

Real Time Biomedical Data Streaming Platform (RIMES): a data-intensive virtual environment

Kejun Dong, Kai Nan
Computer Network Information Center
Chinese Academy of Sciences
China
e-mail: kevin, nankai@cnic.ac.cn

Sameer Tilak, Cindy Zheng, Dong Xu, Jurgen Schulze, Peter Arzberger, Wilfred Li
University of California, San Diego
USA
e-mail: sameer, zhengc, dxu, jschulze, parzberg, wilfred@ucsd.edu

Abstract - In this paper we present Real time bioMEDical data Streaming platform (RIMES), a data-intensive virtual environment, under joint development by Computer Network Information Center and University of California, San Diego. RIMES extends scientists' desktops to biomedical grid computation resources and mass storage systems, by means of streaming biomedical data through internet-based Data Turbine cloud. Researchers may conduct activities related to job submission, data processing and real time visualization within the environment. RIMES may also be suitable to other data-intensive areas such as geosciences and ecoinformatics.

Keywords – biomedical research; data streaming; grid computing; e-science; real time visualization; Data Turbine;

I. INTRODUCTION

Today researchers are increasingly dealing with peta- or exa-scale data generated by scientific instruments and computer simulations, often through grid and cloud computing, which make simulations much easier and faster. It is still challenging to improve the ability to store, transfer and visualize the data efficiently.

For biomedical sciences, researchers can easily manage simulations which are submitted to scheduler-enabled grid or cloud environments as parallel jobs [1] [2]. But it is still a challenge to extend their ability to manage simulation data, such as data acquisition, data transfer and data visualization. In order to improve the situation, a joint group is setup by Collaboration Environment Research Center, Computer Network Information Center, Chinese Academy of Science, China and National Biomedical Computation Resource, University of California, San Diego, USA, focusing on the Real Time Biomedical Data Streaming Platform (RIMES), a data-intensive virtual environment, which enables real time visualization for the on-going experiment data.

In the remainder of the paper we outline the key capabilities and carry out the design and implementation of RIMES. In section II we introduce biomedical data simulations and requirements for high performance data streaming. In section III we elaborate the architecture of RIMES, as well as the design and implementation. In section IV we present how to use the JMOL plug-in platform as a visualization environment based on RIMES API. In section V we give performance tests and summarize the paper in

section VI. Finally, we list related works in section VII and discuss limitations and future plans in section VIII.

II. BIOMEDICAL SIMULATION DATA

With advanced grid computing and cloud computing, researchers submit jobs from different portals and tools, e.g. an OPAL2 Dashboard, a simple GUI provided by the Opal Toolkit [3] for wrapping scientific applications as web services. Routinely, simulation experiments run for a long time and generate lots of data as results. Researchers can download results from the cloud storage to user desktops or servers for visualization. For researchers, there is a long wait time from job submission to end result visualization.

Actually, biomedical researchers know that most of job results are generated dynamically and the intermediate results are already of interest before job completion [4]. These intermediate results may help decide whether to continue or terminate the simulation. It becomes necessary to acquire and process the data once generated, and to simplify the procedure for the biomedical data transmission and visualization.

Here we introduce the different file formats in molecular dynamics simulations, frequently conducted in the avian flu grid project of PRAGMA and other computer aided drug discovery projects.

The files encountered frequently are of the PDB (Protein Data Bank) [5] and DCD [6] formats. The PDB file format is a textual file format describing the three dimensional structures of molecules held in the Protein Data Bank while DCD is binary trajectory files which is identical to the CHARMM-DCD format [7].

As Figure 1 describes, either PDB data or DCD data is composed of one or multiple frames, which makes it possible to acquire and process partial result data.

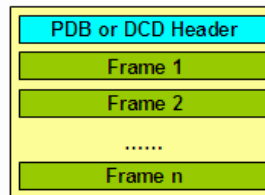


Figure 1. PDB and DCD data format

NAMD (NANoscale Molecular Dynamics) [8] is a parallel molecular dynamics code for large biomolecular systems and it is a typical simulation program widely used by biomedical researchers to simulate molecular dynamics trajectories with different force fields [9]. Figure 2 describes the NAMD-based data workflow. After NAMD job submission, output frames will be appended to output DCD files every N frame intervals.

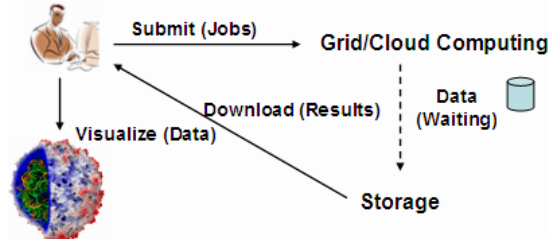


Figure 2. NAMD-based data workflow

In the NAMD case, researchers hope to get the information of jobs that they submit during the NAMD simulations, including job status and partial results. It is necessary to setup such a data-intensive virtual environment capable of using cluster, grid or cloud computing resources, and visualization resources such as desktop clients, tile display walls or StarCAVE.

III. REAL TIME BIOMEDICAL DATA STREAMING

In this section, we discuss a solution to monitor and visualize the on-going dynamical experimental results, usually PDB or DCD files from molecular dynamics trajectories simulations generated by the scientific grid computing software, such as NAMD.

In the following paragraph, we first introduce the fundamentals for the research, and then describe a use case of RIMES, and finally we come to the architecture of the platform. Frame-based characteristic makes it possible to streaming the partial data frames from simulation servers to Data Turbine Cloud by frames. Thus researchers can submit jobs and analyze data frames once it is generated.

A. Fundamentals

Data Turbine stands for Ring Buffer Network Bus Data Turbine Streaming data middleware [10], developed by UCSD, which provides an excellent basis for developing robust streaming data middleware and is the basic middleware in RIMES.

Duckling [11], which stands for Collaboration Environment Core Toolkit, developed by CNIC, is a software suite with a JSR 168 and JSR 286 compliant container for portlets, and virtual organization support. It helps to organize collaboration behavior and realize resource sharing and innovation.

In RIMES, Data Turbine is used as a data streaming framework for biomedical data, and Duckling as a portlet container to offer a web-based easy-to-use portal.

B. Use Case

Figure 3 describes user model for typical biomedical simulation experiments - NAMD simulations, in conjunction with RIMES. Usually researchers download results after job completion. In our case, we setup a dataturbine-based RIMES platform, which connects cloud storage to different data repositories. Researchers can easily get data wanted by frames, either from databases or from on-going simulation data cloud storage, called Data on Demand Service.

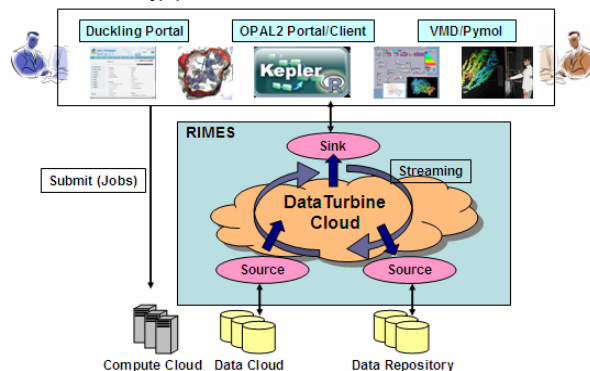


Figure 3. User model of NAMD job simulations

When biomedical data in data repository is updated, appended frames will be streamed to requested RIMES-client or plug-ins, which will get the data synchronized. In conclusion, users can extend user desk to biomedical grid computing and mass storage systems by means of real time biomedical data streaming.

C. Architecture

Figure 4 gives the architecture of RIMES. In the top layer of RIMES architecture, there are several interfaces ready for invocation, including the RIMES-client and other plug-ins, i.e. for JMol [12], RDV [13], StarCAVE [14] [15] and COVISE [16]. If needed, researchers can develop plug-ins to more software based on the RIMES API. The key layers are the API layer and Data Turbine layer, as well as Data Monitor Engine.

The Data Turbine layer is the data streaming core middleware, which consists of several Data Turbine servers, named as a SaaS [17] Data Turbine Cloud. This is the fundamental infrastructure in this environment. The Data Turbine Cloud connects streaming servers around WAN with each other and offers reliable and efficient capability to transfer biomedical frames from one side to another.

Data Monitor Engine is the distributed metadata library of all biomedical data distributed around the repositories or cloud storage. Each repository or cloud storage will create one Data Monitor Engine database once the RIMES daemon is invoked. Data Monitor Engine is not only responsible for data location, but also for data replication of the biomedical data.

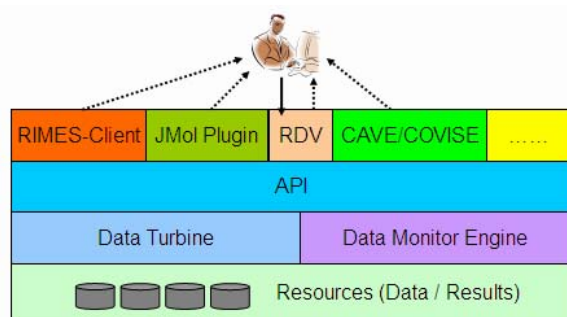


Figure 4. Architecture

D. Channel Map

In Data Turbine, all data is organized in "channel maps". RIMES clients manipulate channel maps as a way to submit or request data, named as sources or sinks respectively.

A source client builds a channel map consisting of one or more named channels. For each channel it provides data of a specified type and quantity. It also specifies a timestamp for the channel map as a whole or for the various pieces (channels and data) separately. After being so built, the channel map is sent from the source client to the server. This process can be repeated, adding new channels or new data to existing channels.

A sink client builds a channel map in order to request data. The channel map consists of named channels and timestamps, which is sent to the server as a request. The response to this request is another channel map, this time with the data filled in for the various channels.

Data Channel Map (DCM) is the most important in RIMES and it's responsible for all reliable data exchange around the Data Turbine Cloud. All metadata about Data Channel Maps will be transmitted by Meta Channel Map (MCM). Protocol Channel Map (PCM) also is needed to communication between repositories or cloud storage.

E. Protocol

Since Data Monitor Engine is distributed around the repositories or cloud storage, communication protocol is needed to exchange the metadata, request and response messages for data subscription.

Data Turbine is used as a reliable message passing fundamental infrastructure, by Protocol Channel Map (PCM), a channel map for communication protocol.

The protocol message is defined as following:

```
+-----+-----+-----+-----+
| SN | KIND | OPER | CONTENT |
+-----+-----+-----+-----+
```

- SN is the serial number of the message. It is unique.
- KIND stands for the kind of messages, which could be C (Channel), channel operation for streaming; or S (Search), information for metadata or others; or F (File), request for data streaming; or R (Response), response for the request message.

- OPER the sub-operation for each kind of KIND. See the definition in TABLE I.
- CONTENT is the content for the message.

TABLE I. OPER EXPRESSION

KIND	OPER	Expression
C	ADD	Adding channel
	DEL	Deleting channel
S	META	Metadata searching
F	NUM	Frame request by specifying frame number, unrecognized number as zero
	META	Metadata request
	HEAD	Header request
R	SN	Response to the message with SN

IV. IMPLEMENTATION AND JMOL VISUALIZATION

Many biomedical users and researchers are interested in visualizations of experimental data, such as molecule dynamics trajectories. The RIMES client and API make it easy to conduct remote molecule trajectory visualization on Tiled Display Walls (TDW) [18], or web-based portals.

The command-line RIMES client is developed to support the basic biomedical file operation through RIMES, including real time data storage, file search, and animated gif file generation through a Java-based RIMES API. Figure 5 is an animated gif output by frames which is fetched from an on-going simulation. Any new frames may be appended to the file once received.

Using the command-line RIMES client, researchers are able to fetch the partial data, perform data processing steps and visualize results through plug-ins for a visualization client or through portlets using the RIMES API.

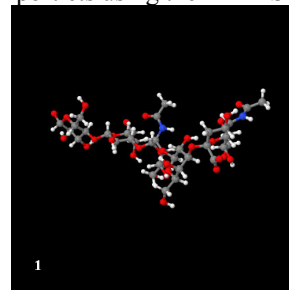


Figure 5. Animated gif output by frames

We have extended the JMol software, a widely used and open-source Java viewer for chemical structures in 3D, to visualize biomedical molecule trajectory frames. The RIMES-enabled JMolApplet has been integrated into a Duckling portal, as a web-based 3D visualization, providing a data-intensive virtual environment for biomedical research. Job submission module also is enabled in Duckling portal as a plug-in to submit job to the back-end clusters.

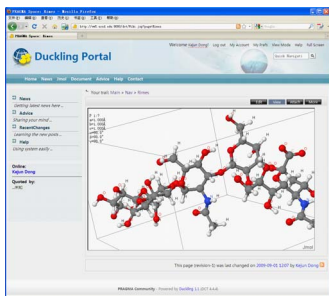


Figure 6. Sink-enabled web-based 3D visualization on Duckling portal

With the JMol plug-in, researchers can get results in real time from the on-going experiments on grid or cloud compute resources, and browse frames through operations such as “previous frame”, “next frame”, “first frame” and “last frame”. Researchers can easily locate the frame that they want to review and utilize all molecular modifications that JMol supports. Furthermore, the “play” operation is also available to play received frames one by one which are similar to animated gif files that we can generate by RIMES-client.

V. TESTING AND DEPLOYMENT

A. Performance Testing

Frame-based streaming performance is the key indicator and the frame streaming data transfer rate can be configured in the RIMES configuration within the maximum limit in order to decrease the load of servers. It is tested that the performance of streaming is acceptable without frame package loss.

In our case, we have deployed two Data Turbine servers and RIMES daemons at UCSD (vm5.ucsd.edu) and CNIC (dt.escience.cn) for biomedical data streaming. Additional information can be downloaded from the link <http://pragma.escience.cn>.

First, we use Iperf [19] to value the network performance between UCSD and CNIC servers. Table II shows the transfer and bandwidth test result between two servers with TCP window size 16Kbyte (default) and the average bandwidth is 4.56Mbits/sec.

TABLE II. TCP NETWORK PERFORMANCE

Source	Destination	Size	Bandwidth
CNIC (dt.escience.cn)	UCSD (vm5.ucsd.edu)	7.23MBytes	4.82Mbits/sec
		4.65MBytes	3.10Mbits/sec
		7.59MBytes	5.06Mbits/sec
UCSD (vm5.ucsd.edu)	CNIC (dt.escience.cn)	7.23MBytes	4.82Mbits/sec
		7.56MBytes	5.04Mbits/sec
		6.80MBytes	4.53Mbits/sec

In order to evaluate the performance of data streaming, we customize three different kinds of PDB files, one 153M file, one 10M file and one 70M file. We use JMol plug-in to receive and visualize PDB files based on UCSD and CNIC Data Turbine servers. To avoid the influence of unexpected network traffic, TCP network performance test and

streaming performance test are operated in the same time slots.

Table III gives the time cost and average transfer rate by RIMES based on Data Turbine. It shows that the top transfer rate of RIMES is near to that of Iperf on WAN. The performance of RIMES in large frames is better than that in small frames. The performance of RIMES is comparable for the best possible throughput given available network bandwidth.

TABLE III. STREAMING PERFORMANCE TEST WITH BIOMEDICAL DATA

File Size	Frame	Time cost	Transfer Rate
380MBytes	60	717s	4.24Mbits/sec
92MBytes	200	571s	1.29Mbits/sec
70MBytes	800	196s	2.86Mbits/sec
50MBytes	10000	137s	2.92Mbits/sec

B. PRAGMA experimental environment

PRAGMA [20] is a community in order to advance the use of grid technologies in applications among a community of investigators working with leading institutions around the Pacific Rim [21]. Researchers have deployed more than five Data Turbine nodes in PRAGMA testbed for large-scale sensor-based observing systems [22]. The performance of Data Turbine routing mechanism has been tested under long-term network failures which last for a duration of an hour or more has been tested and it is ensured that each mirror frame is sent to the downstream server by round-trip confirmation of the receipt of each mirrored data frame.

CNIC and UCSD Data Turbine servers are also part of PRAGMA experimental environments for data streaming. Now the RIMES-based data-intensive environment is ready for more grid-computing simulations and visualizations in biomedical domain.

VI. CONCLUSION

In this paper we present RIMES, a data-intensive virtual environment for biomedical research, which extends scientists’ desktops to biomedical grid computation resources and mass storage systems. A web-based portal based upon Duckling supports job submission, analysis and visualization. It is hopeful that RIMES is also applicable to other domains, especially those with data-intensive workflows, such as astronomy and ecoinformatics.

VII. RELATED WORKS

PRAGMA testbed provides an ideal environment for testing and hosting Open Source Data Turbine streaming data service due to its international footprint and availability as a development platform on 24-7 basis. Open Source Data Turbine Middleware is being used in multiple projects which are related to PRAGMA. The Data Turbine nodes also have been integrated into PRAGMA cloud computing infrastructure as well.

On the other hand, RIMES API has been integrated into other applications, such as RDV (Real Time Viewer), to visualize the biomedical RMSD data, which is generated by VMD (Virtual Molecular Dynamics) [23].

The binary file of RIMES and JMolApplet can be downloaded from the website <http://pragma.escience.cn/> and the source code will be released for open source in the near future.

VIII. LIMITATIONS AND FUTURE PLANS

One criticism that we should discuss is whether data replication using Data Turbine is robust and scalable. During performance test, it isn't applicable to test large amount of data such as replicating over 10 GB trajectories using Data Turbine. But the reliability of huge data replication is really a significant aspect.

Coarse graining of the simulation data can be a good mechanism to improve the performance or the user experience given network bandwidth limitations by screening frames by intervals, especially for replicating huge trajectories.

ACKNOWLEDGMENT

The research was supported by Chinese Academy of Sciences - Visiting Scholars Fund; and by CNIC Director Research Foundation under Grant CNIC_ZR_09002; and by Chinese Academy of Sciences of Special Items of Information - Research and development of e-Science virtual research platform under Grant INFO-115-D01; and by the National High Technology R&D Program of China - Scientific Data Grid and Application of Scientific Research under Grant 2006AA01A120.

REFERENCES

- [1] Zhaohui Ding, Xiaohui Wei, Yuan Luo, Da Ma, Peter W. Arzberger, and Wilfred W. Li, "Customized Plug-in Modules in Metascheduler CSF4 for Life Sciences Applications," *Journal of New Generation Computing*, vol.25, No.4, pp.373-394, Aug. 2007.
- [2] Zhaohui Ding, Xiaohui Wei, Yifan Zhu, Yaoguang Yuan, Wilfred W. Li, and Osamu Tatebe, "Implement the Grid Workflow Scheduling for Data Intensive Applications with CSF4", Fourth IEEE International Conference on eScience, IEEE CS Press, pp.563-569, Indiana, U.S., 7-12 Dec. 2008.
- [3] Opal Toolkit, <http://www.nbc.net/software/opal/>.
- [4] Tiankai Tu, Charles A. Rendleman, David W. Borhani, Ron O. Dror, Justin Gullingsrud, Morten Jensen, John L. Klepeis, Paul Maragakis, Patrick Miller, Kate A. Stafford, David E. Shaw, "A scalable parallel framework for analyzing terascale molecular dynamics simulation trajectories," SC '08: Proceedings of the 2008 ACM/IEEE conference on Supercomputing, IEEE Press, Nov. 2008.
- [5] RCSB Protein Data Bank, <http://www.rcsb.org/pdb/>.
- [6] <http://www.ks.uiuc.edu/Research/vmd/plugins/molfile/dcdplugin.html>
- [7] Chemistry at HARvard Macromolecular Mechanics (CHARMM), <http://www.charmm.org/>.
- [8] Mark Nelson, William Humphrey, Attila Gursoy, Andrew Dalke, Laxmikant Kalé, Robert D. Skeel, and Klaus Schulten, "NAMD - A parallel, object-oriented molecular dynamics program," *International Journal of Supercomputer Applications and High Performance Computing*, 10:251-268, 1996. <http://www.ks.uiuc.edu/Research/namd/>.
- [9] James C. Phillips, Gengbin Zheng, Sameer Kumar and Laxmikant V. Kale, "NAMD: biomolecular simulation on thousands of processors," *Supercomputing '02: Proceedings of the 2002 ACM/IEEE conference on Supercomputing*, IEEE Computer Society Press, Nov. 2002.
- [10] Tony Fountain, Sameer Tilak, Paul Hubbard, Peter Shin, and Lawrence Freuderger, "The Open Source DataTurbine Initiative: Streaming data middleware for environmental observing systems," *International Symposium on Remote Sensing of Environment*, May. 2009. <http://www.dataturbine.org/>.
- [11] Kejun Dong, Kai Nan, Deting Yang, Jianjun Yu, Yongzheng Ma, "Collaboration Environment for e-Science: Duckling," CANS 2009, Beijing, China, Dec.5, 2009. <http://duckling.escience.cn/>.
- [12] Jmol: An Open-Source Java Viewer for Chemical Structures in 3D, <http://jmol.sourceforge.net/>.
- [13] RDV: A Visualization Environment for Scientific and Engineering Data, <http://code.google.com/p/rdv/>.
- [14] Thomas A. DeFanti, Gregory Dawe, Daniel J. Sandin, Jurgen P. Schulze, Peter Otto, Javier Girado, Falko Kuester, Larry Smarr and Ramesh Rao, "The StarCAVE, a third-generation CAVE and virtual reality OptIPortal," *Future Generation Computer Systems*, Volume 25, Issue 2, pp. 169-178, Feb. 2009.
- [15] Luc Renambot, Henri E. Bal, Desmond Germans, Hans J. W. Spoelder, "CAVEStudy: An Infrastructure for Computational Steering and Measuring in Virtual Reality Environments," *Cluster Computing*, Volume 4 Issue: 1, Kluwer Academic Publishers, Mar. 2001.
- [16] COVISE: COLlaborative VISualization and Simulation Environment, <http://www.hlr.de/organization/av/vis/covise/>.
- [17] PAAS: latform as a Service, http://en.wikipedia.org/wiki/Platform_as_a_service.
- [18] Nirimesh Harish, P. Narayanan, P.J., "Garuda: A Scalable Tiled Display Wall Using Commodity PCs", *Visualization and Computer Graphics*, IEEE Transactions on Publication, Volume: 13, Issue: 5, pp.864-877, Sept.-Oct. 2007.
- [19] C-H. Hsu and U. Kremer, "IPERF: A Framework for Automatic Construction of Performance Prediction Models," In *Proceedings of the 1st Workshop on Feedback-Directed Optimization*, October 1998. <http://iperf.sourceforge.net/>.
- [20] Peter Arzberger, Phil M. Papadopoulos, "PRAGMA: Example of Grass-Roots Grid Promoting Collaborative EScience Teams," *CTWatch*. Vol 2, No. 1 Feb 2006. <http://www.pragma-grid.net/>.
- [21] Cindy Zheng, Mason J. Katz, Phil M. Papadopoulos, David Abramson, Shahaan Ayyub, Colin Enticott, Slavisa Garic, Wojtek Goscinski, Peter Arzberger, Bu Sung Lee, Sugree Phatanapherom, Somsak Sriprayoons, "Lessons Learned Through Driving Science Applications in the PRAGMA Grid," *Int. J. Web and Grid Services*, Vol.3, No.3, pp287-312. 2007.
- [22] Corneliu Cotofana, Longjiang Ding, Peter Shin, Sameer Tilak, Tony Fountain, Jennifer Eakins, and Frank Vernon, "An SOA-based Framework for Instrument Management for Large-scale Observing Systems (USArray Case Study)," *IEEE International Conference on Web Services (ICWS)*, Sep.18, 2006.
- [23] William Humphrey, Andrew Dalke, and Klaus Schulten, "VMD - Visual Molecular Dynamics," *Journal of Molecular Graphics*, 14:33-38, 1996. <http://www.ks.uiuc.edu/Research/vmd/>.