

Ingeniería en Sistemas de Información

ReduceMapFast

Documento de pruebas



Cátedra de Sistemas Operativos

Trabajo práctico Cuatrimestral

- 1C2015 -
Versión [1.0]

Requisitos y notas de la evaluación

Deploy y Setup

- Es condición necesaria para la evaluación que **el Deploy & Setup del trabajo se realice en menos de 10 minutos**. Pasado este tiempo el grupo perderá el derecho a la evaluación.
- Los archivos de configuración requeridos para los diversos escenarios de pruebas deberán ser preparados por el grupo con anticipación dejando sólo los parámetros desconocidos (ej: IP) incompletos.
- En la fecha de entrega la conexión a Internet podría estar congestionada para clonar el repositorio desde GitHub. Debido a eso **el grupo debe traer una copia del trabajo en un medio extraíble**, subirlo a una máquina virtual y luego copiar dicho repositorio por red entre las VMs. Ver [Anexo - Comandos Útiles](#)

Compilación y ejecución

- La compilación debe hacerse en la máquina virtual de la cátedra en su edición Server (no se pueden usar binarios subidos al repositorio).
- Para facilitar la visualización de varias terminales de manera simultánea se utilizará la herramienta **PuTTY** para acceder a las consolas de las Máquinas Virtuales.
- Es responsabilidad del grupo verificar que los parámetros de compilación sean portables y conocer y manejar las herramientas de compilación desde la línea de comandos. Ver [Anexo - Comandos Útiles](#)
- Debido a la complejidad y la concurrencia de los eventos que se van a evaluar es imprescindible que el alumno verifique que **su registro (log) permita determinar en todo momento el estado actual y anterior del sistema** y sus cambios significativos.

Evaluación

- Cada grupo deberá llevar **dos** copias impresas de la [planilla de evaluación](#)¹ con los datos de los integrantes completos (dejando los campos “Nota” y “Coloquio” en blanco) y una copia de los presentes tests.
- Las pruebas pueden ser alteradas o modificadas entre instancias de entrega y recuperatorios, y podrán ser adaptadas durante el transcurso de la corrección a criterio del ayudante para lograr validar el correcto funcionamiento y desempeño del sistema desarrollado.
- En los casos en que las modificaciones se vuelvan permanentes, el documento será actualizado y re-publicado para reflejar estos cambios.

¹ Al final de este documento

Pruebas

Prueba 1 - Condición mínima

Esta prueba comprueba el estado determinado como mínimo para que un trabajo práctico sea evaluado. **Cada equipo deberá corroborar que su trabajo cumple con las pruebas aquí descritas antes de inscribirse a una fecha de evaluación**

Configuración inicial

Se requieren 5 máquinas virtuales para ejecutar este test.

VM1	VM2	VM3	VM4	VM5
FileSystem Job1	Marta Nodo1	Nodo2 Job2	Nodo3 Job3	Nodo4

Nodos

Los Nodos deben tener el siguiente esquema de espacio de datos:

Nodo1	Nodo2	Nodo3	Nodo4
1GB	860MB	2.4GB	1.6GB

Job1 - weather-mr

Este job va a ejecutar <https://github.com/sisoputnfrba/weather-mr> con combiner sobre los archivos de temperatura disponibles en [aquí](#)² para obtener el horario de la máxima temperatura del día en cada estación climatológica (WBAN)

Variable	Valor
Mapper	mapper.sh
Reducer	reduce.sh

² El día de la entrega estarán disponibles en la máquina virtual donde sean evaluados.

Combiner	Si
Archivos	[/mr/weather/201301hourly.txt, /mr/weather/201302hourly.txt, /mr/weather/201303hourly.txt, /mr/weather/201304hourly.txt]
Resultado	/output/job1/max-temps.txt

Job2 y Job3 - mr-py-WordCount

Este job se va a ejecutar dos veces una vez **con** combiner (Job2) y otra sin combiner (Job3) sobre algunos archivos de texto disponibles [aquí](#) para obtener la cantidad de veces que se repite cada palabra del alfabeto. Los scripts se encuentran en [este](#) repositorio

Variable	Valor
Mapper	map.py
Reducer	reduce.py
Combiner	Si (Job2) / No (Job3)
Archivos	[/mr/textos/gutenberg.txt, /mr/textos/linux.txt]
Resultado	/output/job2y3/textos-comb.txt (Job2) /output/job2y3/textos-nocomb.txt (Job3)

Desarrollo

- Iniciar el FileSystem, conectar los correspondientes Nodos
- Formatear el FileSystem MDFS desde la consola.
- Crear los directorios /mr, /mr/textos, /mr/weather, /output/job1 y /output/job2y3
- Copiar los archivos de los tests a los directorios correspondientes de MDFS.
- Visualizar por la consola del Filesystem la correcta asignación y distribución de bloques en los diversos Nodos
- Iniciar Marta y luego el Job1. Mientras esté en ejecución el Job1 iniciar Job2 y Job3
- Durante la ejecución de los Jobs, desconectar un Nodo
- Volver a conectar el Nodo desconectado.
- Validar la igualdad de resultado de Job2 y Job3

Prueba 2

Se utiliza la misma configuración que la Prueba 1

VM1	VM2	VM3	VM4	VM5
FileSystem	Marta Nodo1	Nodo2	Nodo3 Job5	Nodo4 Job4

Job4 - mr-c-letterCount

Este job va a ejecutar **con** combiner sobre algunos archivos de texto disponibles [aquí](#) para obtener la cantidad de veces que se repite cada letra del alfabeto. El código y los binarios del mapper/reducer se encuentran en [este](#) repositorio

Variable	Valor
Mapper	mapper
Reducer	reducer
Combiner	Si
Archivos	[/mr/textos/gutenberg.txt, /mr/textos/linux.txt, /mr/textos/kernel.txt]
Resultado	/output/job4/rep-letras.txt

Job5 - twitter-sentiment

Este job va a ser ejecutado **sin** combiner sobre el archivo de tweets tecnológicos. El mapper va a extraer el estado de ánimo de los tweets y el reduce va a obtener aquel que tenga mayor valor positivo agrupado por ubicación y día. Los scripts se encuentran en [este](#) repositorio

Variable	Valor
Mapper	basic_sentiment_analysis.py
Reducer	top-sent.pl
Combiner	No

Archivos	[/sentiment/tweets.csv]
Resultado	/output/job5/tweets-sent.csv

Desarrollo

- Crear los directorios /sentiment, /output/job4 y /output/job5
- Copiar tweets.csv al directorio /sentiment de MDFS.
- Visualizar por la consola del Filesystem la correcta asignación y distribución de bloques en los diversos Nodos
- Iniciar los dos Jobs
- Validar la correcta ejecución y resultados.






Planilla de Evaluación - TP1C2015




Grupo:

Legajo	Nombre y Apellido	Nota

Evaluador:

Coloquio:

Condiciones Mínimas	
Existen conexiones TCP entre los diversos procesos (netstat -nap)	
El FileSystem MDFS soporta correctamente los archivos y directorios	
Al copiar un archivo al FileSystem MDFS este no es alterado (validar mediante md5)	
La distribución de bloques en los Nodos es correcta	
Los Jobs son planificadas por MARTA de manera simultánea respetando la dependencia entre las operaciones de un Job	
Los mappers y los reduce de un Job se ejecutan de manera simultánea en hilos independientes	
Los Nodos ejecutan operaciones de manera completamente simultánea	
Los Nodos pueden ingresar y salir del sistema sin alterar el funcionamiento de los Jobs	
La política de Combiner se aplica correctamente	
El resultado de los Jobs es el correcto	
El resultado de un job con soporte de combiner al ser ejecutado sin combiner es idéntico	

FileSystem MDFS	
Al eliminar un archivo los bloques se liberan	
Al no haber un bloque disponible en tres nodos distintos la operación de copia es abortada	
El Filesystem al iniciar recupera las estructuras persistidas	

Nodo	
Un nuevo nodo puede ingresar al sistema sin alterar el funcionamiento	
Un nodo viejo puede re-ingresar al sistema y formar parte del cluster de procesamiento	

Anexo - Comandos Útiles

Copiar un directorio completo por red

```
scp -rpC [directorio] [ip]:[directorio]
```

Ejemplo:

```
scp -rpC tp-1c2015-repo 192.168.3.129:/home/utnso
```

Descargar **solo** la última versión del código (en vez de todo el repositorio)

```
curl -u '[usuario]' -L -o [archivo] [url_repo]
```

Ejemplo:

```
curl -u 'gastonprieto' -L -o commons.tar  
https://api.github.com/repos/sisoputnfrba/so-commons-library/tarball/master
```

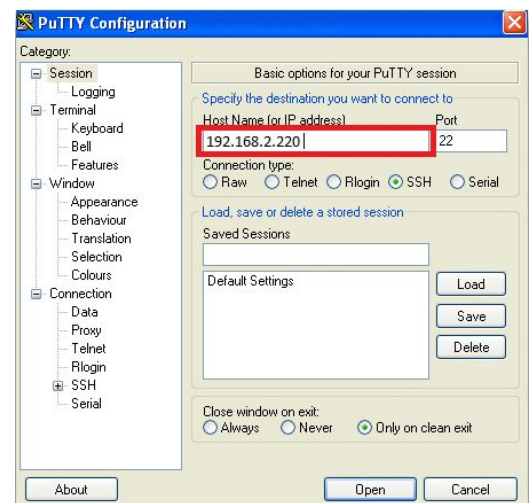
Este comando debe ejecutarse sin salto de línea. Luego **descomprimir con**: `tar -xvf commons.tar`

PuTTY

Este famoso utilitario nos permite desde Windows acceder de manera simultánea a varias terminales de la Máquina Virtual, similar a abrir varias terminales en el entorno gráfico de Ubuntu.

Ya se encuentra en las computadoras del laboratorio y se puede descargar desde [aquí](#)

Al iniciar debemos ingresar la IP de nuestra máquina virtual en el campo **Host Name (or IP address)** y luego presionar el botón **Open** y loguearnos como **utnso**



Se recomienda investigar:

- Directorios y archivos: `cd`, `ls`, `mv`, `rm`, `ln` (creación de symlinks)
- Entorno: `export`, variable de entorno `LD_LIBRARY_PATH`
- Compilación: `make`, `gcc`, `makefile`