

Algoritmos paralelos III Map / Reduce

PROGRAMACIÓN CONCURRENTE EN JAVA - TEMA 5



Universidad
Rey Juan Carlos

Motivación

ALGORITMOS PARALELOS

- Actualmente todo gira en torno a los datos:
 - **Ciencia**: bases de datos de astronomía, genómica, datos medio-ambientales, transporte.
 - **Humanidades**: libros escaneados, documentos históricos, datos sociales.
 - **Negocio**: ventas, transacciones, censos, tráfico de aerolíneas.
 - **Ocio**: películas, música, imágenes.
 - **Medicina**: pacientes, tratamientos, radiografías
 - **Industria**: sensores.

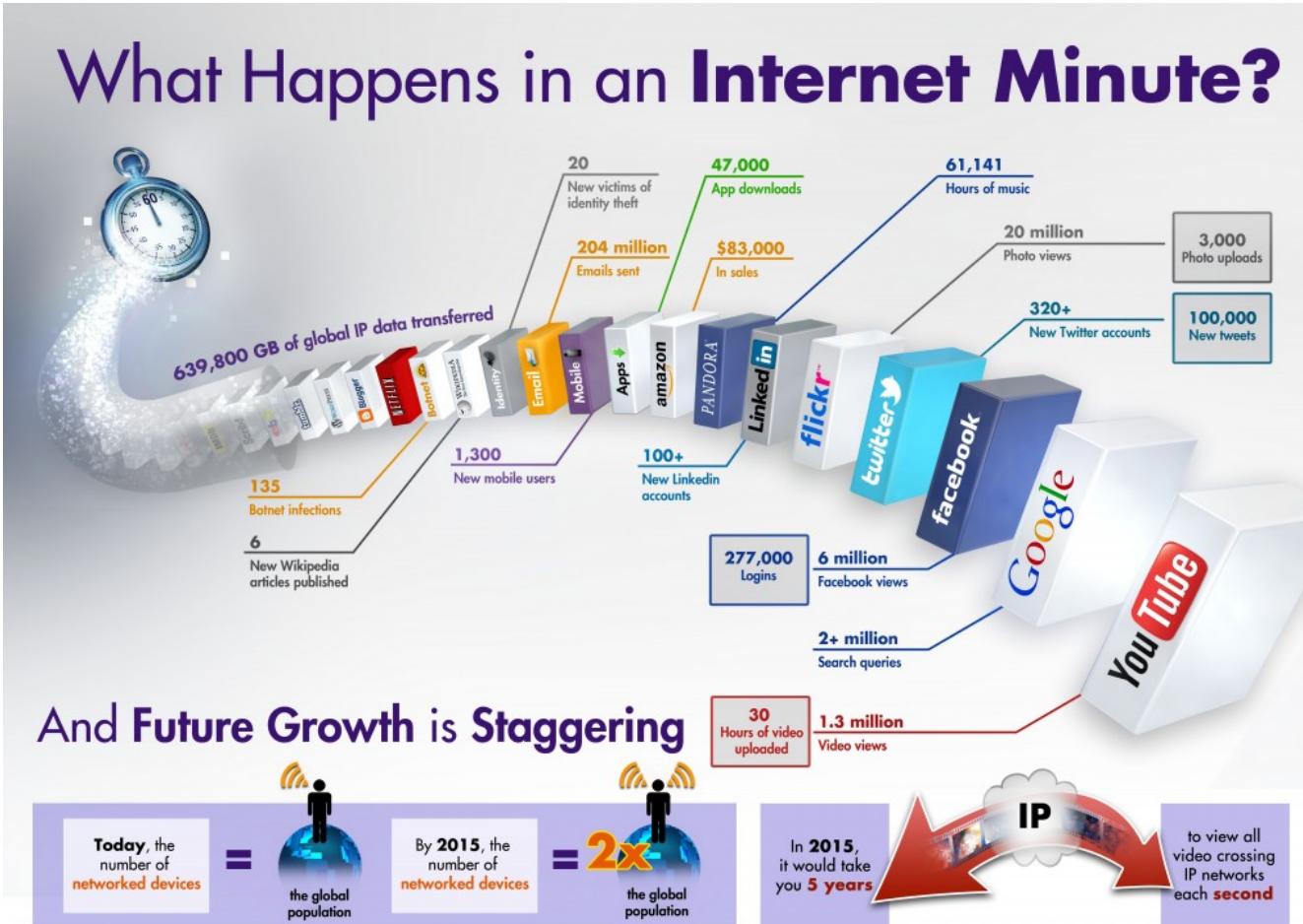
Motivación

ALGORITMOS PARALELOS

- El **crecimiento** de la cantidad de información disponible en los últimos años ha sido **exponencial**.
- Actualmente somos ricos en **datos** pero pobres en **conocimiento**.
 - No disponemos de técnicas suficientemente eficientes como para analizar todos los datos que tenemos disponibles.

Motivación

ALGORITMOS PARALELOS



Motivación

ALGORITMOS PARALELOS

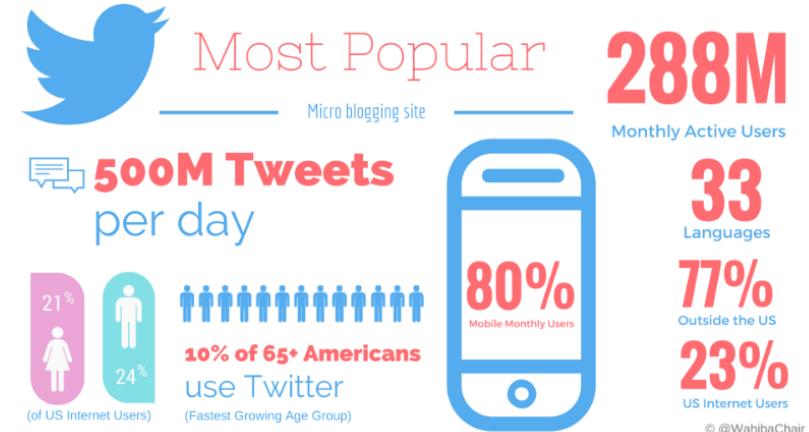
- Big Data se refiere a cualquier colección de datos lo suficientemente **grande** como para no poder ser gestionada utilizando **técnicas tradicionales**.
 - Por ejemplo, el volumen de todos los datos de Facebook o Youtube que recopilamos en un día.
- Sin embargo, Big Data **no solo** se trata de escalabilidad y tamaño de datos, sino que implica una serie de aspectos adicionales
 - Las 4 V's de Big Data

Big Data 4V's

ALGORITMOS PARALELOS

• Volumen

- El tamaño de los datos crece de manera exponencial.
- Se prevé que de 2009 a 2020 el tamaño de los datos disponibles se multiplique por 44.
 - Se pasa de 0.8 zb a 35 zb
 - Zettabyte >
Exabyte >
Petabyte >
Terabyte



Big Data 4V's

ALGORITMOS PARALELOS

• Variedad

- Los datos se presentan en diversos formatos, tipos y estructuras.
- Datos estáticos vs dinámicos
- Una única aplicación puede necesitar diferentes tipos de datos.
- Para poder analizar estos datos, debemos ser capaces de enlazarlos de alguna manera



Big Data 4V's

ALGORITMOS PARALELOS

• **Velocidad**

- Los datos se generan en tiempo real, y necesitamos procesarlos lo más rápido posible.
- Si el análisis se retrasa, no podremos tomar decisiones correctas.
 - **Publicidad:** basándonos en la localización e historial de compras, enviar promociones de compra de alguna tienda cercana.
 - **Monitor de salud:** sensores de actividad que permitan actuar en cuanto se detecte cualquier anomalía.

Big Data 4V's

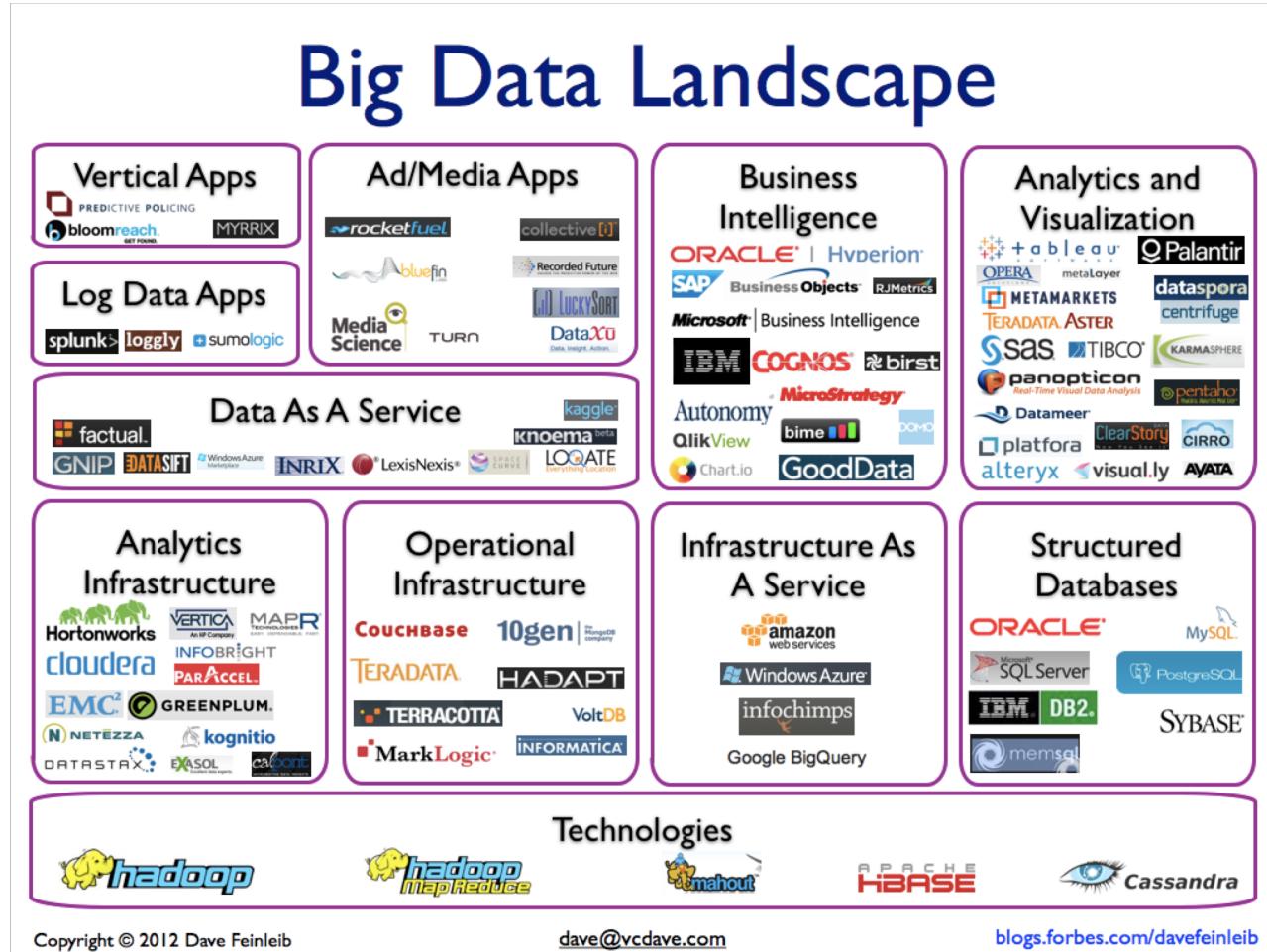
ALGORITMOS PARALELOS

• **Veracidad**

- No todos los datos que obtenemos provienen de fuentes fiables.
- Además, debemos analizar la semántica de los datos.
 - Un comentario de Twitter puede ser irónico, y debemos detectar esa ironía: análisis de sentimientos.

Perspectiva actual de Big Data

ALGORITMOS PARALELOS



MapReduce

ALGORITMOS PARALELOS

- Los sistemas tradicionales suelen disponer de un **servidor centralizado** para almacenar y procesar los datos.
- No están preparados para procesar grandes volúmenes de datos, y suelen ser el **cuello de botella** al procesar muchos ficheros simultáneamente.
- Google resolvió este cuello de botella cuando presentaron MapReduce
 - <http://static.googleusercontent.com/media/research.google.com/en//archive/mapreduce-osdi04.pdf>

MapReduce

ALGORITMOS PARALELOS

- La idea principal de MapReduce es similar a Fork / Join
 - Reducir el tamaño de las tareas y asignárselas a varios computadores de manera concurrente.
 - Los resultados se recuperan e integran para crear el resultado final.

MapReduce

ALGORITMOS PARALELOS

- MapReduce consiste en dos tareas principales: Map y Reduce.
- **Map**
 - Selecciona un conjunto de datos y lo transforma en otro, donde cada elemento se divide en pares clave-valor.
- **Reduce**
 - Selecciona la salida de *Map* como entrada y combina esos pares en conjuntos más pequeños de tuplas

MapReduce

ALGORITMOS PARALELOS

- **Fase de entrada:** un lector de datos parsea los datos y se los envía a *Map* en formato par clave-valor.
- **Map:** selecciona un conjunto de pares clave-valor y los procesa para generar cero o más pares clave-valor.
- **Claves intermedias:** Los pares generados por el *mapper* se conocen como claves intermedias.
- **Combinación:** agrupa datos similares obtenidos en el *map* en conjuntos identificables. Opcional.

MapReduce

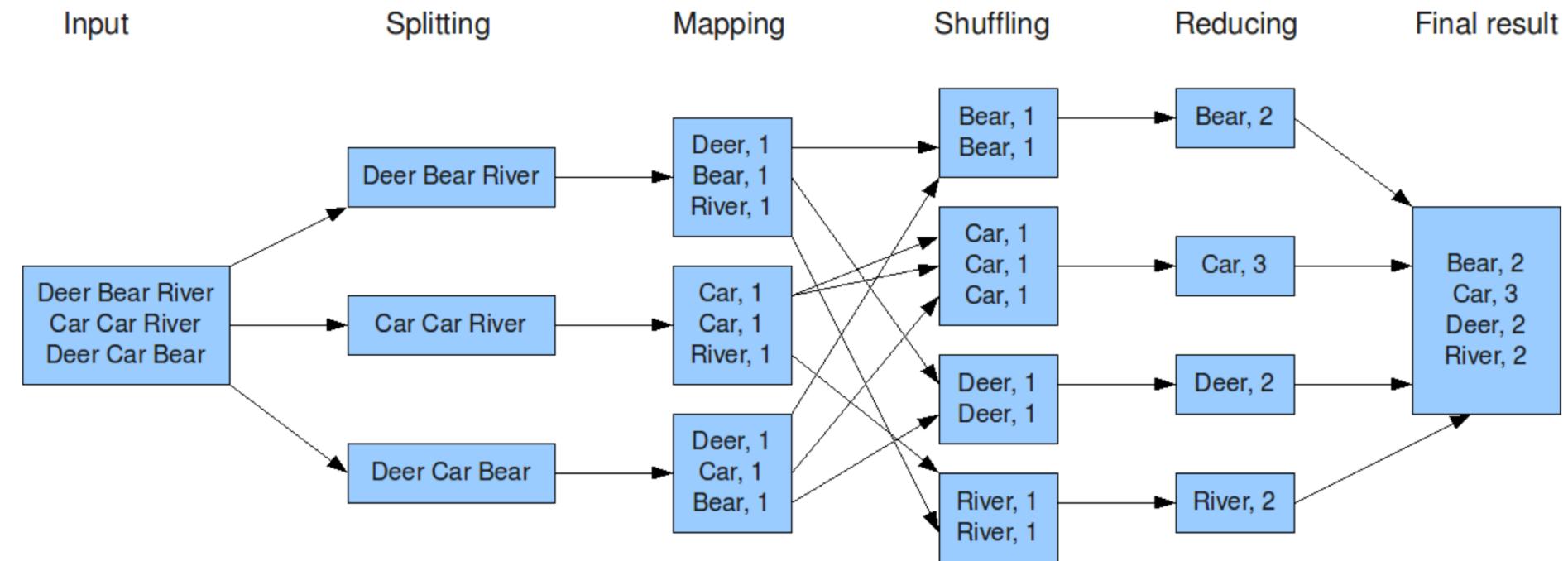
ALGORITMOS PARALELOS

- **Perturbación y ordenación:** La fase *Reduce* comienza en esta etapa. Descarga el conjunto de pares agrupados en la máquina local donde se ejecuta el *reducer*. Los pares individuales se ordenan por clave en una lista que agrupa pares equivalentes juntos para que sus valores se puedan iterar de manera sencilla por el *reducer*.
- **Reduce:** Selecciona los pares agrupados y ejecuta un *reducer* sobre cada uno. Los datos aquí se pueden agregar, filtrar y combinar de diversas maneras, requiriendo un amplio rango de proceso. Al finalizar devuelve cero o más pares clave valor a la etapa final.
- **Fase de salida:** Un formateador convierte los pares clave-valor y los escribe en el formato elegido.

MapReduce

ALGORITMOS PARALELOS

The overall MapReduce word count process



Algoritmos paralelos III Map / Reduce

PROGRAMACIÓN CONCURRENTE EN JAVA - TEMA 5



Universidad
Rey Juan Carlos