Journal of Theoretical and Applied Information Technology
31st May 2013. Vol. 51 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN EFFICIENT EARTH SYSTEM SIMULATION FRAMEWORK BASED ON CCA

[1]SHANSHAN LI, [2]XIAOLING SUN, [3]FUMING CHEN
[1,2,3]Asstt Prof., Department of Disaster Information Engineering, Institute of Disaster Prevention Science and Technology, Hebei, China
E-mail: [1] liss@cidp.edu.cn, [2] sxlliqi@163.com, [3]ds_cfm@sina.com

ABSTRACT

Muti-physics coupling technologies could achieve a more real simulation for Earth system. The complexity of such simulation is a difficult problem. This paper presents an efficient Earth system simulation framework based on CCA (Common Component Architecture) which is applied in large scale scientific computing put forward recently by America. Based on such component schema, this framework could better solve the complexity and provide a "plug-and-play" simulation environment. It could integrate the distributed high performance computing resources, distributed and heterogeneous physical model resources, and finally offer a transparent science research platform for the earth science researchers, which could provide construction, management, analysis service of the muti-physics coupling simulation. A preliminary prototype framework platform was developed. A CME simulation results presented here validate the framework.

Keywords: *Muti-physics, Simulation, Framework, Earth System*

1. INTRODUCTION

Solid Earth system, Earth surface system, sun-Earth system are the three subsystems which constitute a complete Earth system. In this Earth system, different physical domains interact each other through energy transfer and material movement, which they permeated into each other, and jointly promote the occurrence and development of a physical process. Such as mantle convection will cause changes in the Earth's crust, resulting in folds, faults, volcanic eruption, magma intrusion, earthquakes and other phenomena. Solar storms can throw billions of tons of charged particles into the space a few hundred kilometers per second. The erupted X-ray and ultraviolet radiation will do around the earth's magnetic field, making the atmospheric temperature rise. Therefore, the Earth system is a complex multi-physics coupling system. The modeling, simulation and forecasting for this complex system present a huge challenge to the scientists.

As a result, researchers began to use the framework-based technology and component-based software technology to solve the complexity of the muti-physics coupling simulation of the earth system. Currently, there are three well-known frameworks develop by the United States: Space Weather Modeling Framework SWMF [1], Earth System Modeling Framework ESMF [2] , Solid Earth System Modeling Framework GeoFramework [3]. However, they are still having some shortcomings:

(1) Operating mode and performance issues. Each of these frameworks is run on a high-performance computer. It needs to physically import a large number of physical source code and run-time environment into this single high-performance computer. It's not only takes enormous efforts and time, but also the owner of the physic model may be unwilling to open their own source code. In addition, a single high-performance computing power is no longer able to meet the needs of a large number of model's computing.

(2) Physical model development language interoperability problems. Different physical models may use different development languages such as Fortran 77/90/95, Python, C, C + +, Java and so on. Therefore, the interoperability between multiple different languages is necessary for the physical models interaction. However, these three frameworks all have some limitations to development language of the physical model: SWMF requires it must be developed using Fortran 77/90; ESMF requires it must be developed using C

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

/ Fortran 90; GeoFramework requires it must be developed using C / C + + / Python/Fortran77. These limitations restrict the interaction between models.

(3) Framework service issues. These three frameworks are all focus on how to achieve multi-physics model coupling, not considered for researchers to provide a friendly, safe and stable performance, interoperable problem-solving environment.

(4) Interoperability issues between the frameworks. As the three frameworks developed by different development organizations, the framework components were used in different packaging mechanism and operation implementation mechanism. Therefore, these frameworks cannot interoperate between each other, and eventually lead the entire earth system simulation cannot achieve.

Based on above issues, this paper applies the large-scale science component specifications - CCA [4-7] (Common Component Architecture) which is proposed by the United States recently to build an efficient Earth system simulation framework. This framework can solve the above inadequacies in existing researches, effective implement the modeling and simulation of the Earth system.

2. FRAMEWORK ARCHITECTURE

CCA is a component specification for large-scale high-performance scientific computing proposed by America's six laboratories and two universities in 1998. It defines two kinds of ports, service provide port and use port .This two kinds of ports build "service provide and use" schema, and achieve reciprocal relationship between scientific computing components. CCA specification includes two entities: components and frameworks. Components can insert into the framework flexibly through defined port. CCA uses the scientific interface definition language SIDL to describe the interface. It is similar to CORBA IDL. It maintains the neutrality of the programming language, but it extends IDL in parallel mechanism, reflection, object-oriented semantic definition, and scientific data types, such as multidimensional arrays, complex numbers and so on.

The framework this paper proposes follows the CCA specification, adopts service-oriented and peer to peer mode. (1) Service-oriented: different physical model developers who are in their own laboratory at different place could deploy the model in the local high-performance computers, publish model services through the services provide port, while use the services other models provide through use port. (2) Peer to peer mode: the physical model of different physical domains, as well as models from different disciplines, by adding a small amount of interface specifications can be encapsulated as CCA components. Different components interact through service. They are completely equal, there is no primary or secondary.

This framework runs on a distributed computing environment. Figure 1 shows the framework architecture. The structure is divided into four layers: a basic framework layer for coupling interaction, framework service component layer, physical model components layer and GUI graphical user interface layer.

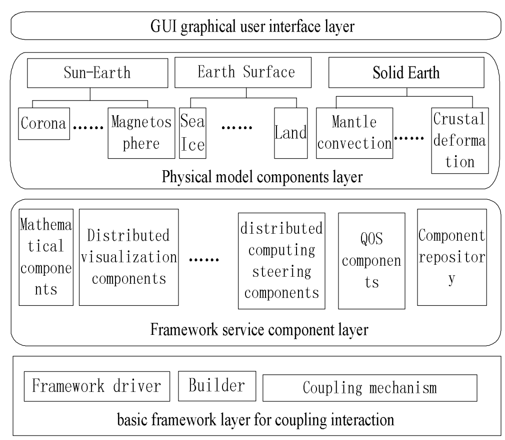

Fig.1: Earth System Simulation Framework Architecture

3. BASIC FRAMEWORK LAYER FOR COUPLING INTERACTION

Basic framework layer for coupling interaction bases on top of distributed computing resources layer provides container for the upper components and shield the underlying complexity and heterogeneity of distributed computing resources. It includes three core components: driver, builder and coupling mechanism. The driver is responsible for startup and shutdown of framework. Builder is responsible for managing components and applications. Coupling mechanism is responsible for remote interaction coupling and communications of model components. We propose an efficient multi-physics coupling interaction mechanism PRMI + + (Parallel Remote Method Invocation + +) which could automatically realize grid re-mapping and parallel distribution remapping

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

of the data at the time when the physical model components are coupling through parallel remote method invocation. It realizes four processes, as shown in Figure 2.

(1)Data description and registration: Component developers need to descript the component's physics data and parallel distribution data when they publish it. Based on those data, the system could generate the mesh descriptor GridDescriptor and data distribution descriptor DAD to achieve the data description. The descriptor is not only very flexible, and can be stored in the memory of each process comprised. This will not occupy a lot of memory space. It's an ideal way to describe the data. When data begin to transmit, the data sender component and data receiver component will register their GridDescriptor and DAD to the framework respectively, making every progress of the data sender are informed of the data information on all the processes of the receiving, and then to make decisions;

(2)Grid re-mapping: Based on the two sides' GridDescriptor, we designed and implemented a grid converter adopting an object-oriented framework Overture [8] which is developed by the U.S. California Lawrence Livermore National Laboratory LLNL. It can automatically convert the data grid when sending data;

(3)Parallel distribution remapping of the data: Using the Hilbert space filling curve and the interval tree, we design and achieve an efficient parallel distribution of re-mapping mechanism, which can automatically calculate the parallel remapping scheduling policy based on two sides' DAD. The principle is that, as the called component is registering its data parallel distributor, according to its DAD, generates one dimensional data indexing space based on Hilbert and detects the overlap space on the interval tree of the caller component. The detected overlap intervals indicate that these two processes overlap on this data block. The core Algorithm of detection is as follows:

a. *if (T≠NULL){ u ← root[T]}*
else end;

b. *if (l ≤ val(u) ≤ r)*
{ R ← (max(l, u. l_δ*),min(r, u.* r_δ*));*
T ← u.leftchild, go to a;
T ← u.rightchild, go to a;
}

c. *if (r < val(u)){*
if(r > u. l_δ*){*
R ← (u. l_δ*, r); }*
T ← u.leftchild, go to a; }

d. *if (r > val(u)){*
if(r < u. r_δ*){*
R ← (l, u. r_δ*); }*
T ← u.rightchild, go to a; }

(4)Data transfer: Follow the remapping parallel scheduling policy, the data sender directly send the overlap data block to corresponding process of the receiver, which greatly improves efficiency.

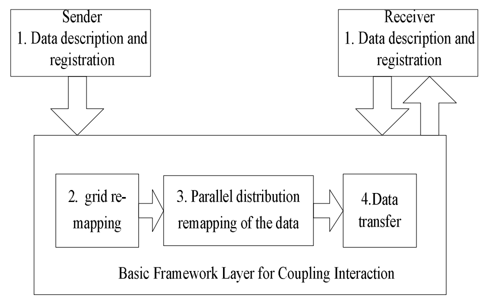

Fig.2: Four Processes of multi-physics coupling interaction mechanism PRMI++

4. FRAMEWORK SERVICE COMPONENT LAYER

(1)Framework service components: provide multidisciplinary services for the upper physical model components, including the following types of services provided by the component:

(2)Distributed visualization component: could provide real-time online visualization service for distributed multi-physical applications simulation, allowing users to get the most intuitive feelings of the results of running;

(3)Mathematical components: are produced by encapsulating mathematical standard libraries according to CCA. It provide services for the development of application, including the numerical components and parallel code development components;

(4)Distributed computing steering component: provides researchers with computing steering services, enabling them in real time of multi-physics simulations could modify the model, boundary conditions, initial conditions and other parameters to test results. They could perform a series of "what if" test which enable they get the most intuitive understanding of the parameters.

(5)Performance monitoring and optimization component: mainly provides analysis and

Journal of Theoretical and Applied Information Technology
31st May 2013. Vol. 51 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

optimization service for the digital software. Through the "customers - consultation - agency" model, component passes its special service requests and QOS constraints metadata to the proxy for dynamic negotiation, and finally instant a component which meets the application-level quality.

(6)Distributed fault-tolerant service component: has the ability to ensure that the entire system when an error occurs, automatically roll back to a previous safe state. This component is based on a checkpoints and rollback recovery strategy policy which we design for the distributed parallel environment. This policy supports the two way parallelism that is the component parallel mechanism PVM or MPI, and the parallel communication mechanism PRMI++. Based on this strategy, distributed fault-tolerant service component is mainly developed by using collaborative models and message count collaboration algorithm.

(7)Component Repository: is a kind of knowledge-based repository which constitutes a "component market". It is an important part for achieving shared components. It store registered components and the main standard interface definition information of components, making services components provide can be found when other components call, or direct inquiries by the user. At the same time, it hides the physic model source code developed by various organizations, making the interaction between components can be transparent. It provides registration service, query service, version management service and other services, while providing the user interface to the user. The core implementation include fuzzy search algorithm, web-based interface of the language interoperability tools, the component type information repository, and other query interface for the software tools.

(8)Language interoperability component: is produced to solve the language interoperability problem between different scientific computing model components developed by different scientific programming language. Most interoperability tools are developed just for two programming languages mapping which achieve point to point languages mapping, such as JNI implements the Java and C / C++ mapping, SWIG achieve a python to C / C++ mapping. Although this point to point mode achieve certain interoperability between two languages, but cannot solve the language interoperability problem between multiple languages. We use Babel [9] tool to achieve this problem. It uses the SIDL to generate glue code between the different languages, based on this to achieve interoperability of various development languages, and arbitrary two languages' mapping is bi-directional.

5. PHYSICAL MODEL COMPONENTS LAYER

Physical model components come from the physical model from different physical domains encapsulated by CCA specification, including the Sun-Earth space physics model components, model components of Earth's surface, physics model components of the solid Earth. This framework allows researchers to do just a small amount of work to generate a new component or encapsulate legacy model code, as shown in Figure 3. First, physic model researchers use SIDL to define components and components' interfaces. Second, they use Babel tool to compile it so that map it into object code. Third, Object code and source code are compiled and linked locally by the language complier, and then create an executable physical model component.

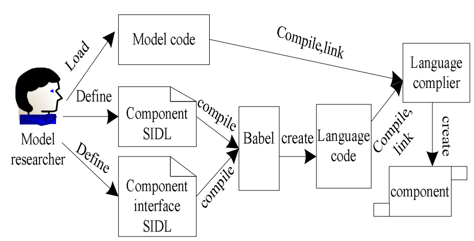

Fig.3: Component Encapsulate Process

6. GUI GRAPHICAL USER INTERFACE LAYER

GUI graphical user interface encapsulates the interface with components, basic framework layer for coupling interaction, and user. By using it, user can instance a remote component, assemble and monitor an application easily just by click, drag and drop the mouse. Figure 4 shows the GUI Mechanism.

Journal of Theoretical and Applied Information Technology
31st May 2013. Vol. 51 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

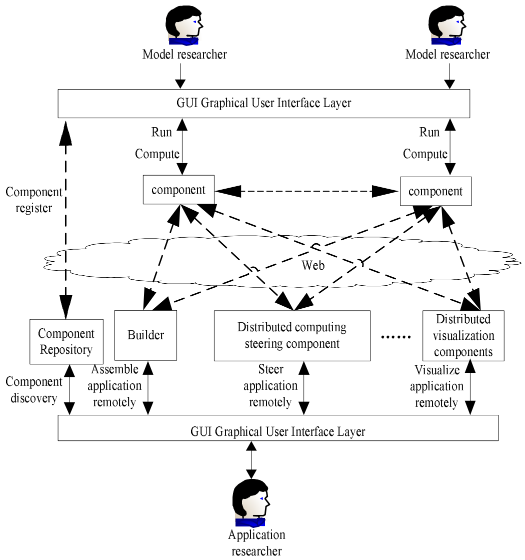

Fig.4: GUI Mechanism

7. EXPERIMENT

We use two models to test: coronal mass ejection model CME (Coronal Mass Ejection) model and a visual model which is based on VTK, Python. The CME is running on a parallel blade computer with 4 blades, the visual model running on a remote PC with 2 CPU. User can operate this application through the GUI, as shown in Figure 5. GUI includes four areas: menu bar, tool bar, component resources area, the assembly editing area. First, User selects the required CME component and visualization component from the component resources area in the left by clicks mouse, and drags it to the assembly editing area. Second, User clicks on the ports of the components to build the connection. After assembling the application, the user can click on the green Go port or the start button on the toolbar to run the entire application. During the simulation run, the GUI brings up a remote visual interface for real-time online visualization; two remote parallel components are coupled, and ultimately realize the numerical simulation of the CME, as shown in Figure 6 (the orange sphere represents Earth, the yellow curve represents magnetic field lines, the gray straight segment represents pole axis).

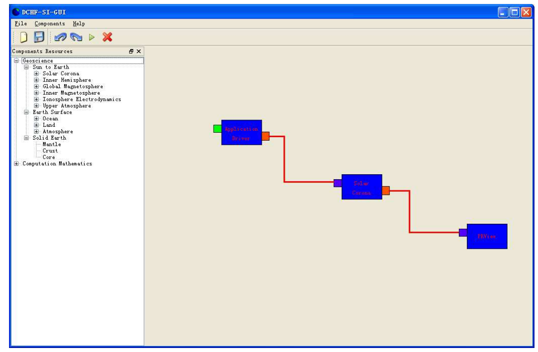

Fig.5: Graphic User Interface

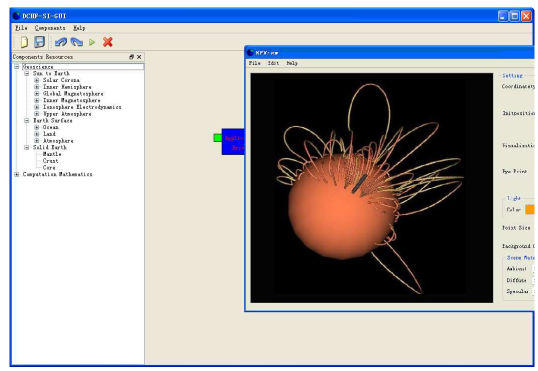

Fig.6: Numerical Simulation of the CME

8. CONCLUSION

The complex Earth system presents the scientists with multiple challenges in muti-physics coupling simulation. This paper proposes an efficient muti-physics coupling simulation framework for Earth system, which leverage the efficacy of the CCA component model to manage the complexity of their distributed simulations, and provide scientists with a transparent research platform on which they can easily develop, operate, manage, visualize, steer and analyze a distributed muti-physics coupling simulation.

9. ACKNOWLEDGEMENT

This work was supported by the China Earthquake Administration Teacher's Fund (Grant NO. 20110112), Special Fund of Fundamental Scientific Research Business Expense for Higher School of Central Government (Projects for young teachers, Grant NO.ZY20110211).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFRENCES:

[1] Toth G, Sokolov I V, Gombosi T I, et al. "Space Weather Modeling Framework: A new tool for the space science community", *Journal of Geophysical Research*, Vol.110, No.A12, 2005, pp. 226-226.

[2] Earth System Modeling Framework Homepage. http://www.esmf.ucar.edu/.

[3] Geoframework Homepage. http://www.geoframework.org.

[4] Bernholdt D, Allan B, Armstrong R, "A Component Architecture for High Performance Scientific Computing", *International Journal of High Performance Computing Applications*, Vol.20, No.2, 2006, pp. 163-202

[5] The Common Component Architecture Forum. http://www. cca -forum.org/.

[6] Armstrong R, Gannon D, Geist, A, "Toward a Common Component Architecture for high performance scientific computing", *Proceedings of the Eighth IEEE International Symposium on High Performance Distributed Computing*, Redondo Beach, August 3-6,1999, pp. 115-124.

[7] Epperly T, Kohn S, Kumfert G, "Component technology for high-performance scientific simulation software", *Proceedings of the International Federation for Information Processing Working Conference on Software Architectures for Scientific Computing*, Kluwer, Ottowa, Ontario, Canada, June 11-13 2000, pp. 69–87.

[8] Oveture Homepage: https://computation.llnl.gov/casc/Overture/.

[9] Babel 1.0 Release Criteria: AWorking Document, http://www.llnl.gov/CASC/components/docs/BabelReleaseCriteria.pdf, 2003

AUTHOR PROFILES:

Dr. Shanshan Li received the degree in Cartography & Geographic Information Engineering from China University of Geosciences for Doctoral Degree, in 2009. She is a research student of China University of Geosciences. Currently, she is an Assistant Professor at Institute of Disaster Prevention. Her interests are computer simulation in Earth science and software engineering.

Xiaoling Sun received the degree in Information Security from Shandong University for Master Degree, in 2009. She is a research student of Shandong University. Currently, he is a Lecturer at Institute of Disaster Prevention. Her interests are in software engineering and information security.

Dr. Fuming Chen received the degree in Cartography & Geographic Information Engineering from China University of Geosciences for Doctoral Degree, in 2009. He is a research student of China University of Geosciences. Currently, he is an Assistant Professor at Institute of Disaster Prevention. His research interests include distributed computing and embedded system.