SAS Virtualization

- This page is meant to organize the discussion around the virtualization of Fermi Science Analysis Systems software. Some pieces of software today have a long history and there is a clear lack of man power to have these run on recent platforms. Some software are stuck on RHEL5, other on RHEL6, others run on modern platform. A detailed status point has to be made on each piece to understand the way forward: maintenance, VM, container.
- Information on this page were first gathered from a number of reference pages:
  - Fermi Software Week Winter 2017
    - Software Week Meeting notes
    - Close out notes
    - Virtualization Discussion
- SLACK channel on containers

Table of content

Task List

- [ ] Make a detailed status for each of the main software package (Summary table below)
- [ ] Make a very simple list of use cases: L1Proc, End user, Developer...
- [ ] Summarize pros and cons of Docker vs Singularity vs Shifter (considering use cases, and what computer centers like most...)
- [ ] Summarize Virtualization and Containerization status at SLAC and CC-IN2P3
- [x] Can the Release Manager create and track the VM/Containers?
- [ ] Test containers at SLAC and CC-IN2P3, using the pipeline
- [ ] Scaling test running many containers at SLAC, and CC-IN2P3, using the pipeline
- [ ] Build a GR container, with everything and/or with a minimal runtime environment
- [ ] Run a small MC production with a GR container using the pipeline
- [ ] Run a small reprocessing task with a GR container using the pipeline

Summary

- Here is summary table with the main software packages
  - see data flow section of the Software Week Meeting notes
    - created by Johan on Tuesday 6th 2017, limiting myself to RHEL5, RHEL6 and RHEL7/CentOS7

Name	Build platform	Running platform	Special dependencies	Upgradable?	Existing VM	Existing container	Links	Comments	Date
FastCopy	RHEL5	RHEL5 or RHEL6 ?	?	✗	✗	✗	FASTCopy processing chain	to be reviewed by experts	06 Jun 2017
Halfpipe	RHEL6	RHEL6	Commercial Qt !?	unlikely ✗			Halfpipe	to be reviewed by experts	06 Jun 2017

Isoc Monitoring	RHEL5	RHEL5, works on RHEL6						to be reviewed by Steve T.	06 Jun 2017
GlastRelease	RHEL6	RHEL6	ROOT v5.34	may be with a lot of work and clean up	✅	❌		huge work, including science verification	06 Jun 2017
DataMonitoring	RHEL6	RHEL6	python2, numpy, ROOT	probably if GR is upgraded	❌	❌		svac/monitor is similar to GR, then mostly python code in dataMonitoring	06 Jun 2017
ScienceTools	RHEL7	RHEL7	RHEL7	NA	Giacomo	Sam's / Matt's		there was also one VM for the Fermi Summer school	06 Jun 2017
ASP	RHEL7	RHEL7	RHEL7	NA				should be similar to ScienceTools, ask Jim	06 Jun 2017

Virtualization vs Containers? which containers?

- Many questions to be tackled under this item:
  - building vs running
  - our software vs 3rd party libraries vs system libraries
  - LAT calibrations, connection to mysql db, file transfer via xroot
- Following June 6th 2017 meeting, we'll initially focused on Containers as it's very unlikely that the SLAC farm will move to Virtual Machines (Brian's comment)
  - For SLAC farm - docker containers for GlastRelease. Need docker registry
  - Use their system run to RHEL6 container, but batch host is RHEL7.
  - Johan: VMs might still be useful for code development and debugging, for GR in particular.
    - indeed, we'll very likely need a RHEL6 VM to build the GR containers at some point (or keep a bare metal node just for building), no?

Virtualization

- Virtual machines running in a cloud environment or on a farm made of virtual nodes
- just a simple VM for developers or end users

Containers

- a few technologies to consider Docker, Shifter (Nersk for HPC) and Singularity
- advantages over a full VM usually are:
  - light weight
  - better performance
  - container has to be carefully built
- Docker: the container
- Shifter: NERSC container optimized for HPC
- Singularity: containers for scientist, compatible with Docker
- Another very interesting link : http://geekyap.blogspot.fr/2016/11/docker-vs-singularity-vs-shifter-in-hpc.html

Tabular Comparison

Name	Docker	Singularity	Shifter
Main Goal	MicroServices, Enterprise applications	Application portability (one image with all dependencies)	Run Docker containers in HPC environment, Improve Docker security
UGE compatible	✅ but CC won't use it	✅	
LSF compatible	✅	✅	
Security	User running docker commands needs to be in special docker group to gain elevated system access LSF improves the Docker security	User runs Singularity image without special privileges.	User run shifter image without special privileges.

What systems need what kinds of containers?

- What is different for developers/users?
- Same image for all the GR uses.
- Don't want to pull a 3GB image to pull FT1, GR is 3x bigger. Just have 1 image at the moment.
- One giant image - good command line interface installed in that image.
- Images built such that the top looks the same between GR and ST. Keep same image.
- Separate builds for debugging purposes?
- GlastRelease is frozen, ST is constantly evolving. Debugging GR is not a problem, debugging ST is important
- Giacomo
    - Mount code at runtime, container doesn't have debugging tools.
    - Container provides environment.
    - Compile inside the container.
    - run debugger inside container.
    - User image has everything - compiled.
- Lightweight container for developers then they can compile. Users have full compiled.
- Debugging in GR and ST is very different
- The computing center will have a cache of docker.
- Every project will say what docker images do you want on the batch nodes?
- Plan for managing cashed images. Work out allocations for collaborations.
- Cost of using docker?

Details for main packages

Halfpipe/FastCopy

- **Halfpipe** : runs on RHEL6, but unlikely to move beyond -> virtualize at RHEL6.
    - Couple APIs need QT, using commercial version
        - Release Manager uses free version of QT, Unsure why using commercial version**,** Might be worth exploring move to free version
            - **Note JB: commercial licence was paid to get support during development, not needed to run the software.**
- **FastCopy** requires RHEL5.
- ISOC ops boxes are mostly under RHEL5. Demonstrated that the tools can be run under RHEL6.
- Backup ISOC is no longer supported.

GlastRelease

- GlastRelease needs virtualizations
    - RHEL 6 is last release that we have the personnel to support
    - A few people running GlastRelease (Developers) - nice use case for Docker. Getting GlastRelease to run on your laptop is painful.
    - GlastRelease carries around Geant4
- Note that in principle RHEL6 and RHEL7 are supposed to be binary compatible, so it might be possible to just run a GR built on RHEL6 on a RHEL7 host...
    - but there might be some issue with some dependencies from the GLAST_EXT
    - doing that would require some validation of the output of GR (whereas it may not be necessary if GR is fully containerized)
- Is there a distinction between Users and Developers for GlastRelease?
    - developer clearly need to build the code, where users just need a working run time environment
- Use Cases
    - **L1 processing, reprocessing in SLAC batch farm**
        - A few numbers on the main steps done in separate jobs in L1: digi, recon/merit, FT1
            - digitization is at the "chunk" level and is just converting LDF binary data into ROOT files : requires <50 parallel jobs per run
            - recon/merit is at the "crumb" and is the heaviest part, involves calibrations and query to mysql db, then reconstruction eats most of the CPU, while merit is straightforward: up to more that 500 parallel jobs per run
            - makeFT1 is at the "chunk" level and is (almost) just converting the merit ROOT file into a FITS file, requires <50 parallel jobs per run
            - taking all this, if L1 ends up processing 3 deliveries with 3 pieces of a run each, we might in principle get up to 3000 cores running in parallel, in practice I'd say we never go over 1500.
        - RHEL6 container on a RHEL7 host
        - do FT1, FT2 files go to xrootd? yes, as far as I know (Johan)
            - Johan: I kind of remember that some step of L1Proc, makeFT1 or makeFT2 actually need both GR and ST to be setup to run... wonder if that's still true?
        - separate containers for L1?
            - Maybe not an issue if we can preload batch nodes. We're guessing ~5 GB image.
            - distinct simplified containers for digi, recon/merit and makeFT1 may actually be a good idea
                - containers for L1 could get rid of the Geant4 and all the MC stuff
                - but it means then to cache 3 different containers but the "recon" container could be potentially not too large
                - data reprocessing would need only "recon" and "FT1" containers
        - Other solution: a simple RHEL6 base container, used just as a runtime environment, while the software is accessed as usual from AFS or NFS (or whatever network file system is available)
    - **Simulations at Lyon, SLAC, GRID**
        - maybe the same as for SLAC - check with Samuel for details

        - simulations require a full GR in the container, as all the steps MC, DIGI, RECON, MERIT, are done within the **same job**
    - **Developers & Users**
        - maybe separate versions for debug symbols and source for developers. Could be on-demand production of this version.

Science Tools

- Focus with ScienceTools is just ease of distribution
- Would it be useful to distribute the tools in VMs? Containers? Both?
    - **Joris : I found this VM : Virtual Machine version 3**
    - **Johan : A VM was also setup** for the Fermi summer school, should find it somewhere
- Are there external dependencies (like xroot-d) that would cause problems with virtualization if backend changes?
    - GR uses xrootd ST does not (Eric)
- We need automated build system for ST: Release manager vs. manual builds
- Use of virtualization is for convenience - which is most useful thing to do? (Richard)
    - Don't depend on NFS/AFS if build container right. Stable for data xrootd
    - getting files/libraries and also output data.
    - Container helps with diffuse model
        - on nodes not on NFS
        - on nodes there's low overhead.
        - Caching image on all of the nodes.
        - Fermi ST image will have the diffuse model in it.
- Caching big files (e.g. templates) in container image. Need a strategy with SCS for this for how containers are cached.

Infrastructure and general questions

Release Manager:

- Release manager doesn't talk to Oracle - but it does talk to a database. Not user friendly.
- Can our Release Manager build containers? (Richard)
    - our Release Manager based on Jenkins (here)
    - Jenkins has plenty of plugins to facilitate building and testing of Docker containers
        - One plugin
        - Tutorial on how to build container with Jenkins (itself running from a container...)
    - Johan: May be just crosscheck with Joanne B. and Max T. , but it should be straightforward

Pipeline

- Needs someone that he could show the pipeline code and train to do heavy lifting when it comes to kicking the pipeline
- Docker containers for something like the batch system may cause some problems, since
- For something like the L1 pipeline, a number of images would need to be launched simultaneously
- Would size of the software cause problems with deployment?
- We would need a system where you restrict loading images to the batch farm to prevent collisions/problems
- There is probably a precedent for this, however, Matt has no experience deploying on this scale
- File size of ~1 GB is best, a few is manageable for production.

Software dependencies

- GPL_TOOLS (staging and logging)
- REPRO common tools
- REPRO task scripts
- GlastRelease
- ScienceTools
- GLAST_EXT software (e.g., python, root)
- Ftools (KIPAC installation)
- ROOT skimmer
- FITS skimmer (possibly unnecessary?)
- evtClassDefs
- calibration and alignment files
- diffuse models
- xroot tools
- xroot /glast/Scratch space
- /scratch on local batch machines
- data catalog query (FT2 file and current version of FITS files)
- mySQL DB (calibration and alignment)
- Fermi astrotools (could probably eliminate)

Farms Status

Farm	Node OS	Network FS	VMs	Container
SLAC	RHEL6	AFS / NFS	No (never says Brian)	Docker
CC-IN2P3	RHEL6 and CentOS7	AFS to be phased out, CVMFS	A OpenStack Cloud is running but not for production	Docker full support, but looking into Singularity
GRID sites	mostly RHEL6, a few CentOS7	many have CVMFS	no	?

- Notes for the SLAC farm
  - Last purchase went into dev cluster
    - many nodes @RHEL6, upgrade to RHEL7 and doing docker with this
    - Still figuring out NFS/AFS sorted out with RHEL7. GPFS?
  - It's good to come up with a plan because of security implications if NFS underneath.
    - Use right docker (UID issues w/security)
  - SLAC has a few nodes for testing docker.
    - AFS on RHEL6 docker
    - read files if world readable.
    - NFS is hardest.
  - Timeline for RHEL7, 12mo? 2018? (Matt)
    - RHEL7 support is dodgy.
    - Configuration stuff is hard part
- Notes for CC-IN2P3
  - Now full support for Docker at Lyon (Fred)
  - **Joris : Lyon wants to use Singularity because they have security issues with UGE + Docker.**
  - **Johan : the CentOS7 queue has Singularity available, will run some tests (for CTA...) soon - July 17th 2017.**

Questions

- **Joris : Is there some security issues with LSF & Docker (https://developer.ibm.com/storage/2017/01/09/running-ibm-spectrum-lsf-jobs-in-docker-containers/)**
- **Joris : We need to verify the compatibility between Singularity (Lyon CC) and Docker**
- etc.