Final Report

Undergraduate Research on Virtual Machine migration by

Google Cloud's Migration Service

CSI 4995

Fall 2017 Semester

Prepared by Cole Pillars

The migration of Virtual Machines on legacy platforms is essential to bringing an application to the Cloud. Google Cloud's VM Migration Service, powered by Cloud Endure, allows VM migration and disaster recovery with relative ease. In order to visualize the time of migration and time to launch, relative to the variables of a given VM, multiple trials were ran. The variables include: VM operating system, VM disk size, data written to disk, and VM virtual CPU size.

The Migration Service migrates a machine through block-level data-replication (as opposed to file-level). Once data-replication has completed, continuous data-replication occurs on the source machine, detecting any changes made. A copy of the machine can be created on Google Cloud using Cloud Endure. Once the Google Cloud machine is ready and without issue, Cloud Endure can start the cutover process which will finalize the migration and redirect users to the new machine. Disaster Recovery is available after migration, as the machine is still in continuous data replication, and previous snapshots created can be launched if a problem occurs with the in use machine.

The Migration Service is split into two phases, the initial data replication phase and the launch machine job. During the initial data replication phase, a VM Migration Agent is installed onto the Source VM via shell to be migrated. The VM Migration Agent copies the VM at the block-level, the progress can be viewed on the Cloud Endure Console, where the launch machine job also takes place. The launch machine job brings the VM online onto the connected Google Cloud Project. The job can be configured under the machines blueprint, which can set virtual CPU size, subnet, and other properties of the new VM. Cloud Endure itself must be configured with the credentials of the Google Cloud Project which hosts the new VM. The initial data replication and the launch machine phases are tested and analyzed.

Methodology/Procedure

To begin, a Google Cloud account must be created. A trial account can be created @ https://cloud.google.com/ by clicking the "TRY IT FREE" button and completing the forms using a Gmail account. The trial account must be linked to a credit or debit card, but no charge will be made unless authorized. The account contains $300 of credits and lasts for a year before being halted. Once an account is created, the Google Cloud Console can be located @ https://console.cloud.google.com.

The first step is to create a Virtual Machine in the default project (source project); this VM will be later migrated to a destination project.

1. On the Google Cloud Console, navigate to Compute Engine > VM Instances.
2. Click "Create", this enters a configuration panel allowing the variables a VM to be chosen. The machine type (1vcpu or 4vcpu), the boot disk image (Centos7 or Ubuntu 16.04), and the boot disk size are all configured here to match the given trial. The "Create" button will launch the desired VM.
3. After the VM is launched and running, clicking the "SSH" button allows shell access to the server. Here the 'dd' command can be leveraged to fill the VM's disk space with junk files.

   *dd if=/dev/zero of=sample.txt bs=1G count=**

   Change the * to the number of gigabytes necessary to fill the hard disk to the desired amount (hard disk space can be viewed with the "*df –h*" command under /dev/sda1, upcoming steps will need to write data to the hard disk, so make sure at least 2~3 gigabytes are left accounting for the preinstalled files)

A secondary project (destination project) must be created to migrate Virtual Machines too, and then connected to Cloud Endure through use of a service account. **This step will only need to be performed a single time.**

1. To create a new project, on the Google Cloud Console, click the drop down with your current project name (default project name should be "My First Project) near the upper left. Then click the "+" button and enter a project name (such as destination-project), then

hit “Create”. Then switch to the destination project from the drop down by selecting it. **Note the ID of the project, located beside the name in the drop down, to use in step 5.**

2. Navigate to IAM & admin > Service accounts. Here we can create a service account to interface with Cloud Endure.
3. Click on “CREATE A SERVICE ACCOUNT”. Enter a name (such as Cloud-Endure), then select role as Project > Owner, and then select “Furnish a new private key” with “JSON” selected. Finally click “CREATE” to finish. **A json file is downloaded to your computer which will be used in step 5.**
4. To connect the destination project with Cloud Endure, first navigate to https://gcp.cloudendure.com/, click “SIGN IN WITH GOOGLE” and select the Gmail account used.
5. After closing the prompts, fill “Google Cloud Platform Project ID” with the ID of the destination project found in step 1. Then for “Google Cloud Platform JSON Private Key” select the file downloaded in step 3. Finally click “SAVE”.
6. Configure Replication Settings. Set the “Live Migration Target”, leave the subnet as “Default”, and then click “SAVE REPLICATION SETTINGS”.
7. Switch to the “Machines” tab and **note the Agent installation token, which is used during the next phase.**

The next step comprises the initial data replication phase of migration, migrating the machine from the source project onto the Cloud Endure Console.

1. On the Google Cloud Console, switch back to the source project (likely “My First Project”) and then navigate to Compute Engine > VM Instances.
2. Click the “SSH” button again to gain shell access to one of the previously created VMs, and then enter the following commands:

    a. *wget -O ./installer_linux.py https://gcp.cloudendure.com/installer_linux.py*

    wget may need to be installed before (in centos7: *“sudo yum -y install wget”*)

    b. sudo python ./installer_linux.py –t <TOKEN>

Change <TOKEN> the Agent Installation token, press enter when command pauses to wait for it

3. Assuming the “GCP CREDENTIALS” and “REPLICATION SETTINGS” were correctly filled, and the commands are entered correctly, the machine should show up on the Cloud Endure Console, and begin the initial data replication phase. The first test may take longer than normal as Cloud Endure has to create a VM on Google Cloud to replicate data.

4. In order to get an accurate time of initial data replication, the log data of the destination project is used. For the destination project, on the Google Cloud Console, navigate to Logging > Logs. Select GCE Disk, the beginning time is the first insert command for this instance, and the end is the last createSnapshot when the data replication is finished for this instance. The corresponding instances can be identified by viewing the logs as the test occurs and noting them.

The final phase comprises the Launch Machine phase. The VM which was replicated to the Cloud Endure Console is launched to the destination project.

1. On the Cloud Endure Console, under the “Machines” tab, select the machine that will be launched, and configure the blueprint. The Machine Type and Subnet should be changed to the corresponding values of the source projects VM.
2. Back under the “Machines” tab, select the VM to be launched, then click “LAUNCH TARGET MACHINES”, then click “Test”, and then click “CONTINUE”. A temporary converter VM is created by Cloud Endure in the destination project which carries out the launch.
3. On the Google Cloud Console in destination project’s VM Instances, the machine will be brought up, allowing it to be SSHed into, and running the applications installed.
4. In order to get an accurate time to launch the VM, view the “Job Progress” tab on Cloud Endure after a test, the start and end time are listed on top.
5. If this test is to be repeated on the same VM, make sure to delete the VM on the Google Cloud Console before running the test, as the test needs to do this, but will create inconsistent result if it isn’t handled before.

Problems/Solutions

As a variable in these tests, data on disk, the ability to create a large file was needed. The ‘dd’ command is able to create large junk files that are perfect for filling disk space.

The trial account for Google Cloud is limited in the amount of CPUs, disk storage, networks, and most other possible metrics. This meant only a small number of VMs could be present on a project at a single time, so after a VMs trials were ran, it must be deleted to free up space for the next VM.

The logging of replication time was integral to this report. The Cloud Endure Console does not log this and it is not logged in the source VM either. However, when this process occurs, disks are written too on the destination project. Comparing when they are created to when the final command is sent allows accurate timing.

Consistent testing is necessary, as often the subsequent tests change variables. For data replication, the VM must be removed from the Cloud Endure Console before a subsequent test. For the test launch, the VM must be deleted off of Google Cloud and the blueprint must be the same. Also, each test must be run individually, because the tests will share Cloud Endure VMs that execute these tasks, causing inaccurate results.

Occasionally, tests would stall at various points inconsistent with the majority of tests. Sometimes the Cloud Endure VMs would turn off randomly and had to be restarted, or a function of a test would hang near indefinitely. These results were discarded, as simply restarting the trial often fixes them.

Test Results

Launch Machine Trials (average of 3 trials)

1vcpu (1core, 3.75GB Mem)	Centos 7	Ubuntu 16.04
10GB Hard Disk 0GB Data	2:47	2:17
10GB Hard Disk 7GB Data	2:20	2:44

1vcpu (1core, 3.75GB Mem)	Centos 7	Ubuntu 16.04
50GB Hard Disk 0GB Data	2:16	2:24

50GB Hard Disk 45GB Data	2:12	2:24

4vcpu (4cores, 15GB Mem)	Centos 7	Ubuntu 16.04
10GB Hard Disk 0GB Data	2:42	2:14
10GB Hard Disk 7GB Data	2:23	2:32

Nearly all trials for differing variables were very close and in a consistent range.

Initial Data Replication Trials (average of 3 trials)

1vcpu (1core, 3.75GB Mem)	Centos 7	Ubuntu 16.04
10GB Hard Disk 0GB Data	3:00	3:20
10GB Hard Disk 7GB Data	3:22	2:57

1vcpu (1core, 3.75GB Mem)	Centos 7	Ubuntu 16.04
50GB Hard Disk 0GB Data	8:56	8:50
50GB Hard Disk 45GB Data	10:33	10:14

4vcpu (4cores, 15GB Mem)	Centos 7	Ubuntu 16.04
10GB Hard Disk 0GB Data	2:54	2:48
10GB Hard Disk 7GB Data	2:57	2:44

The increase of disk size creates a dramatic time increase. The 50GB hard disk tests also seem to imply a potential increase in time due to data on disk, whereas the 10 GB disks experience minimal difference because of this.

Conclusion

With regards to the launch machine phase of migration, the times are all very close and consistent, regardless of the variables. Since the disks on the destination project are written too during the data replication phase, none of this disk writing occurs when a VM is launched. Virtual CPU size and operating system do not appear to majorly affect time.

The data replication phase shows that affect that hard disk size and data on disk can have on the time. When hard disk size is increased to 50GB from 10GB, trial time nearly doubles. This shows that the base hard disk size alone will increase replication time. The 45GB data filled VMs also show that there is likely an increase in time when data is present, as these trials were

on average more than a minute slower. Operating system and virtual CPU size seem to have negligible effect on time.

The key component to total migration time appears to be hard disk size and data on disk. Because the launch machine phase does not write to the disk this data, the test results are near identical, but the data replication phase is primarily a function of hard disk size. A potential strategy to avoid wasted time during migration is too make sure that non-written hard disk space is minimized, as additional disks can be added to a VM on Google Cloud. Operating system and virtual CPU size seem to have a negligible effect on time.