(Big) Data Engineering In Depth

From Beginner to Professional

Moustafa Alaa

Senior Data Engineer at Onfido, London, UK

The Definitive Guide to Big Data Engineering Tasks

Previous video recap!

Chapter: Introduction To Distributed Systems

Section: Course Intro

Chapter Objectives

- Capturing the state of the art in building high performance distributed computing using Hadoop, Spark, and Kafka.

Chapter Objectives

- Capturing the state of the art in building high performance distributed computing using Hadoop, Spark, and Kafka.
- Providing the relevant theoretical and practical background and its best practices.

Chapter Objectives

- Capturing the state of the art in building high performance distributed computing using Hadoop, Spark, and Kafka.
- Providing the relevant theoretical and practical background and its best practices.
- Demonastrating the main concepts and components for distributed systems.

Chapter Objectives

- Capturing the state of the art in building high performance distributed computing using Hadoop, Spark, and Kafka.
- Providing the relevant theoretical and practical background and its best practices.
- Demonastrating the main concepts and components for distributed systems.
- Advancing the understanding of building scalable software systems for large scale data processing and its best practices.

Target Audience

- Software Engineers and Application Developers.

Target Audience

- Software Engineers and Application Developers.
- Data Analysts and DWH/Data Engineers.

Target Audience

- Software Engineers and Application Developers.
- Data Analysts and DWH/Data Engineers.
- Researchers.

Section: Design Simple Distributed System (Case Study Example 1)

Case Study Example 1

- Assume we have a file contains **1TB** of text lines, and we need to convert the text to be the upper case, for example (The -> THE).

Case Study Example 1

- Assume we have a file contains **1TB** of text lines, and we need to convert the text to be the upper case, for example (The -> THE).

- We need to design the program without using any ready distributed system framework.

Case Study Example 1

- Assume we have a file contains **1TB** of text lines, and we need to convert the text to be the upper case, for example (The -> THE).
- We need to design the program without using any ready distributed system framework.
- You can use any number (n) of machines. Assume n specs are 8 GB of memory, hard desk 128 GB, and 2 cores of CPU.

Case Study Example 1

Figure: Simple design for small file(s) processing

Case Study Example 1

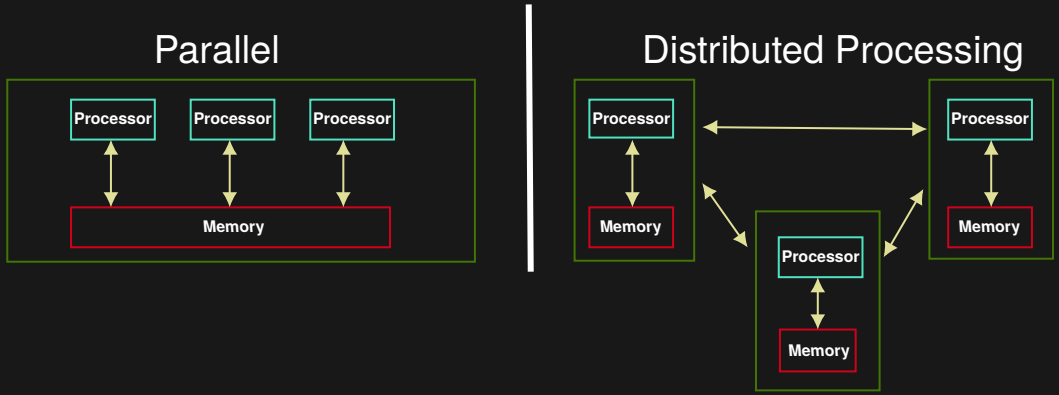


Figure: Parallel processing vs Distributed processing

Case Study Example 1

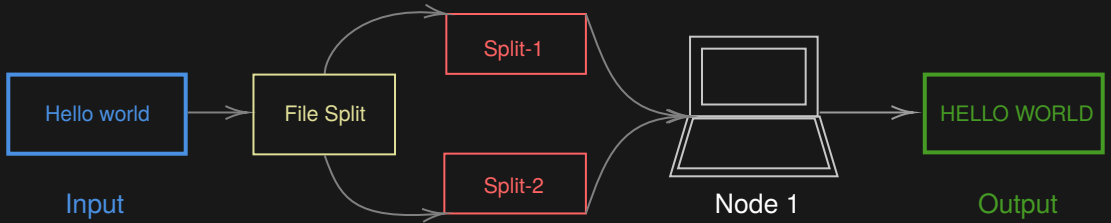


Figure: Adding File Split function to split big files into equal chunks

Case Study Example 1

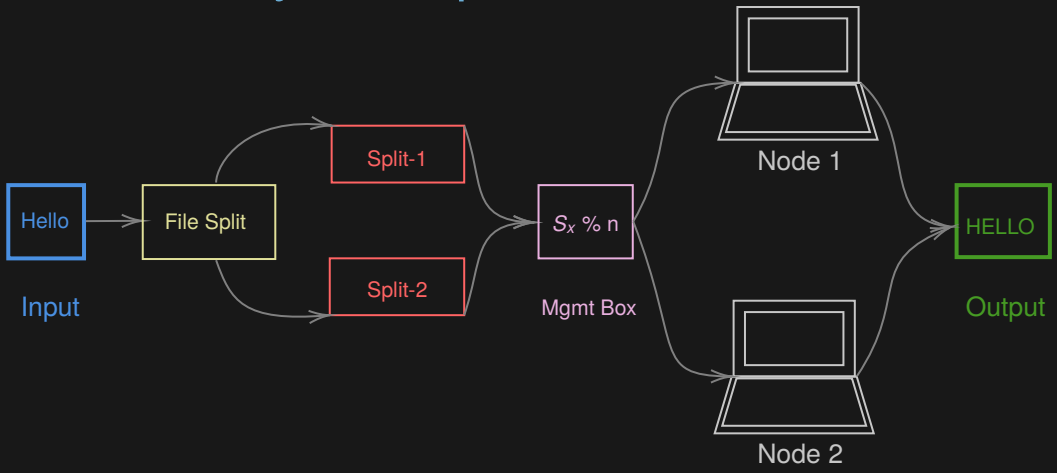
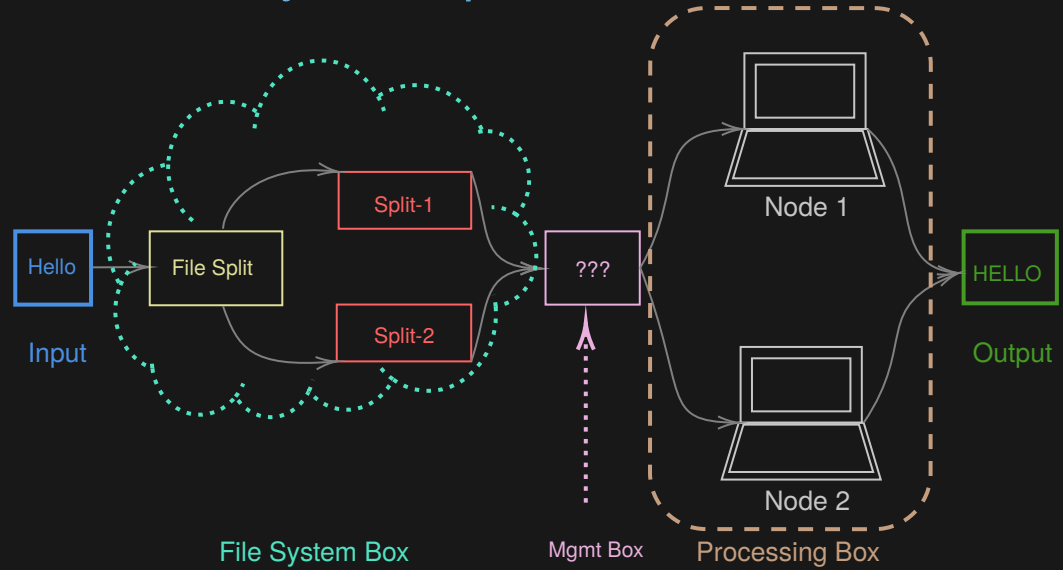


Figure: Adding another node to distribute the processing across serveral nodes (n).

Case Study Example 1



Management box

- How do we distribute the data across the nodes?

Management box

- How do we distribute the data across the nodes?
- How do we know the number of active nodes (or the status of the nodes)?

Management box

- How do we distribute the data across the nodes?
- How do we know the number of active nodes (or the status of the nodes)?
- How do we know if some tasks are stucking?

Management box

- How do we distribute the data across the nodes?
- How do we know the number of active nodes (or the status of the nodes)?
- How do we know if some tasks are stucking?
- How do we track the tasks passed to the nodes?

Management box

- How do we distribute the data across the nodes?
- How do we know the number of active nodes (or the status of the nodes)?
- How do we know if some tasks are stucking?
- How do we track the tasks passed to the nodes?
- What will happen if this box is down? How can we avoid this?

Management box

- How do we distribute the data across the nodes?
- How do we know the number of active nodes (or the status of the nodes)?
- How do we know if some tasks are stucking?
- How do we track the tasks passed to the nodes?
- What will happen if this box is down? How can we avoid this?
- How do we track the available resources (containers) in our cluster?

File System

- How can we store a massive amount of data in distributed systems?

File System

- How can we store a massive amount of data in distributed systems?
- How can we design a file system which supports highly fault-tolerant?

File System

- How can we store a massive amount of data in distributed systems?
- How can we design a file system which supports highly fault-tolerant?
- Do we require special hardware to design a distributed storage system?

File System

- How can we store a massive amount of data in distributed systems?
- How can we design a file system which supports highly fault-tolerant?
- Do we require special hardware to design a distributed storage system?
- How can we design storage systems to support distributed processing?

Data nodes

- How datanodes continuously communicate with the management node?

Data nodes

- How datanodes continuously communicate with the management node?
- How datanodes receive the tasks instructed by the management node?

Data nodes

- How datanodes continuously communicate with the management node?
- How datanodes receive the tasks instructed by the management node?
- Can datanodes store data besides their roles for processing?

Section: Design Simple Distributed System (Case Study Example 2)

Previous video recap!

Case Study Example 1

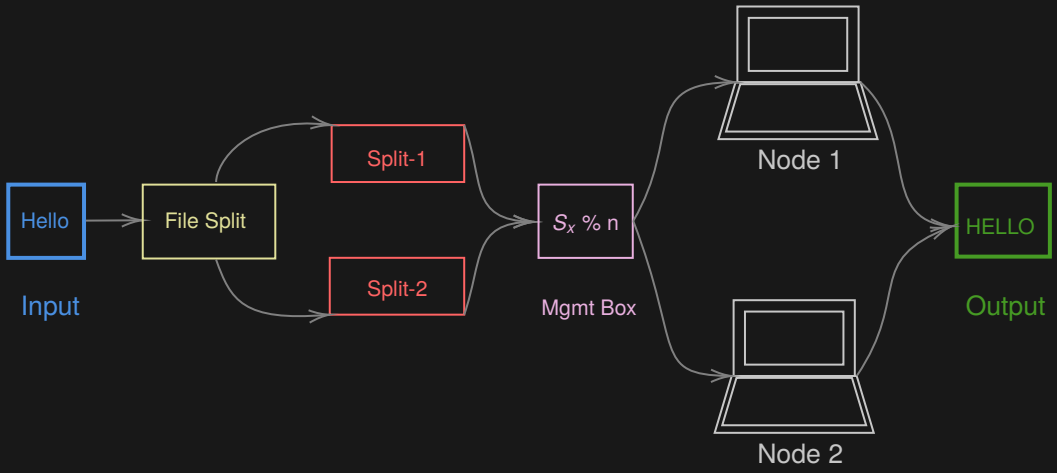
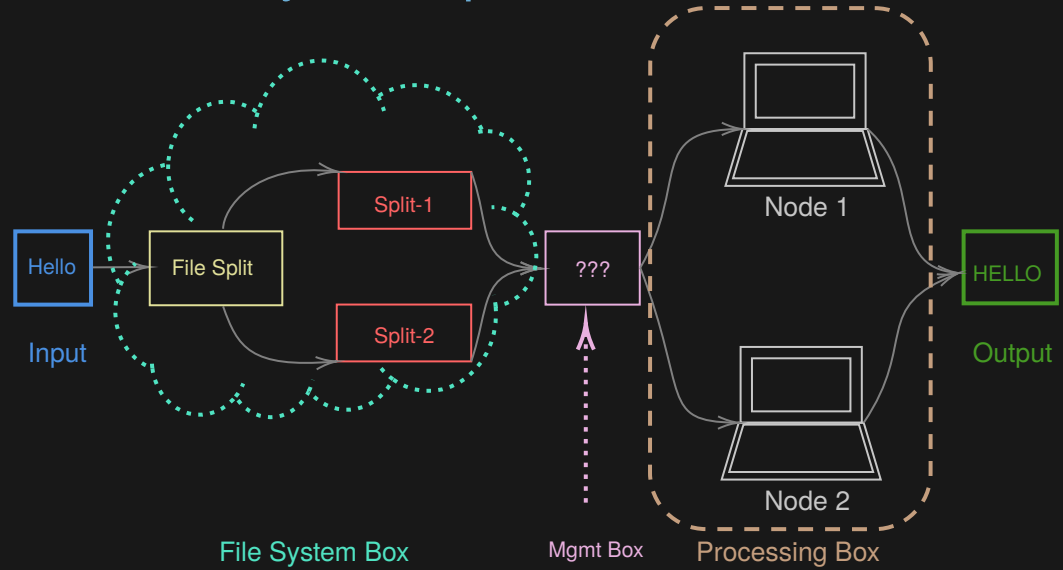


Figure: Convert text to upper text, for example, The -> THE

Case Study Example 1



Case Study Example 2

- Assume we have a file contains **1TB** of text lines, and we need to calculate the word count across the document, for example, The cat came back the very next day -> (the, 2), (cat,1), (came,1), (back, 1), (very, 1), (next, 1), (day, 1).

Case Study Example 2

- Assume we have a file contains **1TB** of text lines, and we need to calculate the word count across the document, for example, The cat came back the very next day -> (the, 2), (cat,1), (came,1), (back, 1), (very, 1), (next, 1), (day, 1).

- One of the distributed architecture solutions for this problem is to use *map-reduce*.

The basic idea of MapReduce

- Assume we need to launch a high-throughput bulk-production sandwich shop.

[1] This example taken from https://reberhardt.com/cs110/summer-2018/lecture-notes/lecture-14/

The basic idea of MapReduce

- Assume we need to launch a high-throughput bulk-production sandwich shop.
- This sandwich has a lot of raw ingredients, and our target is to produce the sandwich as quickly as possible.

[1] This example taken from https://reberhardt.com/cs110/summer-2018/lecture-notes/lecture-14/

The basic idea of MapReduce

- Assume we need to launch a high-throughput bulk-production sandwich shop.
- This sandwich has a lot of raw ingredients, and our target is to produce the sandwich as quickly as possible.
- To make the production very quickly we need to distribute the tasks between the *workers*.

[1] This example taken from https://reberhardt.com/cs110/summer-2018/lecture-notes/lecture-14/

The basic idea of MapReduce

We break this into three stages

- Map.

[1] This example taken from https://reberhardt.com/cs110/summer-2018/lecture-notes/lecture-14/

The basic idea of MapReduce

We break this into three stages

- Map.
- Shuffle/Group (Mapper Intermediates).

[1] This example taken from https://reberhardt.com/cs110/summer-2018/lecture-notes/lecture-14/

The basic idea of MapReduce

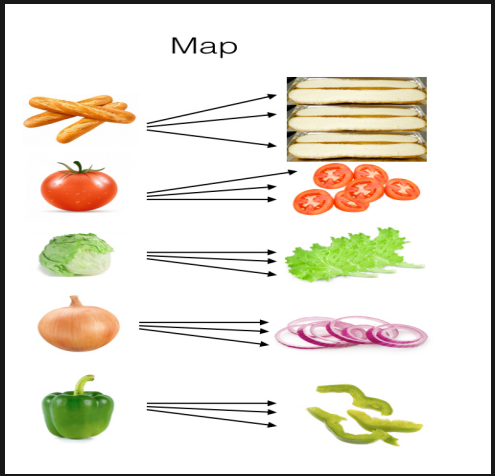
We break this into three stages

- Map.
- Shuffle/Group (Mapper Intermediates).
- Reduce

[1] This example taken from https://reberhardt.com/cs110/summer-2018/lecture-notes/lecture-14/

Map

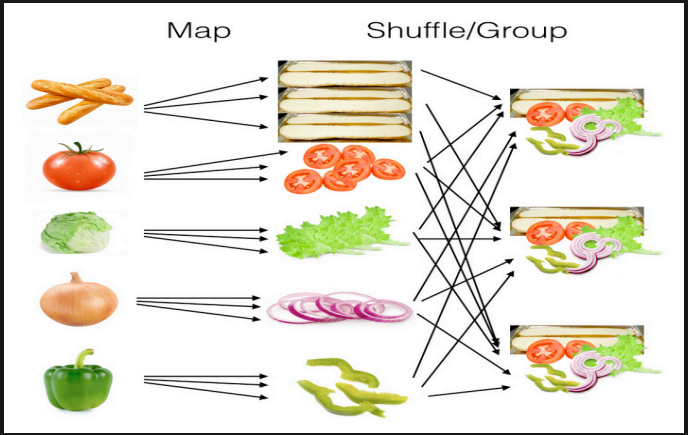
We distribute our raw ingredients amongst the workers.



[1] This example taken from https://reberhardt.com/cs110/summer-2018/lecture-notes/lecture-14/

Shuffle/Group

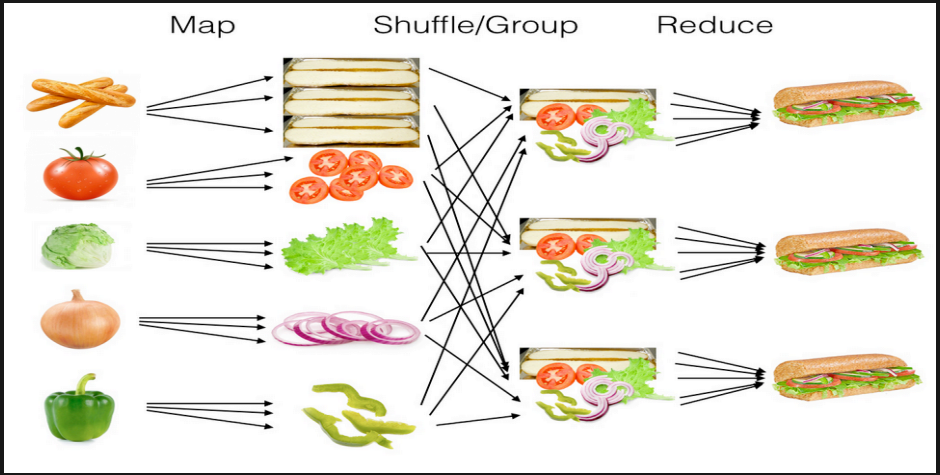
We will organise and group the processed ingredients into piles, so that making a sandwich becomes easy.



[1] This example taken from https://reberhardt.com/cs110/summer-2018/lecture-notes/lecture-14/

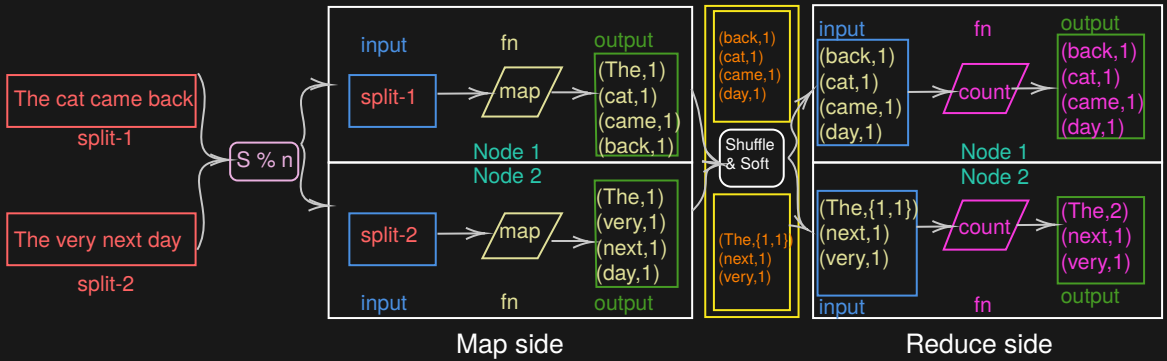
Reduce

we'll combine the ingredients into a sandwich



[1] This example taken from https://reberhardt.com/cs110/summer-2018/lecture-notes/lecture-14/

Case Study Example 2



Thank you for watching!

See you in the next video ☺