SOFTWARE FOR VIRTUAL ACCELERATOR DESIGNING

N. Kulabukhova, A. Ivanov, V. Korkhov, A. Lazarev, SPbSU, Saint-Petersburg, Russia

Abstract

The article discusses appropriate technologies for software implementation of the Virtual Accelerator. The Virtual Accelerator is considered as a set of services and tools enabling transparent execution of computational software for modeling beam dynamics in accelerators on distributed computing resources. Distributed storage and information processing facilities utilized by the Virtual Accelerator make use of the Service-Oriented Architecture (SOA) according to a cloud computing paradigm. Control system toolkits (such as EPICS, TANGO), computing modules (including high-performance computing), realization of the GUI with existing frameworks and visualization of the data are discussed in the paper. The presented research consists of software analysis for realization of interaction between all levels of the Virtual Accelerator and some samples of middleware implementation. A set of the servers and clusters at St.-Petersburg State University form the infrastructure of the computing environment for Virtual Accelerator design. Usage of component-oriented technology for realization of Virtual Accelerator levels interaction is proposed. The article concludes with an overview and substantiation of a choice of technologies that will be used for design and implementation of the Virtual Accelerator.

INTRODUCTION

The key idea of Virtual Accelerator (VA) concept is beam dynamic modeling by the set of several packages, such as COSY Infinity, MAD, etc., based on distributed computational resources, organized on Grid-technology.

Simulation beam dynamics by different packages with the opportunity to match the results (in case of using different resources for the same task) and the possibility to create the set of tasks when the results of using one package can be sent to the input of another is the main use of VA.

Users will get the access to VA resources by unified interface including GUI on different platforms:

- Web-based application: web-services, web browsers' application (Java Webstar technology, Silverlight, etc.).
- Desktop applications: Windows, Linux.
- Mobile platforms: Android, Windows Phone 7.

Note that the VA is considered as information and calculation environment and does not refer to the real-time control systems. On the other hand this control organization must be provided by connection to the specialized software (e.g. EPICS). Such kind of VA is examined in [1] and [2] where authors emphasize on accelerator control development.

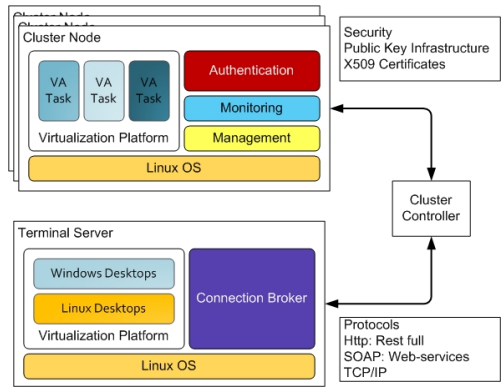

Figure 1: Virtual Accelerator.

In Fig. 1 VA scheme is shown. Cluster controller is a middleware framework developed to interact with calculation programs. This framework provides access by widely used standard protocols such as HTTP, SOAP, TCP/IP.

The same approach to develop a virtual laboratory is discussed in papers [3] and [4] for nuclear physics applications. In article [5] a heterogeneous computing environment and development of the distributed computing systems development are examined.

VIRTUAL ACCELERATOR SCHEME

The usage of VA as a computational resource can be presented as following:

1. Access to the VA resources by the authorization and authentication of users.
2. Specify initial data and settings for calculation:
   - on generalized language for system description (e.g. XML-based),
   - on native package language (COSY Infinity, MAD, OptiM).
3. Choice of the packages and simulation algorithms. Note that user can define the structure on one language (for example, on COSY Infinity notation) and select another program for calculation (e.g. OptiM). In this case conservation of system description will be made by the VA infrastructure invisible for user. And giving the instructions to start the simulation.
4. Monitoring of the task states, data visualization and representation. Access to a logging system with error tracking.

5. Access to the storage of user data that can be represented as simple remote file system with users directories and files. WebDAV protocol can be used for such system.

Security: Access to VA

The VA is available for users as Cloud-service (based on Grid-technology) with standart security features. The authentication techniques are:

- Username and Password
- Public Key Infrastructure and X509 Certificates

Transport level security is provided via SSL encryption. For the prototyping and testing we use OpenSSL program to generate own certificates.

Middleware Implementation: Server Side

Nowadays there exist number of programming languages and technologies. For VA implementation we use Java VM and .NET Framework[1]. Each of these platforms has great built-in capabilities for developing server side software. For example, WCF (Windows Communication Foundations) as a part of .NET Framework provide mapping of business logic in high-level programming languages into web-protocols without any additional effort other than adding declarative information to existing methods.

Client Application

As we mentioned above VA can provide unified access to it services both for web and desktop application. Not

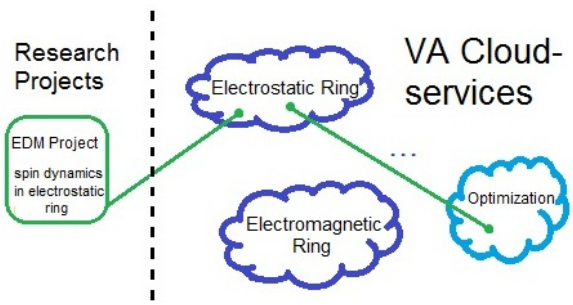

Figure 2: Cloud-services for EDM-project.

all services are required in each case. For example, for EDM-search project[2] it is necessary to include only electrostatic elements, optimization modules and related components (see Fig. 2). Other modules are enabled but available for the research purposes.

An example of the program for EDM project is presented on Fig. 3. It involves electrostatic LEGO objects (elements

[1]Microsoft .NET Framework under Academic Alliance License and Mono as OpenSource platform.

[2]St. Petersburg State University collaborates with Institute for Nuclear Research of Forschungszentrum Jülich, Germany.

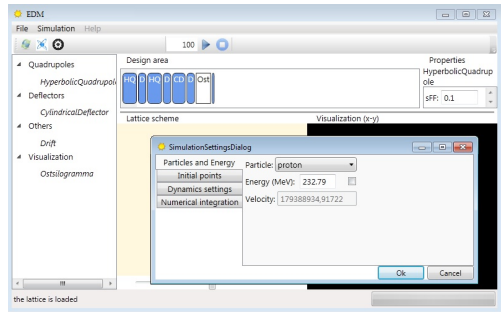

Figure 3: Windows desktop application.

with its parameters, fringe field components), optimization module (e.g. optimization module based on genetic algorithm approach) and visualization tools.

Task Monitoring

For the calculation complex problems (such as long-term beam evaluation) it is important to provide monitoring tools. Researchers at any time can have an opportunity to see progress of the task, restart or cancel it, save obtaining result and etc. VA services must provide real-time interaction with users requests. This requirements provide a complete integrated computing environment for composing,, running, controlling and visualizing applications.

Connection to Physical Equipment

In VA concept it is common interface for simulation model and there should not be differences between access. Both simulation model and equipment adapter (e.g. under the EPICS) must implement the same interfaces. EPICS can be used as a part of the simulation process.

The common user interface of VA allows us to get solutions both in simulation models and in a real machines. This approach provides researcher some mechanism of system identification, parameters optimization and result verification. Such opportunity is impossible without computational models and is a goal aim of the current research.

LEGO PARADIGM IMPLEMENTATION

A LEGO paradigm for VA design is described in [6]. In terms of information technology it corresponds to object oriented design and component programming. Each object represents as independent component with own parameters and behavior. Building the proper class hierarchy allows developer to scale and modify the objects using the already existing solutions. C# and .NET are used for prototyping at the moment. This technology has several advantages compared to Java VM (pointers and memory allocation in the stack, declarative description of the desired functionality, etc.). But in general, these platforms are very similar and the choice depends on the preferences of the development team at all. Note that for implementation of numerical al-

Table 1: Summary of Used Languages

Language	Performance	Web Interfaces	Flexibility
C++	++	--	--
C#	+-	++	++
Java	-+	++	+-

gorithms it will be use C++ as the most powerful language for computing tasks.

A LEGO object is represented as a component object. In object oriented design it means a class with predefined interfaces. Objects has an inheritance hierarchy in that LEGO objects can defined.

LEGO means an ability to choose and change necessary elements without reconstruction of whole structure. For example, in simulation we have a set of integration methods. Of course, each of them mast be used in specific cases that depends on physic problem, mathematical requirements and etc. But for LEGO paradigm they are all one object that knows how to obtain solution and to present it in some form.

SCIENTIFIC VISUALIZATION

Scientific visualization plays an important role in result processing. In the images and animations researcher can diagnose some features associated with the intuitive understanding of the physics of the process rather than they can obtain the same result during statistical data processing.

Visual representation of data help to verify obtaining result and to quickly detect errors that invisible for calculation algorithms but critical to plausibility checking.

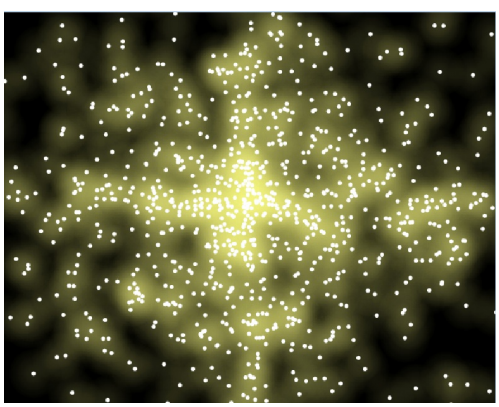

Figure 4: Coordinatewise slice of beam.

In Fig. 4 a coordinatewise ($x - y$ space) slice of beam is shown. In this figure an initial particle distribution is presented. On the picture we can see some clusters of points that may be difficult to find by data mining methods.

CONCLUSION

In the article a scheme of VA organization is described. Data access protocols that will be used are mentioned. Some modules such as global optimization tools, simulation and numerical algorithm are completely developed, other are in a progress. Future plans for this project include extend of the VA tools, implementation of Cloud-service architecture and support developed systems via knowledge base growing, development of methods and approaches in beam dynamics and simulation and validation proposed approach on real machines.

Some approaches that was described above were tested in the distributed computational environment at the faculty of Applied Mathematics and Control Processes on the department of Computer Modelling and Multiprocessor Systems[3].

REFERENCES

[1] P.C. Chiu, C.H.Kuo, Jenny Chen, Y.S. Cheng, C.Y.Wu, Y.K.Chen, K.T. Hsu, "Virtual Accelerator Development For The TPS", IPAC'10, Kyoto, Japan 2010, WEPEB019, p. 2728, http://www.JACoW.org.

[2] C. Gulliford, I. Bazarov, J. Dobbins, R. Talman, N. Malitsky, "The NTMAT EPICS_DDS Virtual Accelerator For The Cornel ERL Injector", IPAC'10, Kyoto, Japan 2010, WEPEB022, p. 2734, http://www.JACoW.org.

[3] V. Korkhov, D. Vasyunin, A. Belloum, S. Andrianov, A. Bogdanov, "Virtual Laboratory and Scientific Workflow Management on the Grid for Nuclear Physics Applications", Distributed Computing and Grid-Technologies in Science and Education: Proc. of the 4th Intern, Dubna, Russia 2010, p. 153.

[4] A. Belloum, D. Groep, et al., "VLAM-G: a grid-based virtual laboratory", Future Generation Computer Systems, V. 19, 2003, P.209–217. Distributed Computing and Grid-Technologies in Science and Education: Proc. of the 4th Intern, Dubna, Russia 2010, p. 153.

[5] A. Bogdanov, A. Lazarev, Myo Tun Tun, La Min Htut, "Development of the Distributed Computing Systems and Running Applications in the Heterogeneous Computing Environment", Distributed Computing and Grid-Technologies in Science and Education: Proc. of the 4th Intern, Dubna, Russia 2010, p. 69.

[6] S. Andrianov, A. Ivanov, E. Podzyvalov, "A LEGO Paradigm For Virtual Accelerator Cocept", in this Proc.

[3] http://www.apmath.spbu.ru/en/staff/andrianov/