Testing the OSG Software Stack Using VM Universe

HTCondor Week 2015

Tim Cartwright
Open Science Grid and Center for High Throughput Computing
22 May 2015

Topics

- Testing OSG Software
- **Using VM Universe**
- **What We Learned**

Testing OSG Software

OSG Software Testing

- Integrated stack of middleware (RPMs) used in variety of deployment scenarios (e.g., CE, XRootD)
- Want light integration testing on "normal" installs, mimicking what admins do:

  install RPMs · config · start services · run clients/jobs · clean up
- Works best on virtual machine (VM): need root, avoid conflicts with other software, can trash host
- One run tests all installed components in a single environment; want many runs, each with unique:

  Operating system, Installed package(s), Repositories used

Using VM Universe

2 Disk Images Give Flexibility

Base OS Image (~6 GB)

(basic OS install, plus start-up config)

Input-Output Image (10 MB)

(input, payload, output)

Making a Base OS Image

- Use Oz to create images from ISO images (https://github.com/clalancette/oz)
- Use text file to describe image, install, and files

```
<template>
  <name>sl_6_x86_64_htcondor</name>
  <description>Scientific Linux 6 base OS</description>
  <os>
    <name>ScientificLinux-6</name>
    <version>6</version>
    <arch>x86_64</arch>
    <install type="url">
      <url>http://.../linux/scientific/6x/x86_64/os/</url>
    </install>
    <rootpw>...</rootpw>
  </os>
  <disk><size>6G</size></disk>
  <files> ... </files>
  <commands> ... </commands>
```

Making an Input-Output Image

- Use tool from **`libguestfs-tools`** RPM
- Makes image (RAW, qcow2, …) with filesystem and, optionally, adds files from directory

```
virt-make-fs --size=10M --format=qcow2 DIRECTORY FILENAME
```

```
/
  run-job
  input/
    ...
  output/
```

Launching the VM Payload

Wanted clean separation between boot and payload

/etc/rc.d/rc.local

```
#!/bin/sh
mount /dev/... /mnt/user
at -f /root/run-user-payload now + 1 minute
```

/root/run-user-payload

```
/bin/env - \
    PATH=/sbin:/usr/sbin:/bin:/usr/bin \
    USER=root \
    PWD=/ \
    LANG=en_US.UTF-8 \
    /mnt/user/run-job >> /mnt/user/run-job.log 2>&1
/sbin/poweroff
```

Extracting Output From a Run

Use Python library from libguestfs-tools RPM

```
import guestfs
g = guestfs.GuestFS()
g.add_drive_opts(image_filename, readonly=1)
g.launch()
filesystems = g.list_filesystems()
g.mount(filesystems[0][0], '/')
for image_path in g.find('/'):
    full_image_path = os.path.join('/', image_path)
    if not g.is_dir(full_image_path):
        image_dir = os.path.dirname(image_path)
        local_dir = os.path.join(output_dir, image_dir)
        if not os.path.exists(local_dir):
           os.makedirs(local_dir)
        g.download(full_image_path,
                   os.path.join(output_dir, image_path))
```

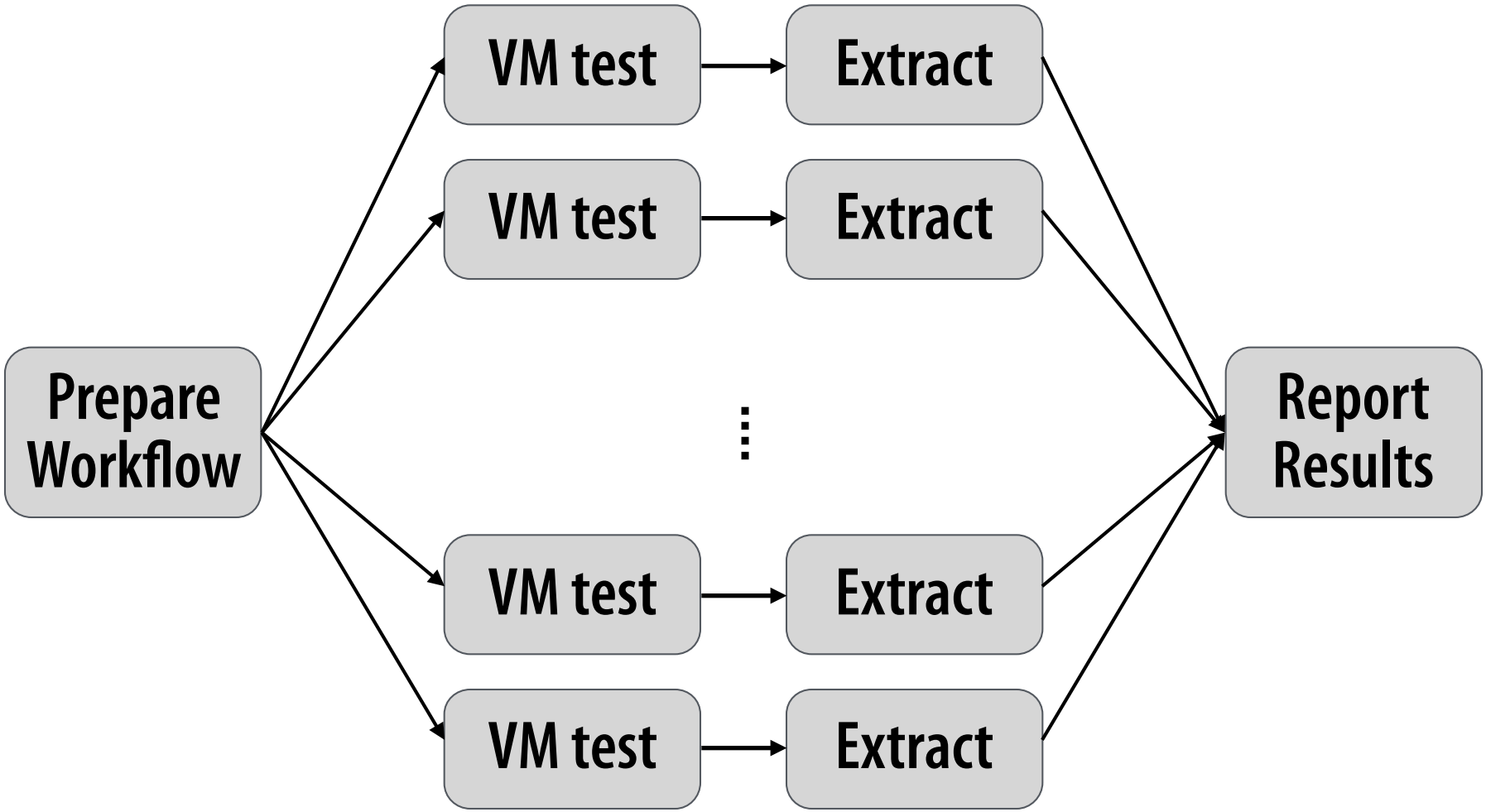
Submitting One VM Universe Job

```
universe              = vm
executable            = osg-test
vm_type               = kvm
vm_memory             = 2048
vm_networking         = true
vm_no_output_vm       = false
vm_vnc                = true
request_disk          = 6GB
log                   = osg-test.log

should_transfer_files   = YES
transfer_input_files    = .../$(platform)_htcondor.dsk, \
                          input-image-$(serial).qcow2
vm_disk                 = $(platform)_htcondor.dsk:vda:w:raw, \
                          input-image-$(serial).qcow2:vdb:w:qcow2
when_to_transfer_output = ON_EXIT
transfer_output_files   = input-image-$(serial).qcow2
transfer_output_remaps  = \
    "input-image-$(serial).qcow2 = result-image-$(serial).qcow2"

queue
```

DAG Structure



What We Learned

Benefits of Using VM Universe

- All the benefits of VMs + HTCondor management
  - Job runs as root
  - Job controls software and runtime environment
  - Job can do anything
  - VMs run on (nearly) all CHTC resources
- Current use:
  - Running 1 DAG of 420 test runs daily
  - Developers can easily initiate workflow runs, too
- Mostly just works … now

Documentation Was Inadequate

- Useful VM universe settings were not documented
- There was little explanation of the bigger picture
  - How to create images
  - How to handle input and output
  - Pros and cons of the different VM hosts
- At the time, there was no way we could have used VM universe without extensive help from CHTC developers and system administrators
- Today?

Initial Configuration Was Hard

- Settling on host and guest configs was difficult
  - Lack of documentation
  - Host and guest must agree on certain things
  - Lots of trial and error with sysadmins
- Networking, in particular, was very painful

```
# cat > /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
TYPE=Ethernet
BOOTPROTO=dhcp
ONBOOT=yes

# rm -f /etc/udev/rules.d/70-persistent-net.rules
```

Base OS Images Are Large

Each job sandbox needs ***copy*** of ***6 GB*** OS image

1. Direct transfer from submit host
   - Failures started at ≥10 simultaneous transfers
   - Resulted in transfer timeouts and job failures
2. Tried caching on execute nodes, but that's a pain
3. Tried Squid, but failed to handle huge images
4. Eventually settled on Gluster shared filesystem

```
requirements = (HasGluster == True)
```

Execution Problems I

Failure to launch

- Used to happen a lot, has gotten better (why?)
- Once KVM goes bad, all VMs on that host will fail
- Added automated email to support queue 😈
- So admins added a machine attribute for this

```
requirements = (HasVirshDefaultNetwork =?= True)
```

Execution Problems II

Launch, then hang (early in boot)

- Rare, but causes job to run forever
- No way to detect from outside (and inside is hung)
- We added a way to hold and release hung VMs

```
periodic_hold = (time() - JobCurrentStartDate > 28800)
periodic_release = (HoldReasonCode == 3) || \
                   ((HoldReasonCode == 6) && \
                    regexp("VMGAHP_ERR_INTERNAL", HoldReason))
```

Execution Problems III

- Launch cleanly but no network
  - Used to happen a lot, has gotten better (why?)
  - We catch this in the payload and bail, but…
- **NO EXIT STATUS CODES!!!!**
  - Cannot tell whether payload succeeded without unpacking output disk and looking there
  - HTCondor developers … ?

Debugging is Hard

- Throw away base OS image after run
  - Too big to transfer and retain
  - However, all run-time state is gone
- What we do
  - Log a LOT of things
  - Copy all interesting files to output image
  - Trial and error!

Conclusions

- VM universe makes possible broad testing in OSG
  - We run O(10) times more test scenarios than before, in hours instead of all day – ***big win!***
  - We can manually start test runs and they typically finish on same work day – ***big win, too!***
- Mostly worked through / around issues
- Needed and still need strong support from admins
- Future of VM universe
  - Little interest in community
  - Waning interest from developers
  - Curse you, Docker! 😀

Credits

- Brian Lin – OSG Software developer
- Todd Miller – HTCondor developer
- Jaime Frey – HTCondor developer
- Nate Yehle – CHTC lead sysadmin
- Aaron Moate – CHTC sysadmin

- Tim Cartwright – OSG Software manager (cat@cs.wisc.edu)