



Big Data

The Dataflow Model

Texto Base: Juan Esquivel Rodríguez
Luis Alexander Calvo Valverde



Google Cloud Dataflows

- Basado en The Dataflow Model que busca la capacidad de procesar datos potencialmente ilimitados
- Procesamiento de tipo de datos "no tradicionales" es un reto
 - Bitácoras Web
 - Flujos continuos de estadísticas desde teléfonos celulares
 - Redes de sensores activos 24/7
- Atributos no enfrentados antes
 - Baja tolerancia a retardo. La información se actualiza en tiempo real
 - Tolerancia al desorden. Paquetes transmitidos no están ordenados en secuencia
 - Correctitud. No es una excusa las complejidades anteriores
 - Optimización de costo. Lo anterior debe poder cumplirse costo efectivamente
- Es muy difícil optimizar todos
 - The Dataflow Model provee infraestructura para balancearlos



Ejemplo - Plataforma Videos

- Plataforma que distribuye videos en línea (e.g. YouTube)
- Se muestran anuncios en ciertos puntos de los videos
- Tipos de reproducción
 - En línea
 - Sin conexión directa
- Es necesario que los anuncios continúen operando, independiente de la presencia de conexión



Requerimientos - Plataforma Videos

- Debe saber cuánto cobrar a cada anunciante por día
- Tener estadísticas agregadas sobre los videos y anuncios mostrados
- Los anunciantes desean saber
 - Qué tan frecuentemente se mostraron sus anuncios
 - Por cuánto tiempo
- Los proveedores de contenido desean saber
 - Frecuencias
 - Largos de reproducción
- Cuáles videos se mostraron con cuáles anuncios?
- Qué segmentos demográficos vieron tales o cuáles anuncios?



Requerimientos - Plataforma Videos

- Estadísticas en tiempo real
 - Deben ajustar sus presupuestos
 - Precios máximos en subastas
 - Cambiar las configuraciones de sus campañas.
- Correctitud es esperada
 - Maneja dinero de los anunciantes y pago a los proveedores de contenido.
- Modelo de programación que sea sencillo para desarrollar aplicaciones de procesamiento y análisis de datos
- Habilidad para manejar datos globalmente distribuidos
- En resumen
 - Sesiones que describen quién vio cada video, con qué anuncio y por cuánto tiempo



Inconvenientes previos (según autores)

- Ejecución batch puede tener alto retardo
 - Deben esperar a que se complete cada grupo de eventos de interés
 - Cuestionable qué tan escalable es este abordaje para streaming
- Es común que cuando se ofrece tolerancia a fallos y escalabilidad se sacrifique correctitud o qué tan expresiva es la interfaz provista.
- Muchos no proveen semántica de exactamente una vez (exactly-once semantics).
 - Asegura que cada dato se procesa una sola vez
 - Si no se cumple, es difícil asegurar correctitud.
- No todos proveen características para manejar ventanas de datos fácilmente
- Hay una ausencia de facilidades para el manejo de los datos expresados en sesiones.



Propuesta Dataflow

- Asumir que los datos son infinitos
 - Muchas veces se asume que en algún momento terminan para poder hacer cálculos
- Debe hacerse debido a
 - Volúmenes de datos
 - Demandas de velocidad de los diferentes consumidores de datos
- Resultados calculados en base al orden en que se generaron los eventos fuente
 - Sin asumir que son finitos u ordenados
 - Parece contradictorio pero lo manejan correctamente



Propuesta Dataflow

- La implementación se divide en cuatro fases:
 - Qué resultados se calculan?
 - Dónde se calculan, con respecto al tiempo en que ocurrieron?
 - Cuándo se materializan, en tiempo de procesamiento?
 - Cómo los eventos se relacionan con resultados previos para refinarlos?
- Flexibiliza la selección de batch, micro-batch y streaming
 - Permite balancear las variables de correctitud, retardo y costo
 - Usa diferentes motores de ejecución para cada uno

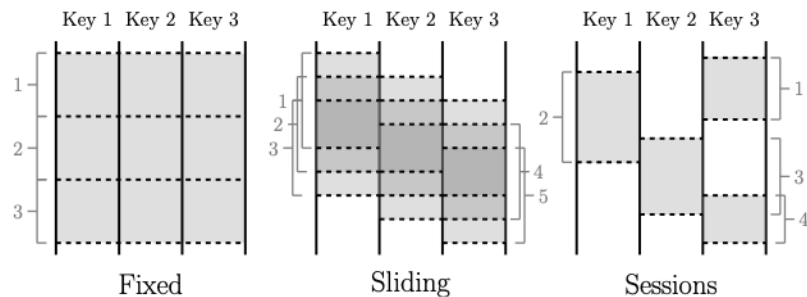


Ventanas (Windowing Patterns)

- Los datos son procesados en grupos llamados ventanas
 - Mayoritariamente basados en proximidad temporal de los eventos
- Divisiones no siempre son necesarias
 - E.g. para filtrar o transformaciones básicas
 - Llegan a serlas cuando se quiere obtener agregados, por ejemplo.
- Tipos de ventanas
 - Fijas
 - Deslizantes
 - Sesiones

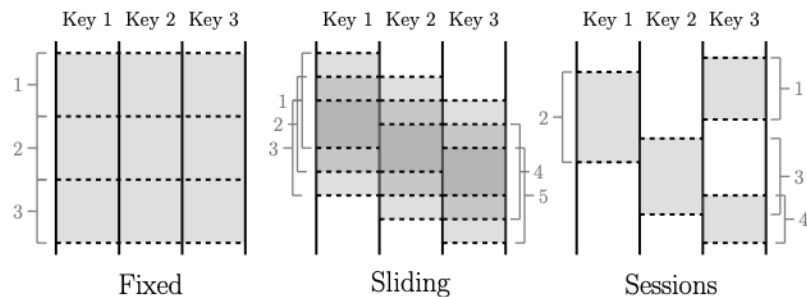
Ventana fija

- Cantidades de tiempo predeterminadas
 - E.g. hora o un día
- Se dice que están alineadas porque todos los datos deben adaptarse al tamaño de la ventana



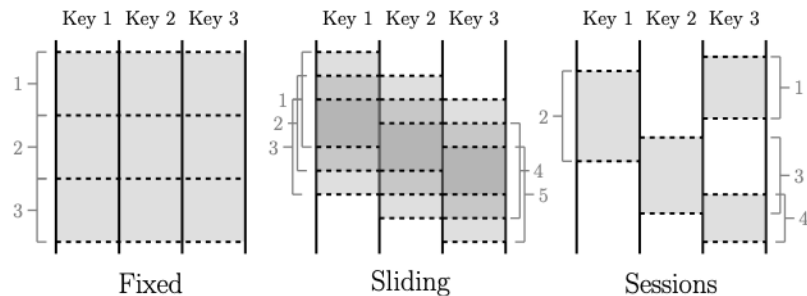
Ventana deslizante

- Combinan un tamaño de ventana más un desplazamiento
 - E.g. una ventana puede abarcar una hora y cada minuto se inicia una nueva ventana de ese tamaño.
- Pueden traslaparse. Se consideran alineadas porque aplican a todos los datos por igual, aún cuando varían en el tiempo.



Sesiones

- Períodos de actividad realizadas por una entidad de interés (e.g. persona)
- Se define un tiempo después del cual se considera terminada la sesión
 - Todos los eventos previo a ello se agrupan
- No se consideran alineadas porque la duración de cada sesión puede ser diferente
- Límites de tiempo no son los mismos para diferentes llaves.





Tiempo Evento vs Tiempo Procesamiento

- Tiempo de evento
 - Cuando el evento realmente ocurre
 - E.g. hacer una compra en línea
- Tiempo de procesamiento
 - Cuando la infraestructura de procesamiento logra atender el evento
 - E.g. cuando un pipeline de Spark obtiene el mensaje para crear agregados por hora
- El planteamiento más seguro es asumir que habrá un retardo
 - La infraestructura debe poder manejarlo



Modelo de ventanas

- Dataflow permite utilizar ventanas que no están alineadas
 - No es el caso en otros sistemas
 - El modelo asume que las ventanas no necesariamente lo están, por generalización
- Implementaciones a bajo nivel pueden realizar optimizaciones
 - Si se sabe que estarán alineados, las implementaciones toman las decisiones
- La administración de ventanas se realiza a través de dos operaciones
 - `Set<Window> AssignWindows(T datum)`
 - Un elemento puede ir a múltiples ventanas.
 - `Set<Window> MergeWindows(Set<Window> windows)`
 - Cuando se deben agrupar los datos se funden
 - Note que esto permite no sólo agrupar por tiempo sino por características de los datos
 - Cada estrategia debe implementar las operaciones de manera apropiada



Generación de datos (Triggering Model)

- La creación de ventanas por sí sola no define cuando se generan resultados
 - E.g. escribir a una base de datos
- Los triggers se disparan en respuesta a señales internas y externas para emitir datos
 - Estimación de completitud de eventos (e.g. cuando llega a un porcentaje)
 - Momentos específicos en el tiempo
 - Cuando ciertos eventos son recibidos (llamados puntuación)
 - Combinaciones lógicas
- Los triggers también definen como los diferentes valores emitidos para una misma ventana se relacionan
 - Desechar
 - Acumular
 - Acumular y retractar



Triggering Model - Desechar

- Cada vez que se dispara los datos se desechan
- Ignora datos para ventanas que existieron previamente
 - Si en el futuro llegan datos para una ventana que ya existió, esa relación es ignorada
- Funciona cuando las emisiones deben ser independientes
 - Desde el punto de vista de los consumidores río abajo
- Es el más eficiente, si cumple los requerimientos
- Ejemplo de los videos en línea no puede utilizar este modo
 - Implicaría que los consumidores tendrían que unir las partes de un video.



Triggering Model - Acumular

- El disparar la emisión no borra los datos
- Objetivo
 - Refinarlos agregados si llegan más datos posteriormente
- El uso estándar es la sobreescritura
 - Cualquier estimado anterior se sustituye por uno más refinado
 - Converge hacia el resultado final paulatinamente



Triggering Model - Acumular y Retractor

- Cada vez que acumula deja una copia en almacenamiento persistente
- Si hubiera que generar un nuevo acumulado, el anterior se retracta
- Necesario si los consumidores río abajo deben deshacer alguna agregación en los casos que los eventos cambien
- En el caso de los videos, si un anuncio no es visto por una suficiente cantidad de tiempo, se debe retractar los datos que indican que sí se vio por una cierta cantidad de segundos
 - E.g. si sólo se quieren contar anuncios que hayan sido vistos por al menos 30 segundos



Referencias

- Akidau, T; Bradshaw, R; Chambers, C; Chernyak, S; Fernández-Moctezuma, R; Lax, R; McVeety, S; Mills, D; Perry, F; Schmidt, E; Whittle, S. The Dataflow Model: A Practical Approach to Balancing Correctness, Latency, and Cost in Massive-Scale, Unbounded, Out-of-Order Data Processing.
<https://www.vldb.org/pvldb/vol8/p1792-Akidau.pdf>