بسم الله الرحمن الرحیم

تکنولوژی کامپیوتر

جلسه‌ی بیست و پنجم

پردازش دسته‌ای – شروع هدوپ

جلسه گذشته

کافکا

جلسه جدید

BATCH PROCESSING

پردازش دسته‌ای یعنی چه؟

معیارهای پر اهمیت و تفاوت با OLTP

معیارهای پر اهمیت و تفاوت با OLTP

ETL – Extract, Transform, Load

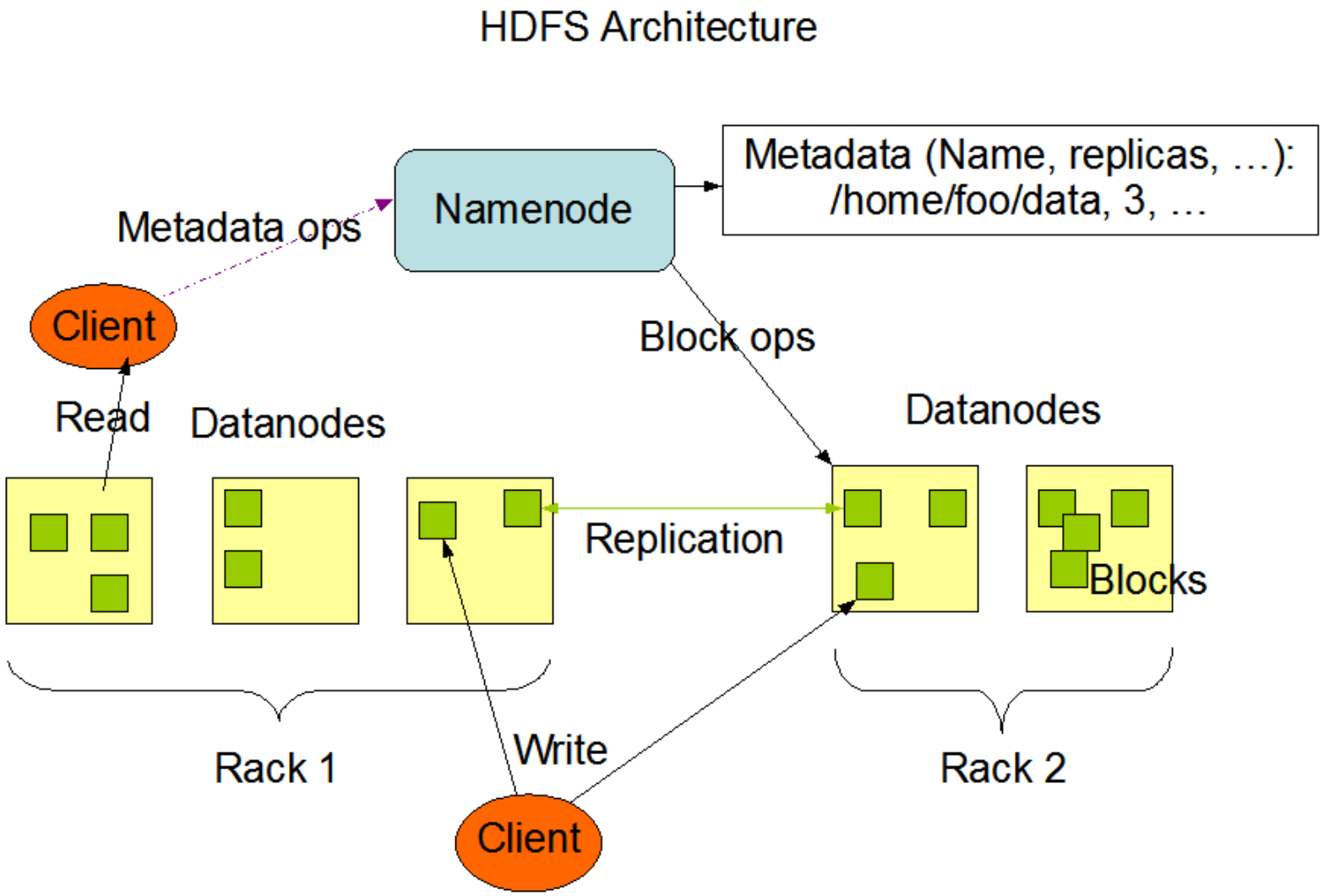
HDFS

- The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware

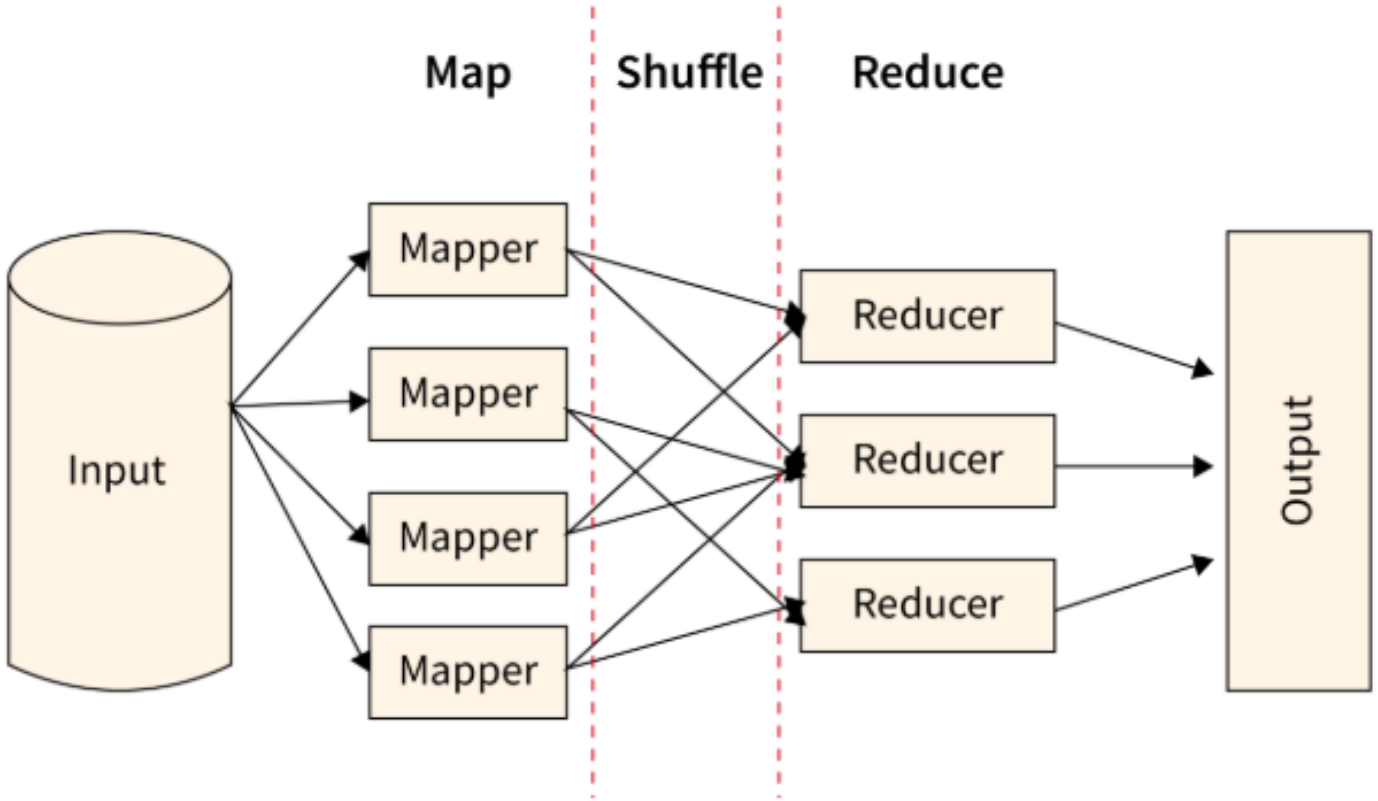
Assumptions and Goals

- Hardware Failure
- Streaming Data Access
- Large Data Sets
- Simple Coherency Model: write-once-read-many
- Moving Computation is Cheaper than Moving Data
- Portability Across Heterogeneous Hardware and Software Platforms

HDFS Components



MAP REDUCE



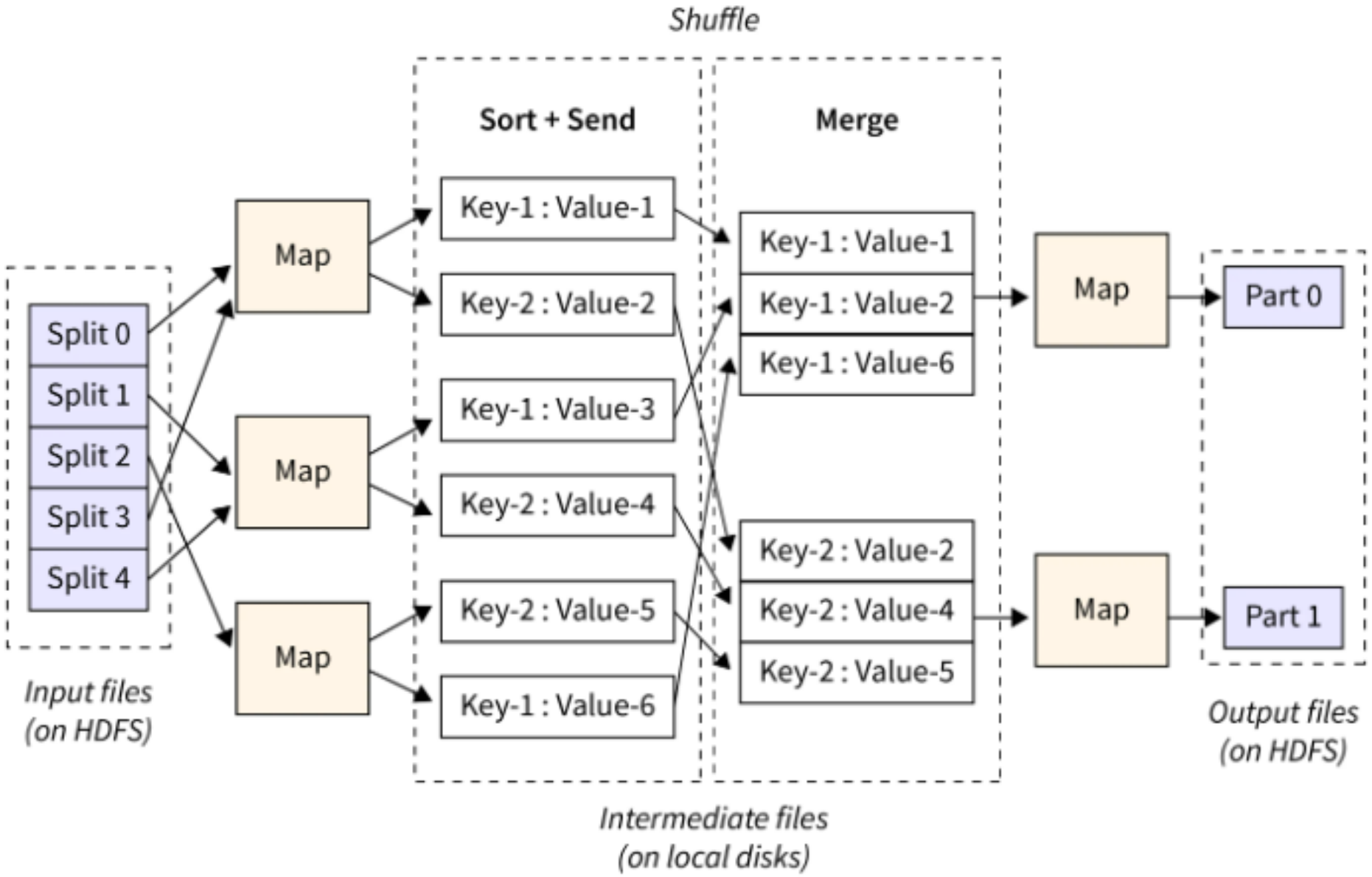
SCALER Topics

Map Stage

- Input data is divided into smaller chunks or blocks.
- Several worker nodes work in parallel to process each chunk independently.
- A "Map" function is applied to each data chunk, generating intermediate key-value pairs.
- The Map function's goal is to extract relevant information from the input data and prepare it for further processing.

Reduce Stage

- After the Map stage, the intermediate key-value pairs are grouped by key.
- The grouped key-value pairs are then shuffled and sorted based on their keys.
- The purpose of the shuffle and sort phase is to bring together all the intermediate values associated with the same key and make them available to the corresponding Reduce function.
- Once the shuffling and sorting are complete, a "Reduce" function is applied to perform aggregation, analysis, or other computations on the grouped data.
- The output is a set of final key-value pairs from the computation.



SCALER Topics

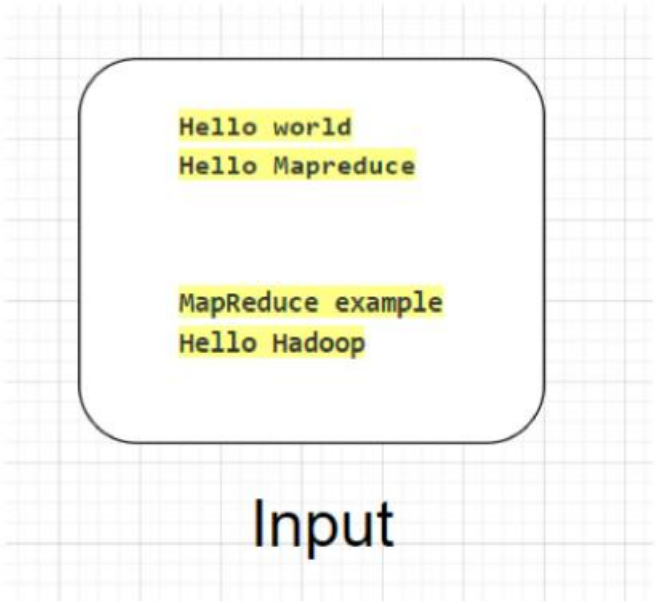
Word Count Example

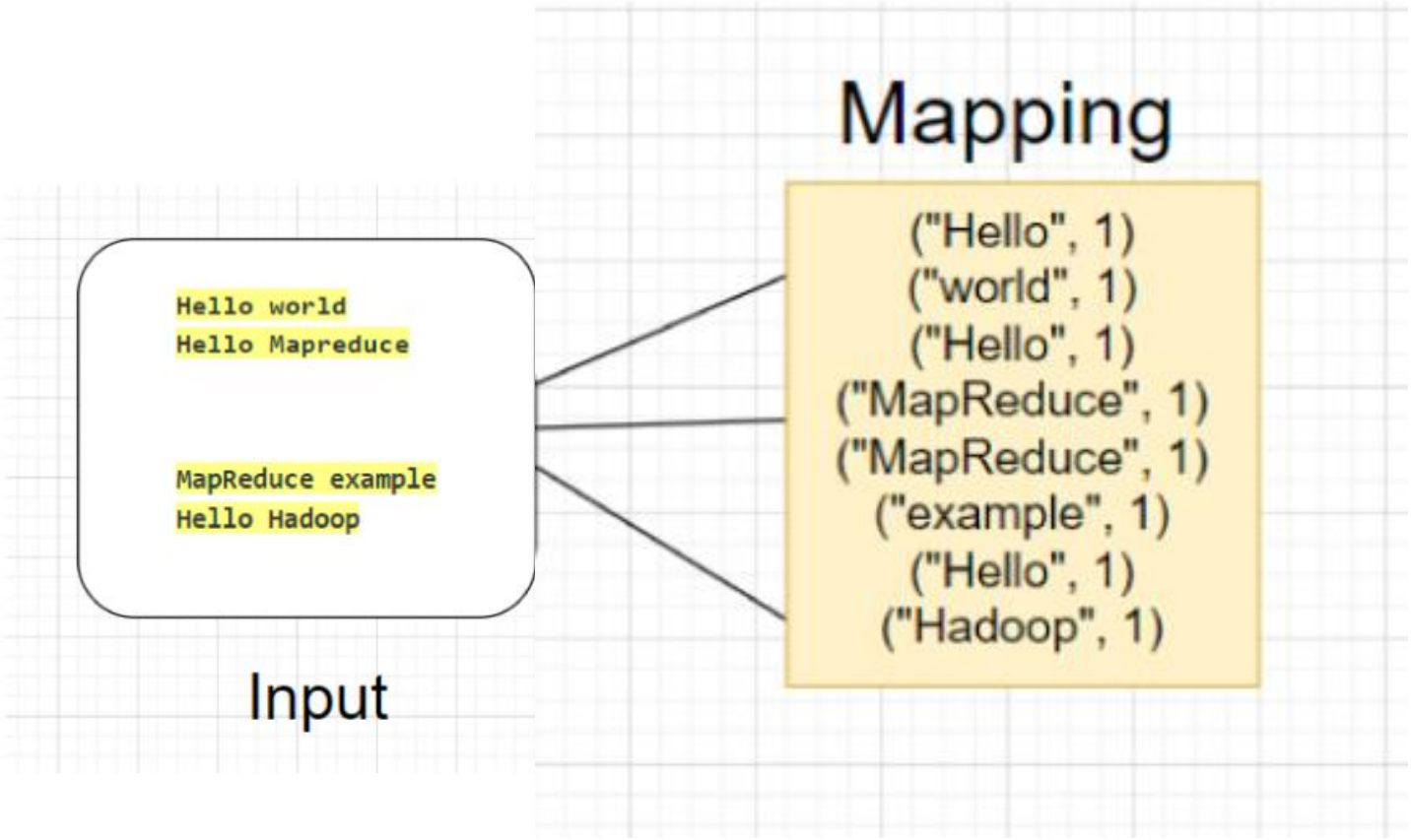
File 1 (document1.txt):

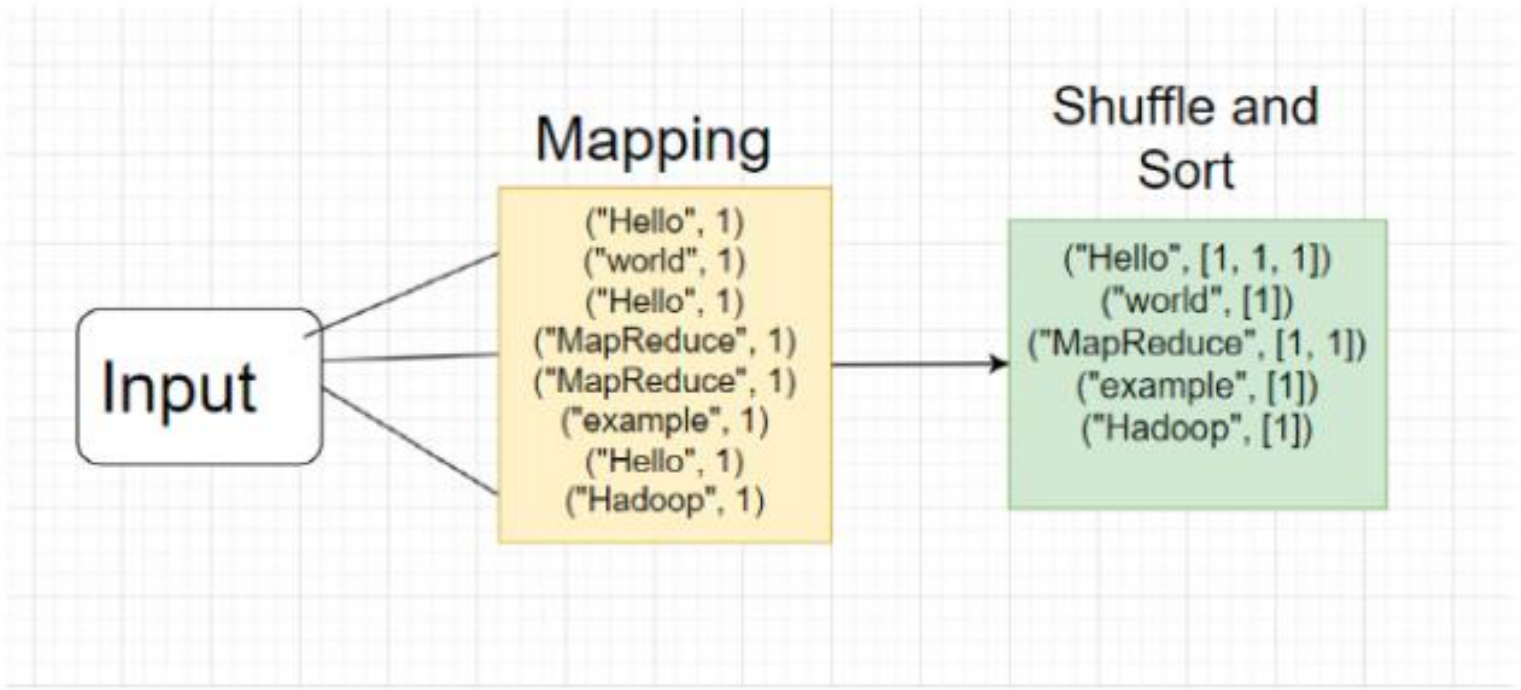
```
Hello world
Hello MapReduce
```

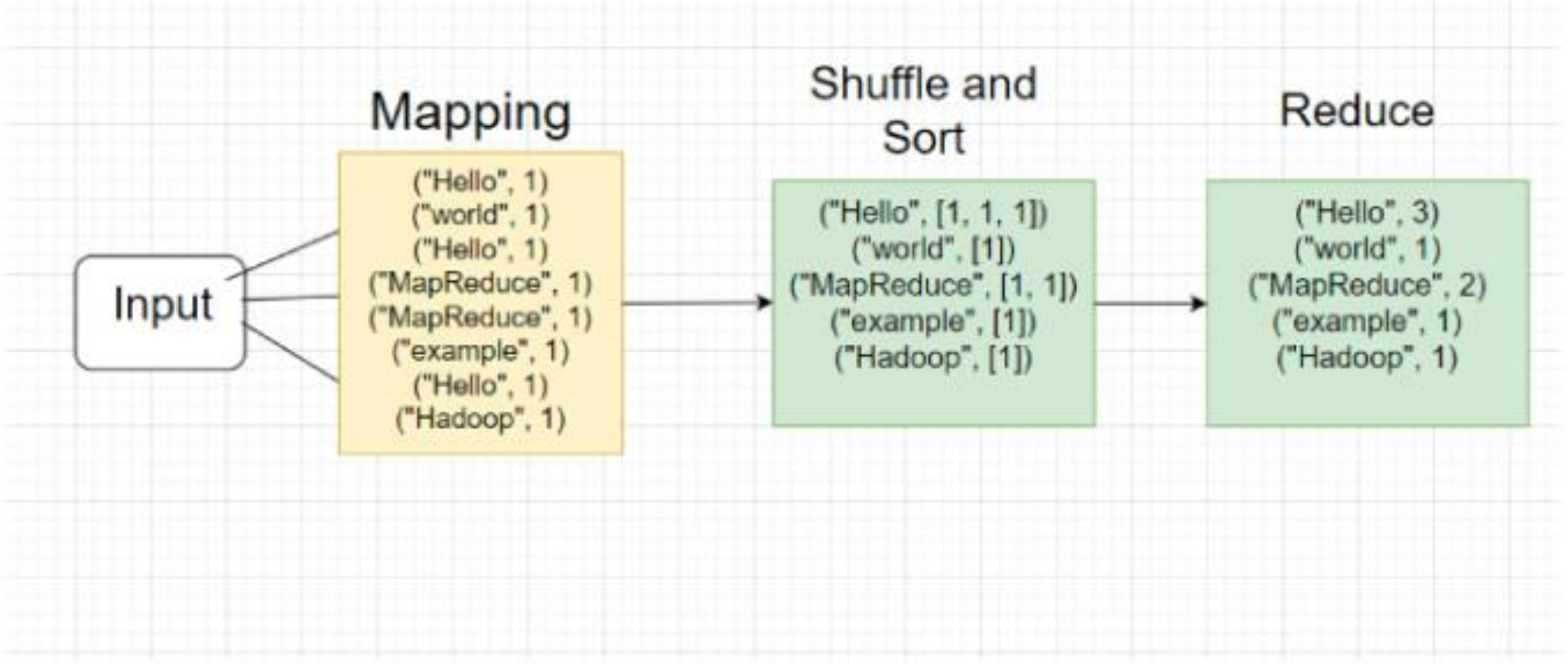
File 2 (document2.txt):

```
MapReduce example
Hello Hadoop
```


Hello world
Hello Mapreduce
MapReduce example
Hello Hadoop
Input


Mapping
Hello world
Hello Mapreduce
MapReduce example
Hello Hadoop
("Hello", 1)
("world", 1)
("Hello", 1)
("MapReduce", 1)
("MapReduce", 1)
("example", 1)
("Hello", 1)
("Hadoop", 1)
Input


Input
Mapping
("Hello", 1)
("world", 1)
("Hello", 1)
("MapReduce", 1)
("MapReduce", 1)
("example", 1)
("Hello", 1)
("Hadoop", 1)
Shuffle and Sort
("Hello", [1, 1, 1])
("world", [1])
("MapReduce", [1, 1])
("example", [1])
("Hadoop", [1])


Input
Mapping
("Hello", 1)
("world", 1)
("Hello", 1)
("MapReduce", 1)
("MapReduce", 1)
("example", 1)
("Hello", 1)
("Hadoop", 1)
Shuffle and Sort
("Hello", [1, 1, 1])
("world", [1])
("MapReduce", [1, 1])
("example", [1])
("Hadoop", [1])
Reduce
("Hello", 3)
("world", 1)
("MapReduce", 2)
("example", 1)
("Hadoop", 1)

Notes

- Typically written in conventional programming language
- Code for mapper and reducer needs to be sent to all systems before job can be run
- Shuffle
- Common to have multiple jobs chained together, typically called a workflow. Various tools designed to manage these, Oozie, Azkaban, Luigi, Airflow, Pinball

Optimization: Combiner

Reduce-Side Joins and Grouping

- Common to have a record of information associated with another record, foreign key in relational model
- In relational DB might use an index, but mapreduce always does a full table scan which is much more expensive
- Only really makes sense to use mapreduce when you are doing it across all users for example, not a join for a single user

Reduce-Side Joins and Grouping

- May need to associate user activity with user profile information
  - *Querying an external data source might be extremely slow especially with the high number of requests put out by mapreduce*
  - *Best to try to promote as much locality as possible*
  - *Better to put a backup of the database in HDFS*

Reduce-Side Joins and Grouping Sort-Merge joins

- MapReduce job can arrange records to be sorted such that the reducer always sees records from user db first followed by activity events in time order, called a secondary sort
- Reducer called once for every user ID and has the first value be the date-of-birth record and then iterate over later activity outputting pairs of viewed url and viewer age in years

Handling skew

Map-Side Joins

Map-side joins - Broadcast hash joins

Map-side joins – Partitioned hash joins

OUTPUT OF BATCH WORKFLOWS

What happen if it is not succeed?

یک مثال که MapReduce چندان هم خوب نیست.

YARN

در عمل