DEVELOPMENT OF A VIRTUAL ACCELERATOR FOR SIRIUS

X. R. Resende*, A. H. C. Mukai, L. N. P. Vilela, I. Stevani
Brazilian Synchrotron Light Laboratory (LNLS), Campinas, Brazil

Abstract

A virtual accelerator is being developed for Sirius, the new 4th generation synchrotron light source being built in Campinas, Brazil [1]. The virtual accelerator is an on-line beam simulator which is integrated into EPICS control system. It consists of a command line interface server with a channel access (CA) layer and with an in-house developed tracking code library written in C++ for efficiency gain. The purpose of such server is to facilitate early development and testing of high level applications for the control system.

INTRODUCTION

Sirius, the new storage ring at the Brazilian Synchrotron Light Laboratory (LNLS), will use EPICS as its control system. Most of the development of the high level applications (HLAs) will take place next year and will be conducted by the accelerator physics group. In the meanwhile, a few client applications have already been implemented to allow analysis of the choices in software development frameworks [2] that were made.

To be able to test and integrate the above-mentioned HLAs in the control system (CS), it was decided that a virtual accelerator (VA) with channel access server layer (CAS) for EPICS should be developed, implemented and made available as soon as possible. The idea is that having a VA allows for early development of control system software, be it for input-output controllers (IOCs) or HLAs.

The VA implements functionalities that can provide simulated process variables (PVs) on the CS that have not been made available yet, thus creating a mock-up CS environment in which early software development is possible. This test environment with VA also has the potential to speed up project development by partially parallelising implementations of applications that consume data from each other.

Commissioning training is also possible using HLAs in the control system provided with the virtual accelerator.

VIRTUAL ACCELERATOR

From the onset it was decided that the virtual accelerator would be composed of two parts: the first, a back-end machine application implementing a simulated virtual accelerator with a channel access server layer (VACA) and the second, a set of front-end virtual IOCs (vIOCS) with which other CS applications are supposed to interact.

Accelerator properties such as beam current and position, injection losses, power supply and RF subsystem setpoints, and so on, are simulated nominally in VACA. The virtual IOCS, on the other hand, encapsulate all PVs that represent the interface between the VA and the rest of the CS. They also add simulated fluctuations to accelerator properties and implement device-dependent parameters, such as excitation curves of the magnets or BPM calibration parameters.

The advantage of this approach is that virtual IOCs can be gradually replaced by their corresponding real IOCs when they become available, and without having to modify core simulation code, since it is implemented separately in VACA.

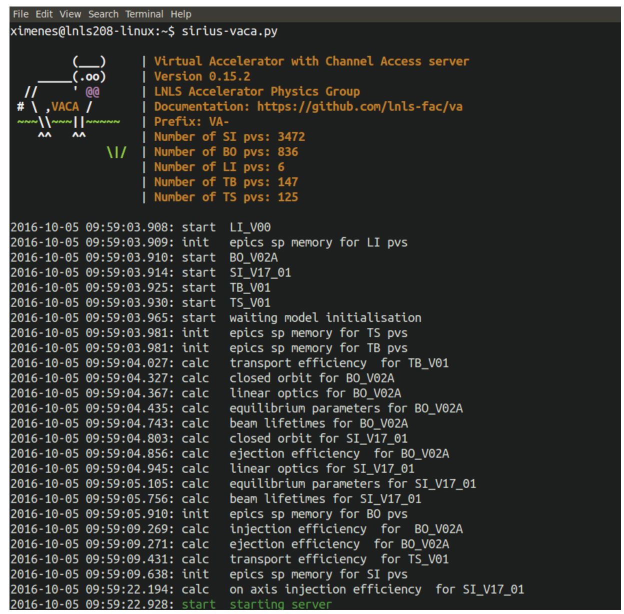

Figure 1: Screen printout of a command line terminal showing a running instance of VACA with the display of its banner and useful information.

VACA

The virtual accelerator with channel access is written in Python3. The choice of a high level programming language allows for rapid development of the intricate functionalities the virtual accelerator has to provide. Python language works as a binding layer between the two main core modules: the CA server and the tracking simulation code. The Python package PCASPy [3] is used to write the CA server module. PVs implemented in VACA have a prefix "VA-" indicating that they represent virtual process variables. For the simulation module, a library called trackcpp [4], that has been developed at LNLS by the accelerator physics group staff, was reused. It is a C++ library of beam dynamics calculations and tracking routines closely based on, and tested against, Tracy [5] and Matlab AT [6] passmethods. This library is converted to a python package using Swig3.0 [7], thus conveying fast and optimised routines that can be conveniently called within Python.

* ximenes.resende@lnls.br

ISBN 978-3-95450-189-2

Figure 1 is a screen print of a command line terminal showing a running instance of VACA with its banner and useful information printout. It shows the number of virtual PVs implemented for each major accelerator subsystem, labeled here SI,BO,LI,TB,TS. Also displayed are current model versions for all subsystems, as well as messages indicating that initial calculations are finished, such as those needed to simulate injection efficiency, for example. At last in the display, a message indicating that VACA is ready to respond to virtual PV queries.

VIOCS

The other part of the virtual accelerator is the set of virtual IOCs that respond to PVs with actual the control system names. So far a few vIOCS have been implemented:

- `si_bpm`, `bo_bpm`, `ts_bpm`, `tb_bpm`: they serve BPM positions for all subsystems that are read from VACA, adding emulated measurement fluctuations.
- `si_current`, `bo_current`: they provide simulated beam currents with fluctuations. Touschek, elastic and inelastic simulated lifetimes are affected by variations of associated parameters such as RF gap voltage and reduced acceptance due to closed orbit variations.
- `si_ps`,`bo_ps`,`ts_ps`,`tb_ps`: provide read/write access to PVs that correspond to power supplies with associated magnet excitation curves.
- `si_rf`,`bo_rf`: implement radio frequency process variables.
- `si_tune`: emulation of the tune measurement IOC.
- `si_beamsize, bo_beamsize`: emulation of beam size measurement IOC.
- `si_lifetime`: emulation of lifetime calculation IOC.

Most of these vIOCS are written using database records that are distributed with EPICS base. `si_lifetime`, on the other hand, is implemented with PCASPy.

CONCLUSIONS

The virtual accelerator described here has been an invaluable asset for the development of high level applications, as described in Ref. [2]. The core of the HLA development is planned to take place in 2017, mainly by the accelerator physics group staff. This development will certainly benefit from having a VA system available. At this point simulation of basic beam processes are implemented. VACA now properly simulates processes such as: parameter-dependent current decays, closed-orbit control with dipolar correctors, beam optics variations with quadrupoles, injection and ejection that depend on magnet and timing configurations. In the future more functionalities and modifications should be added to VA, such as:

- Timing details of the pulsed signals during injection and ejection processes need be considered.
- Approximate coupling expressions now used for beam size estimates should be replaced with more rigorous Ohmi's beam envelop formalism [8] in trackcpp,
- A cleaner separation between VACA and vIOCS is in order. At this points a few excitation curves are implemented in VACA since it has not been decided yet where they will finally be located in the CS. They can either be stored in the corresponding power supply IOCs – in which case they should be moved to the vIOCS for the VA, or stored in some configuration database service.
- Considerations on moving from EPICS database records PCASPy for vIOCS developments. This may simplify the process of writing and deploying applications.
- Recently a few DISCS [9] services have been adopted. In particular, the use of its naming service module allowed for a standardisation of how devices and PVs are named. As a consequence, a major revision of PV names has taken place recently. VA should be updated to contemplate the new PV naming standard.

REFERENCES

[1] A. R. D. Rodrigues *et al.*, “Sirius Status Report”, in *Proc. IPAC'16*, Busan, Korea, May 2016, paper WEPOW001, pp. 2811-2814.

[2] I. Stevani, N. Milas, X. R. Resende, L. N. P. Vilela, “High Level Applications for Sirius”, presented at PCaPAC'16, Campinas, Brazil, Oct. 2016, paper WEPOPRPO22, this conference.

[3] PCASpy - EPICS Portable Channel Access Server python package, `http://pcaspy.readthedocs.io/en/latest`

[4] LNLS Accelerator Physics Group, Tracking library, `https://github.com/lnls-fac/trackcpp`

[5] H. Nishimura, "TRACY, A Tool for Accelerator Design and Analysis", in Proc. EPAC'88, Rome, Italy, Jun. 1988, pp. 803-805.

[6] A. Terebilo, "Accelerator Toolbox for MATLAB", SLAC-PUB-8732, May 2001.

[7] SWIG - Simplified Wrapper and Interface Generator, `http://www.swig.org/`

[8] K. Ohmi, K. Hirata, K. Oide, "From the beam-envelope matrix to synchrotron-radiation integrals" Phys. Rev. E 49, 751, 1994.

[9] V. Vuppala *et al.*, "Distributed Information Services for Control Systems", in *Proc. ICALEPCS'2013*, San Francisco, United States, March 2014, paper WECOBA02, pp. 1000-1003

ISBN 978-3-95450-189-2