects with SimLab partner groups

Support

- Porting/tuning/benchmarking
- Algorithm scaling
- Training

Timeline

2007	JSC `White Paper´
2009	1st three SimLabs created at JSC
Jan 2010	Start of Helmholtz Programme POF II
June 2010	1st SimLab Porting Workshop
Sept 2010	Call for SimLab support – **pilot project** 2010/11
2011	SimLabs Climate, QCD, Fluid Engineering (with RWTH)

SimLab Support Activities

1) Informal code-enabling/diagnostic visits (1-5 days)

2) Long-term partnerships & coops
   - Research groups, institutes, consortia (eg: CECAM)
   - 3rd party projects

3) High-level application support (pilot from Autumn 2010)
   - Proposals in form of self-contained WPs (1-2 PM)
   - Source-code tuning, redesign, refactoring

4) Workshops

JÜLICH
FORSCHUNGSZENTRUM

Simulation Lab Biology

Research

- Protein folding & interaction (docking)
- Structure prediction
- Systems biology

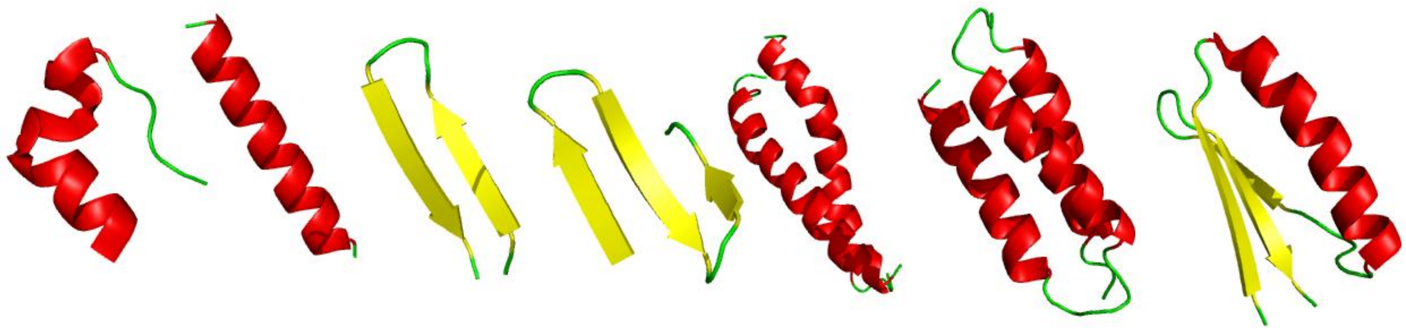
Support

- Libraries, databases, benchmarking
- Monte Carlo, FFT docking, machine learning

Codes

- ***PROFASI***, *SMMP, SVMGrid,* ***LOCUSTRA***

Monte Carlo simulation with ProFASi



ProFASi

- Developer: Sandipan Mohanty
- Language: C++
- DOF: dihedrals and rigid body
- Lund Force Field (Anders Irbäck)
- Strategy to be fast:
  calculate as few things as possible; use cutoffs
- Scales up to 16k cores using replica exchange and multiplexing.
- Energy function not yet parallelized

Simulation Lab Molecular Systems

Research

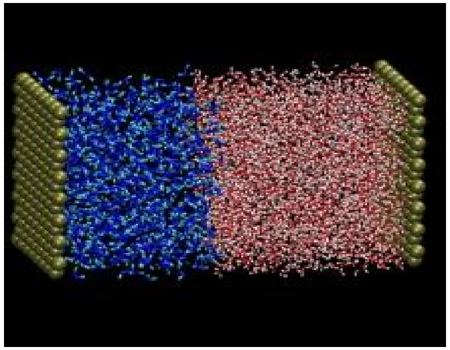
- Macroscopic properties from microscopic information
- Model larger systems / longer timescales
- Integrated multi-scale approaches

Support

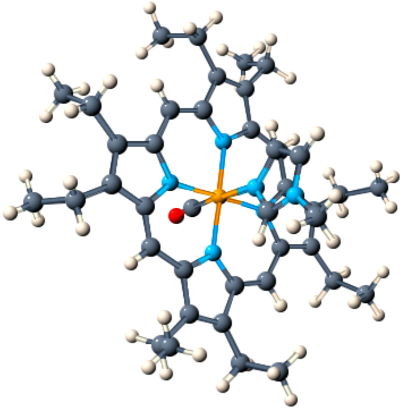
- Methods: electronic structure, force-field, MD
- Quantum-chemical modelling & tools
- Scalable algorithms for supercomputers

Codes

- MP2C, P3MG, TURBOMOLE, COLUMBUS



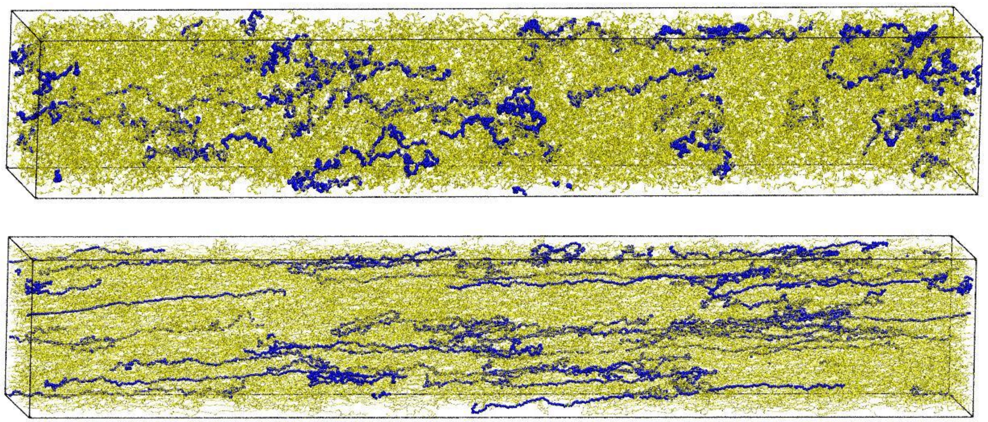
Water-oil interface

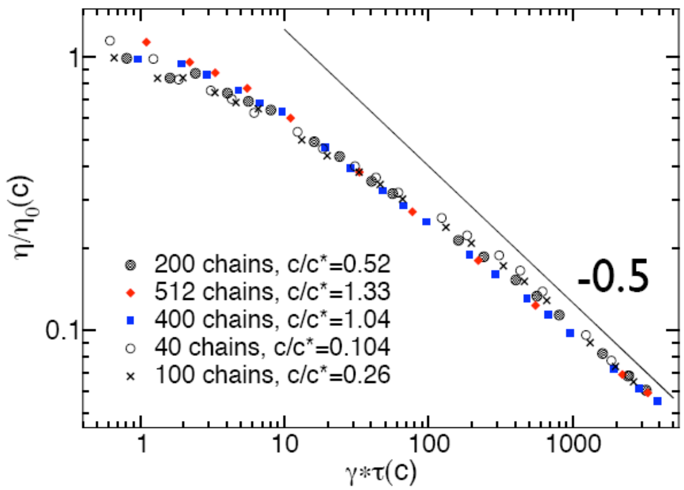


Oxygen transport

Semi-diluted polymer solutions under shear

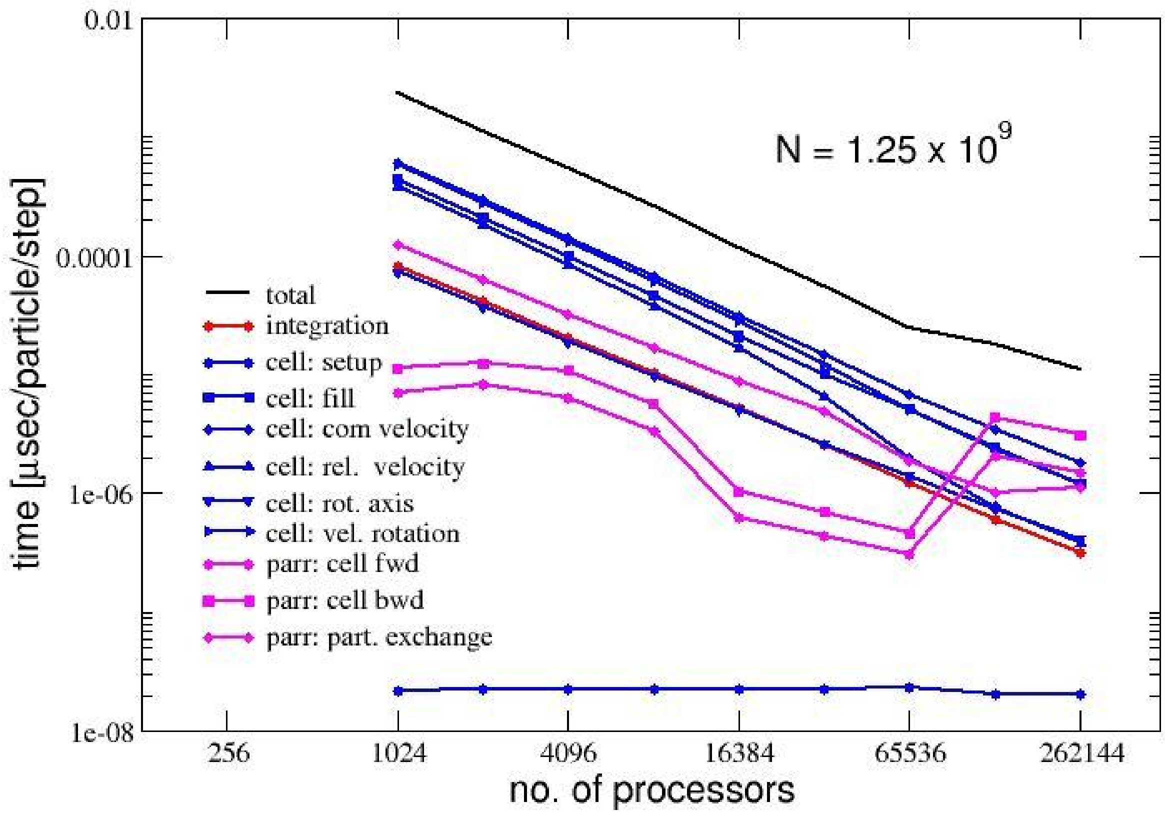
- Shear thinning for large shear rates





- Code: MP2C
- Cooperation with IFF-1 (Gompper)

MP2C strong scaling on BlueGene/P



- Large partitions produce unexpected surprises !
- Possible reason: bad mapping of processes to physical domain

Simulation Lab Plasma Physics

Research

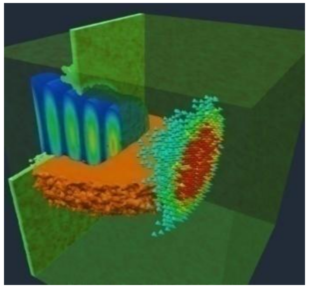
- Kinetic methods: Particle-in-Cell, Vlasov, MD
- Fluid + MHD models
- Transport: Monte Carlo

Support

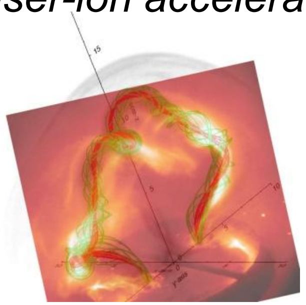
- Plasma model porting & scaling
- Code benchmarking – eg: 3D PIC

Codes

- *PSC*, *ILLUMINATION*, *PEPC*, *racoon*
- *EIRENE, ERO*



Laser-ion acceleration

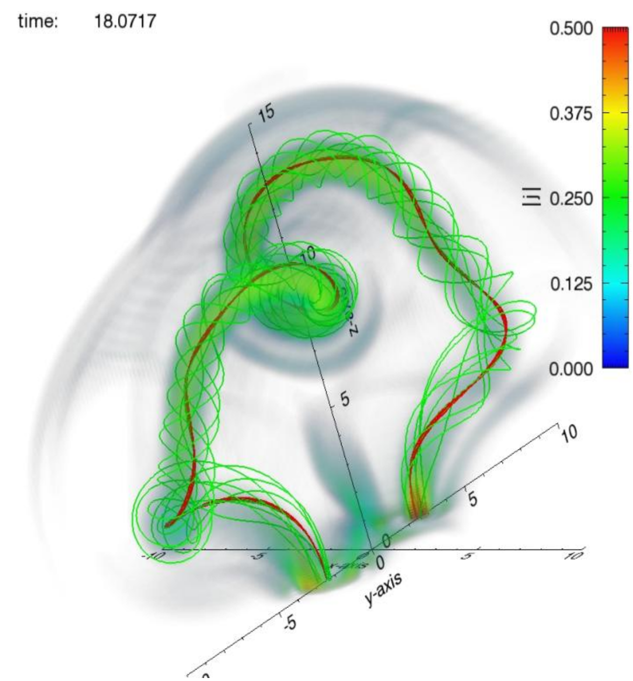


Solar flare modelling

SimLab Actions (from Support Call 2010)

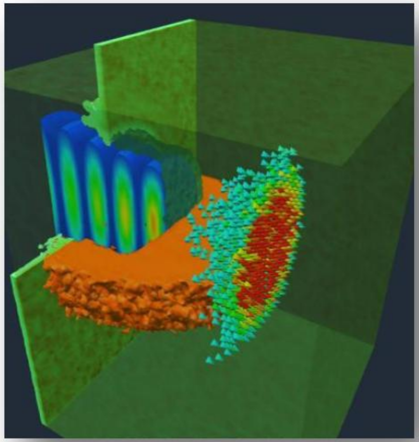
- Partners
  - University Bochum (racoon, jugene)
  - University Warwick (EPOCH, jugene)
  - Research Center Dresden-Rossendorf (PICLS, jugene)
  - University Frankfurt (ralef-2d, juropa)
  - Research Center Jülich (B2/B2.5, juropa)

- Main Topics
  - basic parallelization
  - parallel I/O
  - advanced scaling (>10k cores)
    - *application profiling*
    - *load balancing*
    - *communication structure redesign*



Code Development (Main Projects)

- PEPC
  - N-body (Barnes-Hut) tree code
  - application fields: plasma, gravitational, soft matter physics
  - scales up to 32k cores on jugene



- PSC
  - particle-in-cell code
  - application field: laser-plasma interaction
  - scales up to 4k mpi tasks

In the TEXT project, both codes are further scaled (thread level) using SMPSs

Training Events 2010

- 1^{st} SimLab Porting Workshop
  - www.fz-juelich.de/jsc/simlab-porting-workshop/
  - Hands-on porting & scaling JUGENE + JUROPA
  - Feedback sessions with community members

- Heraeus Summer School
  - Fast methods for long-range interactions in complex systems
  - http://www.fz-juelich.de/conference/wehss

Coming up in 2011

- Climate Science starting January 2011
- 2 new labs in Fluid Engineering and Lattice QCD
- CECAM workshops and guest student programme
- Engagement with Exascale software initiatives (EESI, IESP)

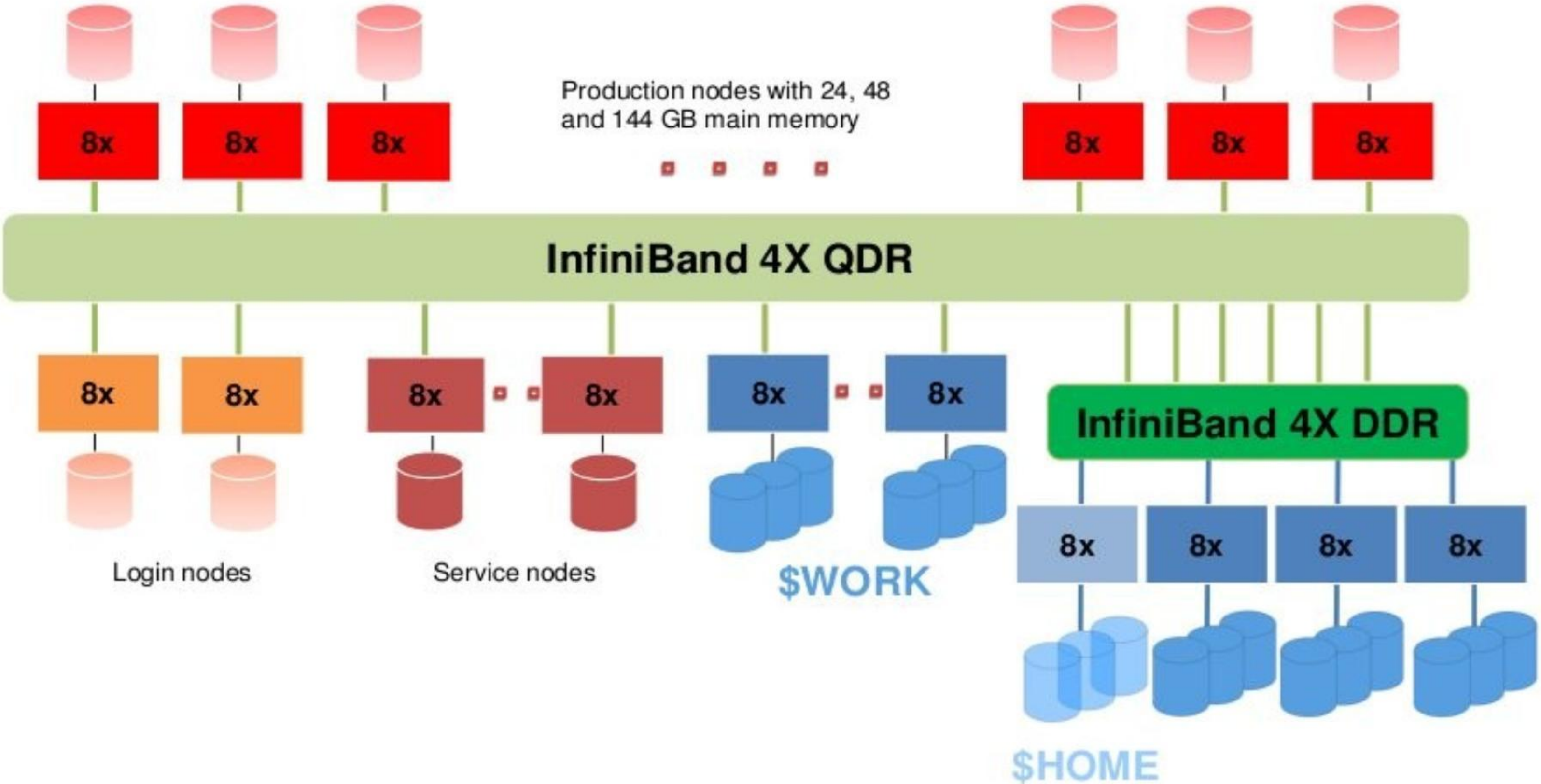
Karlsruher Institut für Technologie

Hardware on HC3 – Nehalem's Impact on Performance

Hartmut Häfner

STEINBUCH CENTRE FOR COMPUTING - SCC

Configuration of HC3 (HP XC3000)

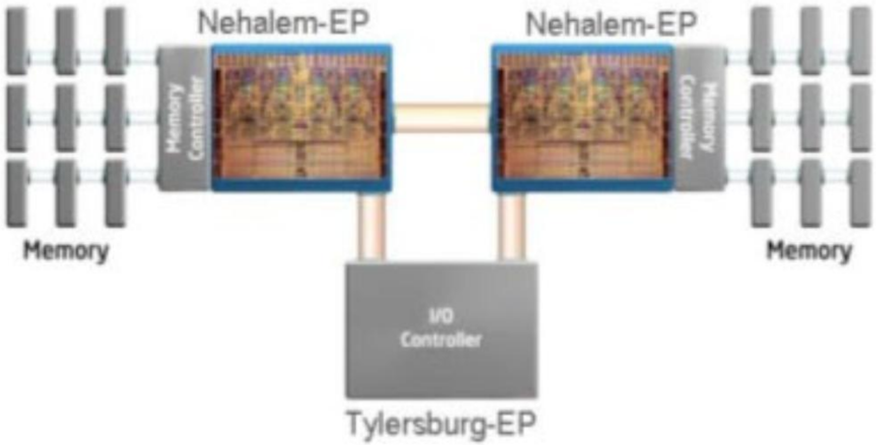


Detailed Configuration of HC3

Karlsruher Institut für Technologie

- 332 nodes in production pool
  - 288 x (2 Quad-Core Intel Xeon 5540, 2.53 Ghz, 24 GB main memory)
  - 32 x (2 Quad-Core Intel Xeon 5540, 2.53 Ghz, 48 GB main memory)
  - 12 x (2 Quad-Core Intel Xeon 5540, 2.53 Ghz, 144 GB main memory)
- 2 nodes in development pool
  - like nodes in production pool with 24 GB main memory
- 2 login nodes
  - like nodes in production pool with 48 GB main memory
- InfiniBand 4X QDR Switch
  - Bandwidth between two different nodes > 3100 MB/s
  - Latency (for messages between nodes) ~ 2 µs
- Parallel Filesystem (Lustre)

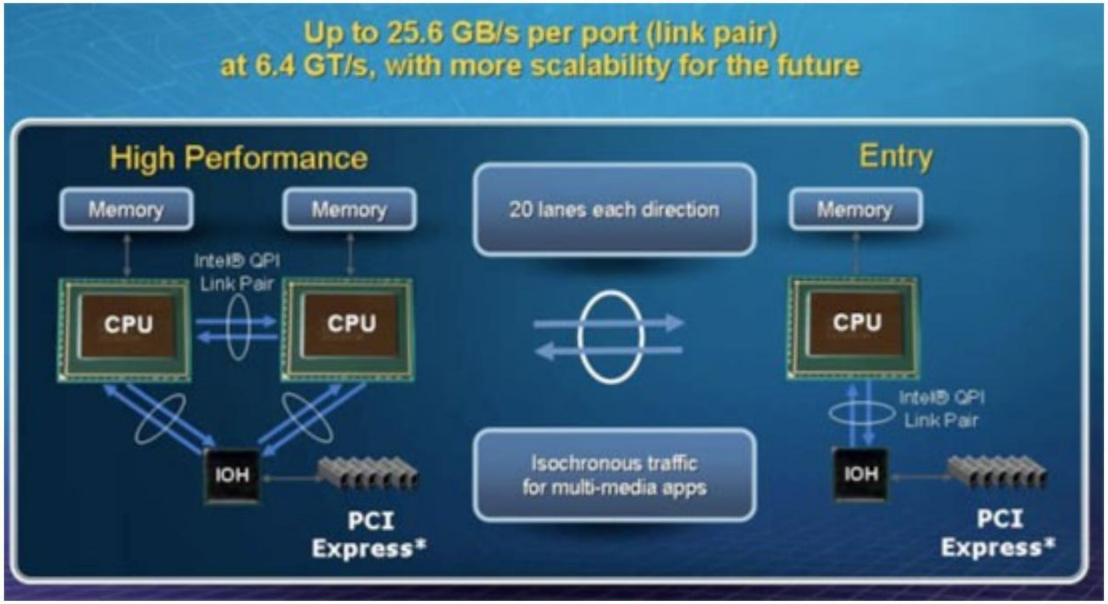
Architecture of Nehalem



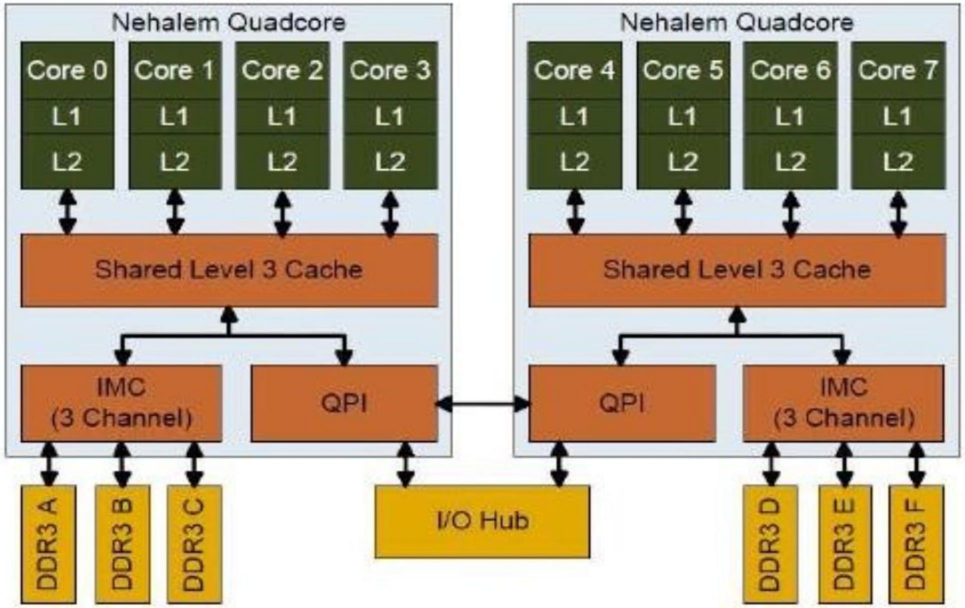
QPI – Quick Path Interconnect

QPI Link is bi-directional with
6.4 GT/s (16 bit parallel) -->
12.8 GB/s per Link in each direction

Local memory bandwidth:
25.6 GB/s (DDR3/1066 MHz)
uni-directional



Architecture of Nehalem (2)



L1-Cache	4 x 32 KB	2 words/cycle bi-directional
L2-Cache	4 x 256 KB	4 words/cycle uni-directional
L3-Cache	8 MB (shared)	1-4 words/cycle uni-directional
Memory	Up to 32 GB	25.6/(20.24...80.96) = 0.31 -1.26 words/cycle

Nehalem's architectural Efficieny

Peak performance of one core: 4*2.53 = 10.12 Gflops

Architectural efficiency for full triad: **c** = **c** + **a*b** (**a**, **b**, **c** are vectors)

Full Triad	Expected Theoretical Peak Performance	Architectural Efficiency
Data from L1-cache	10.12 / 2 = 5.06 Gflops	0.5
Data from L2-cache	10.12 / 2 = 5.06 Gflops	0.5
Data from L3-cache	10.12 / 2 … 8 = 1.26 – 5.06 Gflops	0.125 – 0.5
Data from memory	10.12 * (0.31 … 1.26)/4 = 0.79 - 3.18 Gflops	0.08 – 0.315

Architectural efficiency for dot product: s = s + **a*b** (**a**, **b** are vectors)

Dot Product	Expected Theoretical Peak Performance	Architectural Efficiency
Data from L1-cache	10.12 / 1 = 10.12 Gflops	1.0
Data from L2-cache	10.12 / 1 = 10.12 Gflops	1.0
Data from L3-cache	10.12 / 1 … 4 = 2.53 – 10.12 Gflops	0.25 – 1.0
Data from memory	10.12 * (0.31 … 1.26)/2 = 1.62 – 6.37 Gflops	0.16 – 0.63

Nehalem's measured architectural Efficiency

Measured architectural efficiency for full triad: **c** = **c** + **a*b** (**a**, **b**, **c** are vectors)

Full Triad	Measured Performance	Architectural Efficiency
Data from L1-cache	1.45 Gflops	0.14
Data from L2-cache	1.3 Gflops	0.13
Data from L3-cache	0.2 – 0.95 Gflops	0.02 – 0.09
Data from memory	0.2 - 0.5 Gflops	0.02 – 0.05

Measured architectural efficiency for dot product: s = s + **a*b** (**a**, **b** are vectors)
Measurement of library-routine (BLAS) in square brackets

Dot Product	Measured Performance	Architectural Efficiency
Data from L1-cache	2.2 [3.4] Gflops	0.22 [0.34]
Data from L2-cache	1.95 [2.9] Gflops	0.19 [0.29]
Data from L3-cache	0.5 - 1.6 [1.0 - 2.2] Gflops	0.05 – 0.16 [0.1 – 0.22]
Data from memory	0.5 - 1.15 [0.5 - 1.3] Gflops	0.05 – 0.11 [0.05 – 0.13]

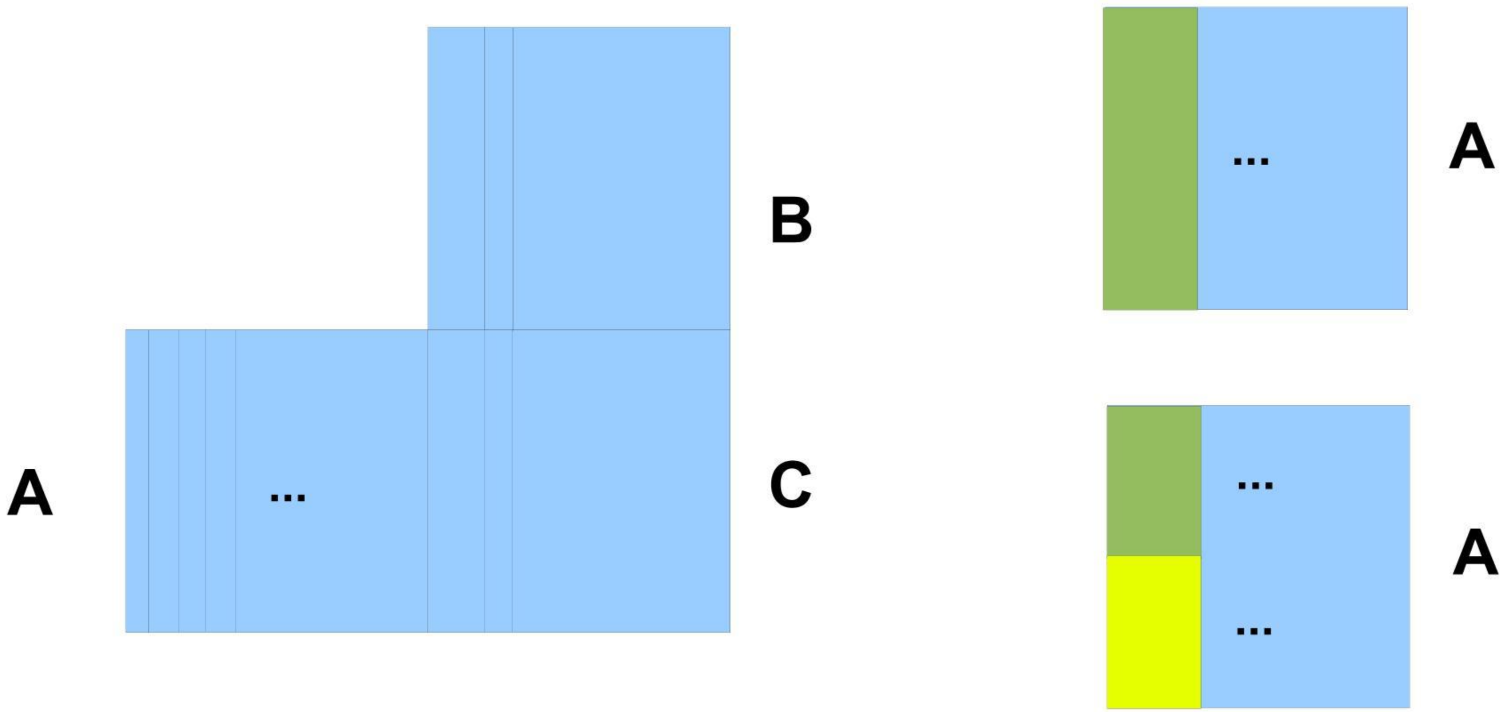
Conclusions

- Implementation of cache reuse is necessary for a good performance
- Cache reuse optimized for L2-cache (instead for L3-cache) should lead to a better performance
- Usage of library-routines – as far as possible – further increases the performance of serial applications

Cache Reuse (optimal)

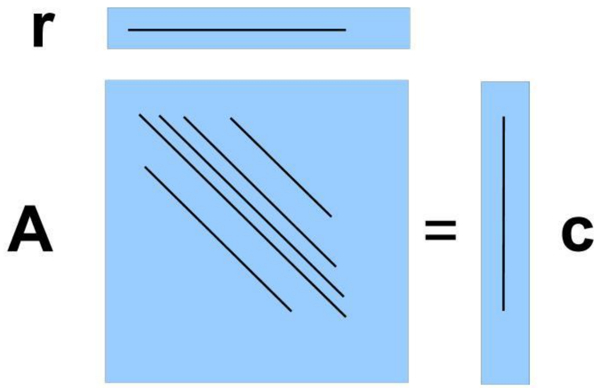
- Matrix-Multiplication **C = A * B**

  c(i,j) = c(i,j) + a(i,k) * b(k,j)

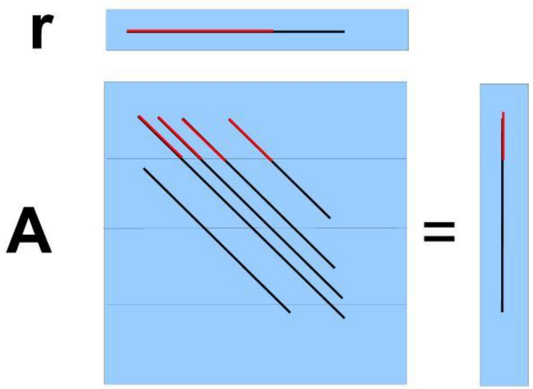


Cache Reuse (partly)

- Matrix-Vector-Multiplication **c=A*r**



Solution for very large matrices:
Striping in blocks of rows (columns)



Memory Requests with different Strides

- Definition of Stride: Stride is the distance in memory (adressing scheme) between successive array elements

Performance losses in percent for vectoroperation (**c**=**a** + **b**) in comparison to stride 1

Stride	2	3	4	5	6	7	8	9	16	17	18	19	20	21	24	25	32	48	49	64	65	128	256	512	2048
Perf. loss vl=100	0	0	0	0	0	0	0	0	0	0	0	0	10	0	40	12	0	0	0	56	0	72	74	78	96
Perf. loss vl=10.000	28	32	53	54	68	72	75	75	75	75	75	75	75	75	75	75	75	72	73	75	75	89	93	97	97

Memory Request with differnet Strides (2)

- Avoid strides ≠ 1 because of reduction of available cache size and heavy performance reduction if data come from memory

- Example for large stride:
  Matrix-Multiplication in dot product form (e.g. 1024x1024 matrices)
  c(i,j) = c(i,j) + a(i,**k**) * b(**k**,j)

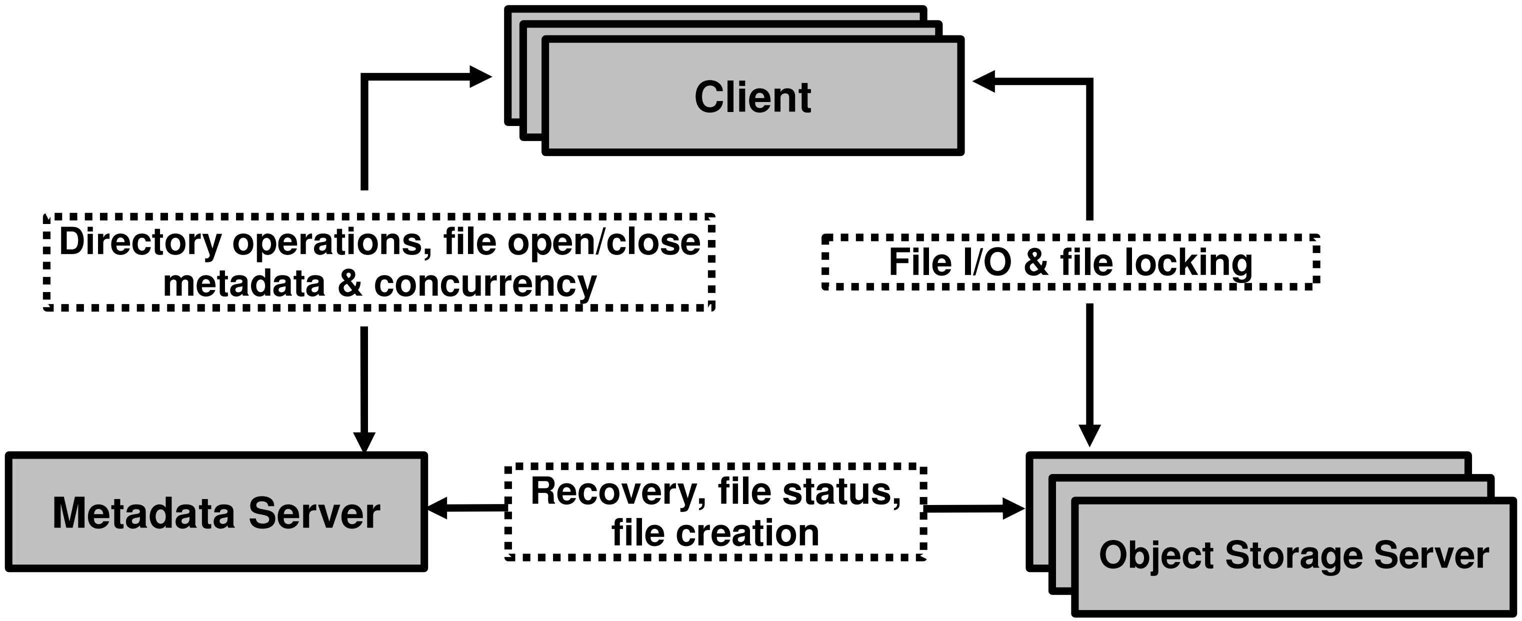
Karlsruher Institut für Technologie

Using file systems at HC3

Roland Laifer

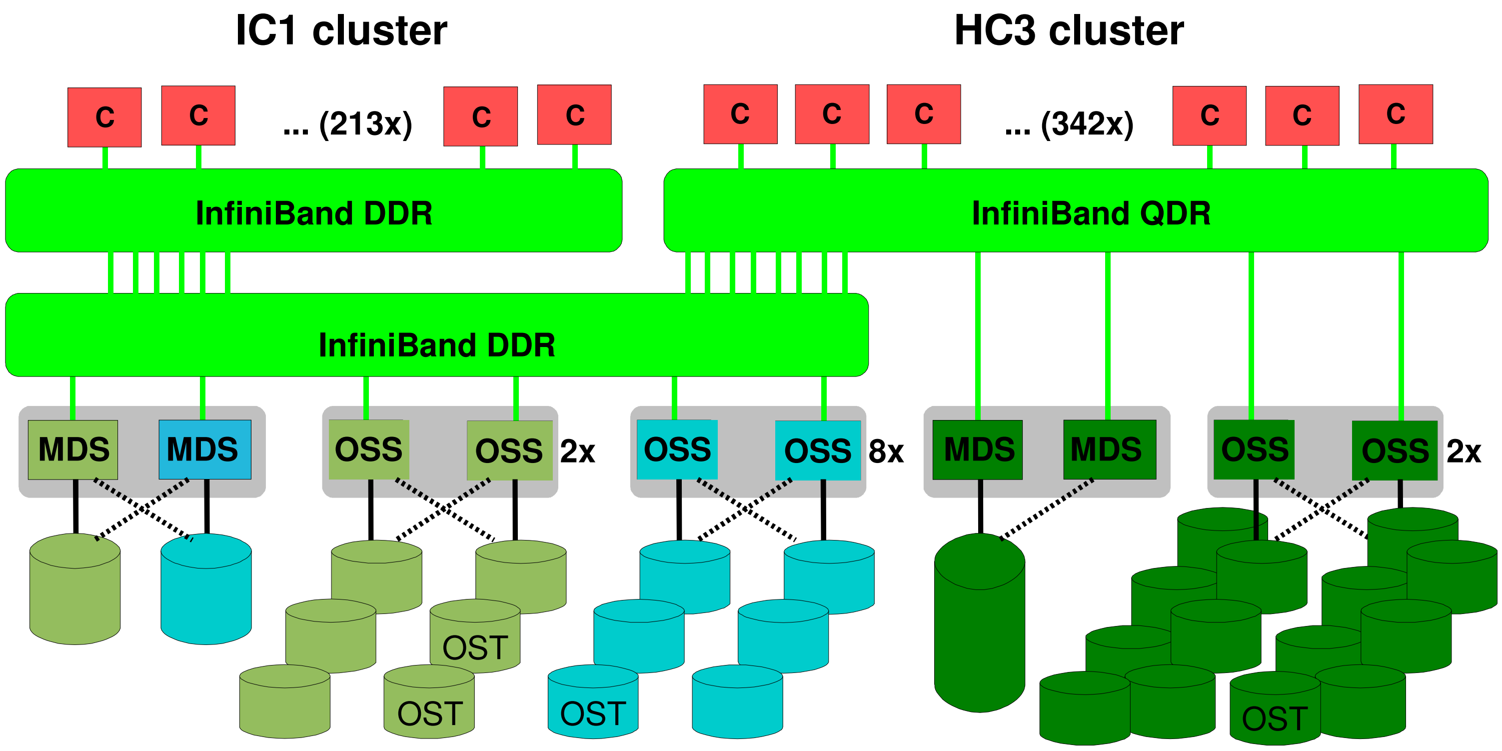
STEINBUCH CENTRE FOR COMPUTING - SCC

Basic Lustre concepts



- Lustre componets:
  - Clients (C) offer standard file system API
  - Metadata servers (MDS) hold metadata, e.g. directory data
  - Object Storage Servers (OSS) hold file contents and store them on Object Storage Targets (OSTs)
  - All communicate efficiently over interconnects, e.g. with RDMA

Lustre file systems at HC3



File system	$HOME	$PFSWORK	$WORK
Capacity (TiB)	76	301	203
Storage hardware	transtec provigo	transtec provigo	DDN S2A9900
# of OSTs	12	48	28
# of OST disks	192	768	290

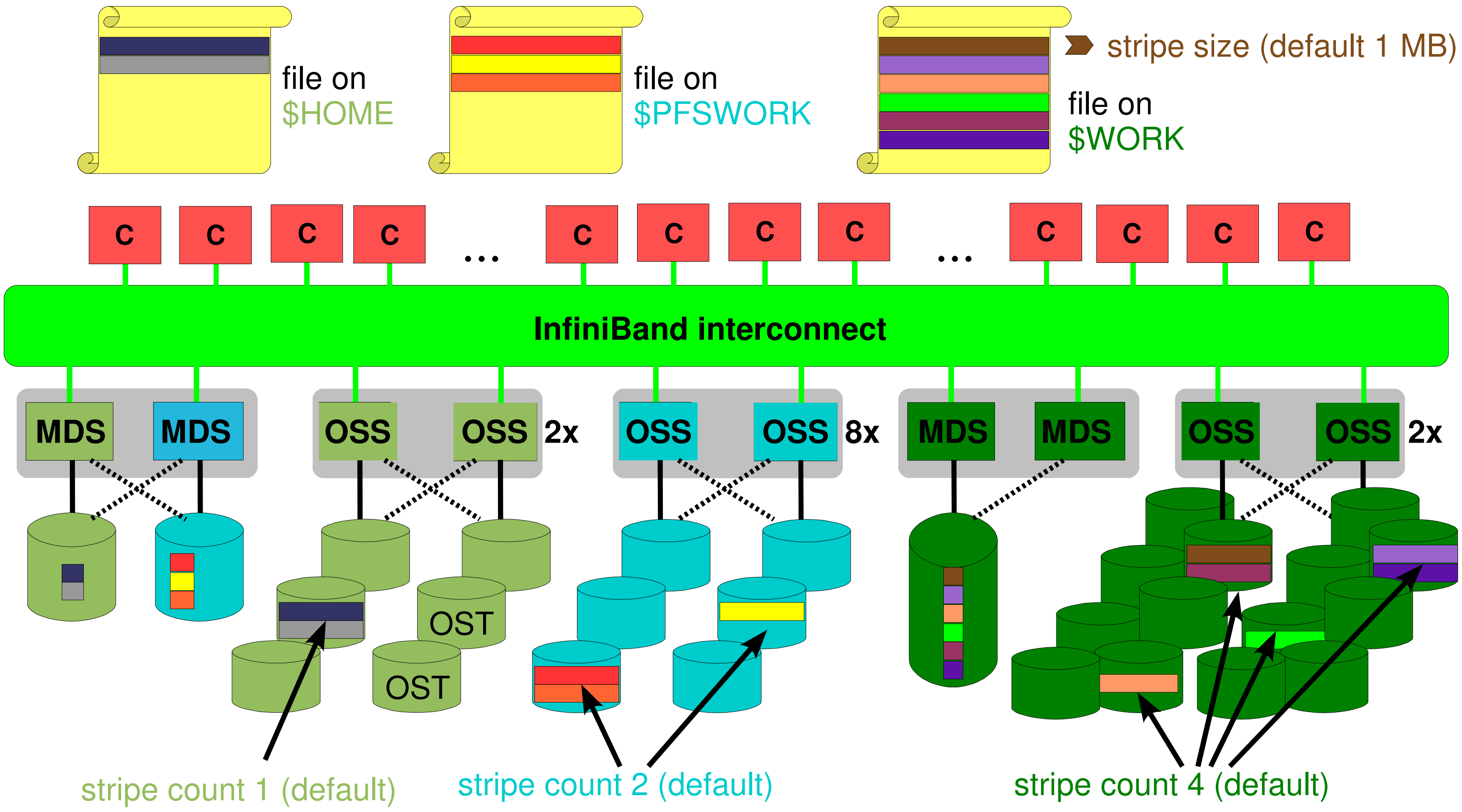
File system properties

Property	$TMP	$HOME	$WORK	$PFSWORK
Visibility	local node	HC3, IC1	HC3	HC3, IC1
Lifetime	batch job	permanent	> 7 days	> 7 days
Capacity	thin/med. | fat | login 129 | 673 | 825 GB	76 TB	203 TB	301 TB
Quotas	no	not enforced	not enforced	not enforced
Backup	no	yes	no	no
Read perf. / node	thin/medium | fat 70 | 250 MB/s	for IC1 nodes 600 MB/s	1800 MB/s	for IC1 nodes 600 MB/s
Write perf. / node	thin/medium | fat 80 | 390 MB/s	for IC1 nodes 700 MB/s	1800 MB/s	for IC1nodes 800 MB/s
Total read perf.	n*70 | n*250 MB/s	1700 MB/s	4800 MB/s	6200 MB/s
Total write perf.	n*80 | n*390 MB/s	1500 MB/s	4800 MB/s	5500 MB/s

Which file system to use?

- Follow these recommendations:
  - Whenever possible use $TMP for scratch data
  - Use $WORK for scratch data and job restart files
  - Use $PFSWORK to share scratch data between clusters
  - Use $HOME for long living data and when backup is required
  - Archive huge and unused data sets

How does striping work?

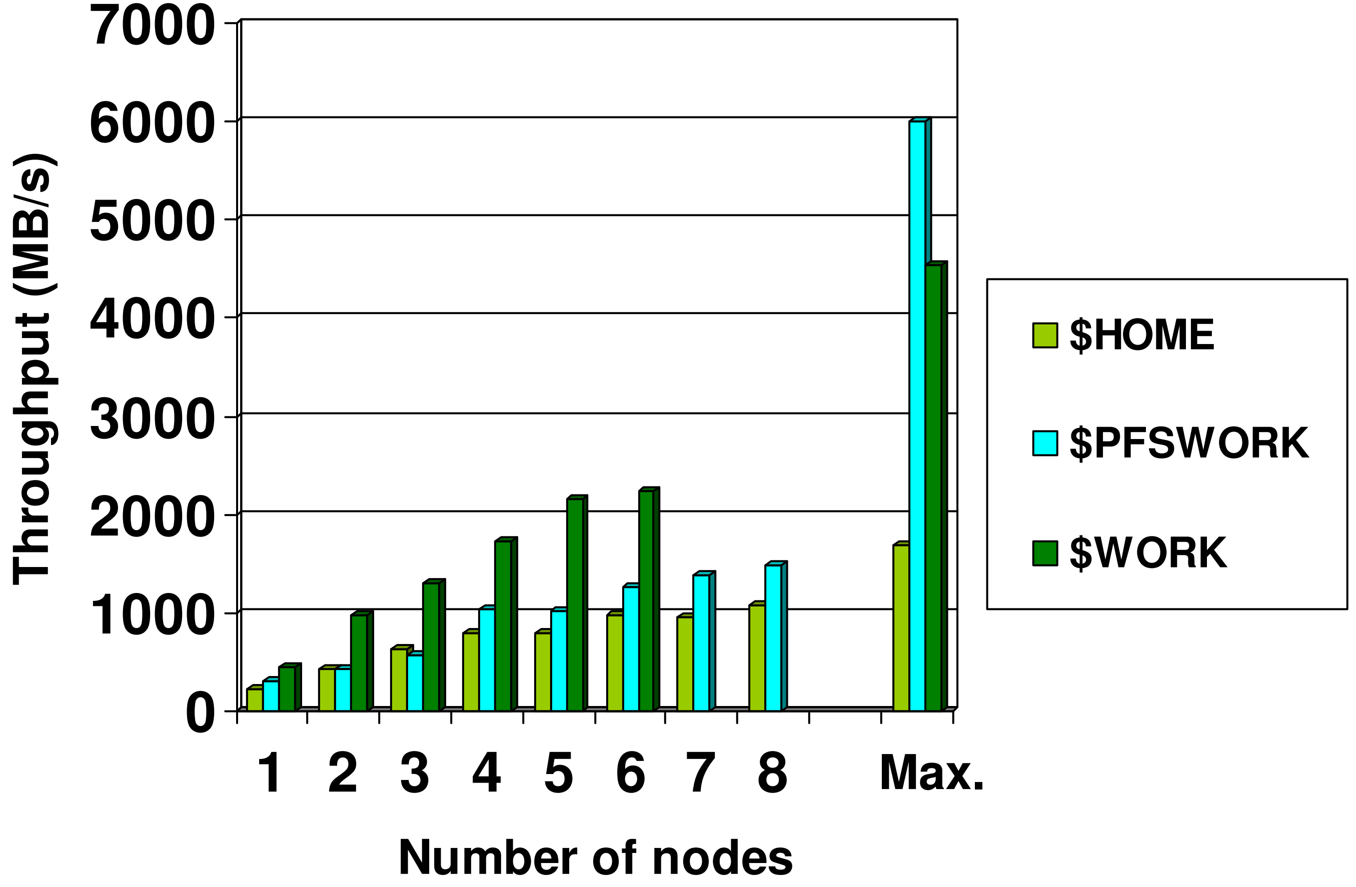


- Parallel data paths from clients to storage

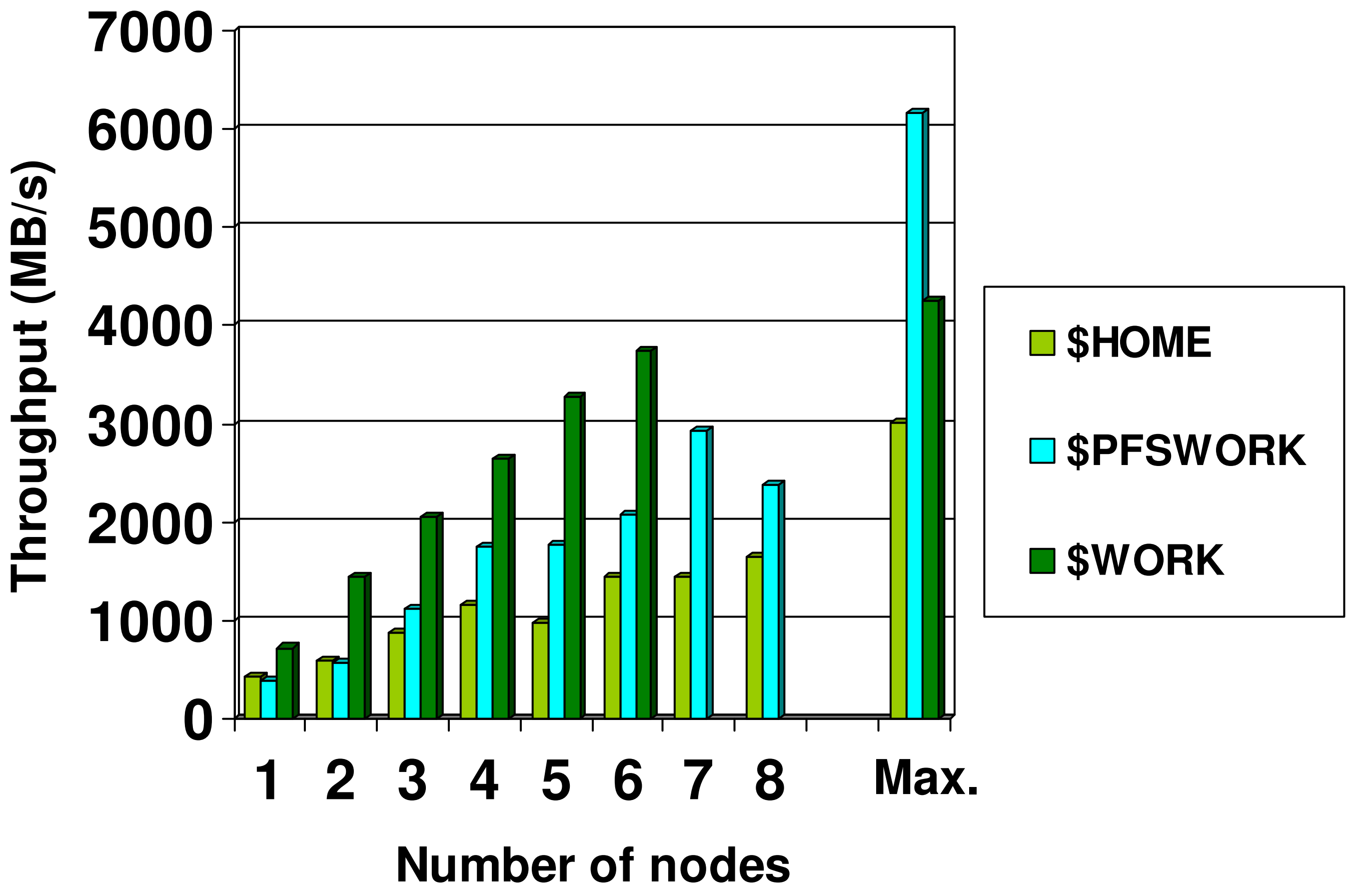
Using striping parameters

- Important hints to use striping
  - Striping parameters inherited from parent directory or file system default
    - To change parameters for existing file, copy it and move it back to the old name
  - Each new file is automatically created on new set of OSTs
    - No need to adapt striping parameters if many files are used in similar way!
- Adapt striping parameters to increase performance
  - OST performance is usually limited by storage subsystem
    - At HC3 pretty similar for all 3 file systems: ~200 MB/s
    - Increasing the stripe count might improve performance with few large files
  - Adapt stripe size to match application I/O pattern
    - Only makes sense in rare cases
- Show and adapt striping parameters
  - Show parameters: *lfs getstripe <file|directory>*
  - Change the stripe count: *lfs setstripe –c <count> <file|directory>*
  - Change the stripe size: *lfs setstripe –s <size> <file|directory>*

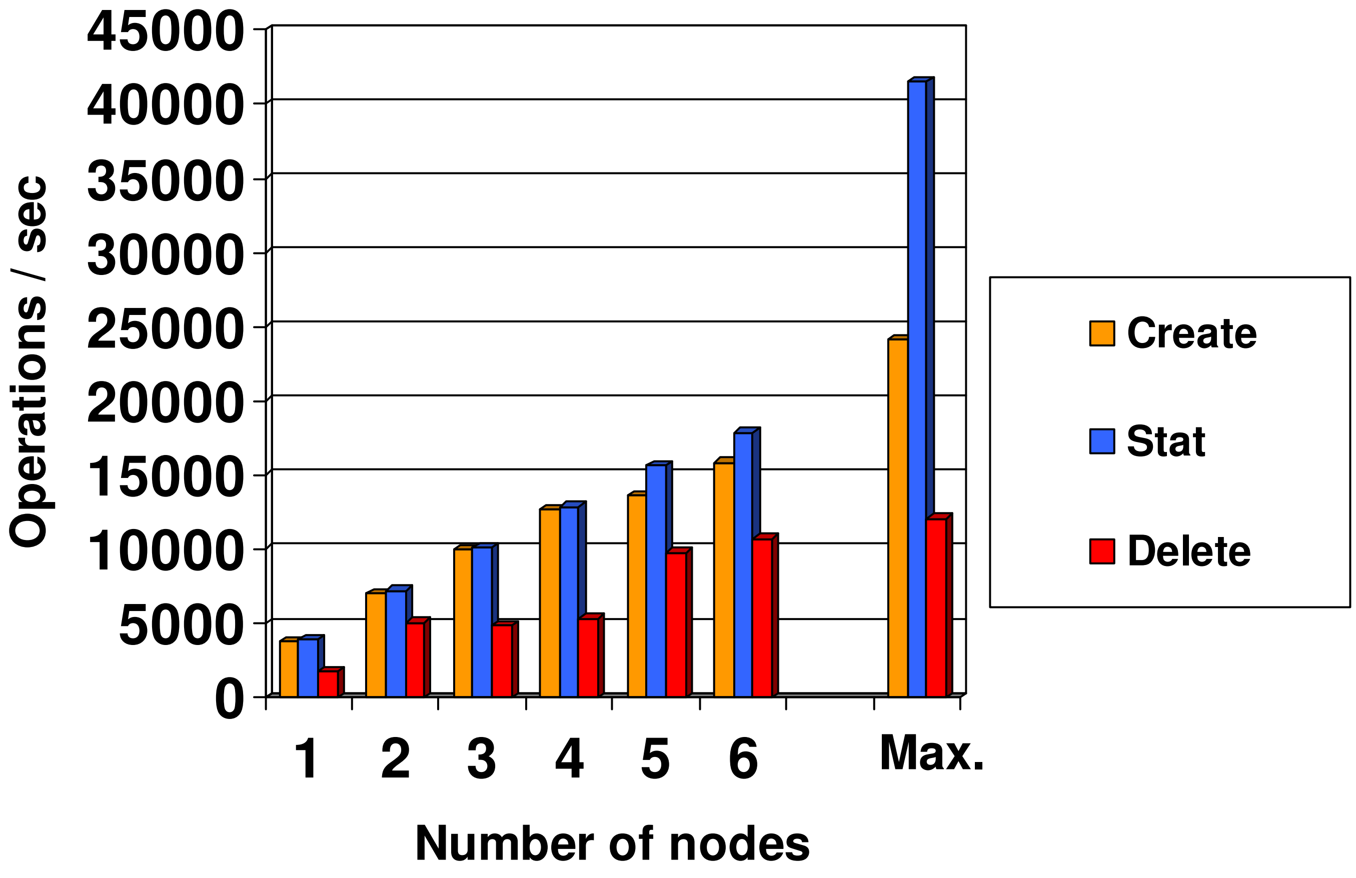
Write performance with default stripe count



Read performance with default stripe count



Metadata performance



Best practises (1)

- Change nothing if you use few (< 100) small (< 10MB) files
- Increasing throughput performance:
  - Use moderate stripe count (4 or 8) if only one task is doing IO
    - Improves single file bandwidth per client
  - To exploit complete file system bandwidth use several clients
    - Different files from different clients are automatically distributed
    - For one shared file use chunks with boundaries at stripe size
    - If many tasks use few huge files set stripe count to -1
  - Use stripe count 1 if lots of files are used in the same way
  - Collect large chunks of data and write them sequentially at once
  - Avoid competitive file access
    - e.g. writing to the same chunks or appending from different clients
    - e.g. competitive read/write access
  - Use $TMP whenever possible
    - If data is used by one client and is small enough for local hard drives

Best practises (2)

- Increasing metadata performance:
  - Avoid creating many small files
    - For parallel file systems metadata performance is limited
  - Avoid searching in huge directory trees
  - Avoid competitive directory access
    - e.g. by creating files for each task in separate subdirectories
  - If lots of files are created use stripe count 1
  - Use $TMP whenever possible
    - If data is used by one client and is small enough for local hard drives
    - This also allows to reduce compilation times
  - Change the default colorization setting of the ls command
    - alias ls='ls -color=tty‘
    - Otherwise ls command needs to contact OSS in addition to MDS

Understanding application I/O behaviour

- Application properties with impact on I/O:
  - Number and size of files
  - Access pattern, i.e. random or sequential
  - Buffer size of file operations
  - Type and number of operations (read, write, create, delete, stat)
  - Number of clients executing the operations
- External factors with impact on application I/O:
  - Overall usage of file system components
    - Storage, OSS, MDS, networks, clients
    - Administrators can tell the current status
- For help to analyse and improve I/O performance
  - contact roland.laifer@kit.edu

SimLabs@KIT

Frank Schmitz

STEINBUCH CENTRE FOR COMPUTING - SCC

Steinbuch Centre for Computing (SCC)

SCC at
Karlsruhe University

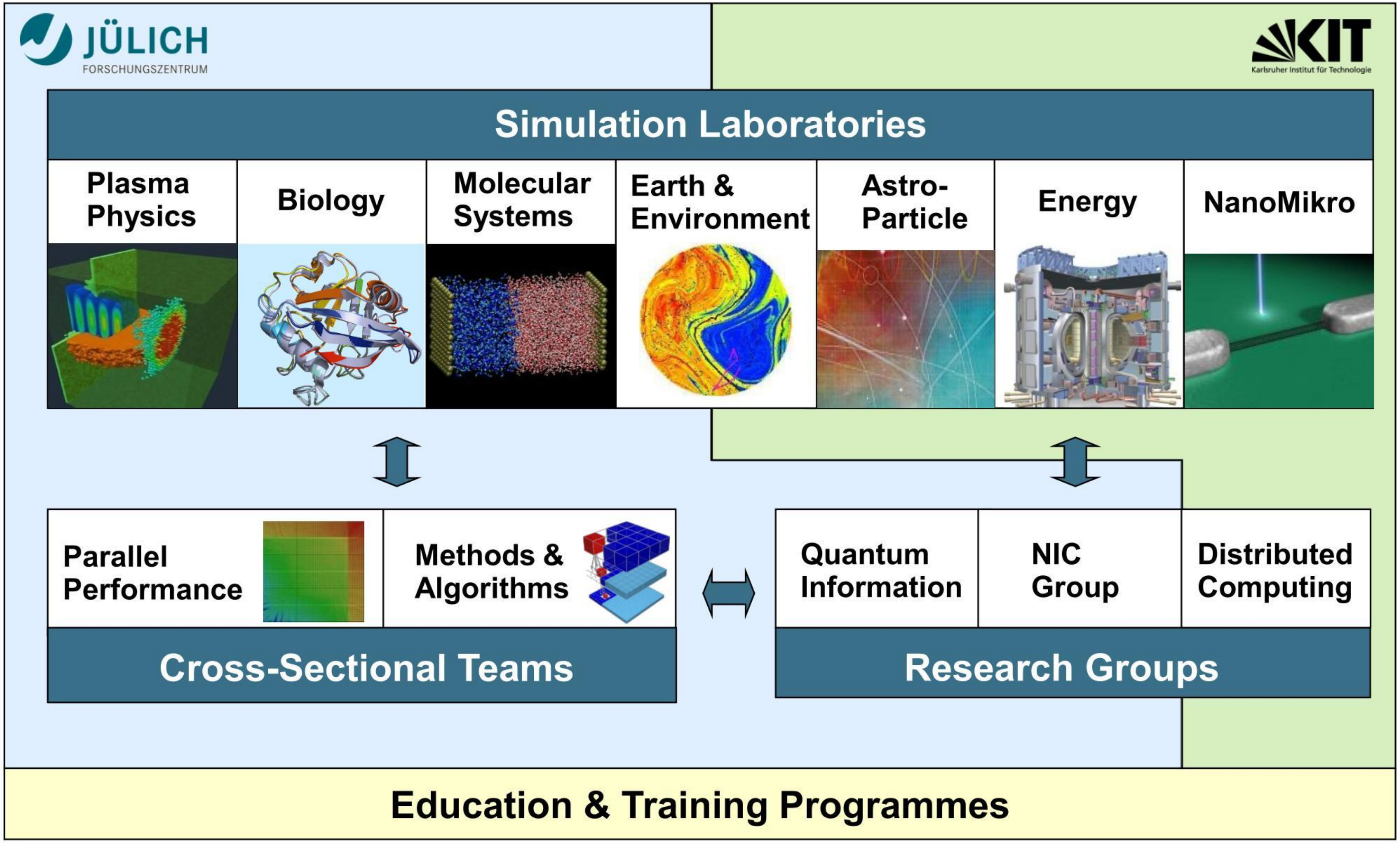
SCC at Research Center

- Founded on January 1st, 2008
- Information Technology Center of KIT
- Merger of the Computing Center of Karlsruhe University and Research Center Karlsruhe
- One of the largest scientific computing centers in Europe

Simulation Laboratory (SimLab)

- One major idea with Forschungszentrum Jülich
- Discussion started autumn 2008
- Program-oriented funding in the Research Field Key Technologies by the Helmholtz Association
- Review in spring 2009
- Funding starts 1.1.2010

SimLabs as part of the Program „Supercomputing“



Simulation Laboratory (SimLab)

- Mission of the SimLabs is R&D&I and the support of applications coming from different scientific areas at KIT. The activities of the four SimLabs are very close to the KIT Centers.

- SimLabs are the glue between high performance (HPC) and data intensive computing (DIC) , science and scientific computing.

- SimLabs support HPC&DIC applications and are part of the SCC service concept, and support scientists by using HPC resources outside KIT.

Simulation Laboratory (SimLab)

- activities in education (Ph.D. students, students, interns)
- integration of projects (Young Investigator Group → confirmed, HPC-5 since 2010, MMM@HPC since 2010→ FP7, other internal KIT-projects)
- SimLabs at KIT are involved in external projects with and without KIT institutes. But the main activities are within KIT.

Simulation Laboratories: An Innovative Community-Oriented Research and Support Structure
Attig, Norbert; Esser, Rüdiger; Gibbon, Paul (2008)
Proceedings of the Cracow Grid Workshop (CGW'07), 16. - 18. Oktober 2007
eds.: M. Bubak, M. Turala, K. Wiatr. - Krakow, Polen, ACC CYFRON ET AGH, 2008. - 978-83-915141-9-1. - S. 1 - 9

Integration of the SimLabs into SCC

SimLab Energy (Olaf Schneider)

SimLab NanoMikro (Ivan Kondov)

SimLab Climate and Environment (Oliver Kirner)

SimLab Elementary Particle and Astroparticle Physics (Gevorg Poghosyan)

Cross-Sectional Team Enhanced Scalability as a connector of the SimLabs

Research Group Distributed Computing

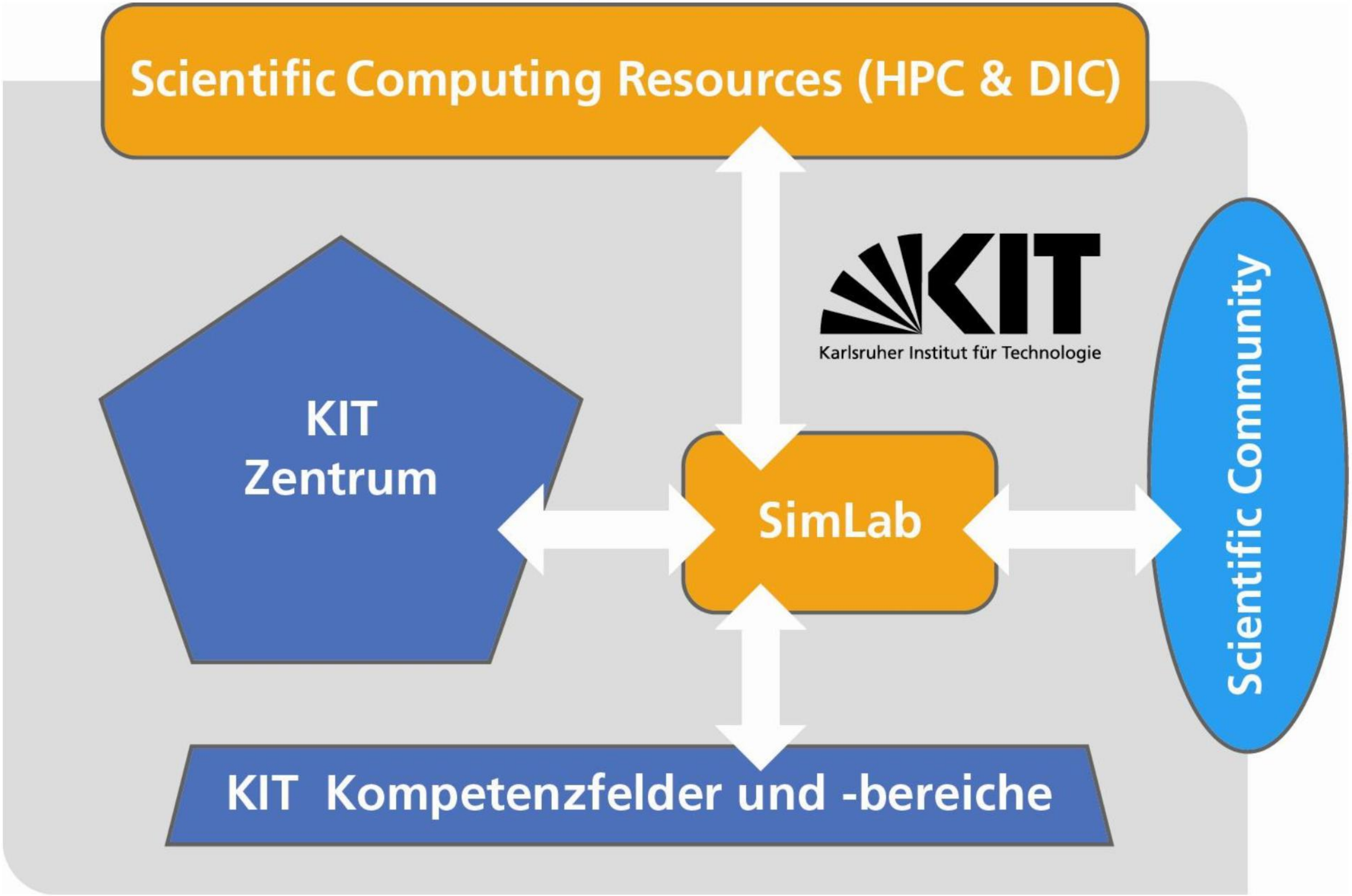
Research Group Cloud Computing

Storage Services (Large Scale Data Facilitie →LSDF)

other SCC Services

Compute Services (HPC)

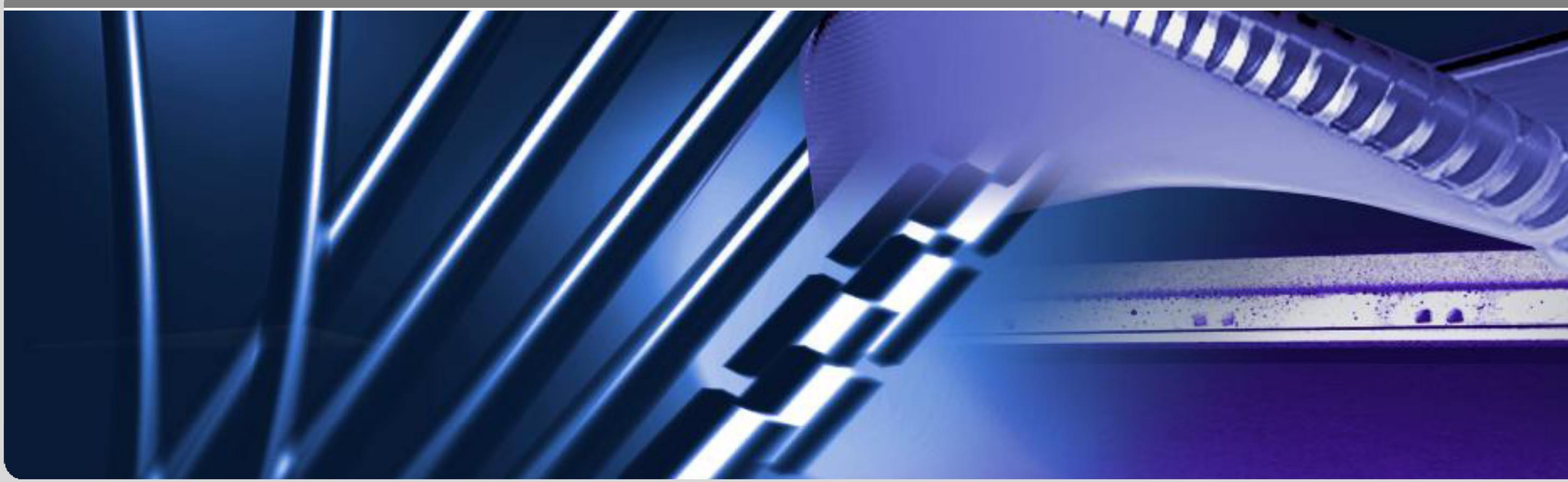
SimLabs as advanced support for Users at KIT



Software Packages on HC3

Paul Weber

STEINBUCH CENTRE FOR COMPUTING - SCC



Overview

- All important distributors of CAE programs provide their applications on cluster computers
- Large performance improvements by parallelization, potentially only SMP
- CAE program packages cover problem solutions in the area of
  - Structural mechanics and heat transfer
  - Computational fluid dynamics (CFD)
  - Electrodynamics and propagation of electromagnetic fields
  - Couplings of these phenomena (multiphysics)
- Mathematical methods: Finite Element Method, Finite Volume Method, Lattice Boltzmann

Course of a project

Geometry and Modelling Meshing Physical properties	Preprocessing	Proprietary Modules universal Preprocessors
Numerical Solution	Analysis	Solver Modules
Results: Visualization Animation	Postprocessing	Proprietary Modules universal Postprocessors

Some important CAE packages on the HC3

ABAQUS, ANSYS	structural mechanics, heat transfer,CFD implicit/explicit, linear/nonlinear Multiphysics, SMP and DMP
ADINA	structural mechanics, heat transfer,CFD implicit/explicit, linear/nonlinear, SMP and DMP Multiphysics
MD Nastran	structural mechanics, heat transfer implicit, linear, SMP and DMP coupling to nonlinear and explicit codes of Dytran and MSC.Marc
MSC.Marc	structural mechanics, heat transfer, CFD, electro-/magnetostatics implicit, nonlinear, SMP and DMP
LS-DYNA	structural mechanics, explicit dynamics, SMP and DMP

Some important CAE packages on the HC3

ANSYS Fluent ANSYS CFX	CFD, fluids, particles, combustion, chemical reactions, multi phase, turbo machinery, SMP and DMP
STAR-CD STAR-CCM+	CFD, fluids, particles, combustion, chemical reactions, phase transitions, SMP and DMP
PowerFLOW	CFD, Lattice Boltzmann, external /internal flows, DMP
COMSOL Multiphysics	general PDEs, specialized modules for structural, CFD, electromagnetics, chemical engineering, heat transfer, MEMS, earth science, fuel cells, Multiphysics, coupling with MATLAB, SMP and DMP
MATLAB	macro language for numerical calculations, many special toolboxes in many areas of engineering, finance, statistics, image processing et al. DMP (and SMP)

Miscellaneous programs

- Preprocessing tools for modelling and meshing
  - ANSYS Workbench, HyperMESH, ANSYS ICEM_CFD, Patran, Gambit/TGRID
- Postprocessing tools for results visualization and animation
  - ANSYS Workbench, HyperVIEW, Patran, EnSight

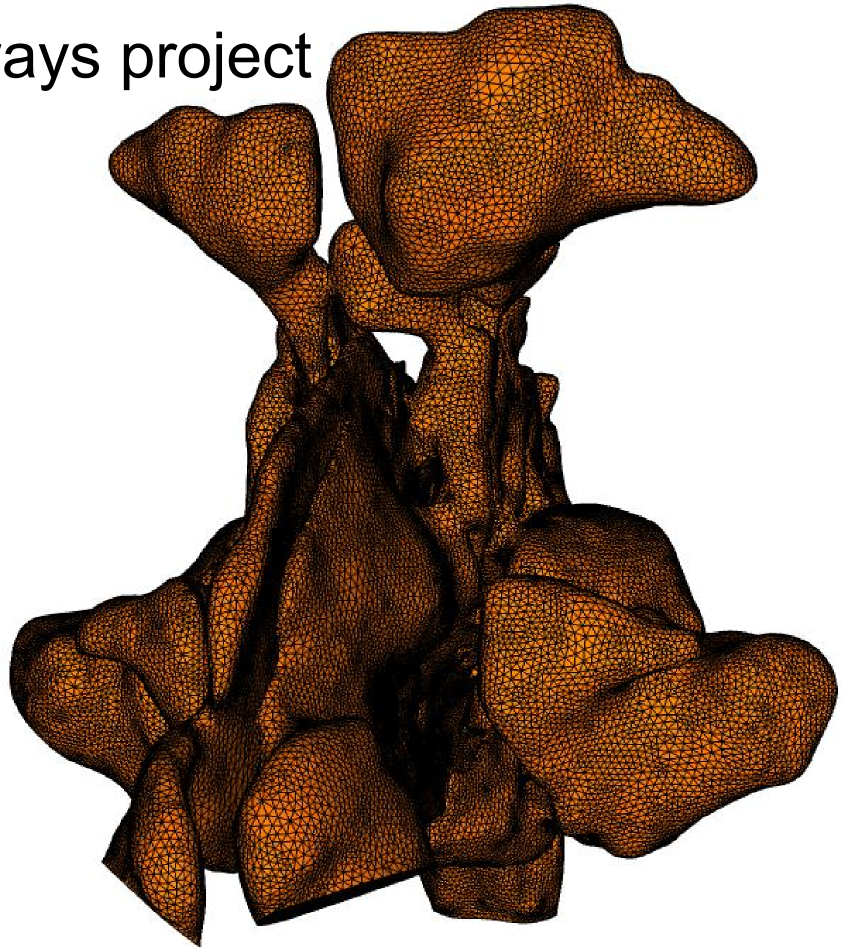
Running CAE programs in the HC3 environment

- Nonstandard batch environment (JMS)
- Starting a batch job with the **`job_submit`** command
  - allocation of memory, cores, cpu time
  - administration of the batch queue
  - initialization of the job
- Launching of the CAE programmes by individual commands
  - Combination of **`job_submit`** and the original program calls
  - request of system resources and setting the job parameters by a single command
- FLUENT Example

```
fluentjob -j NAME -v VERSION -t CPU-TIME -m MEMORY
             [-c CLASS][-d NODE][-T TIME][-p PROCS]
```

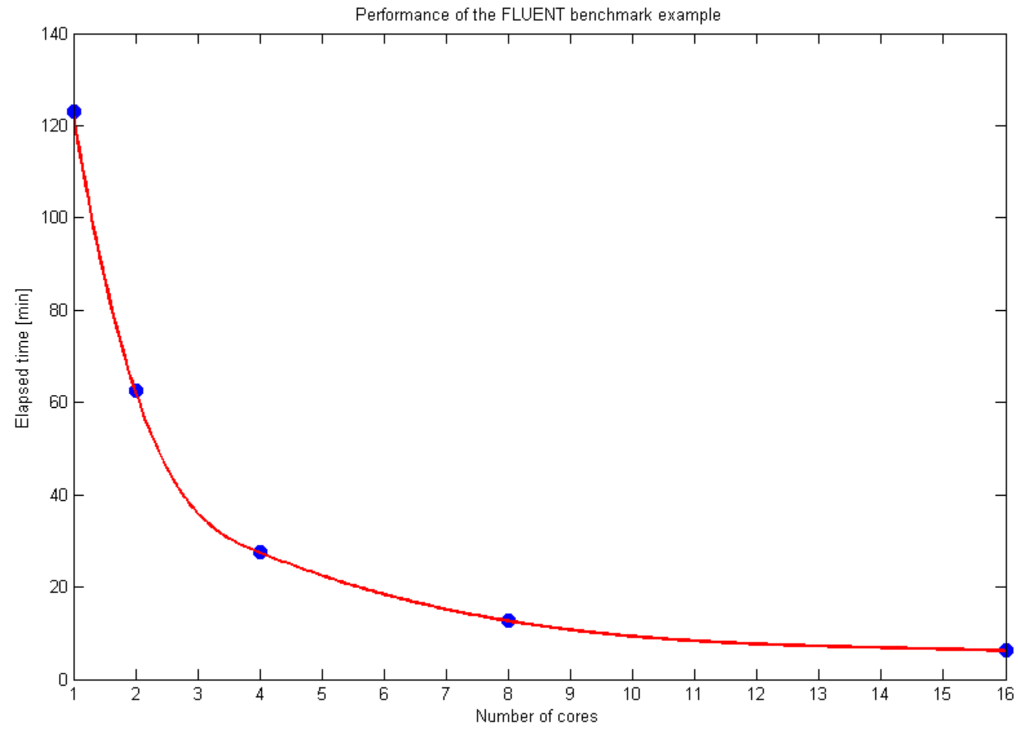
Fluent Example Benchmark

- Model of a respiratory system
- Study for the United Airways project
- about 2 million cells
- only laminar flow,
  no turbulence modelling

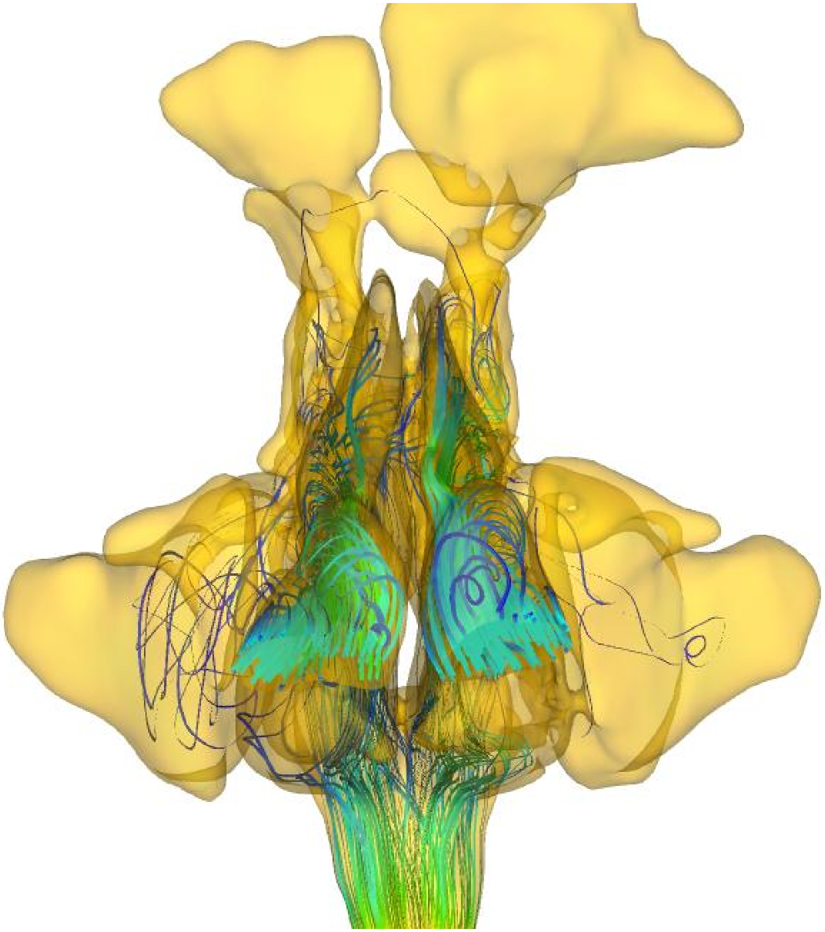


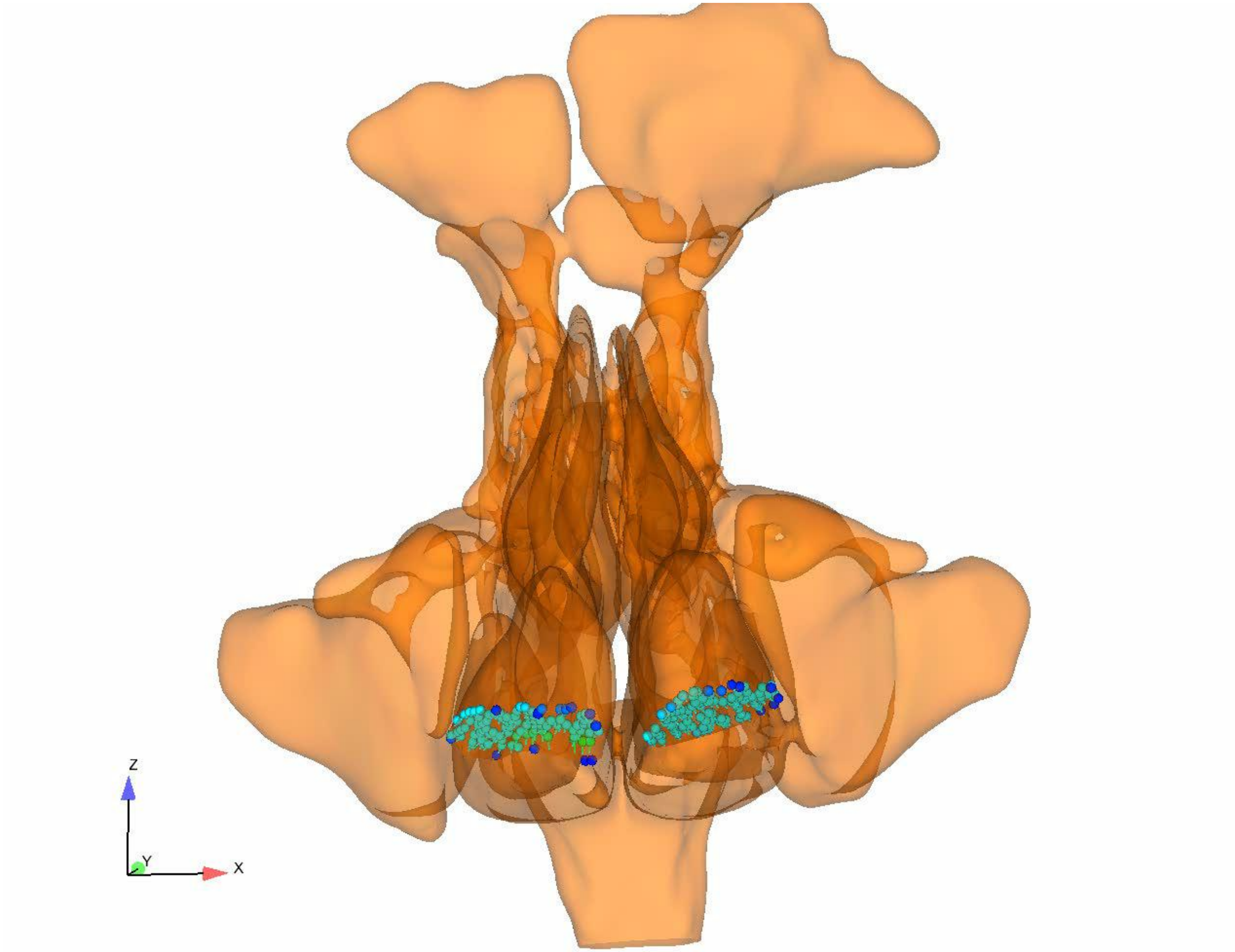
Scaling behavior

- Memory: 16 GByte/processor
- Distribution on up to 16 processors
- each process runs exclusively on one node
- Running processes on up to 8 cores on one node (SMP) shows almost identical performance



Results: velocity




Z
Y
X

Karlsruher Institut für Technologie

Thank you for your attention