COLLABORATIVE WORKBENCH TO ACCELERATE SCIENCE ALGORITHM DEVELOPMENT

4.1

M. Maskey*
University of Alabama in Huntsville, Huntsville, AL

R. Ramachandran
NASA/MSFC, Huntsville, AL

K. Kuo and C. Lynnes
NASA/GSFC, Greenbelt, MD

1 INTRODUCTION

Issues related to Earth science collaboration are twofold. First, data, algorithms, and analysis results are scattered at different locations, usually behind complex security and firewall. Secondly, handful scientific collaboration platforms that offer sharing require scientists to learn new tools. There is high cost and big learning curve associated with these tools. To address these two issues, Earth science Collaborative Workbench (CWB) is being developed. CWB leverages Eclipse Rich Client Platform (RCP) (http://www.eclipse.org/home/categories/rcp.php) to provide an extensible framework that can be customized easily via Eclipse plugins. Earth science analysis tools such as IDL and Python have been available as Eclipse IDE (Integrated Development Environment), which are essentially Eclipse plugins with views. CWB enhances scientist's existing research environment by extending the tools that they are already familiar with, via plugins. These plugins allow CWB to provide collaboration capabilities directly but transparently within scientist's existing research tools using cloud. Cloud integration from CWB is seamless. CWB interfaces to cloud services for storage and compute resources. Additionally, CWB also interfaces with a cloud-based middleware that includes a catalog that tracks and mediates the collaboration.

2 COLLABORATION AND CWB CONCEPT

Science community is increasingly faced with providing communication systems to support collaboration. Collaboration now often goes beyond supporting small teams and can occur in every phase of science process. The speed of collaboration differs with needs. Synchronous collaboration occurs in real time such as simultaneous editing of shared document, videoconference, etc., and typically requires a peer-to-peer connection. Asynchronous collaboration follows a push-pull model. Examples of asynchronous collaboration include email exchanges, blogging, repositories, etc. Enabling collaboration enhances knowledge sharing and innovative outcomes in the increasingly linked Earth science community.

The objective of CWB is to create an open and extensible framework for the Earth Science community via a set of plug-ins. Consequently, the plugins will augment scientist's existing tools to provide collaboration functionalities out of the box. CWB is based on the Eclipse IDE. Already, many science tools are based on Eclipse or plugins to Eclipse. The heart of these tools is the Eclipse RCP.

3 COMPONENTS

Although the goal of CWB is to create a set of eclipse plugins for collaboration, additional supporting modules and components are needed to manage and scale collaboration. Next, we describe the components that are used to support CWB.

3.1 Eclipse RCP

Eclipse RCP includes Equinox, a component framework based on Open Service Gateway Initiative (OSGi) standard, and the ability to deploy native GUI applications to a variety of desktop operating systems (Clayberg, 2008). Eclipse RCP supports thousands of community-contributed plug-ins, making it a popular development platform for many diverse applications including the Science Activity Planner developed at JPL for the Mars rovers (Norris, 2005) and the scientific experiment tool Gumtree (Lam, 2005). By leveraging the Eclipse RCP to provide an open, extensible framework, the CWB supports customizations via plug-ins to build rich user applications specific for Earth

---

* *Corresponding author address*: Manil Maskey, University of Alabama in Huntsville, Information Technology and Systems Center, Huntsville, AL 35899, e-mail: mmaskey@itsc.uah.edu

Science. More importantly, CWB plug-ins can be used by existing science tools built off Eclipse such as IDL or PyDev to provide seamless collaboration functionalities. Open Service Gateway Initiative (OSGi) is the interoperable platform for Eclipse applications. By providing additional components for the integration of legacy application and for the provisioning of new applications, OSGi plays a central role in CWB. OSGi and Eclipse are loosely coupled to provide flexible architecture that allows dynamic deployment of software components. In OSGi, components are referred to as bundles. Bundles can be coupled with other bundles through services and may be extended by bundle fragments.

3.2 Communication Plugins

We extend the Eclipse Communication Framework (ECF) for synchronous real-time communication. We use Openfire implementation of the Extensible Messaging and Presence Protocol (XMPP) (http://xmpp.org/xmpp-protocols/). The CWB also allows for communication based on other providers. The communication plugins provide abilities to create targeted science community, chat, collaborative code editing and many other collaborative features. We have also developed plugins that support asynchronous collaborations. These plugins use catalog for user, group, and artifact management; and cloud resources to store and scale the shared artifacts.

3.3 Catalog

The catalog comprises of user and data model required for collaboration. Besides managing researcher's profile and role, the catalog also manages researcher's activities, projects and science artifacts. The information model of the catalog is based on the notion of an experiment. Experiments serve as containers for collections comprising of one or more logical data sets and science workflows. Workflows consist of programs and other supplementary resources. A workflow chains the science programs describes the flow of data. Catalog maintains metadata for any experiment, workflow, and resources submitted for execution and sharing. The same metadata including user defined tags are indexed for search and discovery of experiments. The catalog supports collaboration by connecting researchers and making metadata available for the shared science artifacts stored on the Cloud infrastructure.

CWB catalog is built using Drupal content management framework, a robust open-source content management framework well supported by developer community and accepted by science community. Drupal allowed us to utilize a number of existing features with minimal code development. These features include the modules that provide REST-based service APIs for the CWB to connect the catalog and built-in roles and permissions, enabling users to control permissions on the shared science artifacts with other individuals, groups, or the entire community.

3.4 Cloud Plugins

Cloud computing has now become a viable option to provide scalable computing infrastructure for enterprises. More and more enterprises are moving toward Cloud computing because of its elasticity and scalability, thereby being able to expand or reduce resources based on demand. Clouds allow multi-tenancy, where multiple users can leverage the same infrastructure and can balance workloads across resources (both are examples of collaboration). CWB cloud plugins provide building blocks for scalable collaboration in conjunction with the catalog.

For our initial implementation, we have configured our CWB to utilize Amazon's EC2 and S3 Cloud infrastructure via the AWS Java API (http://aws.amazon.com). The user's CWB account and corresponding Amazon Identity and Management (IAM) account is managed by the catalog. Execution of workflows is performed in EC2. We have developed CWB job services in EC2 to process and manage workflows submitted via CWB. User-uploaded data and experiment results are stored in S3. Workflows (and programs) can access the data stored in S3 during execution, which in turn, is transferred to EC2. There is no cost associated within Amazon for data transfers between EC2 and S3. Users are assigned a personal space within each experiment ("amazon S3 bucket") where they can create new workflows. Asynchronous sharing of data and workflows occurs by allowing a set of users to copy these resources to the searchable "community" bucket. Collaborators can also directly import the contents in any experiment folder stored in the community bucket to their personal space. Currently, any Python or IDL science code as well as some specialized libraries such as ADAM Data Mining toolkit (Rushing, 2005), can be executed in the EC2.

Advent of cloud technologies has only accelerated the rate at which scientists are dealing with large and more complex resources. Thus, CWB allows for processing of

large and complex data in cloud after examining a representative subset in CWB sandbox locally. However, the location of the processing is transparent to the user. Using CWB cloud plugins, the management of cloud resources is greatly simplified. The CWB model follows “Dropbox” like features, where the notion of cloud is non-existent during user experience.
Figure 1 illustrates the architecture of the CWB.

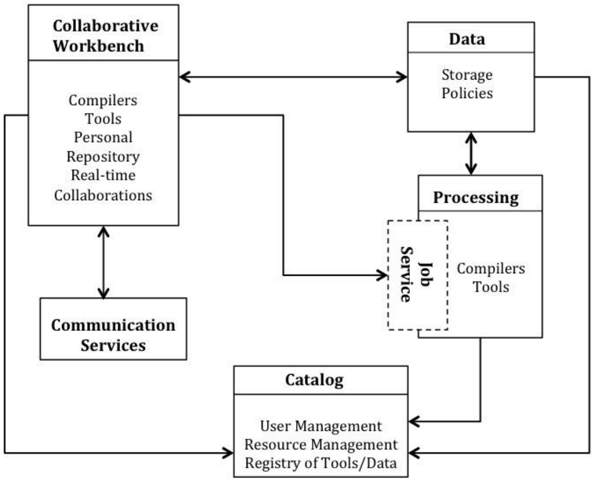


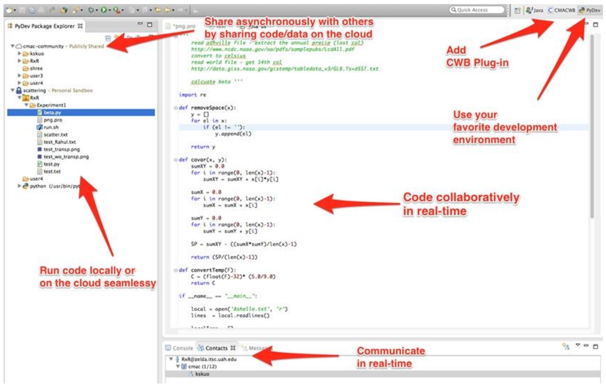
Figure 1. Collaborative Workbench Architecture

4 USE CASE

We demonstrate the value of a CWB via a use case, in which two scientists are involved in algorithm development for NASA’s Global Precipitation Measurement (GPM) mission. GPM, being an international mission involves science expertise of several disciplines with scientists spanning across the world. Each team is accustomed to their own tool and approaches to data management. This diversity makes effective collaboration exceedingly difficult. The CWB addresses all the needs of this situation.

The collaboration consists of a collaborative workflow that comprises of ice crystal modeling to generate scattering properties, precipitation modeling, and evaluation of the models. The scattering properties of precipitation particles are basic quantities for remote-sensing measurements of precipitation. These properties are used by instrument response simulator which is basically composed of a selected radiative transfer model from a suite of models as well as modules for simulating engineering components, to simulate radar and radiometer instrument responses to precipitating atmospheric columns in various weather systems and regions. These simulations in turn form a basis for precipitation retrieval.

Using CWB, scientist at NASA/GSFC, an expert in scattering properties, not only shares the entire directory structure that contains the particles and associated properties, but also shares the programs to extract statistics. The scientist annotates these artifacts with their specific meanings to help instrument simulator scientist at JPL learn to use these scattering properties correctly. The two collaborators can log into their respective CWBs on their own computers, connect and co-develop code together. Note that the collaborators are using the tools they are already familiar with. In this case, the NASA/GSFC scientist uses the CWB within IDL Workbench and JPL scientist uses the CWB within PyDev plugin. All the collaborative features are readily available via contextual menus and eclipse views within both tools. Figure 2 illustrates the CWB components that are used during the use case.



```
    read ashville file - extract the annual precip (last col)
    http://www.ncdc.noaa.gov/oa/pdfs/samplepubs/LcdAll.pdf
    convert to celsius
    read world file - get 14th col
    http://data.giss.nasa.gov/gistemp/tabledata_v3/GLB.Ts+dSST.txt

    calcuate beta '''

import re

def removeSpace(x):
    y = []
    for el in x:
        if (el != ''):
            y.append(el)

    return y

def covar(x, y):
    sumXY = 0.0
    for i in range(0, len(x)-1):
        sumXY = sumXY + x[i]*y[i]

    sumX = 0.0
    for i in range(0, len(x)-1):
        sumX = sumX + x[i]

    sumY = 0.0
    for i in range(0, len(x)-1):
        sumY = sumY + y[i]

    SP = sumXY - ((sumX*sumY)/len(x)-1)

    return (SP/(len(x)-1))

def convertTemp(F):
    C = (float(F)-32)* (5.0/9.0)
    return C

if __name__ == "__main__":

    local = open('Ashelle.txt', 'r')
    lines  = local.readlines()
```

Figure 2. Snapshot of CWB

5 SUMMARY

Cloud-based collaboration via familiar tools means that many of the initial obstacles to entry for collaboration, such as expensive initial investment in training and infrastructure for sharing are eliminated. Instead, it allows for scientists to keep using the tools they are already used to; only with add on transparent features for cloud-based sharing. Furthermore, cloud based collaboration service supports the availability and consistency of the shared data, algorithms, and analysis results among scientists.

We have presented the collaborative workbench (CWB) that augments researcher’s existing tools to provide various means of collaboration without adding any additional burden to the researchers. We envision that using CWB, scientists can be rolled into science algorithm development more easily and exposed to existing science algorithms and workflows. Accelerated science algorithm development will be possible due to the discovery and reuse of these algorithms and workflows within a collaborative environment. We have also demonstrated a science collaboration scenario using CWB.

6 ACKNOWLEDGEMENTS

This work is funded by NASA Grant and the authors would like to acknowledge Dr. Tsengdar Lee and Mike Seabloom for supporting this research. In addition, the authors would like to acknowledge the contributions of Thomas Harris and his IDL team in helping make IDL compatible with the CWB.

7 REFERENCES

Clayberg, E., and D. Rubel. 2008. Eclipse: Building Commercial-Quality Plug-ins: Addison Wesley Professional.

Lam, T., Hauser, N., Götz, A., Hathaway, P., Franceschini, F., Rayner, H., and Zhang, L. GumTree - An Integrated Scientific Experiment Environment. In International Conference on Neutron Scattering. Sydney, Australia, 2005.

Norris, J., Powell, M., Vona, M., Backes, P., Wick, J., "Mars Exploration Rover Operations with the Science Activity Planner," *in Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp.4618-4623, April, 2005.

Rushing, J., Ramachandran, R., Nair, U., Graves, S., Welch, R., Lin, A., ADaM: A Data Mining Toolkit for Scientists and Engineers, Computers & Geosciences, 2005, 31(5) p. 607-618.