A Parallelized and Provenance-Aware Framework for Climate Analysis and Beyond

Brett Yasutake | University of Washington | December 3rd, 2013
Team Members and Faculty Advisors

Team Members: Brett Yasutake
University of Washington
Computer Science and Software Engineering

Niko Simonson
University of Washington
Computer Science and Software Engineering

Faculty Advisors: Dr. Munehiro Fukuda
University of Washington
Computer Science and Software Engineering

Dr. Eric Salathé
University of Washington
Climate Science and Policy

Dr. Hazeline Asuncion
University of Washington
Computer Science and Software Engineering
- Physical Measurements → NetCDF
- Predicted atmospheric conditions
- 100+ year simulation

Climate Model Analysis
Data analysis tools have not kept pace with our ability to capture and store data [Gray et al.]

Early 1990s
Data analysis tools have not kept pace with our ability to capture and store data [Gray et al.]

1996

image source: [3], [4]
Data analysis tools have not kept pace with our ability to capture and store data [Gray et al. 2001].

2001

TAR

~180 km (T63)

image source: [3], [4]
Data analysis tools have not kept pace with our ability to capture and store data [Gray et al. 2007]

AR4

~110 km (T106)

image source: [3], [4]
Improvement Over Existing Tools

- Custom one-off programs
  - Written for a single analysis → No re-usability

- NetCDF Operators (NCO)
  - Limited choices, simple operations
  - Difficult to use, no data provenance

- Parallel NetCDF
  - Difficult to use, no data provenance

- Climate Data Operators (CDO)
  - Difficult to use, no data provenance

image source: [5]
Project Goals

- Perform large-scale, parallelized analysis on NetCDF data
- Create a general framework to enable other researchers to do similar work
- Store data provenance in a manner to support easy query and retrieval [Davis et al.]
- Balance data provenance capture with overhead and storage constraints [Reilly et al.]
Our Solution

- Pacific Northwest Climate Analysis (PNCA)
  - MASS provides under-the-hood parallelization
  - Native NetCDF integration
  - Automatic data provenance collection

- Scientific Data Analysis Framework
  - Use of adapters enables a modular and extendible framework
  - Hides unnecessary programming tasks for climate researchers
Why MASS?

- Allows *further* abstraction of technical details for non-computer scientists
- Existing support backdrop of MASS research team
- Sustainability of the project
<table>
<thead>
<tr>
<th>Area</th>
<th>Brett Yasutake</th>
<th>Niko Simonson</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project Management</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Risk Management</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>- Quality Assurance</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>- Data Provenance</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>- Documentation</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>- Metrics Collection</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td><strong>Parallel Processing</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Architectural Design</td>
<td></td>
<td>○</td>
</tr>
<tr>
<td>- Parallelization</td>
<td></td>
<td>○</td>
</tr>
<tr>
<td>- Algorithm Development</td>
<td></td>
<td>○</td>
</tr>
<tr>
<td>- Schema Development</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>- NetCDF Reader</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td><strong>Data Management</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Data Processing</td>
<td></td>
<td>○</td>
</tr>
<tr>
<td>- Data Distribution</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>- Data Collection</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>- Data Analysis</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td><strong>Data Presentation</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Data Visualization</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>- UI Design</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>
Our Project Management Approach

- Two Week Sprints
- Evolutionary Prototyping
- Pair Programming

- Risk Analysis
  - Test
  - Two Week Sprints
  - Evaluate Alternatives
  - Develop Deliverables and Prototypes

Image source: [8]
Final Design

Pacific Northwest Climate Analysis Framework
Data Flow Diagram

- **User Selects Analytics Module and Input Files**
- **GUI**
- **Analytics Module**
- **Results Adapter**
- **Panoply Viewer (Output)**
- **Execution Adapter**
- **Provenance Adapter**
- **Status Adapter**
- **Input Adapter**
- **netCDF files (Input)**

**Color Legend**
- Yellow: Data Inputs
- Blue: Data Processing
- Green: Data Outputs
Analytics Module Framework

- Two major required functions + user-defined functions

```java
public class YourAnalyticsModule {
    public static void initialize(StatusAdapter status) {
        // Perform any necessary initialization
    }

    public static void runAnalysis(StatusAdapter status) {
        // runAnalysis acts as a 'main' driver function from which user defined functions can be called
    }

    public static void yourUserDefinedFunction1() {
        // User defined functions
        // This is where you manipulate and share data in order to implement your data analysis algorithm.
    }
}
```
Results

- Five prototypes, 44 commits, 6,000 SLOC
- Average analytics module contains **400 SLOC** vs comparable Fortran code that contains **1600 SLOC**
  - Recommended module size for minimum defect density is 200-750 SLOC [Laird et al.]
- Moisture flux calculation for one month of data
  - 13s (PNCA) vs 2m36s (NCO)
Demonstration Video
Key Accomplishments & Challenges

- Improved over existing tools in both **performance** and **usability**
- Proved a new, real world application of MASS
- Created new ways to utilize MASS
- Memory limitations were a challenge
- Custom data visualization was a challenge
Next Steps…

● Immediate practical use
  ○ Moisture Flux, Wind Gradient, Extreme Indices
  ○ New analytics modules

● Future Work
  ○ Utilize mobile agents for complex pattern detection
  ○ Porting to C++ for increased efficiency
  ○ NetCDF → Climate science and beyond
- Niko Simonson
- Dr. Fukuda & the MASS Research Team
- Dr. Salathe & Dr. Spayde
- Dr. Asuncion & Del Davis
- Dr. Erdly & our fellow capstone project colleagues
References

Image Sources

[3] http://eo.ucar.edu/staff/rrussell/climate/modeling/images/ipcc_ar4_wg1_ch1_fig_1_2_big.jpg
What’s in a NetCDF file?

- How many measurements are in a climate model?
  - ~150 variables for each grid cell at each time slice
  - Examples: T2 (temperature at 2 meters), PSFC (surface pressure), SFROFF (surface runoff)

- How many files comprise a NetCDF file and how large are they?
  - 4 files per day → 250MB per day
  - 120 files per month → 30GB per month
  - 1,440 files per year → 360GB per year
  - 100 year simulation → 36TB per 100 years
What were your highest risks?

- NetCDF I/O bottleneck
  - Evolutionary prototypes
- Components take too long to build
  - Scope advisory
- Insufficient architecture analysis
  - Peer reviews, seek architectural design expertise from Dr. Asuncion, Dr. Fukuda
- Concurrency issues
  - Seek technical expertise of MASS project team
- Programmability by climate researchers
  - Seek usability feedback from Physical Sciences faculty member
What were some of the unknown constraints and how did you overcome them?

- Hardware memory limitations
  - Executing the analyses in “chunks”
- Shared data communication with MASS between all computing nodes
  - Implemented an encode/decode scheme in MASS’ exchange all neighbor vector
- Communication between multiple parallel computing spaces
  - Modifications were made to MASS to allow multiple executions