

Fork-join

Técnicas de Programación Concurrente

Fork-join

Paper de 1963

"... in an internal **merge sort** all strings in any pass may be created at the same time."

En honor al paper original,
implementemos un merge sort
concurrente

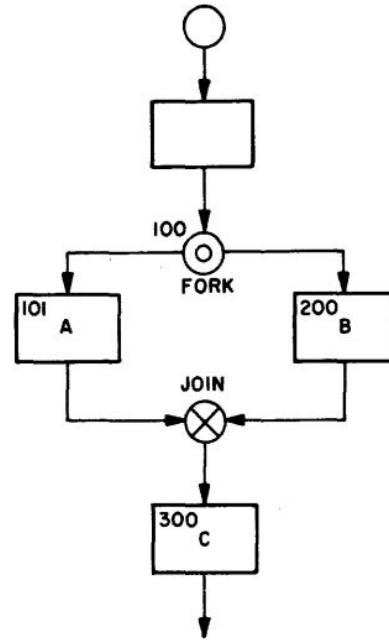
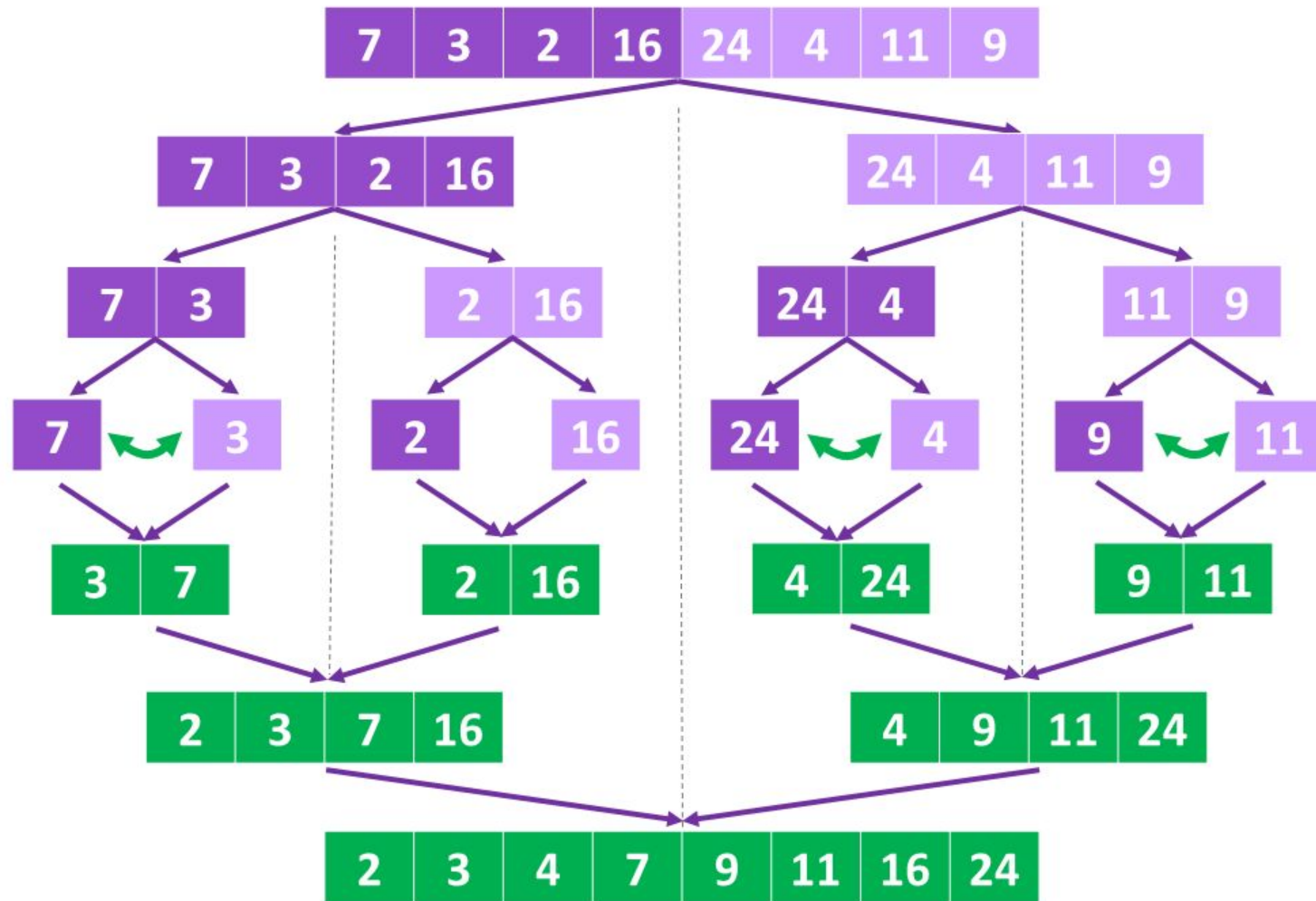


Figure 1. Conventions for drawing fork and join.



Melvin Conway

Merge Sort



Step 1:
Split sub-lists in two until you reach pair of values.

Step 3:
Sort/swap pair of values if needed.

Step 4:
Merge and sort sub-lists and repeat process till you merge to the full list.

MapReduce

Paper del año 2004

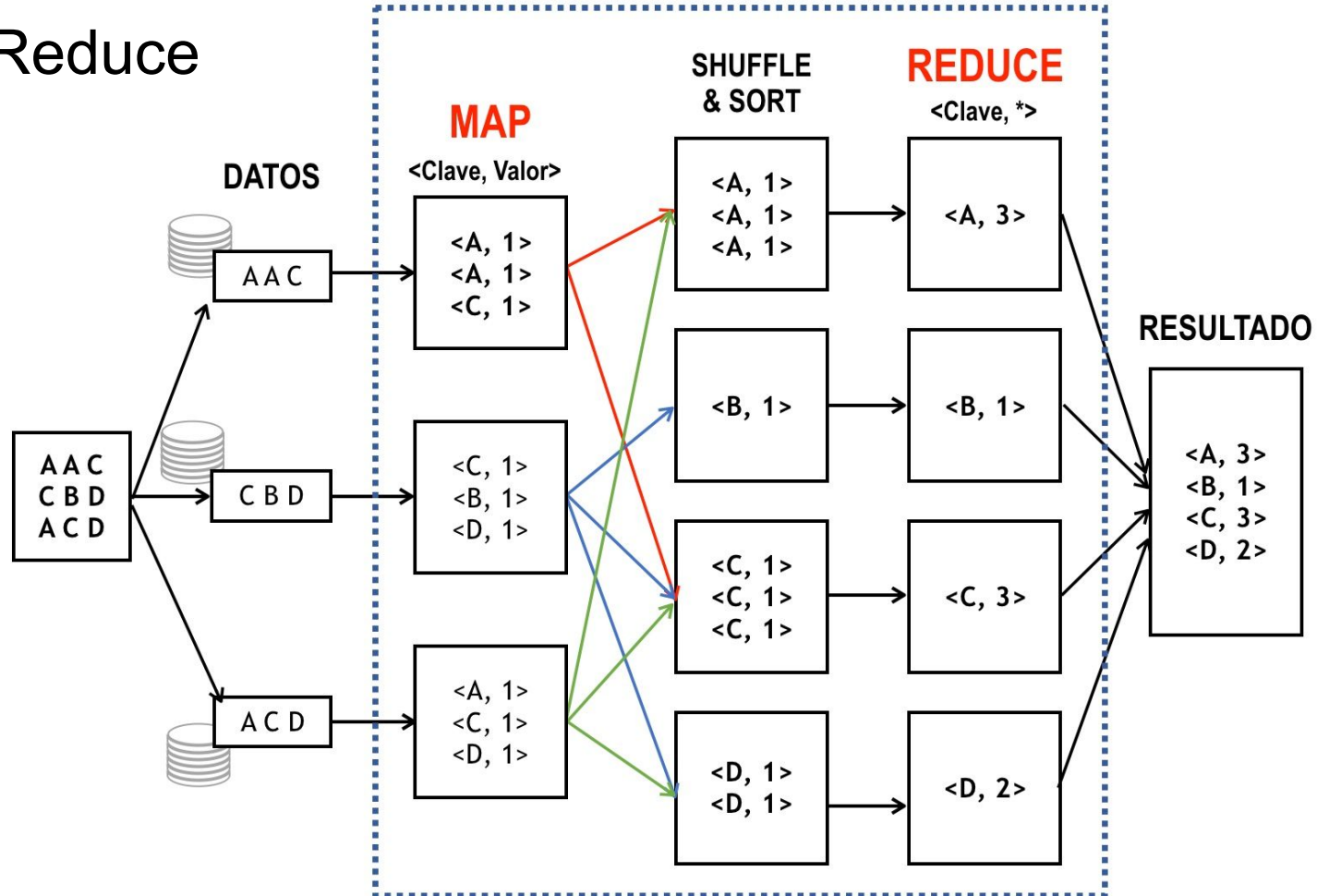
MapReduce is a programming model and an associated implementation for processing and generating large data sets. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key.

Muchas implementaciones, Hadoop siendo la primera open source

"Hijos" más modernos incluyen a Spark

MapReduce

JOB



Word count

Word count es el "Hello World" en el mundo del map reduce

- Map
 - Tomar de potencialmente varios archivos de texto, las líneas
 - De las líneas, las palabras
 - De las palabras sus frecuencias locales
- Reduce
 - Combinar las frecuencias locales en una única tabla con todas las frecuencias

Implementemos el word count!

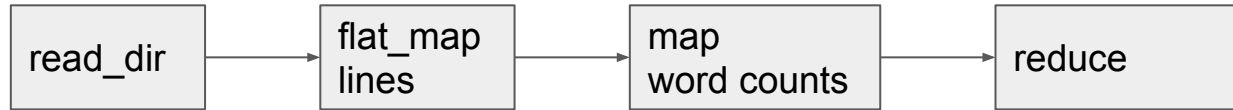
Ryon

- Rayon is a data-parallelism library for Rust. It is extremely lightweight and makes it easy to convert a sequential computation into a parallel one. It also guarantees data-race freedom.
- Rayon makes it drop-dead simple to convert sequential iterators into parallel ones: usually, you just change your `foo.iter()` call into `foo.par_iter()`, and Rayon does the rest

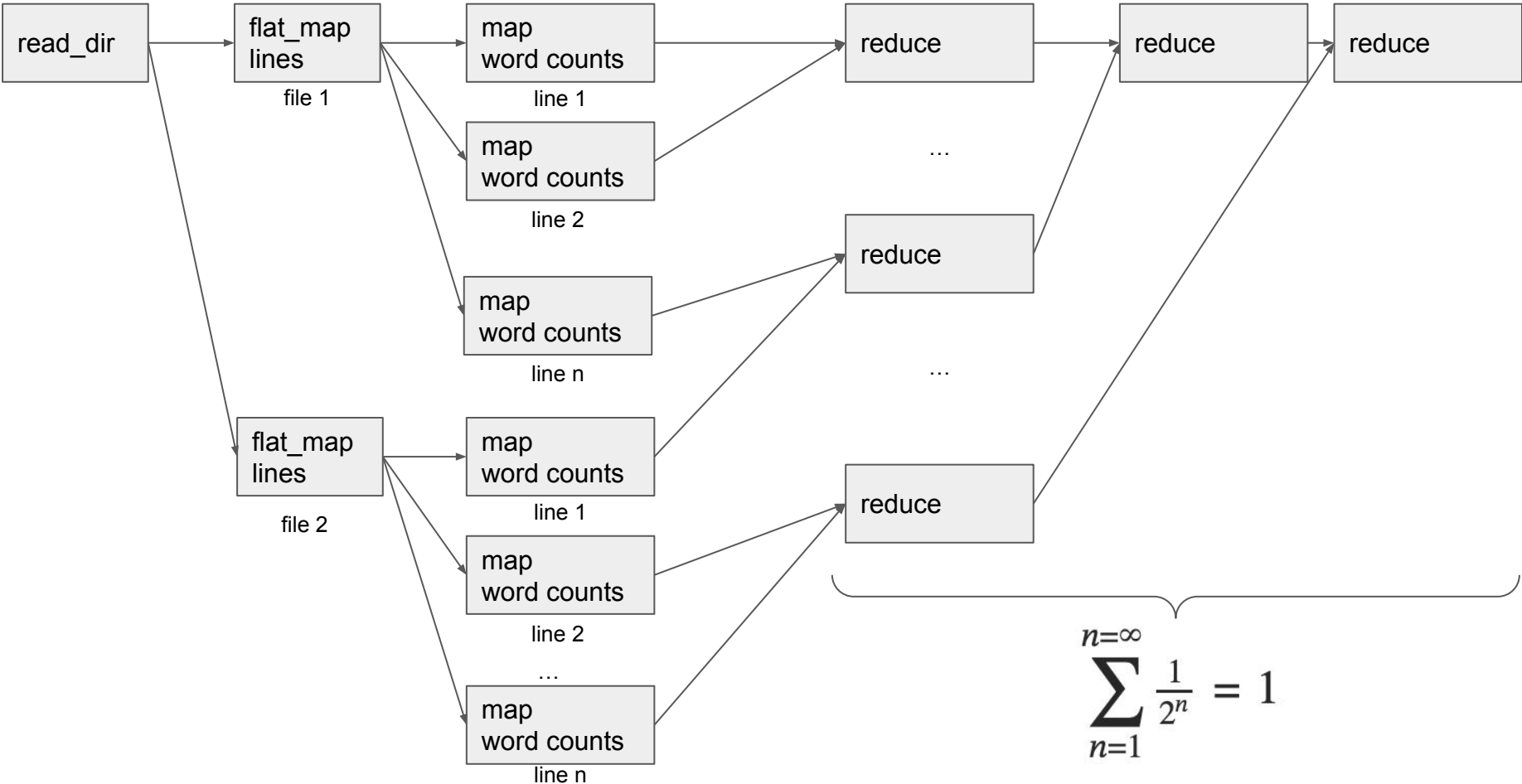
Comparar wordcount secuencial vs paralelo usando rayon.

- Performance de mergesort con Threadpool

DAG - Word count



DAG - Word count Ryon



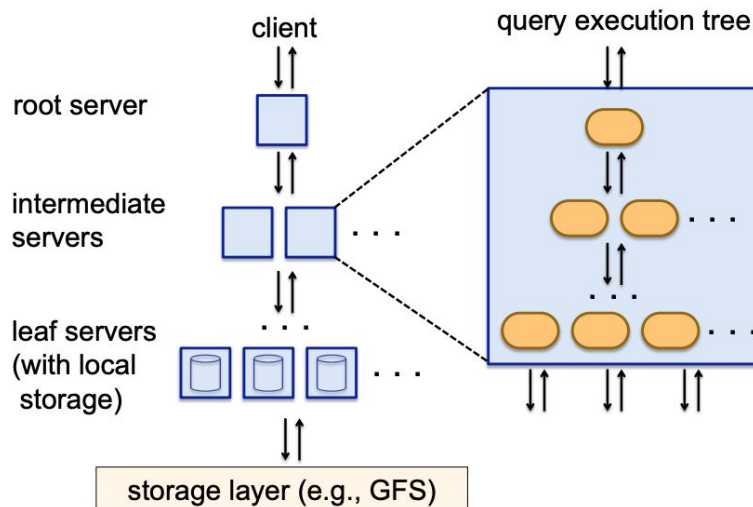
Ejercicio

- Descargar el dataset de Kaggle "Trending YouTube Video Statistics"
<https://www.kaggle.com/datasets/datasnaek/youtube-new>
- Crear un programa map-reduce con Ryon que calcule la cantidad total de vistas por channel_title. Comparar tiempos de ejecución con y sin iteradores paralelos.
- Diagramar el DAG y estimar el mínimo tiempo posible ideal de ejecución en 4 CPUs si son 10 archivos y cada archivo tiene en promedio 35 mil filas.

Dremel

Sistema de consultas en tiempo casi real,
expuesto al público como BigQuery

"A query gets pushed down the tree and is
rewritten at each step. The result of the
query is assembled by aggregating the
replies received from lower levels of the
tree"



Referencias

- A Multiprocessor System Design, M.E. Conway, 1963 -
<https://ia600703.us.archive.org/30/items/AMultiprocessorSystemDesignConway1963/A%20Multiprocessor%20System%20Design%20%28Conway%2C%201963%29.pdf>
- MapReduce: Simplified Data Processing on Large Clusters, Jeffrey Dean and Sanjay Ghemawat, 2004 -
<https://static.googleusercontent.com/media/research.google.com/es//archive/mapreduce-osdi04.pdf>
- Dremel: Interactive Analysis of Web-Scale Datasets S. Melnik et al., 2010 -
<https://storage.googleapis.com/pub-tools-public-publication-data/pdf/36632.pdf>

SIMD - Portable SIMD

MMX, SSE, AVX, NEON, etc.

SIMD - CUDA