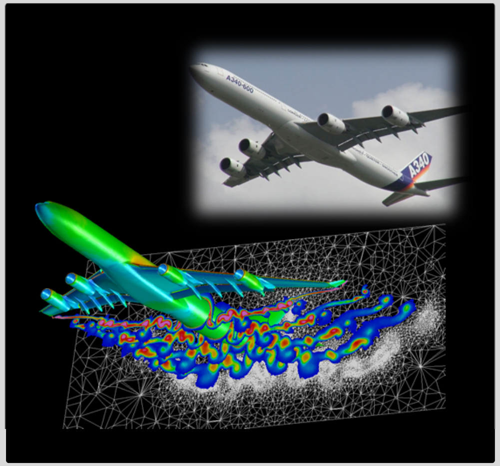


High Performance Computing

ADVANCED SCIENTIFIC COMPUTING

Dr. – Ing. Morris Riedel
Adjunct Associated Professor
School of Engineering and Natural Sciences, University of Iceland
Research Group Leader, Juelich Supercomputing Centre, Germany

SHORT LECTURE 12

Terrestrial Systems & Climate

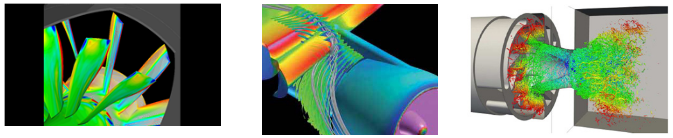
November 7th, 2017
Room TG-227

UNIVERSITY OF ICELAND
SCHOOL OF ENGINEERING AND NATURAL SCIENCES
FACULTY OF INDUSTRIAL ENGINEERING,
MECHANICAL ENGINEERING AND COMPUTER SCIENCE

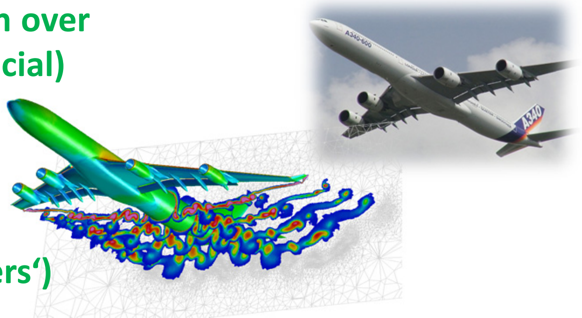
JÜLICH
FORSCHUNGSZENTRUM

Review of Lecture 11 – Scientific Visualization & Steering

- Many Visualization Methods



(visualization over time is crucial)



('a picture is worth 1000 ~~words~~ numbers')

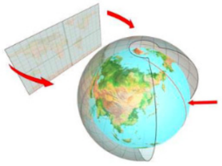
BLENDER: HD visualization

PARAVIEW: GUI for advanced viz

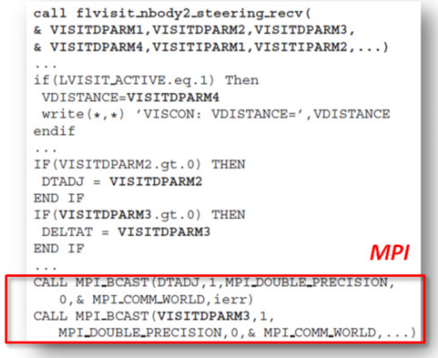
Qt: customized GUI for scientific viz

VTK: advanced scientific viz (low level)

PYTHON: programming language and basic sci viz



- Interactively Steer HPC Applications while executing

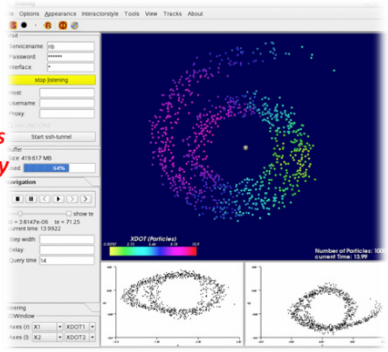


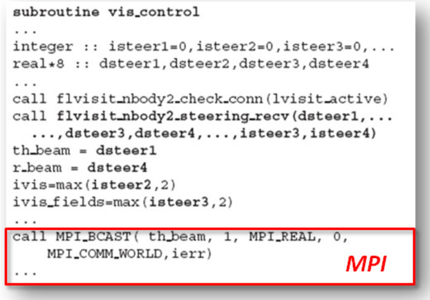
```
call flvisit_nbody2_steering_recv(
& VISITDPARM1,VISITDPARM2,VISITDPARM3,
& VISITDPARM4,VISITIPARM1,VISITIPARM2,...)
...
if(LVISIT_ACTIVE.eq.1) Then
 VDISTANCE=VISITDPARM4
 write(*,*) 'VISCON: VDISTANCE=',VDISTANCE
endif
...
IF(VISITDPARM2.gt.0) THEN
 DTADJ = VISITDPARM2
END IF
IF(VISITDPARM3.gt.0) THEN
 DELTAT = VISITDPARM3
END IF
...
CALL MPI_BCAST(DTADJ,1,MPI_DOUBLE_PRECISION,
   0,& MPI_COMM_WORLD,ierr)
CALL MPI_BCAST(VISITDPARM3,1,
   MPI_DOUBLE_PRECISION,0,& MPI_COMM_WORLD,...)
```

MPI

change parameters interactively

visualize status



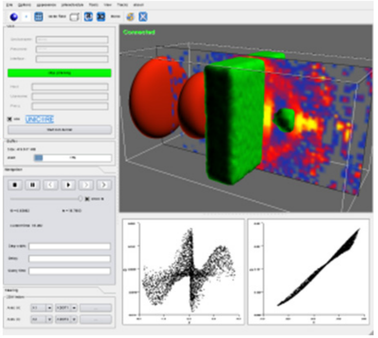


```
subroutine vis_control
...
integer :: isteer1=0,isteer2=0,isteer3=0,...
real*8 :: dsteer1,dsteer2,dsteer3,dsteer4
...
call flvisit_nbody2_check_conn(lvisit_active)
call flvisit_nbody2_steering_recv(dsteer1,...
  ...,dsteer3,dsteer4,...,isteer3,isteer4)
th_beam = dsteer1
r_beam = dsteer4
ivis=max(isteer2,2)
ivis_fields=max(isteer3,2)
...
call MPI_BCAST( th_beam, 1, MPI_REAL, 0,
    MPI_COMM_WORLD,ierr)
...
```

MPI

change parameters interactively

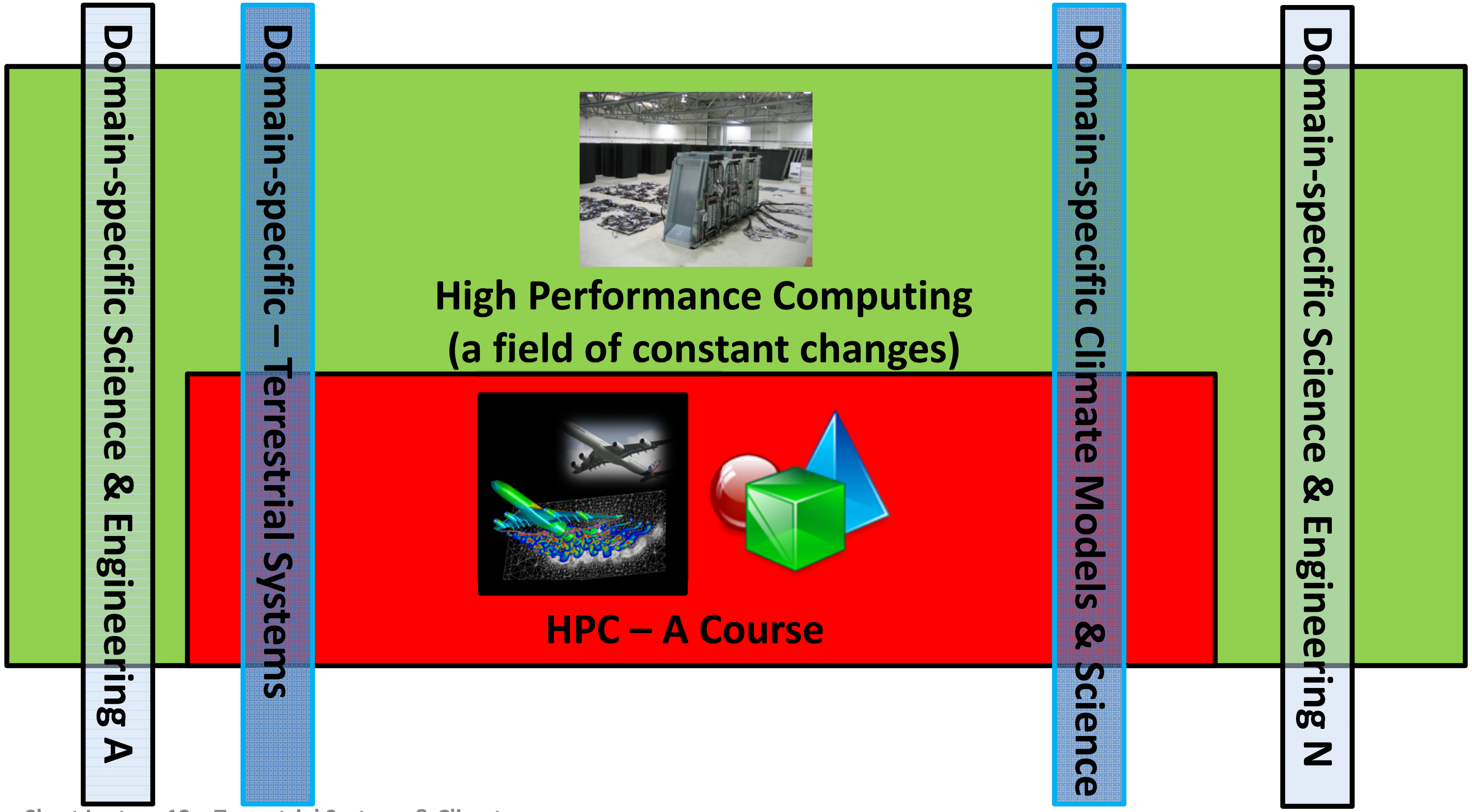
visualize status



modified from [1] CINECA – Scientific Visualization Training *[2] M. Riedel et al., computational steering, 2007*

HPC-A[dvanced] Scientific Computing – Second Part

- Consists of techniques for programming large-scale HPC Systems
  - Approach: Get a broad understanding what HPC is and what can be done
  - Goal: Train general HPC techniques and bits of domain-specific applications



Domain-specific Science & Engineering A

Domain-specific – Terrestrial Systems

High Performance Computing
(a field of constant changes)

HPC – A Course

Domain-specific Climate Models & Science

Domain-specific Science & Engineering N

Outline of the Course

1. High Performance Computing
2. Parallelization Fundamentals
3. Parallel Programming with MPI
4. Advanced MPI Techniques
5. Parallel Algorithms & Data Structures
6. Parallel Programming with OpenMP
7. Hybrid Programming & Patterns
8. Debugging & Profiling Techniques
9. Performance Optimization & Tools
10. Scalable HPC Infrastructures & GPUs
11. Scientific Visualization & Steering
12. Terrestrial Systems & Climate
13. Systems Biology & Bioinformatics
14. Molecular Systems & Libraries
15. Computational Fluid Dynamics
16. Finite Elements Method
17. Machine Learning & Data Mining
18. Epilogue

+ additional practical lectures for our hands-on exercises in context

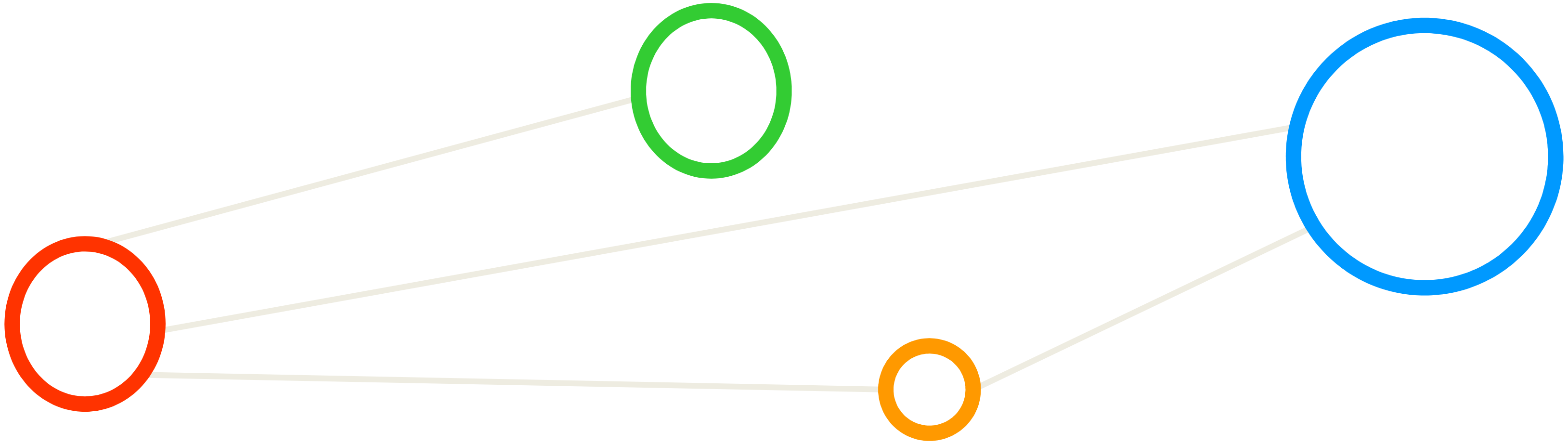
Outline

- Terrestrial Systems
  - Numerical Simulations
  - ParFlow Hydrology Model
  - CLM Land-Surface Model
  - COSMO Weather Model
  - Coupling Different Models
- Climate
  - Numerical Weather Prediction
  - Partial Differential Equations
  - WRF Model Application
  - WRF Parallel I/O & pNetCDF

> ➢ **Short Lecture only shows subsets of libraries & applications in domains**

- **Promises from previous lecture(s):**
- ***Lecture 2:* Lecture 12 about Terrestrial Systems will provide more details on domain decomposition aspects**
- ***Lecture 2:* Lectures 12-17 will provide details on applied parallelization methods within parallel applications**
- ***Lecture 4:* Lectures 12-17 will provide parallel application examples that take advantage of the NetCDF**
- ***Lecture 5:* Lecture 12 will give in-depth details on parallel algorithms used in terrestrial systems and climate**
- ***Lecture 5:* Lectures 12-17 will provide more details on how NETCDF and HDF is used with application data**
- ***Lecture 5:* Lectures 12-17 will provide more details on complexities integrating technical & domain code**
- ***Lecture 7:* Lecture 12 will provide more HPC examples from terrestrial systems and climate simulations**
- ***Lecture 11:* Lecture 12-17 will provide scientific visualizations for different HPC application domain fields**

Terrestrial Systems

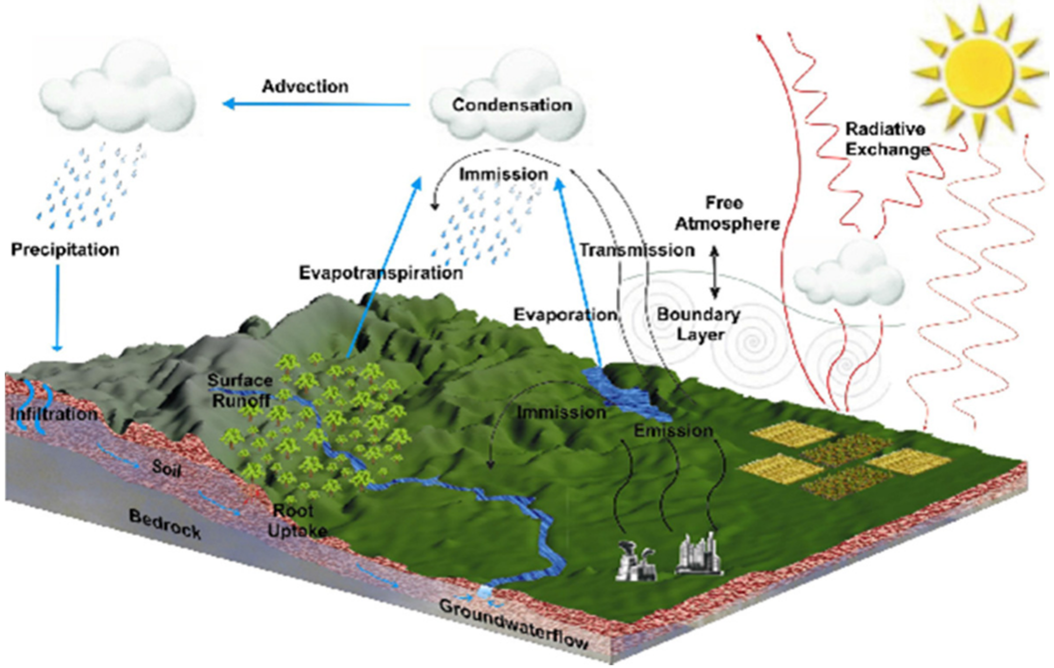


Terrestrial Systems – Motivation

- **Terrestrial systems represent a class of applications that perform numerical HPC simulations of variable complexity of terrestrial systems processes across different scales & regions**

modified from [3] SimLab Terrestrial Systems

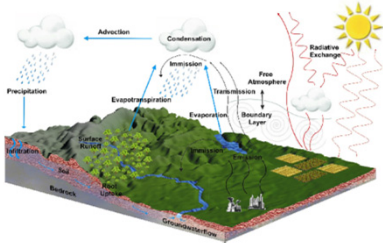
Physical system changes are accompanied by major state changes of land surfaces & ecosystems



State changes of land surfaces & ecosystems and services provided by them have multiple socioeconomic impacts

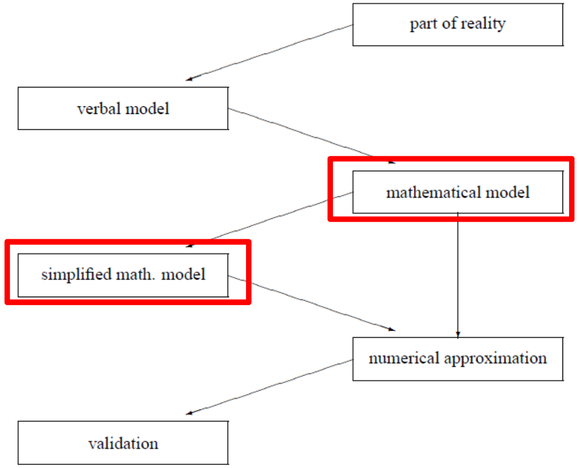
- Selected Motivations
  - Understand global environmental change (e.g. climate) affecting terrestrial systems at all scales
  - Increase understanding of many physical processes on earth (latin terra) that are still poorly understood
  - Work towards better reproducibility of models

Terrestrial Systems – Modelling Dynamical Systems



[3] SimLab Terrestrial Systems

- Evolution in time (and space) is of interest
  - Behavior of a whole ecosystem in time
  - Dynamical systems:
    e.g. economic processes, movement of a fluid, …
  - (cf. Lecture 7 simple Jacobi example & heat equation)
- How to model 'evolution' of a system
  - A dynamical system consists of its state (e.g. input data) and a some 'rules'
  - Rules determine how the dynamic system will evolve over time
  - Rules governing the evolution are 'physical laws/equations for different system elements'


part of reality
verbal model
mathematical model
simplified math. model
numerical approximation
validation

[4] Introduction to SC

- **In order to investigate a real system's behaviour by computing, a mathematical model is needed**
- **A dynamical system is some realistic system whose evolution in time is of interest**

Terrestrial Systems – Need for Numerical Methods in HPC

- **Solving some mathematical problems & equations is too computational intensive → approximate**
- **Numerical methods are methods that obtain numerical approximation solutions to problems**

[4] Introduction to SC

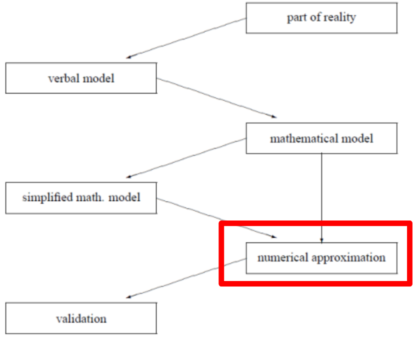
- Behaviour ‘governed by equations‘ are computed
  - Nature is complex & interconnected: simplification
- Behaviour governed by ‘difference equations‘
  - System state only change at discrete instants of time
  - System state ‘not change in time continously‘

(solutions can be computed simply by applying definitions iteratively)

- Behaviour governed by ‘differential equations‘
  - System state evolves ‘continously in time‘

(harder to solve, e.g. initial value problem)

- Selected ‘scientific questions‘ for simulations
  - Under what circumstances will a system evolve into an ‘equilibrium–state’ (state which does not change)
  - Under what circumstances will the system evolve into a ’periodic state’ (states the system return to over time)



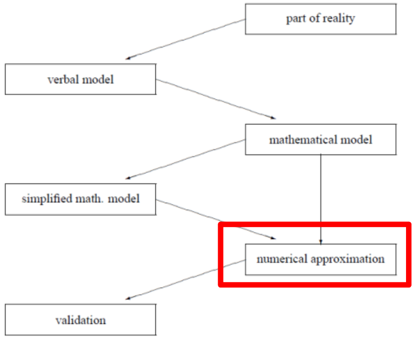
Terrestrial Systems – Partial Differential Equations (PDEs)

- **HPC models often use toolkits (e.g. PETSc) for Partial Differential Equations (PDEs) that are differential equations that contains unknown multivariable functions and their partial derivatives**
- **A general method in HPC modelling use parallel PDEs tools to approximate solutions to problems**

modified from [5] Wikipedia on 'Partial Differential Equation'

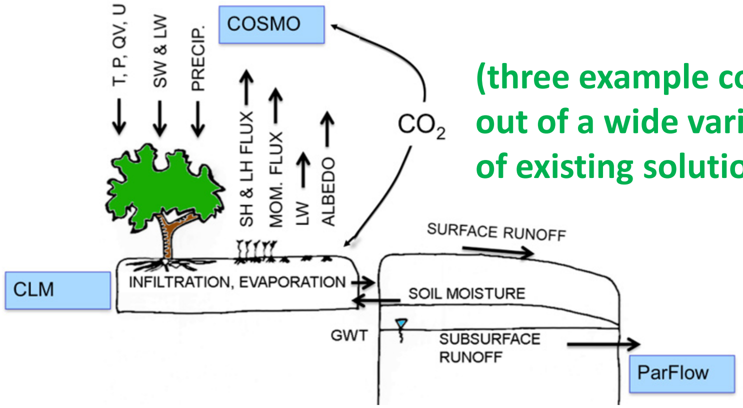
- HPC simulation modelling
  - PDEs enable rates of change (of continous variables)
  - PDEs used to formulate problems involving functions of several variables
  - PDEs describe a wide variety of phenomena (e.g. sound, heat, electrostatics, fluid flow, etc.)
  - PDEs model multi-dimensional dynamical systems
- Differences to 'ordinary differential equations'
  - Ordinary differential equations deal with functions of a single variable and their derivatives
  - Ordinary differential equations model one-dimensional dynamical system

Solving those equations is often too complicated computationally expensive or impossible to analytically compute driving the need for numerical approximation



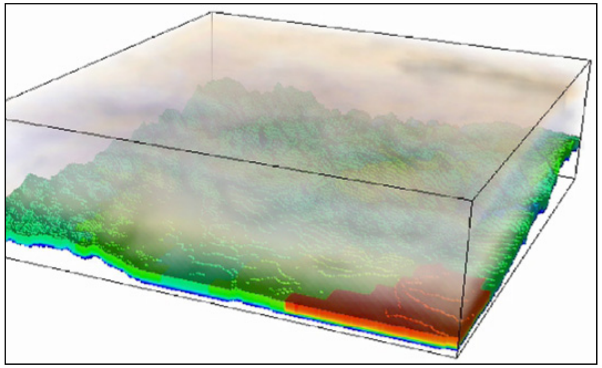
[4] Introduction to SC

Terrestrial Systems – Numerial HPC Simulations



(three example codes out of a wide variety of existing solutions)

[3] SimLab Terrestrial Systems

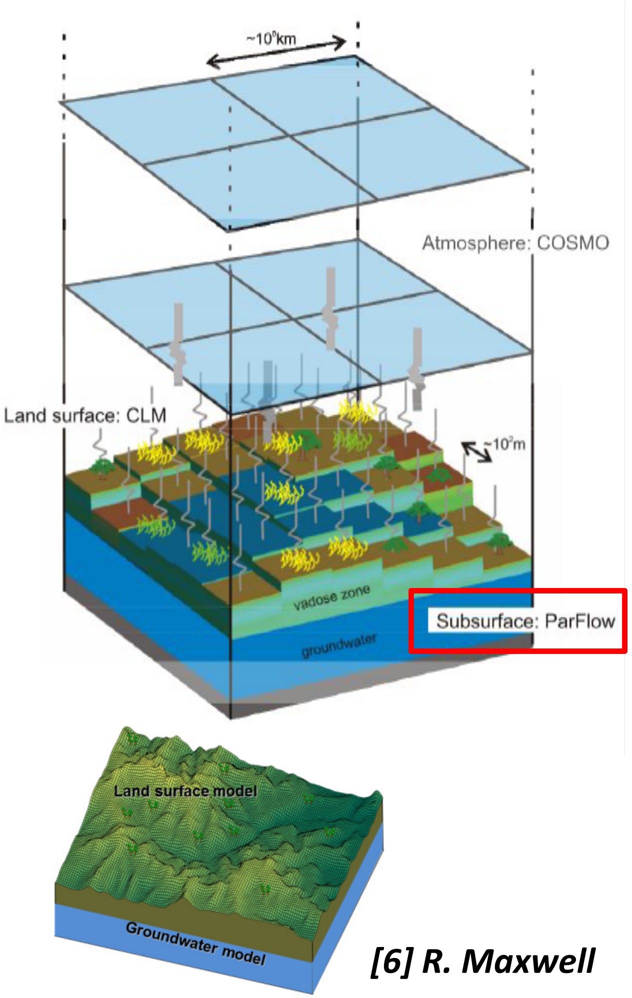


- Numerical models & simulations
  - Investigate multiple terrestrial system processes as a whole
  - Enable sustainable management of terrestrial systems
  - Simulate complex, non-linear transport processes of energy, mass and momentum
  - Create interactions and feedback mechanisms between different compartments of the coupled geo-ecosystem (e.g. subsurface, land-surface, atmosphere, reservoirs, etc.)
  - Varying scales: multiple spatio-temporal scales and high resolutions
  - Potentially long runtimes & use 'ensemble simulations'

➢ **Lecture 13 provides more details on ensemble methods to estimate uncertainties in bio-systems**

Terrestrial Systems – ParFlow Model Parallel Application (1)

- Modelling ‘hydrology‘ processes
  - Parallel watershed flow model (ParFlow)
  - Simulate surface and subsurface fluid flow
  - Use in the assessment and management of groundwater and surface water
  - Investigate system physics and feedbacks
  - Understand interactions at a range of scales
  - Suitable for large scale & high resolution
- Parallel ‘numerical‘ application
  - Developed over 10 years (aka stable code)
  - Offers advanced numerical solvers for massively parallel HPC systems



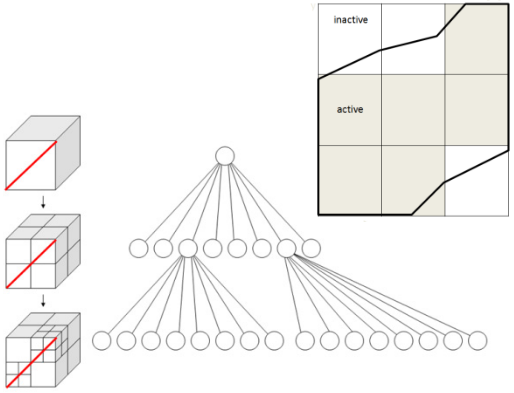
[6] R. Maxwell

[7] ParFlow Web page

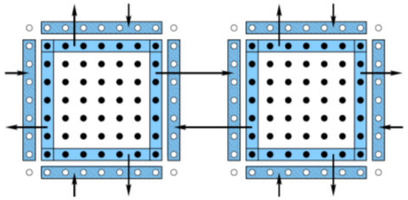
- **ParFlow enables the parallel simulation of hydrology processes with (sub-)surface fluid flows**

Terrestrial Systems – ParFlow Model Parallel Application (2)

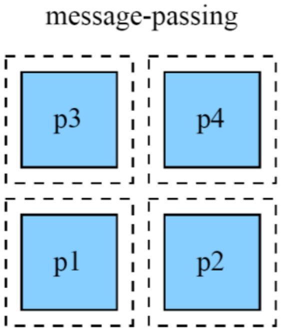
- Parallelization Techniques
    - 3D Grid domain decomposition (cf. Lecture 2)
    - 3D code (use octree-space partitioning algorithm)
    - Implements hybrid programming (cf. Lecture 7)
    - Requirement of ‘halo regions‘ for numerical equations (cf. Lecture 7)

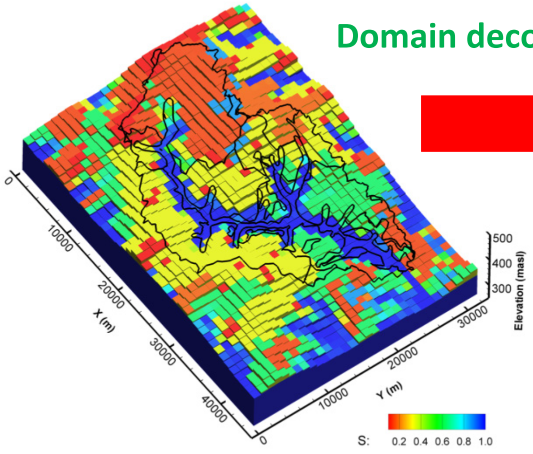


[6] HPC & ParFlow



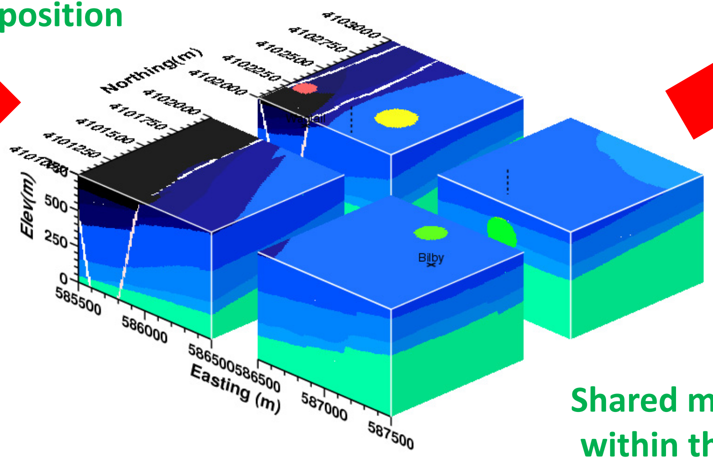
Distributed memory across the grid & halo updates

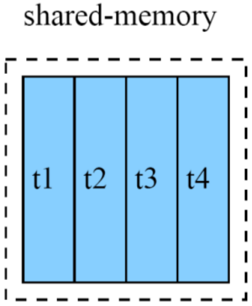




[7] ParFlow Web page

Domain decomposition

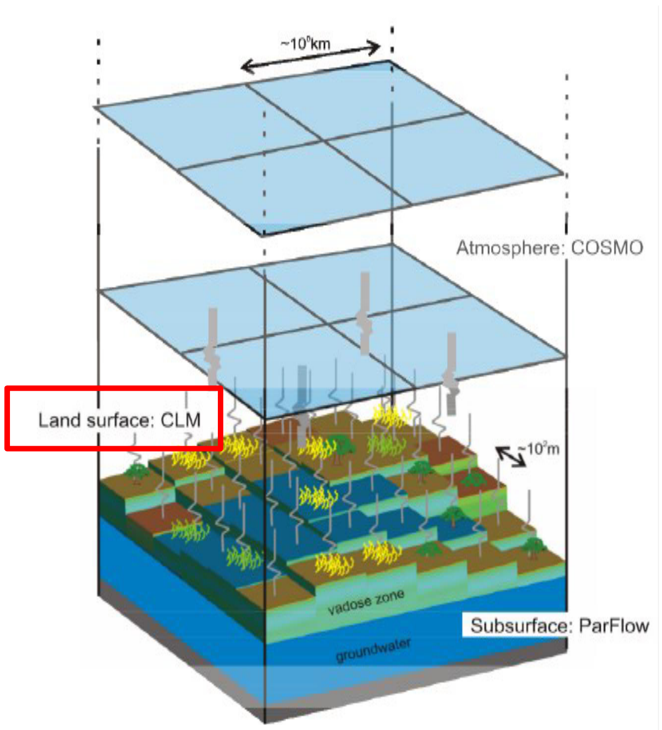


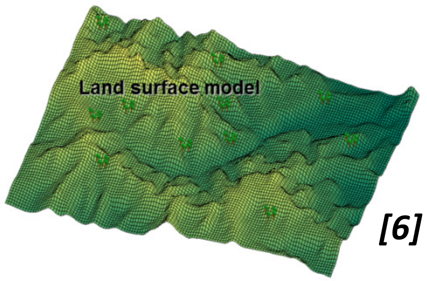


Shared memory within the grid

Terrestrial Systems – CLM Model Parallel Application (1)

- Modelling 'land surface' processes
  - Community land model (CLM)
  - Simulates concepts of ecological climatology
  - Understand how natural & human changes in vegetation affect the climate
  - Examine physical, chemical, and biological processes that affect (or are affected by climate across spatial / temporal scales
  - Investigate terrestrial ecosystems through their cycling of energy, water, chemical elements, and trace gases
  - Explore impact of terrestrial ecosystems as important determinants of climate





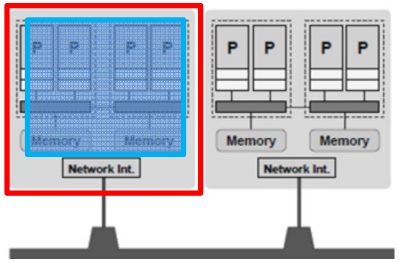
[8] CLM Web page

[6] R. Maxwell

- **CLM enables the parallel simulation of land-surface with physical/chemical/biological processes**

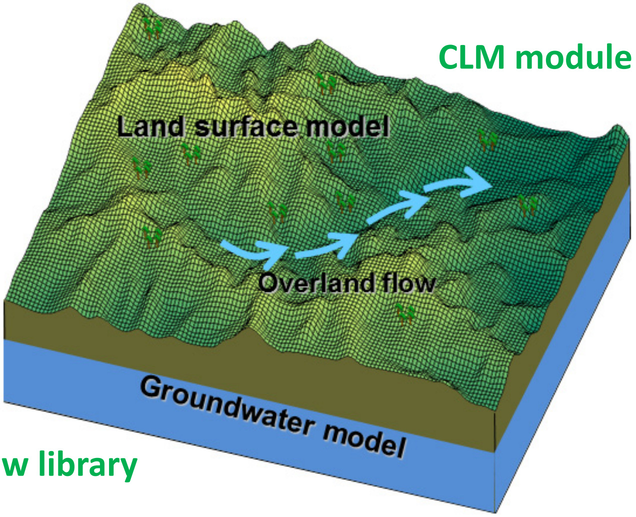
Terrestrial Systems – CLM Model Parallel Application (2)

- Parallelization Techniques
  - Implements ‘hybrid programming‘ (cf. Lecture 7)
  - OpenMP within a node (cf. Lecture 6)
  - MPI routines for parallelism across nodes (cf. Lecture 3)



[6] R. Maxwell

- Coupled as module
  - Code is often fully coupled with ParFlow
  - Coupling is performed in a way that CLM is incorporated into ParFlow as a module (full coupled, fully parallel)
  - E.g. flow of water on land-surface affects groundwater model



CLM module

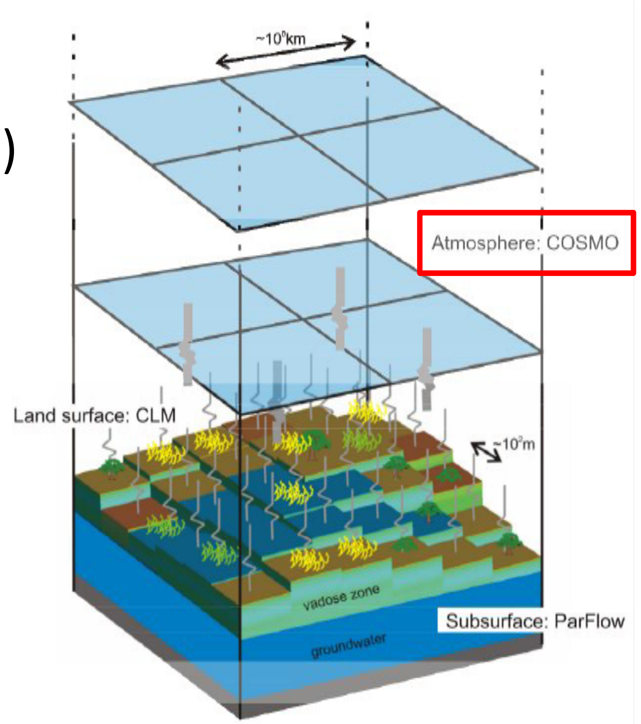
ParFlow library

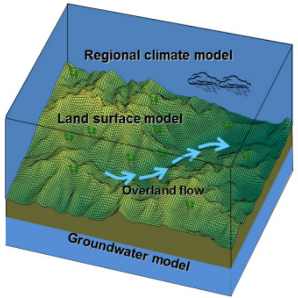
- **Coupling can be done using modules where one library is a module to another in one executable**

Terrestrial Systems – COSMO Model Parallel Application (1)

- Modelling ‘atmosphere‘ processes
    - Consortium for Small-scale Modeling (COSMO)
    - Athmospheric model based on Fortran 90 & numerical methods to simulate weather
    - Use for non-hydrostatic limited-area (e.g. regional atmospheric model)
    - Use a variety of physical processes by parameterization schemes
    - Perform ‘Data assimilation‘ from various observations (simulation alone not sufficient)
    - E.g. radiosonde (wind, temperature, humidity)
    - E.g. aircraft (wind, temperature)
    - E.g. wind profiler (wind)

[9] COSMO Web page

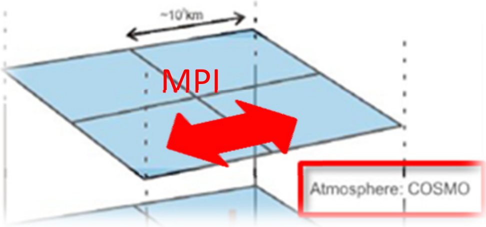




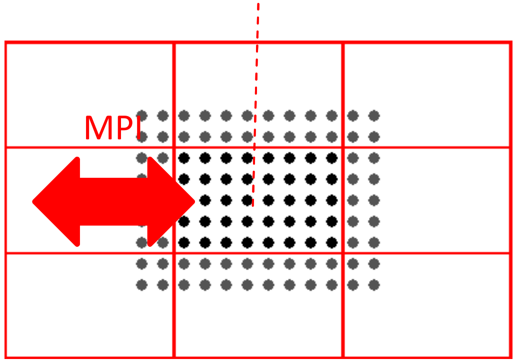
[6] R. Maxwell

- **COSMO enables the parallel simulation of detailed regional atmospheric model processes**

Terrestrial Systems – COSMO Model Parallel Application (2)



- Parallelization Techniques
  - Distributed memory using MPI (cf. Lecture 3)
  - Two dimensional domain decomposition (cf. Lecture 2)
  - Each processor gets a part of data (one subdomain)
  - Solves model equations in parallel
- Use of 'halo technique' (cf. Lecture 7)
  - 2 halo gridlines that belong to neighbor (configurable property)
  - Different computing/solver schemes require different halo lines (2 or 3)
  - Grid points belonging to the halo are exchanged using MPI

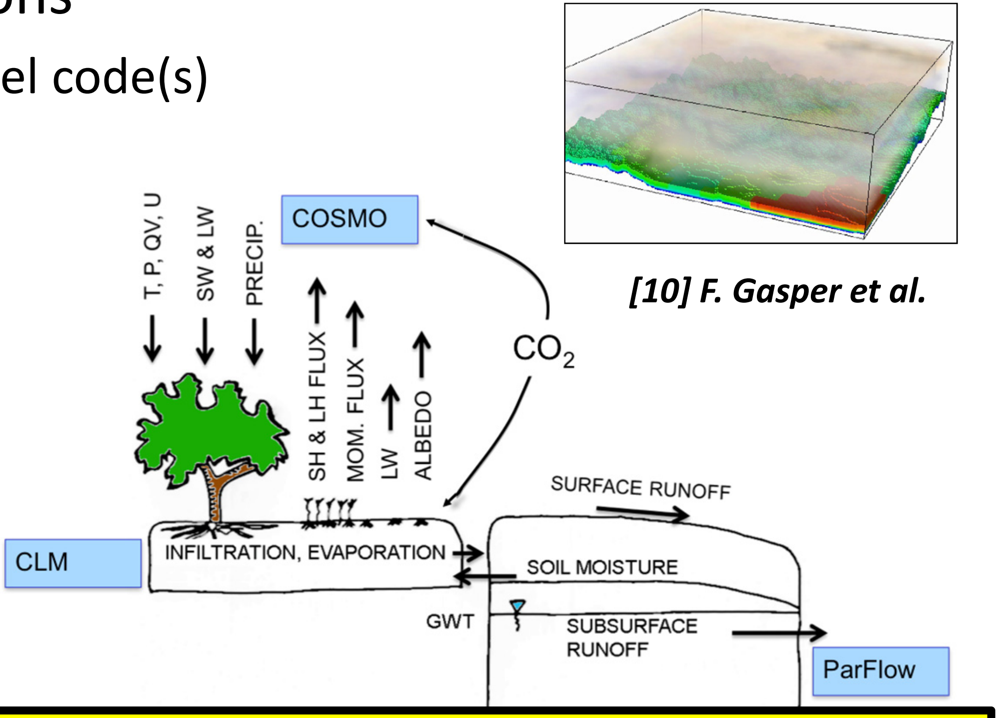


during integration step each processor updates the values of its local subdomain

[9] COSMO Web page

Terrestrial Systems – Coupling Different Parallel Libraries (1)

- Scientific case: understanding water cycle processes and variability across scales including climate and land use impacts
  - Computational challenge: earth system models at regional scale
- Terrestrial systems simulations
  - E.g. TerrSysMP coupled parallel code(s)
  - Many processes are still poorly reproduced by models
  - Overcome isolated research and simulation for specific elements in earth systems
  - Full integrated groundwater-vegetation-atmosphere simulation algorithms

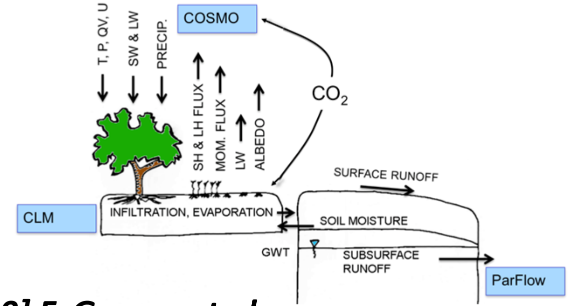


[10] F. Gasper et al.

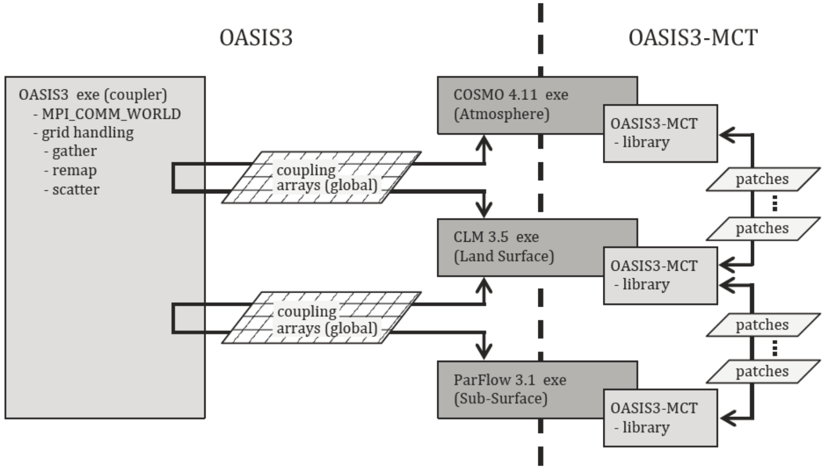
- **Coupled codes execute n different parallel application codes together to simulate one ecosystem**

Terrestrial Systems – Coupling Different Parallel Libraries (2)

- Requires a coupling technique running on a HPC machine
  - Example: OASIS3 coupler for ParFlow, CLM & COSMO
  - 1+3 parallel applications together referred to as ‘TerrSysMP parallel coupled application‘
  - OASIS3 is a separate executable that manages data exchange between others
  - Coupling data arrays are repartitioned to the full domain by OASIS
  - OASIS3-MCT library is part of each component model
  - Coupling arrays only consist of the local fraction of full domain
  - Routed by OASIS to the destination processor



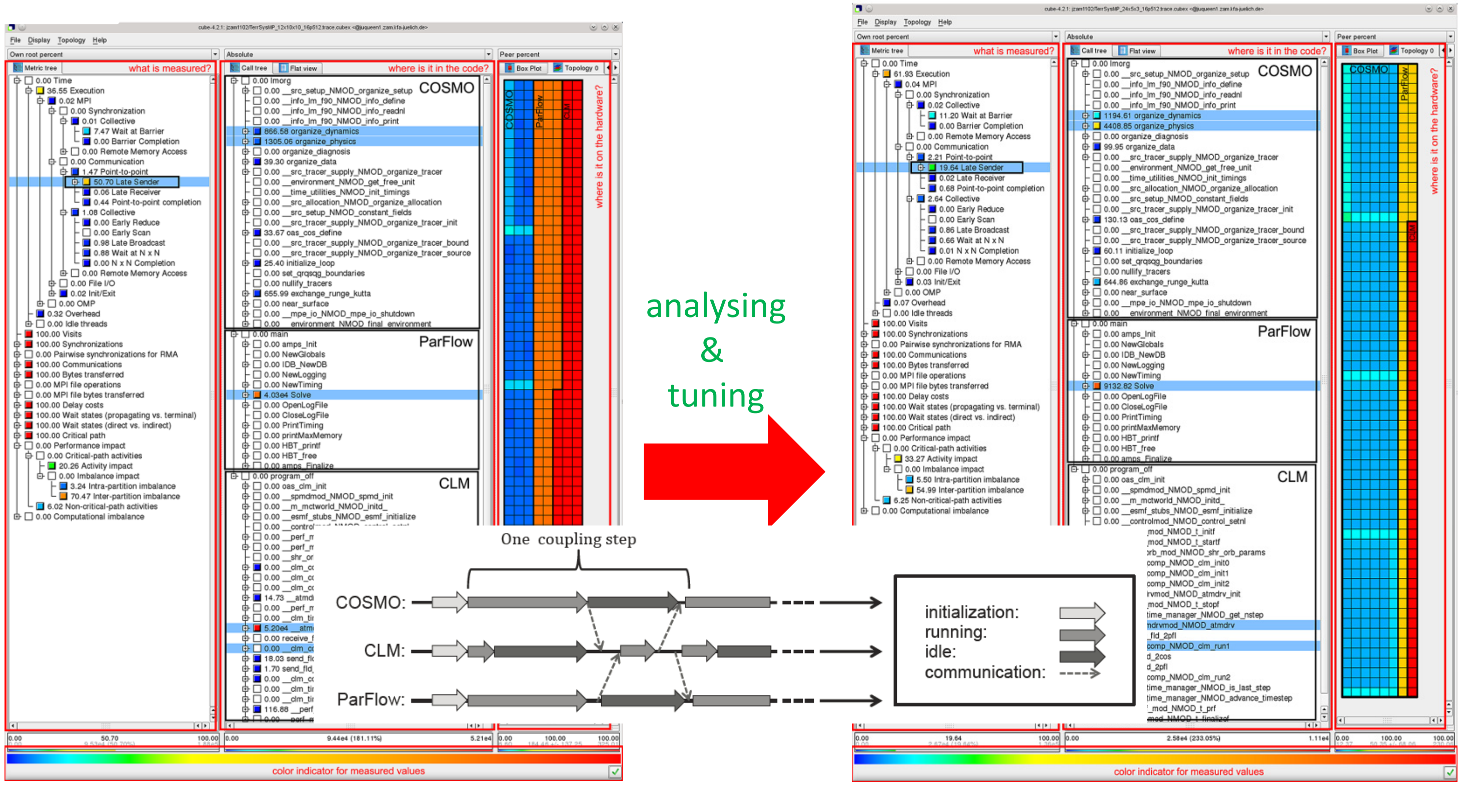
[10] F. Gasper et al.



- **Coupled codes require another separate executable that is a coupler exchanging global data**

Terrestrial Systems – Coupling Different Parallel Libraries (3)

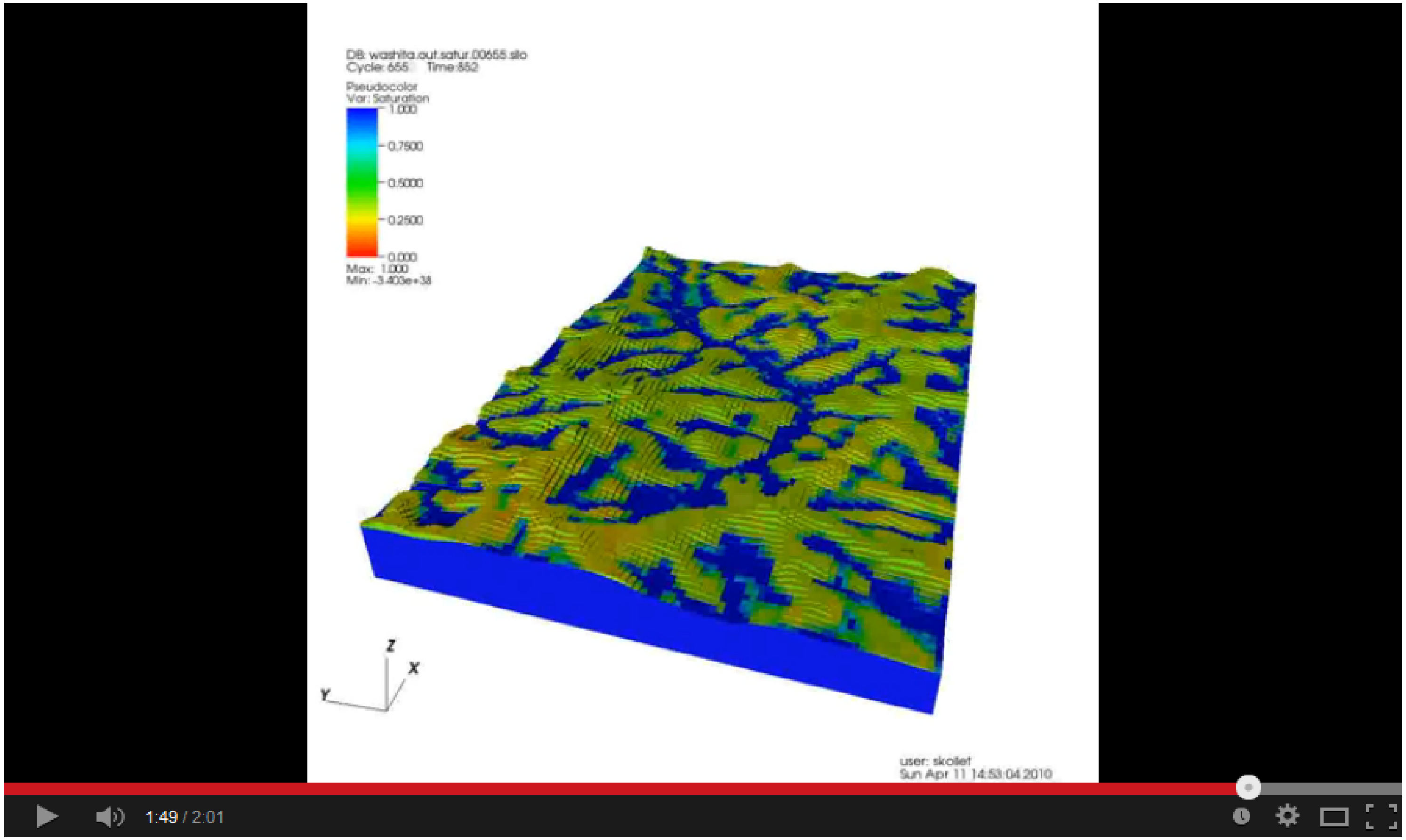
- Performance optimization required (cf. Lecture 9) ***[10] F. Gasper et al.***
  - Using performance analysis tool SCALASCA
  - Resources are distributed according to load (better load balance)
  - LateSender wait state is significantly reduced



Before:
Processor distribution
192 COSMO
160 ParFlow
160 CLM

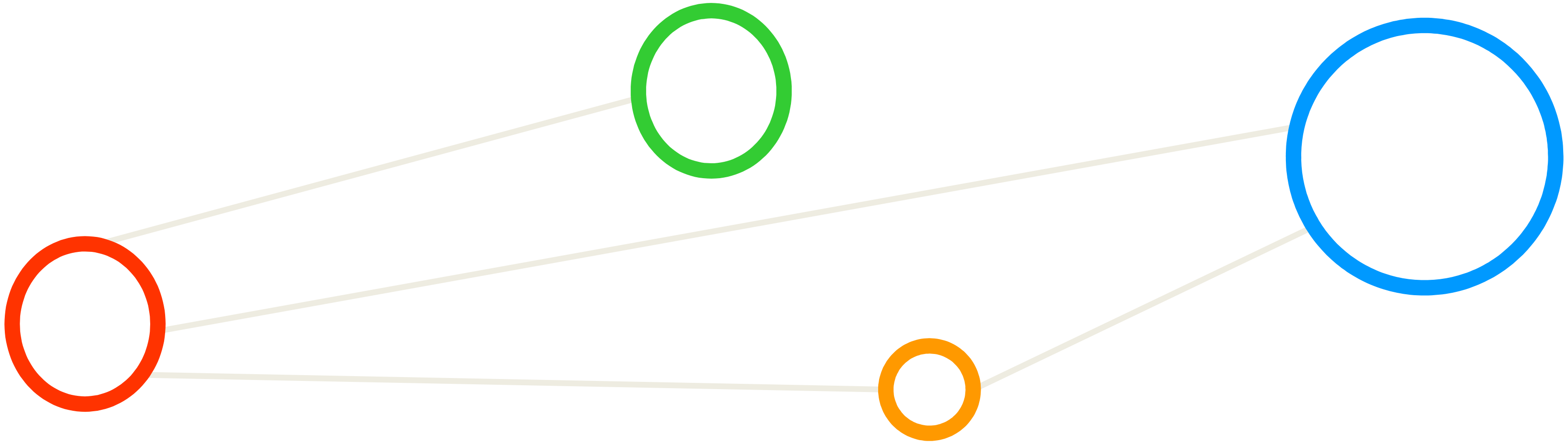
After:
Processor distribution
384 COSMO
80 ParFlow
48 CLM

[Video] Terrestrial Systems with ParFlow coupled with CLM



[11] ParFlow coupled with CLM

Climate

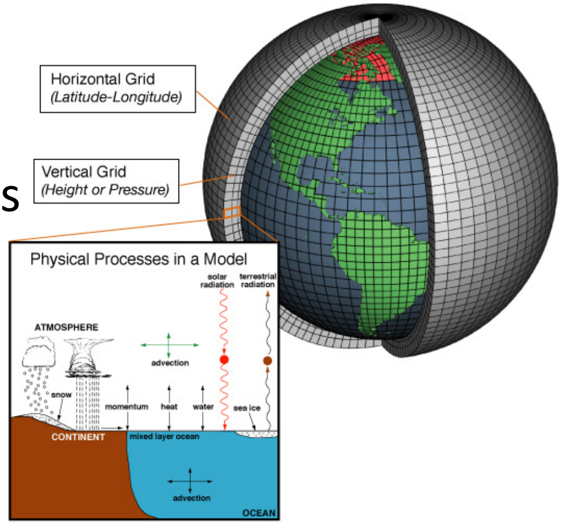


Climate – Numerical Weather Prediction & Forecast

- **Numerical Weather Prediction (NWP) uses mathematical models of the atmosphere and oceans to predict the weather based on current weather observations (e.g. weather satellites) as inputs.**
- **Performing complex calculations necessary for NWP requires supercomputers (limit ~6 days)**
- **NWP belongs to the field of numerical methods that obtain approximate solutions to problems**

modified from [12] Wikipedia on 'Numerical Weather Prediction'

- Application areas
  - Global and regional short-term weather forecast models in operations
  - Perform long-term climate prediction research (e.g. climate change)
- NWP model characteristics
  - Use ordinary/partial differential equations (PDEs) (i.e. use laws of physics, fluids, motion, chemistry)
  - Domain decomposition (cf. Lecture 2): 3D grid cells
  - Computing/cell: winds, heat transfer, solar radiation, relative humidity & surface hydrology
  - Interactions with neighboring cells: used to calculate atmosopheric properties over time



Climate – PDEs in Atmospheric Research (1)

- HPC Atmospheric Models
  - Simulations produce meterological information for given locations
  - Different 'temporal resolutions/scales': future times, days to decades
  - Different 'spatial solutions/scales': meters to kilometers
  - Use primitive equations to enable model evolution over space and time

- **Primitive equations are a set of nonlinear differential equations that are used to approximate global atmospheric flow in atmospheric models and predict/simulate future states of atmospheres**

modified from [13] Wikipedia on 'Primitive Equations'

- Set of Primitive Equations
  1. Conservation of momentum: Describe hydrodynamical flow on the surface of a sphere (e.g. vertical motion smaller than horizontal motion)
  2. Thermal energy equation: Overall temperature of the modelled system in relation to heat sources and sinks
  3. Continuity equation: Describe the conversation of mass

Climate – PDEs in Atmospheric Research (2)

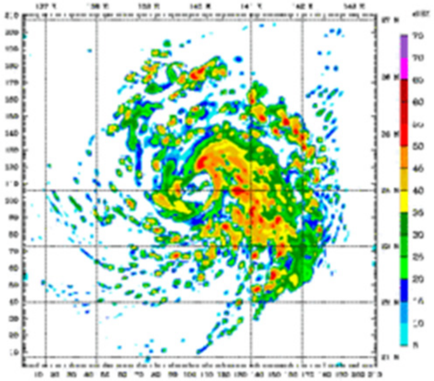
- HPC Model evolution over space and time
  - Based on primitive equations (alongside e.g. gas laws)
  - Simulations shange of density, pressure, potential temperature scalar fields, air velocity (aka wind) vector fields of the atmosphere over time
- Computational challenges
  - Nonlinear PDEs are impossible to solve exactly through analytical methods
  - Idea is to obtain approximate solutions with numerical methods
- Simulation over time
  - Initialization of equations based on analysis data or research question
  - Rates of changes determined via a time increment known as ‘time step‘
  - Approach is repeated until solution reaches the desired forecast time

- **Simulations over time need to maintain ‘numerical stability‘: the length of the time step chosen within the model is related to the distance between the points on the computational grid**

modified from [12] Wikipedia on ‘Numerical Weather Prediction’

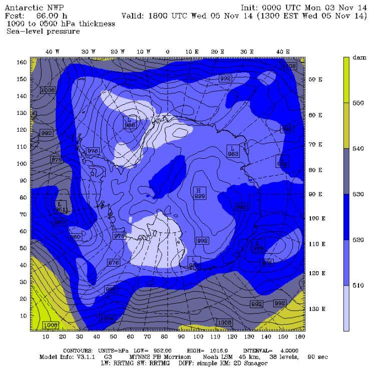
Climate – WRF Model Parallel Application – Examples

[14] Wikipedia on 'WRF'



WRF model output showing simulated radar reflectivity (rBZ) for Typhon Mawar (3.3km grid)

[15] Polar WRF

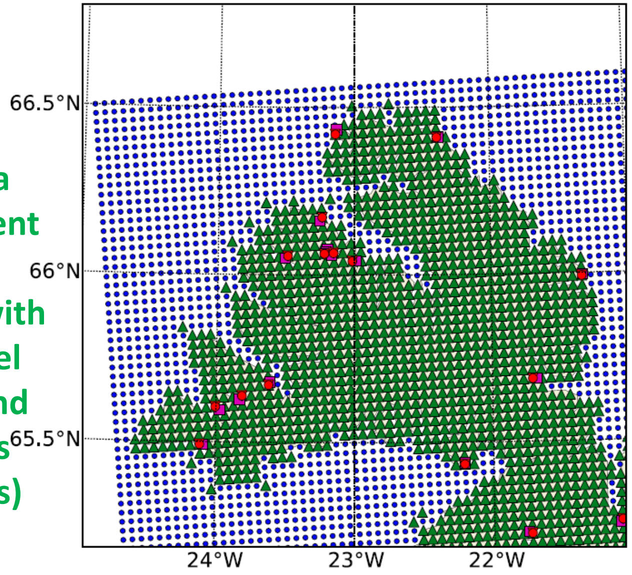


WRF polar model output showing 1000 – 500 hPa thickness & sea level pressure

[16] Evaluation of WRF Mesoscale model

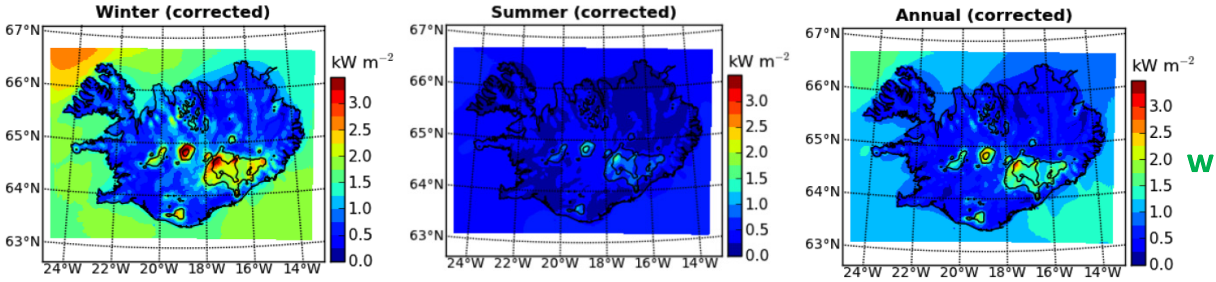
Veðurstofa
Íslands

Align data measurement stations in Iceland with WRF model (closest land gridpoints as red dots)



Veðurstofa
Íslands

[17] Iceland wind energy potential study, 2013



WRF model output showing wind power density across iceland at 50 mAGL

- **Software package Weather Research and Forecasting (WRF) includes parallelization techniques and enables a wide range of meteorological applications across scales (meters – 1000 of KMs)**

Climate – WRF Model Parallel Application – Software

- **The WRF model is a NWP system that enables the simulation and prediction of the atmosphere**
- **It is a scalable parallel HPC simulation for distributed-memory & shared-memory systems**

[18] WRF model Webpage

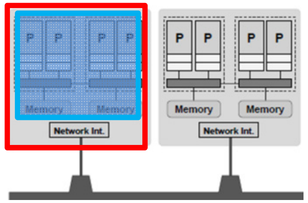
- Weather Research and Forecasting (WRF) model
  - Takes advantage of PDEs (and parallel solvers)
  - Maintained and support as a community model
  - Plug-compatible modules for extensions
  - Research advances have direct path to operations
  - Numerous physics options (link with the broader HPC modeling community) **(e.g. air quality modeling)**
- Selected software package features
  - Available as open-source tool implementing parallelization techniques
  - Implements a modular & hierarchical design
  - Supports a model coupling infrastructure & NetCDF data format support
  - Enables integration into bigger earth system model frameworks

code used for daily weather forecasts by MetOffices worldwide as service to tax payers

tax payers pay some scientists to better the WRF model → ROI / impact over years

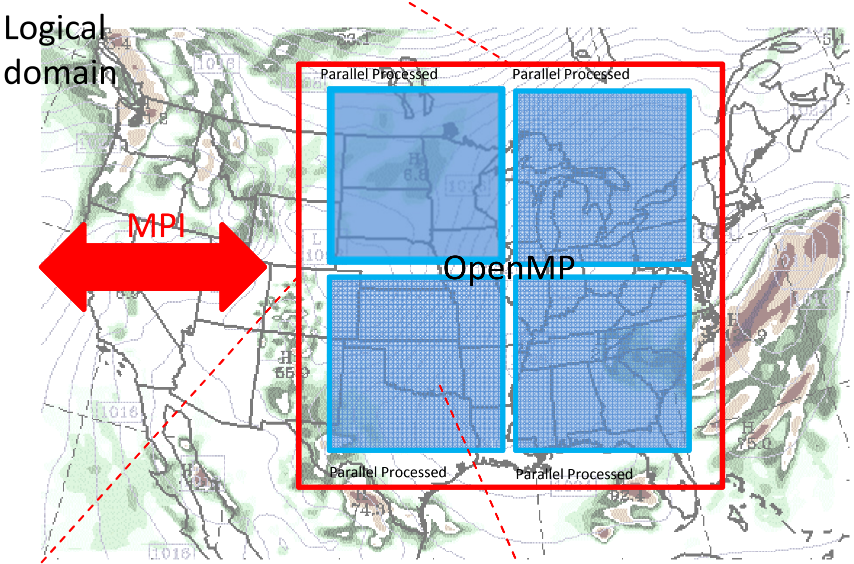
Climate – WRF Model Parallel Application – Parallelization (1)

- Parallel simulation sciences
  - E.g. reflecting real data obtained from observations, analyses, etc.
  - E.g. enable idealized atmospheric conditions
- Approach
  - Implements 'hybrid programming' (cf. Lecture 7)
  - Use of 'domain decomposition' (cf. Lecture 2) dividing work
  - Model domains are decomposed for parallelism on two-levels using 'patches' and 'tiles'



(patch: section of model domain allocated to a distributed memory node)

(mediation layer solver or physics driver)



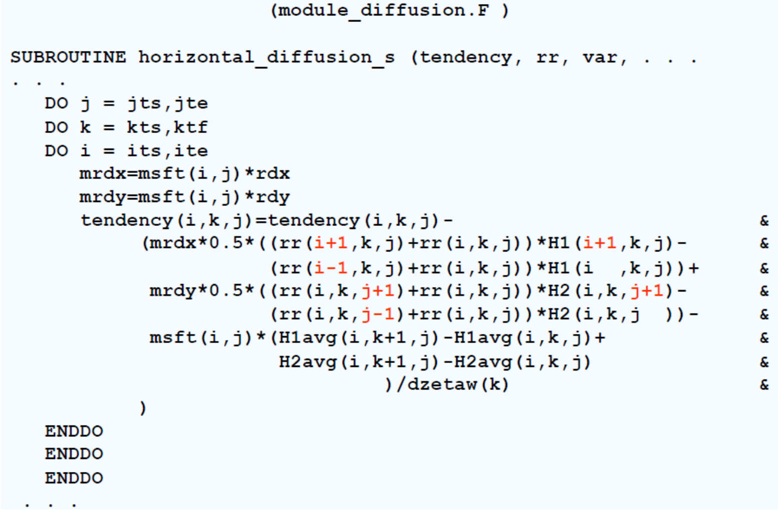
(one patch is divided into multiple tiles)

(tile: section of a patch allocated to a shared-memory processor within a node)

modified from [19] WRF – Code and Parallel Computing

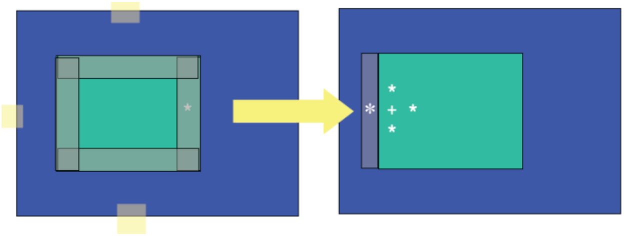
Climate – WRF Model Parallel Application – Parallelization (2)

- Usage for ‘halo‘ (cf. Lecture 7)
  - Code example based on Fortran
  - Horizontal data dependencies
  - E.g. i+1, i+1, etc.: indexed operands may lie in the patch of a neighboring processor
  - Problem: neighbor‘s updates to such an element of the array is not accessible on this processor

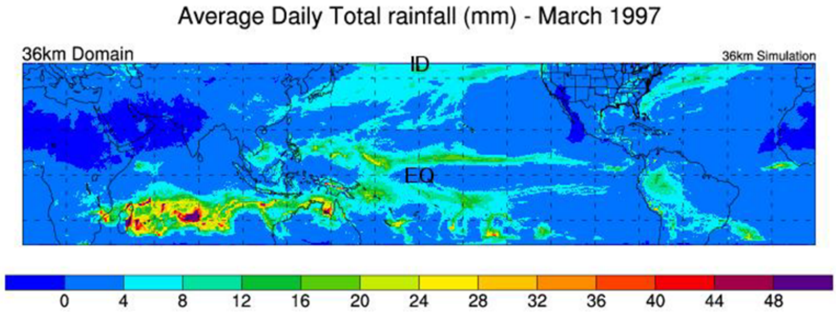


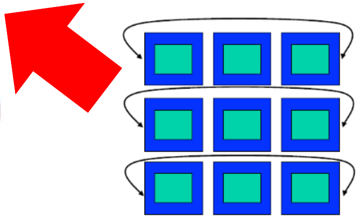
```
                        (module_diffusion.F )

SUBROUTINE horizontal_diffusion_s (tendency, rr, var, . . .
. . .
   DO j = jts,jte
   DO k = kts,ktf
   DO i = its,ite
      mrdx=msft(i,j)*rdx
      mrdy=msft(i,j)*rdy
      tendency(i,k,j)=tendency(i,k,j)-                                  &
           (mrdx*0.5*((rr(i+1,k,j)+rr(i,k,j))*H1(i+1,k,j)-            &
                      (rr(i-1,k,j)+rr(i,k,j))*H1(i   ,k,j))+           &
            mrdy*0.5*((rr(i,k,j+1)+rr(i,k,j))*H2(i,k,j+1)-            &
                      (rr(i,k,j-1)+rr(i,k,j))*H2(i,k,j  ))-            &
            msft(i,j)*(H1avg(i,k+1,j)-H1avg(i,k,j)+                    &
                       H2avg(i,k+1,j)-H2avg(i,k,j)                      &
                                 )/dzetaw(k)                            &
           )
   ENDDO
   ENDDO
   ENDDO
 . . .
```



(halo updates: get values from memory of left processor to memory of right neighbour processor)



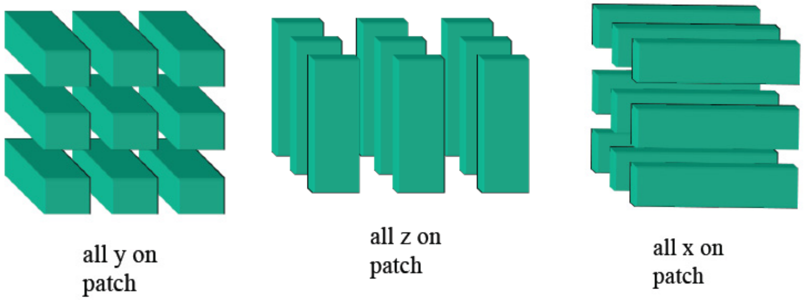


(Periodic boundary updates use interprocess communication)

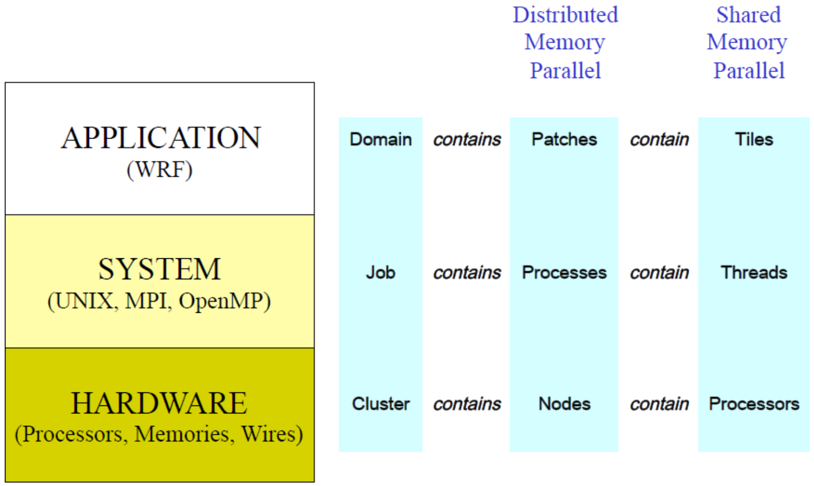
modified from [19] WRF – Code and Parallel Computing

Climate – WRF Model Parallel Application – Parallelization (3)

- Usage for transposes
  - Different parallel transposes are supported
  - Take advantages of MPI



- Overview of parallelization
  - HPC terminology vs. application domain-specific terminology
  - Evolved differently in time



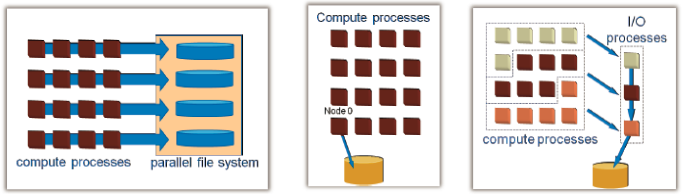
			Distributed Memory Parallel		Shared Memory Parallel
APPLICATION (WRF)	Domain	*contains*	Patches	*contain*	Tiles
SYSTEM (UNIX, MPI, OpenMP)	Job	*contains*	Processes	*contain*	Threads
HARDWARE (Processors, Memories, Wires)	Cluster	*contains*	Nodes	*contain*	Processors

[19] WRF – Code and Parallel Computing

- **One of the most common misunderstandings between the technical HPC community and the application domain-specific communities (e.g. climate) are wrongly interpreted terminologies**

Climate – WRF Model Parallel Application – pNetCDF

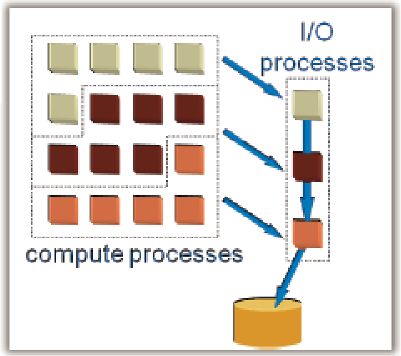
- Need for Parallel I/O (cf. Lecture 4)
  - WRF is output-bound (‘writes costs much’)
- Use Serial & parallel NetCDF (cf. Lecture 4)
  - Provides an I/O layer implemented with parallel NetCDF (pNetCDF)
  - I/O performance gain is considerable against using not pNetCDF

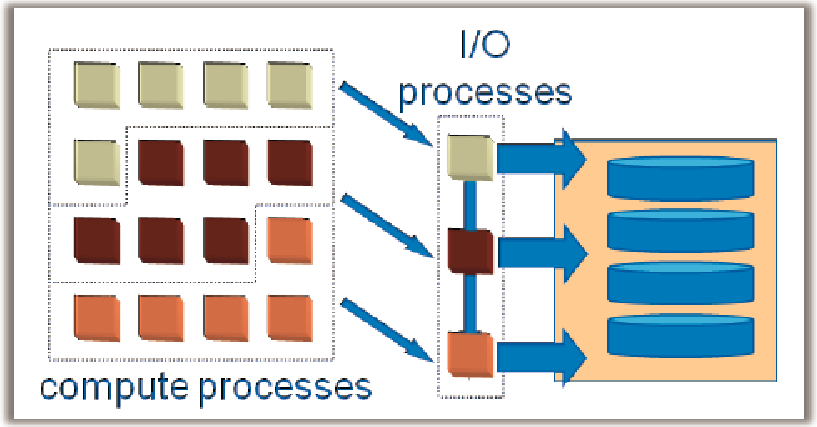


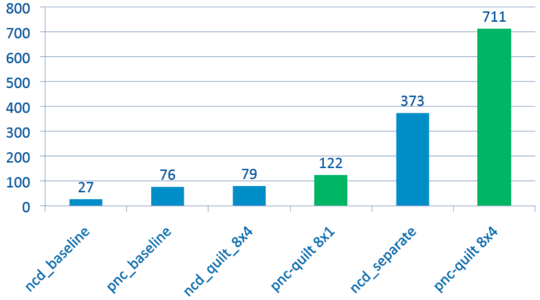
(different options that do not scale)

Serial NetCDF collected and written by gangs of MPI tasks (quilting)

Parallel NetCDF written to single files by all MPI tasks in a gang



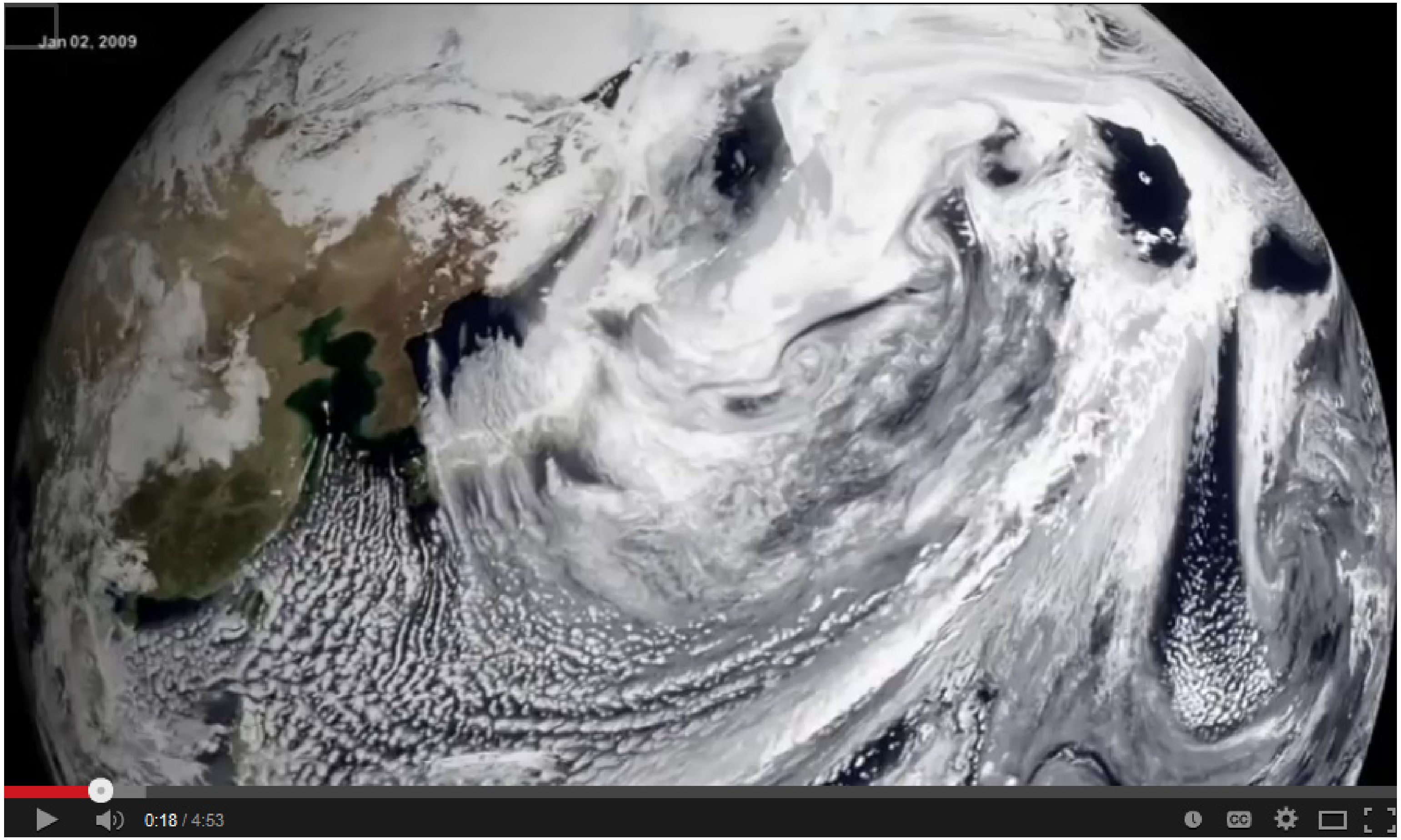




[20] Opportunities for WRF Model Acceleration

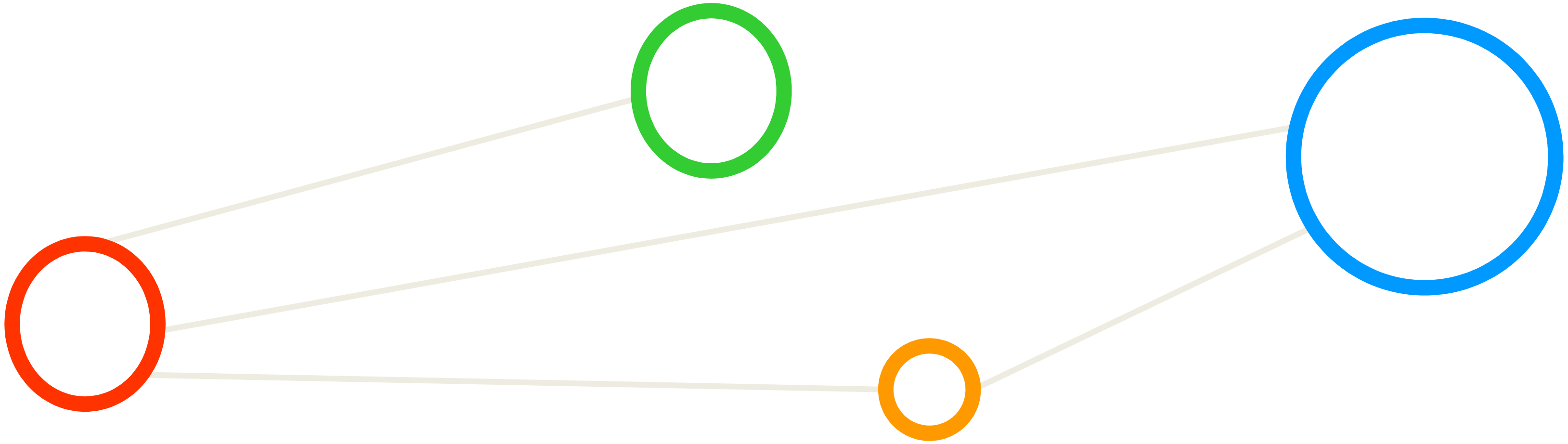
- **Parallel NetCDF can be used to significantly improve I/O output performance of WRF codes**

[Video] Climate Modeling with Supercomputers



[21] Climate modelling with Supercomputers

Lecture Bibliography



Lecture Bibliography (1)

- [1] ‘CINECA – 13th Summerschool on Scientific Visualization‘,
  Online: http://www.hpc.cineca.it/content/training-material
- [2] M. Riedel, Th. Eickermann, S. Habbinga, W. Frings, P. Gibbon, D. Mallmann, F. Wolf, A. Streit, Th. Lippert, Felix Wolf, Wolfram Schiffmann, Andreas Ernst, Rainer Spurzem, Wolfgang E. Nagel, "Computational Steering and Online Visualization of Scientific Applications on Large-Scale HPC Systems within e-Science Infrastructures," e-science, pp.483-490, Third IEEE International Conference on e-Science and Grid Computing, 2007
- [3] Terrestrial Systems Simulation Lab,
  Online: http://www.hpsc-terrsys.de/simlab
- [4] Lecture notes Introduction to Scientific Computing, TU Braunschweig,
  Online: https://www.tu-braunschweig.de/wire/lehre/skripte/index.html;jsessionid=TRIFORK661360156949
- [5] Wikipedia on ‘Partial Differential Equation‘,
  Online: http://en.wikipedia.org/wiki/Partial_differential_equation
- [6] Reed Maxwell, ‘The ParFlow Hydrologic Model: HPC Highlights and Lessons Learned’
- [7] ParFlow Project
  Online: http://computation.llnl.gov//casc/parflow/parflow_home.html
- [8] Community Land Model (CLM),
  Online: http://www.cgd.ucar.edu/tss/clm/
- [9] Consortium for Small-scale Modeling (COSMO),
  Online: http://www.cosmo-model.org/
- [10] F. Gasper et al.: Scaling of the fully coupled Terrestrial Systems Modeling Platform on BlueGene/Q,
  Online: http://www.geosci-model-dev.net/7/2531/2014/gmd-7-2531-2014.pdf
- [11] YouTube Video, ‘ParFlow coupled to CLM‘,
  Online: http://www.youtube.com/watch?v=x2ZL_U6IIR0

Lecture Bibliography (2)

- [12] Wikipedia on ‘Numerical Weather Prediction’,
  Online: http://en.wikipedia.org/wiki/Numerical_weather_prediction
- [13] Wikipedia on ‘Primitive Equations‘,
  Online: http://en.wikipedia.org/wiki/Primitive_equations
- [14] Wikipedia on ‘WRF’,
  Online: http://en.wikipedia.org/wiki/Weather_Research_and_Forecasting_Model
- [15] Webpage ‘Polar WRF‘,
  Online: http://polarmet.osu.edu/PWRF/
- [16] Nikolai Nawri et al., ‘Evaluation of WRF Mesoscale Model Simulations of Surface Wind over Iceland’,
  Online: http://en.vedur.is/media/2012_010_web.pdf
- [17] Nikolai Nawri, Guðrún Nína Petersen, Halldór Björnsson & Kristján Jónasson (2013). The wind energy potential of Iceland.Reykjavík: Veðurstofa Íslands,
  Online: http://www.vedur.is/media/vedurstofan/utgafa/skyrslur/2013/2013_001_Nawri_et_al.pdf
- [18] WRF Model Webpage,
  Online: http://www.wrf-model.org/index.php
- [19] John Michalakes et al., ‘WRF Software: Code and Parallel Computing’,
  Online: http://www2.mmm.ucar.edu/wrf/users/tutorial/201301/gill_software.pdf
- [20] John Michalakes et al., ‘Opportunities for WRF Model Acceleration’,
  Online: http://www2.mmm.ucar.edu/wrf/users/workshops/WS2012/ppts/1.4.pdf
- [21] YouTube Video, ‘Climate Modeling With Supercomputers’,
  Online: http://www.youtube.com/watch?v=izCoiTcsOd8


services
measurement
within
fusion
new
funding
Policy-based
concepts
forms
cross-disciplinary
resources
device
Analysis
Centre
climate
Computer
expertise
year
like
Computational
Test
E.g
Computing
disciplines
results
Approaches
construction
Fusion
Enable
Cross-Disciplinary
EU
methods
big
nodes
PB
community
Environment
Modelling
different
Science
High
centers
Key
scientific
important
files
model
etc
DKRZ
Use
storage
basis
Infrastructure
international
simulations
ENES
technologies
SMAQ
engineering
References
formats
physical
brain
increasing
via
EUDAT
IPCC
Big
Project
Data
processing
Common
TB
Methods
MapReduce
create
computing
Using
European
databases
GB
Supercomputing
Work
images
codes
analysis
stored
Resources
analyze
Large
Summary
data-intensive
Service
Scientific
levels
often
use
DLCL
Level
external
JSC
computational
Research
Matters
Concrete
performance
research
e.g
science
HPC
Juelich
system
datasets
acquisition
compute
Landscape
management
System
Climate
modelling
AR5
Hardware
cores
access
hundreds
Nutshell
Services
Technologies
Earth
large-scale
Structure
current
network
DLCLs
Understanding
structures
directory
project
General
Germany
Simulation
Processing
distributed
Provide
NASA
Energy
systems
day
Health