

HIGH-LEVEL APPLICATION DEVELOPMENT AND PRODUCTION INFRASTRUCTURE AT TRIUMF

E. Tikhomolov, Y. Bylinskiy, A. C. Morton*, T. Planche, T. Tateyama, J. Lee, P. Jung†
TRIUMF, Vancouver, BC, Canada

Abstract

TRIUMF users and operators use a number of high-level applications (HLAs) written in different languages, with complicated graphical user interfaces, to carry out tasks related to delivering ion beams with required characteristics and to process data from TRIUMF's EPICS-based and legacy cyclotron control systems. Some applications have been developed by the EPICS community, and some at TRIUMF. These applications run on different production computers and are developed on different machines. This model no longer satisfies TRIUMF's needs because of the growing number of applications, the long times required for data processing on current machines, the lack of real-time visualization of beam properties and so on. New infrastructure for HLA development has been implemented to address these issues and is working reliably with room for further expansion.

MOTIVATIONS

TRIUMF doesn't have a dedicated group of software developers tasked with immediate response to issues that arise during day-to-day beam delivery. Such issues are resolved by operators and physicists themselves. Thus, a flexible and simple ("user-friendly") software development environment was the main request when it was decided to set up High-Level Application (HLA) development and production infrastructure. At the same time, developers come with different experience, backgrounds and their own favorite tools for development. Thus, the HLA environment should provide some "default set" of tools which are rather common and compatible with other widely-used tools.

HLA SERVERS SET UP AND THEIR DIFFERENT ROLES

The projects that are under development at TRIUMF usually have three components: the code itself (in a number of programming languages), documentation and information, and input/output data. Thus, the common project can be represented as a point in a "three-dimensional space" where each component defines an axis, as illustrated in Fig. 1.

Some projects are just data projects which process experimental data (from TRIUMF's 520 MeV Cyclotron and Isotope Separator and Accelerator facility, or ISAC) for use with third-party applications. Each component has a corresponding location and directory structure: the code is developed on a development server (hladevel), input/output

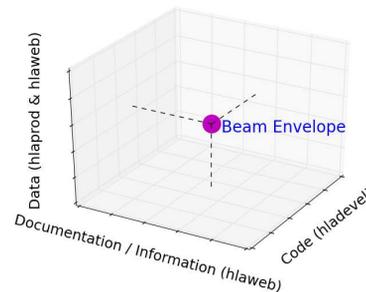


Figure 1: Common HLA project in "3D space".

data are saved on production servers (hlaprod and hlaweb), and documentation and information are maintained on a web server (hlaweb [1]).

To fulfil these tasks rack-mounted Dell servers were chosen [2]. All three servers have identical directories and software. This reduces maintenance time and allows one to easily switch from one server to another in the event of hardware failure. This approach also allows the maintainer to easily expand the infrastructure by cloning the existing servers, and redistributing the load. The differences between the servers are defined only by their different roles. Figure 2 shows these roles and the interactions between servers and user machines and control systems. The HLA development server hladevel and production server hlaprod are accessible only from the TRIUMF network. The web server hlaweb is accessible via the Internet and users can ssh to their accounts. Users have scratch directories (public_html) which may be used e.g. for presenting results at meetings. To reduce security threats it is not possible to connect to any TRIUMF computers from user accounts on hlaweb.

The data and user's scratch directories on the development server, hladevel, are synchronized (using rsync) with hlaweb every 4 hours. Data directories from hlaprod are also synchronized to hladevel. This duplication of data helps to avoid data loss. Nightly backups are done only on hladevel both on local USB drives (for fast restoration) and to a remote location (using TRIUMF's Amanda system [3]).

For version control Git [4] is used. Public and private remote repositories are located on hlaweb and hladevel, respectively. For projects which are expected to involve large number of inter-lab developers Github [5] may also be used.

CentOS 7 was chosen as the operating system for the HLA servers. This is already used at TRIUMF and supported by TRIUMF's Core Networking and Computing Group. To reduce the time spent on system administration the standard tools and applications provided with CentOS 7, at installa-

* Present affiliation: FRIB, MSU, East Lansing, MI, USA

† Present affiliation: Waterloo University, Waterloo, ON, Canada

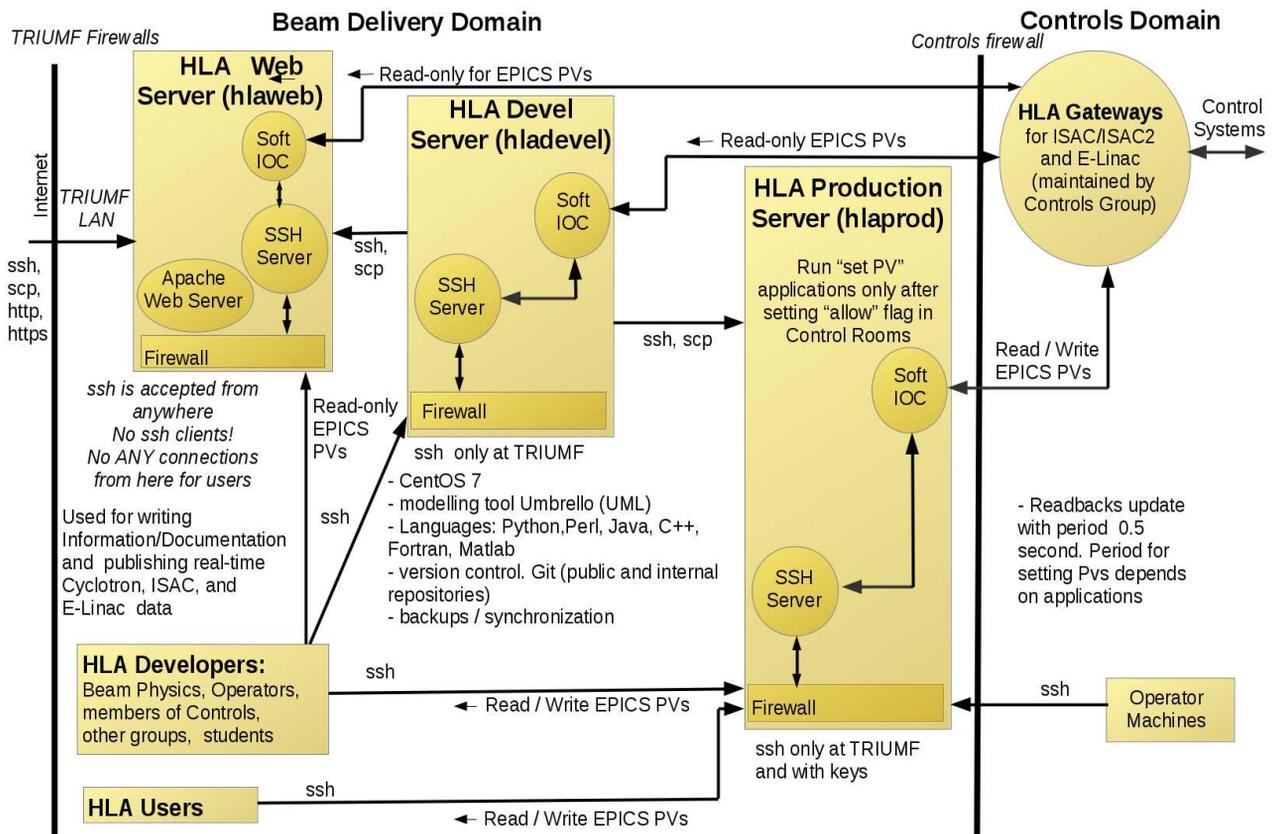


Figure 2: HLA servers and their roles.

tion, are used as much as possible. Users can start remote light-weight Xfce [6] desktops via the remote desktop application X2Go [7]. Graphics-intensive applications (like Matlab [8], Opera [9]) which run very slowly via X-forwarding are started in X2Go session in the "Single Application" mode. Usual tasks like synchronization and backups are done by using short customized scripts.

HLA DEVELOPMENT FLOW

High-level design of any large project starts with creating Unified Modeling Language (UML) diagrams. In our dynamic environment it is very important to create detailed high-level documentation which can be understood by future developers who may not have direct contact with the initial creator of the project. As the default, UML modeller Umbrello [10] is suggested; this comes with standard CentOS installation, and is relatively easy to learn and use. The choice of programming language is based on an evaluation of what is most appropriate for a particular project and on the experience of the current maintainers of similar projects. At present projects are written in Python, Perl, Java, C++, Fortran, and Matlab. For web applications, client side scripting is implemented in JavaScript and JQuery. The purpose of many HLAs is to get and set data from the TRIUMF control systems to collect and process useful analytics and hence improve the performance of the Cyclotron, ISAC, E-Linac,

and, in future, Advanced Rare Isotope Laboratory (ARIEL) facilities.

Any information and documentation about the project is kept on the HLA web server hlaweb. To write information on line an in-house content management system (CMS) using TinyMCE JavaScript package [11] was developed and set up. The frame-based design of web pages is shown in Fig. 3. All modifications are done by clicking buttons in the Edit Bar, access to which is password-protected. When a new project is started, new directories, a frameset and initial Hypertext Markup Language (HTML) files are created with a single click. Editing Side Bar and Information HTML files is done using either TinyMCE or the simple textarea text editor. All of the necessary links for navigating between HTML files are created automatically in a hierarchical and easily maintainable manner. Navigation to necessary information is very efficient using the Navigation and Side Bars. No relational databases are used. For users who have accounts, it is also possible to create, edit, and link files over a SSH or remote desktop session by manually modifying them directly.

INTERFACE TO TRIUMF EPICS

Control systems at TRIUMF have both new and legacy components, Experimental Physics and Industrial Control Systems (EPICS [12]) for ISAC, E-Linac, ARIEL and the old VMS-based Central Control System for the 520 MeV

The screenshot shows a web browser displaying the HLA web interface. At the top, there is a navigation bar with buttons for 'HLA Homepage', 'HLA Projects', 'Beam Physics TRIUMF Homepage', 'Beam Physics Internal Server', 'Magnetic Measurements', 'Cyclotron Data', 'TRIUMF Beamlines', 'OLIS', and 'HLA Content Management System'. Below this, a message states 'This page was visited 203 times'. The main content area is divided into three sections: 'Side Bar' on the left, 'Information Pane' in the center, and 'Edit Bar' on the right. The 'Side Bar' contains a list of 'HLA PROJECTS' and 'COMMON INFORMATION' links. The 'Information Pane' displays a table of project information. The 'Edit Bar' contains buttons for 'Create Sub-Directory/FrameSet', 'Set Default Info', 'Create New Info File', 'Edit HTML as Text', 'Edit Info', 'Edit Side Bar', and 'Edit LaTeX Paper'.

Target Snapshot	ISAC	Chris Payre	Olivier Shelbaya/Spencer Kiy	Operators
Mass Scaling for OLIS	ISAC	 Evgeniy Tikhomolov	Controls Group	Controls
Heater Ramp	ISAC	?	Controls Group	Controls
Tune Display	ISAC	Olivier Shelbaya	Olivier Shelbaya	Operators
SC Linac Phasing Program (PSID)	ISAC	Students	Spencer Kiy	Operators

Figure 3: Frame-based design and editing web pages on hlaweb.

main cyclotron. Several years ago an interface between these legacy and new control systems was developed which can be used for data exchange between them. Thus, for HLAs we need only to communicate with EPICS servers. Interface between HLAs and EPICS is done in two steps (Fig. 2): HLA servers run local EPICS Soft IOCs which communicate with EPICS via gateways. Then HLAs get and set data from or in local Soft IOCs. Such a two-level design provides flexibility and complete decoupling from TRIUMF's Controls Group, which maintains only the gateways. Setting values for control system devices is allowed only from the production server hlaprod and only after the operators in Control Rooms set an "allow" flag with a time-out of 1 hour. This way running HLAs are always under the control of operations staff. For other servers, read-only access to EPICS is allowed. At present TRIUMF control systems have typical update times for read-backs of 500 milliseconds. The desire to maintain a similar update rate for processed data from HLA determines the requirement for data processing speed.

CONCLUSION

The created HLA infrastructure allows software developers to develop applications both by using the default set of tools installed on HLA servers and remotely on the computers of individual users. The sequential versions of projects are saved on HLA servers and are easily accessible by developers. Documentation for projects can be written on-line and is maintained in a very simple way using a dedicated HLA CMS. Several methods are used to run HLAs (which work with devices in TRIUMF control systems) in a secure

manner. If the load on the servers will be increased the expansion of existing infrastructures is quite easy and won't create significant additional time for the maintenance.

ACKNOWLEDGMENT

This work is funded by TRIUMF under a contribution from the National Research Council of Canada.

REFERENCES

- [1] TRIUMF HLA WEB Sever, <http://hlaweb.triumf.ca>
- [2] Dell PowerEdge Servers, <http://www.dell.com/ca/business/p/poweredge-rack-servers?~ck=anav>
- [3] Amanda Network Backup, <http://www.amanda.org/>
- [4] Git, distributed version control system, <https://git-scm.com/>
- [5] Github, <https://github.com>
- [6] Xfce Desktop Environment, <https://www.xfce.org/>
- [7] X2Go - everywherehome, <http://wiki.x2go.org/doku.php>
- [8] Matlab. The Language of Technical Computing, <http://www.mathworks.com/products/matlab/>
- [9] Opera Simulation Software, <http://operafea.com/>
- [10] Umbrello, the UML Modeller, <https://umbrello.kde.org/>
- [11] TinyMCE, Web Editing, <https://www.tinymce.com/>
- [12] EPICS, <http://www.aps.anl.gov/epics/>