

# Pruebas Cóndor – Universo Java

---

## 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 Cóndor instalado previamente.

## Requisitos

Para que las pruebas corran exitosamente se necesita contar con la instalación correcta del aplicativo **Cóndor**

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 instalación de Cóndor. La salida sería 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.

## Definición de Comandos de Cóndor

En la ejecución de las pruebas se usaran 3 comandos básicos de Cóndor el cual se expondrán a continuación.

***condor\_status***: Obtiene la información de las maquinas pertenecientes al clúster y datos adicionales como carga de procesador y *uptime*.

***condor\_q***: Obtiene la cola de trabajos que tiene el Cóndor programados.

***condor\_submit***: Mediante un archivo de configuración procede a ejecutar un trabajo en dicho clúster.

## Listado de Pruebas

El paquete cuenta con Varias pruebas previamente probadas

**Test1**: Prueba el clúster a nivel básico para corroborar que la maquina virtual esta correctamente instalado con un Hola mundo Sencillo.

**Test2**: Prueba una aplicación sencilla desarrollada en lenguaje Java en las maquinas del clúster y fácilmente parametrizable.

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

## Estructura de Directorios

Los Directorios de las pruebas están definidos de esta manera.

- / (Directorio Raiz)
- /test1 (Resultados de la Prueba 1)
- /test2 (Resultados de la Prueba 2)
- /test3 (Resultados de la Prueba 3)
- Hello.class (Aplicativo compilado para realizar prueba 1)
- Principal.class (Aplicativo compilado para realizar prueba 2 y 3)
- /test1/test1.submit (Archivo de configuración para prueba 1)
- /test2/test2.submit (Archivo de configuración para prueba 2)
- /test3/test3.submit (Archivo de configuración para prueba 3).

Esta estructura se debe crear en el *home* del usuario Cóndor creado previamente según las instrucciones del manual.

El aplicativo puede recibir 3 parámetros los cuales son:

**Limites:** Pone un límite tanto superior como inferior para buscar los divisores del primer número digitado. Ej.:/ Principal 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 basico)

#### Archivo test1.submit

```
universe = java
executable = Hello.class
arguments = Hello
output = Hello.$(Process)
error = Hello.error
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 ~]$
[condor@gc1-ce ~]$ cd /home/condor/pruebasJava
[condor@gc1-ce pruebasJava]$
[condor@gc1-ce pruebasJava]$ cd test1/
[condor@gc1-ce pruebasJava]$ chmod 777 test1.submit
```

```
[condor@gc1-ce pruebasJava]$ 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 realizar esta prueba es necesario compilar el programa Principal.java con el compilador por defecto instalado en la máquina

### Archivo test2.submit

```
universe = java
executable = Principal.class
arguments = Principal 999999999
output = logTest2.$(Process)
error = logTest2.error
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 ~]$ cd /home/condor/pruebasJava
[condor@gc1-ce pruebasJava]$
[condor@gc1-ce pruebasJava]$ chown -R condor:condor test2/
[condor@gc1-ce pruebasJava]$ cd test2/
[condor@gc1-ce pruebasJava]$ chmod 777 test2.submit
[condor@gc1-ce pruebasJava]$ 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

```
universe = java
executable = Principal.class
arguments = Principal 999999999999
output = logTest3.$(Process)
error = logTest3.error
should_transfer_files = YES
when_to_transfer_output = ON_EXIT
queue 100
```

Para ejecutar la prueba seguir los siguientes pasos:

```
[root@gc1-ce ~]# pwd
```

```
/root
[root@gc1-ce ~]# su - condor
[condor@gc1-ce ~]$ cd /home/condor/pruebasJava
[condor@gc1-ce pruebasJava]$
[condor@gc1-ce pruebasJava]$ chown -R condor:condor test3/
[condor@gc1-ce pruebasJava]$ cd test3/
[condor@gc1-ce pruebasJava]$ chmod 777 test3.submit
[condor@gc1-ce pruebasJava]$ 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](mailto: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