

SUBORDINACIÓN DE SERVIDORES PARA CLÚSTER EN CLOUD USANDO AWS-EC2

David Jinete¹, Javier Aviles², Alexis Rojsa-Cordero³,

¹Grupo de Investigación FICBPG – Estudiante / Investigador
Institución Universitaria Politécnico Grancolobiano, Bogotá DC, Colombia.
jdjinete@poligran.edu.co

²Grupo de Investigación FICBPG – Estudiante / Investigador
Institución Universitaria Politécnico Grancolobiano, Bogotá DC, Colombia.
javaviles@poligran.edu.co

³Grupo de Investigación FICBPG – Profesor/ Investigador
Institución Universitaria Politécnico Grancolobiano, Bogotá DC, Colombia.
arojasco@poligran.edu.co

Resumen– Este artículo considera, a manera de guía, la implementación de una herramienta llamada StarCluster que nos ayuda a lograr establecer ambiente o sistema de computación robusto que genera altas prestaciones y que permite administrar de manera óptima los recursos de computación en la nube reuniendo lo necesario para que nuestros desarrollos que necesitan gran fuerza o capacidad de cómputo, puedan ejecutarse en menor tiempo.

Para muchos departamentos de TI e investigación se ha vuelto necesario el estar al día en temas de desarrollo e implementación de nuevas tecnologías que puedan procesar grandes volúmenes de datos o realizar cálculos gigantescos; hoy por hoy la tecnología CLOUD o computación distribuida permite a sus usuarios obtener resultados más rápidos que los que podríamos realizar en una máquina convencional y sin la necesidad de gastar

largos periodos de tiempo, dinero e infraestructura en una construcción o implementación de una potencia computacional que brinde ese rendimiento.

La instalación y configuración de esta arquitectura usando StarCluster [1] es nuestro plan de vuelo para subordinación de servidores usando EC2 de Amazon como plataforma de recursos de cómputo. La orquestación de las maquinas adquiridas en la plataforma de computación elástica de Amazon crean según la necesidad de un grupo investigación o TI, un ambiente propicio para aquellos proyectos que requieren gran capacidad de computación. Esta herramienta nos ayuda a establecer en menos tiempo según estos requisitos nuestro Clúster, administrar bien los recursos informáticos y poder reunir lo necesario para que nuestros desarrollos que necesitan gran fuerza o capacidad de cómputo puedan obtener los resultados en menor tiempo.

Palabras Clave -- TI, Computación avanzada, Cloud Computing, computación distribuida, Sistemas distribuidos, administración de sistemas en la Nube, sistemas de altas prestaciones, StarCluster, Amazon EC2.

I. INTRODUCCIÓN

El interés de muchas empresas y departamentos de investigación de poder realizar los cálculos, ordenar, buscar, clasificar más rápido estas operaciones, mejorar los tiempos de respuesta del procesamiento de los datos y al igual que el crecimiento de los volúmenes de datos que hora tras hora o día tras día crecen de una forma exponencial han llevado a estos adentrarse a la computación distribuida, buscar aprovechar mejor los recursos que se tienen y poder tener mejor rendimiento.

El poder optimizar el uso de los recursos también ha llevado a estas empresas a adquirir esta tecnología que nos lleva a no solo tener resultados más pronto, sino que también a ahorrar cientos de miles o hasta cientos de millones de pesos en el mantenimiento de dispositivos o máquinas físicas que con el tiempo van quedando, para aquellos que las adquieren, solo quedan como activos obsoletos. Varias de las cosas que están afectando a los usuarios de servicios de altas prestaciones son:

- La falta de financiación para el hardware y el software.
- Falta de apoyo (nadie para gestionar y optimizar el clúster).
- La falta de disponibilidad flexible de los recursos locales o internos de HPC.
- Un flujo de trabajo que incluye "ráfaga" o despliegue rápido.
- Un flujo de trabajo que requiere escalabilidad en todo momento [2].

Es por ello que surge la pregunta de, ¿cómo obtener, administrar, controlar y gestionar los recursos de computación de alto rendimiento, sin adquirir mucho hardware, sino que sean ofrecidos en un ambiente de computación en la nube?

II. Marco teórico

Para entender cómo funciona estas plataformas hay que saber en qué consisten los elementos que intervienen y el concepto inicial es los sistemas distribuidos, “Un sistema distribuido es en el que las partes dentro de los computadores, enlazados en red, comunican y coordinan funciones, todo a través del paso de mensajes.”. [3] En otras palabras, es la técnica que permite presentar cada recurso de un sistema interconectados para compartir u ofrecer sus servicios de computación o informática.

Algunas de las partes de esta distribución requieren que existan, entre muchos componentes que son parte fundamental de este, como son los núcleos de procesamiento, memoria RAM, almacenamiento en arreglos de discos, las tarjetas de red que nos comunican con otras redes, las plataformas de software, etc. También es necesario que estos servicios no estén expuestos a una red pública y se pueda escalar o de otra manera modificar sus componentes y demás.

Todos estos elementos poseen una configuración que permite establecer en conjunto un pool de recursos para que el usuario pueda administrar (poner en marcha) toda la potencia de cómputo que se ha construido. K jorissen define cloud computing [4] como "un paradigma computacional en el que los recursos

virtualizados dinámicamente escalables se proporcionan como un servicio a través de Internet".

Este conjunto de recursos en el mundo se conoce como la Computación de altas prestaciones de sus siglas en inglés HPC (High Performance Computing), conocida como la práctica de agregar potencia computacional de una manera que ofrece un rendimiento mucho mayor que el que uno podría encontrar en una típica computadora de escritorio o estación de trabajo; esto con el fin de resolver grandes problemas de datos en las ciencias, la ingeniería o los negocios. Por ejemplo los bancos y las empresas de inversión ahora utilizan HPC para analizar datos a la velocidad extremadamente alta necesaria para ejecutar órdenes en el comercio de alta frecuencia (HFT). Los fabricantes de automóviles, aviones y otros equipos industriales utilizan HPC con potentes herramientas de ingeniería automatizada (CAE) que producen complicadas simulaciones de diseño y evaluaciones de prototipos. [5] Otro ejemplo son los análisis que requieren ser controlados por un computador, como es el caso de Modelos matemáticos y experimentales sobre el secado de biomasa [6] que requieren revisar grandes volúmenes de datos

El modelo de HPC hoy en día se muestra como una buena distribución de varios nodos (computadoras o servidores) presentados al servicio de una máquina principal MÁSTER la cual reparte las tareas en un modelo paralelo de computación que permite entregar resultados más rápido, la labor en algunos casos del MÁSTER es también recopilar y clasificar los datos.

Cuando se habla de CLOUD se hace referencia a los servicios que se ofrecen en internet. Cloud Computing tiene que ver con la gestión y administración que se le da al Software y Hardware dentro de una plataforma distribuida, incluye un gran número de computadoras conectadas sobre una red de internet.

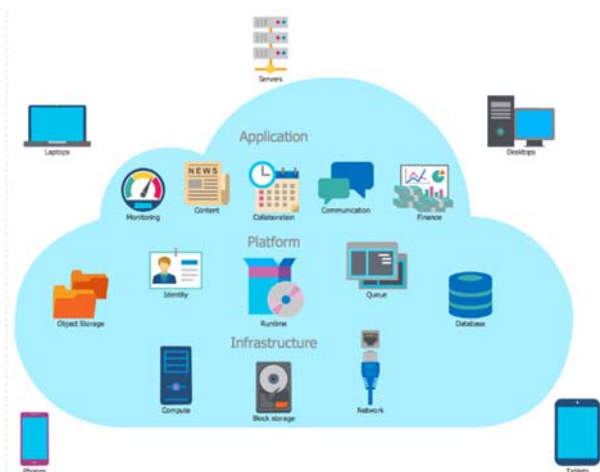


Figura 1 Esquema de Cloud. Fuente: <http://www.conceptdraw.com/examples/cloud-computing>

Hasta este momento no se ve nada fácil, ya que intervienen varios componentes y configuraciones; para llegar a qué todo este proceso sea más amigable y tenga un nivel de practicidad que permita al usuario tener estos recursos y gestionarlos de la mejor manera. Una manera de realizar esta gestión es usar una herramienta de software libre (StarCluster) que le permite al usuario realizar el montaje de un servicio con altos niveles de procesamiento a altas prestaciones en particular en la red de Amazon EC2. Para entender cuantitativamente la capacidad de nuestro cluster de una nube, los investigadores o programadores pueden elegir puntos de referencia que se asemejen al consumo de recursos que generan sus aplicaciones. Según el Caso de estudio de Qiming He [7] los programadores conocen diferentes tipos de referencias que seleccionar las características que se desea que tenga el

cluster.

Cabe resaltar que el desempeño de nuestro clúster puede verse afectado cuando se realizan tareas en paralelo, debido al paso de mensajes entre los nodos tiene su dependencia en la velocidad de la Red. Este problema es fácil de prever siempre y cuando se realicen configuración a conexiones especializadas como Myrinet o Infiniband. [8]

Antes de continuar debemos mencionar que los servicios de computación en la nube son pagos y algunos como en nuestro caso AMAZON nos permite realizar una prueba con una cuenta gratuita por un año, en la cual hemos realizado nuestra implementación.

Hablemos entonces de StarCluster; de acuerdo a la página oficial, StarCluster es un conjunto de herramientas para facilitar construcción, configuración y la gestión de máquinas en Clúster para poder generar un HPC y hacer más sencillo el proceso de configuración de clúster en la nube EC2 de Amazon. [9]

Este proyecto nace del programa de investigación de la Universidad de [10] donde se realizan estudios de Biología, genética, hidrología, informática, etc. Proyectos en los que son necesario realizar cálculos que requieren HPC.

StarCluster es una herramienta diseñada para ayudar a que todos estos componentes junto con su administración dejen de ser una preocupación, permitiendo establecer una configuración entre máquinas y así unirlos para que trabajen en las tareas de forma paralela.

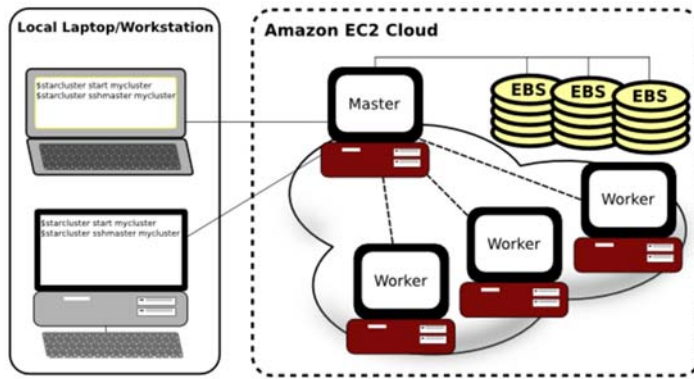


Figura2 Arquitectura de StarCluster. Fuente:

<http://star.mit.edu/clúster/docs/latest/overview.html>

En esta herramienta se realiza el trabajo pesado cuando se trata de crear y configurar un clúster en EC2. [9]

Se encarga de la configuración de cada componente que posee Amazon para la creación de un conjunto de máquinas en su red, entre esos realiza:

- Configuración de reglas de seguridad, la red a la que están conectadas y el acceso a las máquinas.
- Administra el inventario de las máquinas o nodos.
- Las cuentas de usuario de los nodos.
- La administración de las credenciales de acceso SSH.
- El protocolo de transferencia de datos NFS.
- Configura y administra los bloques de almacenamiento persistente de las siglas en ingles EBS.
- Configura y administra el sistema de particiones de los Discos.
- Administra las tareas o procesos que se ejecuten en las máquinas.

Amazon define EC2 como un servicio web que permite brindar a sus subscriptores acceso a gran capacidad informática, que es adaptable según la demanda de recursos que se requiera en cualquier instante de tiempo [11]

III. Implementación de un modelo.

El proceso de instalación de StarCluster inicia desde una máquina que hemos iniciado en un ambiente virtualizado, que se encargará de la orquestación y comunicación a la plataforma de AMAZON. La máquina virtualizada posee como sistema Operativo la distribución 14.04 LTS de Ubuntu Linux.

Nota: Debido a que StarCluster está desarrollado en el lenguaje Python es indispensable contar con la versión de Python 2.7 o superior, la cual necesitaremos en el proceso de instalación de StarCluster, dentro de una ventana de terminal o consola de comandos.

\$ sudo easy_install StarCluster

Otra forma de realizar el proceso de instalación de StarCluster es proceder a descomprimir el archivo descargado de la página oficial de StarCluster que contiene todo el conjunto de dependencias y librerías requeridas para su proceso de instalación.

\$ wget “se coloca la ruta de descarga brindada en la página

<http://pypi.python.org/pypi/StarCluster>”.

\$ tar -xvf StarCluster-0.95.06.tar.gz

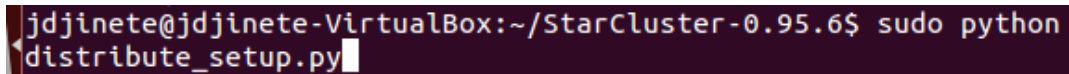
\$ cd StarCluster-X.X.X

Seguido de este proceso se corre el comando:

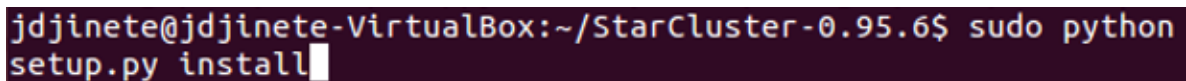
\$ sudo python distribute_setup.py

Con el cual se procede a realizar las configuraciones de la distribución, una vez se termina el proceso de instalación de las herramientas de Python se ejecuta el comando:

\$ sudo python setup.py install



```
jdjinete@jdjinete-VirtualBox:~/StarCluster-0.95.6$ sudo python  
distribute_setup.py
```



```
jdjinete@jdjinete-VirtualBox:~/StarCluster-0.95.6$ sudo python  
setup.py install
```

figura 3: Comandos de inicio de configuración Python. Fuente: Los Autores

Luego de finalizar este proceso de instalación, se puede validar la correcta instalación de StarCluster con ayuda del comando

\$ StarCluster help

Nos indicará un error, indicando que el archivo de configuración no existe:

`/home/user/.StarCluster/config`

Nos da la opción de escribir este archivo de configuración con los datos por defecto, y lo ubica en una ruta específica, lo cual lo revisaremos con el editor vim.

\$ vim ~/.StarCluster/config

La cuenta con AWS ya debe estar creada y dentro del archivo .config se agregan las credenciales o información propia de nuestra cuenta:

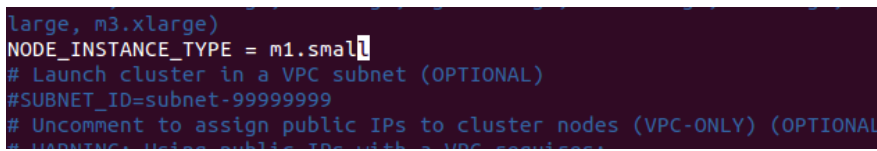
[aws info]

aws_access_key_id = #SU access key id DE AMAZON AQUÍ

aws_secret_access_key = #SU secret access key DE AMAZON AQUI

aws_user_id = #SU user-id DE AMAZON AQUI (SIN LAS LINEAS DE SEPARACION)

En este archivo también se puede proceder a configurar información relevante de las máquinas que van hacer creadas, por ejemplo: El tipo de instancia, capacidad de almacenamiento, la cantidad de nodos, etc. Otro parámetro muy útil en el proceso de instalación de StarCluster es el DEFAULT_TEMPLATE donde podemos usar la palabra “smallcluster” para obtener un pequeño clúster con mínima características de recursos.



```
large, m3.xlarge)
NODE_INSTANCE_TYPE = m1.small
# Launch cluster in a VPC subnet (OPTIONAL)
#SUBNET_ID=subnet-99999999
# Uncomment to assign public IPs to cluster nodes (VPC-ONLY) (OPTIONAL)
# WARNING: Using public IPs with a VPC requires
```

Figura 4. Archivo de configuración de StarCluster .config. Tipos de Nodos a lanzar.

Fuente: Los Autores

```
# out-of-the-box provided you've configured your keypair correctly
DEFAULT_TEMPLATE=smallcluster
# enable experimental features for this release
#ENABLE_EXPERIMENTAL=True
# number of seconds to wait when polling instances (default: 30s)
#REFRESH_INTERVAL=15
# specify a web browser to launch when viewing spot history plots
#WEB_BROWSER=chromium
# split the config into multiple files
#INCLUDE=~/.starcluster/aws, ~/.starcluster/keys, ~/.starcluster/vols

#####
## AWS Credentials and Connection Settings ##
#####
[aws info]
# This is the AWS credentials section (required).
# These settings apply to all clusters
# replace these with your AWS keys
AWS_ACCESS_KEY_ID = #your_aws_access_key_id
AWS_SECRET_ACCESS_KEY = #your_secret_access_key
# replace this with your account number
AWS_USER_ID= #your_userid
```

Figura 5. Archivo de configuración de StarCluster .config parámetros de acceso a AWS

Fuente: Los Autores

Siguiendo con el proceso de configuración de StarCluster, dos parámetros adicionales que son importantes de configurar en este archivo son: el KEY_LOCATION que es donde agregamos la ruta de la key pair que nos brinda AWS EC2; y el KEYNAME que será el nombre o alias usado para acceder con dicha llave.

```
# --help" for instructions on how to create a new ke
# match your key name e.g.:
[key mykeyx]
KEY_LOCATION=~/.ssh/mykeyx.rsa

# You can of course have multiple keypair sections
# [key myotherkey]
# KEY_LOCATION=~/.ssh/myotherkey.rsa

#####
```

Figura 6. Archivo de configuración de StarCluster. Parámetro de ubicación de llave de acceso a AWS Fuente: Los Autores.

En esta imagen se puede apreciar la configuración del parámetro CLÚSTER _SIZE=2 el cual nos está indicando, que se creará un clúster con dos nodos: el nodo máster y el nodo hijo. Adicionalmente, muestra que el usuario de configuración de los nodos será *sgeadmin*.

```
# change this to the name of one of the keypair
KEYNAME = mykeyx
# number of ec2 instances to launch
CLUSTER_SIZE = 2
# create the following user on the cluster
CLUSTER_USER = sgeadmin
# optionally specify shell (defaults to bash)
```

Figura 7. Archivo de configuración de StarCluster. Parámetros de tamaño del Clúster y el Nodo que se va a establecer como principal o Máster. Fuente: Los Autores.

Adicionalmente para realizar la implementación se requiere agregar dentro del mismo archivo las siguientes líneas en la sección “*defining clúster templates*”

```
NODE_IMAGE_ID = ami-6b211202
# instance type for all cluster nodes
# (options: m3.large, c3.8xlarge, i2.8xlarge, t2.micro,
r3.4xlarge, g2.2xlarge, m1.small, c1.medium, m3.2xlarge
m2.2xlarge, t2.small, r3.2xlarge, t1.micro, cr1.8xlarge
, m1.medium, r3.large, c3.xlarge, i2.xlarge, m3.medium,
g1.4xlarge, i2.2xlarge, c3.large, i2.4xlarge, c3.4xlarge
hi1.4xlarge, m2.4xlarge, m1.xlarge, m3.xlarge)
NODE_INSTANCE_TYPE = t2.micro
PLUGINS = mpich2
[plugin mpich2]
SETUP_CLASS = starcluster.plugins.mpich2.MPICH2Setup
#PLUGINS = mpich2
# Launch cluster in a VPC subnet (OPTIONAL)
SUBNET_ID=subnet-99999999
```

Figura 8. Archivo de configuración de StarCluster. Parámetros de la instancia y el plugins a utilizar. Fuente: Los autores.

Para la configuración por defecto que viene en el archivo .config de StarCluster, viene una *id_ami* Ubuntu 13.04, OpenMPI, NFS. En ocasiones la versión de OpenMPI no es compatible para la ejecución de tareas usando procesamiento paralelo, por lo cual es necesario agregar la siguiente línea para incluir el plugin de mpich2:

PLUGINS = mpich2

[plugin mpich2]

SETUP_CLASS = StarCluster.plugins.mpich2.MPICH2Setup

MPI es un proyecto de código abierto diseñado para ofrecer el beneficio de una interfaz de paso de mensajes, esta es muy útil para el desarrollo de tareas en paralelo en sistemas distribuidos. [12]

La siguiente figura muestra una breve descripción de los parámetros del archivo de configuración modificados en nuestra implementación.

Propiedad	Descripción
Vim ~/.starcluster/config	Acceso a configuración archivo de starcluster para configuración de cluster.
AWS_ACCESS_KEY_ID = 23123klj123klj3	Key id de la cuenta AWS.
AWS_SECRET_ACCESS_KEY = asd98asd98sds98	Key access de la cuenta AWS.
AWS_USER_ID=390021971709222	Userld de la cuenta.
[cluster smallcluster]	Etiqueta para definir el nombre del tipo de cluster a configurar.
KEYNAME = mykeyy	Nombre de la keypair descargada de AWS EC2.
CLUSTER_SIZE = 3	Número de nodos del cluster.
CLUSTER_USER = sgeadmin	Username para los nodos.
CLUSTER_SHELL = bash	Interprete de órdenes
NODE_IMAGE_ID = ami-12b6477b	Id de la imagen AWS a configurar en los nodos.
NODE_INSTANCE_TYPE = t2.micro	Tipo de instancia a configurar según características AWS.
Configuración de plugins adicionales.	
PLUGINS = mpich2	Nombre del plugin a llamar.
[plugin mpich2]	Etiqueta para crear el puglin
SETUP_CLASS = starcluster.plugins.mpich2.MPICH2Setup	Fuente del plugin.

Figura 9. Parámetros base para crear nuestro Clúster en EC2 de AWS. Fuente: Los Autores

Una vez realizado los pasos anteriores correctamente de configuración de parámetros del archivo .config de StarCluster, podremos iniciar la creación de nuestro clúster usando el comando:

“StarCluster start -c smallcluster nombre clúster” como lo muestra la imagen.

```
jinetesis@PCJineteTesis:~/StarCluster-0.95.6$ sudo starcluster start -c smallcluster myipcluster
StarCluster - (http://star.mit.edu/cluster) (v. 0.95.6)
Software Tools for Academics and Researchers (STAR)
Please submit bug reports to starcluster@mit.edu

>>> Validating cluster template settings...
>>> Cluster template settings are valid
>>> Starting cluster...
>>> Launching a 2-node cluster...
>>> Creating security group @sc-myipcluster...
Reservation:r-058ecc9a0d5851a30
>>> Waiting for instances to propagate...
2/2 | 100%
>>> Waiting for cluster to come up... (updating every 30s)
>>> Waiting for all nodes to be in a 'running' state...
0/2 | 0%
```

Figura 10. Ejecución de comando StarCluster start. Fuente: Los Autores

Iniciado nuestro clúster, podemos observar como éste es construido desde las mismas instancias de AWS EC2.

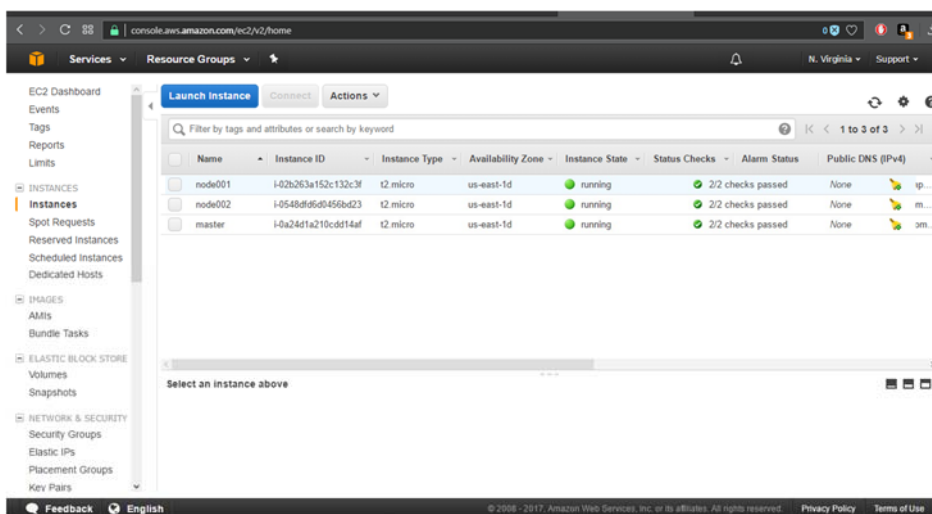


Figura 11. Vista del Dashboard de AWS, con la cuenta y las instancias lanzadas y en ejecución. Fuente: [11]

Implementación del algoritmo en la máquina Máster

Después de iniciado el clúster podemos acceder al nodo máster con el comando:

\$ StarCluster sshmáster nombredemáster

Una vez accedida a la consola de nodos podemos ejecutar modelos de pruebas, en nuestro caso trabajaremos compilando un programa sencillo que nos indica nodo corriendo y la fecha actual.

```
#include <stdio.h> /* printf and BUFSIZ defined there */
#include <stdlib.h> /* exit defined there */
#include <mpi.h> /* all MPI-2 functions defined there */
#include <time.h>

int main(argc, argv)
{
    int argc;
    char *argv[];

    int rank, size, length;
    char name[BUFSIZ];

    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);
    MPI_Get_processor_name(name, &length);

    printf("-----CORRIENDO PROCESO-----\n");
    printf("%s: Somos Starcluster de myipcluster4 proceso %d de %d\n", name, rank, size);

    time_t tiempo = time(0);
    struct tm* tlocal = localtime(&tiempo);

    char output[128];
    printf("-----\n");
    strftime(output, 128, "%d/%m/%y %H:%M:%S\n", tlocal);
    printf("Fecha y hora: %s", output);
    printf("-----\n");
    MPI_Finalize();

    return(0);
}
```

Figura 12. Algoritmo de prueba que se ejecutará en el nodo Máster. Fuente: Los Autores

```

root@master:/home/sgeampirun -n 3 -host master,node001,node002 ./HelloWorld
-----CORRIENDO PROCESO-----
master: Somos Starcluster de myipcluster4 proceso 0 de 3

Fecha y hora: 29/05/17 22:30:59:

-----CORRIENDO PROCESO-----
node002: Somos Starcluster de myipcluster4 proceso 2 de 3

Fecha y hora: 29/05/17 22:31:00:

-----CORRIENDO PROCESO-----
node001: Somos Starcluster de myipcluster4 proceso 1 de 3

Fecha y hora: 29/05/17 22:30:57:

```

Figura 13. El Resultado de la ejecución del algoritmo, llamado HolaMundo.c Fuente: Los Autores

A continuación se muestra las características generadas en AWS de un nodo específico del clúster.

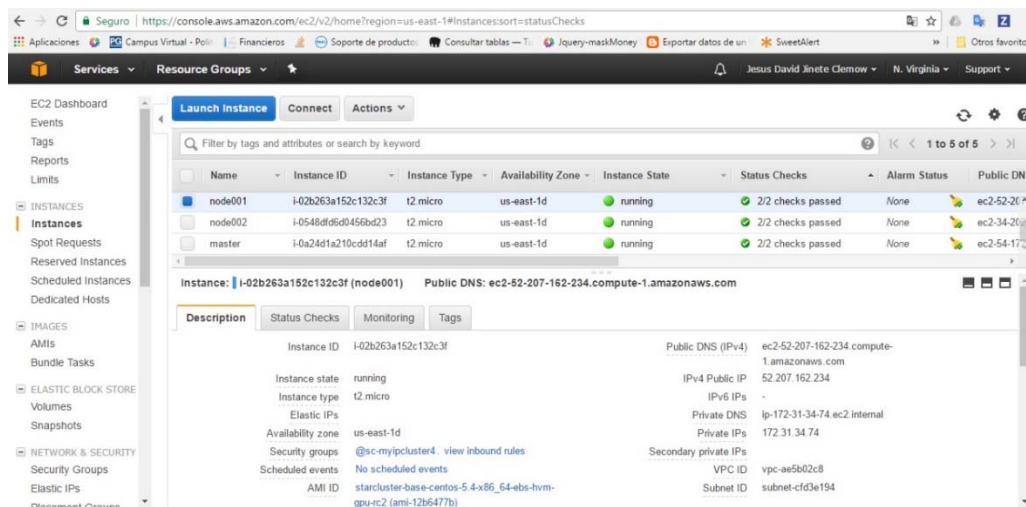


Figura 14. Características del node001 del clúster myipcluster4 en EC2 de AWS, Fuente:

En EC2 de AWS podemos apreciar en su consola administrativa, como ha sido el comportamiento de las ejecuciones y consumos del nodo máster desde su inicio, donde consumió una gran parte de los recursos seguido de pequeños picos que hacen referencia a los momentos en los que se ejecuta el programa y al final como disminuye su consumo con la finalización del mismo, con el comando “StarCluster terminate nombredelclúster”.

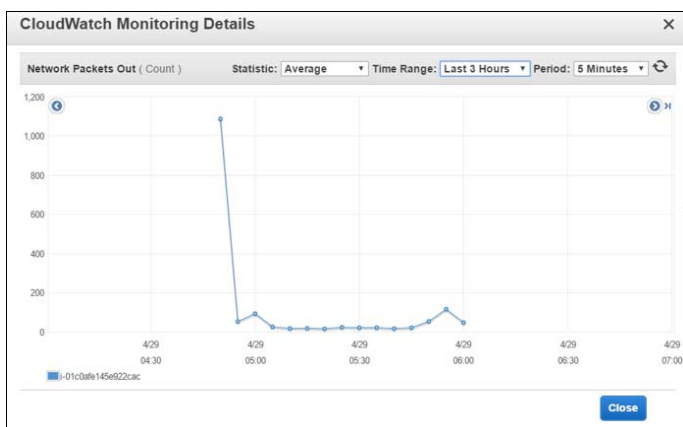


Figura 15. Historial del uso del Procesamiento de uno de los nodos. Fuente: [13]

Conclusiones

Nuestra implementación muestra como a través de estos pasos, se puede realizar una implementación de procesos en paralelo o bien proyecto que requiera altas prestaciones.

Podemos decir que realizar el trabajo de una construcción de un ambiente de clúster se hace más fácil reuniendo una herramienta que permite crear o modificar plantillas de configuración de servidores y una granja de servicios donde podemos obtener los recursos en menos tiempo y no pensar la adquisición de partes o máquinas

físicas que podrían quedarse obsoletas. Nos lleva a pensar más en la definición de un perfil de trabajo, construyendo ambientes que se ajustan a la necesidad gracias también a la elasticidad que brinda la nube publica de Amazon Web Services.

StarCluster es una herramienta que facilita el trabajo de la administración y gestión de clústeres remotos en EC2. El uso de este tipo de herramientas, nos brinda la oportunidad de hacer más sencilla todas esas tareas que pueden llegar hacer tediosas, complejas e importantes para la construcción de un clúster y especialmente uno en la nube. Desde el proceso de configuración de redes internas para la comunicación una red distribuida, adición de recursos o características de cada uno de los nodos como RAM, espacio de almacenamiento, sistema operativo, hasta la posibilidad de adicionar plugins adicionales en los nodos.

Otra importante característica de esta valiosa herramienta, son los paquetes que brinda automáticamente en el proceso de construcción del clúster entre los cuales incluye una configuración de un sistema de archivos de red alojado en la ruta `/home/username/opt` y que permite compartir todo tipo de archivos entre las diferentes maquinas del clúster, paquetes de OpenMPI para el desarrollo de tareas paralelas y el software Sun Grid Engine (SGE) para gestionar de sistemas de recursos computacionales.

Esta herramienta se convierte en el orquestador de la infraestructura de nuestros proyectos de investigación o desarrollo que requieren grandes recursos de computación incluyendo también el almacenamiento

Referencias

StarCluster. (2017, abril 30). *StarCluster*. Retrieved from

<http://star.mit.edu/cluster/index.html>

- [1] StarCluster, "StarCluster," 30 abril 2017. [Online]. Available: <http://star.mit.edu/cluster/index.html>.
- [2] actnowhpc, "A sampling of use cases HPC in the Cloud:A Solution," 30 abril 2016. [Online]. Available: <http://actnowhpc.com/wp-content/uploads/2016/05/Cloud-Case-Studies.pdf>.
- [3] Distributed Systems. George Coulouris, in *Concept and Design*, Boston, Pearson, 2008, p. 20.
- [4] F. a. J. K.Jorissen, "A high performance scientific cloud computing environment for materials simulations," *Elsevier*, vol. 218, no. 9, pp. 1911-1919, 2012.
- [5] Bright Computing, "<http://www.brightcomputing.com/product-offerings/bright-cluster-manager-for-hpc>," 30 abril 2017. [Online]. Available: <https://cdn2.hubspot.net/hubfs/143457/eBook-HPC-v3.pdf>.
- [6] Jorge Armando Villalba Vidales, Nelson Arzola de la Peña, "Modelos matemáticos y experimentales sobre el secado de biomasa," *Revista Científica Ingeniería y Desarrollo*, vol. 33, no. 2, pp. 305-306, 25 abril 2015.
- [7] B. K. D. D. M. Shujia Zhou, *Case Study for Running HPC Applications in Public Clouds*, Chicago: HPDC, 2010.
- [8] C. E. a. C. N. Hill, "Cloud Computing for parallel Scientific HPC Applications," *CiteSeer*, vol. 1, no. 5, p. 5, 2008.
- [9] MIT, "Star," 30 Abril 2017. [Online]. Available: <http://star.mit.edu/cluster/docs/latest/overview.html>.
- [10] Massachusetts Institute of Technology, "Massachusetts Institute of Technology," 2017. [Online]. Available: <http://web.mit.edu/>.

- [11] AWS Amazon, "Amazon Web Services AWS Amazon," 30 Abril 2017. [Online]. Available: <https://aws.amazon.com/es/ec2/>.
- [12] open-mpi.org/, "https://www.open-mpi.org/," 16 mayo 2017. [Online]. Available: <https://www.open-mpi.org/>.
- [13] AWS, "Amazon," 30 abril 2017. [Online]. Available: <https://us-west-2.console.aws.amazon.com/console/home?region=us-west-2#>. [Accessed 30 abril 2017].