



Big Data

Análisis Interactivo de Datos

Texto base: Juan Esquivel Rodríguez
Luis Alexander Calvo Valverde



Motivación Análisis Interactivo

- Mucho esfuerzo es dedicado al diseño de arquitecturas de datos optimizadas para la extracción eficiente
 - Aun así es posible que las consultas tarden una cantidad de tiempo en dar respuesta superior a la que un usuario está dispuesto a soportar
- Una infraestructura para el análisis de datos interactivo es fundamental para satisfacer las necesidades
- En esta lección discutiremos dos tecnologías de Google Cloud cuyos orígenes se remontan a mediados de la década del 2000
 - Dremel
 - BigQuery



Dremel - Inicios

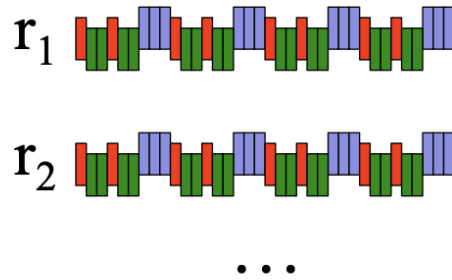
- A principios de los 2000, a lo interno de Google, cuando se quería hacer una consulta sobre un dataset era necesario escribir MapReduce
 - Incómodo cuando se quieren probar múltiples variantes de la consulta base
 - Idealmente podemos crear consultas SQL para ello
- Promesa de Dremel
 - Análisis interactivo de los datos en el orden de los segundos
 - Conjuntos de datos a gran escala
- Principios
 - Fallas constantes en almacenamiento y procesamiento
 - Escalar a miles de nodos, de ser necesario
 - Modelo de datos flexible (no relacional, incluso datos anidados)



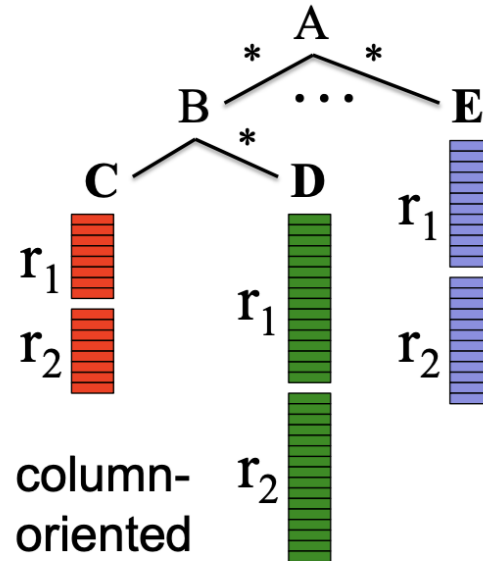
Dremel - Caso de uso

- Dremel se puede utilizar para explorar una tabla con datos crudos
- GFS se considera clave pues permite extraer datos cerca de la fuente
 - No es necesario otra base de datos donde se cargan los datos
- Formato de almacenamiento es compartido por otros sistemas
 - Se utiliza anidamiento, en lugar de columnas

Dremel - Alternativas de almacenamiento



record-
oriented





Dremel - Modelo de datos

- Características
 - Fuertemente tipados y jerárquicos
 - Tipos atómicos estándar: enteros, flotantes, strings, etc.
 - Se pueden expresar registros (*Records*) que son colecciones de campos
 - Campos repetidos, que expresan colecciones del mismo tipo de datos que puede ocurrir múltiples veces dentro de un *Record*
 - Opcionalidad u obligatoriedad de los campos es posible
 - Bibliotecas de serialización y deserialización están disponibles en múltiples lenguajes

Protocol Buffer - Ejemplo Documento

```
DocId: 10      r1
Links
  Forward: 20
  Forward: 40
  Forward: 60
Name
  Language
    Code: 'en-us'
    Country: 'us'
  Language
    Code: 'en'
  Url: 'http://A'
Name
  Url: 'http://B'
Name
  Language
    Code: 'en-gb'
    Country: 'gb'
```

```
message Document {
  required int64 DocId;
  optional group Links {
    repeated int64 Backward;
    repeated int64 Forward; }
  repeated group Name {
    repeated group Language {
      required string Code;
      optional string Country; }
    optional string Url; }}
```

```
DocId: 20      r2
Links
  Backward: 10
  Backward: 30
  Forward: 80
Name
  Url: 'http://C'
```



Dremel - Codificación y Almacenamiento

- Minimizar la cantidad de metadatos necesarios para codificar
 - Datos son opcionales con mucha frecuencia
- Reconstrucción de los objetos eficiente
- Agrupamiento de datos de los mismos campos en un objeto contiguamente
- Optimiza el acceso por columnas

Dremel - Nivel de Repetición

- Cuántos elementos en una colección han pasado hasta el momento.
- Links.Forward
 - Repetición del primer valor se define como 0
 - No ha habido ocurrencias de repetición
 - Los siguientes valores de la misma secuencia tendrán un valor de 1
 - 80 se almacena consecutivamente, aunque no corresponde al mismo objeto conceptual.

DocId	Name.Url	Links.Forward	Links.Backward
value r d	value r d	value r d	value r d
10 0 0	http://A 0 2	20 0 2	NULL 0 1
20 0 0	http://B 1 2	40 1 2	10 0 2
	NULL 1 1	60 1 2	30 1 2
	http://C 0 2	80 0 2	

Name.Language.Code	Name.Language.Country
value r d	value r d
en-us 0 2	us 0 3
en 2 2	NULL 2 2
NULL 1 1	NULL 1 1
en-gb 1 2	gb 1 3
NULL 0 1	NULL 0 1

Dremel - Nivel de Definición

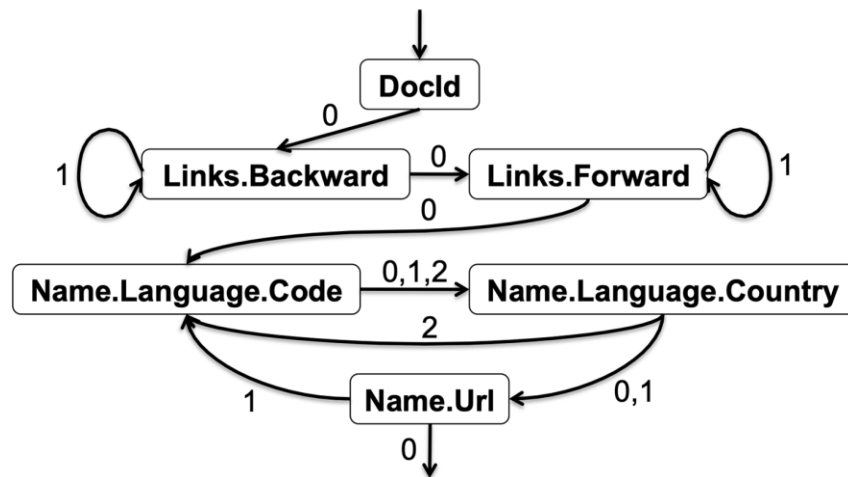
- Cuántos campos en la ruta de búsqueda que podrían no estar definidos realmente están presentes
- Permite saber cuántos elementos opcionales o repetidos están definidos previo al campo de interés

DocId	Name.Url	Links.Forward	Links.Backward
value r d	value r d	value r d	value r d
10 0 0	http://A 0 2	20 0 2	NULL 0 1
20 0 0	http://B 1 2	40 1 2	10 0 2
	NULL 1 1	60 1 2	30 1 2
	http://C 0 2	80 0 2	

Name.Language.Code	Name.Language.Country
value r d	value r d
en-us 0 2	us 0 3
en 2 2	NULL 2 2
NULL 1 1	NULL 1 1
en-gb 1 2	gb 1 3
NULL 0 1	NULL 0 1

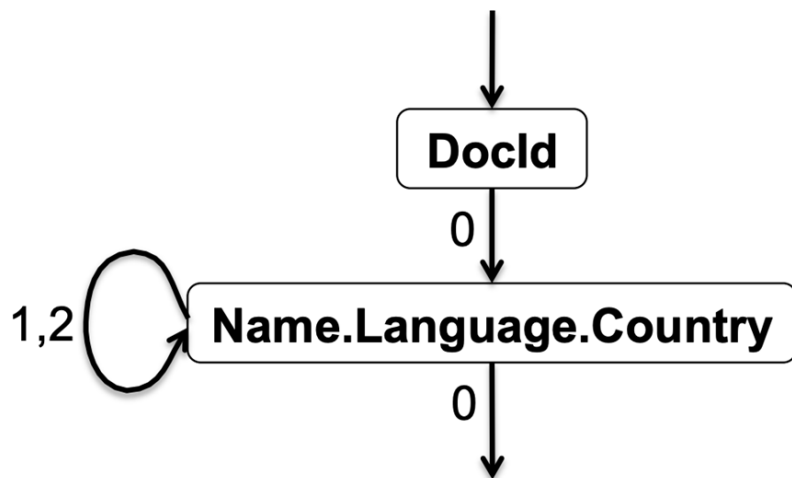
Dremel - Reconstrucción

- Requiere máquina de estados finitos que representa la jerarquía de los datos
- Recorre los datos de forma jerárquica
 - Como un árbol
 - Utiliza los niveles de repetición para el recorrido



Dremel - Reconstrucción

- Datos menos complejos implican ejecución de máquinas más sencillas



DocId: 10	S ₁
Name	
Language	
Country: 'us'	
Language	
Name	
Language	
Country: 'gb'	

DocId: 20	S ₂
Name	

Dremel - Lenguaje de consultas

- Estilo SQL
- El motor hace un mapeo entre SQL y el modelo de datos interno de Dremel

```
SELECT DocId AS Id,  
       COUNT(Name.Language.Code) WITHIN Name AS Cnt,  
       Name.Url + ',' + Name.Language.Code AS Str  
FROM t  
WHERE REGEXP(Name.Url, '^http') AND DocId < 20;
```

```
Id: 10  
Name  
  Cnt: 2  
  Language  
    Str: 'http://A,en-us'  
    Str: 'http://A,en'  
Name  
  Cnt: 0
```

t_1

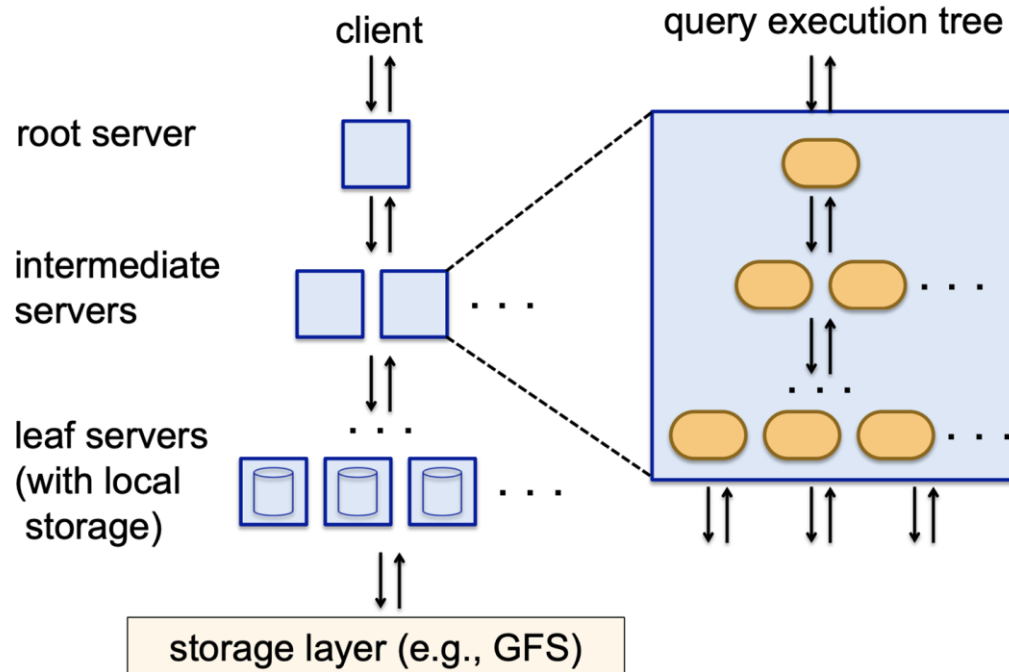
```
message QueryResult {  
  required int64 Id;  
  repeated group Name {  
    optional uint64 Cnt;  
    repeated group Language {  
      optional string Str; }}}}
```



Dremel - Lenguaje de consultas

- La selección recorre el árbol de datos y poda las ramas que no cumplen con las restricciones
 - Plasmado en cláusula WHERE
 - En el ejemplo sólo registros anidados que inician con http se conservan
- Cada expresión del SELECT genera valores al mismo nivel de anidamiento que el campo de entrada con mayor repetición
 - La concatenación de strings en el ejemplo genera valores a nivel de Name.Language.Code
- Dremel soporta agregaciones a nivel de cada registro
 - En el ejemplo, el conteo de códigos de lenguaje se hace para cada nombre

Dremel - Motor de ejecución



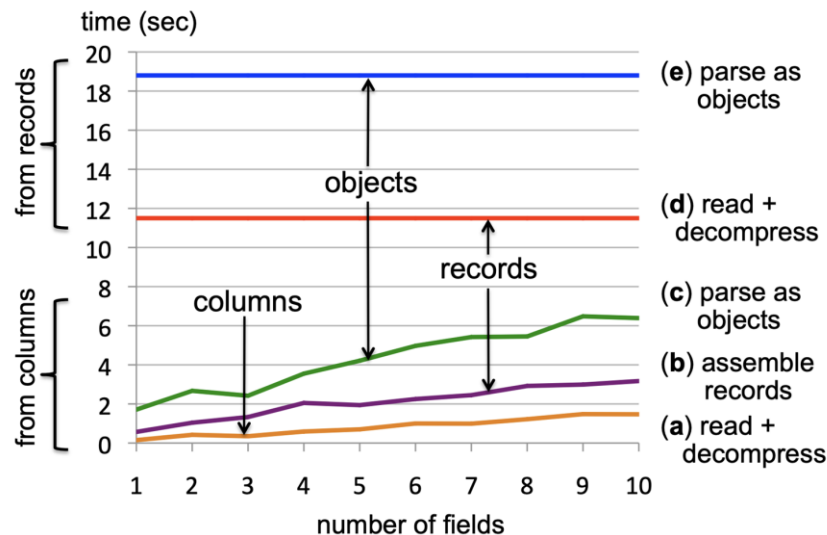


Dremel - Análisis de rendimiento

- Comparación entre leer datos de almacenamiento orientado a columnas versus orientado a registros
- Para un número pequeño de columnas puede haber ganancias de un orden de magnitud.
- Existe un costo en la reconstrucción que debe ser considerado
- Se menciona ganancia de rendimiento en el orden de docenas de columnas, pero no más de eso.

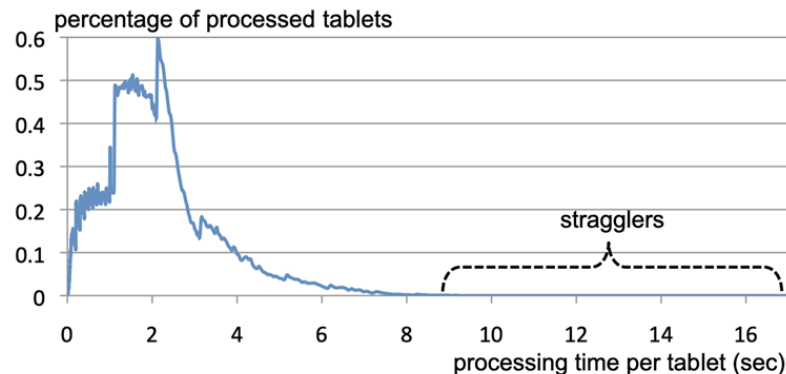
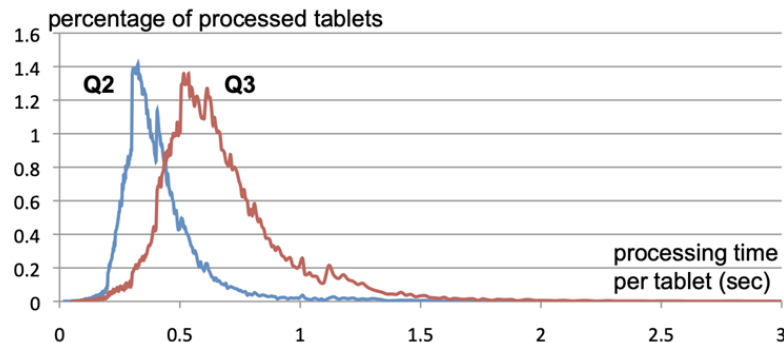
Dremel - Análisis de rendimiento

- (a) y (b) rendimiento leyendo y ensamblando los objetos anidados
- (c) muestra el costo de expresar objetos fuertemente tipados en C++
- (d) y (e) muestran el rendimiento en almacenamiento orientado a registros



Dremel - Análisis de rendimiento

- Otro hallazgo interesante es el rendimiento de los diferentes nodos de ejecución en diferentes consultas
- Se puede observar una distribución claramente normal
- Los autores abordan el problema que generan los nodos que tardan mucho y toman medidas para recalendarizarlos cuando es necesario





Big Query

- Versión "productizada" de Dremel ofrecida a través de Google Cloud Platform
- Un análogo funcional a éste, disponible en Amazon Web Services, es Redshift
 - <https://aws.amazon.com/redshift>
- Como herramienta, BigQuery aborda el problema de procesamiento analítico en línea para exploración y toma de decisiones
- Se puede comparar con otras formas de procesar datos
 - Map Reduce
 - Data Warehousing



Usar Big Query sobre Map Reduce

- Acceso programático no es necesario o es costoso, en tiempo
 - Algunos casos de uso requieren el poder de un lenguaje general
 - SQL es limitado en muchas situaciones
- Tiempos de respuesta rápidos
 - En el orden de segundos si es posible
- Análisis de datos basados en prueba y error o exploración



Usar MapReduce sobre Big Query

- Requerimientos algorítmicos avanzados
 - SQL es un lenguaje "de una pasada" sobre los datos
 - Existen requerimientos que necesitan que se itere sobre los datos múltiples veces
- Uniones de datos muy grandes sobre conjuntos de datos de tamaño muy grande
 - Rendimiento de Dremel tiene una cota en función de la cantidad de campos procesados.
- Procesar datos con el fin de exportar grandes cantidades de ellos



Big Query vs Data Warehousing

- La construcción de un Data Warehouse puede requerir inversión significativa
 - Tiempo en diseño
 - Tiempo en implementación
 - Recursos monetarios para la operación
- Existen 3 maneras comunes de abordar el problema
 - OLAP Relacional
 - OLAP Multidimensional
 - Scan completo



OLAP Relacional

- Basado en el uso de bases de datos relacionales para hacer OLAP
- Fuerte dependencia de la creación de índices
 - Herramienta primaria para cumplir con los requerimientos de rendimiento
- Existe el riesgo que la cantidad de índices necesarios sea considerable
- Requiere una cantidad de almacenamiento que puede ser mayor a la de los datos
- Puede afectar el rendimiento de escritura



OLAP Multidimensional

- Cubre los tradicionales cubos de inteligencia de negocios.
- Requiere un costo significativo de diseño de dimensiones al principio
 - No hay certeza que sea resistente al paso del tiempo
- Ejemplos de dimensiones
 - Identificadores de usuarios
 - Fechas
 - Lugares
- Una vez que se diseña se empiezan a registrar hechos (facts)
 - Totales de tiempos
 - Totales de dinero
 - Conteos
- Después pueden ser agregados para responder a consultas rápidamente



Scan completo

- Única alternativa si las estrategias anteriores no funcionan
 - Hacer un barrido completo de tablas
- La manera más efectiva de implementar esta estrategia es colocar las tablas en almacenamiento rápido (SSD / Flash)
- Normalmente toman la forma de equipo especializado que contiene toda la información
 - Se puede optimizar la manera de almacenar los datos y el throughput de entrada/salida de los discos
- El argumento de BigQuery es que éste se sitúa en la tercera clase
 - Por la infraestructura de Google pueden dar un precio razonable con gran rendimiento.



Referencias

- Melnki, S; Gubarev, A; Long, J; Romer, G; Shivakumar, S; Toltob, M; Vassilakis, T. Dremel: Interactive Analysis of Web-Scale Datasets.
<https://storage.googleapis.com/pub-tools-public-publication-data/pdf/36632.pdf>
- Sato, K. An Inside Look at Google BigQuery.
<https://cloud.google.com/files/BigQueryTechnicalWP.pdf>