Viola Meixian Vuong

**Building Cloud IaaS infrastructures and Computing Models**

*Introduction to Big Data Processing Infrastructures – Report*

**1. Aim**

The aim of this work is to implement a computing model able to align 300M sequences against the hg19 database, using the Burrows-Wheeler Aligner (BWA) software. Both implementation and evaluation in terms of time and cost are discussed. In order to address the computational challenge discussed here the following considerations were considered:

* An alignment of 1000 sequences creates an ascii file of 2.7GB.
* Inside the patient direction there are 1444 read files of 170KB each (more or less), i.e. a total of 245MB.
* Each patient file has 1000 sequences.

**2. Building the cloud infrastructure**

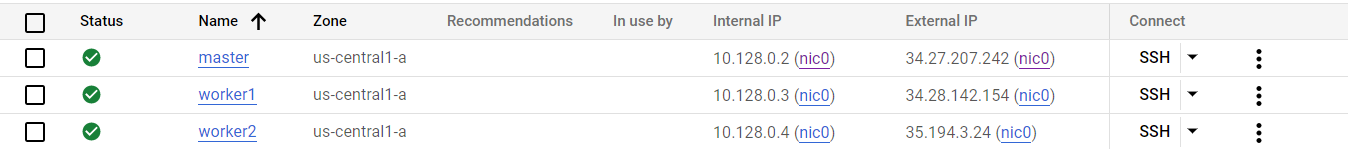
The aim of the project will be achieved building a cloud computing infrastructure. The infrastructure consists in an HTC site, a HPC site, and storage site. For each site, a different Google Cloud project will be created in order to simulate geographically distributed sites.

Immagine che contiene testo

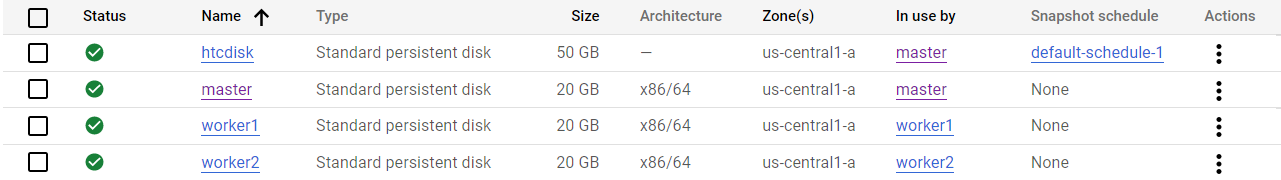
Descrizione generata automaticamente

**2.1. High Throughput Computing (HTC) site**

The HTC site provides CPU (batch farm) and storage (one shared volume). Therefore, three instances were created: one master node (main), followed by two slaves (worker1, worker2). All these instances have 2 vCPUs and 8GB of memory (**e2-standard-2**).



An additional volume (disk of 50GB) was created and attached to the master node.

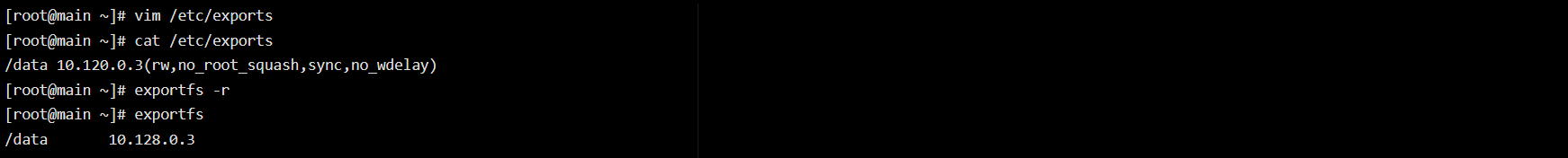


The disk was also partitioned; a new file system was created on top of the disk, followed by the creation of a new mount point for the same. Thus, the fstab file was modified in order to change permission and make it available to every instance in the site.

Immagine che contiene testo

Descrizione generata automaticamente

Access to the disk of both slaves was performed by installing the NFS server. The file system will be therefore exposed to the NFS client installed in these instances. In order to achieved that, the server configuration etc/exports file was edited. The mount point of the file system, the private ID of the slaves were both added in the file.



The directory /data was also created in the workers. To mount the remote file system, the private IP address of the main node was added to the fstab file.

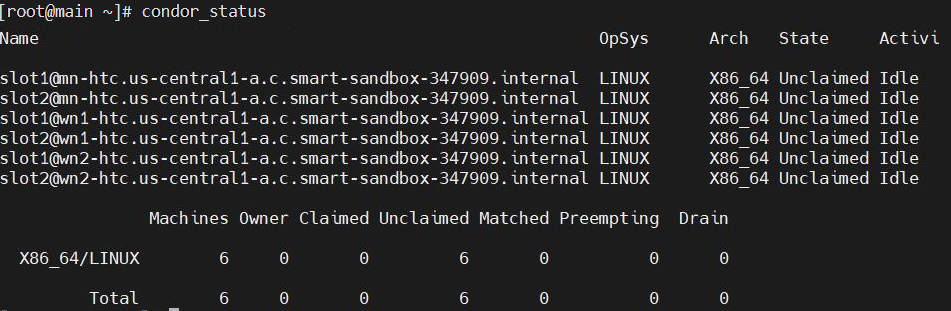
The batch system was managed by HTCondor (installed in all instances):

Immagine che contiene testo

Descrizione generata automaticamenteHTCondor was configured editing its configuration file condor-config. The following lines were added:



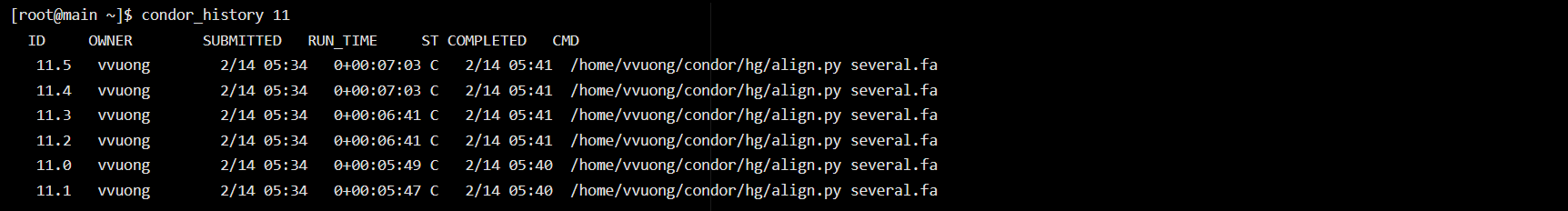
Condor was enabled and started:



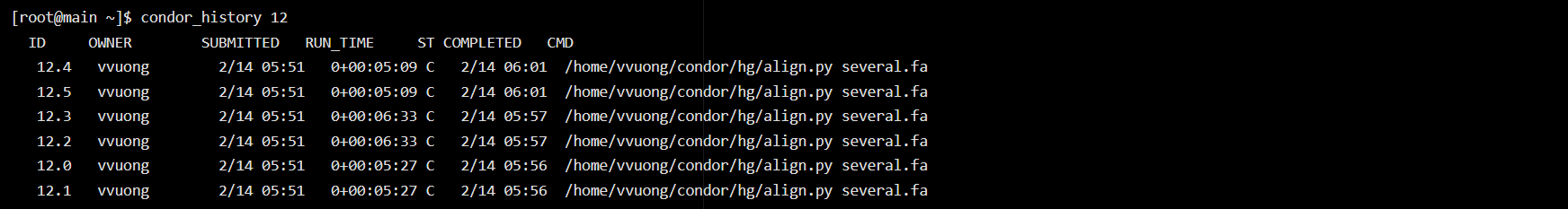
The condor-config file of both slaves was also edited. The private IP of the master node was added:



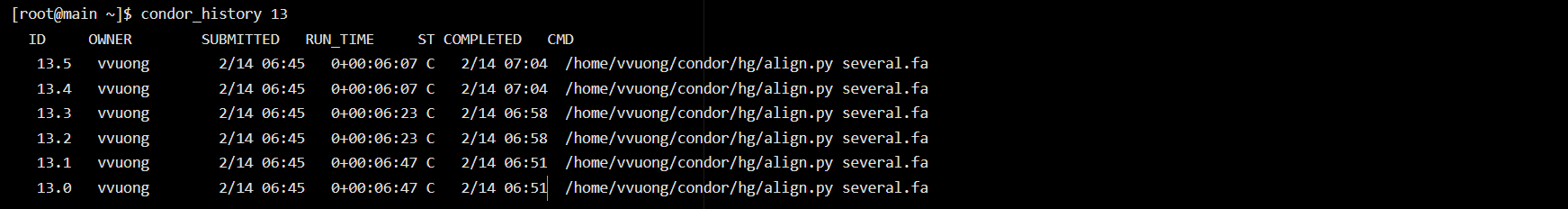
6 jobs were run (changing the QUEUE line in the bwa\\_batch.job to 6). At first, jobs were run in all instances:



then on two nodes:



then in a single node:

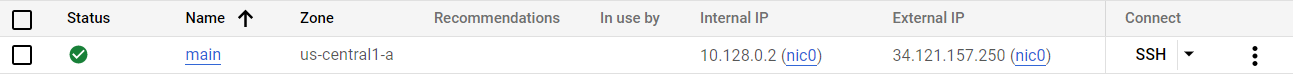


By running BWA with 6 jobs of 1M sequences each, a reduction in terms of running times was observed at the increasing number of cores. Assuming a linear increase, the time to complete the entire challenge varies from 39,6 hours with a single instance to 14,58 hours with three. As the hourly cost for an **e2-standard-2** machine is around 0,07$, the total cost to run the whole computational challenge is 2,77$, 2,92$ and 3,06$, respectively with 1, 2, 3, instances.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Numer of instances** | **Number of cores** | **Time for 6 jobs for 1M alignments (s)** | **Time to complete the challenge (h)** | **Expected cost ($)** | **Total ($)** |
| 1 | 2 | 1140 | 39,6 | 1\*39.6h\*0,07$ | 2,772 |
| 2 | 4 | 600 | 20,83 | 2\*20,83h\*0,07$ | 2,9162 |
| 3 | 6 | 420 | 14,58 | 3\*14,58h\*0,07$ | 3,0618 |

**2.2. High Performance Computing (HPC) site**

The HPC site provides CPU only for parallel jobs. In order to achieve this goal, a single multi-core instance was created (acting as both master and slave). This instance has 4 vCPUs and 4GB in memory (**e2-highcpu-4**). As the node already had 50GB in memory available, no additional disk was attached to it.



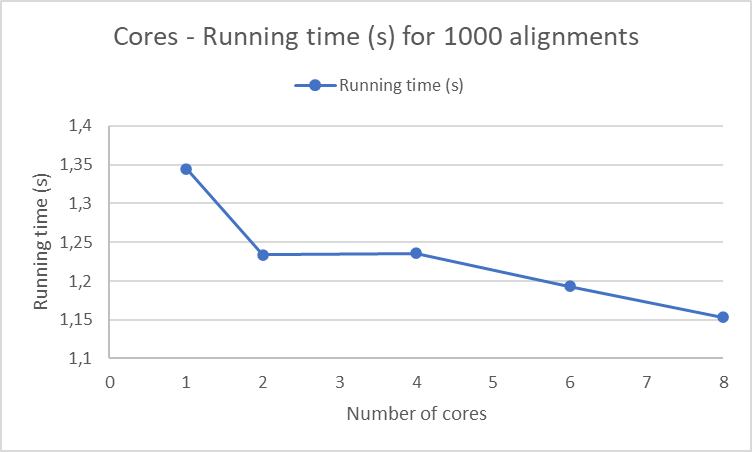
As HTCondor is not necessary in this case, BWA is directly installed on the master (BWA runs only on the main node):

Immagine che contiene testo

Descrizione generata automaticamente

BWA was tested running 1000 alignments:

|  |  |  |  |
| --- | --- | --- | --- |
| **Number of cores** | **Running time (s)** | **Efficiency** | **Speedup** |
| 1 | 1,345 | / | / |
| 2 | 1,234 | 0,5045 | 1,009 |
| 4 | 1,236 | 0,252 | 1,009 |
| 6 | 1,193 | 0,187 | 1,123 |
| 8 | 1,153 | 0,145 | 1,162 |



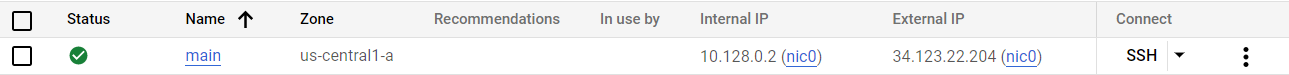
By running BWA on 1000 sequences, the running time equals to 1,345 seconds (time to upload the database, read it for the first time: from now on, it will be stored in the cache and, thus, this pre-processing is not required anymore). Looking at the plotted results, the speedup is modest due to the dimension of the job. Indeed, according to the Amdahl’s law the pre-processing phase (setting up the environment, loading the data, transforming the data) is scalar. This phase affects only one processor. From this moment on, the computational procedure can be parallelized, which means that the running time can be reduced by incrementing the number of cores. Therefore, in order to exploit the parallelization feature of BWA, the alignment tool works best on bigger jobs. 1M alignments were run.

The following plot indeed shows that the running time decrease rapidly by incrementing the number of cores. Notice also the speedup and efficiency.

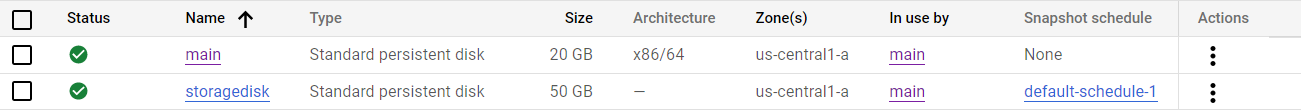
|  |  |  |  |
| --- | --- | --- | --- |
| **Number of cores** | **Running time for 1M (s)** | **Efficiency** | **Speedup** |
| 1 | 174,373 | / | / |
| 2 | 79,514 | 1,096 | 2,192 |
| 4 | 44,812 | 0,972 | 3,89 |
| 6 | 37,605 | 0,773 | 4,636 |
| 8 | 28,597 | 0,762 | 6,097 |

**2.3. Storage site**

This site only provides storage. A single **e2-micro** (2 vCPUs, 1GB in memory) instance was therefore created.



As the storage stores all output files of the alignment jobs run, a volume of 50GB was created, attached, partitioned and mounted as seen in section 2.1.

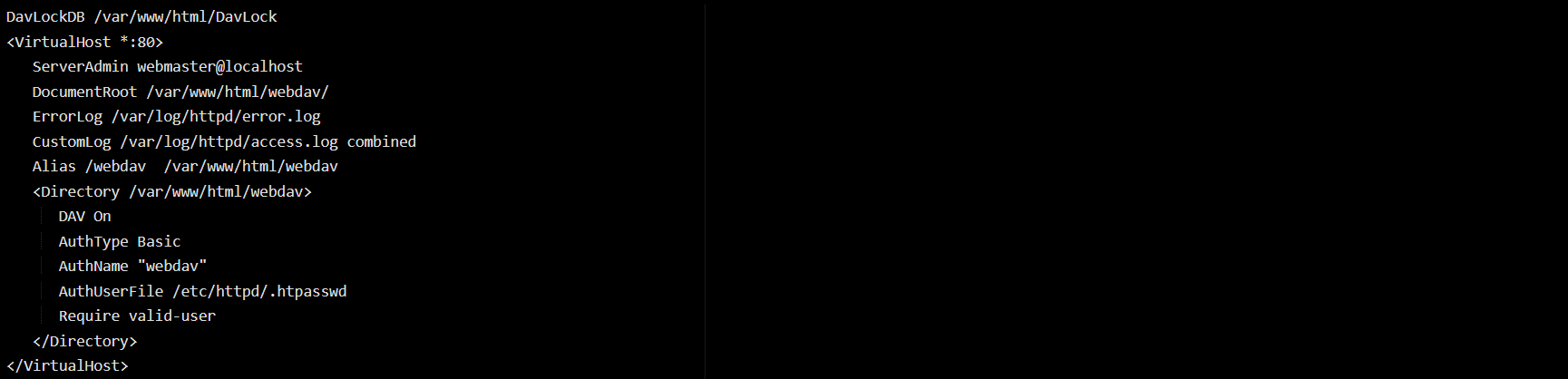


As the output file for 1000 alignments is 2.7GB, completing the entire challenge (300 000 alignments) will require at least 810 000GB of available storage.

To transfer data from all three sites, the data transfer tool WebDAV was installed here:

Immagine che contiene testo

Descrizione generata automaticamenteThe webdav.conf file was edited:

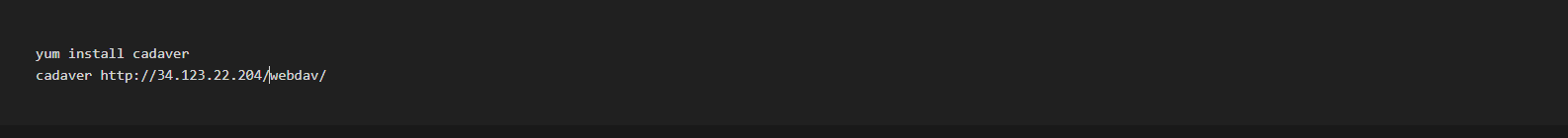


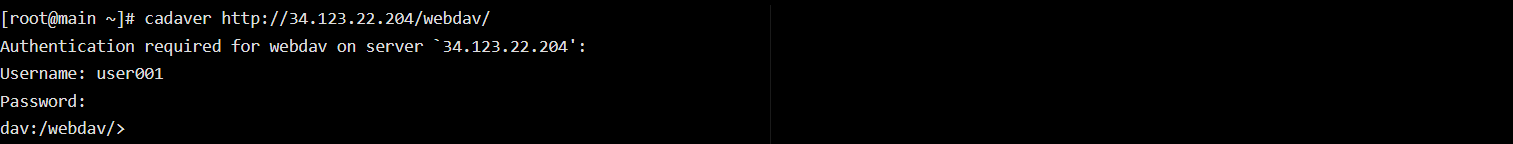
cadaver was installed to access the WebDAV server using the internal ID of the instance:

Immagine che contiene testo

Descrizione generata automaticamente

A firewall rule was installed in order allow the access of the HTC and HPC instances to WebDAV. Both sites can access to the server following the commands:





**3. Implementation of the computing model**

Here, the computational challenge is to align 300M sequences. These 300M sequences can be considered as 300 sets of 1 000 000 alignments, each having dimension of 170MB.

To exploit fully the computational resources of the HTC site and of the HPC site, the workload management system (WMS) pull model would be the best strategy. The WMS is a specific service able to manage the job distribution according to the given policy. It compares the user requirement with the available resources, selects which resources are going to be used, assigns a rank to the matches resources, and distributes jobs. In particular, the pull strategy will hold the job in queue until a resources become available. The pulling operation is performed by pilots, which make sure that resources are indeed available and pull the jobs out of the queue. This strategy would be the best one as jobs will be submitted according to the available resources present in that moment. Indeed, the model makes it easier to optimize the usage of the site resources, optimal for those challenges in which job's time and resources are not known apriori. The WMS could be installed in the third site, and pilots are distributed in the other two sites. As jobs are too big to be distributed through the WMS, WebDAV is exploited to transfer them to these sites. Since each job could be submitted in any working node, and it's not possible to known apriori in which site the input data will be necessary, the optimal data distribution strategy for this case seems to be the apriori push strategy. Following this strategy, all input data are distributed to all the nodes. Input data in the HTC site can be shared to all nodes using the shared volume, containing all necessary files (human genome's BWA indices and BWA alignment tool). This solution, though, does not deal with the problems of scalability and failover. if more than three nodes are present, the best option would be using a Storage Area Network to connect all the nodes and share the input data. Output files are stored on the site, zipped (to save space) through WebDAV to the storage site. The proposed computing model is therefore CPU-driven: jobs are sent where the computing power is available and data is transferred according to an apriori distribution.

**4. Model evaluation to solve the entire challenge (in terms of time and cost)**

To evaluate time and costs needed to complete the challenge of aligning 300M sequences to the human genome, 6 jobs of 1M sequences were run on the HTC site with a different number of instances in the infrastructure. The time required with one node (each one with 2 vCPUs and 8GB of memory) to compute 6 jobs with 1 million alignments is 1140 seconds, meaning that the time to complete 300M alignments is 39,6 hours. Since the hourly cost of an **e2-standard-2** is around 0,07$, the cost to complete the challenge would be of 2,772$. Increasing the number of instances, a slight increase of cost along with a reduction of the running hours is observed.

The total budget estimated for complete the challenge is the sum of the computational cost and the storage cost. Output files should be stored on the HTC and HPC sites, then sent long-term on the storage site. The output files of one job of 1M alignments is estimated to be 2700GB. Therefore, for 300M alignments, storage on HTC and HPC is 810 000GB. The standard storage cost is 0,026$/month, for a total of 0,026$/month\*810 000GB = 21 060$/month. The overall cost to store all processed data on the storage is 0,007$/GB/month, for a total of 0,007$/GB/month \* 810 000GB = 5670$ each month. In addition, the cost for storing jobs (51GB) on the storage is 1,326$/month.

In conclusion, the total budget to run the entire challenge and store the data for one month is approximately 28 058,772$.