

VIRTUAL ACCELERATOR AT NSLS II PROJECT*

Guobao Shen[#], Lingyun Yang, Yongjun Li, BNL, Upton, NY 11973, USA

Abstract

A virtual accelerator has been developed at NSLS II to support tools development from physics study and beam commissioning to beam operation. The physics results are provided using TRACY-II simulation code thru EPICS process variables, which was pioneered originally at Diamond Light Source. The latest virtual accelerator supports all major accelerator components including magnets (Dipole, Quadrupole, Sextuple, correctors, and so on), RF cavity, insertion device, and other diagnostics devices (BPM for example), and works properly for both linear machine and synchrotron ring. Two error mechanisms are implemented, which are random error for each magnet setting, and systematic error to simulate misalignment. Meanwhile, it also provides sort of online model functions including serving beta function, and close orbit data. In NSLS II, there are 5 virtual accelerators deployed, and 3 of them are running simultaneously. Those virtual accelerators have been effectively supporting the tools development such as physics applications, and middle layer services. This paper presents the latest status of virtual accelerator, and our plan for its future development and deployment.

INTRODUCTION

For a large-scale accelerator complex, a high level physics application is critical to tune and operate machine effectively. To support high-level physics application development, a virtual accelerator is important to support physics application development, especially at early stage of each facility. Various virtual accelerators have been development over past years [1, 2]. At NSLS II project, a virtual accelerator, which has been named as J.M.S server, has been development from its early stage [3] with continuing the work done at Diamond Light Source. This J.M.S. server has been using by various sub-systems from our injector to storage ring [4], plays a solid ground to support software development, from our middle layer service to high level physics application and even operation panel, and enables a testing of applications through EPICS control system.

The NSLS II control system is developed using EPICS framework. To simulate the channel access, based on the work at [2], an EPICS device support has been improved to support all major accelerator components, which include bending, quadrupole, sextupole, corrector (horizontal and vertical), insertion device, RF, and diagnostic device such as BPM. Underneath, it used to using TRACY-II [5] library originally, and has been updated to TRACY-III at NSLS II as shown in Fig 1.

*Work supported under auspices of the U.S. Department of Energy under Contract No. DE-AC02-98CH10886 with Brookhaven Science Associates, LLC, and in part by the DOE Contract DE-AC02-76SF00515
[#]shengb@bnl.gov

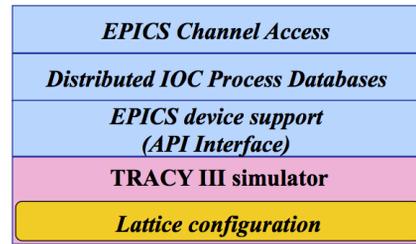


Figure 1: J.M.S. server architecture.

The EPICS processing database is generated automatically thru an application, which is developed with PyQt. This application enables to update a lattice used by J.M.S. server to latest version.

This paper is organized as below. An automatic EPICS database generating application is presented; following by is the improvement of J.M.S. server, which focuses on improvement on supporting of bending magnets and insertion device. After that, the usage of J.M.S. server at NSLS II project is summarized and its performance is analysed. Finally, a summary is given.

EPICS DATABASE GENERATING

For an EPICS based J.M.S. server, it is usually time-consuming to generate an EPICS run-time database, especially for a large facility like NSLS II. However, it is important to keep lattice used by J.M.S. server updated.

At NSLS II project, a graphic interface is developed with PyQt. This application consists of 2 parts, which are lattice analysis part and database generating part respectively. The lattice analysis part takes a lattice in either plain text format, or TRACY-III format. User could use it to convert a text lattice into a TRACY-III lattice, or modify an existing lattice to satisfy requirement.

Once a TRACY-III lattice is obtained, user can load in the lattice by clicking “Read Lattice” button as shown in Fig 2.

This generator generates EPICS records for each device like magnet, and global records for Tune, Orbit, S position, and TWISS (α_{xy} , β_{xy} , η_{xy} , and ϕ_{xy}). The record name follows NSLS II naming convention, and user has flexibility to change name pattern for global records currently.

When generating a database, user has flexibility whether to add random error by checking/unchecking “include noise” radio button. Details could be found in next section.

Since a user may want to generate many instances using same or different lattice, an instance number is provided, which allows user to specify the instance number.

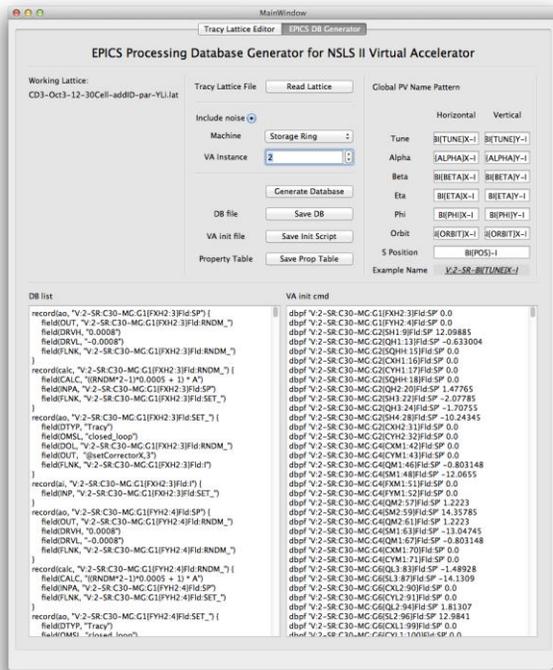


Figure 2: EPICS processing database generating.

Database is generated by clicking “Generate Database” button, and results are displayed in the text field as shown on the 2nd half of Fig 2. The left part of results is a list of all records for the new EPICS database, and the right part is a list, which could be used to initialize all settings.

If everything is good, the result could be saved into files, and it is ready to start a new J.M.S. server with generated database.

J.M.S. SERVER IMPROVEMENT

The J.M.S. server has been improved in many respects. Some key improvements are presented in this section.

TRACY Simulator

Originally, the J.M.S. server was based on TRACY-II version. At NSLS II project, since there are many improvements in TRACY-III, to get the same simulation result, the simulator used in J.M.S. server has been updated to latest TRACY-III.

Error Emulation

To provide more realistic simulation result, two error mechanisms are implemented, which are random error and systematic error respectively.

The random error is to simulate hardware noise, which are setting noise for magnet power supply, and read-back noise for diagnostic device like BPM. As described above, this noise could be enable/disable when generating database. An example used at NSLS II is shown in Fig 3 as below.

```
record(ai, "V:2-SR:C30-BI:G2{PH1:11}SA:YGET_") {
  field(DTYP, "Tracy")
  field(INP, "@getBpmY,11")
  field(SCAN, "Event")
  field(EVNT, "101")
  field(FLNK, "V:2-SR:C30-BI:G2{PH1:11}SA:YRDNM_")
}
record(calc, "V:2-SR:C30-BI:G2{PH1:11}SA:YRDNM_") {
  field(CALC, "((RNDM*2-1)*0.0001 + 1) * A")
  field(INPA, "V:2-SR:C30-BI:G2{PH1:11}SA:YGET_")
  field(FLNK, "V:2-SR:C30-BI:G2{PH1:11}SA:Y")
}
record(ai, "V:2-SR:C30-BI:G2{PH1:11}SA:Y") {
  field(INP, "V:2-SR:C30-BI:G2{PH1:11}SA:YRDNM_")
}
record(oo, "V:2-SR:C30-MG:G2{QH1:13}Fld:SP") {
  field(OUT, "V:2-SR:C30-MG:G2{QH1:13}Fld:RNDM_")
  field(DRVH, "-0.6963044")
  field(DRVL, "-0.5697036")
  field(FLNK, "V:2-SR:C30-MG:G2{QH1:13}Fld:RNDM_")
}
record(calc, "V:2-SR:C30-MG:G2{QH1:13}Fld:RNDM_") {
  field(CALC, "((RNDM*2-1)*0.0001 + 1) * A")
  field(INPA, "V:2-SR:C30-MG:G2{QH1:13}Fld:SP")
  field(FLNK, "V:2-SR:C30-MG:G2{QH1:13}Fld:SET_")
}
record(oo, "V:2-SR:C30-MG:G2{QH1:13}Fld:SET_") {
  field(DTYP, "Tracy")
  field(OMSL, "closed_loop")
  field(DOL, "V:2-SR:C30-MG:G2{QH1:13}Fld:RNDM_")
  field(OUT, "@setQuadrupole,13")
  field(FLNK, "V:2-SR:C30-MG:G2{QH1:13}Fld:I")
}
```

Figure 3: Random error in database.

Figure 3 shows 2 random error examples, which are BPM read-back error and magnet setting error respectively. In those examples, each has been added a $\pm 0.01\%$ error.

Together with noise error, sometime, it is useful also to have a capability to provide J.M.S. server a systematic error to simulate for example misalignment error. A configuration file is provided, and a J.M.S. server is initialized at beginning. Unlike the random error, the random error cannot be changed unless the J.M.S. server is rebooted.

Bending Magnet

For a storage ring simulation, most likely, the bending angle is fixed, and there is no requirement to change the bending angle of a dipole. However, at NSLS II, the requirement was raised with expanding J.M.S. server to support various software developments, particularly for the first-turn beam study.

To satisfy this requirement, a function was added and improved to enable user to do a bending scanning.

Insertion Device

As a 3rd generation synchrotron light source like NSLS II, insertion device (ID) plays critical role. Therefore, supporting for insertion device commissioning software development becomes very important.

The 2nd order radia kickmaps are used to simulate IDs in TRACY-III simulator. The first and second field integrals from the measurement are simulated by four virtual kicks (two for each plane) located at the ID extremities as shown with red arrows in Fig. 4. The

closed orbit distortion due to the imperfection of ID will be compensated by four on-board correctors (as show in green). The calibration function of corrector strength to realize this feed-forward compensation has been successfully implemented and tested on the virtual machine. In the next step, we will use different kickmaps to simulate an ID at different statuses (gap/phase) to calibrate other feed-forward tables if necessary.



Figure 4: Insertion device emulation.

USAGE OF J.M.S. SERVER

The J.M.S. server provides a solid ground to development of high level physics application for beam study, commissioning, and operation, and middle layer service.

Physics Application

Currently, it is serving high level application environment [6], one turn beam study, insertion device application, and so on. A typical application is to display beam orbit and correct it, which is demonstrated as shown in Fig 5.

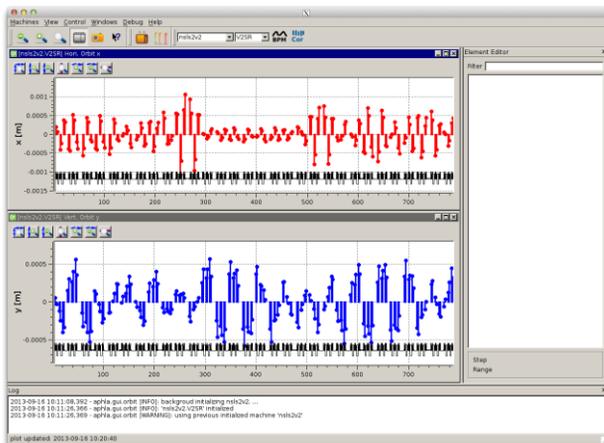


Figure 5: Beam orbit display and correction.

The application shown in Fig. 5 integrates orbit control and magnet control functions. It is initialized with pvs from J.M.S. server, and orbit is displayed with data from J.M.S. server.

Another application is to perform a beam based alignment as shown in Fig 6 [4, 6] which follows the ALS algorithm.

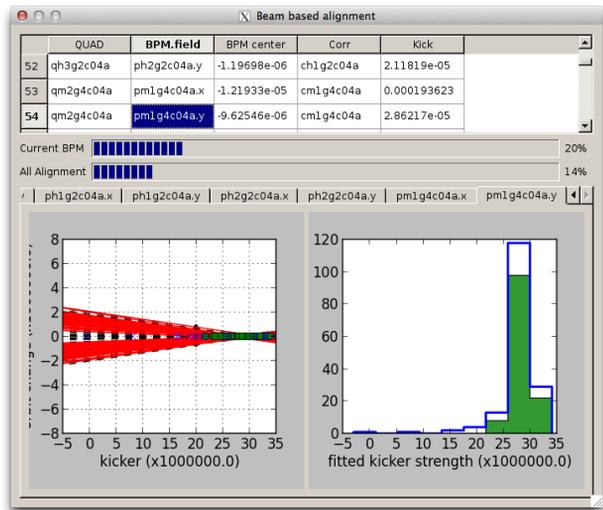


Figure 6: Beam based alignment with J.M.S. server.

For insertion device application, it has been developing in the same environment, and the following tests have been performed against J.M.S. server:

- Verification (and eventual correction) of closed orbit with the installed ID in “Passive State” (i.e. with gap open);
- Measurement of electron beam COD, tune shifts, coupling variation at different ID gaps and phases (or currents in main coils for EM EPU), with respect to the “Passive State”;
- Calculation of feed-forward correction tables for “on-board” ID corrector channels and testing their efficiency.

For the performance issue, as a J.M.S. server, it has its limitation, especially with insertion device as below:

- Measurement of efficiencies of “on-board” ID “corrector channels” based on COD. A 2D calibration table (kick vs. I and gap) needs to be measured and calculated;
- Optics distortion and tune-shift compensation by “current-strips”. Kick-maps are needed for different gap/phase, calibration table for “current strip”.

Middle Layer Service

Meanwhile, since it is wrapped into an EPICS IOC, it provides a platform to test our middle layer services development [7].

Figure 7 shows one use case of CSS (Control System Studio) channel viewer of channel finder service [8]. The channel finder service provides a dictionary service to enable a user to find a correct channel name. With the J.M.S. server, users could be able to test their applications, especially for the physics applications, and seamlessly develop their application against our middle layer services.

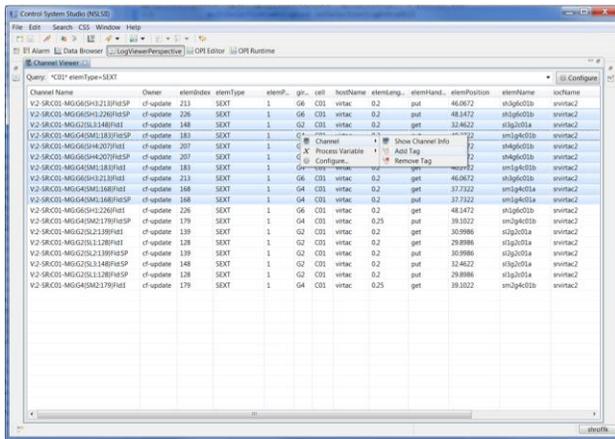


Figure 7: Channel finder service and J.M.S. server.

CURRENT STATUS

J.M.S. Server Instances

Currently, there are 5 J.M.S. servers deployed at NSLS II project as below:

1. Transport line for Linac to beam dump 1 (LBD1);
2. Transport line for Linac to beam dump 2 (LBD2);
3. Linac to booster ring transport line (LBT);
4. Storage ring with insertion devices (V2);
5. Storage ring one turn beam study (V3).

LBD1, LBD2 and LBT share the same front end, and only one instance runs at the same time. The rest are running are running simultaneously.

Performance

The LBD1, LBD2 and LBT are linear transport line, which means it could achieve a high repetition rate. To simulate our real machine, we limit it rate to 10 Hz.

Although the V3 instance is to simulate a synchrotron ring, but it is for first turn study, or first couple turn study. In this case, it is not needed to calculate the closed orbit. Therefore, considering the performance, we treated the machine as a super long linear machine. To simulate a real machine, we limit it rate to 1 Hz even with over 10000 components.

The most complicated instance is the V2. Some optimization has been tuned to achieve an acceptable performance as below:

- Turn off dispersion calculation;
- Turn off radiation calculation;
- Turn off RF cavity off by default;
- Lower number of integration steps to 1 for all sextupoles;
- Lower number of integration steps to 10 for insertion devices;

With above optimization, for V2 instance, which has over 3655 components with 12 insertion devices, and alignment errors for 2 quadrupoles, it can achieve around 1Hz repetition rate. For V2, a 2nd order Radia kick-map is used to calculate insertion device impact, which minimized integration steps to 10.

SUMMARY

The J.M.S. server has been developed for NSLS II project from its early stage, and is actively supporting our high level application development, and other middle layer service development.

There are 5 instances have been deployed in our project, and 3 of them are running simultaneously.

A good performance has been achieved, especially for our storage ring with 12 insertion devices. All those are running continuously, and provide a solid ground for our physics application development, and middle layer service development.

ACKNOWLEDGMENT

The authors would like to acknowledge the contribution of James Rowland at Diamond Light Source, who made original development to this virtual accelerator. James Rowland died in Nov 2012; to memorize his significant contributions to this work, this virtual accelerator has been named the J.M.S.[9]. The authors also want to thank Johan Bengtsson, Jinhyuk Choi at Brookhaven National Laboratory for their contributions on TRACY-III code, and helpful discussions. Finally, but not the last, the authors want to thanks Bob Dalesio, and Samuel Krinsky for their solid supports, and all of our colleagues for their suggestions.

REFERENCES

- [1] N. Yamamoto, "Use of a virtual accelerator for a development of an accelerator control system", Proceedings of the PAC97, Vancouver, Canada, May 1997, p. 2455.
- [2] M. T. Heron, et al., "The Diamond Light Source Control System", Proceedings of EPAC 2006, Edinburgh, Scotland, June, 2006, THPCH13, p. 3068 (2006)
- [3] G. Shen, "A Software Architecture for High Level Applications", Proceedings of the PAC09 Conference, Vancouver, Canada, May 2009 BNL-82325-2009-CP
- [4] L. Yang, et al., "Development Progress of NSLS-II Accelerator Physics High Level Applications", Proceedings of IPAC 2012, New Orleans, Louisiana, USA THPPR018, p. 4005 (2012)
- [5] M. Boge, "Update on TRACY-2 Documentation", SLS Internal Note, SLS-TME-TA-1999-0002 (1999); <http://slsbds.psi.ch/pub/slsnotes>
- [6] L. Yang, et al., "The Design of NSLS-II High Level Physics Applications", ICALEPCS'13, TUPPC130
- [7] G. Shen, et al., "NSLS-II Middle Layer Services", thess proceedings, TUPPC155
- [8] <http://channelfinder.sourceforge.net/>
- [9] B. Dalesio, J. Bengtsson, G. Shen, private communication