Bringing Parallelism to Large-Scale Climate Data Analysis and Visualization

Robert Jacob

2nd IS-ENES Workshop on High Performance Computing for Climate Models
February 1, 2013
Toulouse, France
Motivation

- CAM-SE at 0.125 degrees
  - Single 3D variable: 616 MB
  - Single 2D variable: 25 MB
  - Single history file: 24 GB
  - 1 year of monthly output: 288 GB
  - 100 years of monthly: 28.8 TB
    - CMIP3: 35TB

Output data getting larger

Grids no longer rectangular
ParVis philosophy: Insight about climate comes mostly from computationally undemanding (to plot) 2D and 1D figures.

Why? The atmosphere and ocean have a small aspect ratio; 10,000 km vs. 10 km.
Existing Data Analysis and Visualization (DAV) tools have not kept up with growth in data sizes and grid types.

- NCAR Command Language (NCL)
- Climate Data Analysis Tools (CDAT)
- Grid Analysis and Display System (GrADS)
- Ferret

No parallelism. Assume lat-lon grids
(Parallel Analysis Tools and New Visualization Techniques for Ultra-Large Climate Data Sets)

- Speed up data analysis and visualization through data- and task-parallelism
- AND natively support multiple grids
- AND reconstruct the discretization used in the models.
Approach

 Use existing tools to speed-up development.

 As much as possible, preserve well-established workflows for analyzing climate data, just speed them up.

 There is a problem right now so provide both immediate and long-term help

Argonne Leadership Computing Facility
Hardware Layout

**Intrepid**
- 40 racks/160k cores
- 556 TF

**Eureka (Viz)**
- 100 nodes/800 cores
- 200 GPUs
- 100 TF

**Networks**
- (ESnet, internet2, UltraScienceNet,...)

**I/O Switch Complex**
- 640 @ 10 Gig

- (16) DDN 9900 – 128 file servers
  - /intrepid-fs0 (GPFS) 4.5PB
  - /intrepid-fs1 (PVFS) 0.5PB
  - Rate: 60+ GB/s

- (4) DDN 9550 – 16 file servers
  - /gpfs/home 100TB
  - Rate: 8+ GB/s

- (1) DDN 9900 - 8 file servers

- Tape Library 5PB
  - 6500 LT04 @ 800GB each
  - 24 drives @ 120 MB/s each
NCAR Command Language (NCL)

A scripting language tailored for the analysis and visualization of geoscientific data

1. Simple, robust file input and output
2. Hundreds of analysis (computational) functions
3. Visualizations (2D) are publication quality and highly customizable

- Community-based tool
- Widely used by CESM developers/users
- UNIX binaries & source available, free
- Extensive website, regular workshops

http://www.ncl.ucar.edu/
Metrics and scope of NCL usage

- 10770+ registered users in 128 countries
- A couple hundred email postings a month
- ~1800 downloads a month
Immediate help: task-parallel versions of diagnostic scripts using Swift

- Swift is a parallel scripting system for Grids and clusters
  - for loosely-coupled applications - application and utility programs linked by exchanging files
- Swift is easy to write: simple high-level C-like functional language
  - Small Swift scripts can do large-scale work
- Swift is easy to run: a Java application. Just need a Java interpreter installed.
- Swift is fast: Karajan provides Swift a powerful, efficient, scalable and flexible execution engine.
  - Scaling close to 1M tasks – .5M in live science work, and growing
- Swift usage is growing:
  - applications in neuroscience, proteomics, molecular dynamics, biochemistry, economics, statistics, and more.
Task-parallel atmosphere model diagnostics:

- CESM Atmospheric model working group diagnostics
  - Compares 2 CAM simulations or compares one CAM simulation to observational data
  - Controlled from a top level C-Shell script that calls NCO functions and NCL to create climate average files and over 600 plots that are browsable through a web interface

- ParVis created Swift-based AMWG diagnostics
  - Officially part of version 5.3 of AMWG released in Feb, 2012.
  - Used daily at NCAR
  - Installed on NCAR and DOE machines.
  - Future development will include calling data-parallel functions
AMWG Diagnostics

Set Description
1. Tables of ANN, DJF, JJA, global and regional means and RMSE.
2. Line plots of annual implied northward transports.
3. Line plots of DJF, JJA and ANN zonal means.
4. Vertical contour plots of DJF, JJA and ANN zonal means.
4a. Vertical (XZ) contour plots of DJF, JJA and ANN meridional means.
5. Horizontal contour plots of DJF, JJA and ANN means.
6. Horizontal vector plots of DJF, JJA and ANN means.
7. Polar contour and vector plots of DJF, JJA and ANN means.
8. Annual cycle contour plots of zonal means.
10. Annual cycle line plots of global means.
11. Pacific annual cycle, Scatter plot plots.
12. Vertical profile plots from 17 selected stations.
13. Cloud simulators plots.
15. Annual Cycle at Select Stations plots.

Click on Plot Type

Tables

METRICS
AMWG Diagnostic Package Timings
5 years of .10 degree data on midway
Task parallel ocean model diagnostics

- OMWG diagnostics used non-free software.
- ParVis seeks to use/create only Free and Open Source Software.
- While building Swift version, convert the OMWG diags to all-NCL
  - 87 scripts converted from IDL to NCL
  - All three top-level OMWG control scripts modified to run NCL-based scripts exclusively
  - Graphics appear very similar with identical color table and level spacing to IDL graphics
OMWG diagnostics: Sea Surface Height

Original

NCL
OMWG Diagnostic Package ran with 1 year of 1/10 degree history files (ran on lens)
ParGAL - Parallel Gridded Analysis Library (long term)

- The main product from ParVis.
  - Data parallel C++ Library
  - Typical climate analysis functionality (such as found in NCL)
  - Structured and unstructured numerical grids

- Built upon existing tools
  - MOAB
  - Intrepid
  - PnetCDF
  - MPI

- Provides a data-parallel core to perform typical climate post-processing operations.

- Handles unstructured and semi-structured grids in all operations by building on MOAB and Intrepid. Supports parallel I/O by using PnetCDF.
PNetCDF: NetCDF output with MPI-IO

- Based on NetCDF
  - Derived from their source code
  - API slightly modified
  - Final output is indistinguishable from serial NetCDF file

- Additional Features
  - Noncontiguous I/O in memory using MPI datatypes
  - Noncontiguous I/O in file using sub-arrays
  - Collective I/O

- Unrelated to netCDF-4 work
Mesh-Oriented DataBase (MOAB)

- MOAB is a library for representing structured, unstructured, and polyhedral meshes, and field data on those meshes
- Uses array-based storage, for memory efficiency
- Supports MPI-based parallel model
  - HDF5-based parallel read/write on (so far) up to 16k processors (IBM BG/P)

- Interfaces with other important services
  - Visualization: ParaView, VisIt
  - Discretization: Intrepid (Trilinos package)
  - Partitioning / load balancing: Zoltan

Greenland ice bed elevation (in Paraview/MOAB)  
Jakobshavn ice bed (in VisIt/MOAB)
A Trilinos package for compatible discretizations: a suite of stateless tools for

- Cell topology, geometry and integration
- Discrete spaces, operators and functionals on cell worksets
- Up to order 10 $H(\text{grad})$, $H(\text{curl})$ and $H(\text{div})$ FE bases on Quad, Triangle, Tetrahedron, Hexahedron, and Wedge cell topologies
- High quality cubature, e.g., positive weights only on Tri and Tet cells
- Flexible and extensible design: easy to add tools for new cell shapes and basis functions

Design features

- Multi-indexed Scalar value is the “only” data type in Intrepid. Implemented as multi-dimensional array (MDA): contiguous data layout with multi-index access.
- optimized multi-core kernels; optimized assembly
- Can compute div, grad, curl on structured or unstructured grids maintained by MOAB.

Developers: P. Bochev, D. Ridzal, K. Peterson, R. Kirby

http://trilinos.sandia.gov/packages/intrepid/
ParGAL represents discretizations as they are in the model. Algorithms are aware of grid location of data.

CAM’s Finite Volume Grid

Note: Community should decide on grid metadata standards ASAP
ParGAL Architecture

ParGAL Application

Fileinfo

PcVAR

Analysis

Mesh Oriented datABase (MOAB)

Fileinfo

PcVAR

Analysis

File

User

Native

Intrepid

ParGAL Application

Parallel netCDF

HDF5

PROF

ERR

MEM

LOG
Calculating Streamfunction and Velocity Potential with ParGAL (using Intrepid).

- The finite element method is used to solve the following weak equations for streamfunction and velocity potential using Intrepid

\[ \int \nabla \psi \cdot \nabla \phi \, d\Omega = \int v \cdot (k \times \nabla \phi) \, d\Omega \]

\[ \int \nabla \chi \cdot \nabla \phi \, d\Omega = \int v \cdot \nabla \phi \, d\Omega \]

- Periodic boundary conditions along the latitudinal boundary and Neumann boundary conditions at the poles are used

\[ \int_{\Gamma} \left( \frac{\partial \chi}{\partial n} - v \cdot n \right) \, d\Gamma = 0 \]

\[ \int_{\Gamma} \left( \frac{\partial \psi}{\partial n} - v \cdot t \right) \, d\Gamma = 0 \]

- The weak equations hold on arbitrary subdomains thereby enabling calculations from regional velocity data (e.g. WRF grids)

- Intrepid can support solution of these equations on triangles and quads and eventually on polygons.
Calculating Vorticity with ParGAL

- Calculated locally on each element
- Easily parallelizable
- Global data not required

\[ \text{vorticity} = \frac{1}{r \cos \phi} \frac{\partial v}{\partial \lambda} - \frac{1}{r} \frac{\partial u}{\partial \phi} + \frac{u}{r} \tan \phi \]

- Uses spherical harmonics
- Requires global data
Calculating Vorticity with ParGAL

Total Execution time for reading 4 timesteps of data and calculating the vorticity field on each level of a 768x1152x26 grid (FV 0.25) vs. number of cores.
Calculating Streamfunction

Intrepid
finite element method

NCL (uv2sfvpG)
spherical harmonics

\[ \nabla^2 \psi = \nabla \times \mathbf{v} \]
Calculating Velocity Potential

Intrepid
finite element method

NCL (uv2sfvpG)
spherical harmonics

\[ \nabla^2 \chi = \nabla \cdot \mathbf{v} \]
Calculating Vorticity and Divergence with ParGAL on CAM-SE grid

Vorticity

Divergence

(Rossby wave test case)
ParVis long term project: ParNCL

- ParGAL is just a library.

- ParVis will provide an application using ParGAL to support climate analysis.
  - ParNCL
NCL architecture

User scripts and shared object extensions / File utility scripts

NCL Interpreter

NCL Object Model
- File object
- Variable object
- Multidimensional object
- Graphics object

File I/O
- NIO C library
  - NetCDF4
    - HDF4
    - HDF5
    - HDF4-EOS
    - HDF5-EOS
    - Grib 1
    - Grib 2
    - GDAL/PROJ.4
    - OPeNDAP
    - NetCDF 3
    - NetCDF 4 classic
    - GRIB 1
    - GRIB 2
    - HDF4
    - HDF4-EOS
    - HDF5-EOS
    - shapefile

Analysis
- NFP C wrapper library
- NFP Fortran library
  - Linear algebra (LAPACK/BLAS)
  - FFTs (FFTPACK5)
  - Spherical harmonics (SPHEREPACK)
  - Units conversion library (UDUNITS/NCVIEW)
  - User-contributed C/Fortran code

Visualization
- High-level C Library
  - Graphical objects
  - Data/map transformation objects
  - Triangle library
  - Workstation objects
- Low level C/Fortran graphical libraries
  - contour, vector, map, streamline, font/map databases, primitives
- GKS display C/Fortran library
  - Caireographics (new PS, PDF, PNG, future output formats)
  - X11 window
  - original PS, PDF
ParNCL architecture

User scripts and shared object extensions / File utility scripts

ParNCL Interpreter

ParNCL Object Model
- File object
- Variable object
- Multidimensional object
- Graphics object

Visualization
- High-level C Library
  - Graphical objects
  - Data/map transformation objects
  - Triangle library
  - Workstation objects

Low level C/Fortran graphical libraries
- contour, vector, map, streamline,
  font/map databases, primitives

GKS display C/Fortran library
- Cairographics (new PS, PDF, PNG, future output formats)
- X11 window
- original PS, PDF

PnetCDF

ParGAL analysis
## ParGAL/ParNCL Function Table

<table>
<thead>
<tr>
<th>NCL function Group</th>
<th>NCL Functions</th>
<th>ParCAL Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>File IO</td>
<td>addfile, addfiles</td>
<td>fileinfo, pcvar</td>
</tr>
<tr>
<td>Spherical Harmonic Routines</td>
<td>dv2uv* (4 funcs)</td>
<td>divergence</td>
</tr>
<tr>
<td>Meteorology</td>
<td>uv2dv_cfd</td>
<td>divergence</td>
</tr>
<tr>
<td>Spherical Harmonic Routines</td>
<td>uv2dv* (4 funcs)</td>
<td>divergence</td>
</tr>
<tr>
<td>Meteorology</td>
<td>uv2vr_cfd</td>
<td>vorticity</td>
</tr>
<tr>
<td>Spherical Harmonic Routines</td>
<td>uv2vr* (4 funcs)</td>
<td>vorticity</td>
</tr>
<tr>
<td>Spherical Harmonic Routines</td>
<td>uv2vrdv* (4 funcs)</td>
<td>vorticity, divergence</td>
</tr>
<tr>
<td>General Applied Math</td>
<td>dim_avg, dim_avg_n</td>
<td>dim_avg_n</td>
</tr>
<tr>
<td>General Applied Math</td>
<td>dim_max, dim_max_n</td>
<td>dim_max_n</td>
</tr>
<tr>
<td>General Applied Math</td>
<td>dim_min, dim_min_n</td>
<td>dim_min_n</td>
</tr>
<tr>
<td>General Applied Math</td>
<td>dim_median, dim_median_n</td>
<td>dim_median_n</td>
</tr>
<tr>
<td>General Applied Math</td>
<td>max</td>
<td>max</td>
</tr>
<tr>
<td>General Applied Math</td>
<td>min</td>
<td>min</td>
</tr>
<tr>
<td>Variable Manipulators</td>
<td>delete</td>
<td>pcvar</td>
</tr>
<tr>
<td></td>
<td></td>
<td>gather</td>
</tr>
</tbody>
</table>
ParNCL

- ParNCL supports `addfiles()`, NCL’s multi-format file reader
  - Time slices of the variable read from file works
  - Only reads NetCDF for now (using PNetCDF) from CAM.

- Addition, subtraction of distributed multidimensional data works

- Scaling a distributed multidimensional array by a scalar works

- Subsetting a distributed multidimensional array is supported.

- ParGAL and ParNCL source and ParNCL executable (beta1) all released!
An NCL script executed by ParNCL

```ncl
f = addfiles(diri+filn, "r") ; open file

tt = f[:]->T(0,{500},{-30:30},:); read a section of data.

wks = gsn_open_wks("ps","parvis_t") ; open a PS file

plot = gsn_csm_contour_map(wks, tt(:,,:),False)
```

In the ParNCL interpreter, tt is gathered to one node and passed to normal NCL graphics routines for plotting.

Using ParNCL requires a parallel environment:
Now: > ncl myscript.ncl

With ParNCL: > mpirun -np 32 parcnl myscript.ncl
Future: ParNCL with ESG via LAS
The “vis” in ParVis: Interactive Visualization of Large Geodesic Grid Data
Kwan-Liu Ma, UC Davis

Existing 3D visualization solutions:
- Require a pre-partitioning of each hexagonal cell into multiple tetrahedral cells.
- Do not take advantage of latest GPU features
- Do not offer high-quality rendering

The UC Davis team seeks to provide:
- Advanced visualization of hexagonal grid data
- High quality 3D rendering
- GPU acceleration and parallelization to support Interactive interrogation
Interactive Visualization of Large Geodesic Grid Data
Kwan-Liu Ma, UC Davis

Data: CSU GCRM
Additional ParVis area: Compression

- Completely random data can not be compressed without information loss but many climate output fields are smooth, not random.

- **Lossless** compression can reduce volume of the climate data without information loss
  - Reduce storage, memory, and network requirements to store, process, and transfer the data
  - Compression can potentially speedup analysis and visualization applications
    - Light weight and Integrate well with the applications

- **Lossy** compression can achieve higher compression ratio
  - May be appropriate for some applications.

- Need for compression is here now.
Lossy Compression results

- Error for each value is bounded
- Preliminary results show that we can achieve a compression ratio around 10 when the error bound is 0.1%
- Further improvement is possible with improvement in the second part of our two-stage compression
Incorporating Compression: File System Based Approach

- Change the file systems to return decompressed data
  - Caching to reduce computation overhead
  - Pipelining and Prefetching to reduce latency

- Advantage
  - Application transparent

- Disadvantage
  - Does not reduce communication cost
  - Need to change file systems
Incorporating compression: PNetCDF

- Fetch compressed data through MPI-IO
- Advantages
  - Reduce disk overhead
  - Reduce communication overhead
  - Can be added to existing applications (PIO, ParGAL)
- Disadvantage
  - Challenging when PnetCDF accesses and data compression are not aligned
  - Pipelining is difficult

- Implemented a proof of concept prototype and performed some preliminary measurements
  - Read a 2.7 gb netcdf file with uncompressed data, 39.454 seconds, with compressed data, 27.429 second
ParVis Team

- At Argonne:
  - Rob Jacob, Xiabing Xu, Jayesh Krishna, Sheri Mickelson, Tim Tautges, Mike Wilde, Rob Ross, Rob Latham, Mark Hereld, Ian Foster

- At Sandia:
  - Pavel Bochen, Kara Peterson, Dennis Ridzal, Mark Taylor

- At PNNL
  - Karen Schuchardt, Jian Yin

- At NCAR
  - Don Middleton, Mary Haley, Dave Brown, Rick Brownrigg, Dennis Shea, Wei Huang, Mariana Vertenstein

- At UC-Davis
  - Kwan-Lu Ma, Jinrong Xie

Supported by the Earth System Modeling Program of the Office of Biological and Environmental Research of the U.S. Department of Energy's Office of Science
ParVis released software:
  - Swift-enabled task-parallel diagnostic scripts
    - OMWG and AMWG released through NCAR.
    - Land and Sea ice coming soon
  - ParGAL source code
  - ParNCL source code and binary.

What you can do:
  - Check the website: trac.mcs.anl.gov/projects/parvis
  - Subscribe to ParVis announcement mailing list: parvis-ann