Sistemas Operativos

FIEC, ESPOL, Dra. Cristina Abad R.

Proyecto Final

Resumen del proyecto

Título: Evaluación de rendimiento de 4 programas con BashReduce

Estudiantes: 1 a 4 por proyecto

Puntaje: 30 puntos (30% de la nota del parcial)

Fecha de entrega: Miércoles 27 de enero de 2016, 15h00

Fecha de sustentación: Miércoles 27 de enero de 2016, 15h30

Lenguajes de programación permitidos:
bash, awk, python

Sistemas operativos permitidos:
Linux

Interfaz gráfica: Ninguna

Descripción y alcance del proyecto

Antecedentes

En el 2004, Google publicó el diseño de su plataforma de procesamiento distribuida "MapReduce" [1], la cual revolucionó las áreas de Sistemas Distribuidos, Computación en la Nube y Big Data.

En el 2009, Erik Frey (en ese entonces, desarrollador de last.fm), desarrolló BashReduce [2,3], una versión de MapReduce para bash, la cual permite ejecutar programas, distribuyéndo la carga entre varios cores de un mismo computador, o entre varios computadores.

Descripción del problema

En este proyecto, usted evaluará las mejoras al rendimiento de 4 programas/procesos que se obtienen al ejecutar dichos procesos en múltiples cores y en múltiples computadores, versus el ejecutar el proceso en un solo core de un solo computador.

Alcance

Ud. deberá diseñar, ejecutar y documentar experimentos que le permitan evaluar el rendimiento de 2 a 4 procesos diferentes (uno por cada miembro del grupo, con un mínimo de 2 procesos para quienes trabajen de manera individual). El resultado final será el documento con (al menos) un gráfico para cada una de las 4 evaluaciones. Cada gráfico deberá contar con (al menos) 4 curvas de rendimiento, una para el proceso en un solo core, otra para el proceso en 2 cores, otra para el proceso en la cantidad de cores de la máquina, y otra para el proceso corriendo en múltiples PCs (y posiblemente en múltiples cores en cada una).

Sus experimentos deben estar diseñados de tal manera que permitan apreciar las mejoras al rendimiento en varios escenarios, por ejemplo:

- Procesos CPU-bound
- Procesos IO-bound que trabajan con un archivo de entrada pequeño
- Procesos IO-bound que trabajan con un archivo de entrada grande
- Procesos IO-bound que trabajan con múltiples archivos de entrada

Cada proceso a ser evaluado puede ser un programa en bash, awk, python, o una serie de programas concatenados con pipes.

El documento deberá documentar los resultados, incluyendo un análisis que permita entender la diferencias en los diferentes resultados. Por ejemplo: Si un proceso permite tener mejoras considerables con bashreduce mientras que otro no mejora o hasta empeora. ¿Por qué? Debe incluir este análisis en una sección en el documento.

A más de documento, también revisaré el/los script(s) que hayan usado para correr los experimentos, los cuáles los deberán haber subido a un repositorio en GitHub. Yo ejecutaré dichos scripts para confirmar que puedo reproducir los resultados mostrados en el documento.

IMPORTANTE: Cada estudiante será responsable de uno de los 4 experimentos, con su correspondiente resultados y documentación. La nota de cada estudiante en el grupo será diferente a la de sus compañeros y dependerá de cuán bien está diseñado/ejecutado/evaluado dicho experimento.

Entregables

1. Documento **impreso**, a ser entregado durante la sustentación del proyecto. Secciones del documento: Introducción, metodología/diseño (de los experimentos), resultados, conclusiones.
2. Código fuente en GitHub; entregar URL vía SidWeb.

Rúbrica

Total: 30 puntos

1. Documento: 22 puntos
    a. Sección de metodología/diseño describe y justifica correctamente cada uno de los experimentos: 4 puntos
    b. Experimentos: 8 puntos
        i. Para cada proceso/programa se ha dizeñado y ejecutado experimentos que permite evaluar su rendimiento con bashreduce en múltiples cores: 4 puntos
        ii. Para cada proceso/programa se ha dizeñado y ejecutado experimentos que permite evaluar su rendimiento con bashreduce en múltiples PCs: 4 puntos
    c. Resultados: gráfico correctamente presentado (total: 6 puntos)
        i. Suficientes datapoints: 2 puntos
        ii. Rango de ejes de las X y de las Y bien seleccionados: 2 puntos
        iii. Gráfico permite realizar una comparación de las mejoras de rendimiento en el proceso: 2 puntos
    d. Conclusiones/Análisis: 4 puntos
        i. Conclusiones/Análisis explican de manera satisfactoria lo que se puede observar en los gráficos y proporciona hipótesis y/o pruebas que permiten justificar los resultados obtenidos.
2. Implementación: 8 puntos
    a. README explica cómo correr los scripts en GitHub: 2 puntos
    b. Hay al menos 3 commits por cada estudiante en GitHub: 2 puntos
    c. Scripts en GitHub se pueden ejecutar y permiten obtener los resultados incluidos en el documento: 4 puntos
3. Sustentación: 10 puntos
    a. Estudiante demuestra haber participado activamente en el desarrollo del proyecto y demuestra entender su implementación y resultados: 10 puntos
    b. Estudiante demuestra haber participado parcialmente el desarrollo del proyecto y demuestra entender parcialmente su implementación y resultados: 7.5 puntos
    c. Estudiante demuestra participación limitada en el proyecto y/o no logra comprender los resultados obtenidos: 5 puntos

    d. Estudiante demuestra una colaboración mínima en el desarrollo del proyecto: 2.5 puntos
    e. Estudiante no colaboró en el desarrollo del proyecto: 0 puntos
4. Nota Final: (Nota documento + nota implementación + nota de rendimiento) * (Nota sustentación)/10

Referencias

[1] Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: simplified data processing on large clusters. ACM Symposium on Operating System Design and Implementation (OSDI), 2004. http://research.google.com/archive/mapreduce.html

[2] Erik Frey, BashReduce: MapReduce in bash. Disponible en: https://github.com/erikfrey/bashreduce

[3] Jeremy Zawdony, bashreduce: A Bare-Bones MapReduce. LINUX Magazine. July 2009. Disponible en: http://www.linux-mag.com/id/7407/