Práctica Hadoop Sergio Yunta Martín

Módulo 4 datahack edición 63 Máster Big Data and Architecture

# 2. Dimensionamiento clúster Hadoop

El volumen de datos que vamos a recibir son:

|  | Media Eventos | Tamaño por evento |
| --- | --- | --- |
| Fuente 1 | 10.000 eventos/día | 15 KB |
| Fuente 2 | 120.000 eventos/día | 300 Bytes |
|  | 150.000 eventos/día | 100 KB |
|  | 170.000 eventos/día | 800 KB |
|  | 2000 eventos/día | 1500 KB |

Para calcular el tamaño que necesitamos para almacenar los datos de un año tenemos que tener en cuenta dos cosas:

* La replicación por defecto de Hadoop para garantizar que no se pierda información en el caso de que una máquina se caiga es de **3**. Es decir, necesitamos el **triple de tamaño** en el cluster del que necesitaríamos normalmente para almacenar los datos.
* El tamaño de bloque por defecto es de **128MB**, esto es importante porque define el **tamaño mínimo de escritura** (el de lectura también pero no aplica en este caso), es un factor a tener en cuenta.

Con esto en cuenta, podemos empezar con los cálculos, vamos a ver la tabla anterior calculando cuánto tamaño necesitaríamos para almacenar cada evento al día.

|  | **Tamaño diario (MB)** |
| --- | --- |
| Fuente 1 | 146,48 MB |
| Fuente 2 | 34,33 MB |
|  | 14.648,44 MB |
|  | 132.812,5 MB |
|  | 2.929,69 MB |

Resumiendo:

| **Total diario (MB)** | 150.571,44 MB |
| --- | --- |
| **Total Año (MB)** | 54.958.576,2 MB |
| **Total Año (TB)** | 52,41 TB |

Por lo tanto, cada año se generan **52,41 TB** de datos, pero siguiendo lo que hemos visto en clase y lo mencionado anteriormente, vamos a suponer una replicación típica de 3. Por lo tanto, al año necesitamos de almacenamiento, lo que serían redondeando **158 TB / 79 discos / 4 máquinas**.

Pero esto no es todo, estas 4 máquinas serían únicamente para almacenar todos los datos que vamos a recibir, es decir, *DataNodes*, pero para poder tener un clúster hadoop funcional, necesitamos al menos otra máquina que funcione como *NameNode* por lo que ya sumarían 5 máquinas*.*

Si ya que nos hemos visto forzados a tener 4 *DataNodes* queremos tener Alta Disponibilidad podemos añadir un segundo *NameNode* como *Secondary NameNode* y conseguir así Alta Disponibilidad.

En conclusión, necesitaríamos **6 máquinas** con alta disponibilidad. Si prescindimos de ella, el mínimo indispensable serían 5 máquinas.

# 3. Estimación arquitecturas Hadoop para distintos casos de uso

Repasemos qué herramientas del ecosistema Hadoop aplican a cada caso uno por uno.

## Herramienta de BI (p.ej.: Microstrategy).

Para este primer apartado deberíamos extraer los datos que queremos mostrar en los dashboards. Aquí depende mucho de los requisitos de latencia del dato.

* Si no tiene que ser un dashboard en tiempo real, es decir, no nos importa que los datos tarden un poco en actualizarse, podemos usar **Hive** para poder realizar las consultas y en base a los resultados de las mismas generar las gráficas necesarias.
* Por otro lado, si es muy importante que se refresque en tiempo real, tenemos otra opción, **Impala,** la pega con esta opción es que las consultas se realizan en memoria, se agilizan mucho pero se necesitan muchos más recursos.

Por tanto, para este caso en concreto, no creo que sea tan necesario ese consumo de recursos por lo que podríamos elegir una tecnología como **Hive**. [1] [3]

Microstrategy tenía un servicio llamado Hadoop Gateway que se deprecó en 2021, por lo que se ha descartado.

## Web de consultas sobre pedidos realizados.

En este caso el tiempo de consulta es más importante. Además, me parece muy buena oportunidad para, esperando un gran volumen de pedidos, modelarlos de tal manera que se puedan almacenar en una base de datos NoSQL. Si tiramos por esta opción, tenemos una buena opción en el ecosistema Hadoop como **HBase** [4]. Esto nos permitiría acceder a un gran volumen rápidamente tanto a nivel de lectura como escritura, permitiendo tiempos de respuesta razonables en la web.

## Generación de informes SQL usando R que se ejecutan mensualmente.

Para generar informes en SQL y R necesitamos algo que tenga opción de usar varios lenguajes de programación. Una opción que se presenta es **Spark,** puede ejecutar consultas SQL y R gracias a **Spark SQL y SparkR.** Esta tecnología se integra muy bien con hadoop, de hecho muchos lo consideran dentro del ecosistema [6] y además, el CEO va a estar muy contento porque es una tecnología que se usa muchísimo y estaríamos al día en lo que se usa en este ámbito. Esto nos permitiría sacar informes sobre muchos datos de manera rápida y ágil.

## Recopilación de información de redes sociales.

En este caso está claro que si necesitamos un sistema de recopilación de datos tenemos que buscar entre los servicios disponibles en el ámbito de ingesta de datos y ETLs. Una posible opción es usar **Flume** [7] que parece diseñado para conectar con fuentes remotas y trabajar en streaming, algo importante si hablamos de redes sociales, ya que se generan datos constantemente.

Según las diapositivas de teoría [3] había pensado en **Pig** pero no parece preparado para conectar con fuentes externas, o al menos no he conseguido encontrar mucha información al respecto.

Estas son mis decisiones para cada uno de los casos, he intentado que sean variadas e investigar más herramientas dentro del ecosistema hadoop.

# Referencias

[1] [Ecosistema hadoop | datahack](https://www.datahack.es/ecosistema-hadoop/)

[2] [¿Qué es HBase? | IBM](https://www.ibm.com/es-es/topics/hbase)

[3] Diapositivas de teoría

[4] [Spark SQL & DataFrames | Apache Spark](https://spark.apache.org/sql/)

[5] [SparkR (R on Spark) - Spark 3.5.5 Documentation](https://spark.apache.org/docs/latest/sparkr.html)

[6] [Apache Spark and Hadoop HDFS: Working Together](https://www.databricks.com/blog/2014/01/21/spark-and-hadoop.html)

[7] [Apache Flume](https://flume.apache.org/)