Vanilla to VMWare translation plugin

This document describes the plugin to translate a Vanilla universe job to a VMWare job. The purpose of this document is to serve as a quickstart guide to users of this plugin.

The plugin itself is written as a hook to the JobRouter. On a note of overview, the job router selects a job for translation by the plugin. The JobRouter then invokes the plugin, providing it the classad for the selected job. The plugin uses the classad and some configuration information, to bundle all the files required by the job into a virtual machine, and returns the classad for the virtual machine to the job router. The job router puts this classad onto the queue of jobs to be executed. Eventually, the vmware job gets scheduled and run on some execute host by condor. At this point, the init script placed by the translate hook inside the virtual machine gets invoked. The init script takes care of running the job within the virtual machine and shutting down the virtual machine once the job is done. Once the virtual machine completes execution, the snapshot is shipped back to the submit machine. At this point, the JobRouter invokes the exit hook with the class ads for the original vanilla universe job and the completed vmware job. Within the hook, we extract the output files for the job from the virtual machine's disk and push them to the output folder for the original job. We also propagate the exit status of the job back to the original vanilla job.

Requirements:

This section details the requirements for running the plugin.

- Condor 7.2.0 or newer
- Python (since the hooks are written in python)
- Qemu (more specifically the qemu-img binary. This is used to fiddle with the virtual machine's disk.)
- mtools (This is used to fiddle with the virtual machine's disk.)
- A linux virtual machine which supports condor. (For now linux virtual machines are what are supported, but this will be expanded in the future.) The virtual machine should have a hard disk named “data.vmdk” mounted as “/condor-data”, and writable by all users. The user “condor” must exist on the virtual machine. And since the init script's written in python, the virtual machine should have python installed.

Configuration:

The first step in configuring the plugin is to configure the job router. Please refer to the condor manual for up to date information on configuration. An example way would be to configure the job router to pick up jobs for which a new attribute “WantJobRouter” is defined to be true. And then force the job to always be routed and not run directly by providing a requirement specification that can never be met in the submit file.

Configurations specifically for the hook are listed below

VM_CAPSULE_TEMPLATE = $(LOCAL_DIR)/vmware-capsule
This should point to a folder under which the hook should be able to find a folder named “condor_vm_capsule”. The condor_vm_capsule folder contains the template virtual machine, with one

hard disk named “data.vmdk” and mounted as “/condor-data”. Also note that the partition mounted as “/condor-data” should be formatted as “vfat”, or the mtools will not work.

VM_CAPSULE_SCRATCH_DIR_PREFIX = /scratch/vmathew/jr
This should point to the temporary storage that the plugin can use. Please ensure there's sufficient space in the scratch directory or the plugin is bound to fail.

VM_CAPSULE_TRANSLATE_LOG = $(LOG)/VmCapsuleTransLog
VM_CAPSULE_EXIT_LOG = $(LOG)/VmCapsuleExitLog
The log files for the translate and exit hooks respectively.

VM_CAPSULE_DISK_PART_OFFSET = 32256
The offset of the “vfat” partition within “data.vmdk”. To find this, within the virtual machine run “fdisk” on the hard disk. Issue the command “u” to swith the units displayed to sectors. Find the starting sector of the concerned partition, and multiply that by 512 to get the offset.

VM_CAPSULE_CONDOR_TARBALL = /scratch/vmathew/condor-tarballs/condor.tar.gz
The condor tarball to be used to set up the personal condor instance inside the virtual machine.

VM_CAPSULE_CONDOR_TARBALL_VERSION = 7.2.0
The version number of condor being used in the above tarball. The above tarball is extracted into “/” in the virtual machine by the init script. The assumption here is that this tarball extraction results in the directory to be named as “/condor-<version>”. This is the reason this parameter is used.

VM_CAPSULE_INIT_SCRIPT_PATH =
/scratch/vmathew/condor/condor-7.2.0/local.perdita/vmware-capsule/condor-init.py
The path to the init script to be placed inside the vm. Note here that the vm should be set up such that as it comes up, it will invoke this init script.

VM_CAPSULE_DATA_DISK_MOUNT_POINT = /condor-data
This is the path for the mount point for “data.vmdk” inside the vm.

VM_CAPSULE_CONDOR_CLASSAD_TEMPLATE =
/scratch/vmathew/condor/condor-7.2.0/local.perdita/vmware-capsule/classad.template
Plain text template for the classad for the virtual machine. To create this file, use “condor_submit -dump <filename> <submit-file>” on the virtual machine.