Climate Science I/O

A sketch on Requirements, Trends & Future Challenges

Nathanael Hübbe
nathanael.huebbe@informatik.uni-hamburg.de
Julian Kunkel
julian.martin.kunkel@informatik.uni-hamburg.de

Scientific Computing
Department of Informatics
University of Hamburg

10-01-2013

About us

- Scientific Computing Workgroup
  at the University of Hamburg
- Located at the German Climate Computing Center
- Brings together computer scientists and geo scientists
- Focus on storage related research

informatik
die zukunft

Workflow

- Model input
  - Can come from other model, some need preprocessing
- Model execution
  - Produce data
- Postprocessing
  - Analyze and visualize data
  - Time series generation (data transposition)
- Archive
  - Required for profound scientific working (DFG: 10 years)
  - Example: World Data Center for Climate at DKRZ
  - Active storage possibility: lossless compression
- Distribution
  - Some scientists use only data produced by others

Data

- A number of physical variables stored as arrays
  - Temperature, pressure, wind, ...
- Array dimensions: longitude, latitude, height & time
- Icosahedral models:
  longitude & latitude $\rightarrow$ surface coordinate

Typical data sizes

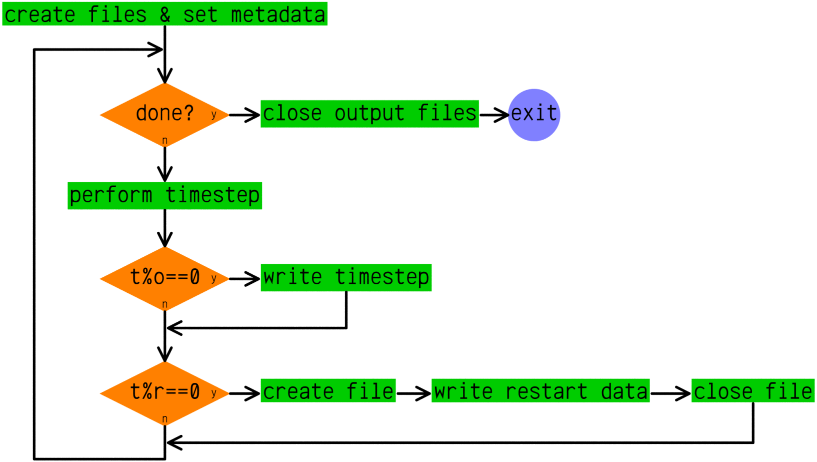
- At least 20 k surface points
  x number of height levels
  x number of physical variables
  x number of timesteps
  $\Rightarrow$ typical timestep sizes $\approx$ 200 MiB
- Not everything calculated can be stored
  $\Rightarrow$ typical dataset sizes $\approx$ 10 GiB to 1 TiB

Expected future data sizes

- Climate scientists want cloud resolving models
  $\Rightarrow$ Required cell size $\approx 1\,km^2$
  $\Rightarrow$ 500 M grid points needed
- A single timestep of a 3D variable $\approx$ 200 GB
- 40 variables, 365 timesteps $\approx$ 3 PB

Computer/Storage/Network technology are the limiting factors

A schematic algorithm of model I/O



Patterns

- Output is done ...
  - ... appending data to files
  - ... timestep by timestep (bursts)
  - ... with large requests
- Input is more diverse
  - May select only parts of a file (postprocessing)
  - May read complete files at once
  - May read timestep by timestep (forcing)

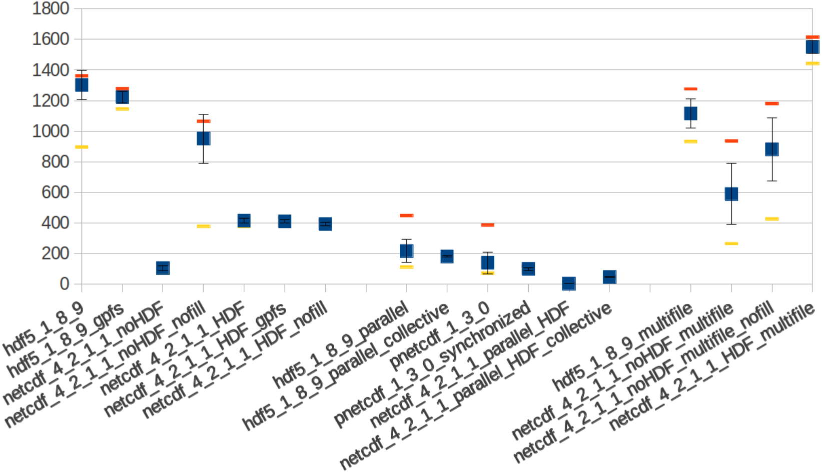
I/O Middleware

- Libraries & file formats
  - Message based: GRIB & BUFR
  - Index based: NetCDF & HDF5
- Metadata
  - Normally added via standardized attributes
    - Example:
      NetCDF Climate and Forecast (CF) Metadata Convention
  - Self contained files force metadata replication
  - GRIB & BUFR metadata is table driven
    $\Rightarrow$ Significantly smaller, replication not an issue

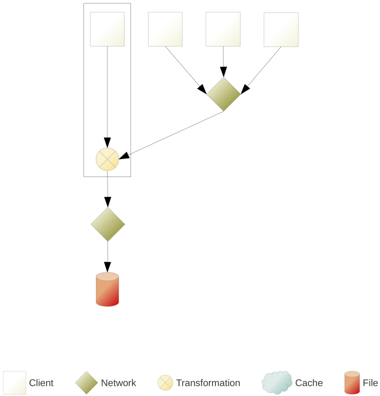
Trends and Challenges

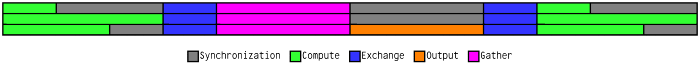
- Sequential input dominates
- Output is shifting towards parallel/asynchronous
- Trend towards overlapping I/O and computation
  - Mostly via dedicated output processes
- Trend towards outsourcing of I/O
  - This is tightly coupled middleware
  - Examples are XIOS & cdi
- Trend towards high level automation
  - Tools that run & monitor complex workflows
  - Examples: cylc, ecflow & SMS
- Roughly one I/O process per node envisioned
  - Depends on postprocessing workload & effective I/O speed

Performance of different library builds on the blizzard p249, parallel runs with 4 processes

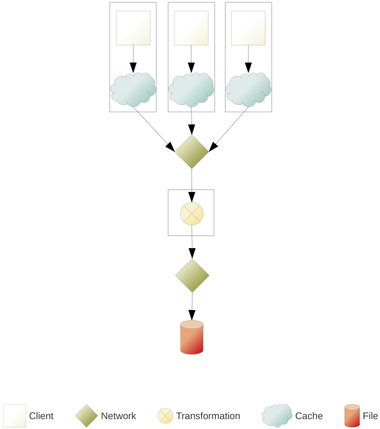


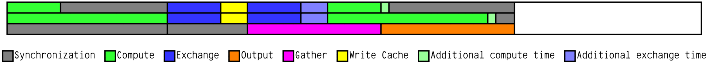
Master gathers synchronously



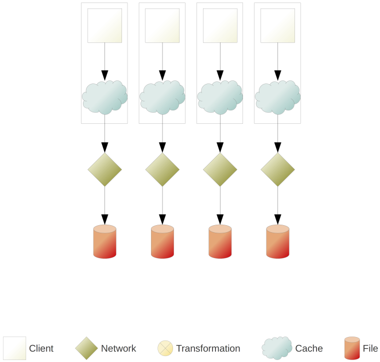


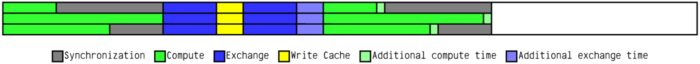
Output process gathers asynchronously





Direct asynchronous output





Summary

- Parallel applications access files sequentially (mostly)
- Large volumes of regular structured data
- Multiple competing file formats & libraries
- Compression is useful, most effective on application level
- Trends towards outsourcing & high level automation
- Best way of I/O parallelization is still unclear