ScienceTools Virtual Machine Appliance

Note: An "appliance" is any prepackaged virtual machine image designed for a specific purpose. So you can have a MySQL appliance that is an OS+installed version of MySQL that you just fire up and start using the database or an e-mail appliance that is a preconfigured e-mail server, etc.

Initally this is just some notes to begin with, things will get more fleshed out as we work more on this.

Pros/Cons of running ST in a VM appliance

Pros:

- You get a out-of-the-box, working installation of the ScienceTools. You don't have to set it up yourself
- You can run any version of the OS/ST (that we make appliances for) on whatever OS you usually use

Cons:

- You have to install a VM software system on your machine
- Slight performance hit. Virtual Machines run at near native speeds these days but there is still a few percent pefromance loss over running in the native OS
- Any disk space and memory used by the VM is unavailable to the host system (disk always and memory while the VM is running)

Virtualization Software

We'll be using VirtualBox from Oracle as the virtualization software. Why:

- It's free for all the major OS's we use (Linux, Windows, Mac). VMware only has free solutions for Windows and Linux, the Mac software is $80.
- Can potentially support more CPU cores in the virtual machine
- Provides a built in mechanism to share directories between the host system and the guest system inside the VM.

You have to have VirtualBox installed on your system in order to run the VM. It can be obtained here: http://www.virtualbox.org. As far as I know (at least for Linux), you must have root access or have an administrator install the software on your system. I don't believe that it can be installed and run by a normal user (unlike VMWare Player which can be) (*This should be checked*).

One Possible Usage Scenario

This is the usage scenario I (Tom S) see as the most likely/useful/plausable. Others may have other ideas and suggestions, please add them.

The ScienceTools Appliance is designed to be a bare bones installation of the OS and a specific version of the Science Tools. Each version of the appliance only has a single version of the Science Tools installed along with the minimum OS capabilities need to run them. Thus upgrading to a new version of the Science Tools is a simple as downloading and running a new version of the appliance.

To use the Science Tools installed on the VM, a user would do the following (*need to check that all these are acually manual processes*)

1. Download and install the specific version of the appliance desired
2. Configure the allocated memory, processors, etc as desired if something different from the defaults is desired or needed.
3. Configure the shared data directory
4. Start the VM
5. Log in as the science tool user
6. Mount the shared data directory
7. Work using the installed tools from the VM plus any scripts and data in the shared data directory.
8. Log off and shut down the VM when done.

If the user wanted to work across multiple machines, the virtual machine appliance could be installed on each one and all that would have to be moved are the relevant scripts and data between the systems.

Demo virtual machine

Version 1

I've uploaded a demo virtual machine to u35. The linux path to the files is **/nfs/farm/g/glast/u35/VM/SL5_32-bit-09-24-00.tar.bz2**. It can be accessed via the web at ftp://ftp-glast.slac.stanford.edu/glast.u35/VM/SL5_32-bit-09-24-00.tar.bz2 . The file is 1.4 GB comressed and expands to about 4.8 GB when uncompressed. The virtual machine has the following characteristics:

- Built with VirtualBox (v4.1.0 r73009)
- Guest OS: Scientific Linux 5.6 32-bit version (Scientific Linux is a RedHat Enterprise Linux clone, SL5==RHEL5 with a different branding)
- Configured Memory: 768MB
- Configured disk: 8GB (1.5GB to swap, rest to OS)
- Installed ScienceTool version: 09-24-00

Notes:

- Root password on the VM is *stroot*
- the username is *STuser*
- the user passowrd is *stuser*
- In this particular incarnation, the GLAST_EXT environment variable is not set automatically, users will have to set it manually using 'export GLAST_EXT="$HOME/GLAST_EXT"' on the command line or adding it to their .bashrc file the first time they log in. This will be fixed in the next one.
- Science Tools are installed in $HOME/ST/
- The Guest Additions (a feature of VirtualBox) that allows the sharing of folders between the guest and host OS's hasn't been installed yet. As such, this machine was really just a test to see if it could be easily setup and have the Science Tools installed and to see if they actually worked. (This will be fixed in the next one).
- I didn't do any updates to the base OS after installation. It is entirely possible there are updates available.
- I didnt' try to trim any excess from the base OS installation. This was more a proof of concept than an actuall implementation. I suspect there is a bunch of things (OpenOffice among others) that could be trimmed to make the distribution size smaller.

Version 2

I've make a new version and uploaded it to u35. The linux path to the files is */nfs/farm/g/glast/u35/VM/*ST_SL5_64b_09-24-00.tar.bzip2. It can be accessed via the web at ftp://ftp-glast.slac.stanford.edu/glast.u35/VM/ST_SL5_64b_09-24-00.tar.bzip2 . The file is 1.3 GB comresssed and expands to about 4.6 GB when uncompressed. The virtual machine has the following characteristics:

- Built with VirtualBox (v4.1.0 r73009)
- Guest OS: Scientific Linux 5.6 64-bit version (Scientific Linux is a RedHat Enterprise Linux clone, SL5==RHEL5 with a different branding)
- Configured Memory: 768MB
- Configured disk: 8GB (1.5GB to swap, rest to OS)
- Installed ScienceTool version: 09-24-00

Notes:

- Root password on the VM is *stroot*
- the username is *STuser*
- the user passowrd is *stuser*
- In this particular incarnation, the GLAST_EXT environment variable has set in the .bashrc file and is in the environment.
- Science Tools are installed in $HOME/ST/
- The Guest Additions (a feature of VirtualBox) is installed
- Base OS fully updated after installation.
- I did trim out a few thing (like OpenOffice and the games) but it didn't do much to the file size.
- Installed the extFiles which the installer doesn't do by default.
- Set the $CALDB variable to point to the correct point in the directory tree.
- Didn't do anything to fix the corrupted numpy installation that Toby reported.

Try it and and provide feedback.

Configurations

What OS configurations do we want/need? Personally I (Tom S) propose that we only provide the VM for a single OS, preferably the most recent Redhat OS we are supporting (currently RHEL 5). However, I (Tom S) don't have access to actual RHEL install media so I'd have to use a clone (namely Scientific Linux) but they are effectively the same thing.

Another related issue is do we provide both 32-bit and 64-bit versions? In theory, VirtualBox can run 64-bit guest OSes on a 32-bit host OS as long as the underlying architecture is 64 bit but it takes a bit more setup on the part of the end user. Plus there is bound to be a performance hit. The question is really, do we need to supply a 32-bit appliance at all? There are fewer and fewer 32-bit machines out there although there are still a lot of 32-bit versions of the OSes running on the 64-bit architectures.

The next issue is what versions of the ScienceTools do we provide appliances for. Obviously we will want to provide them for the Release versions. Is there any reason to provide HEAD and LATEST versions? Personally, I (Tom S) don't think so but others may disagree.

If we are only supplying a single OS, only Release builds and the two architecture options (32 & 64 bit), this will be something that can be easily done by hand. Providing more OSes and especially providing HEAD and LATEST builds would require some investment in time to figure out a way to create the appliances automatically as part of the build process.

Building the ST appliance

The basic plan would be to have a virtual machine with a bare-bones OS installation. This would get updated as needed for OS patches but for the most part would be fairly static.

When a new build of ST was ready to be deployed, we make a copy of the base OS VM, and in the clone install the new ST version, update anything that needed to be changed and deploy this updated version.

This would give us a consistant starting point for all the ST installations and also save time by not having to reinstall the OS each time we did a new distribution.

Things to be done to create the release image:

1) Clone base OS image

2) Install ST version

3) Add bin directory to path

4) Add CALDB variable to point to the correct place

5) Install extFiles directory in GLAST_EXT