

The BE community: current model and future requirements

Juan David González Cobas

Librarian and Integrators Workshop

May 30th, 2017



BE Requirements

BE as a latecomer to the LCG party

BE Requirements BE Model of LCG Usage BE Model: Technicalities Future needs and evolution

A historical outline

ab initio BE's "splendid isolation" from LCG.

beginning 2017 Status review of support of C++ compilers and Python in the Technical Network (TN) environment.

Apr 2017 **BE Seminar on Python tooling**, with EP/SFT presentation.

Aug 2017 Preliminary discussions with EP/SFT to integrate LCG distributions into BE infrastructure.

Oct 2017 Agreement by BE's Python Focus Group to base TN Python distro on LCG.

Oct 2017 Same agreement for C++ compilers more recent than 4.8.

Dec 2017 Release of LCG_92python3 with BE requirements, first beta BE-CO Python distribution.

Apr 2018 Release of LCG_93python3-based Python distribution available on the TN.

BE (mostly TN-bound) constraints

BE Requirements BE Model of LCG Usage BE Model: Technicalities Future needs and evolution

The following constraints were to be taken into account

- Software environment based on SLC5/SLC6/CC7 distributions.
- Relying on *NFS-mounted volumes* and local disks.
- Consisting of servers, consoles and (diskless) front-end computers (FECs).
- No CVMFS, no EOS in the Technical Network (as of 2017).
- Extremely conservative software lifecycle and sysadmin lore (e.g.: no Python in the control system till recently, versions limited to 2.4 up to 2.6, gcc/g++ up to 4.4/4.8, etc.)
- timing-critical applications require local installation (replication into local/RAM disks, no unreliable NFS) → size constraints

Most users dealt with more recent software stacks in the GPN (e.g., jupyter, SWAN, etc.) We will not focus on this, with which everybody is happy. . .

BE non-constraints

BE Requirements BE Model of LCG Usage BE Model: Technicalities Future needs and evolution

- Release cycle *much* slower than LCG (yearly).
- Same model of deployment as the one used for Java in the control system.
- No requirement (yet?) for **big** parts of LCG like ROOT or Geant4.
- Much more restricted set of packages required (e.g. Python: around 105 packages, including dependencies and locally-developed packages).
- Identical environment required in all machines, be it servers, consoles, BE-CO virtual machines or FECs

BE Model of LCG Usage

The BE community

BE Requirements BE Model of LCG Usage BE Model: Technicalities Future needs and evolution

We serve the needs of a community of users coming from diverse areas

BE-ABP Accelerator Beam Physics. Main driving force behind the support requests for newer platforms like Python, esp. for data analysis.

BE-CO The Beam Controls group. Responsible for most services around the accelerator control system (system administration, operating system infrastructure, TN administration, controls hardware, device drivers, timing service, FESA, data management services. . .) FESA, middleware, data management services and operational applications)

BE-RF, BE-BI RF and Beam Instrumentation. Machine development, analysis of offline and live data.

TE-EPC, TE-ABT Power converters and kicker magnets. Not strictly in our Department, but users of services provided by BE-CO.

Builders of scripting tools: <https://wikis.cern.ch/display/ST>

Usage of LCG Distribution Software

BE Requirements BE Model of LCG Usage BE Model: Technicalities Future needs and evolution

From the TN, we basically pull

- a Python distribution (a release LCG_XXpython3)
- a set of packages/versions is requested through the PyFG (see e.g. the [latest list](#))
- a C++ compiler (gcc 7.1 currently).
- some ancillary libraries

and replicate this to all the TN environment.

What is this used for?

BE Requirements BE Model of LCG Usage BE Model: Technicalities Future needs and evolution

All kinds of things, in order of frequency

- processing logged accelerator data (via `pytimber` and `NXCALS` forthcoming Python interface)
- ditto live data (via `PyJAPC` and `PyLSA`)
- MD development, data analysis and exploratory programming
- (experimental) RAD GUI development `PyQt`
- As a general `MATLAB` substitute (`Jupyter` notebook, `SWAN`, `spyder`)
- device driver installation/configuration (`encore`)
- FESA code generation
- hardware testing/prototyping (`PyUAL`, `cheby`, etc.)

General references: <https://wikis.cern.ch/display/ST>,
<https://gitlab.cern.ch/cohtdrivers/cheby>

BE Model: Technicalities

Build systems

BE Requirements BE Model of LCG Usage BE Model: Technicalities Future needs and evolution

Build system? What build system?

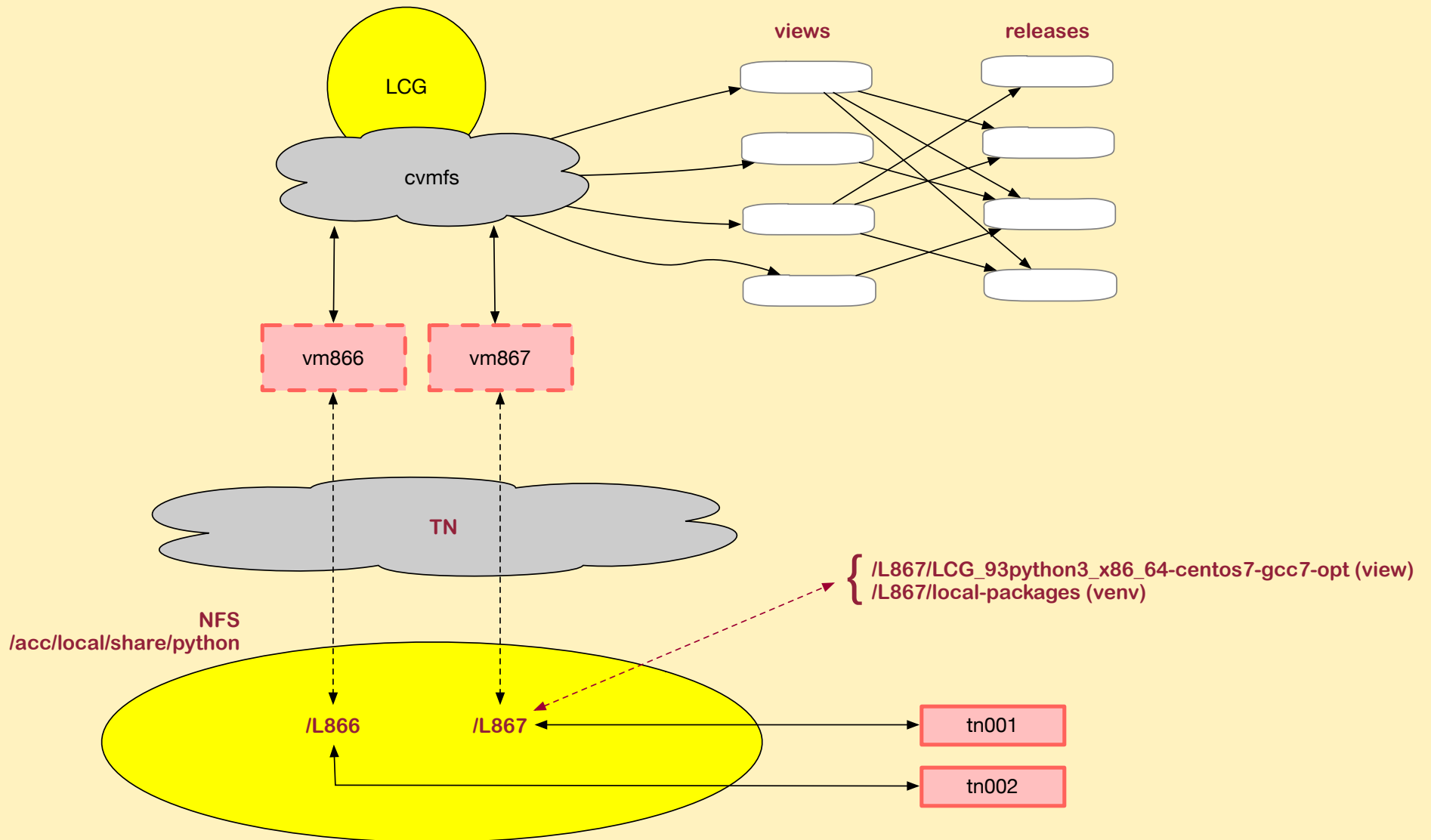
Basically, an ad hoc set of scripts that

- analyze the requested packages
- parse and compile the structure of a complete view
- find the transitive closure of dependencies
- extract the realpath's of each symlink in the view referred to by any package in the (closure of the) requested set
- replicate the symlinks and rsync the realpath's into NFS destination(s)
- (optional) replicate into local disks the NFS structure

This will be greatly simplified by containerized distributions

Sneaking stuff into the TN

BE Requirements BE Model of LCG Usage BE Model: Technicalities Future needs and evolution



Other goodies: nightlies, test and CI

BE Requirements BE Model of LCG Usage BE Model: Technicalities Future needs and evolution

We do not use them (yet). Reasons:

- Lack of time (of workpower, actually)
- We do not release often
- Validation still primitive and ad hoc
- C++ development environment (e.g. for FESA as provided by BE-CO) has its own CI infrastructure

Obviously, for validation of future release, it would prove useful to invest in this.

Future needs and evolution

Future needs and evolution

BE Requirements BE Model of LCG Usage BE Model: Technicalities Future needs and evolution

- The current model seems to be satisfactory and LCG more than enough for current needs.
- It *might* be that requirements for use of Geant4 or ROOT appear (unlikely).
- EOS is slowly getting to the TN; this might change things somehow.
- Integration with CI infrastructure of LCG *must* happen
- Integration with CI infrastructure of BE-CO *must* be studied/defined
- Lots of interest in ARM architecture support (Zynq serious candidate for the Next Platforms Contest)