

# THE EPICS BASED VIRTUAL ACCELERATOR – CONCEPT AND IMPLEMENTATION\*

A. Shishlo\*\*, P. Chu, J. Galambos, T. Pelaia, ORNL, Oak Ridge, TN 37830, USA

## Abstract

A virtual accelerator (VA) concept and an implementation founded on TRACE3D and PARMILA codes are presented. This virtual accelerator is suitable for accelerators with a control system based on EPICS and consists of the EPICS portable channel access server (PCAS), the EPICS client providing communication between a simulation model and PCAS, and the simulation model itself. The virtual accelerators for the SNS linac and experience in using these VAs are discussed.

## COMPONENTS OF A VIRTUAL ACCELERATOR

Development of physics applications for an accelerator that does not yet exist presents a nontrivial task (the SNS is an example). In the case of a control system based on EPICS [1] there is an elegant solution facilitating this task. It is possible to create a model imitating the real machine by using the EPICS PCAS and one of the existing simulation programs. This combination of PCAS and a simulation program has been termed the “virtual accelerator” because it looks like a real machine from the EPICS channel access view, operates with real process variable (PV) names, and produces a reasonable response generated by the simulation model. To provide the network communication between the simulation model and EPICS PCAS we need an EPICS channel access client (third part of virtual accelerator). The structure of VA is shown on Fig. 1.

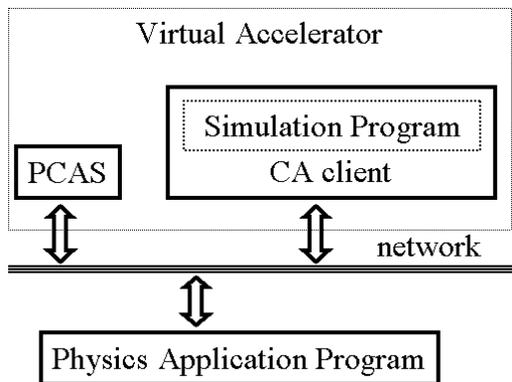


Figure 1. The structure of the virtual accelerator.

A physics application program (see Fig. 1) is a program that is supposed to work with a real machine and communicates only with the PCAS. It is not a part of the

\* SNS is managed by UT-Battelle, LLC, under contract DE-AC05-00OR22725 for the U.S. Department of Energy. SNS is a partnership of six national laboratories: Argonne, Brookhaven, Jefferson, Lawrence Berkeley, Los Alamos, and Oak Ridge.  
\*\*shishlo@sns.gov

virtual accelerator.

There are certain differences between the real and virtual accelerators. First, the simulation program is not as fast as the real machine, therefore the time delay parameters (if any) in the physics application should be increased. Second, the quality of response is defined by the accuracy of the model in the description of the physics of the real accelerator. An advantage of the VA is that the developers can have as many VAs as they need.

## Portable Channel Access Server

The PCAS has been built on the base of the EPICS Release 3.14.1 Portable Server library. For the sake of simplicity PCAS implements only classes related to the synchronous EPICS process variables (PVs) because it is enough for our purposes. It means that our PCAS extends only three abstract base classes from the EPICS channel access server (CAS) library: caServer, casPV, and casChannel. The initialization of the server is performed by using a simple ASCII file that can be prepared by hand or can be generated by the client part of the virtual accelerator. It includes information about EPICS environment variables and the PV's description. In addition to the PV's name the user can specify the lower and upper limits for PV's value, a size for vector PVs, a random error spread for PV's value, and period for applying random errors to simulate noise.

This PCAS has nothing specific for the virtual accelerator and can be used as a usual EPICS portable channel access server.

## Simulation Program

A simulation program for the virtual accelerator should meet several criteria:

- It should be reliable in producing the reasonable physical results.
- It has to be fast enough. The calculation time for the considered part of accelerator should be about several seconds. If it is not we cannot talk about interactivity of the model.
- It should be capable to calculate physical values that determine diagnostic signals necessary for the particular physics application program.

At this time, there are two types of VAs. The first type is based on TRACE3D [2], and the second one uses the PARMILA [3] simulation program. These codes are well known and widely used.

Usually the TRACE3D based VA is used to simulate a trajectory of the center of the beam in the SNS linac. TRACE3D code is fast, but it cannot describe beam losses, and so this VA cannot produce diagnostic signals related to the intensity of the beam.

PARMILA is the linac design and beam dynamics simulation code. However, the VA uses its simulation

capabilities only. It is more complex and slower than TRACE3D, but it enables us to simulate the beam position monitor's (BPM) amplitude and phase values, and signals from both the beam current monitors and wire scanners.

### CA Client

The CA client part of the virtual accelerator provides exchange of information between PCAS and the simulation code. It gets PV's values from PCAS, transforms them if it is necessary, and puts them as input parameters into the simulation program. After the simulation code finishes calculation, the CA client should extract diagnostic information and put it into the corresponding channel access server's PVs. These actions put into "loop forever" will produce an impression of the live accelerator if the user deals with PVs on the PCAS only. There are two extremely different ways to implement this algorithm.

In the first way the CA client deals with the input and output files of the simulation program only. In the beginning of the calculation loop, the CA client has to prepare the simulation input file with the values corresponding to the PV values from the PCAS at the appropriate places in this file. Then it has to start the simulation program executable file by using the system call. After the simulation code finishes the calculation and dumps output files onto the hard drive, the CA client has to analyze these output files, extract necessary diagnostic information, and put this information into PCAS. This approach does not require any modification in the simulation code, and this is a big advantage of this approach. On the other hand, parsing of the output files can be difficult, and these files may not contain all of the necessary information. For instance, none of the existing simulation programs generates BPM phases and amplitudes.

In the second way the CA client and the simulation code are integrated into one executable and do not use intermediate files for information exchange. For this case, the source code of a simulation program has to be modified to include methods for input data definition and extraction of diagnostic information. These new methods should not interfere with the original source code, so the results of calculations for modified and unmodified simulation programs have to be the same. This approach provides greater flexibility in producing the simulated diagnostic signals, because the CA client has access to all internal information in the simulation code. It enables us to create a new diagnostic method for each of the necessary diagnostic values. The disadvantage of this approach is that the developer should have knowledge about not only input and output files of the simulation program, but also about the details of the program implementation. Therefore, this approach can be used only for a relatively simple simulation code.

In the described virtual accelerators, the second method is used in the VA based on the TRACE3D code and a combination of both methods for the PARMILA based

VA. The different approaches were used because TRACE3D has a very simple structure for input data describing an accelerator lattice and can be easily modified.

### TRACE3D CA CLIENT

The structure of the CA client based on the TRACE3D code is shown on Fig. 2.

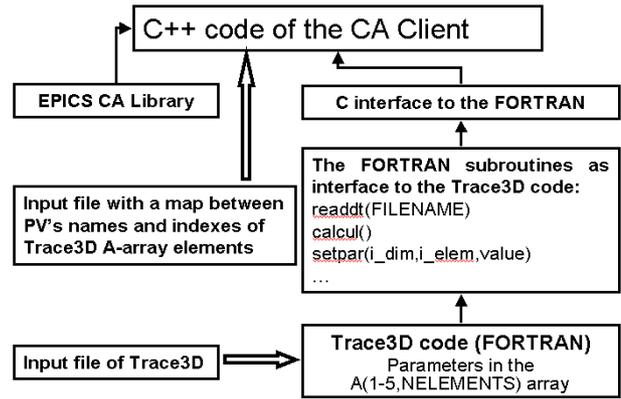


Figure 2. The structure of the CA client and simulation program for the TRACE3D based VA.

As it is said above, the source code of TRACE3D has been modified. The main program has been transformed into the FORTRAN function "calcul" and the part of code that reads the TRACE3D input data has been moved into the "readdt" function. The call of new function that grabs coordinates of the centroid of the beam has been inserted into the main loop of the TRACE3D, which passes through all elements of the accelerator lattice.

A number of new FORTRAN subroutines that work with the TRACE3D data structure were added. They provide the ability to set and read values inside the 2D A-array where TRACE3D keeps all data about the accelerator's lattice elements. Thus, the CA client can synchronize this data with PVs in the PCAS.

There are two files needed to initialize this CA client. The first file is a usual unmodified TRACE3D input file, and the second one is an ASCII map file with PV names for "set" and "read back" variables and indexes of the A-array associated with these PVs. The second file also includes necessary parameters to transform PCAS PV values to the TRACE3D values such as coefficients between current and field strengths for electromagnets.

The algorithm of the calculation cycle of the TRACE3D based CA client is revealed in Fig. 3. In the beginning, it reads data from two input files and initializes the internal structures. Then, it puts initial values into PCAS PVs in accordance with the TRACE3D input file data. Eventually, it starts the main loop of calculations. During this loop, the CA client reads data from PCAS, puts them into the TRACE3D data structure, starts TRACE3D, and places calculated BPM responses into PCAS. There are beam center initial coordinates PVs in PCAS that do not exist in a real accelerator and are

initialized from the TRACE3D input file, but they enable outside applications to change these coordinates without interruption of the VA calculation cycle if necessary.

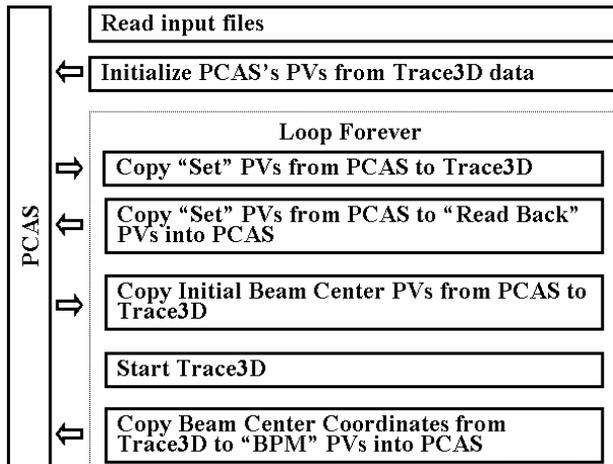


Figure 3. The algorithm of calculations of the TRACE3D based CA client.

### Using of the TRACE3D Based CA Client

The TRACE3D based VA is mostly used in the development and debugging of orbit correction application programs. It is very suitable for this purposes, because it is very fast. It takes less than one second to get the beam center trajectory for the SNS MEBT part of the SNS linac that includes about 100 lattice elements.

## PARMILA CA CLIENT

The structure of the PARMILA based CA client is similar to the one based on TRACE3D (see Fig. 2.). It includes the modified PARMILA FORTRAN code, additional FORTRAN subroutines providing access to the data about coordinates of macro-particles, and the C interface to the FORTRAN subroutines that enable the CA client to control PARMILA calculations. In either case, one executable file for the CA client is created.

The difference is that the PARMILA based CA client does not work with data structures describing an accelerator lattice directly. Instead, it reads the input PARMILA file with special marks on the places where parameters from PCAS should be placed, keeps this file in memory, at necessary moments edits this file replacing marks by numbers, and dumps the new version of the file onto the hard drive as the new input file for PARMILA. The algorithm of the PARMILA based CA client is shown in Fig. 4. This approach demands minimal changes in the PARMILA source code; only a few lines with calls to the external diagnostic routine were added. This diagnostic routine has access to the inner PARMILA array where the 6D coordinates of the macro-particles are kept. Using this information the diagnostic method calculates average x and y coordinates that are x and y BPM signals, BPM's

amplitudes and phases (the second harmonic of FFT of the longitudinal macro-particles distribution), and the wire scanner signals (the RMS distributions for each of the x and y coordinates).

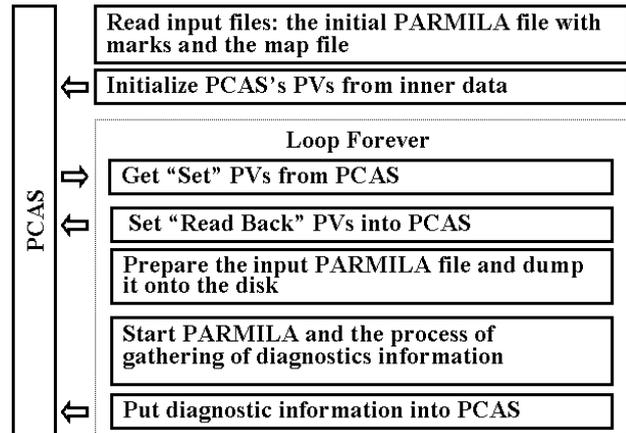


Figure 4. The structure of the CA client and simulation program for the PARMILA based VA.

### Using of the PARMILA Based CA Client

The PARMILA based VA provides more realistic simulation of the diagnostic signals, but it is slower than the TRACE3D based one. It takes several or several tens seconds to calculate the propagation of macro-particles through the MEBT. The calculation time depends on the number of macro-particles and the space-charge treatment approach. This VA is used in development and debugging of the RF cavities tuning applications.

## SUMMARY

Using EPICS as a base for the control system of an accelerator provides the ability to create realistic simulation software (VA) that can be used instead of the real machine during development of the physics application programs. The combination of the portable channel access server and the accelerator simulation code with the channel access client has been called a "virtual accelerator". These VAs are widely used in application development for SNS and appear to be a useful tool.

## ACKNOWLEDGEMENT

Authors thank E. Tanke and S. Aleksandrov (ORNL) for many useful discussions during the VA development.

## REFERENCES

- [1] <http://www.aps.anl.gov/epics>.
- [2] D.P. Rusthoi K.R. Crandall. TRACE 3-D Documentation, LA-UR-97-886. LANL, 1997
- [3] Harunori Takeda. PARMILA, LA-UR-98-4478. LANL, 1999.