Status of VmBatch – A virtualization tool for grid jobs

Bjarte Kileng

Bergen University College

April 1, 2016

What is VmBatch

- System for running batch jobs inside virtual machines.
- A new virtual machine will be created for each job.
  - Each job will run inside its own virtual machine.
- The virtual machine is removed when job is finished.
- Has been tested in our private AliEn testbed to run AliEn jobs.

What is VmBatch

- System for running batch jobs inside virtual machines.
- A new virtual machine will be created for each job.
  - Each job will run inside its own virtual machine.
- The virtual machine is removed when job is finished.
- Has been tested in our private AliEn testbed to run AliEn jobs.

What is VmBatch

- System for running batch jobs inside virtual machines.
- A new virtual machine will be created for each job.
  - Each job will run inside its own virtual machine.
- The virtual machine is removed when job is finished.
- Has been tested in our private AliEn testbed to run AliEn jobs.

What is VmBatch

- System for running batch jobs inside virtual machines.
- A new virtual machine will be created for each job.
  - Each job will run inside its own virtual machine.
- The virtual machine is removed when job is finished.
- Has been tested in our private AliEn testbed to run AliEn jobs.

Why VmBatch

- Homogenize the execution environment.
- Sandbox the execution environment.
  - Improved security.
  - Less interference between jobs.
- Use CernVM for running AliEn jobs.
  - Make it easier to include new resources.
- Less complex than a full private cloud infrastructure.

Why VmBatch

- Homogenize the execution environment.
- Sandbox the execution environment.
  - Improved security.
  - Less interference between jobs.
- Use CernVM for running AliEn jobs.
  - Make it easier to include new resources.
- Less complex than a full private cloud infrastructure.

Why VmBatch

- Homogenize the execution environment.
- Sandbox the execution environment.
  - Improved security.
  - Less interference between jobs.
- Use CernVM for running AliEn jobs.
  - Make it easier to include new resources.
- Less complex than a full private cloud infrastructure.

Why VmBatch

- Homogenize the execution environment.
- Sandbox the execution environment.
  - Improved security.
  - Less interference between jobs.
- Use CernVM for running AliEn jobs.
  - Make it easier to include new resources.
- Less complex than a full private cloud infrastructure.

How it works

- When the worker node Torque service (*pbs_mom*) receives a job, VmBatch will:
  - Start a virtual machine.
  - Run the job inside the virtual machine.
  - Stop and remove the virtual machine when the job has finished.
- VmBatch can use VM images located on the worker node, but can also download images from the net.
  - VmBatch can work with a stock CentOS/SLC and CernVM image.
  - No prior adaption of images is required.
- VmBatch can also create the empty disk image used by CernVM for the CernVM-FS cache.

How it works

- When the worker node Torque service (*pbs_mom*) receives a job, VmBatch will:
  - Start a virtual machine.
  - Run the job inside the virtual machine.
  - Stop and remove the virtual machine when the job has finished.
- VmBatch can use VM images located on the worker node, but can also download images from the net.
  - VmBatch can work with a stock CentOS/SLC and CernVM image.
  - No prior adaption of images is required.
- VmBatch can also create the empty disk image used by CernVM for the CernVM-FS cache.

How it works

- When the worker node Torque service (*pbs_mom*) receives a job, VmBatch will:
  - Start a virtual machine.
  - Run the job inside the virtual machine.
  - Stop and remove the virtual machine when the job has finished.
- VmBatch can use VM images located on the worker node, but can also download images from the net.
  - VmBatch can work with a stock CentOS/SLC and CernVM image.
  - No prior adaption of images is required.
- VmBatch can also create the empty disk image used by CernVM for the CernVM-FS cache.

Setup for testing of stability

- Torque server and one worker node with XEN.
- Run 20 000 jobs through VmBatch.
  - Have had failures after 8 000 jobs. Therefore 20 000.
  - Test will create, start and remove 20 000 virtual machines, one for each job.
- 5 jobs are kept in the job queue at any time.
- 2 jobs are run concurrently.

Setup for testing of stability

- Torque server and one worker node with XEN.
- Run 20 000 jobs through VmBatch.
  - Have had failures after 8 000 jobs. Therefore 20 000.
  - Test will create, start and remove 20 000 virtual machines, one for each job.
- 5 jobs are kept in the job queue at any time.
- 2 jobs are run concurrently.

Setup for testing of stability

- Torque server and one worker node with XEN.
- Run 20 000 jobs through VmBatch.
  - Have had failures after 8 000 jobs. Therefore 20 000.
  - Test will create, start and remove 20 000 virtual machines, one for each job.
- 5 jobs are kept in the job queue at any time.
- 2 jobs are run concurrently.

Setup for testing of stability

- Torque server and one worker node with XEN.
- Run 20 000 jobs through VmBatch.
  - Have had failures after 8 000 jobs. Therefore 20 000.
  - Test will create, start and remove 20 000 virtual machines, one for each job.
- 5 jobs are kept in the job queue at any time.
- 2 jobs are run concurrently.

Observed problems

- Libvirt suddenly dies.
- Libvirt can stop responding.
- Libvirt can exhaust host memory.
  - Does Libvirt/XEN on CentOS6 have a memory leak problem?
  - If killed by kernel (OOM killer), restarting the services does not always work. Host must be restarted.
- If Libvirt crashes when starting VM, the result can be a XEN Zombie.
  - Invisible to Libvirt, but
  - take up all assigned resources
  - A normal reboot of host is not possible. Require "**reboot --force**".
- SSH into the virtual machine might require more than one attempt.
  - Rarely. Has happened twice during all tests.
- Mounting the NFS share might require several attempts.
  - Rare, but does happen.

Observed problems

- Libvirt suddenly dies.
- Libvirt can stop responding.
- Libvirt can exhaust host memory.
  - Does Libvirt/XEN on CentOS6 have a memory leak problem?
  - If killed by kernel (OOM killer), restarting the services does not always work. Host must be restarted.
- If Libvirt crashes when starting VM, the result can be a XEN Zombie.
  - Invisible to Libvirt, but
  - take up all assigned resources
  - A normal reboot of host is not possible. Require "**reboot --force**".
- SSH into the virtual machine might require more than one attempt.
  - Rarely. Has happened twice during all tests.
- Mounting the NFS share might require several attempts.
  - Rare, but does happen.

Observed problems

- Libvirt suddenly dies.
- Libvirt can stop responding.
- Libvirt can exhaust host memory.
  - Does Libvirt/XEN on CentOS6 have a memory leak problem?
  - If killed by kernel (OOM killer), restarting the services does not always work. Host must be restarted.
- If Libvirt crashes when starting VM, the result can be a XEN Zombie.
  - Invisible to Libvirt, but
  - take up all assigned resources
  - A normal reboot of host is not possible. Require "**reboot --force**".
- SSH into the virtual machine might require more than one attempt.
  - Rarely. Has happened twice during all tests.
- Mounting the NFS share might require several attempts.
  - Rare, but does happen.

Observed problems

- Libvirt suddenly dies.
- Libvirt can stop responding.
- Libvirt can exhaust host memory.
  - Does Libvirt/XEN on CentOS6 have a memory leak problem?
  - If killed by kernel (OOM killer), restarting the services does not always work. Host must be restarted.
- If Libvirt crashes when starting VM, the result can be a XEN Zombie.
  - Invisible to Libvirt, but
  - take up all assigned resources
  - A normal reboot of host is not possible. Require "**reboot --force**".
- SSH into the virtual machine might require more than one attempt.
  - Rarely. Has happened twice during all tests.
- Mounting the NFS share might require several attempts.
  - Rare, but does happen.

Observed problems

- Libvirt suddenly dies.
- Libvirt can stop responding.
- Libvirt can exhaust host memory.
  - Does Libvirt/XEN on CentOS6 have a memory leak problem?
  - If killed by kernel (OOM killer), restarting the services does not always work. Host must be restarted.
- If Libvirt crashes when starting VM, the result can be a XEN Zombie.
  - Invisible to Libvirt, but
  - take up all assigned resources
  - A normal reboot of host is not possible. Require "**reboot --force**".
- SSH into the virtual machine might require more than one attempt.
  - Rarely. Has happened twice during all tests.
- Mounting the NFS share might require several attempts.
  - Rare, but does happen.

Observed problems

- Libvirt suddenly dies.
- Libvirt can stop responding.
- Libvirt can exhaust host memory.
  - Does Libvirt/XEN on CentOS6 have a memory leak problem?
  - If killed by kernel (OOM killer), restarting the services does not always work. Host must be restarted.
- If Libvirt crashes when starting VM, the result can be a XEN Zombie.
  - Invisible to Libvirt, but
  - take up all assigned resources
  - A normal reboot of host is not possible. Require "**reboot --force**".
- SSH into the virtual machine might require more than one attempt.
  - Rarely. Has happened twice during all tests.
- Mounting the NFS share might require several attempts.
  - Rare, but does happen.

Towards a stable solution

- Handle the XEN zombies.
- Monitor the memory consumption of Libvirt, and restart if necessary.
- Abort hanging Libvirt calls and restart the hypervisor services.
- Allow several attempts when restarting the hypervisors, mounting NFS shares and connecting to VMs with SSH.

Towards a stable solution

- Handle the XEN zombies.
- Monitor the memory consumption of Libvirt, and restart if necessary.
- Abort hanging Libvirt calls and restart the hypervisor services.
- Allow several attempts when restarting the hypervisors, mounting NFS shares and connecting to VMs with SSH.

Towards a stable solution

- Handle the XEN zombies.
- Monitor the memory consumption of Libvirt, and restart if necessary.
- Abort hanging Libvirt calls and restart the hypervisor services.
- Allow several attempts when restarting the hypervisors, mounting NFS shares and connecting to VMs with SSH.

Towards a stable solution

- Handle the XEN zombies.
- Monitor the memory consumption of Libvirt, and restart if necessary.
- Abort hanging Libvirt calls and restart the hypervisor services.
- Allow several attempts when restarting the hypervisors, mounting NFS shares and connecting to VMs with SSH.

Running the stability test

- Have successfully run the test with CentOS6 as host and guest.
  - VmBatch can use three different methods for contexualizing its guests.
  - All the three methods have been tested with 20 000 jobs.
- Have also successfully run a test with CernVM.
- VmBatch has also been tested on SLC5, but not as extensively.

Running the stability test

- Have successfully run the test with CentOS6 as host and guest.
  - VmBatch can use three different methods for contexualizing its guests.
  - All the three methods have been tested with 20 000 jobs.
- Have also successfully run a test with CernVM.
- VmBatch has also been tested on SLC5, but not as extensively.

Running the stability test

- Have successfully run the test with CentOS6 as host and guest.
  - VmBatch can use three different methods for contexualizing its guests.
  - All the three methods have been tested with 20 000 jobs.
- Have also successfully run a test with CernVM.
- VmBatch has also been tested on SLC5, but not as extensively.

Testing robustness

- The stability tests test the normal behaviour of VmBatch.
- Necessary to test also the robustness:
  - How does VmBatch handle failed jobs?
  - How does VmBatch recover if aborted?
  - How does it handle problems with network connections, timeouts, missing resources?
  - What if bad configuration choices? Are the error messages useful?

Testing robustness

- The stability tests test the normal behaviour of VmBatch.
- Necessary to test also the robustness:
  - How does VmBatch handle failed jobs?
  - How does VmBatch recover if aborted?
  - How does it handle problems with network connections, timeouts, missing resources?
  - What if bad configuration choices? Are the error messages useful?

Setup for testing robustness

- Running and arbitrarily killing jobs.
  - Non-aborted jobs should run without problems.
- Running jobs and aborting VmBatch at critical points in the code.
  - Non-aborted jobs should run without problems.
- Rerunnable job and time out in the VmBatch code.
  - Timeout can happen when e.g. downloading large images.
  - The resubmitted job should run without problems.
- Rerunnable job and time out each time for the same job.
  - The next new job should run without problems.

Setup for testing robustness

- Running and arbitrarily killing jobs.
  - Non-aborted jobs should run without problems.
- Running jobs and aborting VmBatch at critical points in the code.
  - Non-aborted jobs should run without problems.
- Rerunnable job and time out in the VmBatch code.
  - Timeout can happen when e.g. downloading large images.
  - The resubmitted job should run without problems.
- Rerunnable job and time out each time for the same job.
  - The next new job should run without problems.

Setup for testing robustness

- Running and arbitrarily killing jobs.
  - Non-aborted jobs should run without problems.
- Running jobs and aborting VmBatch at critical points in the code.
  - Non-aborted jobs should run without problems.
- Rerunnable job and time out in the VmBatch code.
  - Timeout can happen when e.g. downloading large images.
  - The resubmitted job should run without problems.
- Rerunnable job and time out each time for the same job.
  - The next new job should run without problems.

Setup for testing robustness

- Running and arbitrarily killing jobs.
  - Non-aborted jobs should run without problems.
- Running jobs and aborting VmBatch at critical points in the code.
  - Non-aborted jobs should run without problems.
- Rerunnable job and time out in the VmBatch code.
  - Timeout can happen when e.g. downloading large images.
  - The resubmitted job should run without problems.
- Rerunnable job and time out each time for the same job.
  - The next new job should run without problems.

Towards a robust solution of VmBatch

- Solved a synchronization problem related to NFS cache delays.
- The default Torque scheduler (*pbs_sched*) continues to resubmit a rerunnable job if VmBatch time out each time.
  - Stopped the test before job TTL expired.

Towards a robust solution of VmBatch

- Solved a synchronization problem related to NFS cache delays.
- The default Torque scheduler (*pbs_sched*) continues to resubmit a rerunnable job if VmBatch time out each time.
  - Stopped the test before job TTL expired.

Running the robustness tests

- Have successfully run the tests using the Maui scheduler, and with CentOS6 as host and guest.
- Does the default Torque scheduler (*pbs_sched*) have a memory leak problem?
  - Had to monitor *pbs_sched* memory usage on the Torque server, and restart if to high.

More testing needed

- Not all VmBatch working modes have been tested for stability and robustness.
- Need also to test longer running jobs.
- Have not tested error behaviour with bad configuration choices.
- February – XEN 4.6 for CentOS6 is available from CentOS.
  - Not tested.
  - Includes a newer version of Libvirt.

More testing needed

- Not all VmBatch working modes have been tested for stability and robustness.
- Need also to test longer running jobs.
- Have not tested error behaviour with bad configuration choices.
- February – XEN 4.6 for CentOS6 is available from CentOS.
  - Not tested.
  - Includes a newer version of Libvirt.

More testing needed

- Not all VmBatch working modes have been tested for stability and robustness.
- Need also to test longer running jobs.
- Have not tested error behaviour with bad configuration choices.
- February – XEN 4.6 for CentOS6 is available from CentOS.
  - Not tested.
  - Includes a newer version of Libvirt.

More testing needed

- Not all VmBatch working modes have been tested for stability and robustness.
- Need also to test longer running jobs.
- Have not tested error behaviour with bad configuration choices.
- February – XEN 4.6 for CentOS6 is available from CentOS.
  - Not tested.
  - Includes a newer version of Libvirt.

VmBatch status

- VmBatch seem to work and is stable.
- If anyone would like to use or test VmBatch, I would be glad to help.
- Available through SVN – Latest VmBatch version.

VmBatch status

- VmBatch seem to work and is stable.
- If anyone would like to use or test VmBatch, I would be glad to help.
- Available through SVN – Latest VmBatch version.

VmBatch status

- VmBatch seem to work and is stable.
- If anyone would like to use or test VmBatch, I would be glad to help.
- Available through SVN – Latest VmBatch version.

What next?

- KVM on CentOS7?
- XEN on CentOS7?
- KVM on SLC6?
- Slurm in addition to Torque?
- Or, something else?

What next?

- KVM on CentOS7?
- XEN on CentOS7?
- KVM on SLC6?
- Slurm in addition to Torque?
- Or, something else?

What next?

- KVM on CentOS7?
- XEN on CentOS7?
- KVM on SLC6?
- Slurm in addition to Torque?
- Or, something else?

What next?

- KVM on CentOS7?
- XEN on CentOS7?
- KVM on SLC6?
- Slurm in addition to Torque?
- Or, something else?

What next?

- KVM on CentOS7?
- XEN on CentOS7?
- KVM on SLC6?
- Slurm in addition to Torque?
- Or, something else?

Backup

- Some numbers on the next slides.
- Average VM startup times, etc.

Some numbers – Background

- Due to NFS cache delays, the VM can be running for some time before VmBatch is notified.
  - Default setup give a maximum NFS cache delay of 60 seconds.
  - For the tests, the maximum NFS cache delay was set to 20 seconds.
- The times on the next slides are measured from VmBatch starts till it is notified that the guest is up and running.

CentOS6 guests and host

- 20 000 jobs finished in 6 days, 8 hours and 43 minutes.
- Average VM startup time: 38.41 seconds.
- Shortest VM startup time: 24 seconds.
- Longest VM startup time: 66 seconds.
- No restarts of Libvirt by VmBatch.

CernVM guests, CentOS6 host

- The 20 GByte CernVM-FS cache disk was prepared in advance.
- 20 000 jobs finished in 10 days, 9 hours and 53 minutes.
- Average VM startup time: 75.25 seconds.
- Shortest VM startup time: 51 seconds.
- Longest VM startup time: 345 seconds.
- 78 restarts of Libvirt by VmBatch due to high memory usage.