



UAB

Universitat Autònoma
de Barcelona

Antonio Espinosa

Apache Impala

Apache Impala

- Está diseñado para optimizar la latencia de las consultas, no tanto la escalabilidad y el ancho de banda
- Integrada en el ecosistema de Hadoop
- Permite el uso de lenguaje SQL para consultar los datos
- Los datos se leen del disco en una fase inicial y luego permanecen en la memoria del servidor durante buena parte del análisis de los datos



Requerimientos de Impala

- La mayor parte de los datos tiene que caber en memoria principal
- Requiere servidores con mucha más capacidad de memoria que los servidores Hadoop MapReduce
- Una consulta Impala no puede recuperarse de la caída de un nodo como en Hadoop. Si un servidor falla, la consulta generará un error
- Se recomienda usar Impala para consultas que se realizan de forma rápida. Reiniciar una consulta en caso de error no supone un grave inconveniente



Arquitectura de Impala

- Un proceso *daemon* en cada nodo del *cluster*
- *Statestore*: componente que gestiona la salud de los daemons
- *Catalog service*: gestiona los cambios de las consultas SQL a los nodos del cluster



Entorno de programacion

- Necesita procesos de servicio activos(daemons): los servicios de Impala siempre se están ejecutando.
- No hay coste de arranque pero necesitamos dedicar recursos del servidor
- El motor de consultas es propio y está realizado en C++
- Uso de LLVM: se utiliza LLVM para compilar las consultas y todas las funciones relacionadas para optimizarlas previamente a su ejecución



Ejemplo de uso: Join

```
select
    Nombre1, Nombre2
from
    Tabla1 JOIN Tabla2
Where
    Tabla1.id = Tabla2.id
```



Operación Broadcast join

- Operación join de dos conjuntos de datos:
- Impala toma el conjunto de datos menor y lo distribuye entre todos los servidores disponibles. Estos datos se almacenan en memoria principal en forma de una tabla hash
- Cada daemon de Impala lee su subconjunto local de datos y usa la tabla hash para encontrar aquellas filas que coinciden entre los dos conjuntos de datos
- Este proceso de lectura y búsqueda de coincidencias se realiza en bloques de 1GB



Join en tiempo de ejecución

```
select
    Nombre1, Nombre2
from
    TablaGigante JOIN TablaMini
Where
    TablaGigante.id = TablaMini.id
```



Filtrado en tiempo de ejecución

- Impala determina qué tabla tiene menos valores **ID** distintos: Gigante o mini
- Decide usar mini para construir una tabla hash en memoria de sus valores
- Lee bloques de tabla gigante de disco y evita comparar atributos de **IDs** no existentes



Cuándo usar Impala

- Impala debería ser una de las primeras opciones al evaluar motores de búsquedas NoSQL debido a su buen rendimiento
- Cuando necesitamos un sistema que tenga una alta capacidad de realizar muchas consultas de forma concurrente
- Útil en organizaciones donde existen muchos usuarios que lanzan consultas sobre un conjunto de datos distribuido en HDFS



A close-up of a laptop screen displaying the word "MOOC" in a bold, sans-serif font. The screen is framed by a thick black border. The background of the image is a solid light green color.

MOOC

Apache Impala

A close-up of a laptop keyboard with hands typing. The laptop screen displays the UAB logo and the text "MOOC Escola de Postgrau". The background of the image is a solid light green color.

UAB
Universitat Autònoma de Barcelona

MOOC
Escola de
Postgrau

UAB

Universitat Autònoma
de Barcelona