MUSIC

&

BIG DATA

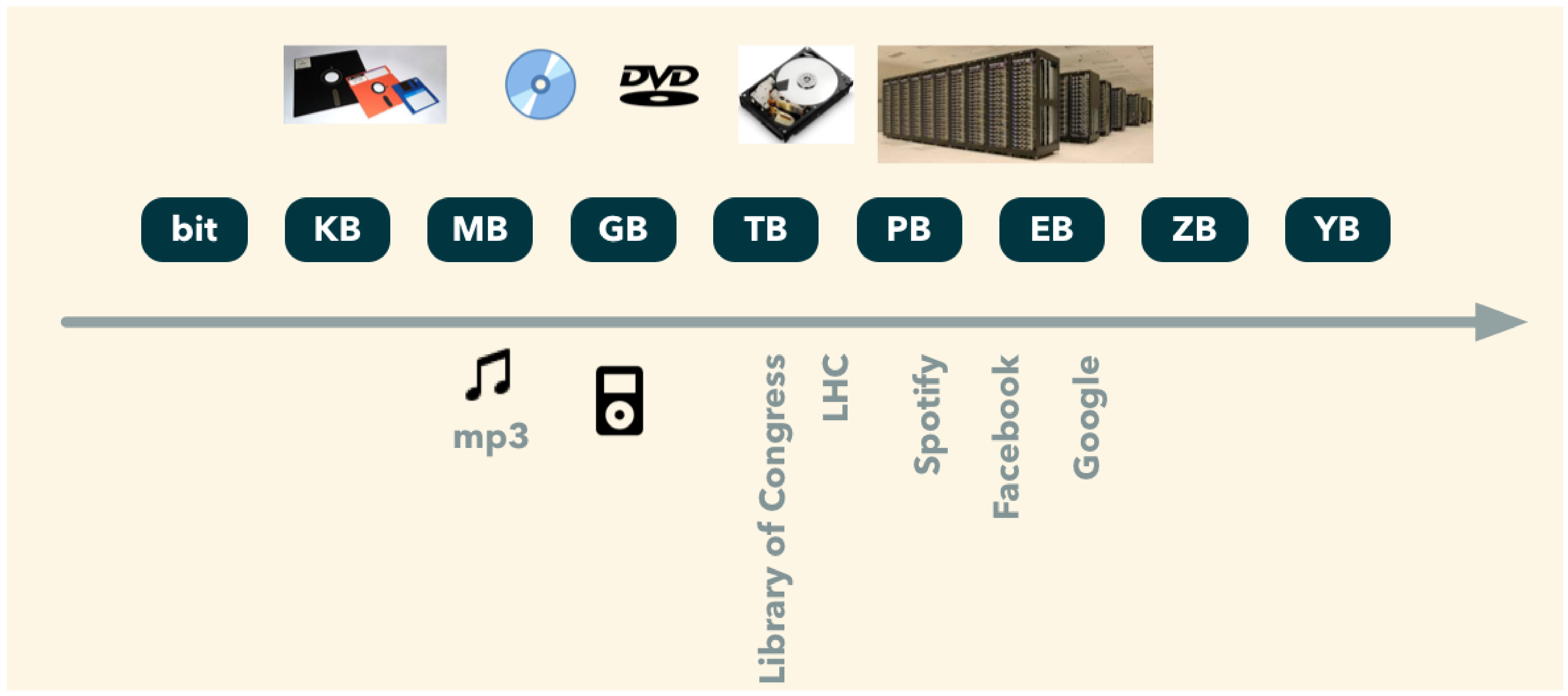
BIG DATA

"describes large amount of data (structured or unstructured) that are difficult to process using traditional database and software"

BIG DATA

"Big data usually includes data sets with sizes beyond the ability of commonly used software tools to capture, curate, manage, and process data within a tolerable elapsed time."

BIG DATA



bit
KB
MB
GB
TB
PB
EB
ZB
YB
mp3
Library of Congress
LHC
Spotify
Facebook
Google

BIG DATA - THE 3 VS... (AND MORE)

VOLUME

- large data sets
- data are not sampled

VELOCITY

- rapidly changing
- available in real-time

VARIETY

- different type: text, images, audio, video, ...
- (un)structured: JSON, XML / images, audio, music

... AND MORE VS

- **Veracity**
  - how much trust can be put in the data
- **Value**
  - eventually drives revenues or new features for companies
- **Variability**
  - no fixed data or schema
  - evolution in time

BIG DATA IN MUSIC

-

WHERE ?

CONTENT

- Audio
- Lyrics
- Metadata

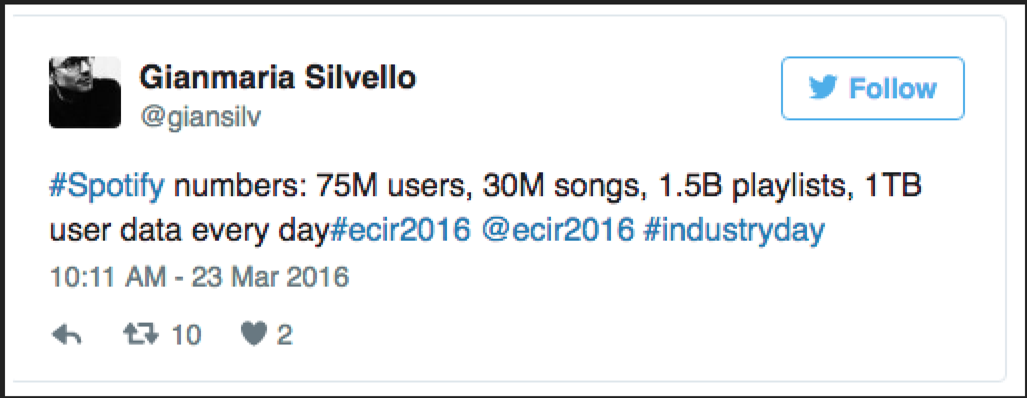
EVENTS

- Listening patterns
- Application events
- User activity
- Social media data

DERIVED DATA

- Crowd sourced data
- Recommendations
- Playlists
- User content

EXAMPLE: BIG DATA @ SPOTIFY



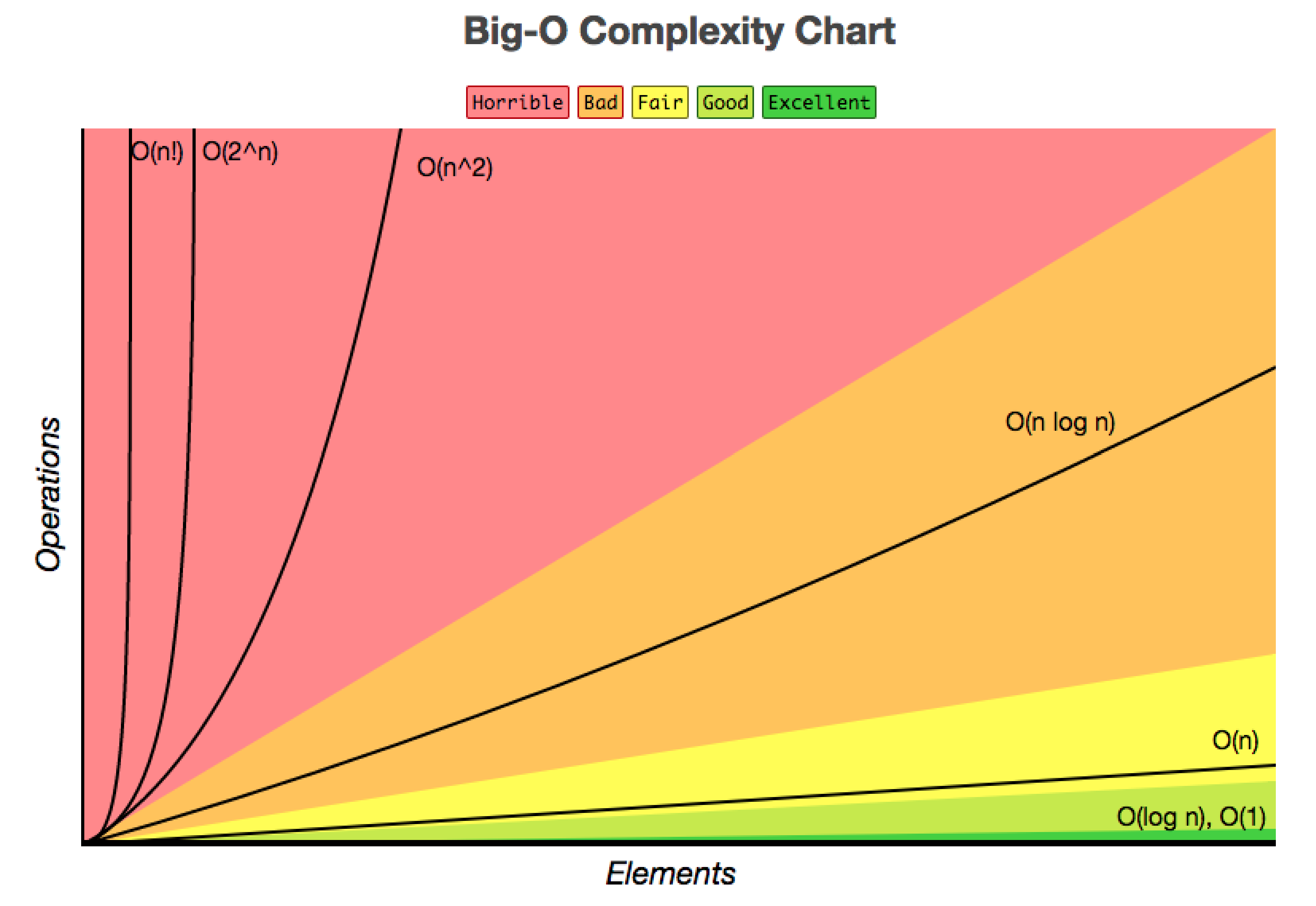
- 42PB Storage
- 200TB data generated / day
- 1300 Servers

HOW TO SCALE?

ALGORITHMS & DATA STRUCTURES

- Algorithmic & Complexity
- Data Structures

COMPLEXITY


Big-O Complexity Chart
Horrible
Bad
Fair
Good
Excellent
O(n!)
O(2^n)
O(n^2)
O(n log n)
O(n)
O(log n), O(1)
Operations
Elements

DATA STRUCTURES

Common Data Structure Operations

Data Structure	Time Complexity								Space Complexity
	Average				Worst				Worst
	Access	Search	Insertion	Deletion	Access	Search	Insertion	Deletion	
Array	Θ(1)	Θ(n)	Θ(n)	Θ(n)	O(1)	O(n)	O(n)	O(n)	O(n)
Stack	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Queue	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Singly-Linked List	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Doubly-Linked List	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Skip List	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(n)	O(n)	O(n)	O(n)	O(n log(n))
Hash Table	N/A	Θ(1)	Θ(1)	Θ(1)	N/A	O(n)	O(n)	O(n)	O(n)
Binary Search Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(n)	O(n)	O(n)	O(n)	O(n)
Cartesian Tree	N/A	Θ(log(n))	Θ(log(n))	Θ(log(n))	N/A	O(n)	O(n)	O(n)	O(n)
B-Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)
Red-Black Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)
Splay Tree	N/A	Θ(log(n))	Θ(log(n))	Θ(log(n))	N/A	O(log(n))	O(log(n))	O(log(n))	O(n)
AVL Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)
KD Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(n)	O(n)	O(n)	O(n)	O(n)

PROGRAM OPTIMIZATION

- CPU
- Memory
- IO
- Network

PARALLELISM

- Parallelism
  - Multithreading
  - Multiprocessing

VERTICAL SCALING

- More CPU
- More Memory
- More Storage

NEW PARADIGMS

- Dedicated hardware - bespoke designs
  - GPU
  - FPGA (Field Programmable Gate Array)
- New paradigm
  - DNA Computing
  - Quantum Computing

HORIZONTAL SCALING

- Clusters
- Sharding
- Share Nothing
- Distributed Systems

BIG DATA & HADOOP

- Fundations
  - Google File System (2003)
  - Google MapReduce (2004)
  - Google BigTable (2005/2006)
- Open Source implementation
  - Apache Nutch (web crawler)
  - Development moved to the Hadoop project in 2006

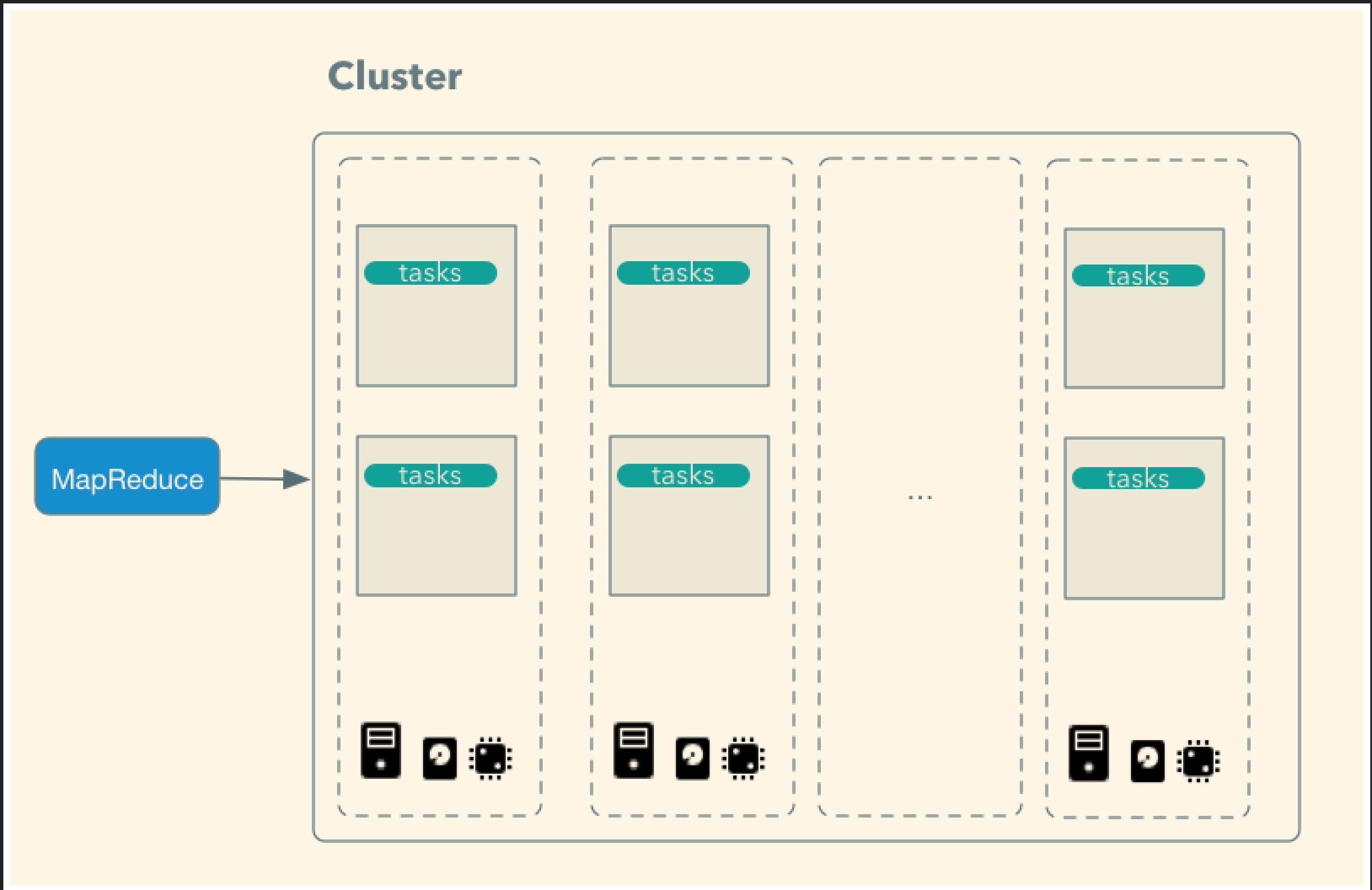
MAPREDUCE

A programming model for processing and generating large data sets with a parallel, distributed algorithm on a cluster

Takes advantage of the locality of data, processing it near the place it is stored in order to reduce the distance over which it must be transmitted.

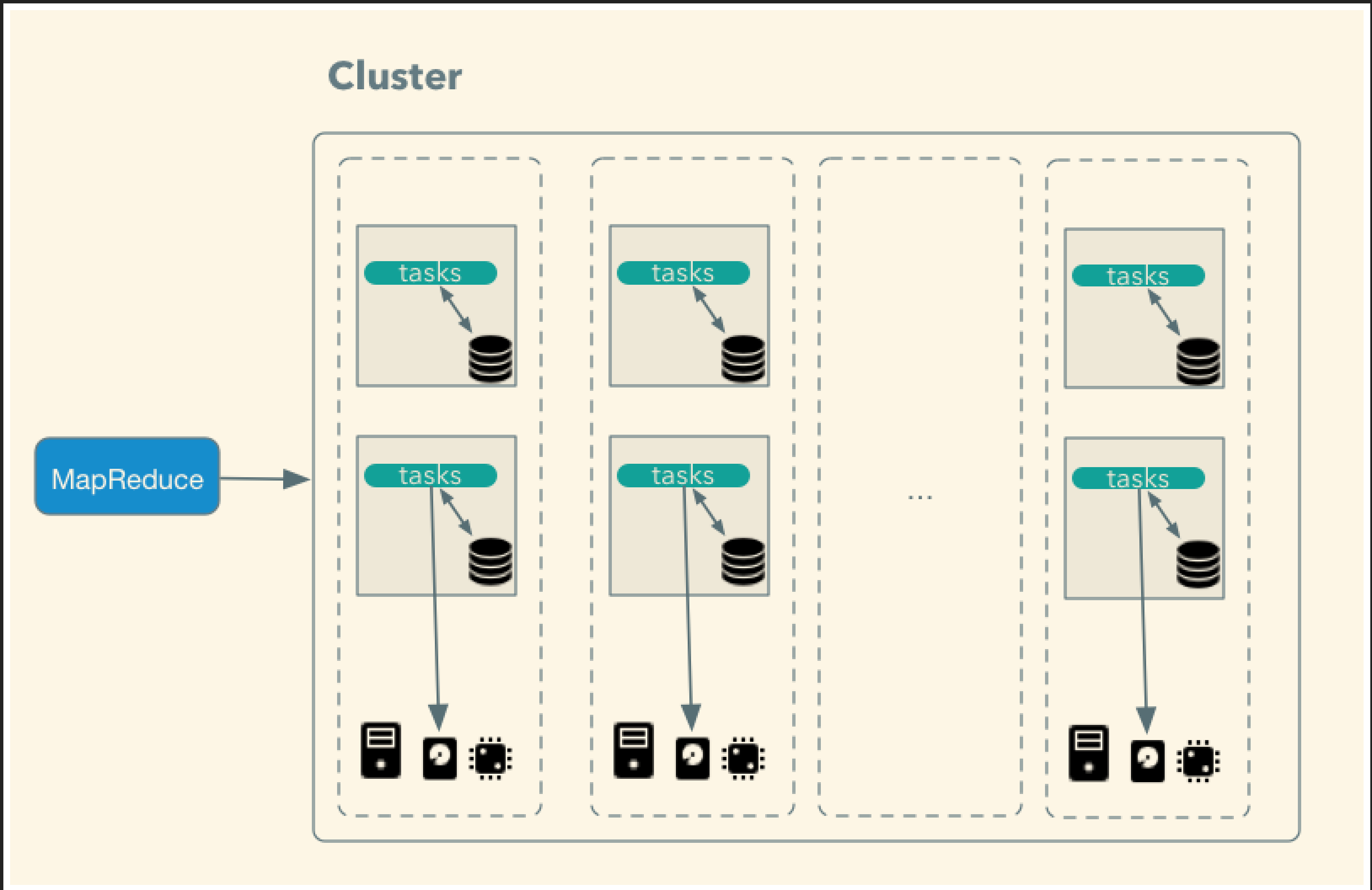
MAPREDUCE

Parallel computations on a cluster

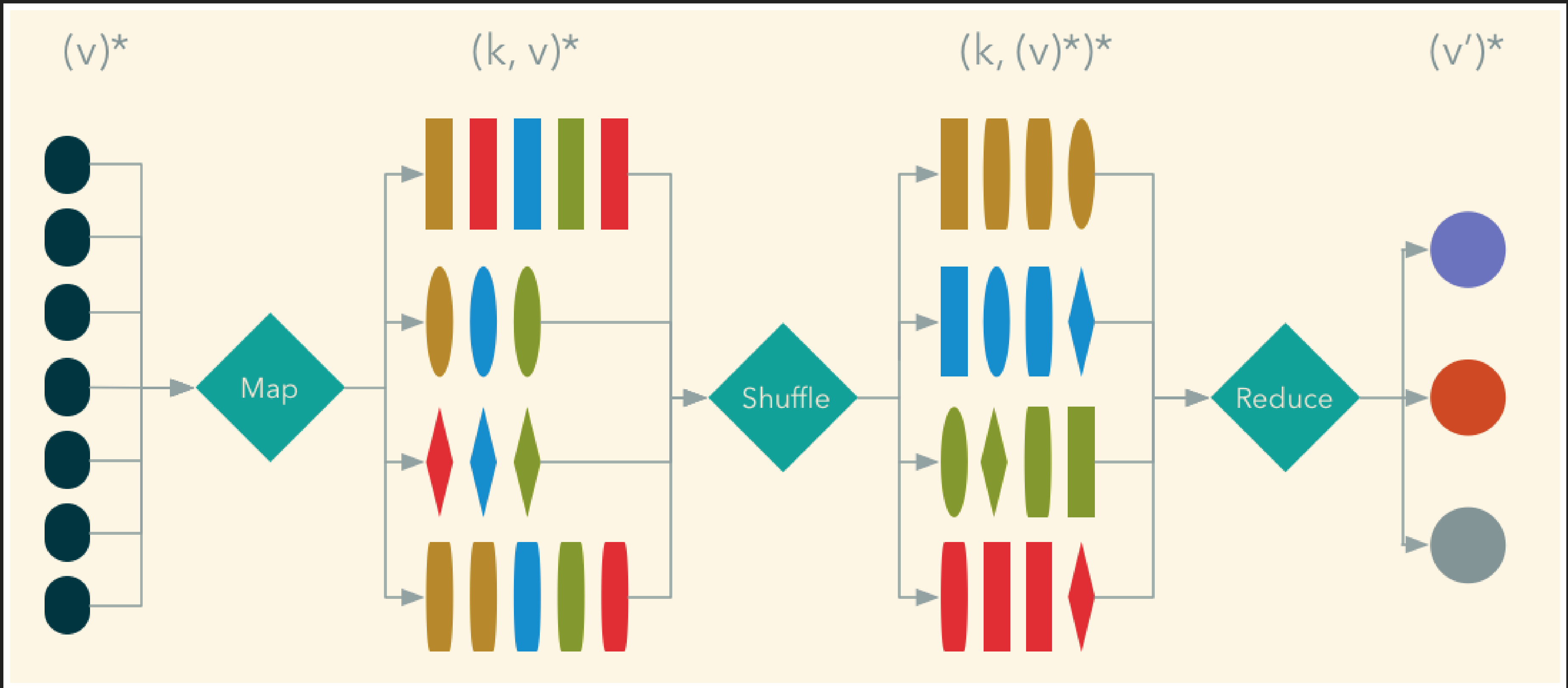


MAPREDUCE

Data locality



MAPREDUCE

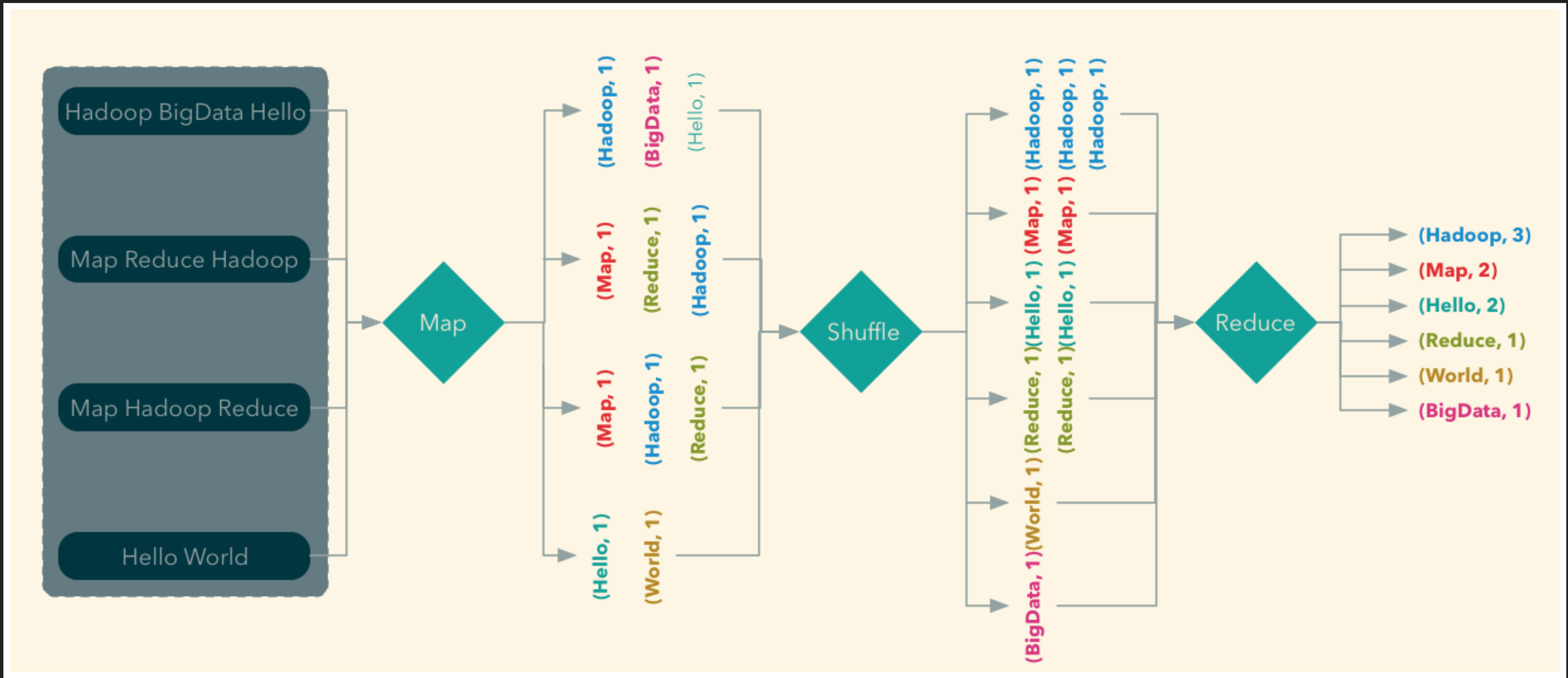


MAPREDUCE - WORD COUNT

```
def map(document):
    for word in document:
        emit(word, 1)

def reduce(word, values):
    count = 0
    for value in values:
         count += value
    emit(word, count)
```

MAPREDUCE - WORD COUNT



MAPREDUCE - WHAT NOW?

Relatively simple computational model

but

Many problems can be translated to it!

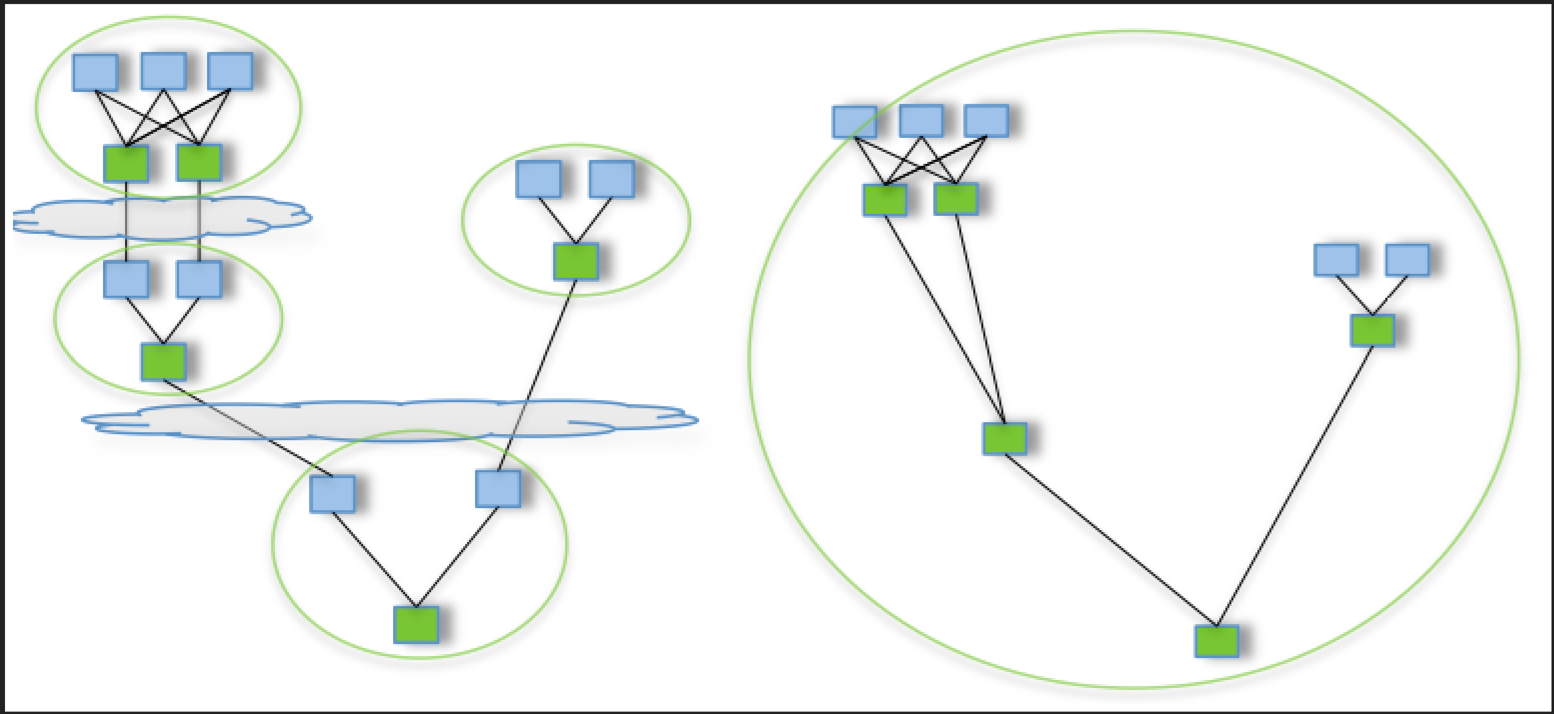
- SQL
- ETL (Extract / Transform / Load)
- Machine Learning
- Bespoke analysis
- ...

MAPREDUCE - LIMITATIONS

- MapReduce tasks independent from each others
- Network/IO intensive in some cases (Shuffle)
- Lack of iterative/in-memory computation

NEW FRAMEWORKS - DAG

Direct Acyclic Graph



NEW FRAMEWORKS - DAG

- Generalization of MapReduce concept
- Jobs are aware of all the tasks involved
- Allows global optimization
- Better use of resources

Implementations:

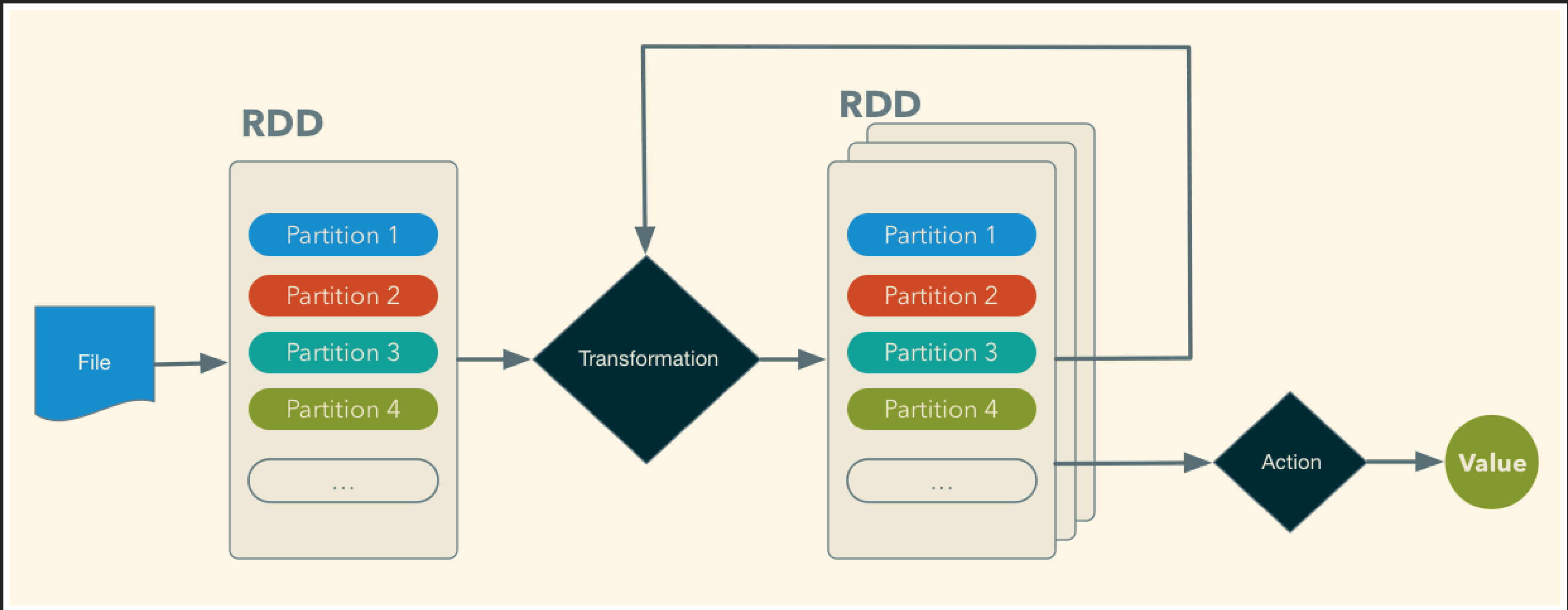
- Dremel, Spark, Tez, Drill, ...

BIG DATA & SPARK

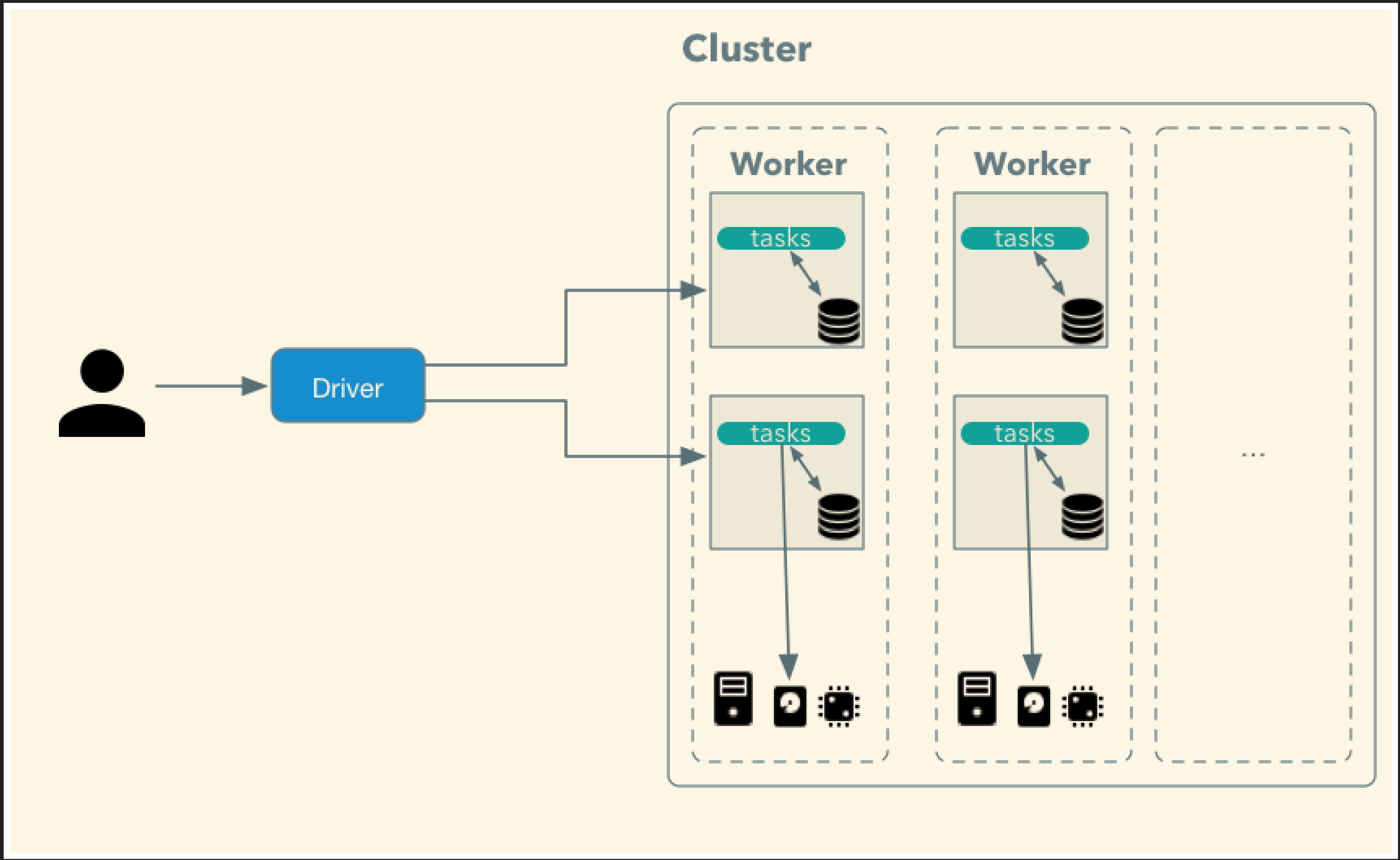
- Fundations
  - Berkeley's AMPLab from 2009
  - Open sourced and moved as an Apache project in 2013
- Improvements on the MapReduce paradigm
  - In memory cluster computing
  - Iterative algorithms
  - Interactive & Exploratory analysis
  - Batch & Streaming

SPARK - RDD (RESILIENT DISTRIBUTED DATASETS)

a fault-tolerant collection of elements that can be operated on in parallel



SPARK - DRIVER & WORKERS



SPARK - HIGH LEVEL LIBRARIES

- SQL
- Streaming
- Machine Learning
- Graph

DEMO!

QUESTIONS?

LAB !