BitTiger

讲座群

讲座负责人

BITTIGER

MapReduce

Agenda

What is MapReduce

Why MapReduce was created

The Job: Indexing the web

The programming model

MapReduce and Hadoop

MapReduce v1, v2 & Spark

Q&A

What is MapReduce

a programming model and

an associated implementation for processing and generating large data sets

with a parallel, distributed algorithm on a cluster.

What is MapReduce

a programming model and

an associated implementation for processing and generating large data sets

with a parallel, distributed algorithm on a cluster.

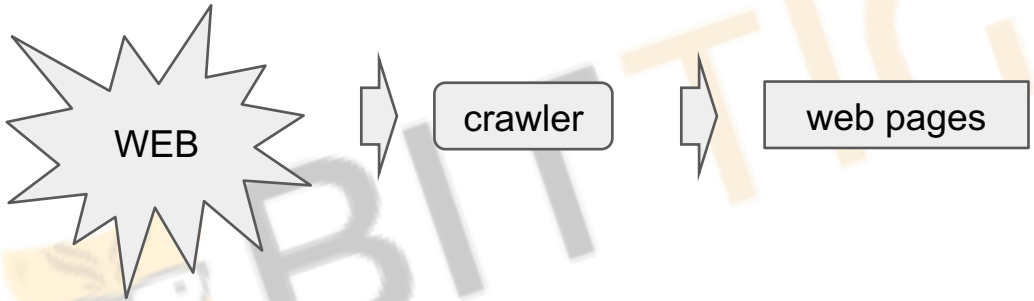
Too abstract!

What does it do ?

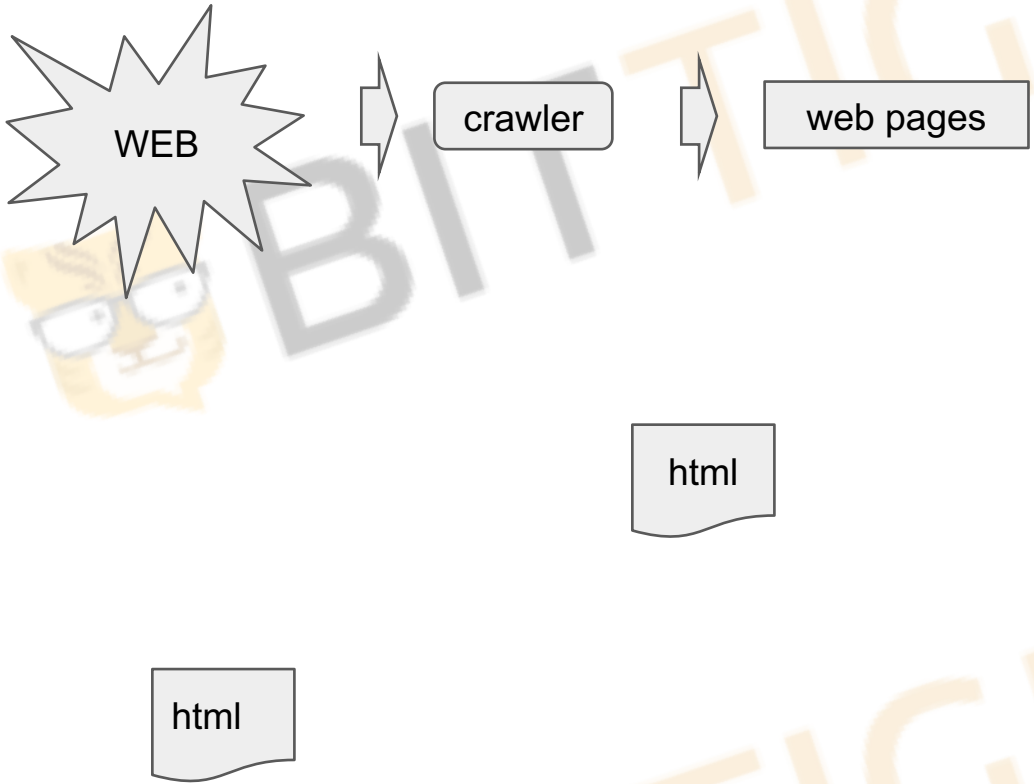
Indexing the web

Crawler + Indexer

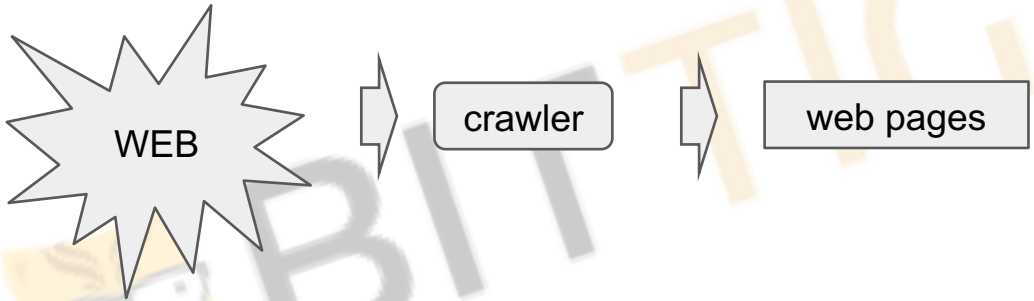
Crawler → traverse

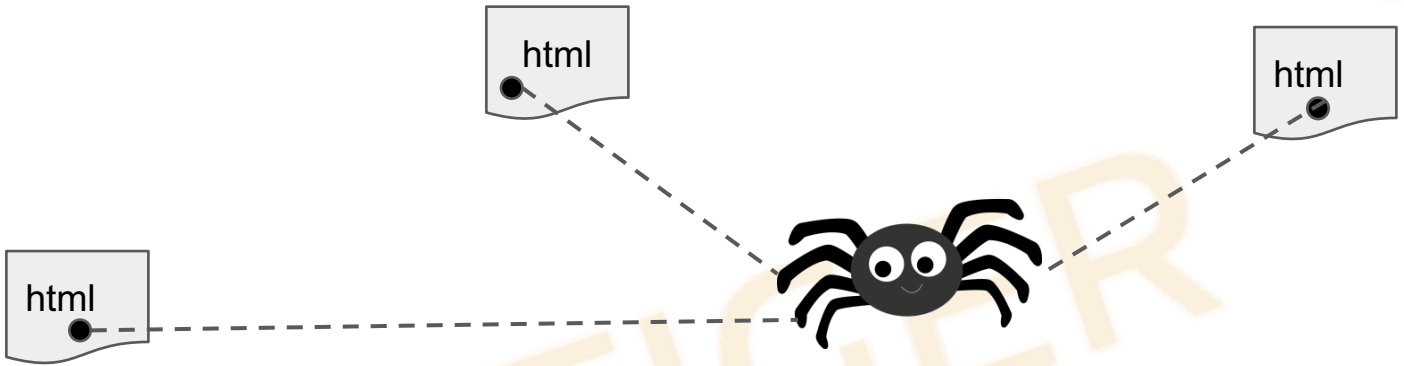


Crawler → traverse

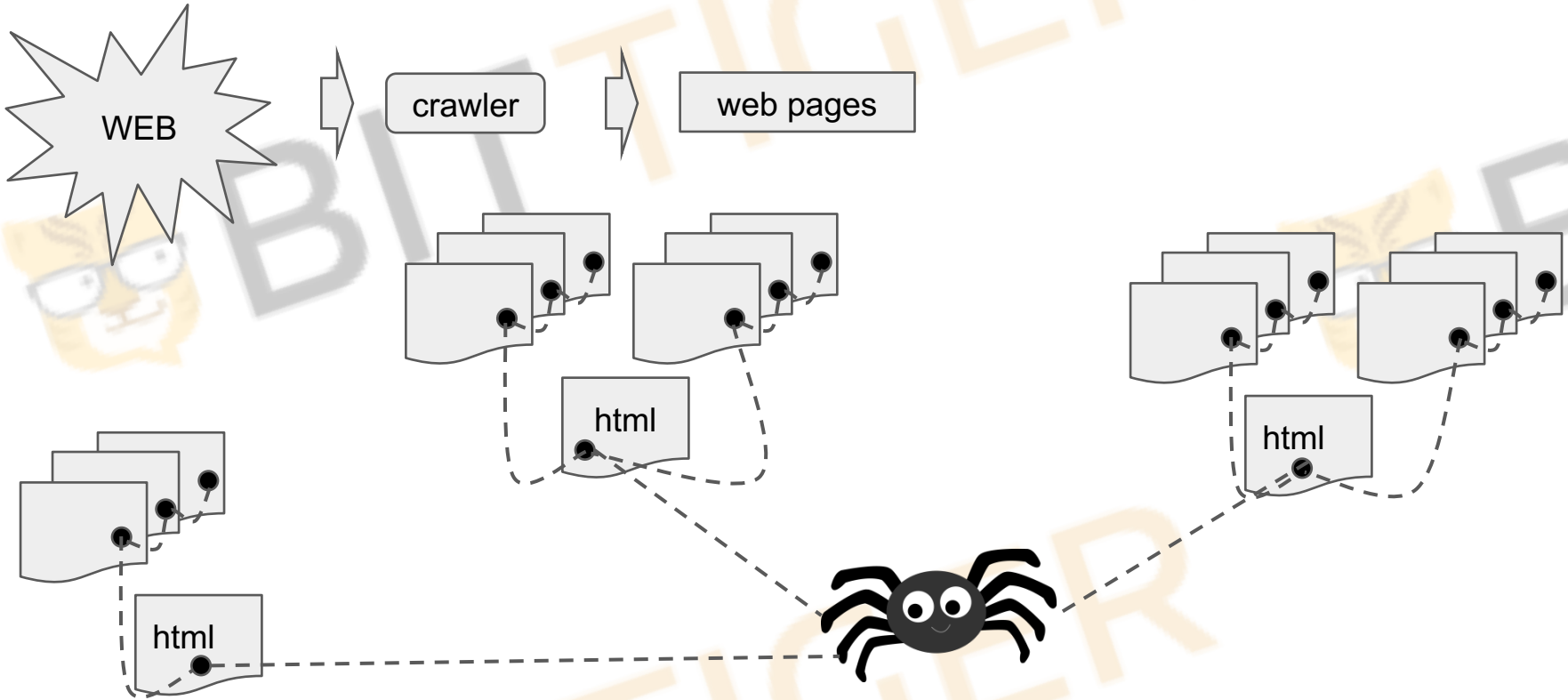


Crawler → traverse

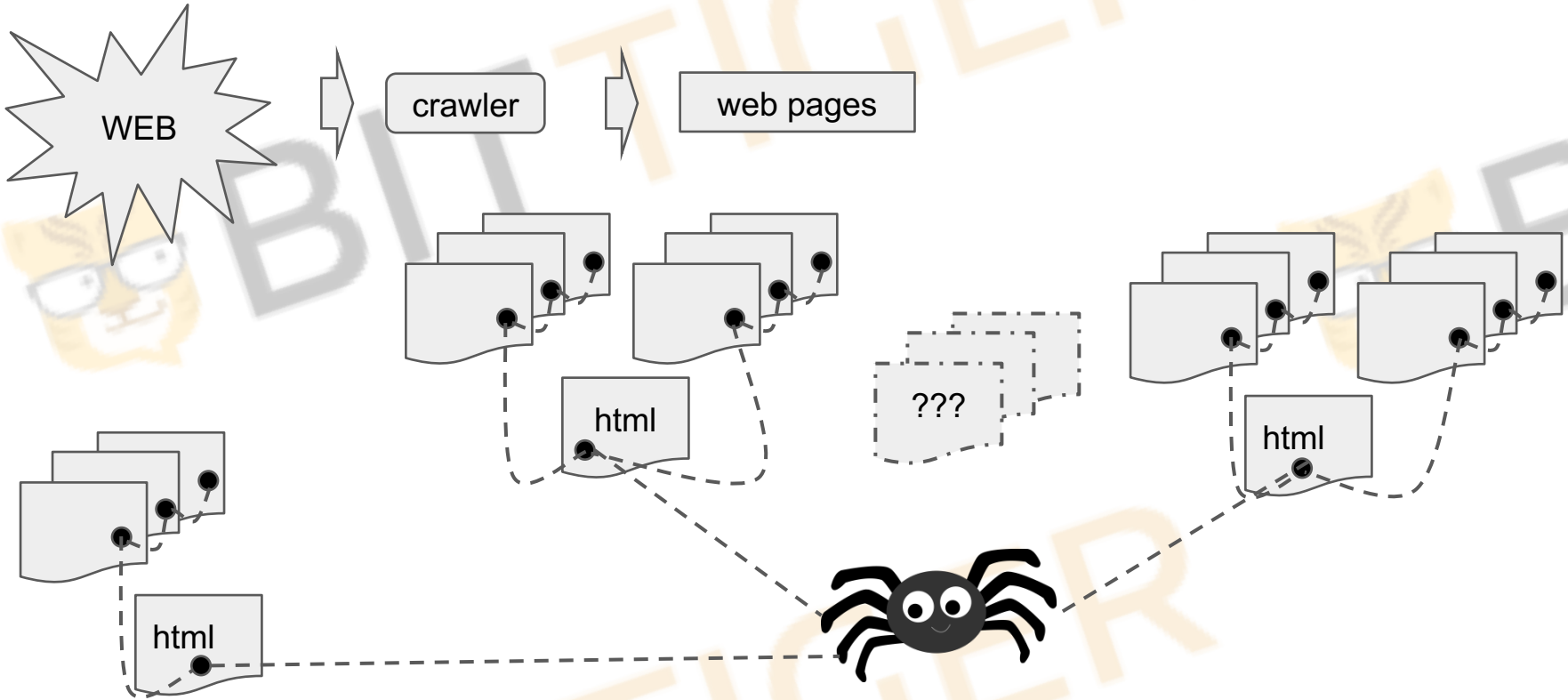




Crawler → traverse

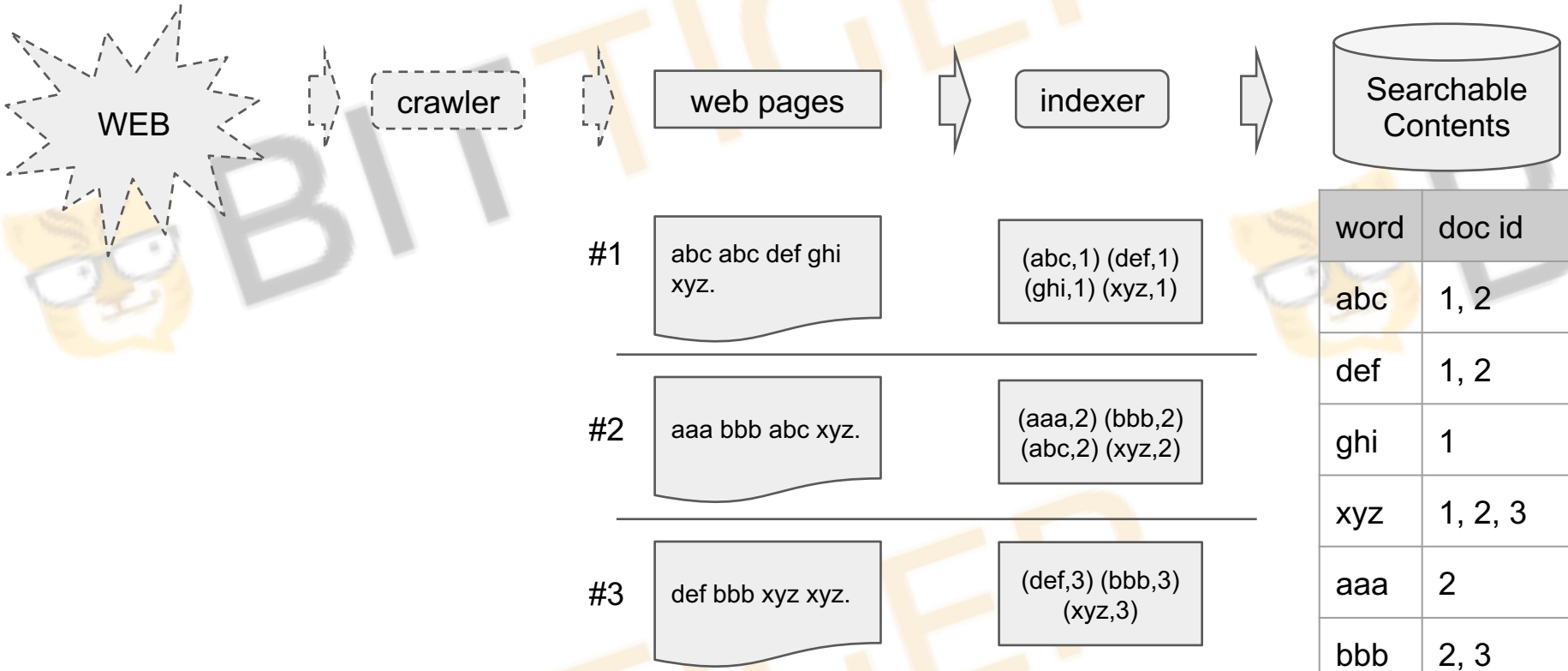


Crawler → traverse



Indexing the web: building index

Indexing the web: building index



word	doc id
abc	1, 2
def	1, 2
ghi	1
xyz	1, 2, 3
aaa	2
bbb	2, 3

Indexing the web: searching

query:
abc bbb

word	doc id
abc	1, 2
def	1, 2
ghi	1
xyz	1, 2, 3
aaa	2
bbb	2, 3

abc: ?

bbb: ?

Indexing the web: searching

query:
abc bbb

word	doc id
abc	1, 2
def	1, 2
ghi	1
xyz	1, 2, 3
aaa	2
bbb	2, 3

abc: 1, 2

bbb: 2, 3

doc id	relevance
2	100%

Indexing the web: searching

query:
abc bbb

word	doc id
abc	1, 2
def	1, 2
ghi	1
xyz	1, 2, 3
aaa	2
bbb	2, 3

abc: 1, 2

bbb: 2, 3

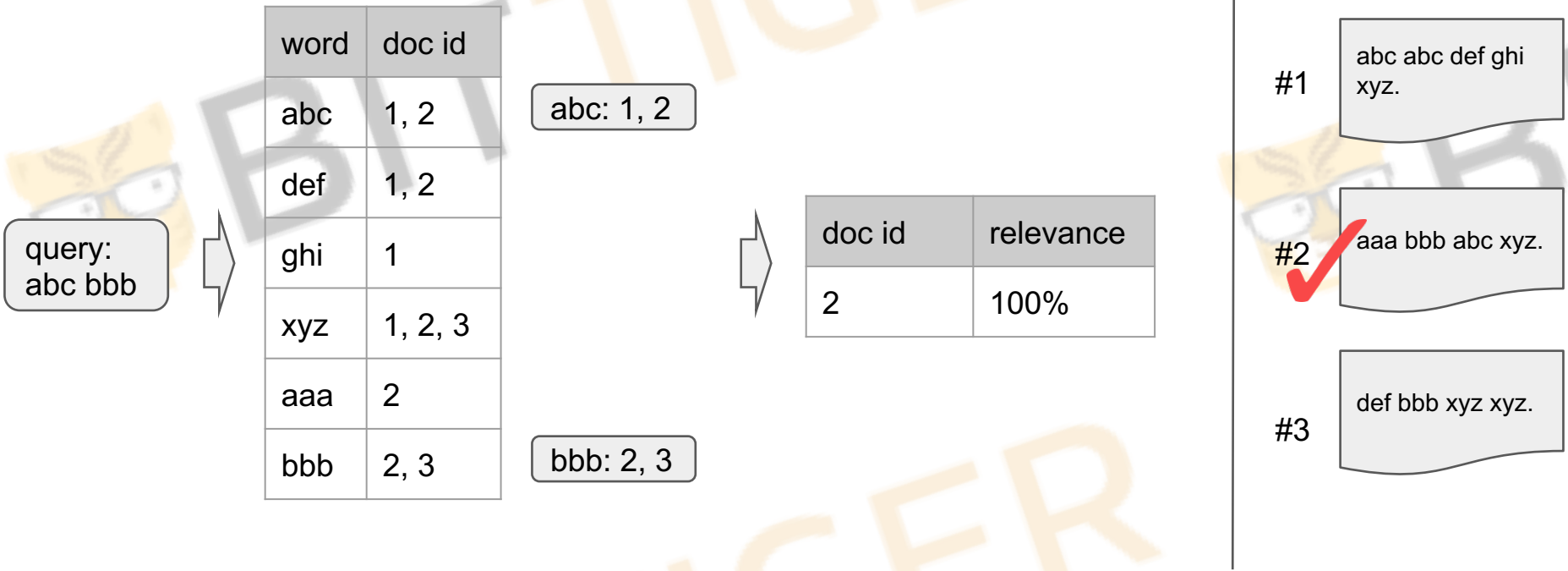
doc id	relevance
2	100%

#1 abc abc def ghi xyz.

#2 aaa bbb abc xyz.

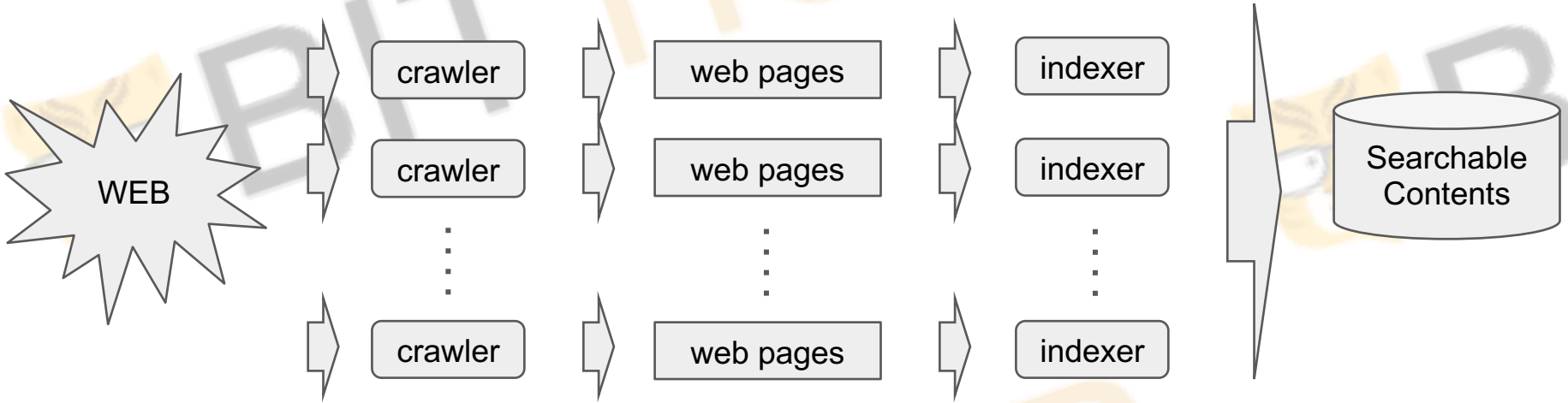
#3 def bbb xyz xyz.

Indexing the web: searching

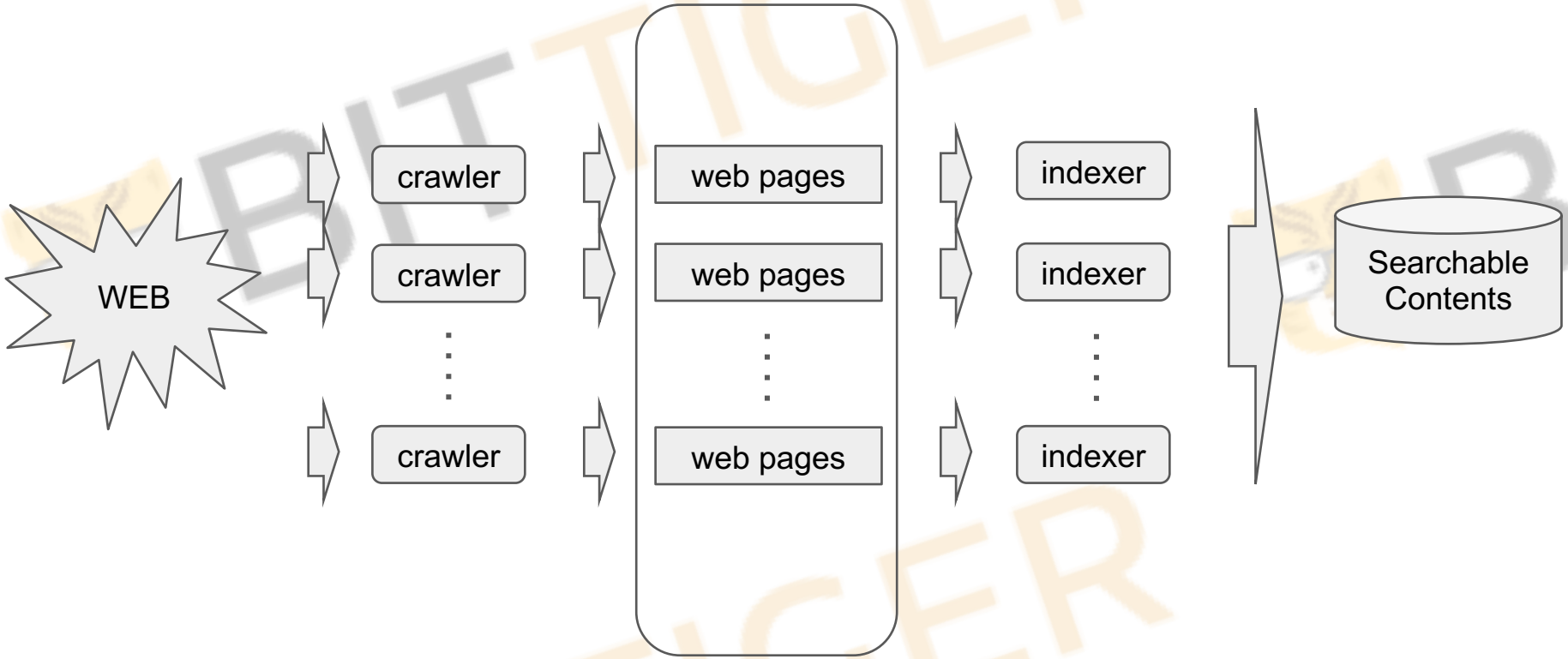


*Inverted index is commonly saved as sorted list

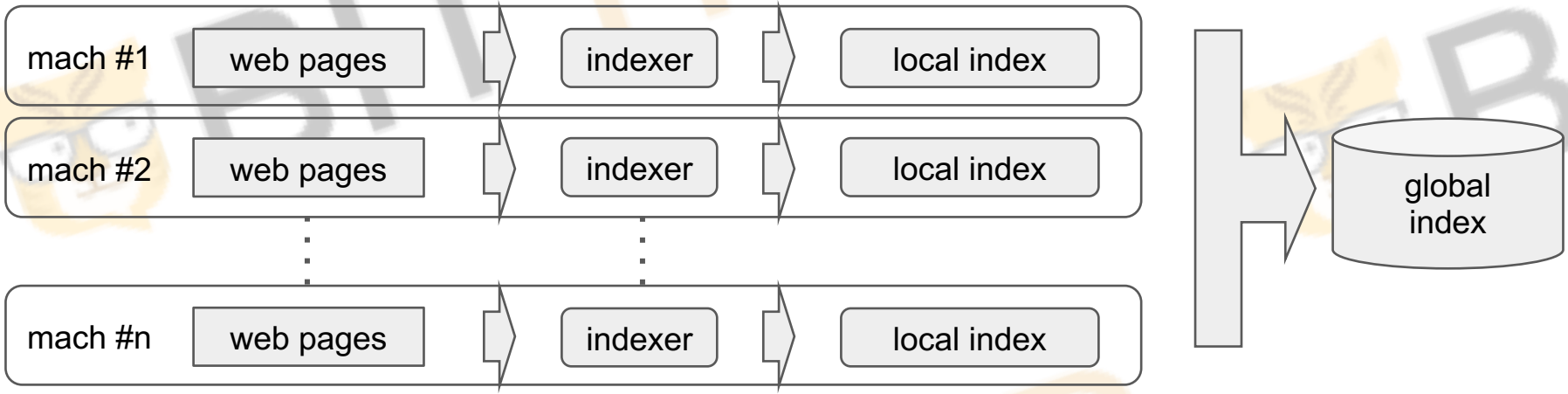
Indexing the web: parallel



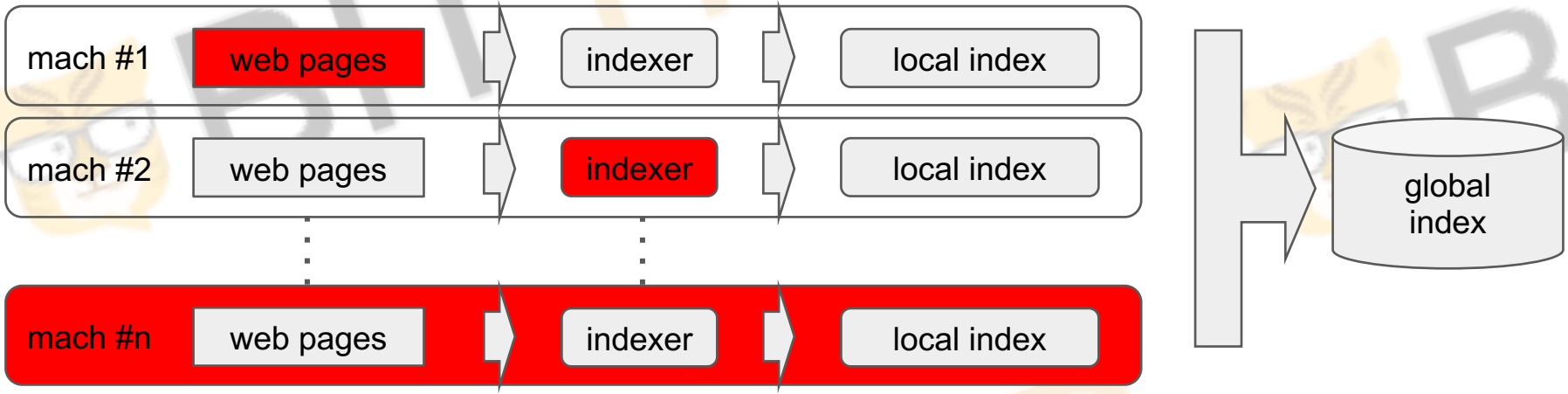
Indexing the web: parallel and separation



Parallel processing



Parallel processing - problems



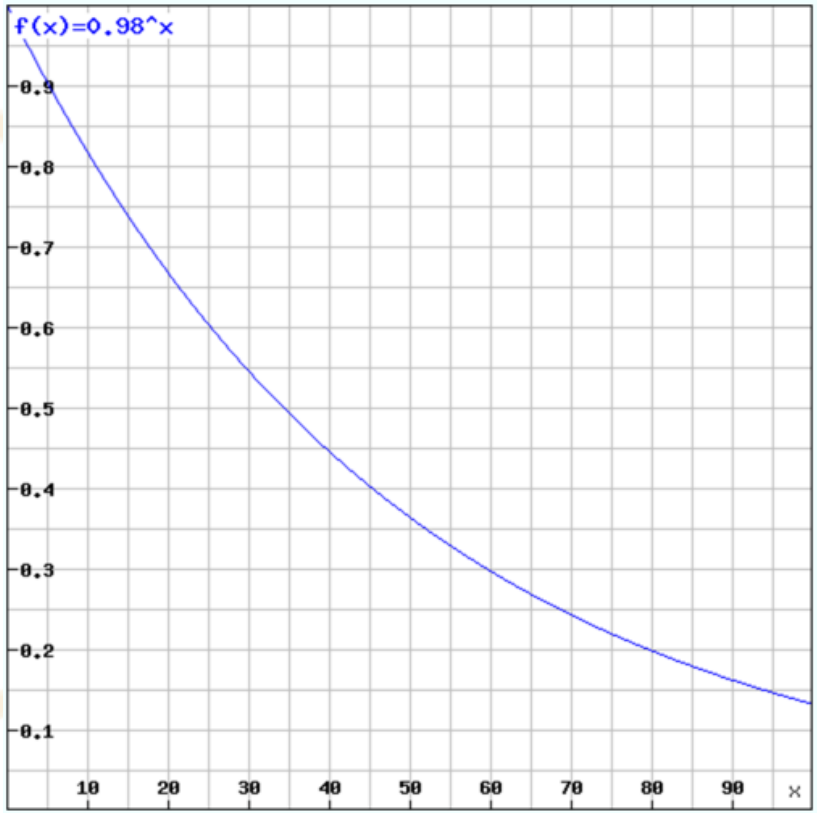
More machines, more problems

Assume: single machine 98% reliable

- Reliability decays exponentially

Source of failures

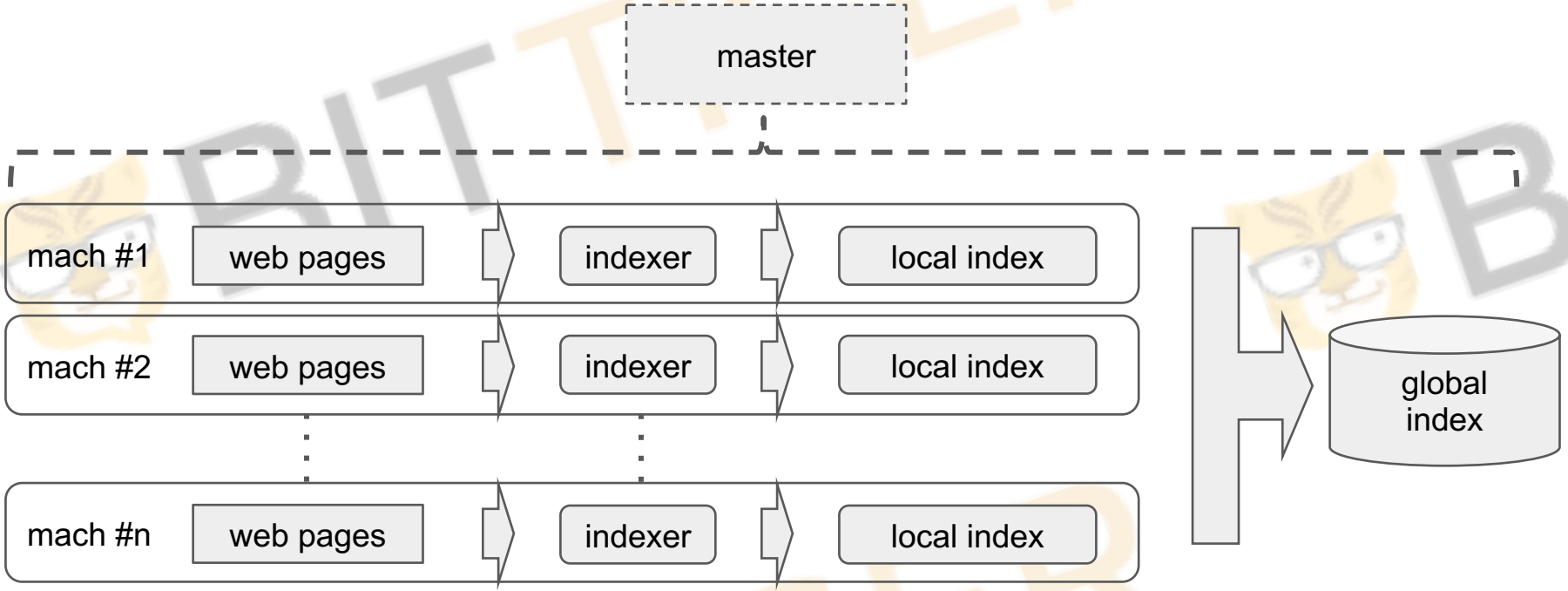
- Hardware failure
- Network partition
- Process hanging
- ……



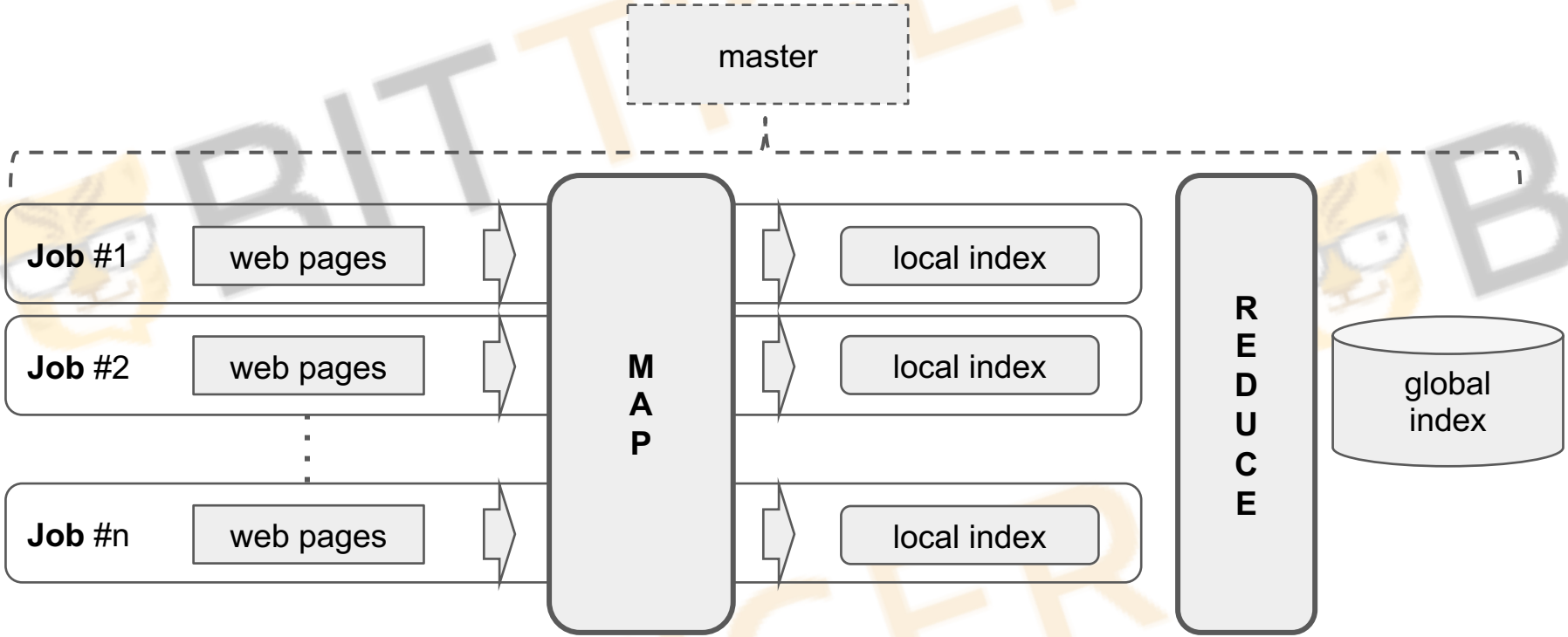
The Joys of Real Hardware Typical first year for a new cluster

- ~0.5 overheating (power down most machines in <5 mins, ~1-2 days to recover)
- ~1 PDU failure (~500-1000 machines suddenly disappear, ~6 hours to come back)
- ~1 rack-move (plenty of warning, ~500-1000 machines powered down, ~6 hours)
- ~1 network rewiring (rolling ~5% of machines down over 2-day span)
- ~20 rack failures (40-80 machines instantly disappear, 1-6 hours to get back)
- ~5 racks go wonky (40-80 machines see 50% packetloss)
- ~8 network maintenances (4 might cause ~30-minute random connectivity losses)
- ~12 router reloads (takes out DNS and external vips for a couple minutes)
- ~3 router failures (have to immediately pull traffic for an hour)

Parallel processing - master



MapReduce

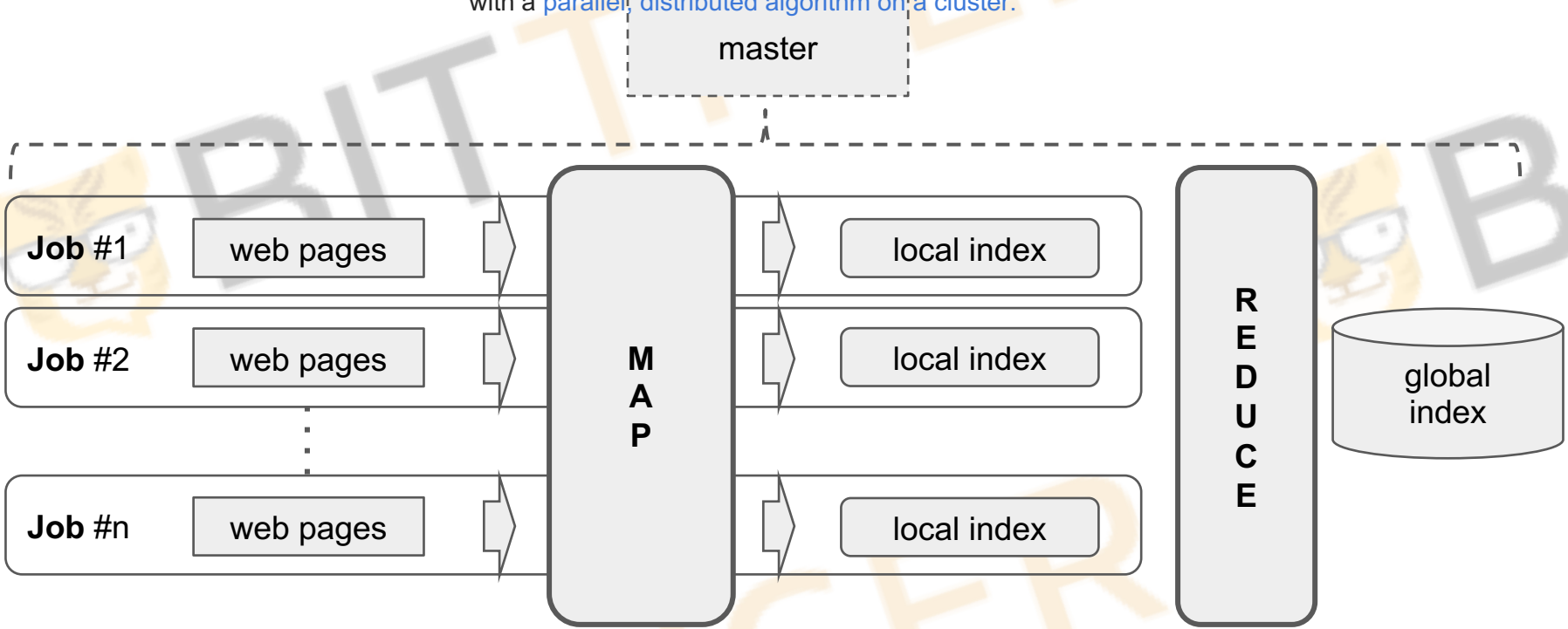


MapReduce

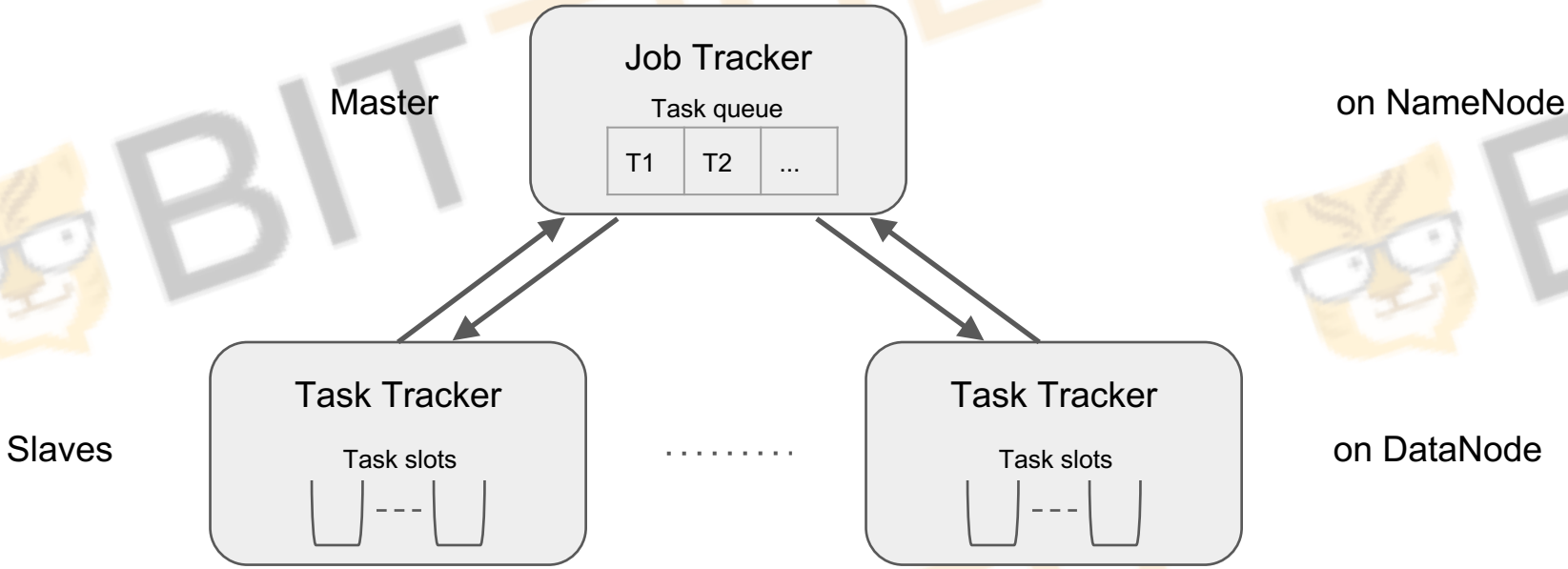
a programming model and

an associated implementation for processing and generating large data sets

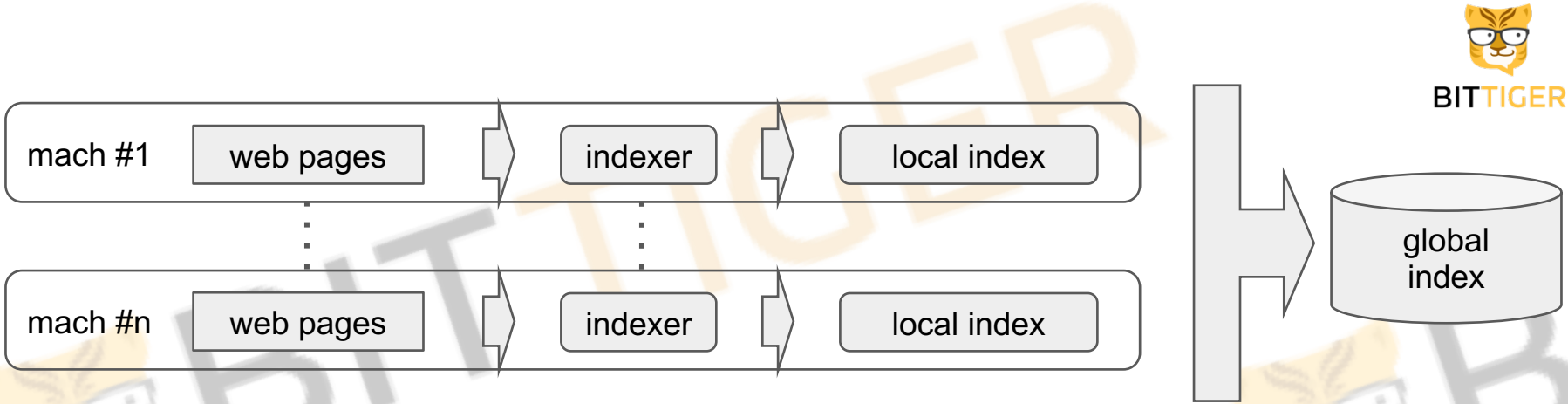
with a parallel, distributed algorithm on a cluster.

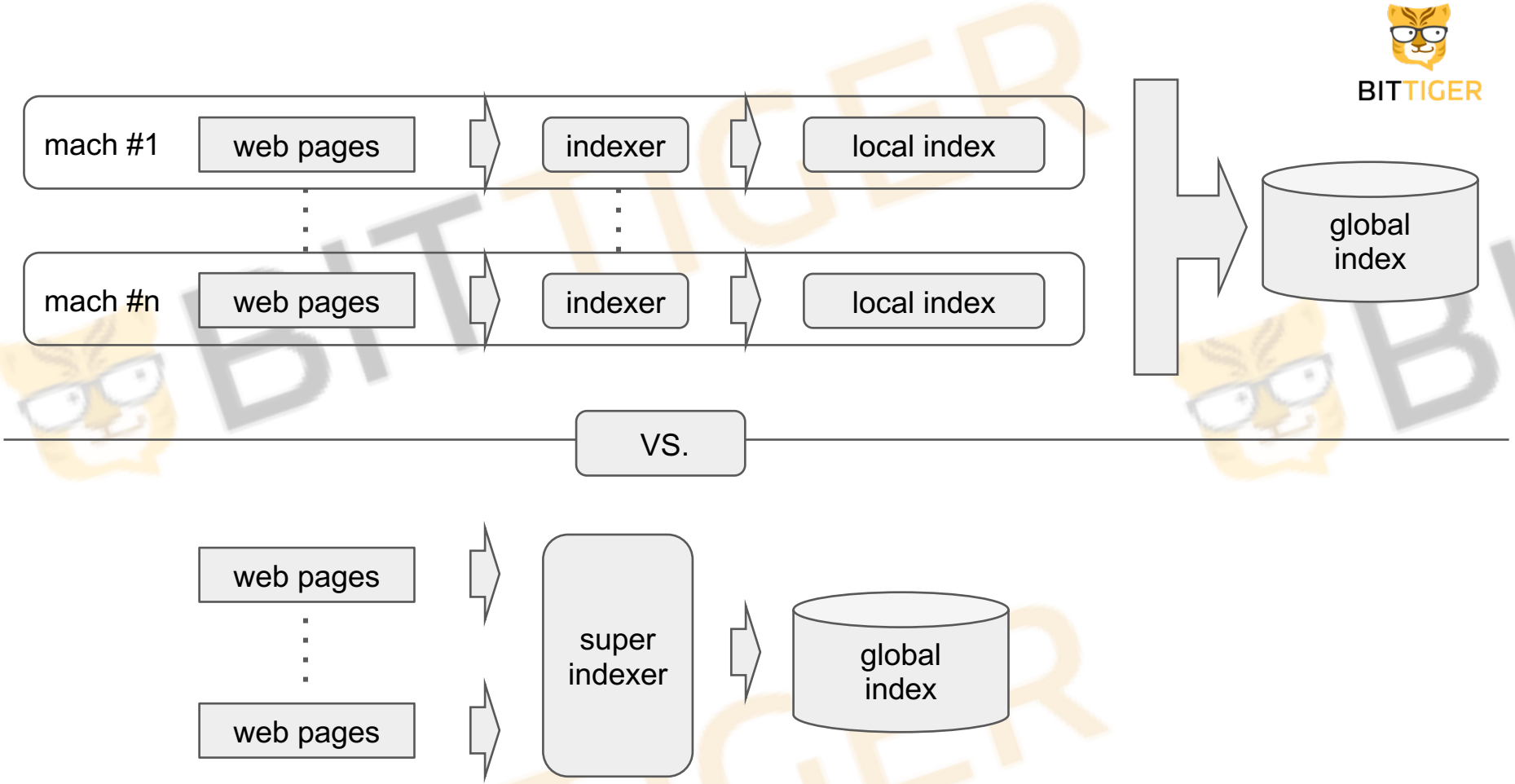


MapReduce

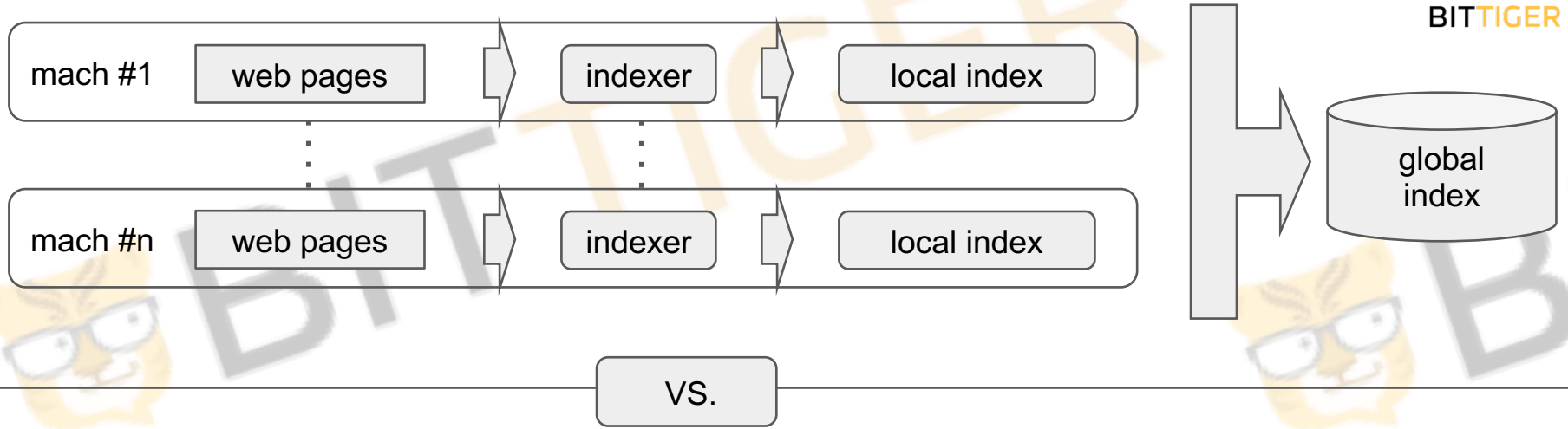


Task is dispatched to DataNode ....





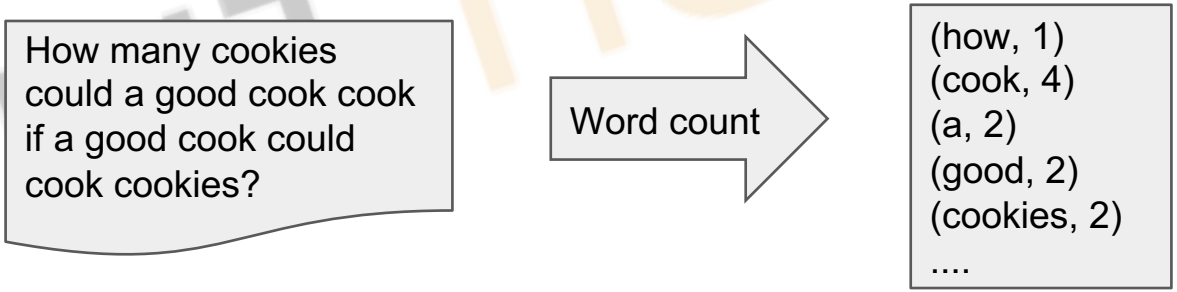
Shipping software to data



Shipping data to software

Back-of-envelope calculation

Task: word count over 1TB of text



(Raw data: 1TB) 1TB over 100Mb network → 80,000 sec (1 day = 86400 sec)

(Result size: 1 million * (6 + 4)B = 10 MB) 10MB over 100Mb network → ~1 sec

Back then

# of cores	1
RAM	1GB
HDD	8TB (RAID 1 with 8 drives)
Cost	$3000
Software	Nutch (Lucene) *
Result	100 page / second

Nutch was started by Cutting and Cafarella, and later turned into Hadoop.

Back then

# of cores	1
RAM	1GB
HDD	8TB (RAID 1 with 8 drives)
Cost	$3000
Software	Nutch (Lucene) *
Result	100 page / second

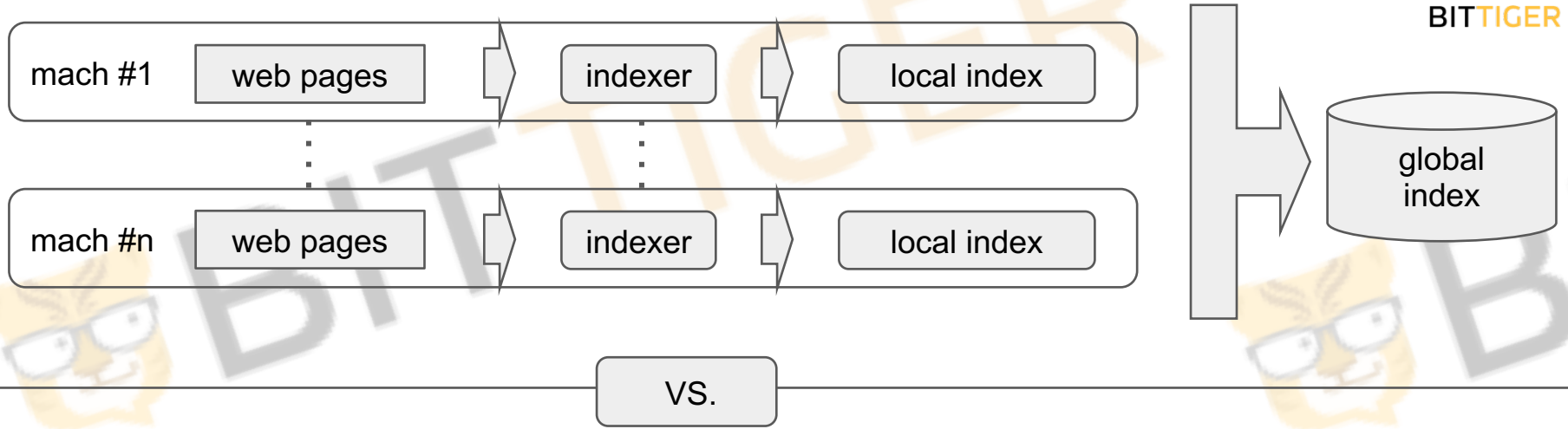
multi-core processor was not even a thing back then

Commodity hardware
(xeon/pentium 2006 dual core)

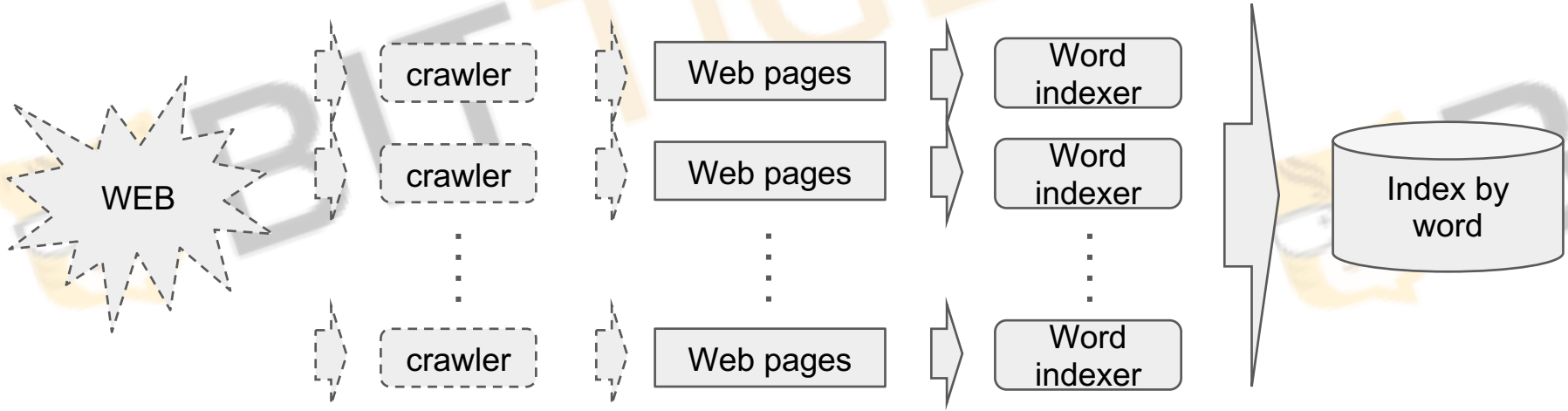
Mainframe
(IBM power4 2001 dual core)

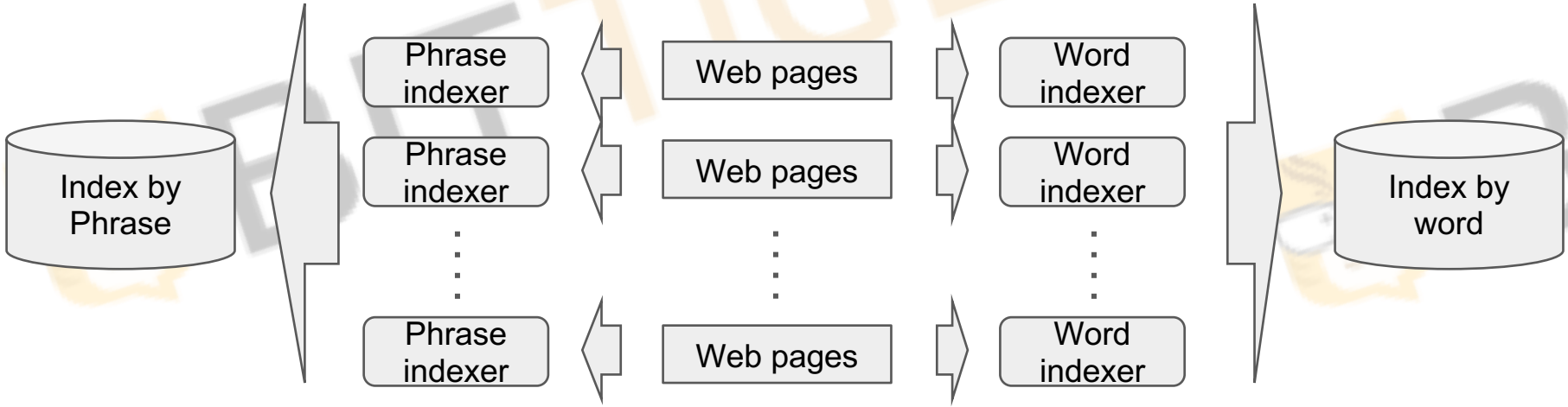
Nutch was started by Cutting and Cafarella, and later turned into Hadoop.

Shipping software to data ✓

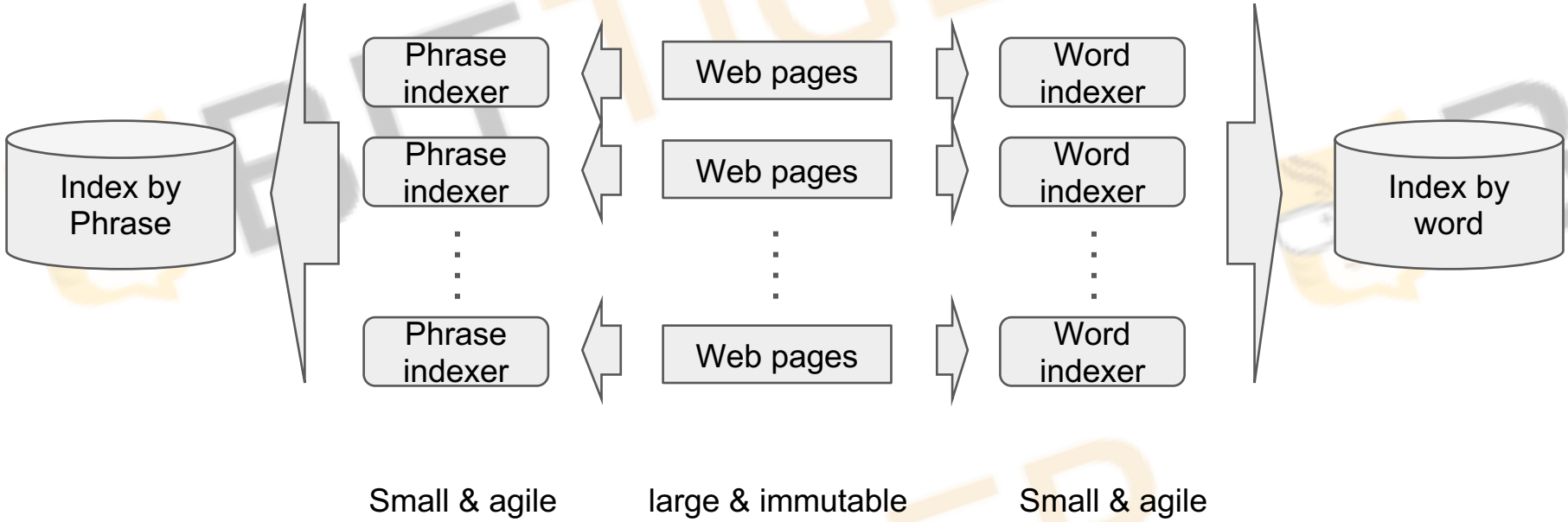


Shipping data to software

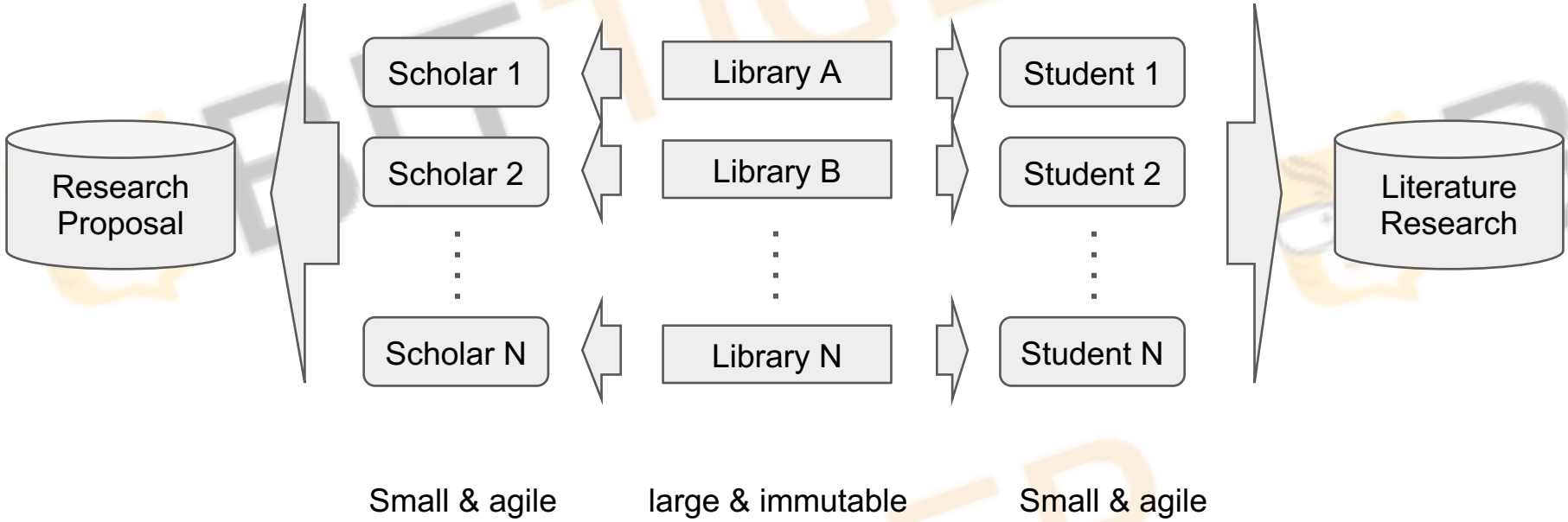




In computer world



In physical world



MapReduce checklist

Data:

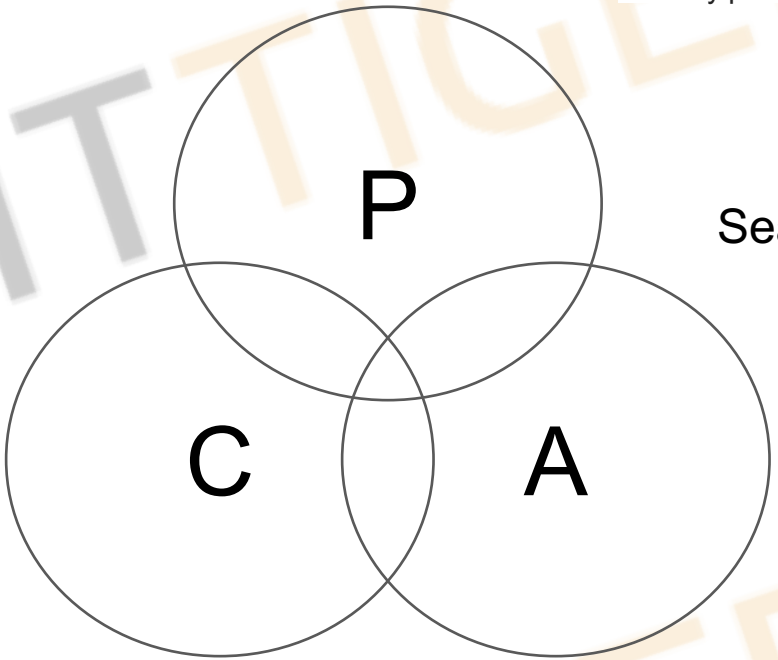
- Large
- Stationary
- Distributed

Actions:

- non-transactional operations (e.g. analytics)

CAP Theorem

Partition tolerance:
the system continues to operate despite arbitrary partitioning due to network failures



Search engine

Consistency:
every read receives the most recent write or an error

Availability:
every request receives a response, without guarantee that it contains the most recent version of the information

Hadoop

MapReduce has frequently been associated with *Hadoop* since its debut on the computing stage.

Hadoop is an open source implementation of MapReduce and is currently enjoying wide popularity

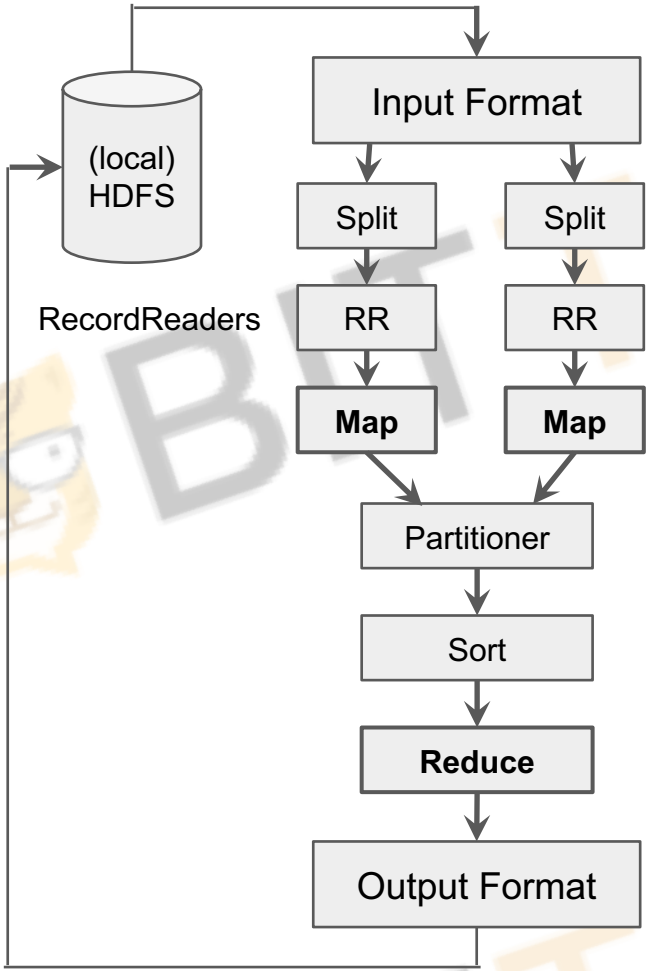
Hadoop presents MapReduce as an analytics engine and under the hood uses a distributed storage layer referred to as Hadoop Distributed File System (*HDFS*)

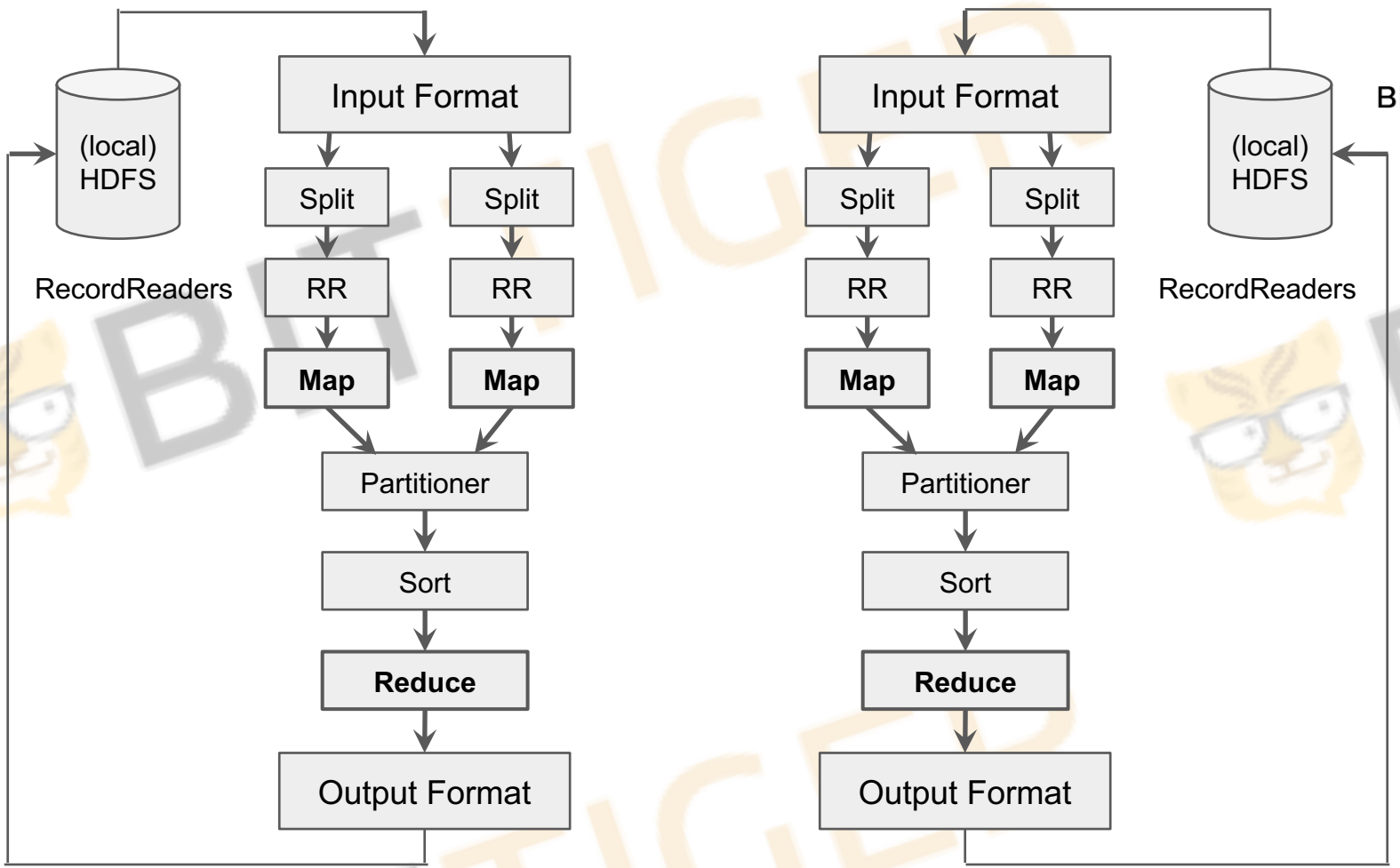
HDFS mimics Google File System (*GFS*)

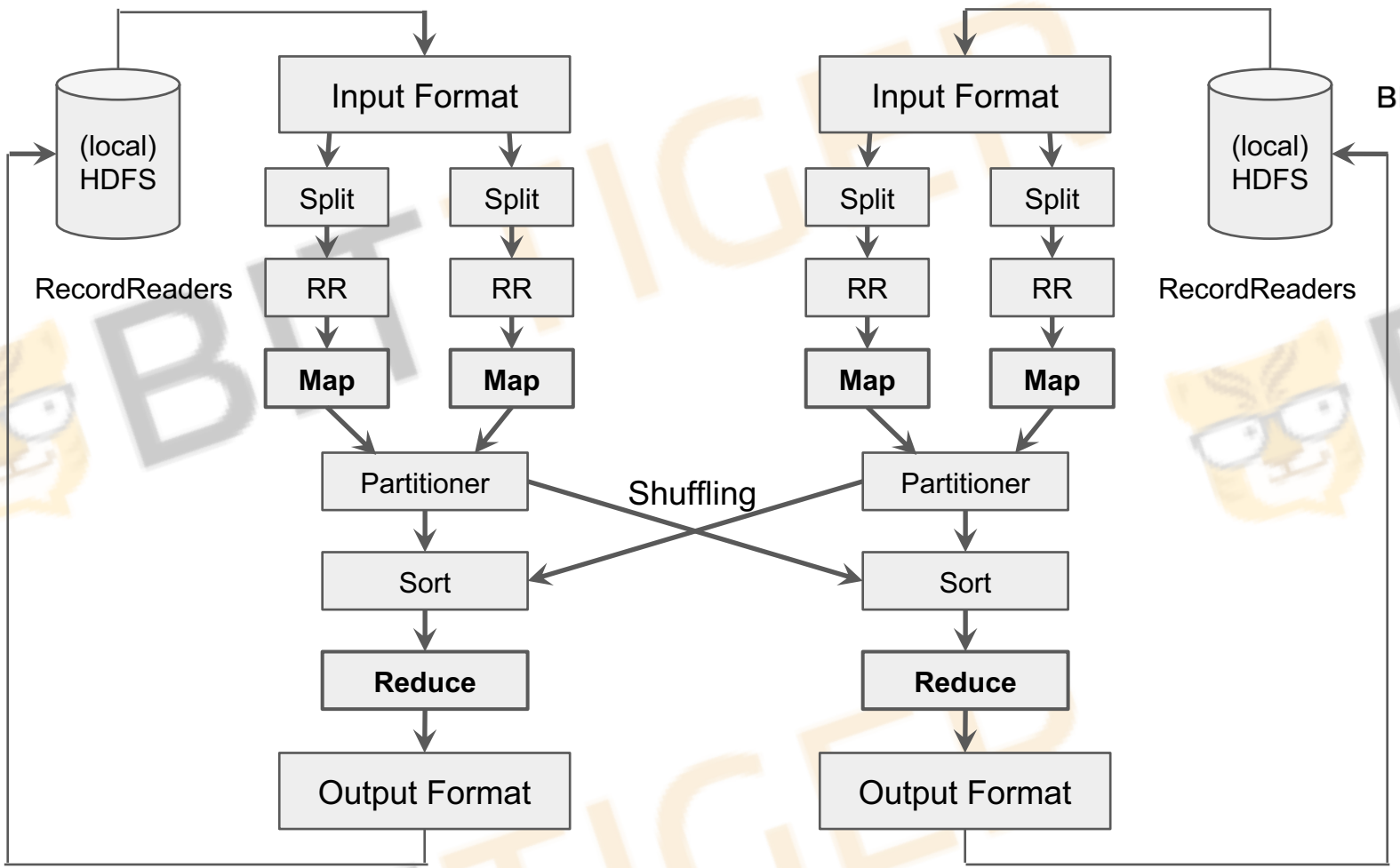
Distributed Storage Layer (GFS or HDFS)

BITTIGER

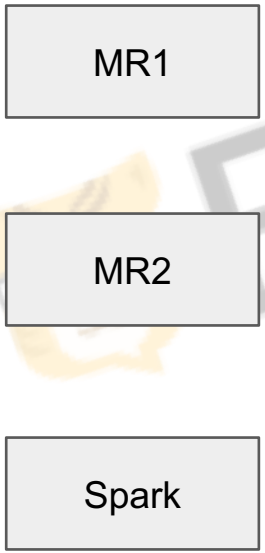
- Required by the amount of data
- Proper UNIX FS abstraction
- Replication + recovery







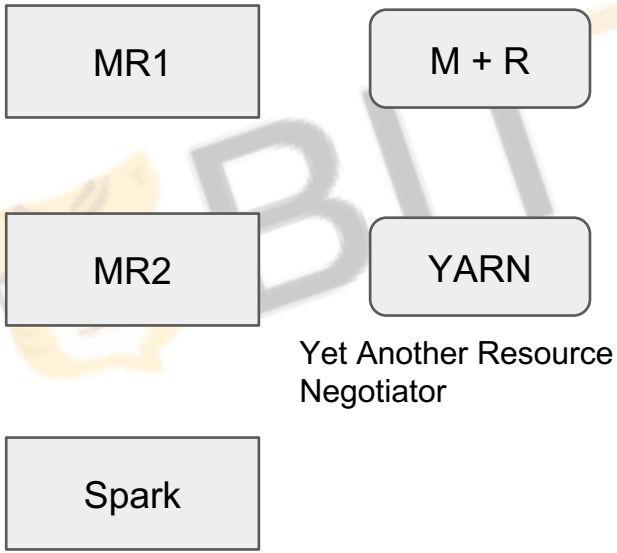
MR1 & MR2 & Spark



M + R

Abstracts data processing pipelines to two logical steps

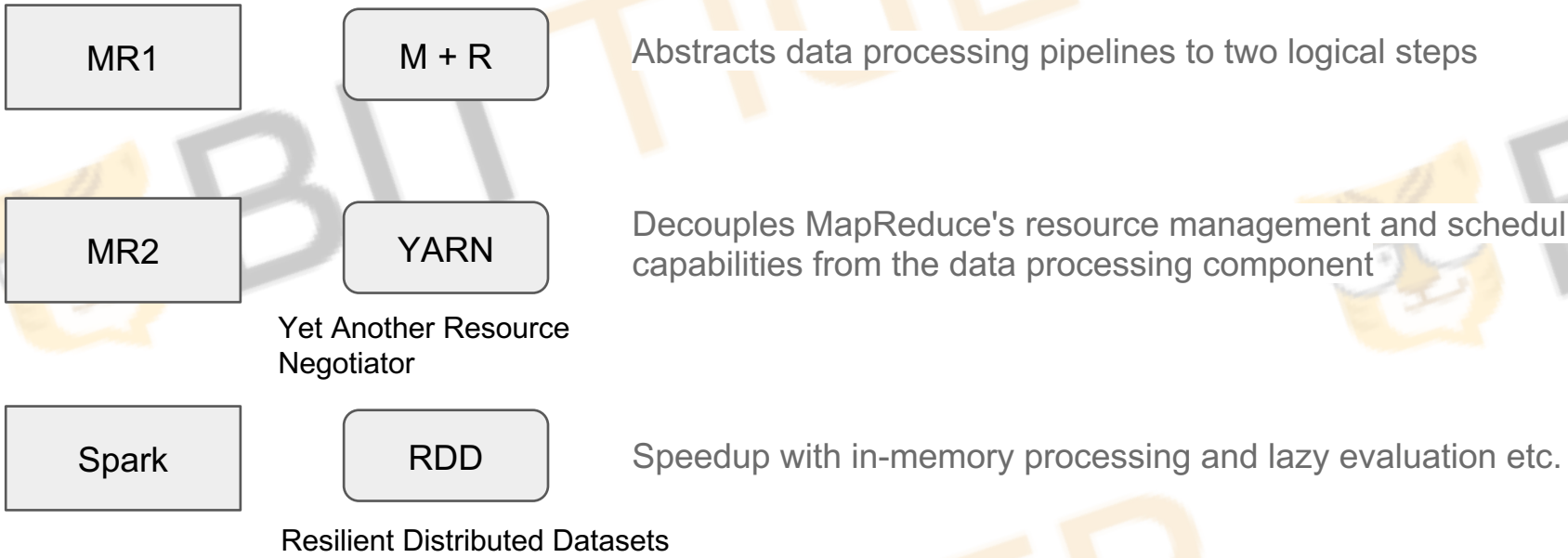
MR1 & MR2 & Spark



Abstracts data processing pipelines to two logical steps

Decouples MapReduce's resource management and scheduling capabilities from the data processing component

MR1 & MR2 & Spark



Summary

MapReduce

a programming model and

an associated implementation for processing and generating large data sets

with a parallel, distributed algorithm on a cluster.

Summary

MapReduce

a programming model and

an associated implementation for processing and generating large data sets

with a parallel, distributed algorithm on a cluster.

created to index the web

grew to a corner stone of data science.

BITTIGER