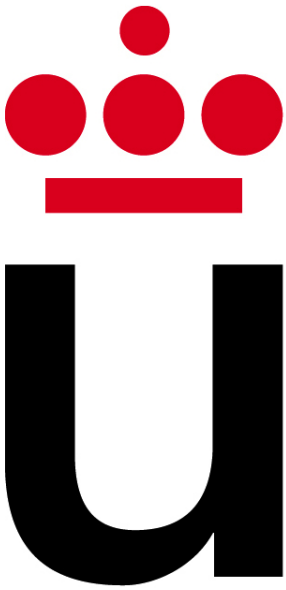
Foro de preguntas

Análisis de Big Data

Preguntas Tema 4

José Ignacio Escribano



Móstoles, 26 de marzo de 2016

Índice

1. Preguntas

1.1. Supongamos que cierto algoritmo de data mining se aplica sobre un gran conjunto de datos de 1PB de tamaño, almacenado en un sistema de ficheros distribuido en un clúster (por ejemplo, HDFS) y se va a analizar con un entorno que permite procesado de datos en memoria (por ejemplo, Spark). En cada iteración del algoritmo, se aplica una serie de pasos sobre un subconjunto de los datos, para luego, iterativamente, refinar el resultado en sucesivas pasadas muestreando de nuevo datos del conjunto original. ¿Sería necesario almacenar en este caso el total de 1PB de datos en memoria para agilizar los cálculos? ¿En qué principio nos podemos basar para argumentar nuestra respuesta?

No es necesario tener el total de datos en memoria, debido al principio de localidad. La naturaleza iterativa de estos algoritmos hace que accedamos a los mismos datos una y otra vez (localidad espacial). Además, cuando un dato no esté en un nivel de memoria superior, se bajará a uno inferior, subiendo el dato junto con sus datos adyacentes (localidad espacial), haciendo que se produzcan menos fallos en futuras llamadas.

1.2. Una alternativa al procesado de datos en memoria cuando no podemos almacenar toda la información en la RAM del sistema son las bases de datos de consultas aproximadas. BlinkDB, un proyecto de la UC Berkeley sigue esta estrategia. Busque información sobre BlinkDB y responda a estas preguntas: ¿Podemos acotar el error cometido en las consultas a este tipo de bases de datos? ¿Qué utilidad tienen respecto a las bases de datos tradicionales que siempre devuelven respuestas exactas a las consultas?¿Podemos acotar el tiempo de espera para recibir la respuesta a la consulta?

BlinkDB es un motor de consultas SQL con errores y tiempos de respuesta acotados sobre grandes volúmenes de datos.

La idea detrás de BlinkDB es que para tomar decisiones correctas no es necesario tener respuestas perfectas. Usando esta idea, BlinkDB permite intercambiar precisión por tiempo de respuesta.

BlinkDB es adecuada en aplicaciones donde prima la velocidad sobre la precisión. Algunos de estos escenarios son:

- Informes en tiempo real: es necesario tomar decisiones en muy poco tiempo, y es aceptable asumir un margen de error (por ejemplo, 5 %).
- Machine Learning, por ejemplo en sistemas de recomendación.
- En aplicaciones con colas largas, como por ejemplo, un ranking con valoraciones de distintos cantantes
- Etc.

BlinkDB consigue esto gracias a dos ideas clave:

1. Un framework que construye y mantiene un conjunto de muestras multidimensional de los datos originales a lo largo del tiempo.
2. Una estrategia dinámica de selección de las muestras que selecciona el tamaño muestral apropiado basándose en la precisión de las consultas o en los requisitos del tiempo de respuesta.

Por ejemplo, si queremos conocer la media del tiempo de sesión de los usuarios de la ciudad de San Francisco, y queremos una respuesta en menos de 1 segundo, haríamos una consulta de la siguiente forma:

```
SELECT avg(sessionTime)
FROM Table
WHERE city = "San Francisco"
WITHIN 1 SECONDS
```

Esta consulta devolvería algo como

$$234 \pm 15.32$$

Es decir, BlinkDB nos calcula un intervalo de confianza de la consulta que solicitemos.

Si, por el contrario, quisiéramos especificar el error o el nivel de confianza del intervalo, la consulta sería algo así:

```
SELECT avg(sessionTime)
FROM Table
WHERE city = "San Francisco"
ERROR 0.1 CONFIDENCE 95.0%
```

Fuentes:

[1]: https://www.cs.berkeley.edu/~sameerag/blinkdb_eurosys13.pdf
[2]: http://arxiv.org/pdf/1203.5485v2.pdf
[3]: https://www.quora.com/What-are-some-successful-usecases-for-Berkel
[4]: http://es.slideshare.net/Hadoop_Summit/t-1205p212agarwalv2
[5]: http://nwds.cs.washington.edu/files/nwds/pdf/UW-Google-published.p
[6]: http://telruptive.com/2013/04/06/a-big-data-base-that-is-fast-but-