TP3 - Algoritmos en sistemas distribuidos

Sistemas Operativos - Segundo Cuatrimestre de 2014

Límite de entrega: Martes 25 de noviembre a las 23:59

Introducción

Los directivos de Rededit han lanzado un concurso de análisis de datos para entender el funcionamiento de la comunidad. Estos tiene un gran problema, generan gran cantidad de datos diariamente que no saben cómo procesar.

Se desea evaluar una solución sobre el exitosísimo paradigma *Map-Reduce* y para esto, se nos facilitaron los datos necesarios para realizar una prueba de concepto.

El siguiente trabajo se dividirá en dos secciones, una donde se implementarán pequeños algorítmos de análisis de datos y otra sección donde se hará un análisis de escalabilidad y performance de la arquitectura.

Datos

Rededit es un sitio de noticias y entretenimiento donde los usuarios registrados suben contenido en forma de vínculos o texto. Los usuarios votan positivamente (*upvote*) o negativamente (*downvote*), lo que genera un *ranking* de contenido. A su vez, las entradas de contenido se organizan en áreas de interés llamadas *subreddits*.

Los datos que obtuvimos son algunas entradas de *Rededit*. Cada entrada posee los siguientes campos:

- `image_id`: Id de la imágen. Entradas con el mismo ID corresponden a la misma imágen.
- `unixtime`: Timestamp (UNIX) del momento de la creación.
- `rawtime`: Timestamp (ISO) del momento de la creación.
- `title`: Título.
- `total_votes`: Número de *upvotes* + *downvotes*.
- `reddit_id`: Id de la entrada en *Rededit*.
- `number_of_upvotes`: Número de *upvotes*.
- `subreddit`: subreddit.
- `number_of_downvotes`: Número de *downvotes*.
- `localtime`: Timestamp local (UNIX) del momento de la creación.
- `score`: Número de *upvotes* − *downvotes*.
- `number_of_comments`: Número de comentarios recibidos.
- `username`: Nombre del usuario que publicó el contenido.

Implementación Map-Reduce

El lote de datos se encuentra almacenado en db.reddit, para acceder desde la terminal:

```
$ mongo
> use reddit
```

En la colección *posts* se encuentran la información como un Json que tiene la estructura definida previamente:

Para ver el primer elemento en lote de datos deberán hacer:

```
> db.posts.find()[0]
{
        "_id" : ObjectId("5287f32503e912097aeee725"),
        "username" : "jaymzwilson",
        "rawtime" : "2012-06-11T22:31:01-07:00",
        "score" : 54,
        "title" : "Searched Rape Prevention on Google and this showed up. Honestly, WTF.",
        "number_of_comments" : 5,
        "unixtime" : "1339428661",
        "subreddit" : "WTF",
        "image_id" : "17008",
        "number_of_upvotes" : 77,
        "localtime" : "1339453861",
        "number_of_downvotes" : 23,
        "reddit_id" : "uxhvz",
        "total_votes" : 100
}
```

Utilizando estos datos deberán realizar los siguientes análisis:

1. encontrar el subreddit con mayor score promedio.
2. Encontrar los doce títulos con mayor score de la colección de posts con al menos 2000 votos.
3. Para los diez mejores scores, calcular la cantidad de comentarios en promedio por sumisión.
4. Entre los usuarios con a la sumo 5 sumisiones, encontrar el que posea mayor cantidad de upvotes.
5. Para todos los subrredit que poseen un score ente 280 y 300, indicar la cantidad palabras presentes en sus títulos

Para implementar cada uno de los análisis, deberán crear la función *Map* en un archivo *map.js* y la función *Reduce* en un archivo *reduce.js* y de ser necesario la función *Finalize* en un archivo *finalize.js*. Como su extensión lo indica, estas funciones deben contener código Javascript. Para ejecutar los análisis cuentan con el script de Python *runner.py*.

```
$ python runner.py
```

Para mas información, consultar el código fuente y:
Map-Reduce en Mongo: http://docs.mongodb.org/manual/core/map-reduce/
JavaScript: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide.

Investigación

Se pide lee el siguiente paper : *Job Scheduling for Multi-User MapReduce Clusters* (http://www.icsi.berkeley.edu/pubs/techreports/ICSI_jobschedulingfor09.pdf) y exponer los siguiente puntos.

1. Explique cuál es el entorno y la situación donde se plantea el problema. (Motivación)
2. Identifique situaciones donde el problema se dispara y la consecuencias que esto genera.
3. Identifique del problema
4. Comente el background histórico del problema

5. Explique otros problemas asociados a la búsqueda de una mejor solución
6. Explique las soluciones propuestas
7. Evalúe soluciones propuestas
8. Discusión
9. Conclusiones

Informe

Se pide que entreguen un informe completo con la explicación detallada y la justificación de las funciones que utilicen en cada uno de los ejercicio. Además deben presentar los resultados obtenidos en cada caso. Por último deben presentar una explicación del paper sugerido utilizando como guía los puntos marcados en la sección anterior.