Virtual Accelerator: Software for Modeling Beam Dynamics

Nataliia Kulabukhova, Vladimir Korkhov, Sergey Andrianov

Saint-Peterburg State University

St. Petersburg, Russia

e-mail: kulabukhova.nv@gmail.com, vladimir@csa.ru, sandrianov@yandex.ru

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. The main purpose of a virtual accelerator is to conduct computational experiments to simulate the beam dynamics using various software packages with the ability to compare the results of calculations (in case solutions of the same problem are obtained by various means), and the ability to create a task flow (solutions of one package can be used as an input for the subsequent calculation in another package). Control system toolkits EPICS (Experimental Physics and Industrial Control System [9], realization of the Graphical User Interface(GUI) with the existing frameworks and visualization of the data are discussed in the paper. The presented research consists of a software analysis for realization of interaction between all levels of the Virtual Accelerator.

We present the concepts and the design of the Virtual Accelerator, describe the prototype implementation and show early results.

Keywords

distributed computing; beam accelerator; modeling; grid computing; cloud computing.

1. INTRODUCTION

It is not a secret that in many different areas scientists try to simulate a lot of physical processes, from little ones, such as pendulum motion till the very huge like the interaction between planets. In beam physics it is necessary to simulate the dynamics of beam. In order to control large-scale accelerators efficiently, a control system with a virtual accelerator model was constructed by many facilities [1, 2]. In many papers by the words Virtual Accelerator an on-line beam simulator provided with a beam monitor scheme is meant. It works the parallel with the real machine. The machine operator can access the parameters of the real accelerator through the client and then feed them to the virtual accelerator, and vice versa. Such a virtual machine scheme facilitates developments of the commissioning tools; enables feasible study of the proposed accelerator parameters and examination of the measured accelerator data. That is the common scheme of virtual accelerators used in different laboratories [3]. Until now there is no virtual accelerator working without a real machine. Our goal is to construct a virtual accelerator application used independently from any machine.

2. CONCEPT OF THE VIRTUAL ACCELERATOR

The key idea of the Virtual Accelerator (VA) concept is modeling of beam dynamics with the help of several software packages, such as COSY Infinity [4], MAD [5], etc, composed in pipelines and enacted on distributed computing resources. The main use of the VA is simulation of beam dynamics by different packages with the opportunity to match the results (in case of using different solution methods for the same problem) and the possibility to create pipelines of tasks when the results of one processing step based on a particular software package can be sent to the input of another processing step. The VA is considered as information and computing environment and does not refer to real-time control systems. However, real-time control can be provided by the connection to specialized software (e.g. Experimental Physics and Industrial Control Systems – EPICS [6]).

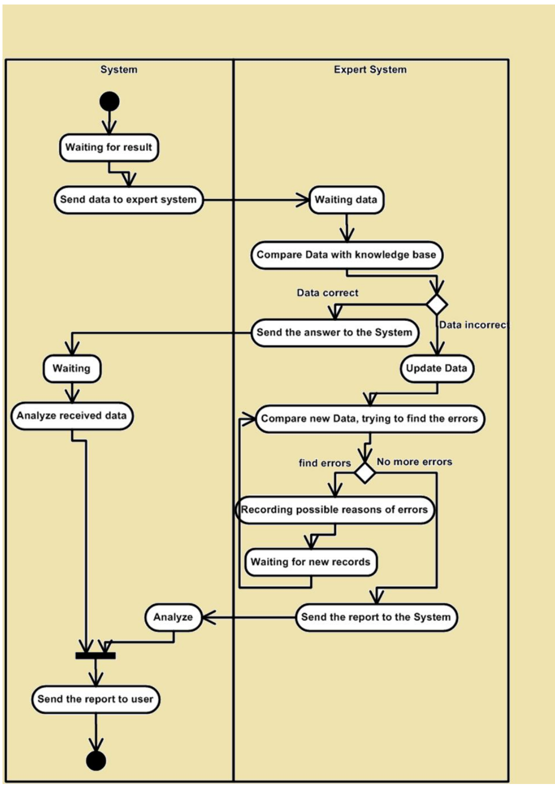

Figure 1. Knowledge base

The general idea of the software implementation is based on the Service-Oriented Architecture (SOA) that allows using Grid and Cloud computing technologies and enables remote access to the information and computing resources. Distributed services establish interaction between mathematical models and a low-level control system. The VA user interface allows getting solutions both from simulation models and from real accelerator machines. This approach gives researchers an ability for system identification, parameter optimization, and result verification, which is impossible without computational models. Similar approach to develop a virtual laboratory is discussed in [8] for nuclear physics applications. 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. Users will get the access to VA resources by a unified interface including GUI on different platforms. Figure 1 shows a scheme of workflow of Knowledge base. The Virtual Accelerator usage scenario is the following:

1. The user gets access to the VA user interface (authenticated and authorized by using a symbolic password, a graphical password, etc.);

2. The user sets initial conditions and parameters for the calculation. The options are:

1) using "generic" description language that can be converted into a specific language used in the individual packages: MAD, COSY, etc.;

2) directly in the language of one of the packages, which will be used for calculations;

3. The user selects a package (or a set of packages), which will perform calculations;

4. The user instructs the system to run calculations using the packages and given initial data. This is done using either a dedicated resource (e.g. cluster), which is selected manually, or automatically selected resource based on information about requirements of the application.

5. After starting the calculation the user may wish to see the intermediate results. This capability depends on the abilities of the packages. It is important to track errors that occur. As practice shows, the most difficult is to figure out why something does not work. Carefully organized collection of error messages must be maintained. To collect this information, annotate data and results of computations the so called provenance systems are used [7].

7. VA offers the possibility of organizing a workflow – sequentially running packages, where the next step uses the data obtained on the previous step(s). The means to convert data formats between packages must be provided.

8. Calculation results can be visualized by means of VA. It is particularly important to be able to visually compare the results of calculations of a problem in different packages.

Grid environment allows coordinated resource sharing and problem solving among groups of trusted users within Virtual Organizations. Such environments enable global distributed collaborations involving large numbers of people and large scale resources, and make data and computing intensive scientific experiments feasible. One of the important research topics in e-Science is to develop effective Grid enabled Problem Solving Environments (PSE), also called Virtual laboratories, for different scientific domains. Organizing software utilities (e.g. simulators, visualization and data analysis tools) as a meta experimental environment, a PSE allows a scientist to plan and conduct experiments at a high level of abstraction [8].

3. METHODS REALIZATION

It has already been said that Virtual Accelerator in itself can use different packages to calculate. We tried to analyze some of most popular packages used to simulate beam dynamics and to compare them. We took these:

- MAD X;
- OptiM;
- Cosy Infinity;
- MaryLie [10];
- TRANSPORT;
- Matrix Formalism.

The two programs TRANSPORT and MAD use the same format for beamline specification. MaryLie is based on Lie algebra, but instead of OptiM it has a multi-platform Graphical User Interface. So, it can be used on Linux machines. As well as MAD X can be used. Problem is that practically every program has differences in input data format. In VA this problem will be solved by unified GUI in the way that user won't even know what program the VA is using during the simulation time.

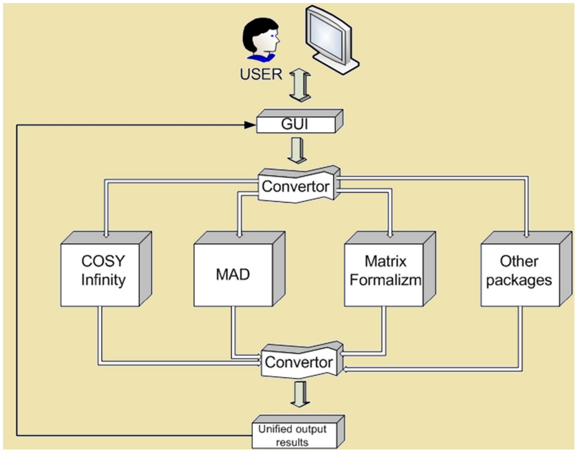

Figure 2.

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 a conservation of system description will be made by the VA infrastructure invisible for the user. VA will give the instructions to start the simulation. How it will be is shown on Figure 2.

4. MATRIX FORMALISM AND PARALLELIZATION

The natural parallel and distributed structures of beam physics problems allow the use of parallel and distributed computer systems. But the usual approaches based on traditional numerical methods demand using the resources of supercomputers. This leads to the impossibility of using such multiprocessing systems as computational clusters. There are two classes of problems in beam physics which demand very extensive computer resources. The first class includes long-time evolution problems; the second is concerned with the computer realization of optimization procedures for beam lines. Examples of the first type of problem include multi-turn injection and extraction of the beam in circular accelerators. Usually, these problems do not consider space charge effects. For advanced applications it is essential to study beam dynamics in high-intensity accelerators. Such machines are characterized by large beam currents and by very stringent uncontrolled beam loss requirements. An additional difficulty of numerical simulation is connected with a long-time beam evolution that requires a computation of hundreds of thousands or millions of turns. It requires the use of high-performance computers for beam evolution study. The problems of similar multi-turn evolution such as transverse stability with nonlinear space charge, uncontrolled beam losses due to space-charge-induced halo generation, etc. can also be mentioned. These problems are peculiar to modern high-intensity machines and require careful investigations of long-time evolution effects. From the computational point of view there are some problems related to the choice of models for beams with space charge [11], the presentation form of the beam propagator, and so on.

Matrix formalism [12] is a high-performance mapping approach for ODE solving. It allows to present solution of the system in the following form

$$X = \sum_{i=0}^{k} R^{1i}(t) X_0^{[i]}$$

where R^{1i} are numerical matrices. So, this approach can be easily implemented in a parallel code. Due to the fact that only matrix multiplication and addition are used, GPU programming is especially suitable for this purpose. The research has shown that there is no great benefits via parallelization of computational code for one particle by using GPU. In this case overhead on data sending is significant. On the other hand matrix formalism allows to process a set of initial points, where parallelization is more preferably. Let's introduce a set of initial particle

$$M = (X_0^1 X_0^2 \dots X_0^p) \qquad (1)$$

According to the equation (1) the resulting points can be calculated

$$M = \sum_{i=0}^{k} R^{1i}(t) \left((X_0^1)^{[i]} (X_0^2)^{[i]} \dots (X_0^p)^{[i]} \right)$$

Note that the sizes of matrices in this equation is much greater than in (1) when a set of initial particles is quite large. The use of matrix formalism allows to build some important criteria in terms of matrix elements. This is significantly reducing computational time.

5. CONCLUSION

From our point of view the model of Virtual Accelerator should consist of a simulation engine that reads in all relevant lattice settings from the accelerator control system and computes data corresponding to the real diagnostic data output from the real accelerator. Or, if no accelerator exists at that time; a mathematical model will be given to compare the results. Here would ideally be a one-to-one map between all relevant control parameters and output data from the machine and its simulated counterpart. This model inherently assumes that the simulated lattice updates automatically any time the physical lattice settings are modified. If we speak about a mathematical model, it means that at any time we can change settings and continue the computations. This model should represent an extremely valuable tool to the beam physicist by providing a useful way to benchmark the simulation engine as well as providing a detailed description of the dynamics in the accelerator in real time. By saying real time, it means the time the system needs to get the problem, to analyze it and to give the message about it on the screen. It does not allow the beam physicist to directly use the simulation engine to guide beam operation because the virtual lattice settings are limited to those read in from the physical machine.

The future development of the research can be based on writing software using different parallel techniques and complete implementation of the described approaches.

4. ACKNOWLEDGEMENT

Authors thank Dmitry Vasunin for many useful discussions about Virtual Accelerator development. And Computations were partly carried out on cluster HPC-0011654-001 of Saint-Petersburg State University, Faculty of Applied Mathematics and Control Processes.

REFERENCES

[1] NP.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] H. Harada, K. ShigakiF, Noda, H. Hotchi, H. Sako, H. Suzuki, K. Furukawa, S. Machida, “VIRTUAL ACCELERATOR AS AN OPERATION TOOL AT J-PARC 3 GEV RAPID CYCLING SYNCHROTRON (RCS)”, EPAC'06, Edinburgh, Scotland, 2006, WEPCH128, p.~2224, {http://www.JACoW.org}.

[3] A.Shishlo, P. Chu, J. Galambos, T. Pelaia, “THE EPICS BASED VIRTUAL ACCELERATOR CONCEPT AND IMPLEMENTATION”, Proceedings of the 2003 Particle Accelerator Conference, pp. 2366-2368.

[4] K. Makino, M. Berz, “COSY INFINITY Version 9”, Nuclear Instruments and Methods A558 (2005) pp. 346-350

[5] Grote, H., and F. Schmidt. “MAD-X-an upgrade from MAD8”, Proceedings of the 2003 Particle Accelerator Conference, Portland, OR. 2003

[6] B. Dalesio, “EPICS V4 Expands Support to Physics Application”, Data Acquisition, and Data Analysis. ICALEPCS 2011

[7] M. Gerhards, V. Sander, T. Matzerath, A. Belloum, D. Vasunin, A.Benabdelkader. “Provenance opportunities for WS-VLAM: an exploration of an e-science and an e-business approach”, Proceedings of the 6th workshop on Workflows in support of large-scale science Pages 57-66, ACM New York, NY, USA 2011

[8] V. Korkhov, D. Vasyunin, A. Belloum, S. Andrianov, A. Bogdanov, “Virtual Laboratory and ScientificWorkflow 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,{http://www.JACoW.org}.

[9] C. Gulliford, I. Bazarov, J. Dobbins, R. Talman, N. Malitsky, ”The NTMAT EPICS-DDS Virtual Accelerator for the Cornell ERL Injector”, IPAC'10, Kyoto, Japan 2010, WEPEB022, p.~2734, {http://www.JACoW.org}.

[10] R. D. Ryne, A. J. Dragt, A. Adelmann et al., “RECENT PROGRESS ON THE MARYLIE/IMPACT BEAM DYNAMICS CODE”, Proceedings of ICAP 2006, Chamonix, France, p.157 {http://www.JACoW.org}.

[11] N. Kulabukhova, “SPACE CHARGE DOMINATED ENVELOPE DYNAMICSUSIN GGPUs”, Proc. of IPAC’13, Shanghai, China, {http://www.JACoW.org}.

[12] N. Kulabukhova, “GPGPU IMPLEMENTATION OF MATRIX FORMALISM FOR BEAM DYNAMICS SIMULATION”, Proc. of ICAP’12, Rostok, Germany, {http://www.JACoW.org}.

[13] N. Kulabukhova, A. Ivanov, V. Korkhov, A. Lazarev, “Software for Virtual Accelerator Designing”, ICALEPCS'11, Grenoble, France,2011, WEPKS016, p.~816, {http://www.JACoW.org}.