XSEDE Cloud VM Repository

Introduction

New XSEDE computational resources with cloud provisioning capabilities, including *Bridges* at PSC and *JetStream* at the University of Indiana and TACC, are now coming online. This follows the deployment of previous NSF-funded experimental cloud infrastructure projects, including *FutureGrid*, *Chameleon* and *CloudLab*, which have provided test environments for exploring ways in which cloud technologies and infrastructure can be architected, constructed, maintained, and operated to serve the needs of the national computational science community.

Development, construction, testing, maintenance, vetting and cataloguing of cloud virtual machine (VM) images are non-trivial tasks. Many VM images share common base operating systems and services, to which specific permutations of tools, applications, libraries, services, and configuration data are added to create customized, replicable VM *appliances*, intended to serve specific tasks and roles in a computing infrastructure. To facilitate sharing and reuse of cloud VM images and appliances generated for XSEDE service providers (SPs) and users, an XSEDE Cloud VM Repository (XCVMR) has been proposed.

Usage Model

The XCVMR library will initially contain a collection of “off-the-shelf” VMs based on common operating systems with basic installations. Customized variants of these VMs will be constructed and added to the library over time by XSEDE Service Provider (SP) staff in response to user needs, usage statistics, and observations made by SP staff.

To enable provisioning of customized VMs for users based on those made available in the XCVMR library, the XCVMR will maintain a package index per VM client operating system (OS) platform. Packages and configurations options available for each client OS will be made available for selection and installation onto an existing VM image from the XCVMR library. The resulting custom VM image will be stored on behalf of the user and made available to the user’s XSEDE project group. A copy of the VM may also be stored as a new baseline (midline?) VM image for further customization in the XVMR library.

Upon activation, initialization and use of a VM image by a user, the copy of the VM image executed will be retained for the user in their allocated XSEDE VM storage space. Modifications made by the user to the contents of their VM will thereby be retained between executions of the VM image. Users may duplicate their VMs in their allocated storage to retain snapshots if needed, within their allocated storage quota limits.

Cloud VM Construction Styles

Existing Cloud VM/appliance collections typically contain VMs constructed in one of four styles: By way of analogy, we will refer to these styles as *gourmet*, *main+sides*, *buffet*, and *plate* style VMs.

- A *gourmet* VM has been assembled as a well-designed, cohesive whole system image with the operating system, software and services built and configured to run as-is or with minimal, necessary “salt” (customization) added by the user. Gourmet VMs are largely standalone appliances and are not intended to be modified further beyond regular updates provided by the VM developer.

- A *main+sides* VM has its base operating system and purpose-oriented software and services installed and configured. To this, the user can add various software and custom configurations to suit their needs from a menu of available optional packages made available by the VM developers or cloud SP. Updates to the VM may be limited to those provided for the base installation and packages on the limited menu by the VM developer or cloud SP.

- A *buffet* VM is provided with only the operating system and necessary software and configuration to start-up the VM in the cloud infrastructure or hypervisor. To this, the user can add various software and custom configurations to suit their needs from a menu of available optional packages made available by the VM developers or cloud SP. Users may add packages from external repositories as needed; maintenance of these external packages is left to the VM user.

- A *plate* VM is one with only a stock installation of an operating system. Operating system vendors commonly make these available as cloud images for download and full customization by the VM user. Updates to the base operating system are available from the OS vendor, and everything else is left to the VM user to install and maintain.

VM developers will typically begin with a plate VM and build it up to one of the other styles, with associated package libraries, package management tools (e.g., Puppet, Salt, Chef, Ansible, etc.) and system/inventory management tools (e.g., Spacewalk) employed as needed.

Security Considerations

For every style of VM deployed, and every cloud or hypervisor infrastructure on which the VMs are executed, appropriate security policies and measures must be established and agreed to by VM infrastructure operators and users alike. These are essentially all the same as those that are commonly in place for conventional computing systems, but there are some additional requirements that are unique to the portable and readily-customizable VM arena.
VMs transported between different hosting infrastructures and administrative zones are likely to be exposed to different security risks and controls. The degree to which cloud SP operators may

be permitted access to VMs and their data may be similarly restricted, given the sensitivity of data (e.g. HIPAA-protected) or licensing restrictions for user-installed software.

A default installation of OpenStack permits authorized users to deploy VM images from the OpenStack Glance image service or imported from a user-specified URL. This poses several security concerns for OpenStack Cloud operators, as vulnerabilities and unauthorized service behaviors may be included in user-supplied or other external VM images. To support operation of user-supplied and other externally retrieved (US/ER) VMs, a separate, quarantined virtual cloud environment must be defined, with heightened security monitoring, logging, SP notification and response capabilities. It may not be possible to examine the contents of US/ER VM images in advance, nor to determine the behavior of installed software and services, the interactions of users with the VMs, and the access by unknown third parties to the VMs once they are operating. Clear policies and acceptable use agreements must be established in advanced and enforced by XSEDE and SPs before users are permitted to operate US/ER VMs on XSEDE systems.

Licensing Considerations

XSEDE SPs with site-specific license pools for commercial software will need to implement license allocation and monitoring methods to ensure compliance with license distribution limits on VMs, as needed on a per-VM or per-user basis. Users who have their own licenses for software installed on VMs may employ those licenses with installed software on VMs if permitted to do so under their license agreements, and assuming that the VM, SP cloud or VM host systems can “see” the correct license servers to validate access.

Regulation and use of software licenses must be covered in the XSEDE end-user license agreement (EULA) and any related service-level agreements (SLAs) that XSEDE users and SPs sign and agree to adhere to.

XCVMR Operations

To ensure the long term viability of the XCVMR, XSEDE staff resources must be allocated and dedicated to these vital, technically challenging activities, in addition to applicable repository infrastructure systems, services, software, storage and network resources.

Cloud VM Interoperability
(TBD)

Jason’s Thoughts
(TBD)

One Proposed Solution

Recently the OpenStack Foundation has ushered in new projects that pertain directly to VM image creation and management. Murano, an application catalog service, strives to enable developers and cloud administrators to publish cloud ready applications in a browsable, categorized context. The Murano catalog will provide the functionality to build and abstract repeatable, complex, custom environments. Deployment of these environments can be automated in OpenStack through templates via a software orchestration service such as Heat. The Heat orchestration service works with Murano in conjunction with cloud-init scripts to instantiate application creation at VM runtime.

Creating a repository of Heat templates and cloud-init scripts for an XSEDE wide service while presenting a unified application catalog would provide a robust list of choices for user VMs. The proposed solution would also minimize the steady state stack of images which require constant patching and synchronization between XSEDE SPs.

Related Work

Open Cloud Computing Interface (OCCI)

OCCI is an open, platform-independent, RESTful protocol and API for interfacing between external users and a cloud service provider's internal cloud management framework. It's goal is to permit remote OCCI client users to request and use cloud resources without having to know how to navigate each cloud service provider's implementation. OCCI specifications are published by the Open Grid Forum. Details are described at http://occi-wg.org/

Cloud Data Management Interface (CDMI)

Supported by the Storage and Networking Industry Association (SNIA), CDMI "defines the functional interface that applications will use to create, retrieve, update and delete data elements from the Cloud" (http://www.snia.org/cdmi). CDMI media types are described in IETF RFC 6208 (https://tools.ietf.org/html/rfc6208).

EGI FedCloud

Launched in May, 2014, the European Grid Initiative (EGI) Federated Cloud (FedCloud) (http://www.egi.eu/solutions/fed-cloud/) is an open, standards-based, cooperative cloud infrastructure spread over 20 infrastructure providers in Europe, supporting computational research in the European community.
The EGI FedCloud maintains a web-based repository of VM images, appliances and containers at https://appdb.egi.eu/ . Cloud VM images and appliances made available via the EGI AppDB may be searched and retrieved at https://appdb.egi.eu/browse/cloud . Although many of the

VMs listed are *plate* style VMs with only a base operating system on them (indeed several appear to be simply republished from vendor sites), others include complete working computational applications, e.g., Weather Research and Forecasting (WRF), and virtual software appliance collections provided by CERN for LHC experiment analyses.

Virtual Machine image Repository & Catalog (VMRC)

Developed by the Grid and High Performance Computing Group (GRyCAP) at the Universitat Politècnica de València (UPV), VMRC (http://www.grycap.upv.es/vmrc) is a client-server database and searchable catalog system for VM images. It offers match-making query capabilities to identify appropriate VM images and appliances for users seeking specific permutations of VM virtual hardware characteristics, operating system and installed packages. These capabilities may be incorporated into a dynamic VM scheduling broker to retrieve and initiate appropriate VMs and appliances, customize the VMs to fill in missing requirements, and execute tasks in a cloud environment.