CLARA VIRTUAL ACCELERATOR

R.F.Clarke, G.Cox, M.D.Hancock, P.W.Heath, B.G.Martlew, A.Oates, P.H.Owens, W.Smith,
J.T.G.Wilson, STFC Daresbury Laboratory, Warrington, UK

Abstract

STFC Daresbury Laboratory is developing a novel Free Electron Laser (FEL) test facility focussed on the generation of ultra-short photon pulses of coherent light with high levels of stability and synchronisation. The main motivation for CLARA[1] is to test new FEL schemes that can later be implemented on existing and future short wavelength FELs. Particular focus will be on ultra-short pulse generation, pulse stability, and synchronisation with external sources. Knowledge gained from the development and operation of CLARA will inform the aims and design of a future UK-XFEL. To aid in the development of high level physics software, EPICS, a distributed controls framework, and ASTRA, a particle tracking code have been combined to simulate the facility as a virtual accelerator.

This paper will discuss how a EPICS [2], a distributed controls framework, and ASTRA [3], a particle tracking code, are used to simulate electron bunches in a virtual accelerator. The simulation is currently used by the accelerator physics group as a rapid application development framework for high level physics software. It is also being used by the controls group to test high level interfaces to CLARA. The simulation contains magnets; diagnostic screens; BPMs and camera beam positions; and low level RF. The next stage is to develop a more detailed simulation for the cameras and BPMs. Another objective is to distribute the transport code to external servers to increase the number of simulated particles.

INTRODUCTION

The Compact Linear Accelerator for Research and Applications, CLARA [1] is being developed using Daresbury's existing expertise and experimental experience of electron accelerators and FELs. Of note are VELA, the Versatile Electron Linear Accelerator, which will share a common front-end with CLARA. Daresbury has also worked with an IR FEL on the superconducting accelerator ALICE [4, 5].

CLARA consists of several major subsystems that themselves contain many individual devices. The FEL will requires tight tolerances on all these devices to operate. A suite of high level physics software and automation will be needed to control the FEL. The controls and accelerator physics groups wanted the ability to test and develop concepts in simulation before they are deployed. These requirements drove the development of the virtual accelerator.

CLARA consists of timing, magnets, beam position monitors (BPM), beam shaping, vacuum control, RF systems, electron gun, environmental controls and lasers. The operation of all these devices is handled by EPICS, a distributed control framework. EPICS provides tools to develop custom device drivers and the control logic for connected devices under time critical constraints. EPICS also provides a uniform interface to manage operations over a dedicated controls network using process variables (PVs). Using PVs over the dedicated CLARA control network allows operator applications, data-logging, high level software or even other EPICS controlled devices to communicate and interact across the machine seamlessly.

The core of EPICS is a server called the IOC, see Figure 1. The IOC will signal its presence over a control networks to other EPICS compatible clients and manage PV requests. CLARA will have many tens of IOCs and thousands of PVs running simultaneously when the machine is in operation. This is itself a major challenge that the virtual accelerator is helping to address.

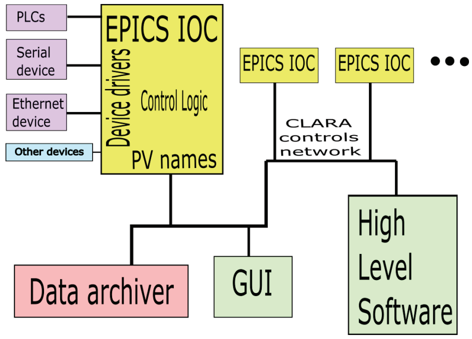

Figure 1: EPICS IOC and PVs.

PREVIOUS VIRTUAL ACCELERATORS: EMMA

Simulations of accelerators have been employed at KEKB [6], the SNS linac [7], Diamond [8], J-PARC [9], TPS, Taiwan [10].

Two main functions of a simulation tool were described by Yamamoto:

- For rapid application development (RAD)

"Tightly integrated modelling code in a control system, or a virtual accelerator, is also useful as a RAD tool in the construction of an accelerator control system. Application programmer and/or an accelerator physicist can develop a code without waiting for the completion of hardware installation. Realistic response using the VA will help the development of high level applications in the control system."

- A simulation as a "Flight Simulator"

"Trainee of the accelerator operation can learn the response of the system even in a situation which should not occur in the real machine. 'Flight simulator' allows accelerator physicist to examine his/her algorithm without disturbing the operation."

At EMMA the simulation tool was used for developing software but was not used for operations as there was a lack of manpower to complete model. It was found that the model results did not match the real machine but progress was made towards implementing a more model[11].

DISTRIBUTION, NETWORKING AND VERSION CONTROL

The virtual accelerator simulation runs on Linux under VirtualBox[12]. This allows all IOC setup and networking to be maintained by the controls group inside a virtual machine. To use the virtual accelerator, the virtual machine is downloaded and installed on a local PC. It automatically starts all IOCs and network configurations in the background on startup. EPICS clients, such as PyEPICS[13] can then access the virtual accelerator on the host PC.

PV names must be unique so this must be checked when adding new IOCs. The virtual accelerator runs its own IOCs supplying virtual PVs. These are direct copies of the CLARA PVs. Once these simulations are out in the wild there is the possibility that a machine running a simulation could accidentally be connected to the CLARA controls network. In the worst case virtual PVs could be confused for CLARA PVs. A naming clash, in any event, would cause peculiar behaviour during operation and be hard to diagnose. The accelerator physics group also requested the ability to run multiple simulated IOCs on the same virtual network raising the possibilities of further naming clashes. Spawning multiple simulations would cause naming clashes inside and across virtual machines running on the host PC.

The first release used the VirtualBox virtual network card. A “host only card” can be selected that will keep all internet connections from the virtual machine within the host PC. This worked well initially but the virtual machine did not have a connection to the host’s network card. It was found that the virtual PVs were still visible outside the machine when the virtual machine was allowed to connect to the network card no matter how the VirtualBox configuration was set.

The solution was to use EPICS version 3.15. In this version it is possible to restrict an IOC to only respond to PVs that originate from a specific IP address. The IPs of the virtual accelerator can be set so they do not respond to requests sent form the CLARA control network. This provides the first level of guard incase the virtual accelerator is connected the CLARA network.

An IOC is also locked to a specified port numbers. This provides additional guard against accidental connection and also allows multiple virtual IOCs to run under the same virtual machine. For IOCs on CLARA the default port is 5064 and a beacon is transmitted on port 5065. The virtual IOCs are set to run on ports greater than 6000 by default with no beacon. If the user wants multiple instances of virtual IOCs running they can simply change the port numbers in the startup scripts.

A final guard was added by prefixing all the simulation PVs with “VM_”. This is now logged as part of the CLARA naming convention. See figure 2.
The typical way for a script to access a real PV would look like:

```
EPICS_CA_ADDR_LIST=<CLARA IP XXX.XXX.XXX.XXX>
caget("CLA-S02-DIA-SCR-01:V:ACTPOS")
```

Then simulation PVs on server 1 and 2 would be accessed as:

```
Server1, VM on port 6002

EPICS_CA_ADDR_LIST=<SERVER1 XXX.XXX.XXX.XXX>
EPICS_CA_SERVER_PORT=6002
caget(VM_CLA-S02-DIA-SCR-01:V:ACTPOS)

Server1, VM on port 6003

EPICS_CA_SERVER_PORT=6003
caget(VM_CLA-S02-DIA-SCR-01:V:ACTPOS)

Server2, VM on port 6002

EPICS_CA_ADDR_LIST=<SERVER2 XXX.XXX.XXX.XXX>
EPICS_CA_SERVER_PORT=6003
caget(VM_CLA-S02-DIA-SCR-01:V:ACTPOS)
```

Git hub

For the initial version of the virtual machine software updates were simply uploaded as a new VM. After the simulation was released to the physics group the number of users rapidly increased and this method became unsustainable. Github was picked for software version control.

Currently, there is of the virtual accelerator that has the git repository set up. Users download this version, open the virtual machine on their PC. They can then use github to pull the latest updates from the controls group server. This is a reliable and fast way to distribute updates as the user-base increases.

SIMULATIONS FOR HIGH LEVEL SOFTWARE AND TRANSPORT CODES

The virtual accelerator simulates various devices and feeds them into the virtual EPICS control system providing identical PVs to CLARA. The virtual accelerator has been combined with a particle transport code called ASTRA to simulate the electron beam. Other simulation codes and custom scripts are added to ASTRA under a custom framework called the online model.

The virtual accelerator does not need the online model for most investigations. The values for basic PV operations such as axis positions, magnet current settings and interlock have simplified operating modes that can be used to develop

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA204

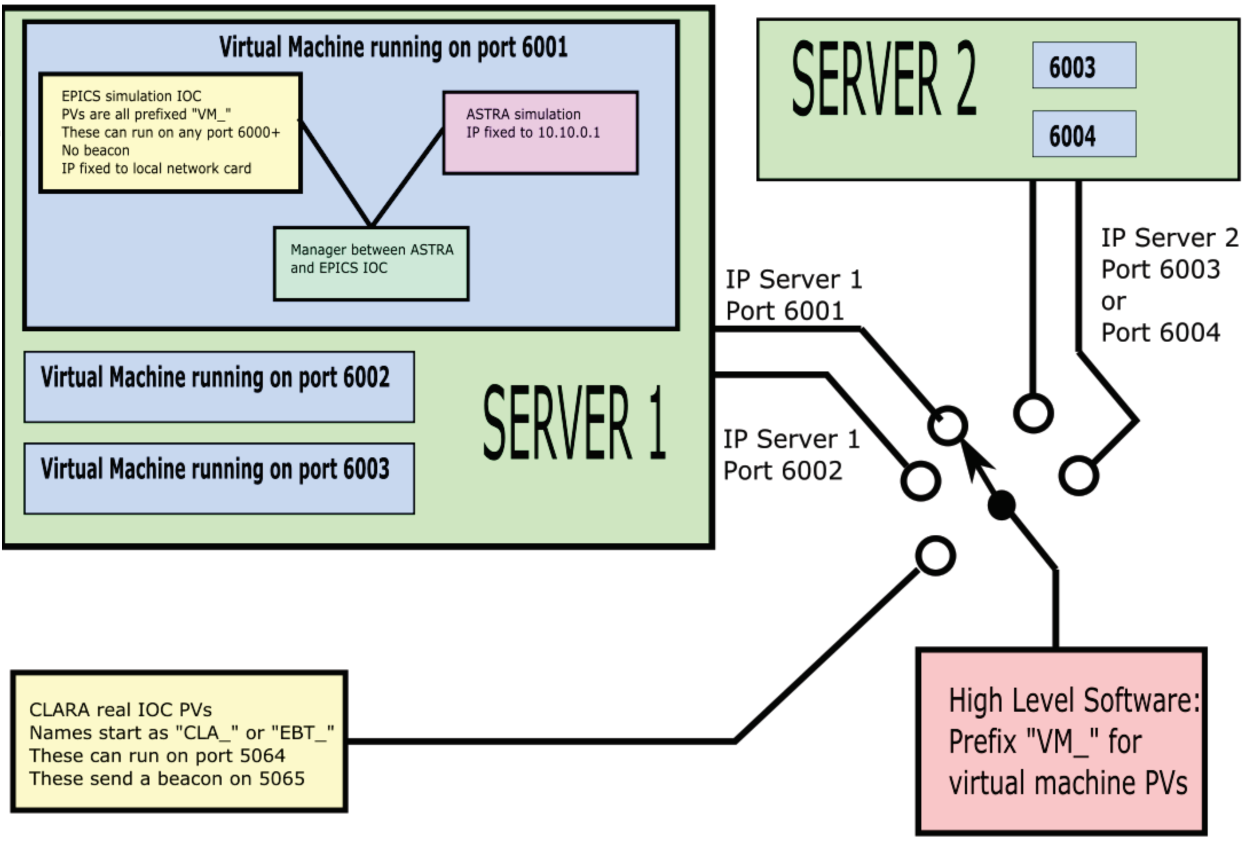

Figure 2: The high level software can select from simulations running on different IPs or ports and from CLARA PVs. In the worst case scenario the “VM_” prefix will protect the real machine’s IOC PVs.

graphical user interfaces while CLARA is under construction. Another example involved the use of virtual magnets. The magnet current can be set to ramp up and have a preset noise level. This was used to for the offline development of a custom PVs for the CLARA/VELA magnets. These PV were part of a new interface that is now commissioned on the real machine. This interface allows for the automatic degaussing of all the CLARA/VELA magnets via PVs sent from an operator in the control room. The operator triggers the degauss operation on the relevant IOC in the rack-room. The IOC starts a state machines which checks if the magnet needs to be degaussed, monitors various interlocks and checks if any devices would be affected by the operation. The IOC runs the desired degauss algorithm, records when the degauss completes or any errors that occurred and reports back, via PVs over the controls network. All the complexity of degaussing the magnet manually has been hidden from the operator. CLARA and VELA have largely different magnet power supplies and interlocks and the operation has to be customised for each magnet type. The operators want the specifics of the operation to be hidden and simply be able to trigger them. The virtual machine was used to test the sequence of operations and to speed up development by performing magnet ramps faster than real-time. The virtual accelerator also allowed the response to various power supply interlock triggers to be simulated.

Adding the online model in combination with EPICS also provides useful information for the development of the BPM and camera interfaces. An example is trajectory monitoring solutions such as orbit beam corrections.

The virtual accelerator currently contains the CLARA phase 1 and VELA magnets, diagnostic screens, BPMs, gun and low level RF. The next goal is to develop a more powerful simulation for the cameras and BPMs. Currently the online model typically simulates 250pC bunches. These are made of about 1000 macro-particles and take roughly 10 seconds to track on a standard PC. The next iteration of the virtual accelerator will run the online model separately allowing simulations of the order of 10^3 or 10^5 more macro-particles.

OVERVIEW AND FUTURE OF THE CLARA SIMULATION

Current Status

- The virtual accelerator is under Git version control
- The virtual machine can run as many IOCs as needed with accidental access to the CLARA control network

blocked under EPICS 3.15. The current EPICS version on CLARA is being upgraded from version 3.14 to 3.15

- The first device PVs to be brought into the virtual accelerator were for the magnets. These were run without electron transport simulations and used a basic algorithm to generate online ramping noise, see figure 3. This initial survey of all the magnets allowed us to identify that the legacy VELA PV names and new CLARA names were inconsistent. A program to update the legacy PVs was started on these machines and a small set of high level interface PVs are being developed for use with scripts and GUI design. The virtual accelerator was also used to test and build state machines that tie different magnet power supply units together for the high level interface. Using the virtual accelerator allows rapid tests on error conditions and state changes that are hard or impossible to generate with the real IOCs.
- The BPMs and camera positions have been added. The RF and magnets provide ASTRA and the online model with the initial conditions to generate electron trajectories. These trajectories are fed back to the virtual BPM and cameras as beam positions. These simulations are being used for trajectory planning such as orbit beam corrections for the accelerator physics group.
- The various motion controls for the diagnostic screens were brought under a common interface. Some of the devices had several diagnostic devices on a single axis and some are dual axis devices. Some devices are set on the horizontal and vertical plane and will clash if both are extended together. To set a device of a specific axis the conjugate axis needs to be retracted. Other motion controls were simple pneumatic "pop in/pop out" devices.
- A standard interface for all these devices was developed within the virtual accelerator before being deployed on the real machine. One of the benefits of this interface was to simplify the ability to script scans of the beam by jogging the collimators or slits. A state machines was developed to control the positioning of device on an axis and to manage the conjugate axes. Testing the state machines in the virtual accelerator was much faster, much like the magnet case, as axis motion could be speeded up and error states triggered to test the response.

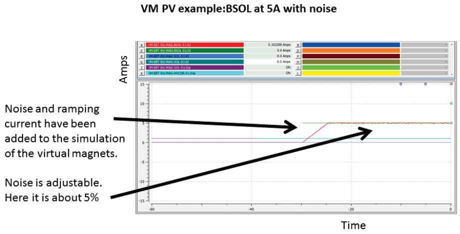

Figure 3: Simulation of the magnets using only noise and ramping for GUI development.

Future Work

- The online model simulations will benefit from more computing power than provided by a single PC for some concepts. Currently it is possible to simulate 250pC bunches at about 1000 macro particles. Having the online model on dedicated, high-performance servers will allow several orders of magnitude increase in simulation of particles. Different users will require different ASTRA and online simulation set-ups. The latest version of the virtual machine has been modified to allow networked connections for external simulations.
- The BPMs and cameras will be modified to provide more realistic simulations such as noise.

ACKNOWLEDGEMENTS

With thanks to ASTeC, STFC Daresbury Laboratory.

REFERENCES

[1] J. Clarke *et al.*, "The conceptual design of clara, a novel fel test facility for ultra-short pulse generation," pp. 496–501, Jan. 2013.

[2] Argonne National Laboratory. (2015). Experimental physics and industrial control system, `http://www.aps.anl.gov/epics/`

[3] K. Floettmann. (2016). A space charge tracking algorithm astra, `http://www.desy.de/~mpyflo/`

[4] F. Jackson *et al.*, "The status of the alice r&d facility at stfc daresbury laboratory," in *IPAC 2011 - 2nd International Particle Accelerator Conference*, Sep. 2011.

[5] ASTeC. (2015). The versatile electron linear accelerator (vela), `https://www.astec.stfc.ac.uk/Pages/The-Versatile-Electron-Linear-Accelerator-(VELA).aspx`

[6] Yamamoto et al. (1997). Use of a virtual accelerator for a development of an accelerator control system, `http://accelconf.web.cern.ch/accelconf/pac97/papers/pdf/3P042.PDF`

[7] A. Shishlo, P. Chu, J. Galambos, and T. A. Pelaia, "The EPICS Based Virtual Accelerator - Concept and Implementation," *Conf. Proc.*, vol. C030512, p. 2366, 2003.

[8] P. Goryl, A. Wawrzyniak, M. Sjöström, and T. Szymocha, "An Implementation of the Virtual Accelerator in the Tango Control System," in *Proceedings, 11th International Computational Accelerator Physics Conference (ICAP 2012): Rostock-Warnemünde, Germany, August 19-24, 2012*, 2012, MOSBC3. `http://jacow.org/ICAP2012/papers/mosbc3.pdf`

[9] H. Harada *et al.*, "Current status of virtual accelerator at j-parc 3 gev rapid cycling synchrotron," in *2007 IEEE Particle Accelerator Conference (PAC)*, Jun. 2007, pp. 215–217. DOI: `10.1109/PAC.2007.4440163`.

[10] P. C. Chiu, C. H. Kuo, Jenny Chen, Y. S. Cheng, C. Y. Wu, Y. K. Chen, K.T. Hsu. (2010). Virtual Accelerator Development for the TPS, `https://accelconf.web.cern.ch/accelconf/IPAC10/papers/wepeb019.pdf`

[11] D. Kelliher, "Accelerator software interest group iii," Feb. 2015.

[12] Oracle. (2017). Virtualbox, `https://www.virtualbox.org/wiki/Downloads`

[13] University of Chicago. (2016). Pyepics, `http://cars9.uchicago.edu/software/python/pyepics3/`