



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC3413 — Implementación de Sistemas de Base de Datos

LABORATORIO 6

Laboratorio: Bases de datos paralelas y distribuidas.
Publicación: Lunes 26 de mayo.
Ayudantía: Viernes 30 de mayo.
Entrega: **Domingo 8 de junio a las 23:59 horas.**

Introducción

En este laboratorio estudiaremos el almacenamiento y la ejecución de consultas simples en el sistema Hadoop¹.

Introducción a Hadoop

Apache Hadoop es un "framework" constituido por un conjunto de librerías Java para almacenar y procesar grandes cantidades de datos en un entorno formado por clusters de computadores. Hadoop se constituye de cuatro elementos principales:

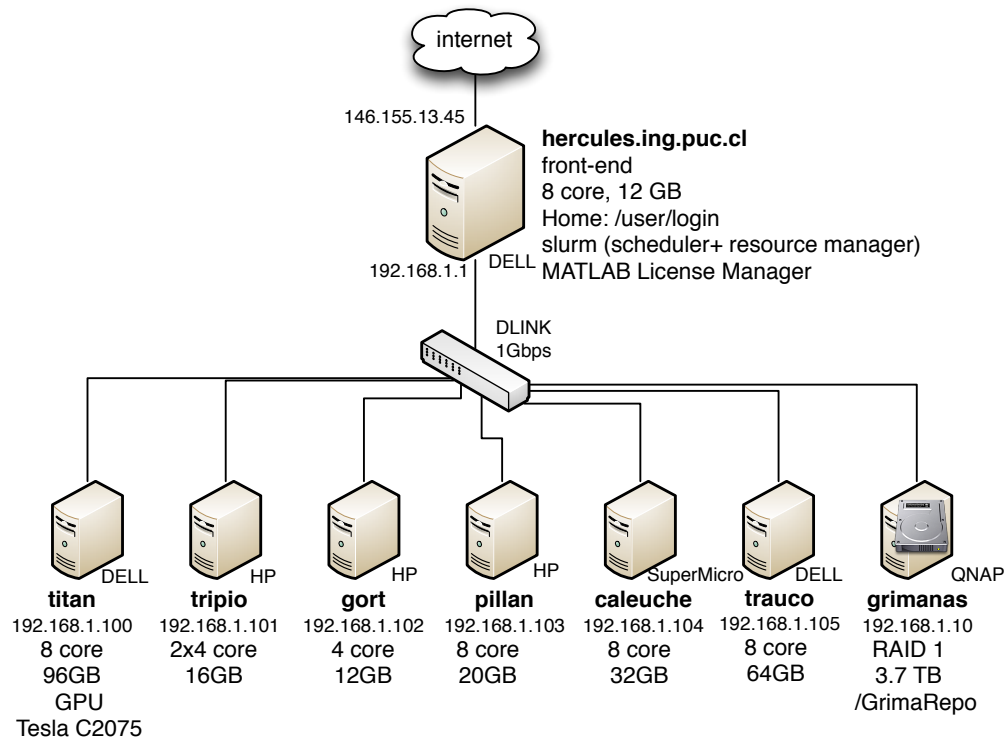
- Hadoop Common: conjunto de utilidades que dan soporte a los módulos de Hadoop.
- Hadoop Distributed File System (HDFSTM): El sistema de archivos distribuidos de Hadoop, inspirado en el Google File System (GFS) [GGL03].
- Hadoop YARN: Un framework para la gestión de tareas (scheduling) y gestión de los recursos del cluster.
- Hadoop MapReduce: sistema para el procesamiento de forma paralela de grandes cantidades de datos (basado en YARN). Derivado del artículo de Google Map Reduce [DG08].

Además, Hadoop tiene proyectos asociados como Apache Mahout (implementa algoritmos de machine learning), Apache HBase (una base de datos distribuida utilizando el concepto de BigTable [CDG+06]), Pig (entorno de programación de alto nivel para codificar programas MapReduce) entre otros.

MapReduce y Hadoop

MapReduce es un modelo de programación orientado a aquellas aplicaciones que permiten el procesamiento en paralelo de grandes cantidades de datos. Esto es debido a que en este modelo cada subtarea se puede procesar de forma independiente por distintos procesos workers, los cuales se ejecutan en distintos nodos distribuidos en un cluster de computadores. MapReduce está inspirado en las primitivas map y reduce del ámbito de la programación funcional. En una aplicación MapReduce, un conjunto de procesos *mapper* recibe un fragmento del conjunto de datos de entrada y los procesa en paralelo de manera independiente. El resultado es un conjunto de valores intermedio, que son utilizados por otro conjunto de procesos *reducer* para obtener el resultado final.

¹<http://hadoop.apache.org>



Cluster GRIMA

Como se ha comentado antes, Apache Hadoop es una implementación open source del modelo MapReduce. Para obtener una mejor idea de cómo funciona Hadoop y el modelo MapReduce se recomienda la lectura de la documentación del sistema², el tutorial³ y la implementación del ejemplo más típico de Hadoop: WordCount⁴. Para esto tan solo es necesario descargar Hadoop y realizar una instalación local.

Para este laboratorio se tiene acceso al cluster de GRIMA en el cual está disponible una instalación de Hadoop. Este cluster consiste en varias máquinas y la instalación de Hadoop se encuentra disponible entre las máquinas Hercules, Titan y Tripio. La Figura muestra la organización del cluster.

Tareas

Tarea 1. Joins y intersección en MapReduce. Para esta tarea, el alumno deberá implementar el algoritmo en MapReduce visto en clases para computar:

1. el join natural $R(A, B) \bowtie S(A, C)$ entre las relaciones $R(A : \text{INT}, B : \text{String})$ y $S(A : \text{INT}, C : \text{String})$.
2. la intersección $R(A) \cap S(B)$ entre las relaciones $R(A : \text{String})$ y $S(B : \text{String})$.

El input y output de la operación deben ser extraídos y almacenado en el *Distributed File System* (DFS) de Hadoop.

Tarea 2. Evaluación de performance de algoritmo MapReduce En esta tarea el alumno deberá evaluar la eficiencia de los algoritmos implementados en la Tarea 1 y graficar el comportamiento de la solución en un gráfico “número de tuplas” versus tiempo. Cada gráfico deberá considerar distintos:

²<http://hadoop.apache.org/docs/stable/index.html>

³http://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html

⁴http://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html#Example%3A+WordCount+v1.0

1. Tamaños de tuplas. Se deberá considerar 4 tamaño de tuplas distintas.
2. Distribuciones de las tuplas. Para esto se deberá considerar 4 distribuciones distintas: todos los valores distintos (join uno-a-uno), todos los valores iguales (producto cruz), selectividad del 0.1 % y selectividad del $\frac{N}{1000}$ % donde N es el número de tuplas.

En total son 16 gráficos distintos por cada operador.

La evaluación de cada caso deberá desarrollarse en el cluster de GRIMA para el cual se proveerá acceso a cada alumno. Debido a que el tamaño de las relaciones es considerable (del orden de GB), los datos para la evaluación de sistema serán comunes y proporcionados por el ayudante del curso. Más detalles sobre la evaluación y las distintas pruebas se entregarán durante la ayudantía del Laboratorio.

Para la entrega de esta tarea, el alumno debe subir a GitHub un documento con los gráficos obtenidos en esta experiencia junto con un pequeño análisis de la experiencia (no más de dos páginas).

Evaluación

Para la evaluación del laboratorio, la ponderación de cada tarea en la nota final del laboratorio será la siguiente:

- Tarea 1: 50 %.
- Tarea 2: 50 %.

Ayudantía y preguntas

El día viernes 30 de mayo se realizará una ayudantía donde se darán mas detalles sobre hadoop y el laboratorio. **Para preguntas fuera del laboratorio se pide usar el foro del curso (google group IIC3413).** No se responderán e-mails privados con dudas.

Referencias

- [CDG⁺06] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber. Bigtable: A distributed storage system for structured data. In *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation - Volume 7*, OSDI '06, pages 15–15, Berkeley, CA, USA, 2006. USENIX Association.
- [DG08] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008.
- [GGL03] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The google file system. *SIGOPS Oper. Syst. Rev.*, 37(5):29–43, October 2003.