Collaborative Workbench to Accelerate Science Algorithm Development

Manil Maskey*, Rahul Ramachandran, Kwo-Sen Kuo, Chris Lynnes

Oct 28, 2014
Earth Science Technology Forum

mmaskey@itsc.uah.edu
Outline

• Motivation
• Collaboration
• CWB Core
• CWB-enabled Applications
  • GLM Validation Tool
  • CWB-AES Integration
  • Giovanni Integration
  • NEOS$^3$ Integration
• Conclusions and Plans
Motivation

- Scientists increasingly find themselves working with a more diverse set of data, complex science algorithms, and models.
  - Each new dataset, algorithm, or model - own knowledge
  - Acquiring this knowledge - difficult and time consuming
  - One way - social collaboration (virtually)

- Adopting social networking and collaboration
  - Only a few have been successfully customized to address and impact Earth science research modalities
  - New set of analysis tools to leverage the collaboration infrastructure
  - Adoption: inhibited by the steep learning curve
**Approach**

- **Collaborative Workbench (CWB)**
  - Augment a scientist’s current computational research environment
  - Provide a science algorithm development environment that seamlessly integrates the researcher's desktop with a cloud infrastructure.
  - Focus on enabling sharing of research artifacts + collaborative development of algorithms
  - Interpretation of data and analysis results as well as their knowledge (in the form of annotations).
  - Accommodate various modes of collaboration, in order for it to be effective.
  - Let scientists focus on science, rather than learning collaboration tools.
-COLLABORATION-
YOU + OTHER + GOAL
COLLABORATION HAS DIFFERENT LEVELS
You must be present
  Chat
  Skype
  Messenger
  Meeting

You will be present
  Email
  Forum
  Dropbox
Managing Collaboration

Versioning + Provenance

Ownership
Who created/modified it
Assumptions
Lineage
Quality
Replication Recipe
....
Science Collaborations

- Occur when 2 or more people work together to achieve a common goal, result or project
- Cover all levels of collaboration
- Find ways to share reuse data, programs, workflows, experiments ..
- Require feedback and iteration
  - Members review each others work and make revisions
- Track – who, what, when, how, why, which
Collaboration Systems

- Need to provide a mechanism for:
  - Communication
  - Content management
  - Workflow for revisions
Components of Collaboration Systems

• **Data**
  - Cloud for scaling

• **Software**
  - Cloud for sharing, scaling, and executing

• **People**
  - Different roles, levels of control, groups, …
CWB CORE
Collaborative Workbench (CWB) to Accelerate Science Algorithm Development

Sharing Knowledge is at the heart of science, yet it is challenging for researchers to effectively share information and tools.

**Goals**
- An architecture for scalable, controlled collaboration
- Selective sharing of science resources
  - among individuals
  - within science teams
  - with the entire science community.
- Software that fits how researchers currently do scientific analysis to promote adoption

**Benefits**
- Accelerate science algorithm development by distributed science teams
- Reduce redundancy
- Improve productivity
- Securely share all science artifacts (data, information, workflow, virtual machines)
- Generalizable to support collaborative science algorithm development for other mission and model enterprises
Project Objectives

• Investigate different science collaboration modalities:
  – Shared Resources, Local Computation – Dropbox Model.
  – Shared Resources, Cloud Computation
  – Shared Virtual Machine Instances

• Build **core** components required for an Earth Science Collaboratory
  – Collaborative Workbench (CWB)
  – Cloud-Interfaces
  – Catalogs
Workbench - CWB

- Sophisticated code editing, navigation, and management tools for development and execution.
- Familiarity
- Eclipse platform
  - Can be specialized for particular requirement.
- Plugin architecture
CWB – An integrated development environment
Core Components

• Communication Plugins
  – Shared Code Editing, Chats, Send files

• Catalog for Data/User/Job management
  – Search, Version, Provenance

• Development tools
  – IDL, Python, ADaM, Visual workflow composer

• Cloud job submission
  – Submit job to multiple instances (multiple software stacks, versions)
  – ENVI Service Engine backend

• Cloud Resource Management

• Share public/individual
  • Share Local, Share VM
Versioning

Data and source code versioning using git

Why git?
- git allows distributed versioning
- Integrates with S3
- Files are compressed
- Only change (deltas) stored

• Plugins
  - Hide all the version control nomenclature from scientist and provide versioning implicitly – very important
  - CWB + Cloud + Git with single sign on
CWB versioning and sharing using Git

1. Save (git commit)
2. Get a version (git reset)
3. Upload (git push & S3 policy=private)
4. Share (S3 policy=public)

Amazon S3
Remote Repository

CWB

Local Repository

Workstation

Git Account Management
Content Management System
Auto Provenance for CWB

Collaborative Workbench

- Source Code
- Temporary Files
- Compiler/Interpreter

Server

- Provenance Wrapper
- Provenance Tracer
- PROV-N Generator
- SVG Generator
- PNG Generator
- PROV Library
- PROV to SVG Converter
- SVG to PNG Converter

- Implemented as a service
- Any client can call the service
- Integrated into IDE such that provenance generation is inherent
- Ideal for Data Center processes and workflows

CWB: Auto Provenance Generation (Python)
Sharing/Versioning

target = open("sample_output.txt", 'w')
for x in range(0, 3):
target.write("Writing data to a test file" + "\n")
target.close()
Provenance

CWB Provenance Demo
Applications and Integrations

• GLM Verification and Validation

• CWB-AES

• Giovanni upload
  – Upload dataset
  – Run workflow on dataset

• Giovanni
  – Upload
  – GLIDER

• NEOS³
  – netCDF packaging of scattering database
  – Python module: job handling
GLM Verification and Validation

- Validation of the Geostationary Lightning Mapper (GLM) instrument, scheduled to launch on GOES-R in 2016.
- Close collaboration with LIS SCF.
- Uses real time flash rate datasets, such as Earth Networks Total Lightning Network, the National Lightning Detection Network (NLDN) and Vaisala’s Global Lightning Dataset (GLD360) that have been ingested into a GIS database.
- Displays thematic layers of each dataset.
- Performs real time analysis of discrepancies between GLM and ground based sensors; display interactive statistics, histograms.
- Subsets spatially and temporally.
- Saves anomaly as image/video.
- Uses CWB features to share artifacts.
- Controlled access.
GLM Verification and Validation Architecture
GLM Verification and Validation
CWB-AES Integration
Automated Event Service (AES)

- AIST, PI: Tom Clune
- Observing and studying events
- System being developed to methodically mine custom defined events
CWB-AES Integration
Starting CWB

 Authentication and authorization Handled by CWB

 Point to the appropriate HPD End Point
My Experiment on Somali Jet Detection

Import AES Library and use All the functions to write my own Script to query data, and execute optimized detection algorithms
Example methods

- Execute optimized data parallel queries on a remote HPC
- Integrate local data processing and analysis seamlessly

Run my script from CWB
Results saved in MyExperiment folder

Visualize plots with CWB
Provides interactive python shell to try methods while writing the script

Also enables user to use additional Python libraries – Ex: matplotlib
Giovanni Integration

- Giovanni
  - Goddard Interactive Online Visualization ANd aNalysis Infrastructure
  - Popular online visualization and analysis tool
  - Exploration and comparison of remote sensing data + model output

- Integration with CWB
  - CWB python to execute a Giovanni workflow
  - CWB drop-in to upload scientist’s own data
  - Giovanni modifications to support workflows with scientist’s own data
  - User-Supplied Data Use Cases
    - Upload “my” data to Giovanni for comparison with data already in Giovanni (completed)
    - Upload my data to Giovanni to allow other users to interact with it (90% completed)
Giovanni Integration++

- Earth System Grid Federation
  - NextGen ESG project, PI = C. Mattmann
  - Main Goal: publish DAAC data to ESGF
  - Spinoff: ESGF data turns out to be “Giovanni-friendly”

- CWB “Integration” with ESGF
  - Acquire data from ESGF
  - Upload ESGF data via CWB/Giovanni User-Supplied Data Uploader
  - Run ESGF data in Giovanni workflows

ESGF-Giovanni interoperability from NextGen ESG project
GLIDER Integration

• GLIDER – Globally Leveraged Integrated Data Explorer for Research
  – Desktop tool to easily visualize, analyze and mine satellite imagery
  – Allows image enhancements and pattern recognition algorithms to be applied on the satellite imagery
  – Visualization with NASA World Wind 3D tool
  – RCP application consisting of Eclipse plugins
  – Integration to CWB included dropping the GLIDER plugins into CWB dropins folder
CWB Giovanni-GLIDER plugin: Use Case

• Demonstrate that researchers can use CWB to download data using python code to their workspace and apply GLIDER plugin for image view of the data and use World Wind plugin to visualize and compare.

• Steps:
  – Access Giovanni service via Python Module – Download data to personal space in CWB
  – Convert the result data to GLIDER format
  – Use GLIDER (Eclipse Plugin) to visualize and enhance the data
  – Display the enhanced image on the NASA World Wind globe (Eclipse Plugin)
  – Overlay multiple images on the globe to see the correlation.
CWB Giovanni Python Module

Access Giovanni via Python module with PyDev

Data:
TRMM 3B42 Daily Rainfall Estimate V7 0.25 deg.
Cloud Effective Radius Combined QA Mean

Temporal Bounds:
2005-08-20 to 2005-08-30

Spatial Bounds:
127.9688, -23.9062, -156.7969, 16.1719

Execute Python Code Locally

Check the log for completion
CWB GLIDER Plugin

GLIDER functionalities readily available via contextual menus
CWB GLIDER Plugin – Image Enhancement

[Diagram showing image enhancement process]
CWB GLIDER – NASA World Wind Plugin

- Layer images
- Visualize with context
NEOS³

- NEOS³: NASA Earth Observing System Simulation Suite
  - AIST: PI Simone Tanelli
  - Web-based integrated simulator for Earth remote sensing applications.
  - Equipped with start-of-the-art modules to enable the realistic simulation of satellite observables.
  - Providing an advanced, sophisticated, and user-friendly simulator package to be used by both scientists for research-oriented applications and by system engineers for an instrument design purpose.
  - Accessible via a web interface and capable of distributing computationally intensive tasks to remote servers such as those at the NASA Advanced Supercomputing (NAS) Division.
CWB-NEOS\(^3\) integration

- Creating NEOS\(^3\) Jobs using templates
- Polling NEOS\(^3\) for completion
- Retrieving outputs to CWB
- Demo
Lessons Learned

“To complicate is easy, to simplify is hard. To complicate, just add, Everyone is able to complicate. Few are able to simplify.”

-Bruno Munari
Conclusion and Plans

• A framework for Science Collaboration
  – Augments existing tools
  – Reduced learning curve
  – Scalable
  – Extendable (Plugin Architecture)
  – Reusability for customizations

• More simplifications

• Integration with other tools based on plugin architecture
Thanks to NASA CMAC and ESTO for their support in our efforts to develop this collaboration framework