TP3 - Algoritmos en sistemas distribuidos

Sistemas Operativos - Segundo Cuatrimestre de 2013

Límite de entrega: Domingo 8 de Diciembre 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. Un usuario se puede definir como *downvoter*, si vota mas veces negativamente que positivamente, *upvoter*, si realiza lo contrario o *neutral* si la cantidad de votos positivos y negativos son iguales. ¿Es la comunidad en promedio más *upvoter* o más *downvoter*?
2. Promedio de comentarios por submission.
3. El usuario con la suma(score) más alta de todos.
4. Para el **Jueves 21 de Noviembre** deberán diseñar un análisis y consultarlo en clase. Luego de ser aprobado por los docentes, deberán implementarlo.

Para implementar un análisis, deberán crear la función *Map* en un archivo `map.js` y la función *Reduce* en un archivo `reduce.js`. Para ejecutarlos, cuentan con el script de Python `runner.py`.

```
$ python runner.py
```

Para mas información, consultar el código fuente.

Análisis de escalabilidad

La compañía ha decido que el esquema planteado es el adecuado: La utilización de MongoDB para el soporte de los datos y una arquitectura basada en máquinas virtuales.

1. El *deploy* de las máquinas virtuales se propone lanzar en una plataforma de *HaaS* (Hardware as a Service). Para esto, se solicita una búsqueda bibliográfica de distintas alternativas de *HaaS* y su evaluación de performance teniendo en cuenta las capacidades de la solución elegida, MongoDB. Para esto, considere que el tamaño de los datos actuales no superará un crecimiento lineal anual.
2. Analice qué problemas podría traer esta solución si en un futuro la compañía decidiera cambiar la implementación *HaaS* por una de hardware propio.