APPLICATIONS OF SCIENTIFIC WORKFLOW TECHNOLOGY IN PROCESSING ATMOSPHERIC SCIENTIFIC DATA

Xiaoguang Lin*, Yuanchun Zhou, Jianhui Li

**Scientific Data Center, Computer Network Information Center, Chinese Academy of Sciences, Beijing, China*
Email: lxg@sdb.cnic.cn

ABSTRACT

Scientific Workflow is a flexible tool for accessing scientific data, and executing complex analysis on the retrieved data. E-Science oriented Scientific Workflow serves the scientists, and makes it much easier to analyze and manage the scientific data. Data-intensive scientific experiments with large-scale scientific data processing, in particular the atmospheric scientific experiments, are iterative scientific data processing procedures, including some fixed steps, such as data input, data pre-processing, data computation, data analysis, data visualization, result-dataset output, etc. This paper introduces a mechanism to compute and analyze atmospheric scientific data based on scientific workflow technology, and an implementation to provide atmospheric scientists and researchers an intuitive and easy-to-use web based toolkit.

Keywords: Scientific Workflow, Kepler, E-Science, Atmospheric Scientific Data Processing

1 INTRODUCTION

With the fast development of domain sciences and the continuous accumulation of knowledge, the scientific data is more large-scale, and the science processes are becoming increasingly complicated. A lot of data-intensive scientific experiments based on large-scale scientific data processing, in particular the atmospheric scientific data analysis processing as seen from the Figure 1, are iterative scientific data processing procedures, including some steps with different granularity, such as data input, data pre-processing, data computation, data analysis, data visualization, result-dataset output, etc. The only different things in each iterative processing are integrating different computational models and analyses, and invoking different process tools. How to reuse the components with computational models and analyses information to promote research efficiency is a serious problem to solve for the domain researchers. The scientific workflow technology is introduced in this paper to deal with this problem.

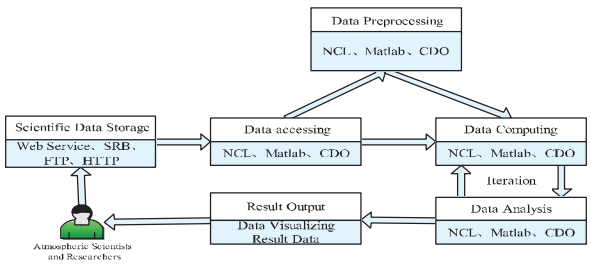

Figure1. Atmospheric scientific data processing diagram

This paper focuses on the scientific workflow technology and its application in atmospheric scientific data processing. The remainder of this paper is organized as follows. Section 2 introduces the scientific workflow

technology and the Kepler system, one of the mature scientific workflow systems in the international community at present. Section 3 provides a mechanism to process large-scale scientific data based on scientific workflow technology. This mechanism uses the Kepler system as the workflow engine, and integrates some mature tools of data mining, data integration, data analysis and data visualization. The implementation of this mechanism in atmospheric science domain provides atmospheric scientists and researchers an intuitive and easy-to-use web interface to compute and analyze large-scale scientific data with different computation models.

2 Scientific Workflow and the Kepler System

The Scientific workflow, a flexible tool for accessing scientific data, and executing complex analysis on the retrieved data, is becoming increasingly important as a unifying mechanism to combine scientific data management, analysis, simulation, and visualization tasks. Scientific workflow systems are problem-solving environments, supporting domain scientists and researchers in the creation and execution of scientific workflows (Ludascher, 2005).

Historically, business workflows can go back to office automation systems, and more recently gained momentum in the form of business process modeling and business process engineering (Bowers, 2005). Scientific Workflow uses the idea of business workflow to describe and control the science experiment and process execution. However, different from the business workflow focus on control-flow, scientific workflows is much more dataflow-oriented.

Scientific workflows exhibit particular traits, e.g., they can be data-intensive, compute-intensive, analysis-intensive, and visualization-intensive (Ludascher, 2005). Consequently, workflow steps can have very different granularities and may be implemented as shell scripts, web services, local application calls, or as complex sub-workflows. Each scientific workflow consists of analytical steps that may involve database access and querying, data analysis and mining, and intensive computations performed on high performance cluster computers (Bowers, 2005).

There are many mature academic scientific workflow systems in the international community, such as Kepler, Taverna, Triana, etc. With respect to their modeling paradigm and workflow execution models, these systems are closer to visual dataflow programming languages for scientific data and services. In particular, the Kepler system is the most popular one, for its providing a convenient program method with a special component called *actor* to extend specific scientific workflow system (Rygg, 2007).

Kepler is a cross discipline project that aims to simplify access to scientific data and the analysis of the retrieved data. The Kepler environment is built upon the PtolemyII platform, developed for modeling heterogeneous and concurrent systems and engineering applications, and the Kepler project has extended PtolemyII towards scientific workflows through adding support for web service invocations and access to Grid resources. Components to deal directly with the business logic in Kepler are objects called *actors* and the communication between actors and execution of a workflow is controlled by an object called *director*. In Kepler, users develop workflows by selecting appropriate *actors*, and joining them together to form the desired workflow. Actors have *input ports* and *output ports* that provide the communication interface to other *actors*. Each customized workflow model is comprised of a *director* and at least one *actor*. When workflow execution, the director controls the data flowing between *actors* and deploys the iteration of each *actor*. Taken together, workflows, actors, ports, connections, and directors represent the basic building blocks of actor-oriented modeling. For a domain science, the only thing for the programmers to do is to extend these interfaces to encapsulate domain scientific appropriate components (*actors*) (Kepler, Web Site).

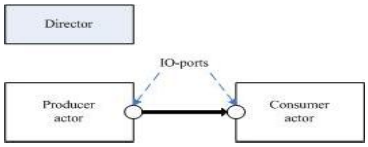

Figure1. Kepler model diagram

In Kepler/PtolemyII system, the customized scientific workflow model instances are stored in XML file format, which satisfies the Modeling Markup Language (MoML) XML schema (PtolemyII, Web Site).

3 Design and Implementation

3.1 Requirement

The aim of our work is to provide an intuitive web based workflow environment for atmospheric scientists and researchers, from accessing distributed atmospheric scientific data, executing analytical and computational steps iteratively according to the definite workflow model, to downloading the result dataset and display the visualization result. An overview of these requirements is presented in Table 1.

Table1. The requirements of the atmospheric scientific data processing

Purposes	To access distributed atmospheric scientific data
	To integrate some components to form a workflow intuitively
	To execute workflow and manage the execution
	To add new mathematic algorithm models dynamically
	To store and view workflows through web pages
Features	Workflow engine (Kepler): to execute the customized workflow by executing *actors* iteratively
	Model assembly: to form definite workflow model instances by assembling some components(*actors*) based on web
	Dynamic component addition: to add new mathematic algorithm models dynamically
	Workflow management: to store and view well customized workflow instances

Considering that atmospheric scientists and researchers lack sufficient computer technology, an intuitive web based tool should be provided. Web based tools do not need to be installed, and the researchers should only work on the browsers (like IE) to use the tools. So, all things should be implemented as BS (Browse-Server) architecture.

To integrate workflow engine and web front seamlessly, we should develop a middleware tool kit of Kepler, which is an open source project. The middleware could be used to list existing related components (*actors*) by redefined categories, add a new component (*actor*) according to the given algorithm model script, execute a submitted workflow instance and manage the execution.

The web front module should enable researchers a good user experience. Through it, users can assemble components to form a workflow instance, add a new component dynamically, submit execution of the instance and manage it, look over the distributed scientific data, and view the saved workflow instance.

3.2 Architecture

As analyzed in section 1, a data-intensive scientific experiment is really an iterative scientific data processing. The steps of the experiment based on large-scale data are relatively fixed, from data accessing, data input, data computing, and data analysis, to result dataset visualization display. The only different things of each experiment are different algorithm models for data analysis and computation. According to these, we present a mechanism to compute and analyze scientific data using scientific workflow based technology. The figure 2 is the common overview of the mechanism architecture.

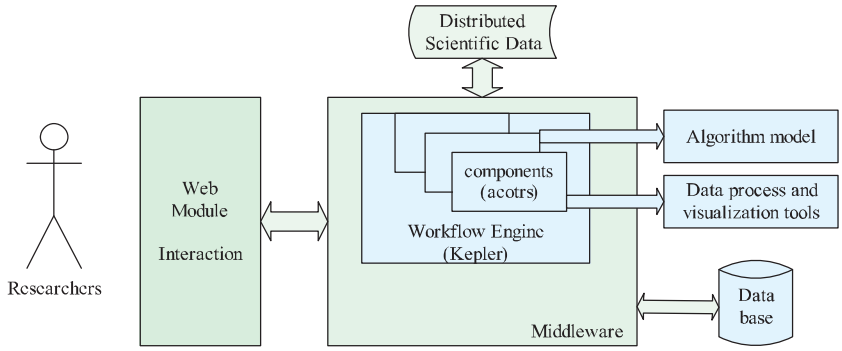

Figure2. The common architecture overview

The common architecture of the mechanism is comprised of three main layers: data resource layer, workflow engine and middleware layer, and web interaction layer. The data resource layer mainly implements storing and accessing the physical resources and the distributed resources on the internet; the workflow engine and middleware layer provides a steady scientific workflow engine and supports the communication between resources, the engine and the web; the web interaction layer provides users a well designed interface to assembling, viewing, executing, managing workflow instances.

We have succeeded in developing an atmospheric scientific data analysis and computational environment based on the common architecture using Java. From section 3.3 to 3.5, we introduce the layered implementation of the environment in detail.

3.3 Data resource layer

There are many referential physical resources in the atmospheric scientific data analysis and computational environment: large-scale local scientific data, algorithm models for steps of each experiment, software tools for data processing and visualization, and other files or users information.

Most atmospheric scientific data (like IPCC, NCAR) are large-scale, in particular, each size of the NCAR data files is at least a few hundreds million bytes. Under normal circumstances, researchers have to fetch the definite web sites to download the wanted data temporarily when the experiment needs it. Thus, a lot of research time and resources will be wasted, and the efficiency of the scientific research will be affected greatly. To deal with it, we (Scientific Data Center, CNIC, CAS) try to provide a massive storage environment with data mirrors, and users no longer need to download the remote data to local disk. What's more, we have provided a high-performance computational environment. On the other hand, researchers could upload their only scientific data to our storage environment to share it to others.

The algorithm model is one of the most important things in our atmospheric scientific data analysis and computation environment. In atmospheric science, there are too many algorithm models, from simple addition, subtraction, multiplication, division operations, to complicated statistical functions. They could be stored in two forms: the persistent format in database (using Mysql), and the format of *actors* in Kepler container. The middleware of this environment could transform the two formats each other when required. And through the web interaction and the middleware, user could add new algorithm models dynamically.

In view of generality, we take the NCL (NCAR Command Language), a free interpreted language designed specifically for scientific data processing and visualization, as the tool to process atmospheric scientific data and visualize the result data. The full NCL script includes 3 parts: file input and output, data analysis, and visualization. We could use one simple command line to invoke the NCL easily in Linux OS. Via the middleware, the invoking of

the NCL is encapsulated into a special *actor* of Kepler.

Some other important data are stored in the database, such as categories of *actors*, detailed information of *actors*, personal information of users and so on. The middleware provides a set of database processing methods to store and access these data.

3.4 Workflow engine and middleware layer

The Kepler system is used as the workflow engine in our environment. We use web front module instead of Kepler's own intuitive GUI (inherited from PtolemyII) to design and execute workflow instances. So, Kepler is a background running program in our environment. Kepler sees to receive customized workflow instances (XML file in MoML schema), resolve the definition of the flow, instantiate the flow, control the instance execution, schedule tasks, and do other relative things. Once the workflow execution starts, *actors* are "fired" and processed iteratively in Kepler.

We have developed a middleware to integrate data resources, Kepler and the web front seamlessly. The middleware provides some important functions as follows:

1) Manage the classification of actors.

As presented above, actors (with the algorithm models information) have two formats. The middleware supports the transformation between the two formats. In database, actors are classified to be a tree by their own functions, like Figure3.

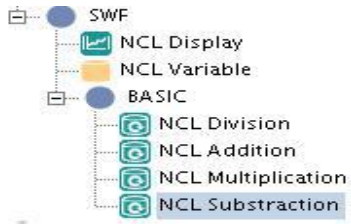

Figure3. The tree structure of actors

2) Process the execution of the workflow instances.

The middleware builds a uniform interface for users to shield the details of Kepler. Table2 shows main states of the workflow engine. When workflow instances are executed, each has an execution listener to monitor states of the engine during the whole execution phase.

Table2. Main states of managing the engine

State	Description
IDLE	Indicator that there is no currently active execution.
INITIALIZING	Indicator that the execution is in the initialize phase.
RESOLVING_TYPES	Indicator that type resolution is being done.
ITERATING	Indicator that the execution is in iteration.
PAUSED	Indicator that the execution is paused.
EXITING	Indicator that the execution is in the wrapup phase and about to exit.

The middleware also provides a unifying interface to control the execution, like starting the execution, pausing it, canceling it, or restarting it.

3) Add a new NCL script model dynamically.

The algorithm models with NCL script are encapsulated into *actors* in our environment. And the middleware

supports that scientists and researchers add new algorithm models by making up new actors dynamically.
The middleware defines a standard to the new algorithm model with NCL script, which specifies the formats of input variables, output variables, parameters, and others. After users upload the script through web, the middleware encapsulates it automatically to a java source file satisfied the *actor* format. And then, the middleware invokes the tool class of java to compile the source file dynamically to be a .class file. At the same time, detailed information of this *actor* is stored into database.

3.5 Web interaction layer

The Kepler system provides an intuitive GUI (inherited from PtolemyII) to design and execute workflow instances. But, considering users' convenience and our enormous computational and storage resources, we develop a web based GUI to enhance the user experience, which is one of the best features of our environment.
This web interaction layer is a traditional MVC (Model-View-Controller) application, built on Spring, one of the most popular lightweight J2EE frameworks. This layer not only provides a easy-to-use web interface to design, execute, manage, and view workflow instances, but gives users a one-stop service web environment for source data, result data, user information management, and other services.
The design and view module is developed based on VML (Vector Markup Language), which is a markup script language supporting the IE browser, and AJAX (Asynchronous JavaScript and XML) technology. Users could do cut, copy, paste, drag, alignment, and other visual editing operations on the graphic elements, representing *actors* from the middleware. A customized workflow instance is stored as a XML file in MoML schema in this module. And then, the instance will be submitted to the middleware, where the Kepler engine will be invoked to resolve and execute the instance. After Kepler finishes the execution, the result (dataset or image) will be exhibited to users in this module.
This layer also provides users interfaces to add new algorithm models and manage them. Indeed, the logic of the addition and management is realized by the middleware.

3.6 Example

Some result images are given in this part. Figure 4 shows a simple atmospheric scientific workflow instance chart. Three different variables from the input .nc files are chosen in the beginning, then several computational steps are iteratively processed, and the result data set is stored and visualized at last. Figure 5 shows the final visualization result image.

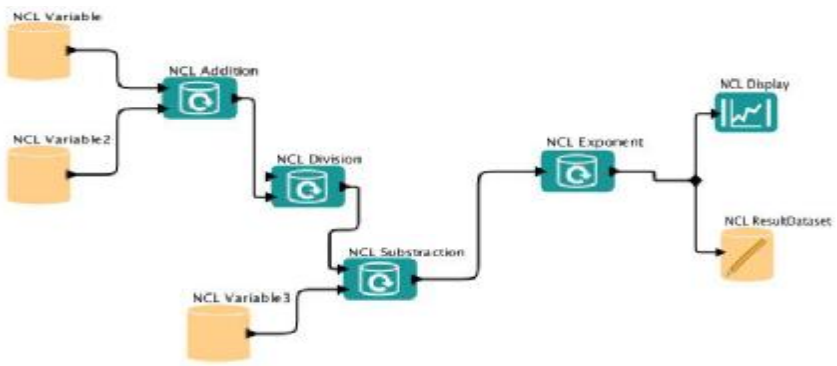

Figure4. A simple flow chart

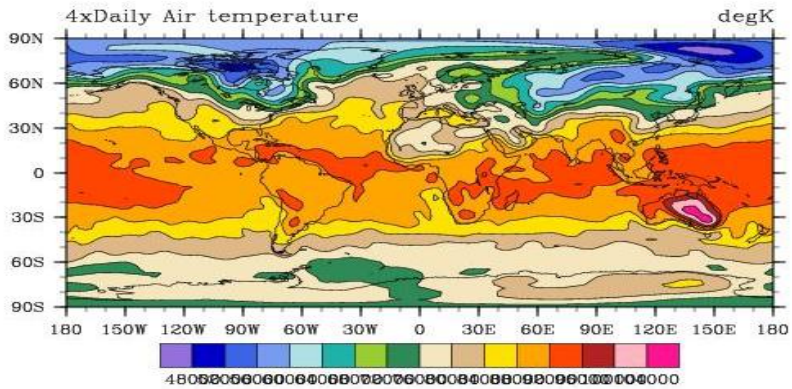

Figure5. A final result image

4 Conclusion

The atmospheric scientific data computation and analysis environment based on the scientific workflow technology uses the mass storage and high-performance computation environment, supports users to add algorithm models dynamically, and achieves the reuse of models. Consequently, the research capabilities and efficiencies of atmospheric scientists and researchers are greatly promoted.

The environment is being developed at present, and will be extended to many other domain sciences in the future.

5 ACKNOWLEDGEMENTS

We would like to thank Professor Gang Hang, Doctor Pengfei Wang, and Xia Qu from Institute of Atmospheric Physics, Chinese Academy of Sciences, for their open idea, discussion, cooperation, and contribution. The work reported in this paper is supported by the Youth Foundation of Computer Network Information Center, Chinese Academy of Sciences under Grant No. O714041701.

6 REFERENCES

Ludascher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Jones, E., Lee, E., Tao, J., Zhao, & Y., (2005) Scientific Workflow Management and the Kepler System. *Concurrency and Computation: Practice & Experience 18*(10), 1039-1065.

Bowers, S., & Ludäscher, B., (2005) Actor-Oriented Design of Scientific Workflows. *Conceptual Modeling-Er 2005*: Springer Berlin / Heidelberg, Volume 3716/2005, 369-384.

Kepler Project [EB/OL]. From the WWW, http://kepler-project.org/

PtolemyII Project. From the WWW, http://ptolemy.berkeley.edu/ptolemyII/

Wang, L., Peng, Z., Luo, M., Ji, W., & Huang, Z., (2006) A Scientific Workflow Framework Integrated with Object Deputy Model for Data Provenance. *Advances in Web-Age Information Management*: Springer Berlin / Heidelberg, Volume 4016/2006, 596-580.

Liu, X., Dou, W., Chen, J., Fan S., Cheung, S., & Cai, S., (2007) On design, verification, and dynamic modification of the problem-based scientific workflow model. *Simulation Modeling Practice and Theory*, 1068-1088.

Rygg, A., Roe, P., & Wong, O., (2006) GPFlow: An Intuitive Environment for Web Based Scientific Workflow. *Grid and Cooperative Computing Workshops, 2006. GCCW '06. Fifth International Conference*, Changsha, China