

Pruebas Cóndor – Universo Vanilla

Contenido

Requisitos	2
Definición de Comandos de Cóndor	2
Listado de Pruebas	2
Estructura de Directorios	3
Definición de Pruebas.	3
Prueba 1	3
Prueba 2	4
Prueba 3	4
Autor	5

Este documento contiene algunos pasos con las cuales se probara la disponibilidad y confiabilidad del clúster C ndor instalado previamente.

El Objetivo principal es probar el universo *vanilla*, Universo que se encuentra en C ndor y generar evidencia de que ya existen cl ster activos y disponibles.

Requisitos

Para que las pruebas corran exitosamente se necesita contar con la instalaci n correcta del aplicativo **C ndor**,

El primer paso es verificar que los nodos se encuentran disponibles para recibir trabajos, para esto lanzamos el comando *condor_status* el cual se preinstala una vez se realiza la instalaci n de C ndor. La salida ser  algo parecido a:

```
[root@gc1-ce ~]# condor_status
Name           OpSys  Arch  State   Activity LoadAv Mem  ActvtyTime
gc1-c001.javeriana LINUX   INTEL Unclaimed Idle    0.000 501 0+00:18:32
gc1-c002.javeriana LINUX   INTEL Unclaimed Idle    0.020 501 0+00:18:17
gc1-c003.javeriana LINUX   INTEL Unclaimed Idle    0.000 501 0+00:18:15
gc1-c004.javeriana LINUX   INTEL Unclaimed Idle    0.000 501 0+00:18:06
gc1-c005.javeriana LINUX   INTEL Unclaimed Idle    0.000 501 0+00:18:02
      Total Owner Claimed Unclaimed Matched Preempting Backfill
      INTEL/LINUX    5    0    0    5    0    0    0
```

Una vez esto este cumplido podremos ejecutar las pruebas sin problema alguno.

Definici n de Comandos de C ndor

En la ejecuci n de las pruebas se usaran 3 comandos b sicos de C ndor el cual se expondr n a continuaci n.

condor_status: Obtiene la informaci n de las maquinas pertenecientes al cl ster y datos adicionales como carga de procesador y *uptime*.

condor_q: Obtiene la cola de trabajos que tiene el C ndor programados.

condor_submit: Mediante un archivo de configuraci n procede a ejecutar un trabajo en dicho cl ster.

Listado de Pruebas

El paquete cuenta con Varias pruebas previamente probadas

Test1: Prueba el cl ster a nivel de sistema operativo, devuelve los nombres de las maquinas para saber que maquinas est n activas y trabajando correctamente en el cl ster.

Test2: Prueba una aplicaci n sencilla desarrollada en lenguaje C en las maquinas del cl ster y f cilmente parametrizable.

Test3: Escala la prueba anterior lanzando N numero de procesos del mismo aplicativo para medir el rendimiento del clúster y realizar pruebas de stress al mismo.

Estructura de Directorios

Los Directorios de las pruebas están definidos de esta manera.

```
/ (Directorio Raíz)
/test1 (Resultados de la Prueba 1)
/test2 (Resultados de la Prueba 2)
/test3 (Resultados de la Prueba 3)
test (Aplicativo compilado para realizar prueba 2 y 3)
test1.submit (Archivo de configuración para prueba 1)
test2.submit (Archivo de configuración para prueba 2)
test3.submit (Archivo de configuración para prueba 3).
```

Esta estructura se debe crear en el *home* del usuario creado previamente.

El aplicativo puede recibir 3 parámetros las cuales son:

Límites: Pone un límite tanto superior como inferior para buscar los divisores del primer número digitado. Ej.:/test 1234567 3 123.

Definición de Pruebas.

Una vez se ejecute el comando *condor_submit* se debe verificar que el headnode (SCHEDD) previamente configurado halla procesado la cola de mensajes correctamente con el comando *condor_q*.

Prueba 1 (Prueba el clúster a nivel de sistema operativo)

Archivo test1.submit

```
Executable = /bin/hostname
Universe = vanilla
output = logTest1.$(Process)
error = logTest1.$(Process)
Log = logTest1.log
should_transfer_files = YES
when_to_transfer_output = ON_EXIT
Queue 5
```

Para ejecutar la prueba seguir los siguientes pasos:

```
[root@gc1-ce ~]# pwd
/root
[root@gc1-ce ~]# su - condor
[condor@gc1-ce ~]$ pwd
/home/condor
[condor@gc1-ce ~]$ chmod 777 test1.submit
```

```
[condor@gc1-ce ~]$ cd test1/
[condor@gc1-ce test1]$ condor_submit ../test1.submit
```

NOTA: cambiar el parámetro *Queue* con el número de *working nodes* en el clúster.

Prueba 2 (Prueba una aplicación sencilla)

Para ejecutar esta prueba es necesaria la compilación del archivo test.c (carpeta resources) con el compilador de gcc instalado por defecto.

Archivo test2.submit

```
Executable = test
Universe = vanilla
output = logTest2.$(Process)
error = logTest2.$(Process)
Arguments = 1234567 3 123
Log = logTest2.log
should_transfer_files = YES
when_to_transfer_output = ON_EXIT
Queue 5
```

Para ejecutar la prueba seguir los siguientes pasos:

```
[root@gc1-ce ~]# pwd
/root
[root@gc1-ce ~]# su - condor
[condor@gc1-ce ~]$ pwd
/home/condor
[condor@gc1-ce ~]$ chmod 777 test2.submit
[condor@gc1-ce ~]$ cd test2/
[condor@gc1-ce test1]$ condor_submit ../test2.submit
```

NOTA: cambiar el parámetro *Queue* con el número de *working nodes* en el Clúster.

Prueba 3 (Escala la prueba anterior lanzando N numero de procesos)

Archivo test3.submit

```
Executable = test
Universe = vanilla
output = logTest3.$(Process)
error = logTest3.$(Process)
Arguments = 1234567 3 123
Log = logTest3.log
should_transfer_files = YES
when_to_transfer_output = ON_EXIT
Queue 10000
```

Para ejecutar la prueba seguir los siguientes pasos:

```
[root@gc1-ce ~]# pwd
/root
[root@gc1-ce ~]# su - condor
[condor@gc1-ce ~]$ pwd
/home/condor
[condor@gc1-ce ~]$ chmod 777 test3.submit
[condor@gc1-ce ~]$ cd test3/
[condor@gc1-ce test1]$ condor_submit ../test3.submit
```

NOTA: cambiar el parámetro Queue con el número de veces que se quiere lanzar la prueba en el clúster (aquí pueden poner un número bastante grande con el fin de probar el clúster)

Autor

Diego Alberto Rincón Yáñez

diego-rincon@javeriana.edu.co

Departamento de Ingeniería de Sistemas y Computación

Grid Colombia - Pontificia Universidad Javeriana

Bogotá DC. - Colombia

Marzo 2011 – Versión 2.0