Towards probabilistic weather forecasts – new developments of the Numerical Weather 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

- Observations
  - Global observing system
  - Observation errors
  - Redundancies
  - Bias corrections

- Data base
- Process
- HPC

- Quality control
- Estimation of the initial values

- Data assimilation
- Model solution

- Forecast
- Archive

- Post-processing
- Products for weather advisers, end users, international exchange

- Verification
- Quality assurance

- Archive
- Process
- HPC
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

<table>
<thead>
<tr>
<th>Medium and long range</th>
<th>+1 d - +2W</th>
<th>Deterministic, probabilistic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>+1 W - +12 W</td>
<td>Probabilistic</td>
</tr>
<tr>
<td>Under development</td>
<td>+1 M - +12 M</td>
<td>Probabilistic</td>
</tr>
<tr>
<td><strong>Short range</strong></td>
<td>+5 h - + 72 h</td>
<td>Deterministic, probabilistic</td>
</tr>
<tr>
<td><strong>Very short range</strong></td>
<td>+1h - + 24 h</td>
<td>Deterministic</td>
</tr>
<tr>
<td>Under development</td>
<td>+2h - + 24 h</td>
<td>Probabilistic</td>
</tr>
<tr>
<td><strong>Challenge</strong></td>
<td>+10min - +120min</td>
<td>Deterministic</td>
</tr>
</tbody>
</table>

ECMWF

European NMHS
Deterministic Nonperiodic Flow

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.
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.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$X$</th>
<th>$Y$</th>
<th>$Z$</th>
<th>$N$</th>
<th>$X$</th>
<th>$Y$</th>
<th>$Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0045</td>
<td>0174</td>
<td>0035</td>
<td>0483</td>
<td>3092</td>
<td>0117</td>
<td>0075</td>
<td>0352</td>
</tr>
<tr>
<td>0107</td>
<td>-0091</td>
<td>-0083</td>
<td>0287</td>
<td>3098</td>
<td>0123</td>
<td>0076</td>
<td>0365</td>
</tr>
<tr>
<td>0168</td>
<td>-0092</td>
<td>-0084</td>
<td>0288</td>
<td>3171</td>
<td>0134</td>
<td>0082</td>
<td>0383</td>
</tr>
<tr>
<td>0230</td>
<td>-0092</td>
<td>-0084</td>
<td>0289</td>
<td>3268</td>
<td>0153</td>
<td>0069</td>
<td>0435</td>
</tr>
<tr>
<td>0292</td>
<td>-0092</td>
<td>-0083</td>
<td>0290</td>
<td>3333</td>
<td>-0114</td>
<td>-0079</td>
<td>0342</td>
</tr>
<tr>
<td>0354</td>
<td>-0093</td>
<td>-0083</td>
<td>0292</td>
<td>3400</td>
<td>-0117</td>
<td>-0077</td>
<td>0350</td>
</tr>
<tr>
<td>0416</td>
<td>-0093</td>
<td>-0083</td>
<td>0293</td>
<td>3468</td>
<td>-0125</td>
<td>-0083</td>
<td>0361</td>
</tr>
<tr>
<td>0478</td>
<td>-0094</td>
<td>-0082</td>
<td>0295</td>
<td>3541</td>
<td>-0129</td>
<td>-0073</td>
<td>0378</td>
</tr>
<tr>
<td>0540</td>
<td>-0094</td>
<td>-0082</td>
<td>0296</td>
<td>3625</td>
<td>-0146</td>
<td>-0074</td>
<td>0413</td>
</tr>
<tr>
<td>0602</td>
<td>-0095</td>
<td>-0082</td>
<td>0298</td>
<td>3695</td>
<td>0127</td>
<td>0079</td>
<td>0370</td>
</tr>
<tr>
<td>0664</td>
<td>-0096</td>
<td>-0083</td>
<td>0300</td>
<td>3772</td>
<td>0136</td>
<td>0072</td>
<td>0394</td>
</tr>
<tr>
<td>0726</td>
<td>-0097</td>
<td>-0083</td>
<td>0302</td>
<td>3853</td>
<td>0144</td>
<td>0077</td>
<td>0407</td>
</tr>
<tr>
<td>0788</td>
<td>-0097</td>
<td>-0081</td>
<td>0304</td>
<td>3926</td>
<td>0129</td>
<td>0072</td>
<td>0380</td>
</tr>
<tr>
<td>0851</td>
<td>-0099</td>
<td>-0083</td>
<td>0307</td>
<td>4014</td>
<td>0148</td>
<td>0068</td>
<td>0421</td>
</tr>
<tr>
<td>0914</td>
<td>-0100</td>
<td>-0081</td>
<td>0309</td>
<td>4082</td>
<td>-0120</td>
<td>-0074</td>
<td>0359</td>
</tr>
<tr>
<td>0977</td>
<td>-0100</td>
<td>-0080</td>
<td>0312</td>
<td>4153</td>
<td>-0129</td>
<td>-0078</td>
<td>0375</td>
</tr>
<tr>
<td>1040</td>
<td>-0102</td>
<td>-0080</td>
<td>0315</td>
<td>4233</td>
<td>-0144</td>
<td>-0082</td>
<td>0404</td>
</tr>
<tr>
<td>1103</td>
<td>-0104</td>
<td>-0081</td>
<td>0319</td>
<td>4307</td>
<td>0135</td>
<td>0081</td>
<td>0385</td>
</tr>
<tr>
<td>1167</td>
<td>-0105</td>
<td>-0079</td>
<td>0323</td>
<td>4417</td>
<td>-0162</td>
<td>-0069</td>
<td>0450</td>
</tr>
<tr>
<td>1231</td>
<td>-0107</td>
<td>-0079</td>
<td>0328</td>
<td>4480</td>
<td>0106</td>
<td>0081</td>
<td>0324</td>
</tr>
<tr>
<td>1295</td>
<td>-0111</td>
<td>-0082</td>
<td>0333</td>
<td>4544</td>
<td>0109</td>
<td>0082</td>
<td>0329</td>
</tr>
<tr>
<td>1360</td>
<td>-0111</td>
<td>-0077</td>
<td>0339</td>
<td>4609</td>
<td>0110</td>
<td>0080</td>
<td>0334</td>
</tr>
<tr>
<td>1427</td>
<td>-0116</td>
<td>-0079</td>
<td>0347</td>
<td>4675</td>
<td>0112</td>
<td>0076</td>
<td>0341</td>
</tr>
<tr>
<td>1493</td>
<td>-0120</td>
<td>-0077</td>
<td>0357</td>
<td>4741</td>
<td>0118</td>
<td>0081</td>
<td>0349</td>
</tr>
<tr>
<td>1566</td>
<td>-0125</td>
<td>-0072</td>
<td>0371</td>
<td>4810</td>
<td>0120</td>
<td>0074</td>
<td>0360</td>
</tr>
<tr>
<td>1643</td>
<td>-0139</td>
<td>-0077</td>
<td>0396</td>
<td>4881</td>
<td>0130</td>
<td>0081</td>
<td>0376</td>
</tr>
<tr>
<td>1722</td>
<td>-0140</td>
<td>0075</td>
<td>0401</td>
<td>4963</td>
<td>0141</td>
<td>0068</td>
<td>0406</td>
</tr>
<tr>
<td>1798</td>
<td>-0135</td>
<td>-0072</td>
<td>0391</td>
<td>5035</td>
<td>-0133</td>
<td>-0081</td>
<td>0381</td>
</tr>
<tr>
<td>1882</td>
<td>-0146</td>
<td>0074</td>
<td>0413</td>
<td>5124</td>
<td>-0151</td>
<td>-0076</td>
<td>0422</td>
</tr>
<tr>
<td>1952</td>
<td>-0127</td>
<td>-0078</td>
<td>0370</td>
<td>5192</td>
<td>0119</td>
<td>0075</td>
<td>0358</td>
</tr>
<tr>
<td>2029</td>
<td>-0135</td>
<td>-0070</td>
<td>0393</td>
<td>5262</td>
<td>0129</td>
<td>0083</td>
<td>0372</td>
</tr>
<tr>
<td>2110</td>
<td>-0146</td>
<td>0083</td>
<td>0408</td>
<td>5340</td>
<td>0140</td>
<td>0079</td>
<td>0397</td>
</tr>
<tr>
<td>2183</td>
<td>-0128</td>
<td>-0070</td>
<td>0379</td>
<td>5419</td>
<td>-0137</td>
<td>-0067</td>
<td>0399</td>
</tr>
<tr>
<td>2268</td>
<td>-0144</td>
<td>-0066</td>
<td>0415</td>
<td>5495</td>
<td>0140</td>
<td>0081</td>
<td>0394</td>
</tr>
<tr>
<td>2337</td>
<td>-0126</td>
<td>0079</td>
<td>0368</td>
<td>5576</td>
<td>-0141</td>
<td>-0072</td>
<td>0405</td>
</tr>
<tr>
<td>2412</td>
<td>-0137</td>
<td>0081</td>
<td>0389</td>
<td>5649</td>
<td>0135</td>
<td>0082</td>
<td>0384</td>
</tr>
<tr>
<td>2501</td>
<td>-0153</td>
<td>-0080</td>
<td>0423</td>
<td>5752</td>
<td>0160</td>
<td>0074</td>
<td>0443</td>
</tr>
<tr>
<td>2569</td>
<td>-0139</td>
<td>0076</td>
<td>0387</td>
<td>5816</td>
<td>-0110</td>
<td>-0081</td>
<td>0332</td>
</tr>
<tr>
<td>2639</td>
<td>-0129</td>
<td>0082</td>
<td>0371</td>
<td>5881</td>
<td>-0113</td>
<td>-0082</td>
<td>0339</td>
</tr>
<tr>
<td>2717</td>
<td>-0136</td>
<td>0070</td>
<td>0395</td>
<td>5948</td>
<td>-0114</td>
<td>-0075</td>
<td>0346</td>
</tr>
<tr>
<td>2796</td>
<td>-0143</td>
<td>-0079</td>
<td>0402</td>
<td>2871</td>
<td>0134</td>
<td>0076</td>
<td>0388</td>
</tr>
<tr>
<td>2862</td>
<td>-0132</td>
<td>-0072</td>
<td>0426</td>
<td>2846</td>
<td>-0135</td>
<td>0074</td>
<td>0388</td>
</tr>
</tbody>
</table>

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 $C'$. 
E. N. Lorenz (1963):

„The computations have been performed on a Royal McBee LGP-30 electronic computing machine. Approximately one second per iteration, aside from 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)

Initial values and Boundary conditions

Model physic

IFS
GME
GFS
UM

Lhn_coeff=0.5
Visualisation

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 2\textsuperscript{nd} 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/](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)

- 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

Vorticity
Divergence
optionally averaged
Gradient

Vorticity

Divergence

Gradient

tangential wind
RBF reconstruction
Demonstration example for grid nesting
Vorticity at lowest model level on day 20

high-resolution (35 km)

coarse-resolution (140 km)

Nested (140 / 35 km)
Vorticity at lowest model level on day 20 (zoom)

nested (innermost domain; 35 km)  high-resolution (35 km)
Current parallelization levels

- Hardware aspects abstraction
- Performance aspects

- Node
- Chip Module
- Core

- Multi Node
- MPI-Network
- MPI-shmem

- Cache/Memory domain
- OpenMP
- Cache/Vector blocking
- Loop blocking

- MPI domain decomposition
- Blocking over innermost loop and levels
- Innermost loop length
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 prediction system
  - $10^8$ degrees of freedom

- Variational problem with $10^8$ variables
  - Most probable state of the atmosphere is defined by the minimum of
    \[
    J(x) = (x - x_B)^T B^{-1} (x - x_B) + (y_O - H(x))^T O^{-1} (y_O - H(x))
    \]
Recent Developments in data assimilation

→ Prediction of flow dependent model error covariance matrix $\textbf{B}$
  
  → 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 resources
  - Increasing resolution
  - Comprehensive physics
  - Ensemble prediction systems (EPS) on all scales
  - High resolution Ensemble data assimilation
with Acknowledgement to
D. Majewski
S. Theis
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

2004
IBM Power 4+
JUMP, 9 TFlop/s

2005/6
IBM Blue Gene/L
JUBL, 45 TFlop/s

2007/8
IBM Blue Gene/P
JUGENE, 223 TFlop/s

2009
IBM Blue Gene/P
JUGENE, 1 PFlop/s

Intel Nehalem Clusters
JUROPA
200 TFlop/s
HPC-FF
100 TFlop/s

File Server
GPFS

File Server
GPFS, Lustre
Use by Scientific Discipline

Leadership-Class System

- Astrophysics
- Biophysics
- Chemistry

General-Purpose Supercomputer

- Earth & Environment
- Plasma Physics
- Soft Matter
- Fluid Dynamics

JUGENE
~40 Projects

JUROPA
~85 Projects

Granting period
07/2009 – 04/2010

Oversubscription factor > 5

Use by Scientific Discipline

- Elementary Particle Physics
- Computer Science
- Condensed Matter
- Material Science

29.11.2010
Simulation Labs at JSC
Primary Application Support at JSC

User Support

- Advisor
- Methods and Optimisation
- Technical Support

Training

- High-level Workshops
- Tutorials and Seminars
- Courses
Performance Development in Top500

- 1 Eflop/s
- 100 Pflop/s
- 1 Pflop/s
- 100 Tflop/s
- 10 Tflop/s
- 1 Tflop/s
- 100 Gflop/s
- 10 Gflop/s
- 1 Gflop/s
- 100 Mflop/s

SUM

N=1

N=500
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 → scalable algorithm design
- More sophisticated user-support structures: Simulation Labs
Domain-specific Research and Support

Simulation Laboratories

<table>
<thead>
<tr>
<th>Plasma Physics</th>
<th>Biology</th>
<th>Molecular Systems</th>
<th>Climate Science</th>
<th>Fluid Engineering</th>
<th>High Energy Physics</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Plasma Physics" /></td>
<td><img src="image" alt="Biology" /></td>
<td><img src="image" alt="Molecular Systems" /></td>
<td><img src="image" alt="Climate Science" /></td>
<td><img src="image" alt="Fluid Engineering" /></td>
<td><img src="image" alt="High Energy Physics" /></td>
</tr>
</tbody>
</table>

- together with RWTH Aachen University
- together with Cyprus Institute – CaSToRC

Cross-Sectional Teams

<table>
<thead>
<tr>
<th>Parallel Performance</th>
<th>Application Optimisation - Parateam</th>
<th>Methods &amp; Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Parallel Performance" /></td>
<td><img src="image" alt="Application Optimisation" /></td>
<td><img src="image" alt="Methods &amp; Algorithms" /></td>
</tr>
</tbody>
</table>
SimLab Teams

• Biology
  - Olav Zimmermann
  - Godehard Sutmann

• Molecular Systems
  - Paul Gibbon
  - Thomas Müller

• Plasma Physics
  - Lukas Arnold

• Climate Research
  - Lars Hoffmann (ICG-1)
  - Jan Meinke
  - Sandipan Mohanty
  - Annika Schiller
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
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 (raccoon, 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

• 1st 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](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)
Hardware on HC3 – Nehalem's Impact on Performance

Hartmut Häfner
Configuration of HC3 (HP XC3000)

Production nodes with 24, 48 and 144 GB main memory

InfiniBand 4X QDR

Login nodes
Service nodes

$WORK

InfiniBand 4X DDR

$HOME

8x
8x
8x
8x
8x
8x
8x
8x
8x
8x
8x
8x
8x
8x
8x
8x
8x
8x
8x
8x
8x
Detailed Configuration of HC3

- 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)

<table>
<thead>
<tr>
<th>Cache Level</th>
<th>Size</th>
<th>Access Rate</th>
<th>Directionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1-Cache</td>
<td>4 x 32 KB</td>
<td>2 words/cycle</td>
<td>bi-directional</td>
</tr>
<tr>
<td>L2-Cache</td>
<td>4 x 256 KB</td>
<td>4 words/cycle</td>
<td>uni-directional</td>
</tr>
<tr>
<td>L3-Cache</td>
<td>8 MB (shared)</td>
<td>1-4 words/cycle</td>
<td>uni-directional</td>
</tr>
<tr>
<td>Memory</td>
<td>Up to 32 GB</td>
<td>25.6/(20.24...80.96) = 0.31 -1.26 words/cycle</td>
<td></td>
</tr>
</tbody>
</table>
Nehalem's architectural Efficiency

Peak performance of one core: $4 \times 2.53 = 10.12$ Gflops

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

<table>
<thead>
<tr>
<th>Full Triad</th>
<th>Expected Theoretical Peak Performance</th>
<th>Architectural Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data from L1-cache</td>
<td>$10.12 / 2 = 5.06$ Gflops</td>
<td>0.5</td>
</tr>
<tr>
<td>Data from L2-cache</td>
<td>$10.12 / 2 = 5.06$ Gflops</td>
<td>0.5</td>
</tr>
<tr>
<td>Data from L3-cache</td>
<td>$10.12 / 2 \ldots 8 = 1.26 - 5.06$ Gflops</td>
<td>0.125 – 0.5</td>
</tr>
<tr>
<td>Data from memory</td>
<td>$10.12 \times (0.31 \ldots 1.26)/4 = 0.79 - 3.18$ Gflops</td>
<td>0.08 – 0.315</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Dot Product</th>
<th>Expected Theoretical Peak Performance</th>
<th>Architectural Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data from L1-cache</td>
<td>$10.12 / 1 = 10.12$ Gflops</td>
<td>1.0</td>
</tr>
<tr>
<td>Data from L2-cache</td>
<td>$10.12 / 1 = 10.12$ Gflops</td>
<td>1.0</td>
</tr>
<tr>
<td>Data from L3-cache</td>
<td>$10.12 / 1 \ldots 4 = 2.53 - 10.12$ Gflops</td>
<td>0.25 – 1.0</td>
</tr>
<tr>
<td>Data from memory</td>
<td>$10.12 \times (0.31 \ldots 1.26)/2 = 1.62 - 6.37$ Gflops</td>
<td>0.16 – 0.63</td>
</tr>
</tbody>
</table>
Nehalem's measured architectural Efficiency

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

<table>
<thead>
<tr>
<th>Full Triad</th>
<th>Measured Performance</th>
<th>Architectural Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data from L1-cache</td>
<td>1.45 Gflops</td>
<td>0.14</td>
</tr>
<tr>
<td>Data from L2-cache</td>
<td>1.3 Gflops</td>
<td>0.13</td>
</tr>
<tr>
<td>Data from L3-cache</td>
<td>0.2 – 0.95 Gflops</td>
<td>0.02 – 0.09</td>
</tr>
<tr>
<td>Data from memory</td>
<td>0.2 - 0.5 Gflops</td>
<td>0.02 – 0.05</td>
</tr>
</tbody>
</table>

Measured architectural efficiency for dot product: \( s = s + a \times b \) (a, b are vectors)

Measurement of library-routine (BLAS) in square brackets

<table>
<thead>
<tr>
<th>Dot Product</th>
<th>Measured Performance</th>
<th>Architectural Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data from L1-cache</td>
<td>2.2 [3.4] Gflops</td>
<td>0.22 [0.34]</td>
</tr>
<tr>
<td>Data from L2-cache</td>
<td>1.95 [2.9] Gflops</td>
<td>0.19 [0.29]</td>
</tr>
<tr>
<td>Data from L3-cache</td>
<td>0.5 - 1.6 [1.0 - 2.2] Gflops</td>
<td>0.05 – 0.16 [0.1 – 0.22]</td>
</tr>
<tr>
<td>Data from memory</td>
<td>0.5 - 1.15 [0.5 - 1.3] Gflops</td>
<td>0.05 – 0.11 [0.05 – 0.13]</td>
</tr>
</tbody>
</table>
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 \( C = A \times B \)

\[
c(i,j) = c(i,j) + a(i,k) \times b(k,j)
\]
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 | 1 | 2 | 8 | 25 | 6 | 5 | 1 | 2 | 0 | 4 | 8 |
|--------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|1| 2| 8| 25| 6| 5| 1| 2| 0| 4| 8 |
| Perf. loss \(vl=100\) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10 | 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 | 75 | 75 | 75 | 75 | 75 | 75 | 75 | 89 | 93 | 97 | 97 |
Memory Request with different 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)
  \[ c(i,j) = c(i,j) + a(i,k) \times b(k,j) \]
Using file systems at HC3

Roland Laifer
Basic Lustre concepts

Lustre components:
- 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

<table>
<thead>
<tr>
<th>File system</th>
<th>$HOME</th>
<th>$PFSWORK</th>
<th>$WORK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity (TiB)</td>
<td>76</td>
<td>301</td>
<td>203</td>
</tr>
<tr>
<td>Storage hardware</td>
<td>transtec provigo</td>
<td>transtec provigo</td>
<td>DDN S2A9900</td>
</tr>
<tr>
<td># of OSTs</td>
<td>12</td>
<td>48</td>
<td>28</td>
</tr>
<tr>
<td># of OST disks</td>
<td>192</td>
<td>768</td>
<td>290</td>
</tr>
</tbody>
</table>
## File system properties

<table>
<thead>
<tr>
<th>Property</th>
<th>$\text{TMP}$</th>
<th>$\text{HOME}$</th>
<th>$\text{WORK}$</th>
<th>$\text{PFSWORK}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visibility</td>
<td>local node</td>
<td>HC3, IC1</td>
<td>HC3</td>
<td>HC3, IC1</td>
</tr>
<tr>
<td>Lifetime</td>
<td>batch job</td>
<td>permanent</td>
<td>&gt; 7 days</td>
<td>&gt; 7 days</td>
</tr>
<tr>
<td>Capacity</td>
<td>thin/med.</td>
<td>fat</td>
<td>login</td>
<td>129</td>
</tr>
<tr>
<td>Quotas</td>
<td>no</td>
<td>not enforced</td>
<td>not enforced</td>
<td>not enforced</td>
</tr>
<tr>
<td>Backup</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Read perf. / node</td>
<td>thin/medium</td>
<td>fat</td>
<td>for IC1 nodes</td>
<td>70</td>
</tr>
<tr>
<td>Write perf. / node</td>
<td>thin/medium</td>
<td>fat</td>
<td>for IC1 nodes</td>
<td>80</td>
</tr>
<tr>
<td>Total read perf.</td>
<td>n*70</td>
<td>n*250 MB/s</td>
<td>1700 MB/s</td>
<td>4800 MB/s</td>
</tr>
<tr>
<td>Total write perf.</td>
<td>n*80</td>
<td>n*390 MB/s</td>
<td>1500 MB/s</td>
<td>4800 MB/s</td>
</tr>
</tbody>
</table>
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

- Stripe size (default 1 MB)
- Stripe count: 1 (default), 2 (default), 4 (default)
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

![Graph showing write performance with default stripe count](image)

- **$HOME**
- **$PFSWORK**
- **$WORK**

Throughput (MB/s) vs. Number of nodes
Read performance with default stripe count

Throughput (MB/s)

Number of nodes

$max.$

$HOME$

$PFSWORK$

$WORK$
Metadata performance

Operations / sec vs Number of nodes

- Create
- Stat
- Delete
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=tt'
    - 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)

- 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 Laboratories

<table>
<thead>
<tr>
<th>Plasma Physics</th>
<th>Biology</th>
<th>Molecular Systems</th>
<th>Earth &amp; Environment</th>
<th>Astro-Particle</th>
<th>Energy</th>
<th>NanoMikro</th>
</tr>
</thead>
</table>

Parallel Performance | Methods & Algorithms

Cross-Sectional Teams

Quantum Information | NIC Group | Distributed Computing

Research Groups

Education & Training Programmes

29.11.2010, SimLabs@KIT, Frank Schmitz
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)
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 Facilities → LSDF)
other SCC Services
Compute Services (HPC)
SimLabs as advanced support for Users at KIT

Scientific Computing Resources (HPC & DIC)

KIT Zentrum

KIT Kompetenzfelder und -bereiche

Scientific Community

29.11.2010, SimLabs@KIT, Frank Schmitz
Software Packages on HC3

Paul Weber
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)
Course of a project

- Geometry and Modelling
  - Meshing
  - Physical properties

- Numerical Solution

- Results: Visualization
  - Animation

Preprocessing

Analysis

Postprocessing

- Proprietary Modules
  - universal Preprocessors

- Solver Modules

- 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

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