

InfiniCloud: Leveraging the Global InfiniCortex Fabric and OpenStack Cloud for Borderless High Performance Computing of Genomic Data and Beyond

Kenneth Ban

*Institute of Molecular and Cell Biology, A*STAR, Dept. of Biochemistry, Yong Loo Lin School of Medicine, National University of Singapore
Address, Singapore*

E-mail: fkban@imcb.a-star.edu.sg

Jakub Chrzesczyk

*National Computational Infrastructure (NCI)
Australian National University, Australia*

E-mail: jakub.chrzesczyk@anu.edu.au

Andrew Howard

*National Computational Infrastructure (NCI)
Australian National University, Australia*

E-mail: andrew.howard@anu.edu.au

Dongyang Li

*National Computational Infrastructure (NCI)
Australian National University, Australia*

E-mail: dongyang.li@anu.edu.au

Tin Wee Tan

*A*STAR Computational Resource Centre (A*CRC), Dept. of Biochemistry, Yong Loo Lin School of Medicine, National University of Singapore
Address, Singapore*

E-mail: tinwee@bic.nus.edu.sg

At the Supercomputing Frontiers Conference in Singapore in 2015, A*CRC (Singapore) and NCI (Canberra, Australia) successfully presented InfiniCloud, a geographically distributed, high performance InfiniBand HPC Cloud which aims to enable borderless processing of genomic data as part of the InfiniCortex project. This paper provides a high-level technical overview of the architecture of InfiniCloud and how it can be used for high performance computation of genomic data in geographically distant sites by encapsulation of workflows/applications in Virtual Machines (VM) coupled with on-the-fly configuration of clusters and high speed transfer of data via long range Infiniband.

*International Symposium on Grids and Clouds (ISGC) 2015
15 -20 March 2015,
Academia Sinica, Taipei, Taiwan*

1. Introduction

The advent of big data has driven the need for flexible high performance computing platforms in order to analyze large amounts of data using user defined workflows, particularly in the emerging field of genomics and healthcare informatics. These workflows typically require a specific stack of applications with their Operating System-specific dependencies, which can be different for each pipeline and can frequently change over time as updates are released. In addition to the heterogenous nature of applications, such workflows demand high CPU performance paired with large memory capability as well as a high-performance interconnect for analysis of large genomic/healthcare datasets.

In response to this growing need for high performance and flexible computing for analysis of large datasets, A*CRC and NCI teams collaborated to define a new cloud computing platform called InfiniCloud, which combines high performance (HPC) cloud computing powered by OpenStack with the high speed/low latency Infiniband network architecture. This platform was optimized to provide high performance computing with minimal overhead within virtual instances, coupled with native Infiniband protocol to provide high speed interconnect and transfer of data between the instances.

We have extended the InfiniCloud platform to address another major challenge in the computation of large datasets such as genomic data: the need for efficient high speed transfers to and from the HPC cloud computing platform. To do this, we leveraged the long range Obsidian Longbow E100 Infiniband extenders to enable unprecedented high speed transfer of large datasets and VM images across trans-Pacific distances between two geographically distant InfiniCloud platforms in Singapore and Canberra via Seattle. This capability enables borderless high performance cloud computing by high speed transfer of large datasets together with workflows/applications encapsulated in VMs. Furthermore workflows/applications in VMs can be parallelized in virtual instances by the on-the-fly setup of cluster compute nodes.

We envision that the InfiniCloud platform combined with long range Infiniband as part of a global fabric (InfiniCortex) will enable seamless distributed high performance computing amongst geographically distant InfiniCloud nodes, breaking down borders on the path to exascale computing to meet the challenge of big data computing.

2. InfiniCloud Platform

The InfiniCloud platform was developed on compute cluster hardware at the NCI and A*CRC based on OpenStack cloud computing software stack with custom modifications.

2.1 Hardware Components:

Currently, InfiniCloud consists of two sites: one located at the NCI (National Computational Infrastructure), in Canberra, Australia (Figure 1) and the second at A*CRC, Singapore (Figure 2). The total count of compute cores available is 264, supporting 3TB of memory and a local storage capacity of 15TB (SSD and HDD). All instances are connected to the shared 56Gbit FDR IB fabric.



2.1.1 Server Specifications

The overall design of each site is similar, utilizing a common InfiniBand interconnect. The server configurations are detailed in Tables 1 and 2.

Table 1. NCI Compute Cluster Hardware Configuration

Servers	10x Fujitsu PRIMERGY CX250 machines
CPU	Intel Xeon E5-2650
Memory	256GB or 128GB
Interconnect	FDR IB
Local storage	4x Intel DCS3700 or 6x 2.5 inch 10k SAS

Table 2. A*CRC Compute Cluster Configuration:

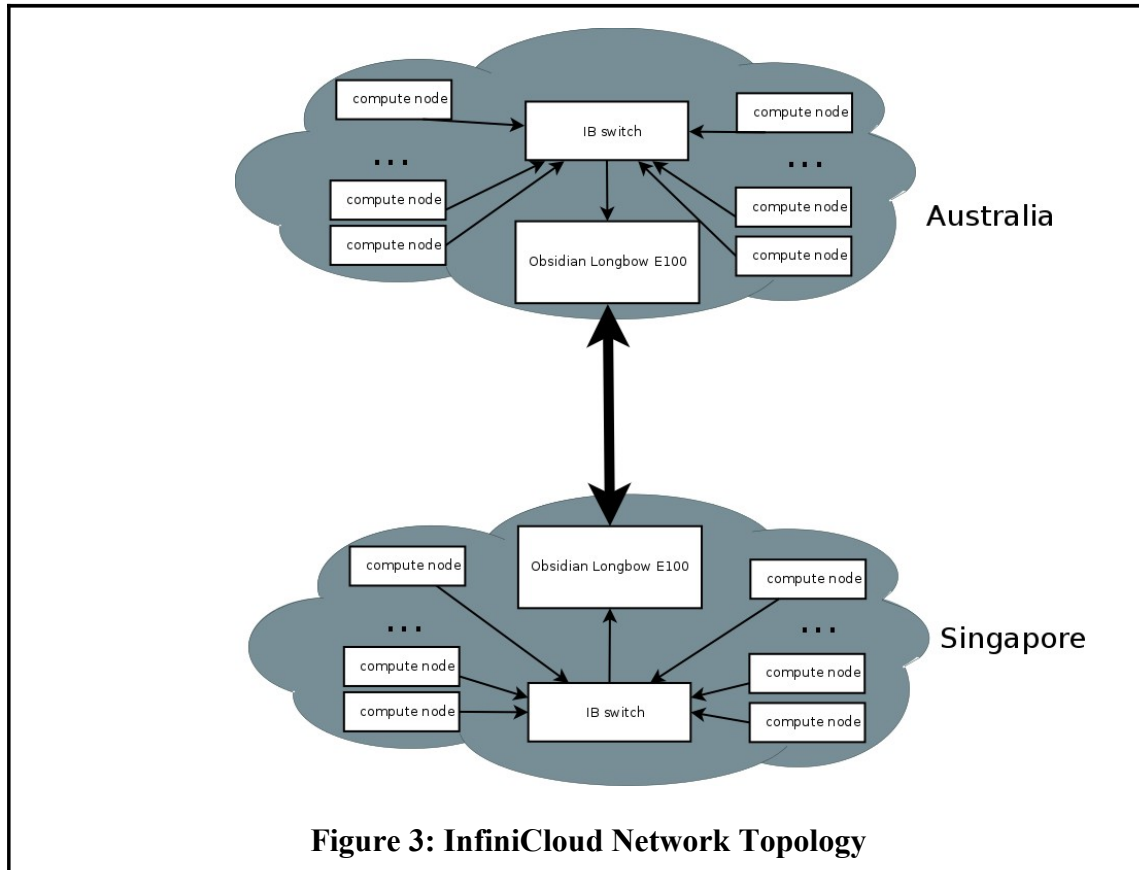
Servers	5x SGI C1104-GP1
CPU	Xeon E5-2680
Memory	128GB
Interconnect	FDR IB
Local storage	1x Intel DCS3500 or 3x Intel DCS3500

2.1.2 Local Area Network Components (each site):

POS (ISGC2015) 019

Table 3: Network Configuration

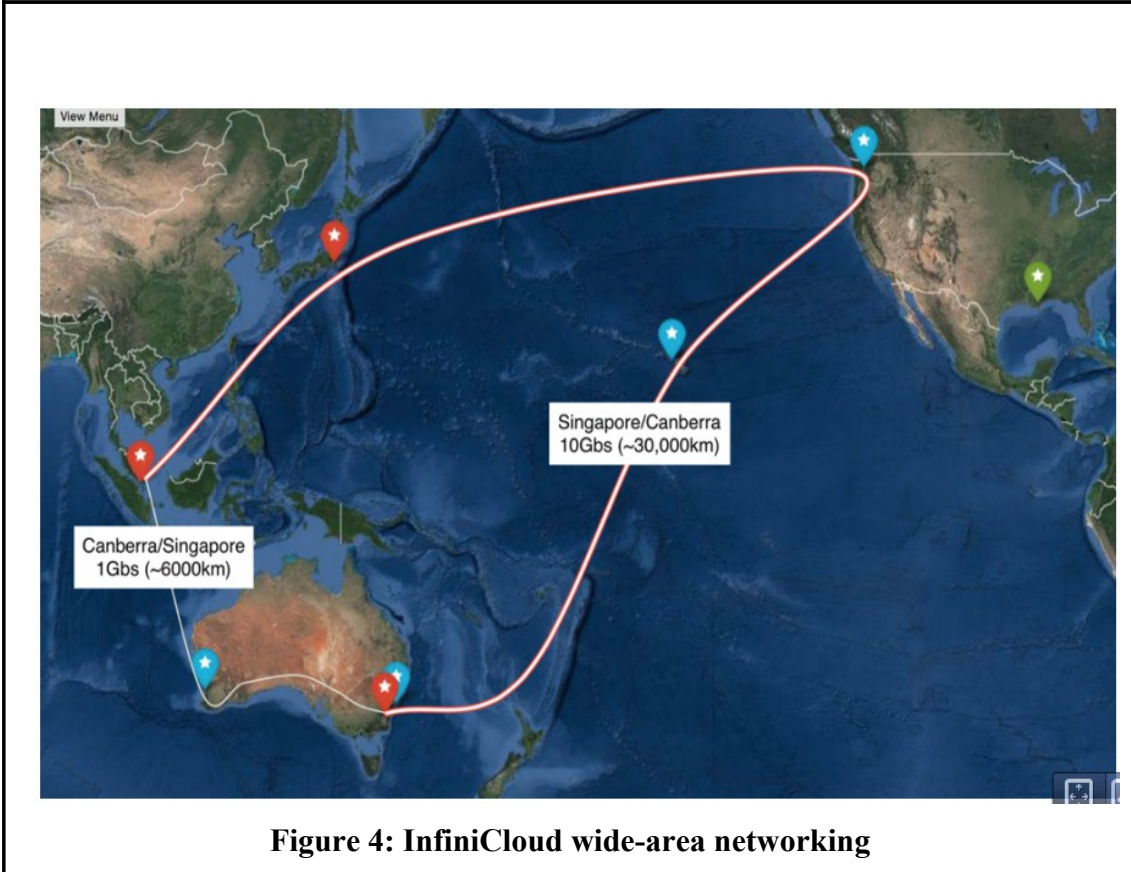
Switching	FDR IB
Range extender	Obsidian Strategics Longbow E100
Subnet manager	OpenSM (active: AU; standby: SG)



POS (ISGC2015) 019

2.1.3 Global Area Network:

To enable the global InfiniBand connection, the A*CRC and NCI teams worked closely with AARNet (AU), SingAREN (SG) and Pacific NorthWest GigaPop (PNWGP) in Seattle (USA) to secure a dedicated 10Gbit/s layer 2 link between Canberra and Singapore (Figure 4). Due to the network topology connecting Australia (with the majority of the bandwidth provided to the more densely populated East Coast of Australia), the link was routed via the longer eastern path, crossing the Pacific Ocean twice through PNWGP in Seattle with an RTT of 305ms. In contrast, the more direct western path through Western Australia, Indian Ocean and Guam while exhibiting better delay characteristics has limited capacity and is only capable of providing a 1Gbit/s connection.



2.2 InfiniCloud Installation and Configuration

All InfiniCloud systems run the following software stack (Table 4):

Table 4: InfiniCloud Software Stack

Operating System	CentOS 6.6 x86_64
InfiniBand drivers	Mellanox OFED 2.4
OpenStack version	Icehouse + InfiniCloud specific patches

Both InfiniCloud clusters consist of one dedicated management node, one dedicated controller node and a variable number of compute nodes (ranging from 4-8). All node classes are integrated to form a fully featured HPC Cloud.

- The management node is used for bare metal provisioning and cluster-wide configurations.
- The controller node provides API, CLI and GUI access to the Cloud and is responsible for managing all the core areas of cluster operation: identity management, scheduling, VM image storage, network management and providing an orchestration layer.

- Compute nodes provide CPU, RAM, storage and high performance SRIOV networking to the virtual instances.

Building the InfiniCloud cluster required a high degree of customization in order to enable native InfiniBand capability in virtual instances, as well as to provide access to the global InfiniBand network connecting Australia and Singapore. The diagram and table below (Table 5; Figure 5) show the components that were modified to enable full native Infiniband capabilities:

Table 5: OpenStack Customizations

Neutron Server	enable SRIOV and native IB capability	Controller side
Neutron Networker	enable EoIPoIB support	Networker side
Nova Compute	enable SRIOV and native IB capability	Compute side
Neutron Agent	enable SRIOV and native IB capability	Compute side
DHCP	enable IPoIB support	Networker side
eswitchd	customize Pkey assignments to create global IB fabric	Networker side

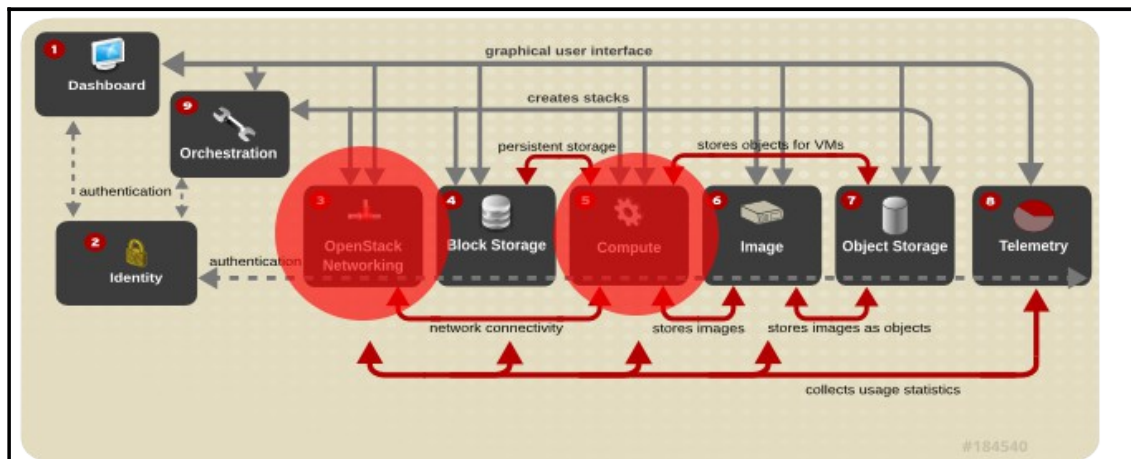


Figure 5: Overview of OpenStack components with customizations highlighted in red. Image adapted from access.redhat.com

3. InfiniCloud Infiniband Capabilities

After cloud provisioning is complete and all the customizations required for global InfiniBand communications are in place, the system has the ability to provide virtual instances on demand, connected over InfiniBand, with full ability to communicate to remote instances using RDMA over a trans-Pacific 10Gbit/s network.

POS (ISGC2015) 019

3.1 Native IB capability in VM:

Infiniband IP assignments for 2 virtual OpenStack instances (NFS-DATA-AU and NFS-REF-AU) in Infinicloud:

```
[kban@ibcm04 ~(keystone_kban)]$ nova list
+-----+-----+-----+-----+-----+
| Name                | Status | Task State | Power State | Networks                |
+-----+-----+-----+-----+-----+
| NFS-DATA-AU         | ACTIVE | -           | Running     | longbow=192.168.200.37 |
| NFS-REF-AU          | ACTIVE | -           | Running     | longbow=192.168.200.38 |
+-----+-----+-----+-----+-----+
```

IP configuration in one of the virtual instances (NFS-REF-AU):

```
[kban@ibcm04 ~(keystone_kban)]$ ssh 192.168.200.38
[root@nfs-ref-au ~]# ifconfig
Ifconfig uses the ioctl access method to get the full address information, which limits hardware addresses to 8 bytes.
Because Infiniband address has 20 bytes, only the first 8 bytes are displayed correctly.
Ifconfig is obsolete! For replacement check ip.
ib0      Link encap:InfiniBand  HWaddr A0:00:78:10:FE:80:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
      inet addr:192.168.200.38 Bcast:192.168.200.255  Mask:255.255.255.0
      inet6 addr: fe80::fa16:3e00:a3:6dfe/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:2044  Metric:1
      RX packets:65071 errors:0 dropped:0 overruns:0 frame:0
      TX packets:13377 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1024
      RX bytes:33891502 (32.3 MiB)  TX bytes:2237704 (2.1 MiB)
```

3.2 Native IB capability in VM:

We demonstrate high bandwidth capability (~6 GB/sec) between 2 virtual instances:

```
-----
                RDMA_Write BW Test
Dual-port       : OFF           Device       : mlx4_0
Number of qps  : 1             Transport type : IB
Connection type : RC           Using SRQ    : OFF
TX depth       : 128
CQ Moderation  : 100
Mtu            : 2048[B]
Link type      : IB
Max inline data : 0[B]
rdma_cm QPs    : OFF
Data ex. method : Ethernet
-----
local address: LID 0x05 QPN 0x0a5e PSN 0x90c425 RKey 0xb8011700 VAddr 0x007f476c430000
remote address: LID 0x1a QPN 0x0cac PSN 0x94503d RKey 0x7001182b VAddr 0x007fa941670000
-----
#bytes  #iterations  BW peak[MB/sec]  BW average[MB/sec]  MsgRate[Mpps]
65536   5000          5984.52          5976.36              0.095622
```

POS (ISGC2015) 019

3.3 High (10Gbit) bandwidth capability – global:

Integral to the data transfer component is the use of the Obsidian Strategies DSYNC utility which utilises the RDMA (Remote Direct Memory Access) capabilities to provide long range Infiniband RDMA transfers between Infiniband-connected virtual instances.



Here we demonstrate the high speed transfer of 381 GB of genomic data in under 9 minutes from the A*CRC InfiniCloud (Singapore) to NCI (Canberra, Australia) via the 10G link going through Seattle (~30,000 km) using the DSYNC utility.

```
[root@test01 ~]# dsync --direct-io --option Xfer::RDMA::Buffer-Size=5368709120 --option Xfer::RDMA::IO-Block-Size=10485760 192.168.200.144:/scratch/kuba/reference_dset/ /scratch/kuba-test/
Finished generating remote file list. 40 files, 3 directories, 381GB.
Finished checking local files. Need to get 40 files, 381GB.
Transfer xfer-ib-rdma network usage 3050B in 0s (10.0kB/s)
Transfer xfer-ib-rdma network usage 381GB in 8m19s (764MB/s)
Done. Transferred 381GB in 8m27s (752MB/s)
```

The remarkable performance observed with long range Infiniband RDMA provides a significant improvement over standard TCP/IP protocols.

4. Using InfiniCloud for Parallelized Workflows in Genomic Analysis

The InfiniCloud platform provides a high performance cloud computing environment for flexible workflows, coupled with unprecedented high speed transfer of big data sets over large geographical distances. A key application that takes advantage of these high performance characteristics is the analysis of genomic sequences which has seen an exponential growth in demand with the advent of next generation sequencing technologies.

	
<p>Sanger sequencing 10 years for 1 genome (3 Gbp) ~ USD \$2.7 billion</p>	<p>HiSeq X 32 genomes/week < USD \$1,000/genome</p>

The rapid development of next generation of sequencing technologies has dramatically reduced the cost of sequencing genomes. Previously, it took ~USD \$2.7 billion and 10 years to sequence one human genome, but currently the cost has dropped several orders of magnitude to ~USD \$1,000 per genome with the introduction of platforms such as the Illumina HiSeq X sequencer. This drop in cost coupled with the ability to sequence a complete human genome in a few days has driven the adoption of genomic sequencing in research labs as well as hospitals.

Although the cost and speed of sequencing has dramatically improved, the transfer and computation of the genomic data remains a bottleneck in translating that data into the insights needed for improving patient care. In addition, the computational analysis of genomic data for clinical use requires enforcement of reproducibility standards in addition to the data provenance and security guarantees needed to comply with ethical and legal privacy issues. A computational platform for clinical genomics needs to meet the following challenges:

- High speed data transfers from sequencing data stores to the computational platform
- Reproducible and well documented workflows that can be run on different hardware platforms
- Easy provisioning of compute clusters for processing genomic data from multiple samples using parallel workflows
- High CPU and network performance for rapid analysis of large datasets
- Mechanisms for data provenance and security (e.g. using ephemeral containers) for computation at remote sites

4.1 Provisioning of instances and on-the-fly setup of cluster compute nodes for parallel workflows

POS (ISGC2015) 019

To address these challenges, we implemented a software stack on top of the InfiniCloud platform that leverages the use of VM instances or containers to encapsulate workflows, together with automated provisioning of virtual instances and the setup of virtual compute clusters for parallelized workflows.

We adapted Elasticcluster (<http://gc3-uzh-ch.github.io/elasticcluster/>) for use on InfiniCloud to enable easy provisioning of instances and setup of virtual clusters for parallel workflows (Figure 6). In our custom configuration, Elasticcluster was used to provision instances and set up a virtual cluster consisting of a frontend node and a user-defined number of compute nodes. To enable cluster computing for parallel workflows, we configured Elasticcluster to install and setup the SGE job scheduler, Ganglia monitoring tools, and the IPython notebook shell (Table 6).

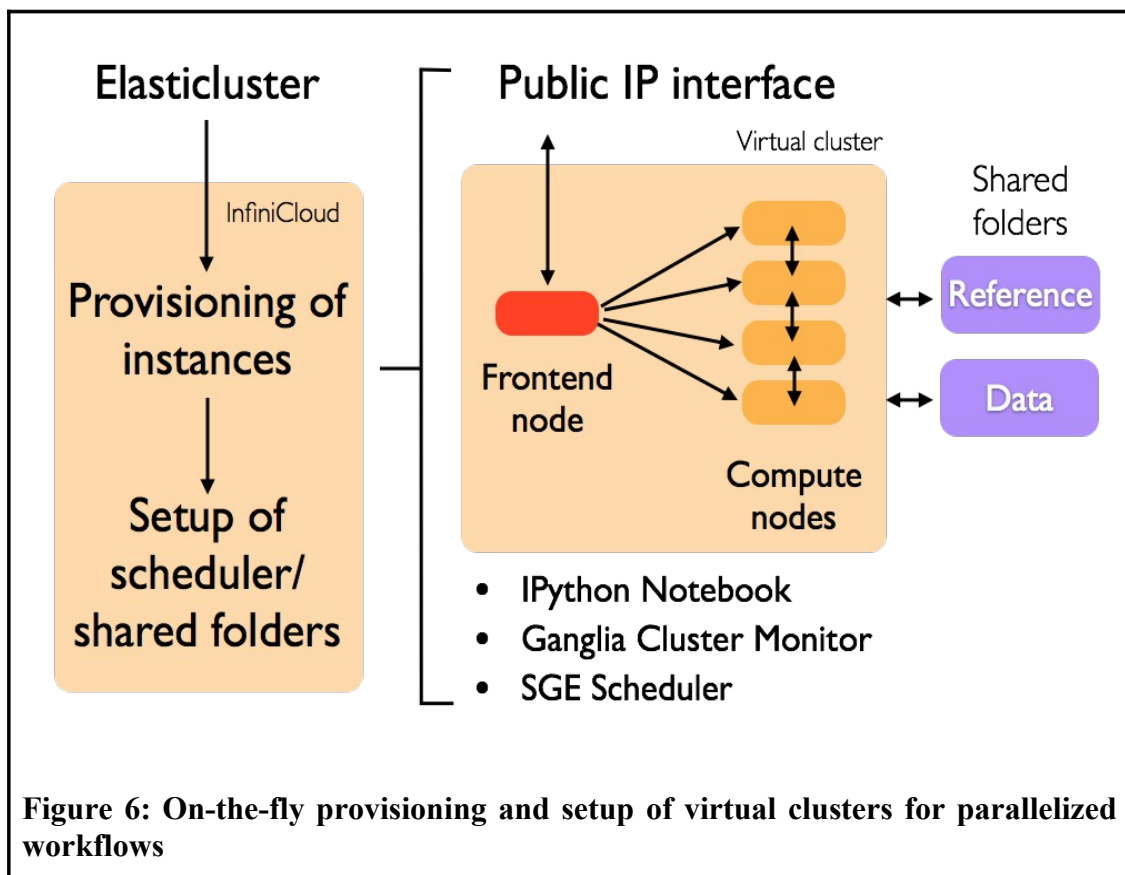


Figure 6: On-the-fly provisioning and setup of virtual clusters for parallelized workflows

Table 6: Cluster computing stack

IPython	Notebook shell (BASH/Python/R) IPython parallel engine	2.4.1
SGE	Grid engine job scheduler	6.2u5
Ganglia	Cluster monitoring (CPU/Memory/Network)	3.1.7

We demonstrate the ease of provisioning and setup of a virtual compute cluster on InfiniCloud using Elasticcluster:

```
[kban@ibcm04 ~(keystone_kban)]$ elasticcluster start biopipeline -v
Starting cluster `biopipeline` with 4 compute nodes.
Starting cluster `biopipeline` with 1 frontend nodes.
(this may take a while...)

INFO:gc3.elasticcluster:Starting node compute001.
INFO:gc3.elasticcluster:Starting node compute002.
INFO:gc3.elasticcluster:Starting node compute003.
INFO:gc3.elasticcluster:Starting node compute004.
INFO:gc3.elasticcluster:Starting node frontend001.
INFO:gc3.elasticcluster:_start_node: node has been started
INFO:gc3.elasticcluster:_start_node: node has been started
INFO:gc3.elasticcluster:_start_node: node has been started
INFO:gc3.elasticcluster:_start_node: node has been started
INFO:gc3.elasticcluster: start node: node has been started
INFO:gc3.elasticcluster:Ignoring unknown ssh-rsa host key for 192.168.200.42:
0b8ffdd5897184aa819a23a63968ac73
INFO:gc3.elasticcluster:Connection to node compute001 (192.168.200.42) successful.
INFO:gc3.elasticcluster:Ignoring unknown ssh-rsa host key for 192.168.200.41:
0b8ffdd5897184aa819a23a63968ac73
INFO:gc3.elasticcluster:Connection to node compute002 (192.168.200.41) successful.
INFO:gc3.elasticcluster:Ignoring unknown ssh-rsa host key for 192.168.200.36:
0b8ffdd5897184aa819a23a63968ac73
INFO:gc3.elasticcluster:Connection to node compute003 (192.168.200.36) successful.
INFO:gc3.elasticcluster:Ignoring unknown ssh-rsa host key for 192.168.200.43:
0b8ffdd5897184aa819a23a63968ac73
INFO:gc3.elasticcluster:Connection to node compute004 (192.168.200.43) successful.
INFO:gc3.elasticcluster:Ignoring unknown ssh-rsa host key for 192.168.200.44:
0b8ffdd5897184aa819a23a63968ac73
INFO:gc3.elasticcluster:Connection to node frontend001 (192.168.200.44) successful.
```

Following provisioning of virtual instances, the setup of virtual cluster is done automatically using Ansible Playbooks as a backend (abbreviated output):

```
Configuring the cluster.
(this too may take a while...)

Setup of SSH for all nodes
...

TASK: [Setup SSH host based authentication (server configuration file; enable
`HostbasedAuthentication` option)] ***
changed: [frontend001]
changed: [compute002]
changed: [compute004]
changed: [compute001]
changed: [compute003]
...
```

POS (ISGC2015) 019

```

Setup of Ganglia Monitor

TASK: [Install ganglia monitor (CentOS)] *****
changed: [compute002] => (item=ganglia-gmond)
changed: [compute004] => (item=ganglia-gmond)
changed: [compute001] => (item=ganglia-gmond)
changed: [compute003] => (item=ganglia-gmond)
...

Setup of SGE Job Scheduler

TASK: [Install GridEngine RPM packages] *****
changed: [frontend001] => (item=gridengine-qmaster,gridengine-execd,gridengine-qmon)
...

Setup IPython Notebook Shell Web Interface

PLAY [IPython controller Playbook (standard user operations)] *****

TASK: [create ipython profile] *****
changed: [compute001]
changed: [compute002]
changed: [compute004]
changed: [compute003]
changed: [frontend001]
...
    
```

At completion of setup, the user is able to access the frontend node via SSH:

```

INFO:gc3.elasticcluster:Cluster correctly configured.
Your cluster is ready!

Cluster name:      biopipeline
Frontend node: frontend001
- compute nodes: 4
- frontend nodes: 1

To login on the frontend node, run the command:

    elasticcluster ssh biopipeline

To upload or download files to the cluster, use the command:

    elasticcluster sftp biopipeline
    
```

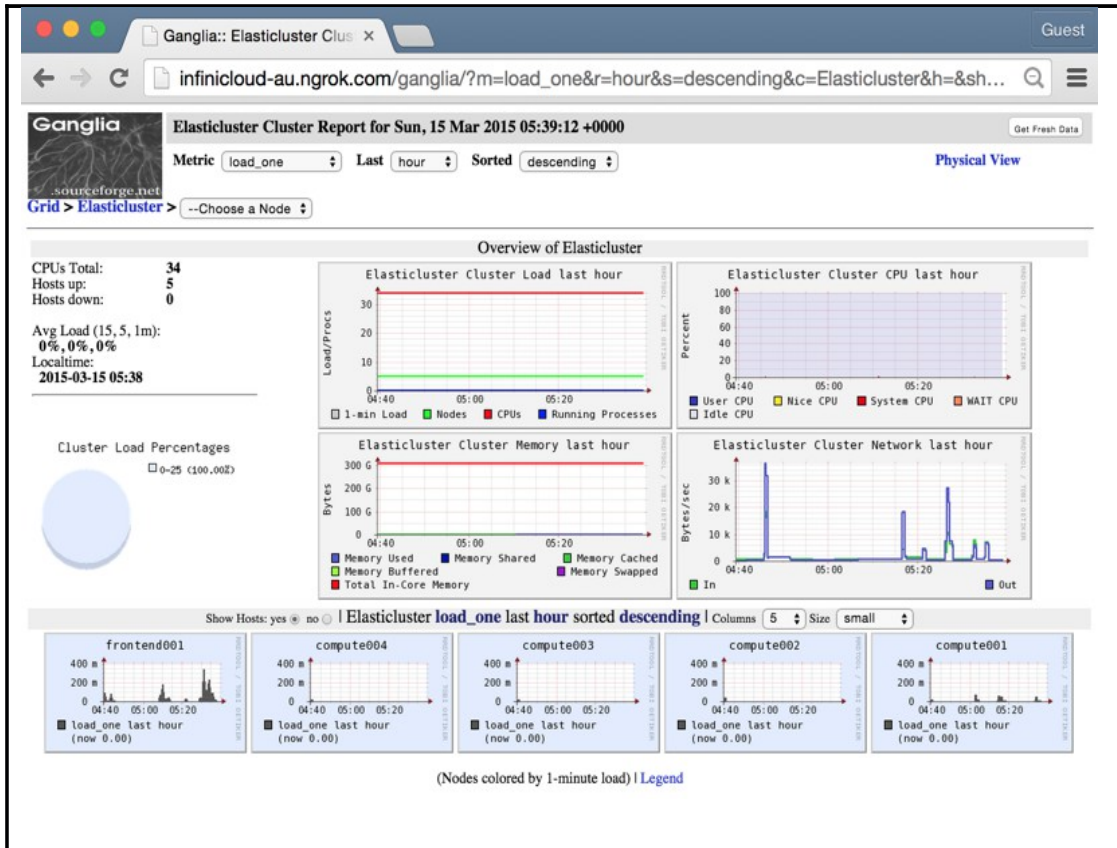
Control of cluster and job submission through frontend node using the SGE scheduler:

```

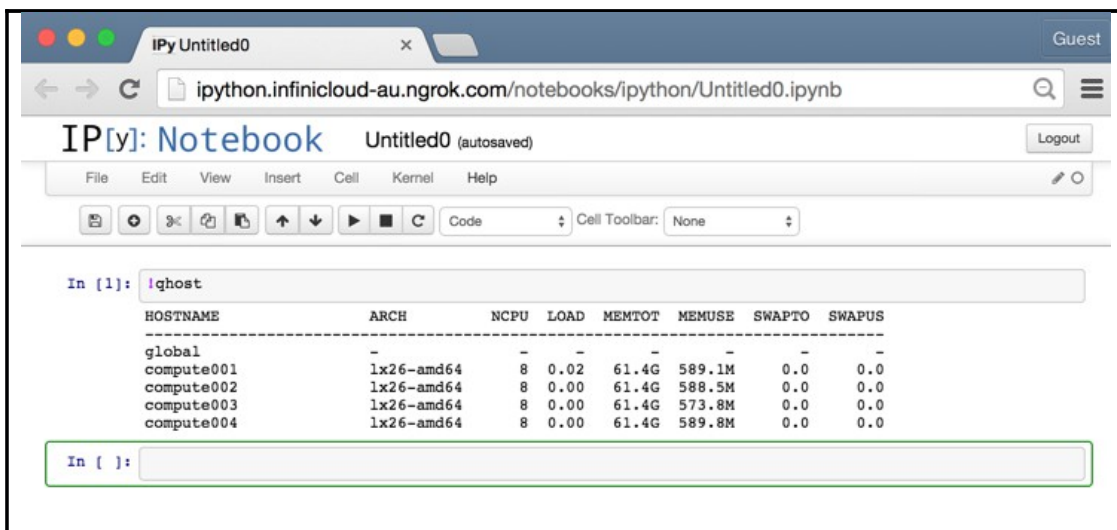
[kban@ibcm04 ~(keystone_kban)]$ elasticcluster ssh biopipeline

[root@frontend001 ~]# qhost
HOSTNAME          ARCH          NCPU  LOAD  MEMTOT  MEMUSE  SWAPTO  SWAPUS
-----
global            -             -     -     -       -       -       -
compute001        lx26-amd64    8     0.01  61.4G   573.3M  0.0     0.0
compute002        lx26-amd64    8     0.02  61.4G   573.8M  0.0     0.0
compute003        lx26-amd64    8     0.02  61.4G   577.7M  0.0     0.0
compute004        lx26-amd64    8     0.00  61.4G   575.7M  0.0     0.0
    
```

Besides SSH access, the setup provides a web interface for cluster monitoring using Ganglia:



Finally, the setup provides a versatile IPython Web Notebook interface for BASH/Python/R scripting:



POS (ISGC2015) 019

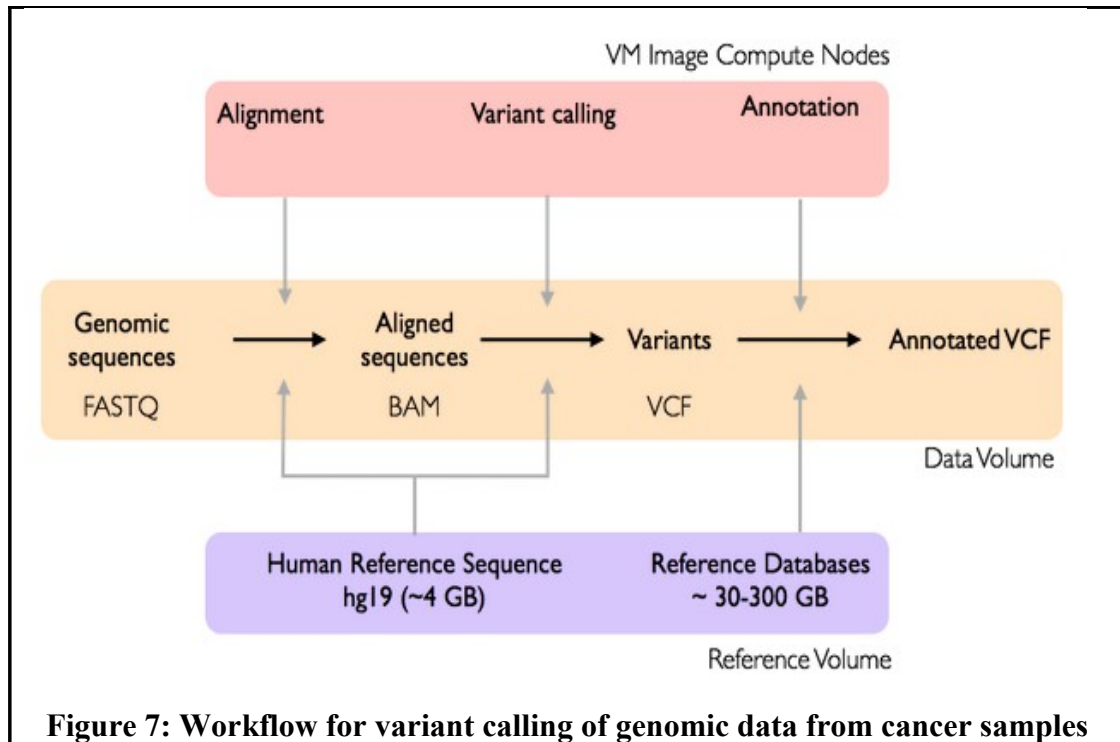
4.2 Implementation of variant calling genome analysis pipeline

Next, we demonstrate how the on-the-fly provisioning and setup of a virtual machine which can be used to parallelize a genomic analysis workflow. We chose a clinically relevant workflow, called variant calling, that takes genomic sequences from cancer samples and detects mutations in genes that could be used to determine the prognosis of a patient, or to identify potential chemotherapy drugs that could be used for treatment. Because each cancer sample can be analyzed separately, the workflow is amenable to simple asynchronous parallelization without any interprocess communication.

In this workflow, genomic sequences are processed in a pipeline through a series of steps using different applications to identify and annotate mutations (Figure 7). We use a pipeline application to orchestrate the steps in processing and to distribution the processing to the compute nodes using the SGE scheduler.

1. Genomic sequences from each cancer sample are processed with an aligner - an application that compares the sequences to a human reference genome sequence and identifies the position and alignment of each sequence from the cancer samples.
2. The files from each cancer sample are processed by a variant caller program, which compares the aligned sequences to the human reference genome sequence to identify variations (substitutions, insertions, deletions) in the cancer samples.
3. The variant files from each cancer sample are annotated. A specialized application compares each variation to multiple databases to identify what the potential effects of each mutation have on regions in the genome.

The applications are pre-installed in the VM images together with their dependencies to enable portability between InfiniCloud platforms. The reference datasets required by the aligner, variant caller, and annotation tool, are located in a data volume that can be mirrored between InfiniCloud platforms. The genomic dataset is isolated in a separate volume which also stores the results of the analysis (Figure 7); this isolation provides the flexibility for maintaining data provenance and security



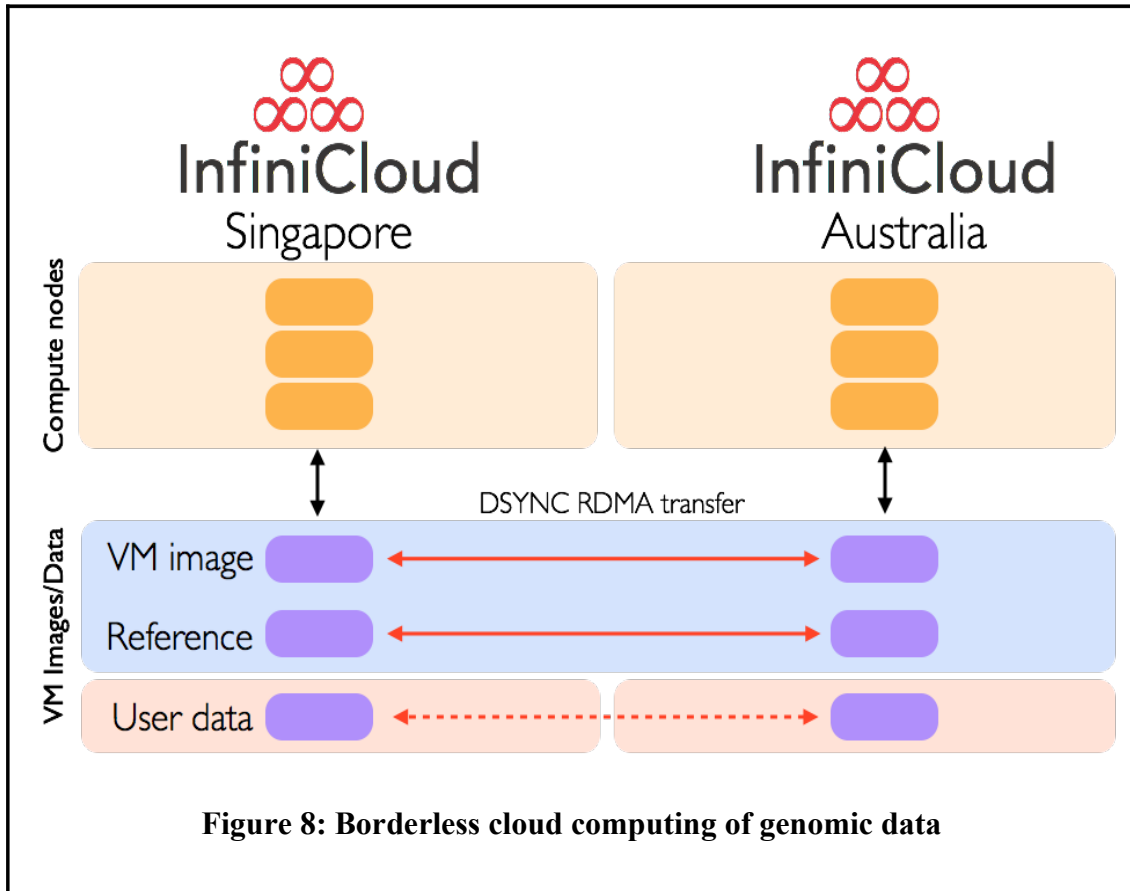
4.3 Demonstration of genome analysis workflow for remote cloud computing

We demonstrate the computation of genomic sequences on the InfiniCloud platform in Canberra, Australia from Singapore by remote provisioning of instances, setup of the cluster, and mounting of reference/data volumes (Figure 8):

- The VM images are mirrored from Singapore to Australia so that both sites have the same application/workflow backends for genomic analysis
- The common reference volume is automatically mirrored from Singapore to Australia and attached to the frontend and compute nodes
- The data volume is synchronized according to a user-defined workflow and attached to the frontend and compute nodes
 - Genomic data is transferred from Singapore to Australia for computation
 - Results are transferred back from Australia to Singapore
 - Data volume on remote site can be deleted in cases where genomic data cannot be stored offsite for data provenance and security reasons

POS (ISGC2015) 019

o



For the analysis, the genomic data is transferred from Singapore into the data volume (Australia) using dsync. Here, we achieve a transfer of ~233 GB of data in 5.5 minutes (~696 MB/sec) from Singapore to Australia via Seattle (~30,000 km):

```
[root@nfs-data-au data]# ls -l
total 0

[root@nfs-data-au data]# dsync --direct-io --option Xfer::RDMA::Buffer-Size=5368709120 --option
Xfer::RDMA::IO-Block-Size=10485760 192.168.200.156:/data/ .
Finished generating remote file list. 3564 files, 220 directories, 233GB.
Finished checking local files. Need to get 2314 files, 233GB.
Transfer xfer-ib-rdma network usage 123kB in 0s (404kB/s)
Transfer xfer-ib-rdma network usage 233GB in 5m24s (717MB/s)
Done. Transferred 233GB in 5m34s (696MB/s)

[root@nfs-data-au data]# ls -l
total 8
drwxr-xr-x 3 root root 4096 Mar 16 08:16 ipython
drwxr-xr-x 8 root root 4096 Mar 16 08:16 projects

[root@nfs-data-au data]# df
Filesystem      1K-blocks      Used Available Use% Mounted on
/dev/vda1       10185764      924476  8737300  10% /
tmpfs           9991928         0  9991928   0% /dev/shm
/dev/vdb        361243236    227723700 115169456  67% /data
```

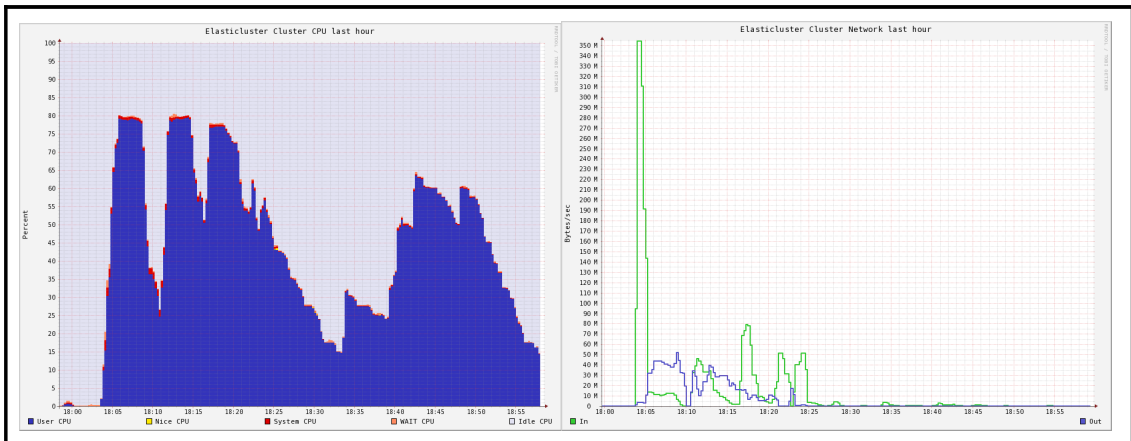
POS (ISGC2015) 019

Here we run the pipeline remotely on InfiniCloud (Australia) for variant calling on genomic sequences from cancer samples using bpipe, which interprets the workflow script specified in the variant_calling.pipe file. The abbreviated output is shown below:

```

=====
|                               Starting Pipeline at 2015-03-15 21:04                               |
=====
----- Stage align [AD365_S3_L001] -----
...
----- Stage variant [AD363_S2_L001] -----
...
----- Stage annotate [AD363_S2_L001] -----
...
===== Pipeline Finished =====
21:48:07 MSG: Finished at Sun Mar 15 21:48:07 UTC 2015
21:48:07 MSG: Outputs are:
              annotate/AD407_S5_L001.avinput
              annotate/AD407_S5_L001.hg19_multianno.csv
              annotate/AD363_S2_L001.hg19_multianno.csv
              annotate/AD422_S6_L001.avinput
              annotate/AD422_S6_L001.hg19_multianno.csv
              ... 8 more ...
    
```

Ganglia monitoring of the workflow showing aggregate CPU and network loads on all compute nodes:



Example of mutations detected in a tumour suppressor gene (TP53) in a cancer sample which generally signifies a bad prognosis:

Chr	Start	End	Ref	Alt	Func.refGene	Gene.refGene	ExonicFunc.refGene	snp138
chr17	7574025	7574025	C	-	exonic	TP53	frameshift deletion	.
chr17	7577531	7577531	G	-	exonic	TP53	frameshift deletion	.
chr17	7579472	7579472	G	C	exonic	TP53	nonsynonymous SNV	rs1042522
chr17	7579644	7579659	CCCCAGCCCTCCAGGT	-	intronic	TP53	.	rs146534833
chr17	7579801	7579801	G	C	UTR5	TP53	.	rs1642785

POS (I S G C 2 0 1 5) 0 1 9

Upon completion of the analysis, the results in the data volume are transferred back to the point of origin (Singapore) and the volume is deleted. This workflow provides one approach to maintain data provenance at the site of origin while harnessing the high performance computational resources at remote sites.

5. Conclusions

We present a new cloud computing platform called InfiniCloud, which combines High Performance Cloud Computing powered by OpenStack with the high speed/low latency of an Infiniband network architecture. This platform delivers High Performance Computing with minimal overhead within virtual instances, coupled with native Infiniband protocol for high speed interconnect transfer of data between the instances.

The InfiniCloud platform incorporates long range Infiniband extension and enables unprecedented high speed transfers of large datasets and VM images across global distances. The A*Star and NCI teams have successfully demonstrated data transfer across trans-Pacific distances (~30,000 km) between two geographically distant InfiniCloud platforms located in Singapore and Canberra, Australia. This capability enables borderless high performance cloud computing supporting high speed transfer of large datasets together with workflows/applications encapsulated in Virtual Machines. This encapsulation allows parallelization of virtual instances and on-the-fly instantiation of cluster compute nodes using Elasticcluster.

We envision that the InfiniCloud platform combined with long range Infiniband as part of the InfiniCortex global InfiniBand fabric, will enable seamless distributed cloud based High Performance Computing amongst geographically distant InfiniCloud nodes, breaking down borders and illuminating the path to Exascale computing to meet the challenge of supporting today's and tomorrow's big data computing needs.