EPICS 2022 BUILD AND DEPLOYMENT WORKSHOP

APS-UPGRADE EPICS BUILD / DEPLOYMENT ENVIRONMENT “C2”

ANDREW JOHNSON
ASD/Controls Group
Argonne National Laboratory

20 September 2022

OUTLINE

- The APS Upgrade and C2 environment
- C2’s design principles
- Some details about our solutions

THE APS UPGRADE AND C2 ENVIRONMENT

THE APS UPGRADE

- A project to replace the 25+ year old electron storage ring of the US Department of Energy's Advanced Photon Source at Argonne National Laboratory, upgrading the facility to increase the brightness of the X-ray beams by up to 500 times.
- The new storage ring will use a multi-bend achromat lattice, with 15 different types of magnet (1,321 in total), to steer and focus the beam down to 4×15 microns (the current beam is 10×275 microns).
- Injection of electron bunches into the ring is changing from the current top-up approach to swap-out, which requires some upgrades to the linac and booster.
- Installation should begin 17 April 2023 and take 12 months to complete.
- The accelerator control system software is being upgraded from EPICS 3.14.12 to EPICS 7, and introduces PVAccess for new high-speed, high-volume data acquisition systems based on PVDatabase and AreaDetector.
- Channel Access will continue to be used for most slow control communications.

"C2": THE SECOND APS ACCELERATOR CONTROL SYSTEM ARCHITECTURE

- We needed a short name for the new architecture, for filesystem paths etc.
- C2 is both evolution and revolution, EPICS 7 C2 IOCs coexist with older IOCs.
- C2 replaces a collection of IOC and application management tools and scripts, mostly written and maintained in-house since the 1990s, with common public open-source software packages (plus new tools and scripts):
  - Conda for application packaging and deployment using environments
  - EPICS Sumo for IOC support module build automation
  - Git and GitLab for version control, wikis and issue tracking
- We will continue to deploy the software onto NFS shared file-systems
  - We have multiple shared installations; most systems run RHEL-8, I-Tech's Libera BPMs and LLRF run Ubuntu, separate development and production.
- We are introducing CSS-Phoebus and Python for some new applications. MEDM, ALH, StripTool, SDDS and Tcl/Tk scripts will continue to be used and maintained by our Accelerator Operations & Physics group.

PRINCIPLES OF THE C2 DESIGN

C2 DESIGN PRINCIPLES AND RULES

- All software built or installed in production must come from an on-site source or package repository
  - Local Conda package repository
    - Packages can be uploaded that were obtained from Conda-Forge
    - We create Conda recipes from pip Python packages (e.g., epics-sumo) and for some Perl modules
    - Our Conda installations only search our local repository for packages
  - Local web archive for tar files, boot-strap scripts etc.
    - Some builds install code from tar files (e.g., zeromq)
  - Local GitLab instance, used for all EPICS sources and FPGA designs
    - Externally-maintained EPICS support modules are imported from release tar-files into an upstream branch of a Git repo in GitLab
    - Use Git to merge new releases with our local changes and configuration settings on the Git main branch, then tagged for Sumo to build
    - Our sumo DEPS.DB file only contains URLs to our GitLab server

C2 DESIGN PRINCIPLES AND RULES

- Production installations of support modules and IOCs will be built by operations staff using a limited-access account.
- IOC application developers:
  - Control the version of each support module their IOCs are built from through the Sumo configure/MODULES file.
  - Are responsible for testing their IOCs when updating them to use new support module versions.
  - Cannot modify the IOC files in the production build area; all changes to production IOCs must be made by committing to the Git repository.
  - May build and boot production IOCs out of a personal build area, but only while testing and fixing problems.
- C2 IOCs write boot information to the same shared directories as older IOCs, updated crawlers monitor and extract data from there.

MORE DETAILS

CONDA

- Multi-platform packaging system, not just for Python applications
- Our “C2 tool” uploads packages to our repo, and creates and deploys Conda versioned environments intended for different kinds of users, e.g.
  - The “ioc” env provides Sumo, Git, VisualDCT, CA and PVA client tools, iocConsole and other IOC-related programs
  - The “gui” env has CSS-Phoebus, IRMIS
  - The “ioc-services” env is for servers like logstash that the IOCs connect to or that monitor IOCs status
- Versioned environments allow us to install and test updates of the tools without affecting running systems, and lets users run older versions if necessary.
  - The environment version number is the date/time the manifest file that specifies all the package versions was created/committed to Git
  - An un-versioned soft-link points to the “recommended” tagged version
- Conda packaged binaries built for linux-x86_64 work on both RHEL and Ubuntu! We use a lightly patched Perl from Conda to build Base (but not Conda’s gcc)

IOC AND SUPPORT MANAGEMENT

- Sumo works great, saves us lots of effort; Goetz Pfeiffer has been very helpful.
  - So far during development we have only created a limited number of builds, especially in our production area (7 build tags; 25 in development area)
  - **Question**: Do sites that have been using Sumo in production create new builds as and when they are required by the IOCs being updated, or do you periodically build and install tags with the latest compatible versions of all your IOC support modules?
  - **Question**: A few of our new Data Acquisition (DAQ) applications use ZeroMQ and JsonCpp in places. How do other sites integrate builds of non-EPICS packages like these into your EPICS application builds?