ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-MOAR02

MODERNIZING DIGITAL VIDEO SYSTEMS AT THE NATIONAL IGNITION FACILITY (NIF): SUCCESS STORIES, OPEN CHALLENGES AND FUTURE DIRECTIONS

V. Gopalan, A. Barnes, G. Brunton, J. Dixon, C. M. Estes, M. Fedorov,
M. Flegel, B. Hackel, D. Koning, S. Townsend, D. Tucker, J. Vaher,
Lawrence Livermore National Laboratory, P.O. Box 808, Livermore, CA 94550, USA

Abstract

The National Ignition Facility (NIF), the world's most energetic laser, completed a multi-year project for migrating control software platforms from Ada to Java in 2019. Following that work, a technology refresh of NIF's Digital Video (DVID) systems was identified as the next important step. The DVIDs were facing long-term maintenance risk due to its obsolete Window XP platform, with over 500 computers to be individually upgraded and patched, 24 camera types with a variety of I/O interfaces and proprietary drivers/software with their licensing needs. In this presentation, we discuss how we leveraged the strengths of NIF's distributed, cross platform architecture and our system migration expertise to migrate the DVID platforms to diskless clients booting off a single purpose-built immutable Linux image and replacing proprietary camera drivers with open-source drivers. The in-place upgrades with well-defined fallback strategies ensured minimal impact to the continuous 24/7 shot operations. We will also present our strategy for continuous build, test, and release of the Linux OS image to keep up with future security patches and package upgrades.

INTRODUCTION

Digital Video (DVID) systems are an integral part of the NIF control system. They participate in a variety of automatic loops (e.g., automatic alignment), provide critical diagnostics to study the laser, and allow operators to observe the state of the system in real time. The cameras may be used for still captures, streaming video or both. The capture may be precisely timed using triggers, synchronized to the arrival of the laser pulse, or may be manually captured by an operator on demand. It can be used for spatial measurements – e.g., for measuring the spatial aberrations of the NIF beam's optical wavefront. With over 500 DVIDs deployed along the laser path leading to the target, the DVIDs provide machine vision functions to NIF including optics inspection, automation, application-oriented machine vision processing, and vision-guided automatic alignment/positioning systems.

LONG TERM MAINTENANCE CHALLENGE

The NIF Integrated Computer Control System (ICCS) underwent a phase of modernization which concluded in 2019 [1]. As part of that work, the low-level, hardware-facing Front-End-Processors (FEPs) were migrated from Ada to Java based software and associated Java frameworks and libraries. The DVID FEPs also benefitted from the overall NIF ICCS modernization, however, the unique nature of the DVIDs called for further upgrades to DVID hardware and software to counter obsolescence and avoid future maintenance jeopardy. The various challenges specific to DVIDs are described in the following sections.

Obsolete OS

The DVID FEPs run on the unsupported Windows XP Operating System (OS) with inherent security issues and lack of hardware support as newer computer hardware discontinue support for older operating system versions. Non-availability of machines that can install Windows XP on them means that replacing or upgrading hardware will only become a bigger challenge with time. Stop gap measures such as upgrading random access memory (RAM) in existing systems to improve application performance also does not help much due to XP's limitation on maximum supported memory (4GB).

Large Number of FEP Images

Each DVID FEP has its own Windows image installed in the local hard disk and these OS images are required to be individually upgraded and patched. Deployment of OS/driver upgrades and patches are expensive and difficult due to the large number of machines that needs to be individually patched and tested for every modification.

Multiple Camera Interface Types

The DVIDs use several interface types to talk to the different types of cameras. Figure 1 shows how the interface types are split across all the DVID FEPs.

Interface type	No. of FEPs
FireWire	487
GigE	318
Analog-FireWire	110
Analog-PCIe	2

Figure 1: Camera interface types and usage count.

Cameras that use FireWire, also commonly known as IEEE 1394, use DCAM protocol that describes the exchange of data over the FireWire bus interface. GigE based cameras use GigE Vision [2], a standard for video transfer and device control over Ethernet networks. Analog cameras typically provide an RS-170 analog signal, which then

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-MOAR02

gets digitized for further image processing by using a Video-To-FireWire converter or a PCI Express (PCIe) frame grabber computer adapter card.

Multiple Camera Types

The NIF control system employs multiple camera types from multiple manufacturers as shown in Fig. 2. The selection of a particular camera is based on the requirements for the specific application associated with the control function such as image resolution, streaming frame rates, or specialized features such as radiation or vibration tolerance. In some specific use cases, cameras have NIF specific customizations to satisfy the requirements for use in the control system. Each of these camera types comes with its associated Windows drivers/software and often have proprietary licensing schemes that are different across the different manufacturers or camera software vendor.

Interface type	Camera Manufacturers	Camera Models
FireWire	4	7
GigE	4	15
Analog-FireWire	1	1
Analog-PCIe	1	1

Figure 2: NIF DVID camera types.

Hard Disks – A Potential Point of Failure

The hard disks in the DVIDs are a potential point of failure and can cause downtimes due to data loss or corruption. With over 500 DVIDs deployed, the collective MTBF of the disks is a potential risk to the NIF's high expectations for availability and reliability, which essentially dictates a "zero tolerance for error". Disk based systems continue to pose a challenge to meet this level of fault tolerance, unless fault tolerant, redundant storage is implemented in every DVID.

Aging FEP Computer Hardware

The deployed FEP machines belong to different generations of CPU and hardware, and many of them do not have enough memory (RAM) or CPU horsepower to run a modern OS. Regardless of the OS upgrade path chosen, newer OS technologies demand more capable machines.

From the perspective of running a modern OS on it, Fig. 3 shows the percentage of FEP hardware that needs to be replaced or upgraded.

POTENTIAL STRATEGIES, OPTIONS AND CONTRAINTS

Hardware/software changes typically require facility downtime to support modifications, tests, and qualification. Continuous run of experiments at NIF means that an extended downtime for upgrade of hardware and software of the video systems is not an acceptable option. The DVID

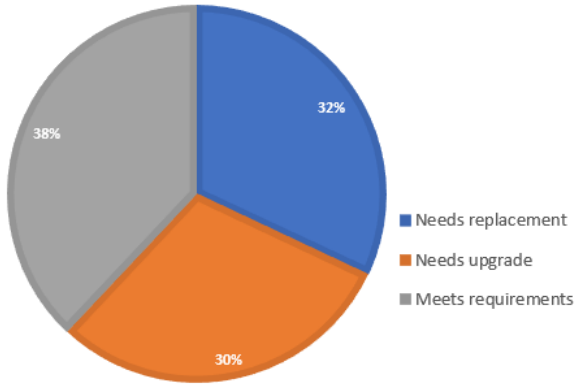

Figure 3: DVID FEP machines.

upgrades will also need to be "In-Place" in parallel with scientific operations – an approach that the facility has tolerated well as part of the prior NIF modernization phase [1].

With the success of "In-Place" upgrade strategy as described in [1], the digital video control system modernization naturally adopts the same overall strategy and best practices, with incremental deployment of new hardware/software components keeping compatibility with the existing system. This allows a smooth upgrade and progressive replacement with minimal impact to the running system.

OS Upgrade / Replacement

With the DVIDs currently running Windows XP, the obvious upgrade path may appear to be to install a newer version of Windows. However, this will not solve the challenge of maintaining a large number of separate OS images. Furthermore, opensource support for Windows based driver software for cameras is limited, and the proprietary nature of Windows will limit our ability to create a custom OS image with tightly controlled image size, tailored to run on the specific set of FEP hardware with available memory and CPU performance constraints.

A compelling option for DVID OS candidate originated from an ICCS maintained custom Linux OS distribution based on Gentoo Linux [3] that has already been successfully deployed to VME FEPs in NIF, including support for network boot mode. The already available Diskless Gentoo OS (DGOS1) offered us a platform where we would be able to incrementally add support for specific features required by DVID systems.

The 3rd option was to explore other Linux distributions such as RHEL, Debian and other Linux distributions. With Gentoo based systems already successfully deployed in NIF, this alternative option which involves significant additional effort, would be justified only if the existing Gentoo based OS could not be enhanced to meet the DVID requirements.

Even if Linux is selected as the primary DVID OS, the possibility of having specialized DVID systems that can only function with proprietary/commercial software needs to be considered as well. We expect a small percentage of

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-MOAR02

DVIDs (less than 5%) that may still need a Windows based solution to support specialized camera, drivers, or image processing software.

Camera Drivers

There were two broad categories to consider: Open-source drivers and proprietary drivers.

Proprietary Drivers Proprietary drivers include drivers provided by the camera manufacturer or supplied by a third-party software vendor. The proprietary drivers typically come with license costs and restrictions (e.g., node locked, hardware dongles, license built into camera).

Open-Source Drivers This includes drivers available with one of the many open-source licenses either distributed via open internet (e.g., through GitHub), or distributed by the manufacturer itself in source code.

The manufacturer provided drivers only work with the specific vendor's cameras, so even though the driver is standard compliant, it is generally not possible to use it across cameras from different vendors.

With the above considerations, the best approach was determined to be using open-source drivers for FireWire and GigE cameras, that would be manufacturer / model agnostic, if the cameras are standards compliant.

Camera Maintenance Tools

To enable testing of cameras in production during installation, repair or troubleshooting, operators need GUI (Graphical User Interface) tools that can query the camera parameters, capture image, and perform streaming tests. The Windows DVID with the manufacturer or third-party vendor supplied software provides a variety of GUI tools for debugging the cameras outside the ICCS framework. If we replace Windows with Linux and open-source drivers, these tools must be replaced with equivalent functionality. Since the Diskless DVID will be headless, it is also required to launch the GUI remotely.

We considered the following options for developing the camera GUI tools:

1. Use GUI sample applications that come with the opensource camera drivers. The UI must be launched remotely using X11 forwarding.
2. Use manufacturer provided Linux based applications, either by running binaries or building from source. The UI must be launched remotely using X11 forwarding.
3. Build our own custom GUI app based on web-based technologies and launch the UI remotely on the client web browser.

The UI application in option 1 and 2 above has dependencies on additional Linux libraries such as Gnome, SDL and others, and the Gentoo system should support these dependencies so that the app can be launched successfully. This could increase the OS image size significantly.

MODERNIZED ARCHITECTURE

FEP Operating System

The legacy Windows OS will be replaced with Linux OS, which will be a single image that will be configured and built to run on all DVIDs. This new Diskless Gentoo OS Version 2 (DGOS2) will be derived from the VME diskless Gentoo (DGOS1) with the following enhancements:

- Upgrade to latest Gentoo packages
- The CPU architecture type is selected from an Intel architecture version that ensures compatibility with existing DVID machines. The "core2" architecture was chosen, which provides a good balance of compatibility with older processors while not taking a performance hit on the newer processors with architecture later than core2.
- Upgrade Linux kernel to a long-term support (LTS) version: v4.14 LTS with Projected EOL Jan 2024 was chosen.
- Enable FireWire support in the kernel.
- Support additional network interfaces required for GigE DVIDs. This requires a private subnet to be created for attaching GigE cameras.
- Tune kernel network buffer size to optimize GigE performance and prevent frame drops.
- Enable NIS user authentication.
- Add FireWire hot-plug support.

The distribution builder builds all the packages from source with the selected CPU architecture. This is required, since a package binary that is built for a slightly different hardware configuration can cause the boot to fail or specific packages to malfunction.

Booting Mechanism

The DVIDs boot off a common Linux OS image over the network. The network boot is implemented using Preboot Execution Environment (PXE) [4], Dynamic Host Configuration Protocol (DHCP) and Trivial File Transfer Protocol (TFTP).

When the diskless DVID client is booted up, the PXE boot process will request the IP Address from the DHCP Server (Fig. 4). The DHCP server responds by sending the

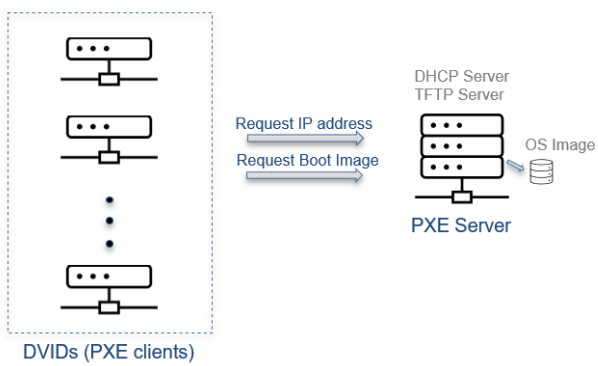

Figure 4: Diskless clients boot architecture.

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-MOAR02

IP Address and information about the boot image. In the second step of PXE boot, the boot image is requested and loaded into the DVID's Random Access Memory (RAM) by using TFTP. After all the image data loaded, the DVID automatically loads the OS and executes the OS boot sequence.

Driver Level Redesign

The proprietary drivers in the FEP will be swapped out and replaced with opensource variants. After prototyping and verifying functionality and performance, the following opensource drivers were selected:

- FireWire: **libdc1394** library with support of over 500 camera models, provides a complete API for controlling cameras conforming to DCAM specifications.
- GigE: **Aravis** library implements the Gigabit Ethernet protocols used by GigE cameras.

Although the opensource drivers support most of the required features, there are specific instances where we had to enhance it. For e.g., libdc1394 does not support access to shutter time base register which is required for supporting full range of exposure times for some cameras.

CORBA Interfaces and FEP Java Code

ICCS uses CORBA middleware technology to build a distributed control system network. Carrying over CORBA IDL interfaces intact from legacy Windows to Linux FEP software is a vital requirement for achieving smooth In-Place migration [1].

Existing Windows based FEPs were designed such that the Java support classes were selected and instantiated based on the camera vendor. Each support class is an API class whose native methods were Java Native Interface (JNI) calls to a lightweight ICCS-maintained wrapper that dispatches the calls to vendor-supplied and vendor-maintained library calls (DLL) for command/control.

Moving to Linux requires a re-architecture since most camera vendors do not provide out-of-the-box support for Linux. ICCS already maintains a vendor-agnostic approach to instantiating support classes. We designed the Linux FEP to leverage the existing methodology, abstract it to support all new Linux DVID FEPs including generic support for multiple camera bus types.

Cameras are described by text-based, human-readable YAML files, with one YAML file per camera make/model which is then stored in the file system. These camera description files are controlled alongside source code, deployed to production along with software release and gets loaded and parsed by FEP during initialization. Contents of the file determines support class instantiation and supported feature configuration for command and control of each camera defined per process in the database. FEP loads ICCS-maintained thin wrapper library (written in C, compiled for Linux) that dispatches native calls from Java FEP to the FireWire/GigE driver.

FEP Hardware Upgrade

For each DVID that is converted to Diskless Linux, the FEP computer will be either replaced with a newer machine, upgraded with higher RAM (16 GB) or used without any changes. The modification depends on the category of the machine as described in Fig. 3.

Performance Considerations

Two major areas required performance considerations: CPU performance and network performance.

CPU Performance Windows video FEPs used Intel's Integrated Performance Primitives (IPP) library to perform image compression (JPEG) and decimation. Current Windows FEP implementation utilizes IPP version 4 that dates back to 2003. To be able to run IPP on Linux, we discovered that this requires an update to bring the latest IPP support into the image processing library compiled for Linux. After the update was successfully integrated into the Linux FEP, verification of new image processing libraries was performed with the help of Automatic Alignment tests by comparing centroid locations obtained from test images decimated with current Windows 32-bit version of image processing libraries with those using new 64-bit Linux version of the library.

Image capture and streaming tests (Fig. 5) were conducted with various FEPs to measure, improve and verify that the capture response times, and streaming frame rates met or exceeded the Windows FEPs that were being replaced.

Network Performance The Windows FEP used a vendor supplied performance driver that optimized GigE Vision network traffic for adapters that use specific Intel chipsets. The advantage of the performance driver is that it offloads network traffic processing to the adapter and lowers the CPU load. With Linux FEPs, our testing did not indicate a need for the performance drivers, and the streaming performance was comparable to Windows FEPs without using the performance driver.

host	configuration			fps	measured* fps	diff from windows
	ROI	compression	bit depth			
PixeLINK 959G on Linux FEP, Dell R200)	1600 x 1200	ON	16	4	4	0.14
				5	5	0.35
				6	6	1.04
				7	6.93	1.07
				8	6.92	0.61
				9	6.833	0.433
	1600 x 1200	ON	8	10	8.8	-0.77
				11	8.75	-1.49
				12	8.58	-1.54
				13	8.6	-1.68
				14	8.55	-2.28
	1600 x 1200	OFF	16	4	3.95	0.92
				5	4.77	1.89
				6	5.22	2.32
				7	5.4	2.47
				8	7.54	4.68
				9	7.61	4.78
	1600 x 1200	OFF	8	10	9.46	3.63
				11	10.24	4.49
				12	10.78	4.99
				13	11.55	5.82
				14	13.16	7.33

Figure 5: Fragment of streaming performance test results.

Database and FEP Identity

A change in database schema not required for the Linux conversions since the strategy is to maintain the configuration and capability of the FEP. Reusing the same tables between legacy Windows and new Linux implementations also allows for easy switching between Windows/Linux software.

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-MOAR02

However, there will be minor changes to database to configure Linux specific start-up parameters. This will impact the data stored in the tables – but not the schema itself.

Taxonomical name and identity of the FEP will not change and the Linux FEPs will continue to be identified using the same identify of the Windows FEPs that they are replacing. This ensures that the conversion is transparent to the other components in the control system. This aspect is also very important for the In-Place upgrade strategy to work.

VERIFICATION AND DEPLOYMENT

Test and verification are first done in the offline (non-production) environments which consists of development, integration, and QA. One of more setups of each DVID type is available across these environments, which assure robustness of these conversions before we apply it in production. A full suite of manual and automated endurance tests is run and compared against the corresponding Windows FEP to ensure that the In-Place replacement will work as expected.

Deployment to NIF requires extensive coordination between controls hardware teams, IT, operators, subsystem leads, and testers during conversion. A typical Windows DVID FEP to Linux conversion consists of the following steps:

1. Shut down FEP's running processes
2. Configure BIOS to PXE boot instead of booting from disk.
3. Configure DHCP server to add the FEP's IP and PXE boot configuration
4. Reboot the FEP and verify successful boot-up into Diskless Linux.
5. Apply database changes to modify Linux FEP configuration parameters.
6. Startup the FEP's processes
7. Verify functionality using image capture and streaming tests.

Status of Deployments to NIF

The Diskless Linux solution has been successfully deployed to DVIDs in NIF, and functionality and performance is now proven by operation in the NIF's control system that is running a 24x7 scientific experimental schedule. With the modernized solution, our team has successfully accomplished a technology refresh of NIF's Digital Video systems and effectively solved the Digital Video systems' long term maintenance challenge.

FUTURE DIRECTIONS

Diskless Gentoo OS Roadmap

The official Gentoo Linux releases use a rolling release model, where packages are made available to the distribution as soon as they are released upstream. It has been a challenge to keep DGOS up to date with the official releases, since new versions of DGOS are currently released only as needed - typically for adding specific new features required by NIF or for applying security patches. Going forward, our plan is to release DGOS at least every 6 months to keep it current with the official Gentoo packages.

The DGOS releases are currently not integrated into the NIF ICCS continuous integration (CI) system, and verification depends primarily on manual tests. CI integration is a work in progress, and we plan to progressively integrate the DGOS build, test and release into the overall CI system which will enable us to support faster rate of release and quicker fault isolation.

Diskless Video FEP Enhancements

As part of the diskless video conversions, a large subset of DVIDs were successfully configured to handle two cameras simultaneously, significantly reducing the number of deployed FEP hardware which reduces maintenance costs and thermal stress. The upgraded DVID computers offer the possibility of further consolidation of camera functionality into the same FEP (for e.g., handle 4 cameras in the same DVID). The possibility of such efficiency improvements will be explored in the future.

Current implementation only supports FireWire and Gig cameras. Support for analog/PCIe cameras is planned in the future.

Camera Maintenance GUI Tools

We are working on deploying a maintenance GUI framework, based on web technologies, that will securely expose the camera command and control over a RESTful API, and allow testing of cameras in production during installation, repair or troubleshooting remotely from a web browser.

ACKNOWLEDGMENT

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. LLNL-CONF-827589.

REFERENCES

[1] M. Fedorov et al., "In-Place Technology Replacement of a 24x7 Operational Facility: Key Lessons Learned and Success Strategies From the NIF Control System Modernization", *presented at the 17th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'19)*, New York, NY, USA, Oct. 2019, paper WEDPL01.

[2] GigE Vision Standard, https://www.automate.org/a3-con-tent/vision-standards-gige-vision

[3] Gentoo Linux, https://www.gentoo.org/

[4] Preboot Execution Environment (PXE) Specification, http://www.pix.net/software/pxeboot/archive/pxespec.pdf