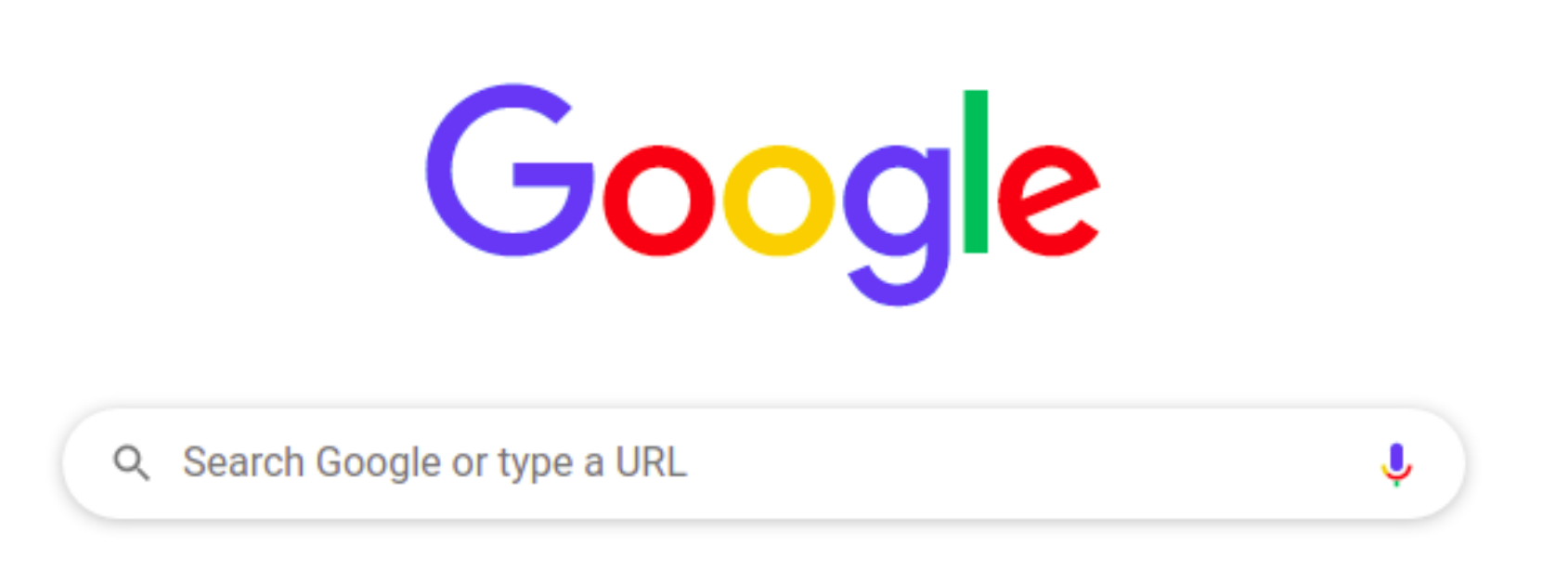
Unidad 4: Algoritmos para flujos de datos

Muestreo

Blanca Vázquez

Abril 2020

Introducción



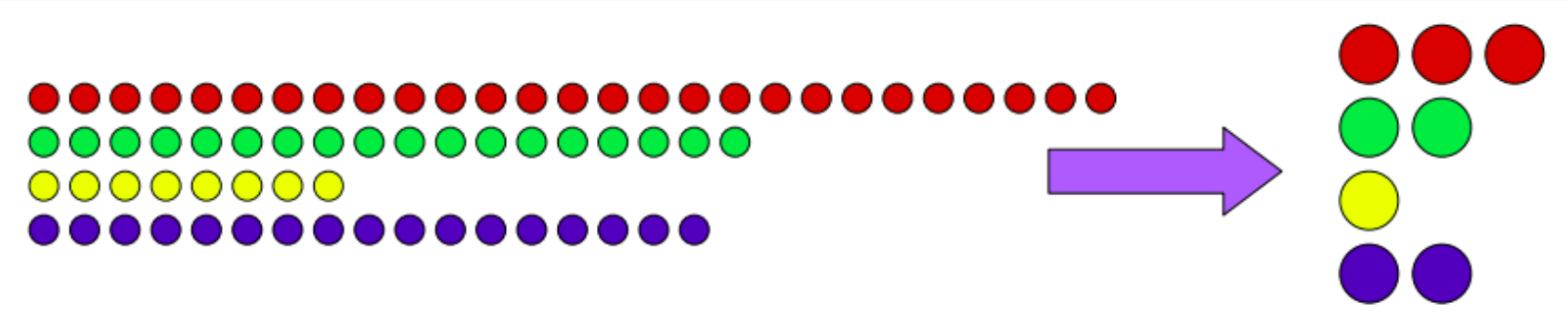
Google procesa **81,226 búsquedas** por segundo

Introducción

Uno de los principales retos del procesamiento del flujo de datos parte del **almacenamiento** y las futuras consultas que podemos hacer.

Muestreo

Debido a que en la mayoría de los casos no es posible almacenar todos los flujo de datos que se reciben, uno de las soluciones es hacer **muestreo**.



Muestrear consiste en almacenar una porción de los datos que se reciben

Muestreo

- **Ventaja:** el costo computacional es bajo, debido a que estamos solo usamos una porción de los datos recibidos.
- **Desventaja:** ¿cómo saber que tan largo es el flujo de datos?, ¿cada cuánto tiempo debemos muestrear?, ¿cómo hacemos el muestreo, por tipo de proveedor?

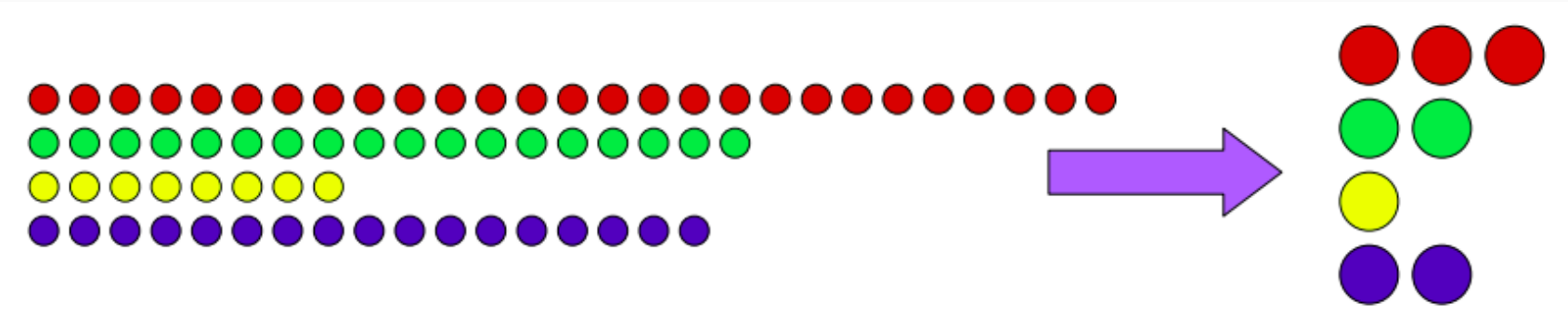
Muestreo

Existen algunas soluciones que aplican muestreo:

- Muestreo de tamaño fijo
- Muestreo aleatorio
- Ventanas deslizantes
- Histogramas
- Modelos de resolución múltiple

Muestreo de tamaño fijo

Consiste en muestrear un porción fija de los elementos recibidos (digamos 1 de cada 10 recibidos)



Muestreo de tamaño fijo (ejemplo)

- Motor de búsqueda: Google
- Entrada: flujos de datos en forma de tupla

User (IP)	Query	Time

- Pregunta: ¿Qué fracción de las consultas de un usuario son duplicadas?

Muestreo de tamaño fijo (ejemplo)

¿Qué fracción de las consultas de un usuario son duplicadas?

- Supongamos que cada usuario realiza X número de consultas únicas y D número de consultas repetidas
- El total de consultas que el usuario hace es: X + 2D
- Usando el muestreo, mantendríamos 1/10 de todas las consultas
  - x/10 (de todas las consultas únicas)
  - 2D/10 (de las consultas duplicadas)

Muestreo de tamaño fijo (ejemplo)

¿Qué fracción de las consultas de un usuario son duplicadas?

- De las D consultas duplicadas, en la muestra solo tendríamos D/100
  - D/100 = 1/10 * 1/10 * D
- De todo el conjunto de preguntas repetidas, 18D/100 aparecería realmente una vez.
  - 18D/100 = es la probabilidad de que una de las ocurrencias estará en el 1/10 del stream seleccionado, mientras que el otro estará en el 9/10 (que no fue seleccionado)
  - 18D/100 = ((1/10 * 9/10)+(9/10 * 1/10)) * D

Muestreo de tamaño fijo (ejemplo)

¿Qué fracción de las consultas de un usuario son duplicadas?

- La respuesta basada en muestreo fijo sería:

$$\frac{D}{\frac{X}{10} + 19D}$$

- Por lo tanto:

$$\frac{\frac{D}{100}}{\frac{X}{10} + \frac{D}{100} + \frac{18D}{100}} = \frac{D}{10X + 19D}$$

Obtención de muestras representativas

¿Qué fracción de las consultas de un usuario son duplicadas?

- Como observamos hacer el muestreo tomando una muestra de cada usuario, puede arrojar resultados pocos confiables.
- Y si en lugar de tomar 1/10 de las búsquedas de cada usuario, tomamos 1/10 de todos los usuarios.
  - Es decir, vamos a almacenar TODAS las búsquedas
  - Descartando las consultas del resto de usuarios 9/10
  - Tomando la IP del usuario (como ID), usamos una función hash que almacene las consultas hacia cubetas
  - Como resultado, tendríamos una muestra más representativa.

Muestreo aleatorio

Una de las técnicas más comunes en el muestreo aleatorio es el 'Reservoir Sampling'

- Consiste en muestrear los primeros M elementos recibidos y los mantiene en memoria (reserva)

Reservoir Sampling - análisis

- Almacena los primeros **s** elementos del stream en la muestra **S**.
- Supongamos que hemos visto **n-1** elementos, y ahora recibimos el n^{th} elemento (n>s)
- Con probabilidad **s/n**, mantenemos el elemento n^{th}, reemplazando uno de los **s** elementos en la muestra S

Ventanas deslizantes

Es una técnica útil para el procesamiento de flujos de datos, en dónde las consultas se realizan sobre una **ventana** de tamaño *w*. Cuando un nuevo elemento llega en el tiempo **t** este expira en el momento **t + w**.

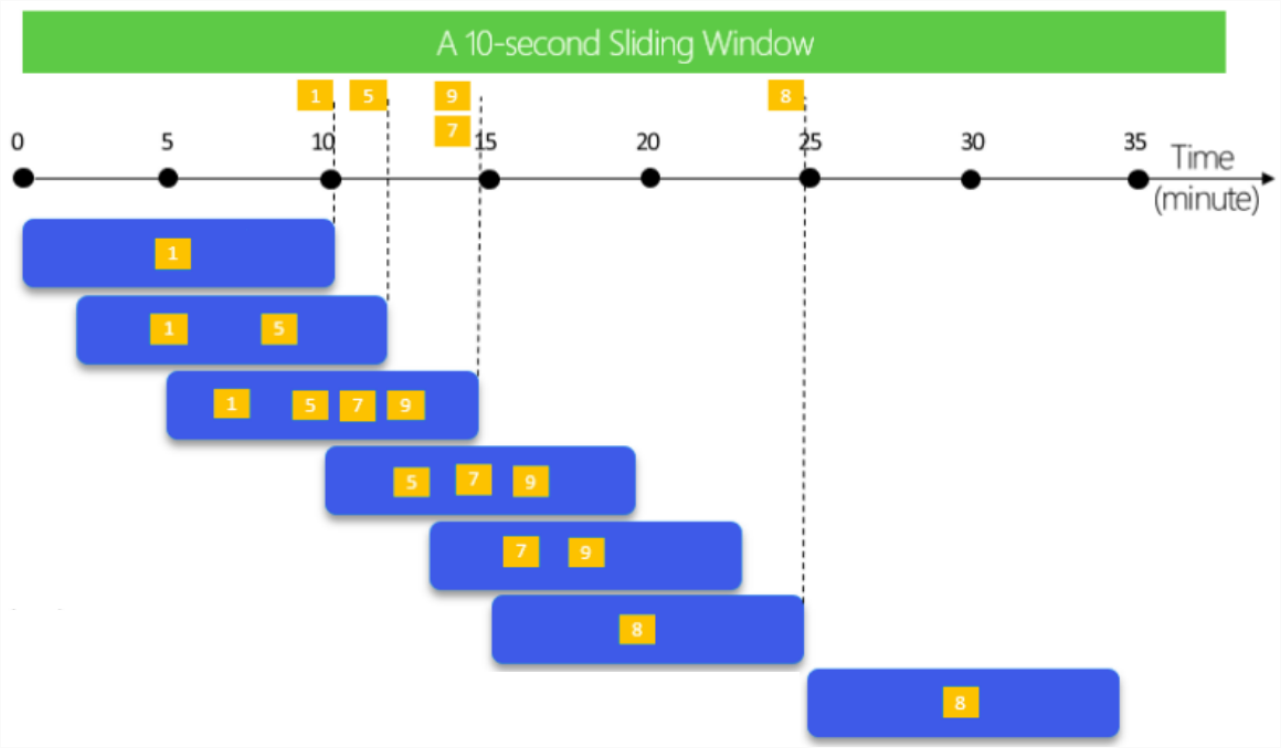


Imagen tomada de Azure Stream Analytics

Ventanas deslizantes

En este ejemplo el tamaño de la ventana deslizante es 6, observamos el traslape entre datos.

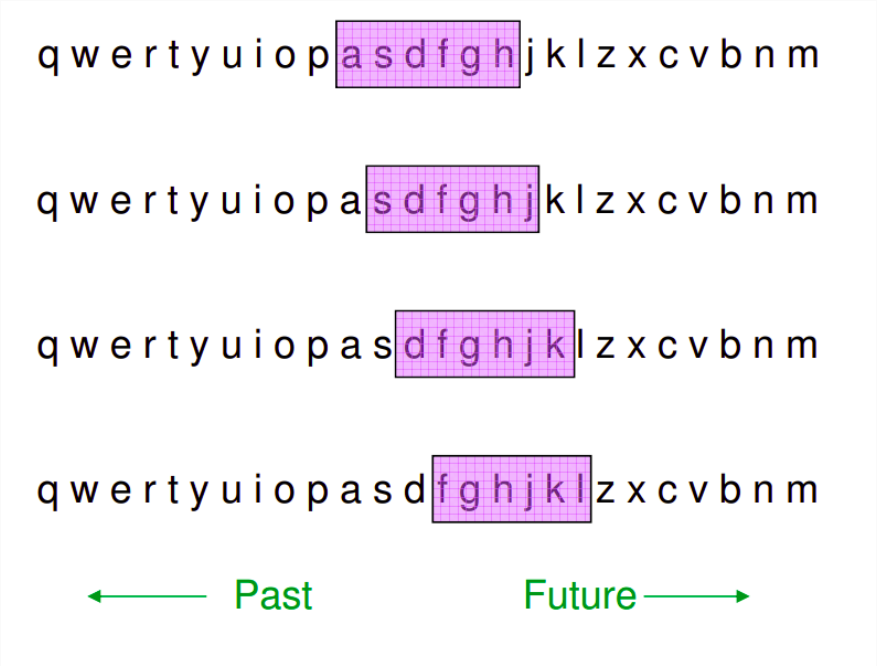
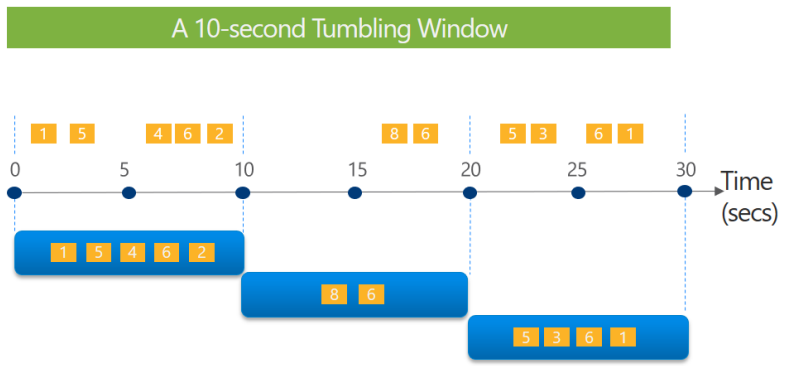


Imagen tomada de J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org

Ventanas de saltos de tamaño constante

Se usan para segmentar una transmisión de datos en segmentos de tiempo distintos y realizar una función con ellos. Una de las características principales es que no existe traslape entre los datos.

Tell me the count of tweets per time zone every 10 seconds




```
SELECT TimeZone, COUNT(*) AS Count
FROM TwitterStream TIMESTAMP BY CreatedAt
GROUP BY TimeZone, TumblingWindow(second,10)
```

Imagen tomada de Azure Stream Analytics

Ventanas de sesión

Agrupan eventos que llegan a la misma hora, filtrando los periodos en dónde no se recibe ningún dato. En este caso se deben fijar los parámetros de tiempo de espera y duración máxima.

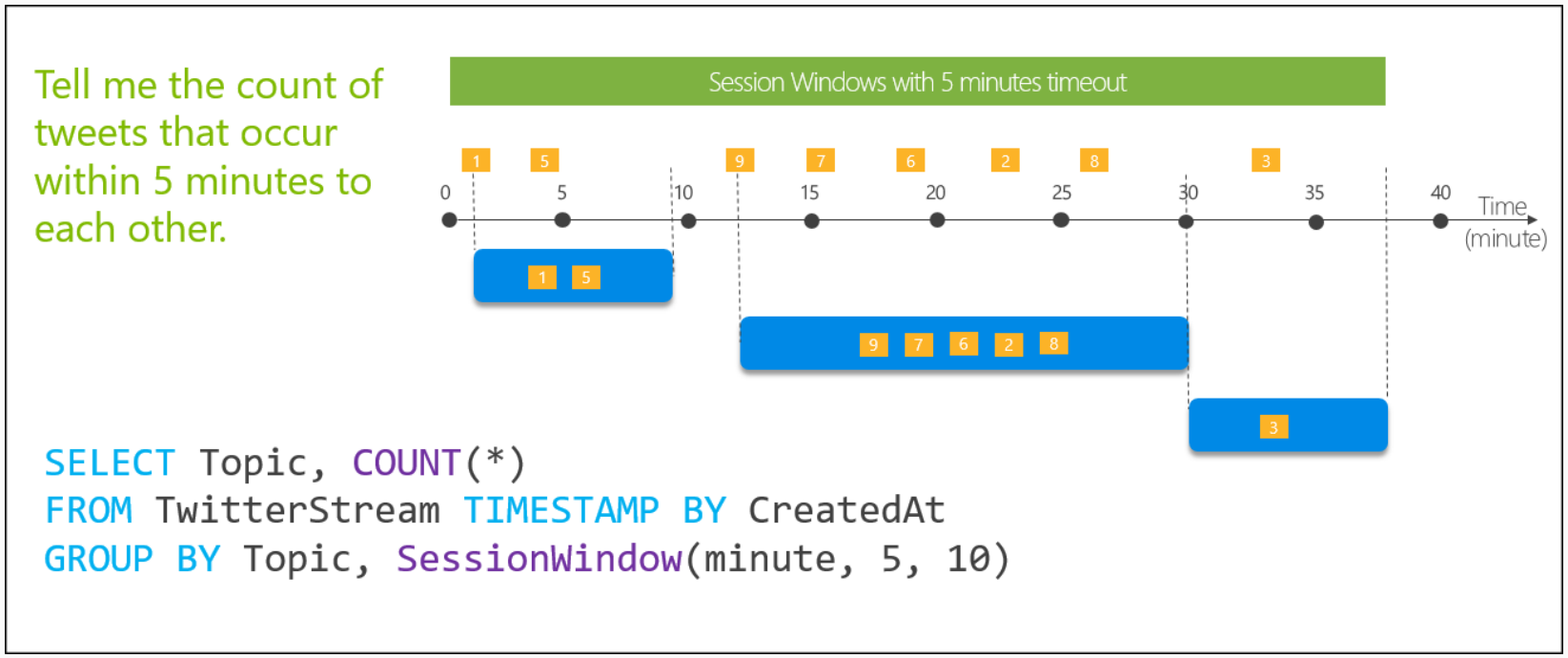


Imagen tomada de Azure Stream Analytics

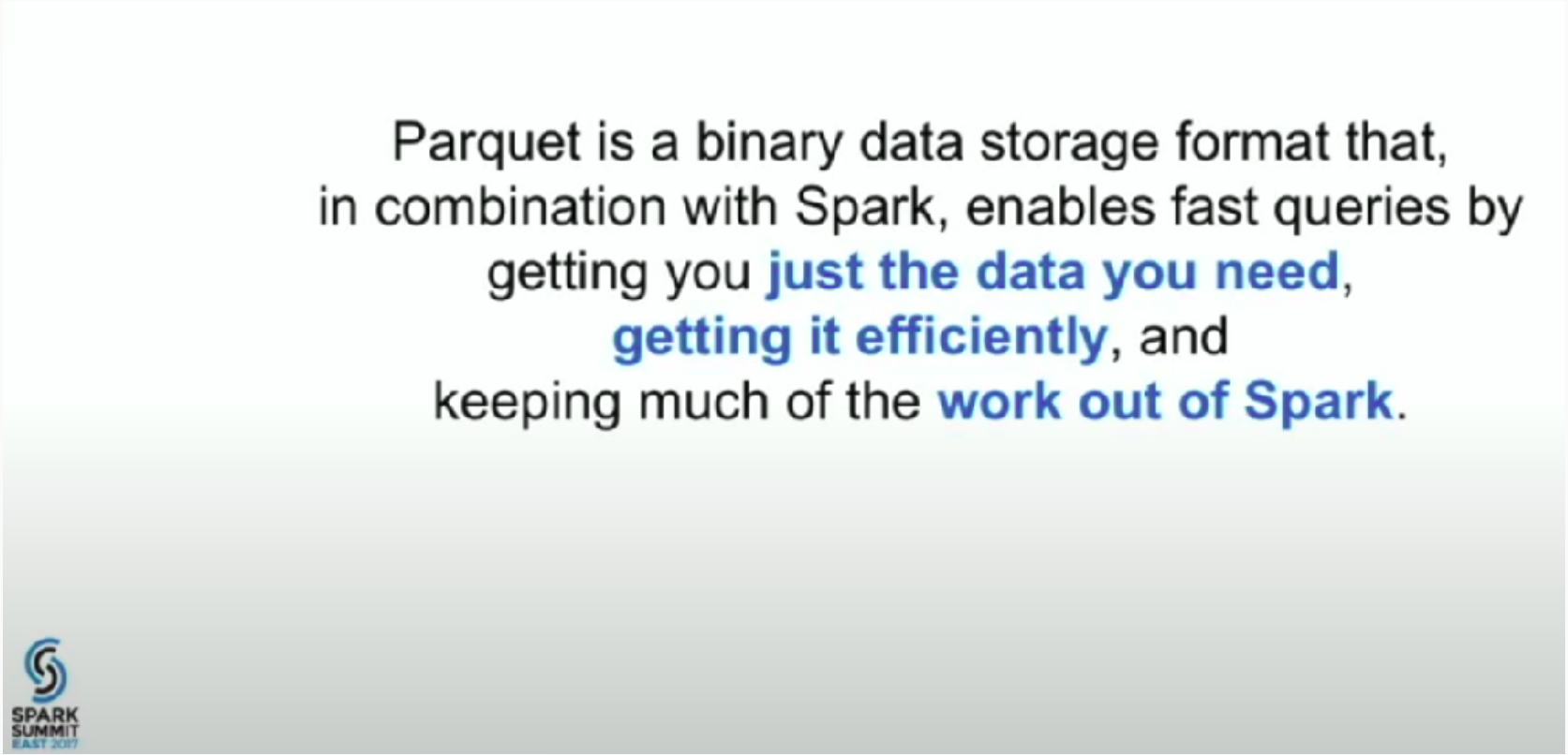
Parquet

Apache Parquet

- Initial effort by Twitter & Cloudera
- Open source storage format
  - Hybrid storage model (PAX)
- Widely used in Spark/Hadoop ecosystem
- One of the primary formats used by Databricks customers

databricks

Parquet


Parquet is a binary data storage format that,
in combination with Spark, enables fast queries by
getting you just the data you need,
getting it efficiently, and
keeping much of the work out of Spark.
SPARK
SUMMIT
EAST 2017

Parquet

	Parquet
Usability	Good!
Administration	None!
Spark Integration	FANTASTIC!!
Resource Efficiency	WONDERFUL!! (Storage, I/O, Data cardinality)
Scalability	FANTASTIC!!
CO$$$$T	¢ ¢ ¢
QUERY TIME	GOOD!!

SPARK SUMMIT EAST 2017