(Hadoop) MapReduce

DS 5110: Big Data Systems

Spring 2025

Lecture 7b

Yue Cheng

Some material taken/derived from:

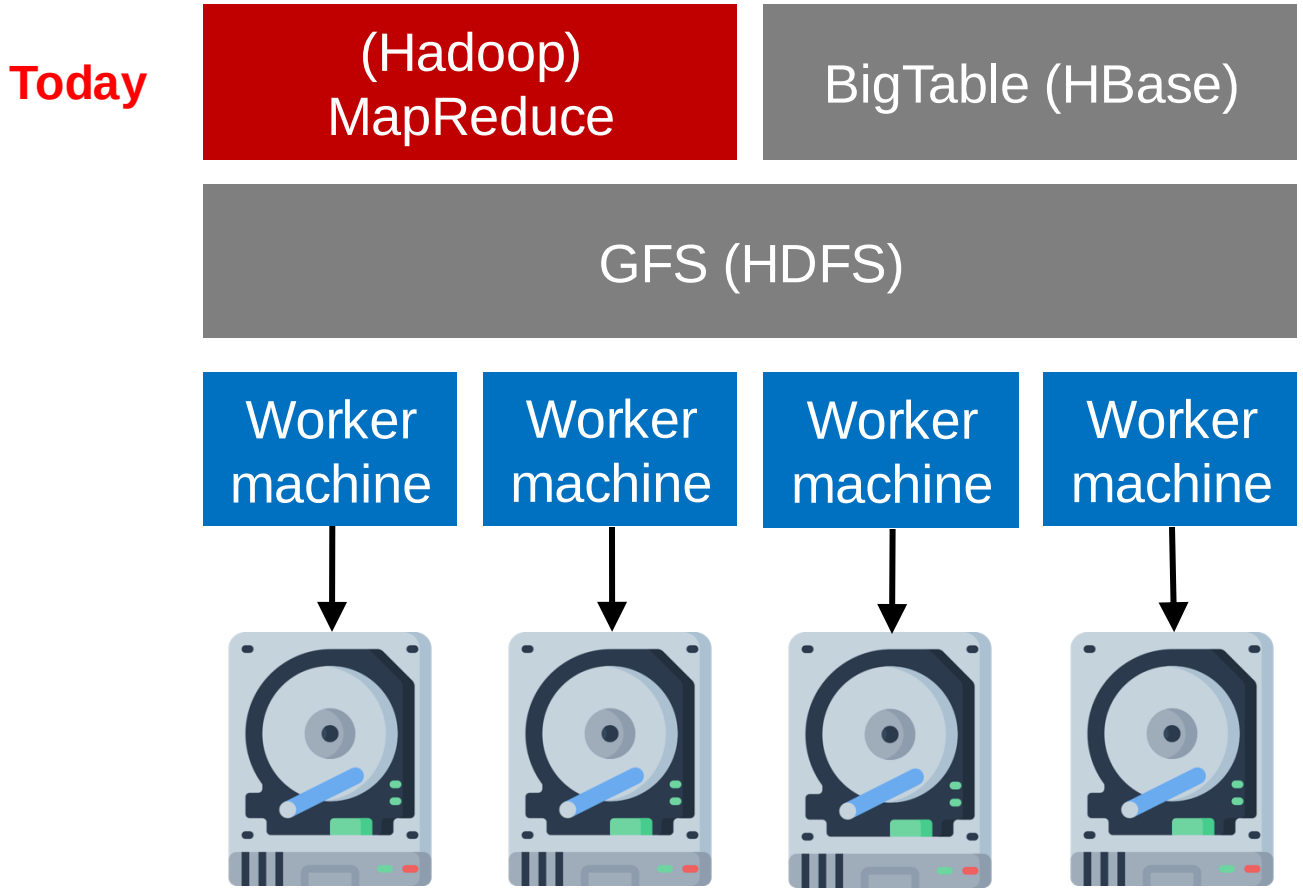
- Wisconsin CS 320 by Tyler Caraza-Harter.

HDFS demo...

Learning objectives

- Describe the role mappers and reducers have in MapReduce jobs
- Understand how MapReduce interacts with HDFS (GFS)

MapReduce


Today
(Hadoop) MapReduce
BigTable (HBase)
GFS (HDFS)
Worker machine
Worker machine
Worker machine
Worker machine

How does big data system answer questions?

SQL

A query, "`SELECT * FROM …`"

Results

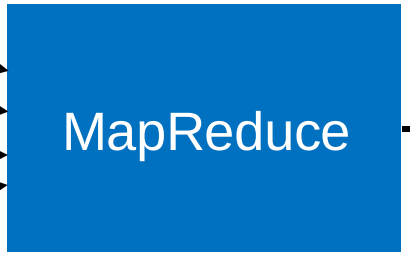
MapReduce

map function code

reduce function code

mapper/reducer counts

a file (or many)



Output files

How does big data system answer questions?

SQL

A query, "`SELECT * FROM …`" → Database → Results

MapReduce

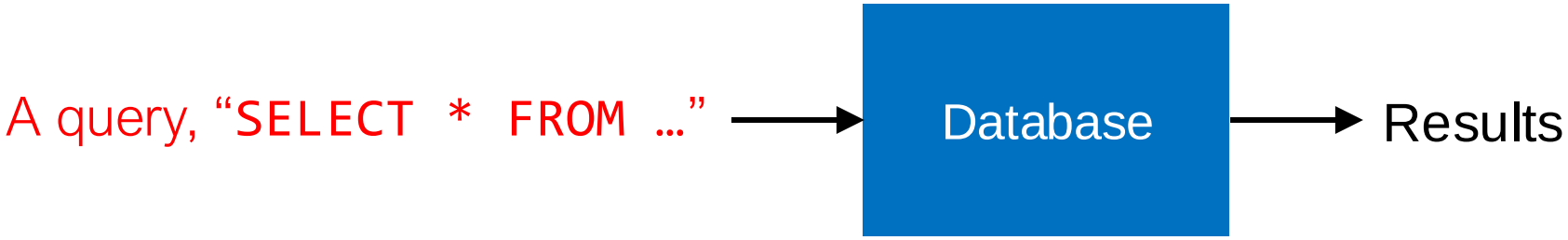
map function code
reduce function code
mapper/reducer counts
a file (or many)

→ MapReduce → Output files

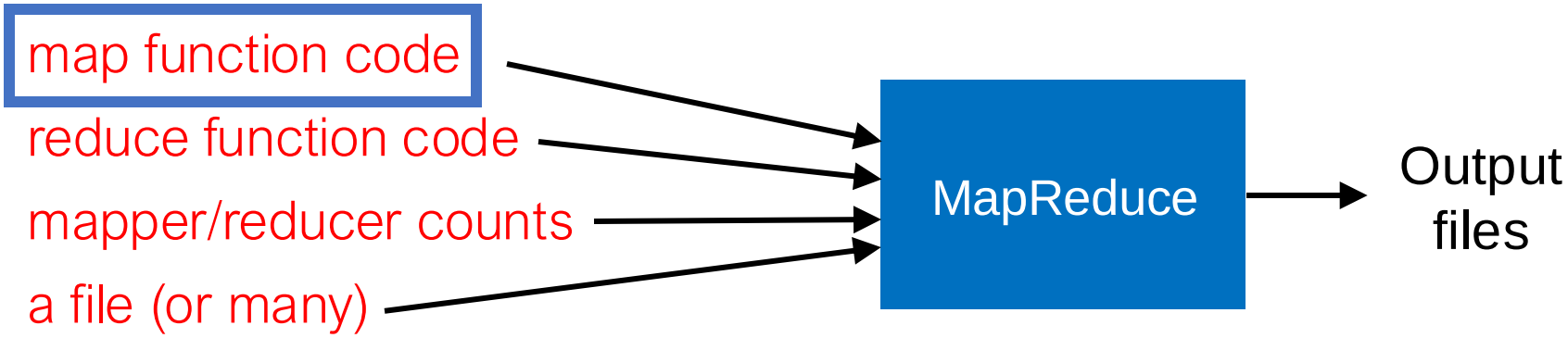
Input/output files are generally stored in HDFS

How does big data system answer questions?

SQL



MapReduce



Mappers by example: What are the colors of the squares?

input.csv (in HDFS)

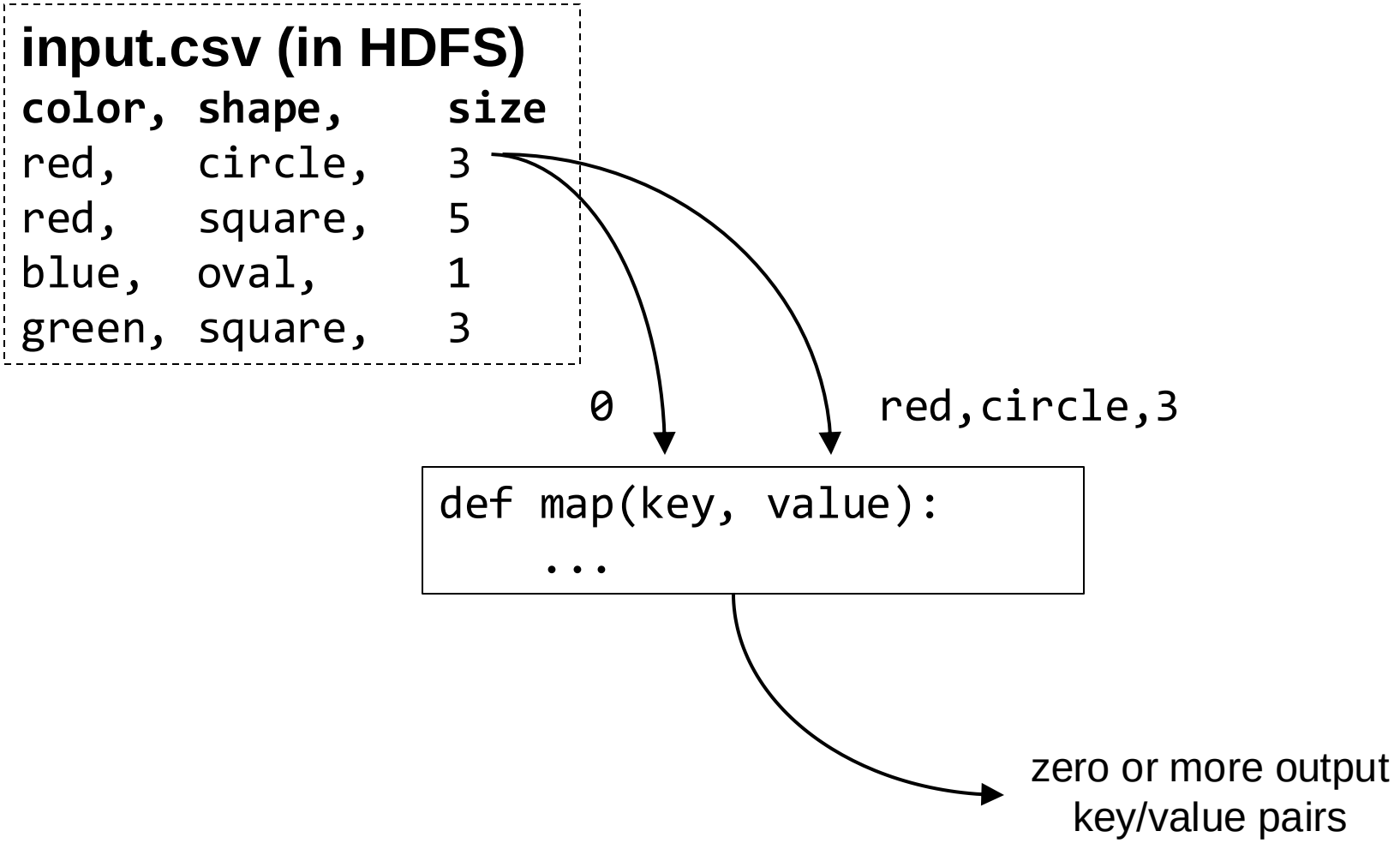
```
color, shape,    size
red,   circle,   3
red,   square,   5
blue,  oval,     1
green, square,   3
```

```
def map(key, value):
    ...
```

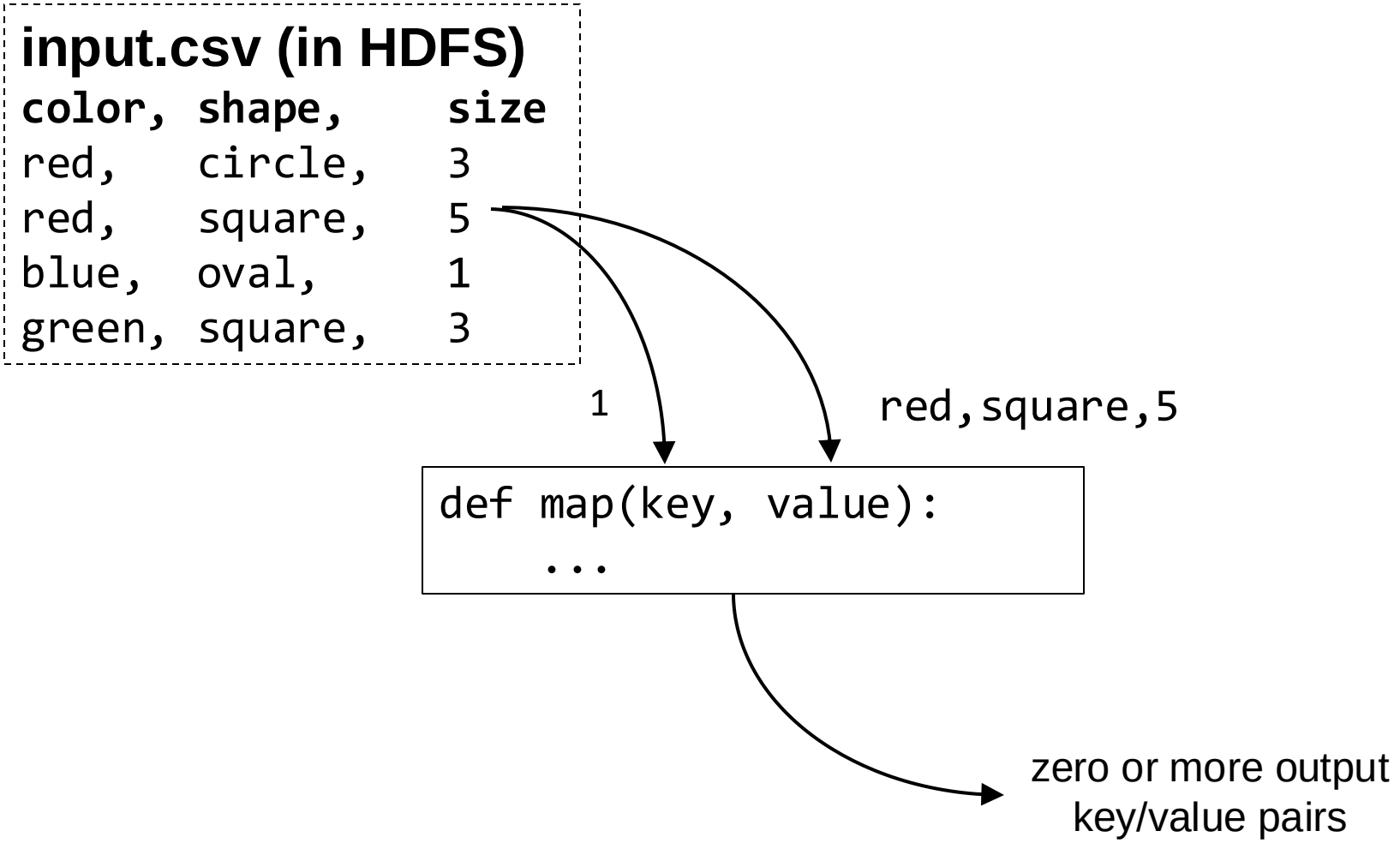
In SQL:

```
SELECT color FROM table WHERE shape = “square”
```

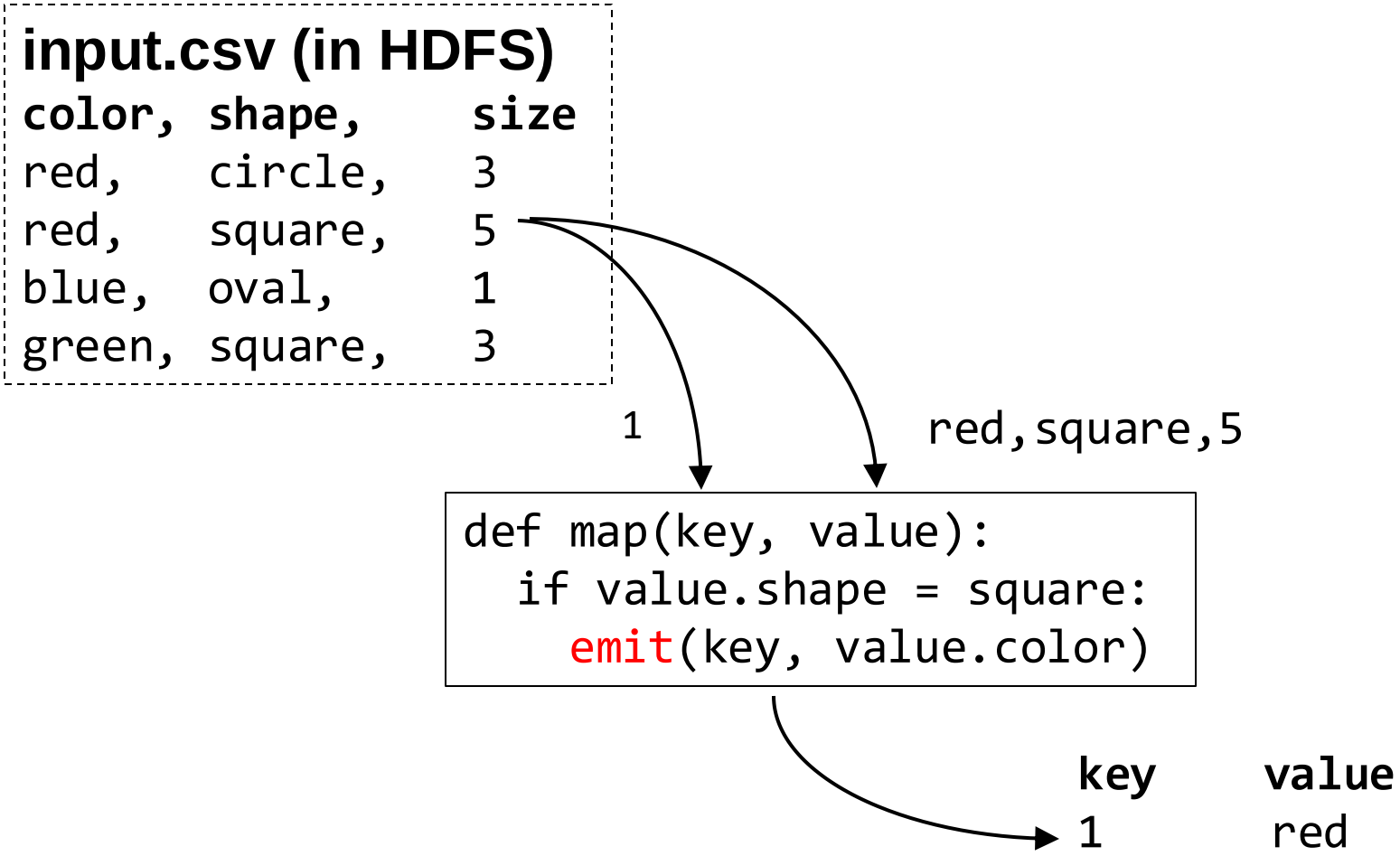
Mappers by example: What are the colors of the squares?



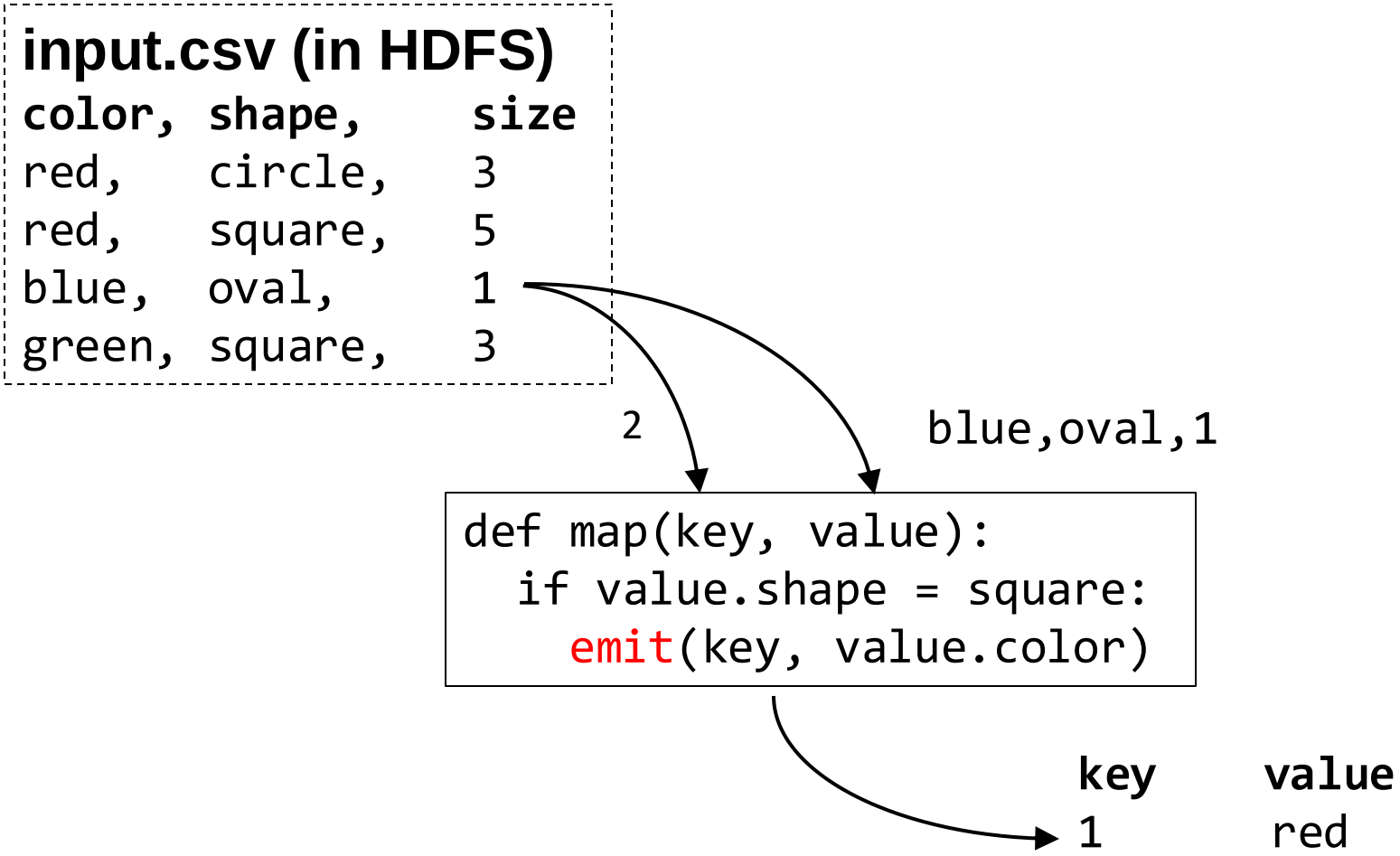
Mappers by example: What are the colors of the squares?



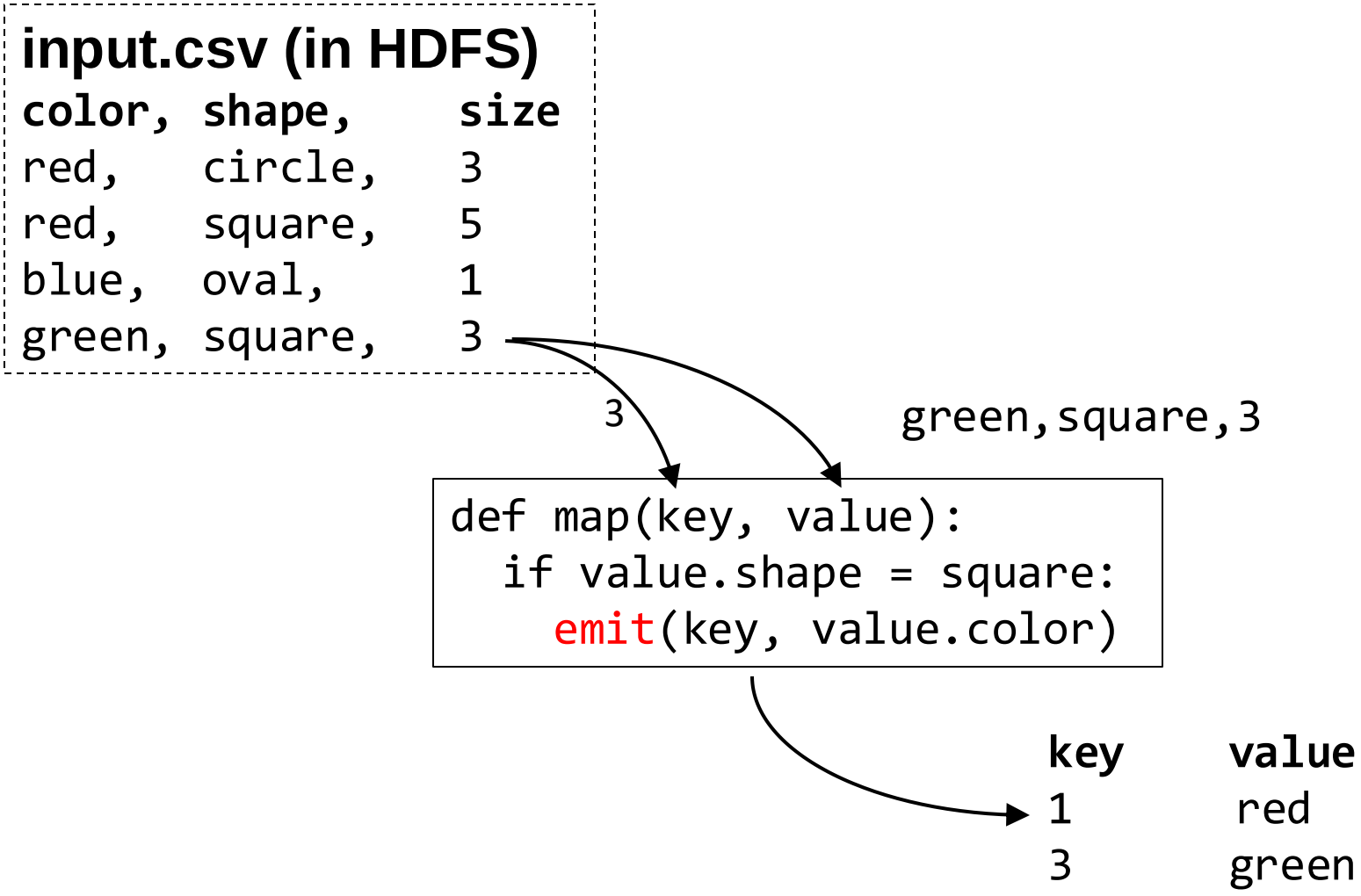
Mappers by example: What are the colors of the squares?



Mappers by example: What are the colors of the squares?



Mappers by example: What are the colors of the squares?



Mappers by example: What are the colors of the squares?

input.csv (in HDFS)

```
color, shape,    size
red,   circle,   3
red,   square,   5
blue,  oval,     1
green, square,   3
```

What if the data is huge?

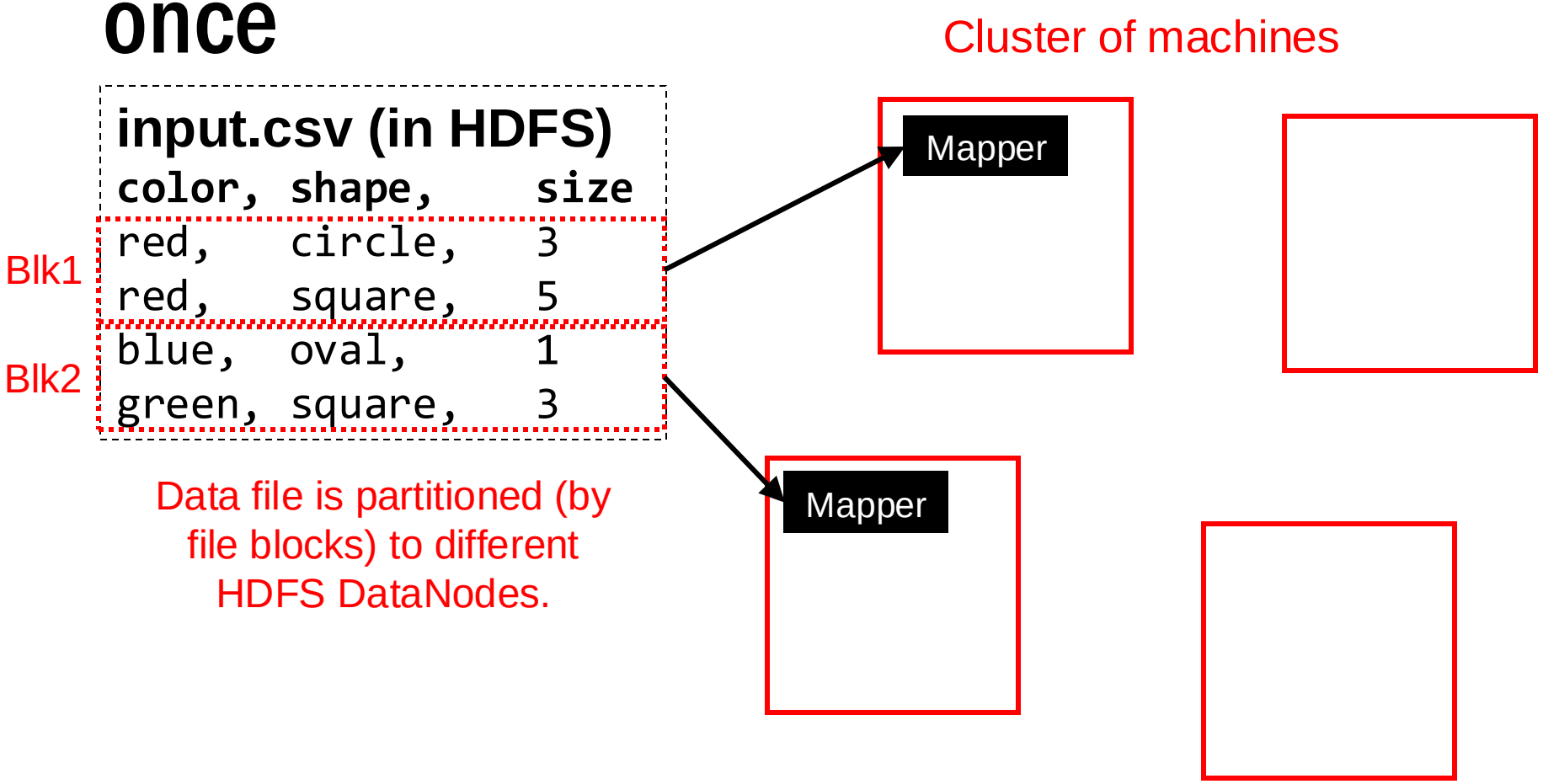
3

green,square,3

```
def map(key, value):
  if value.shape = square:
    emit(key, value.color)
```

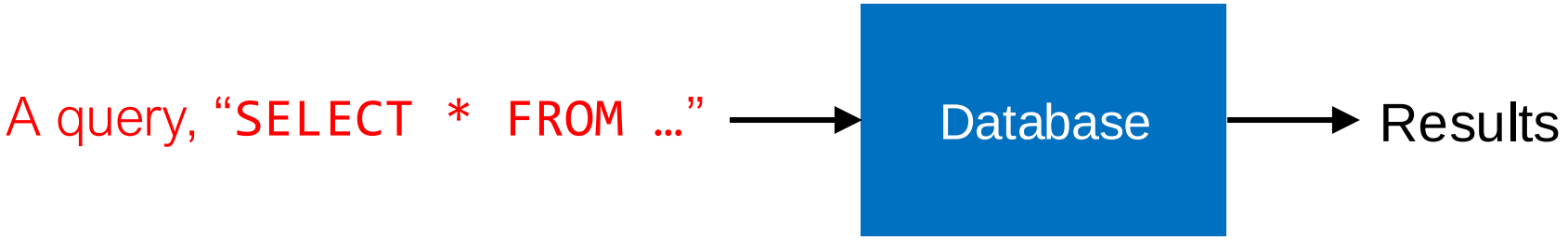
key	value
1	red
3	green

Mappers run on multiple machines at once

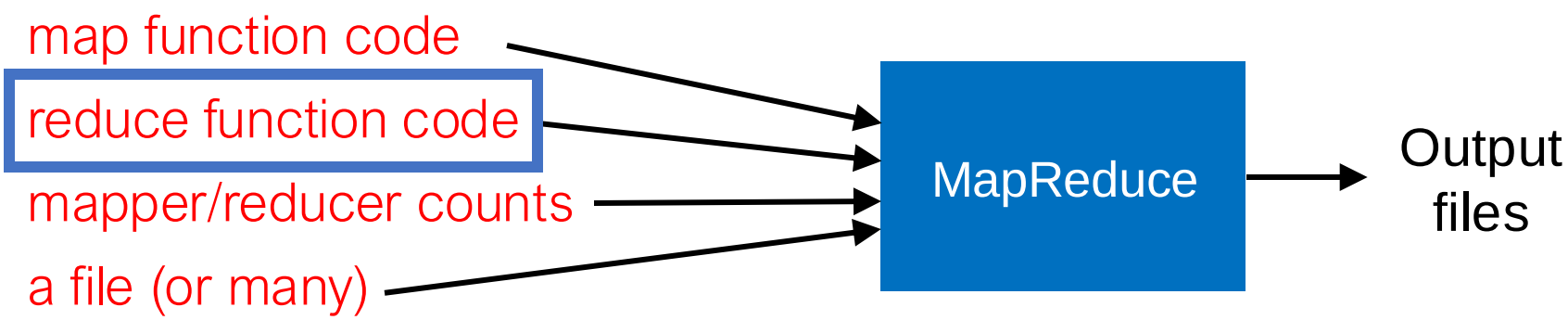


How does big data system answer questions?

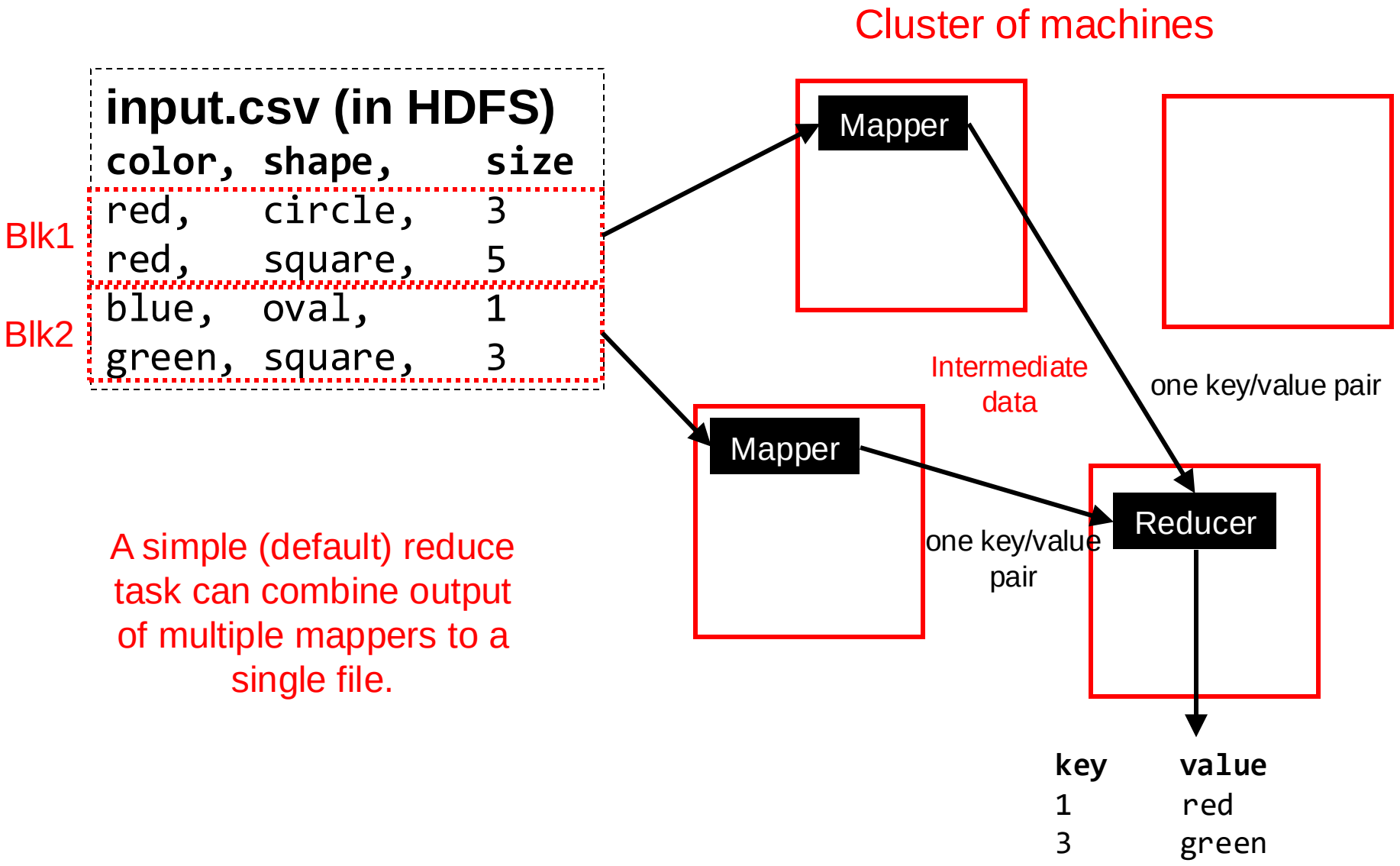
SQL



MapReduce



Reducers



A simple (default) reduce task can combine output of multiple mappers to a single file.

Reducers

Reducers can output exactly their input,
OR have further computation.

```
def reduce(key, values):
  for row in values:
    emit(key, row)
```

Reducers

input.csv (in HDFS)

```
color,  shape,    size
red,    circle,   3
red,    square,   5
blue,   oval,     1
green,  square,   3
```

```
def map(key, value):
  emit(value.color, value)
```

key	value
blue	blue, oval, 1
green	green, square, 3
red	red, circle, 3
red	red, square, 5

```
def reduce(key, values):
  count = 0
  for row in values:
    count = count + 1
  emit(key, count)
```

Intermediate data is **grouped** and **sorted** by key.

Reduce will be called 3 times (once for each group). The call could happen in one reduce task (or be split over many).

Reducers

input.csv (in HDFS)

```
color,  shape,    size
red,    circle,   3
red,    square,   5
blue,   oval,     1
green,  square,   3
```

```
def map(key, value):
  emit(value.color, value)
```

key	value
blue	blue, oval, 1
green	green, square, 3
red	red, circle, 3
red	red, square, 5

```
def reduce(key, values):
  count = 0
  for row in values:
    count = count + 1
  emit(key, count)
```

key	value
blue	1

Intermediate data is **grouped** and **sorted** by key.

Reducers

input.csv (in HDFS)

```
color,  shape,    size
red,    circle,   3
red,    square,   5
blue,   oval,     1
green,  square,   3
```

```
def map(key, value):
  emit(value.color, value)
```

key	value
blue	blue, oval, 1
green	green, square, 3
red	red, circle, 3
red	red, square, 5

```
def reduce(key, values):
  count = 0
  for row in values:
    count = count + 1
  emit(key, count)
```

key	value
blue	1
green	1

Intermediate data is **grouped** and **sorted** by key.

Reducers

input.csv (in HDFS)

```
color,  shape,    size
red,    circle,   3
red,    square,   5
blue,   oval,     1
green,  square,   3
```

```
def map(key, value):
  emit(value.color, value)
```

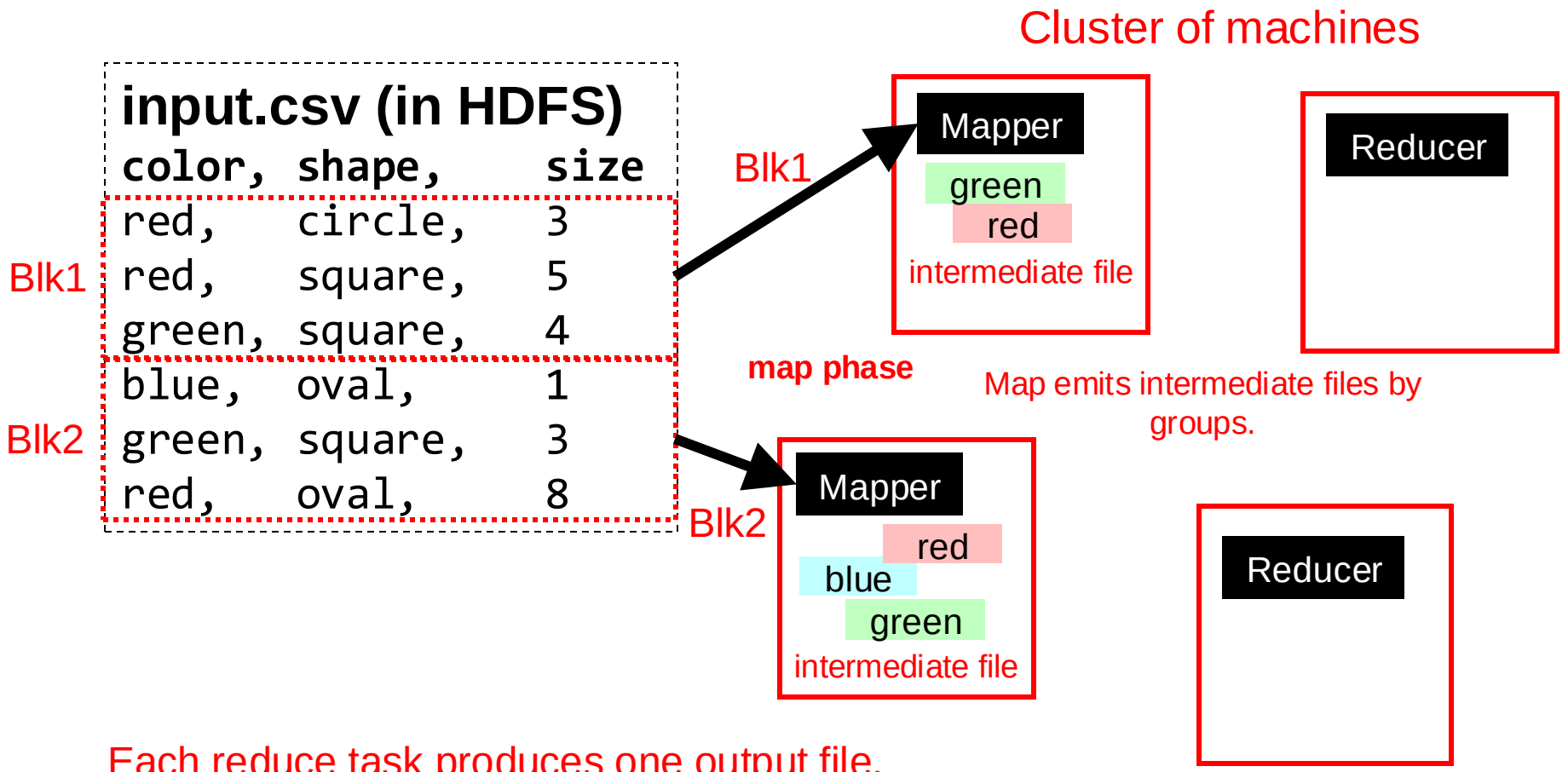
key	value
blue	blue, oval, 1
green	green, square, 3
red	red, circle, 3
red	red, square, 5

Intermediate data is **grouped** and **sorted** by key.

```
def reduce(key, values):
  count = 0
  for row in values:
    count = count + 1
  emit(key, count)
```

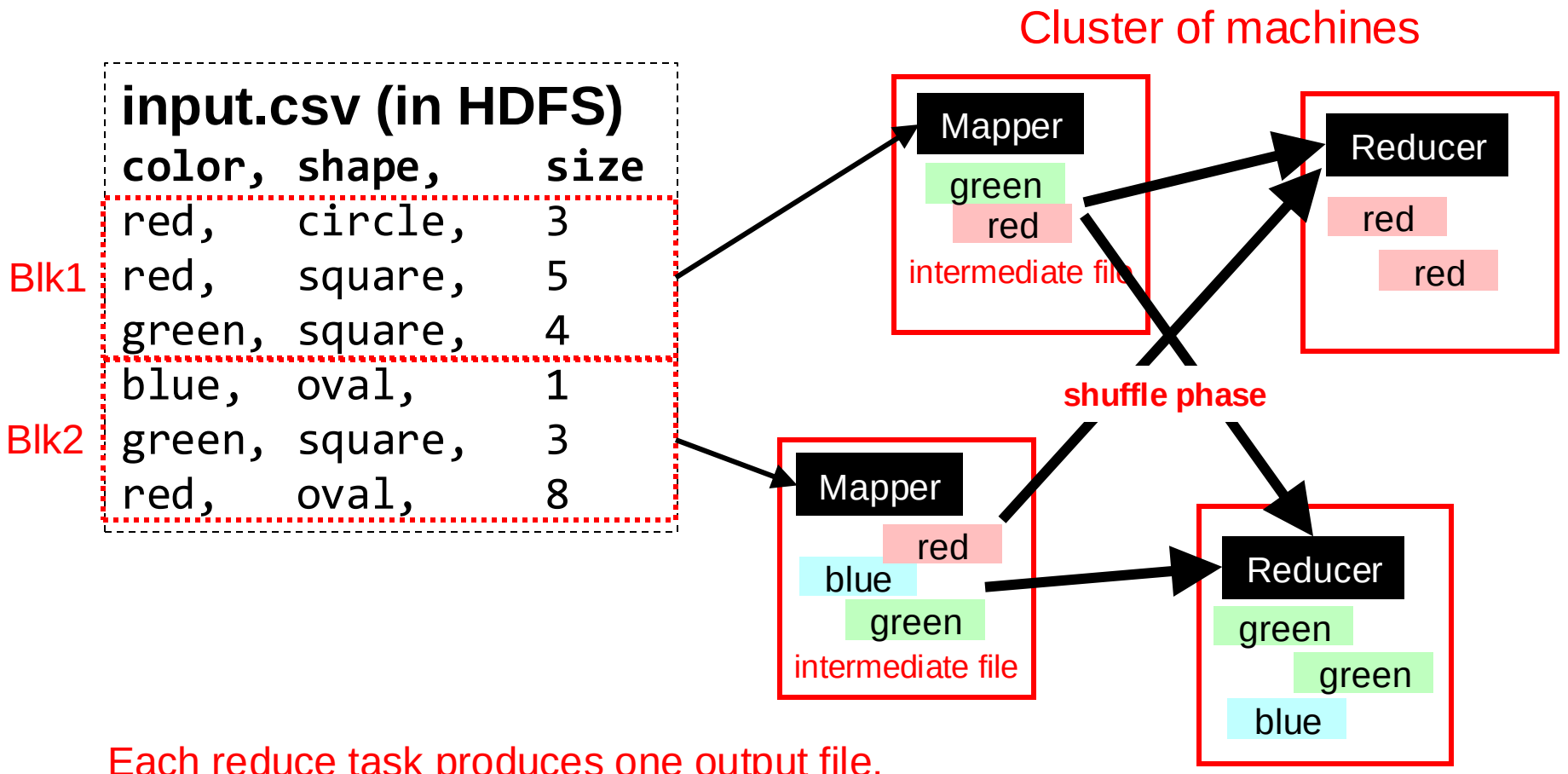
key	value
blue	1
green	1
red	2

Multiple reducers (for big intermediate data)



Each reduce task produces one output file.
A reduce task might take multiple keys.
All intermediate rows with the same key go to the same reducer.

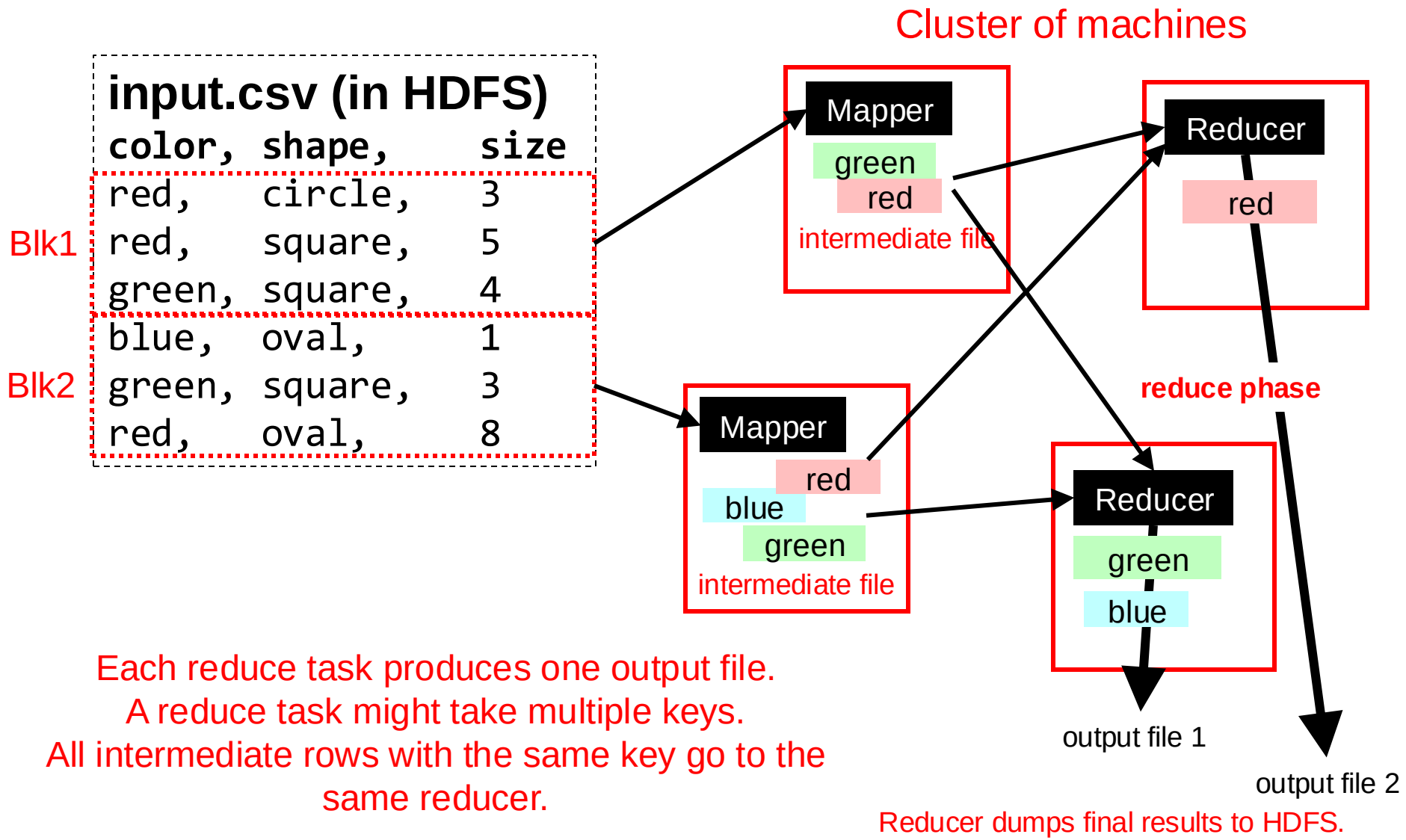
Multiple reducers (for big intermediate data)



Each reduce task produces one output file.
A reduce task might take multiple keys.
All intermediate rows with the same key go to the same reducer.

Reducer collects all intermediate files of its assigned keys (groups).

Multiple reducers (for big intermediate data)

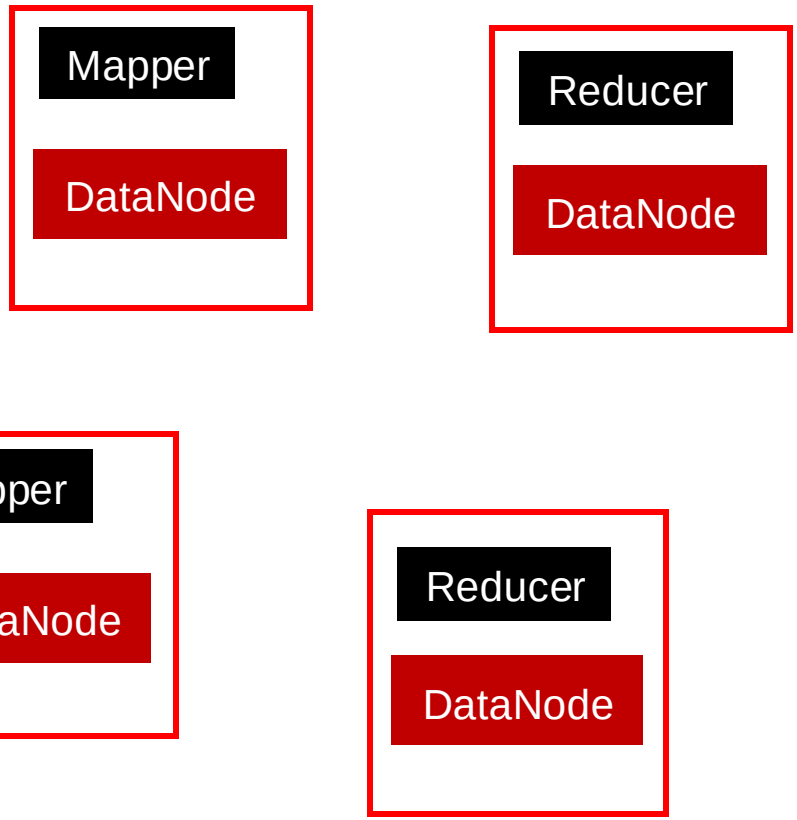


Each reduce task produces one output file.
A reduce task might take multiple keys.
All intermediate rows with the same key go to the same reducer.

Data locality: Avoid network transfer

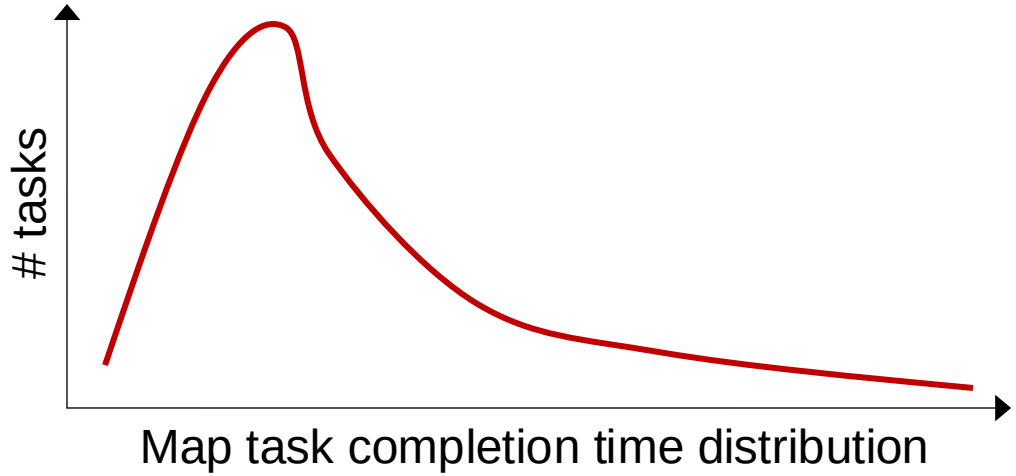
Run on same machines

- Layered subsystems
- MapReduce executor
- HDFS DataNode

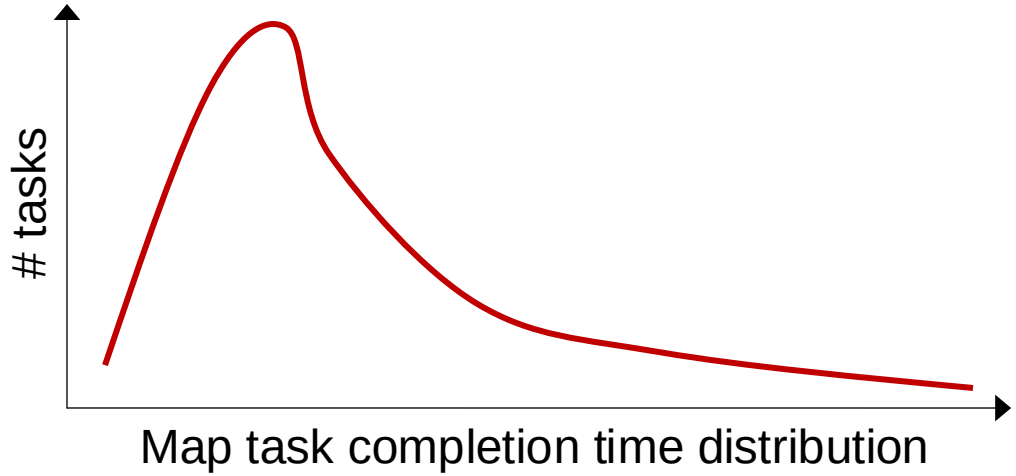


Try to run mappers on machine where DataNode has needed data. Uses local disk but not network.

Stragglers



Stragglers



- **Tail execution time** means some executors (always) finish late (**recall tail latency**)

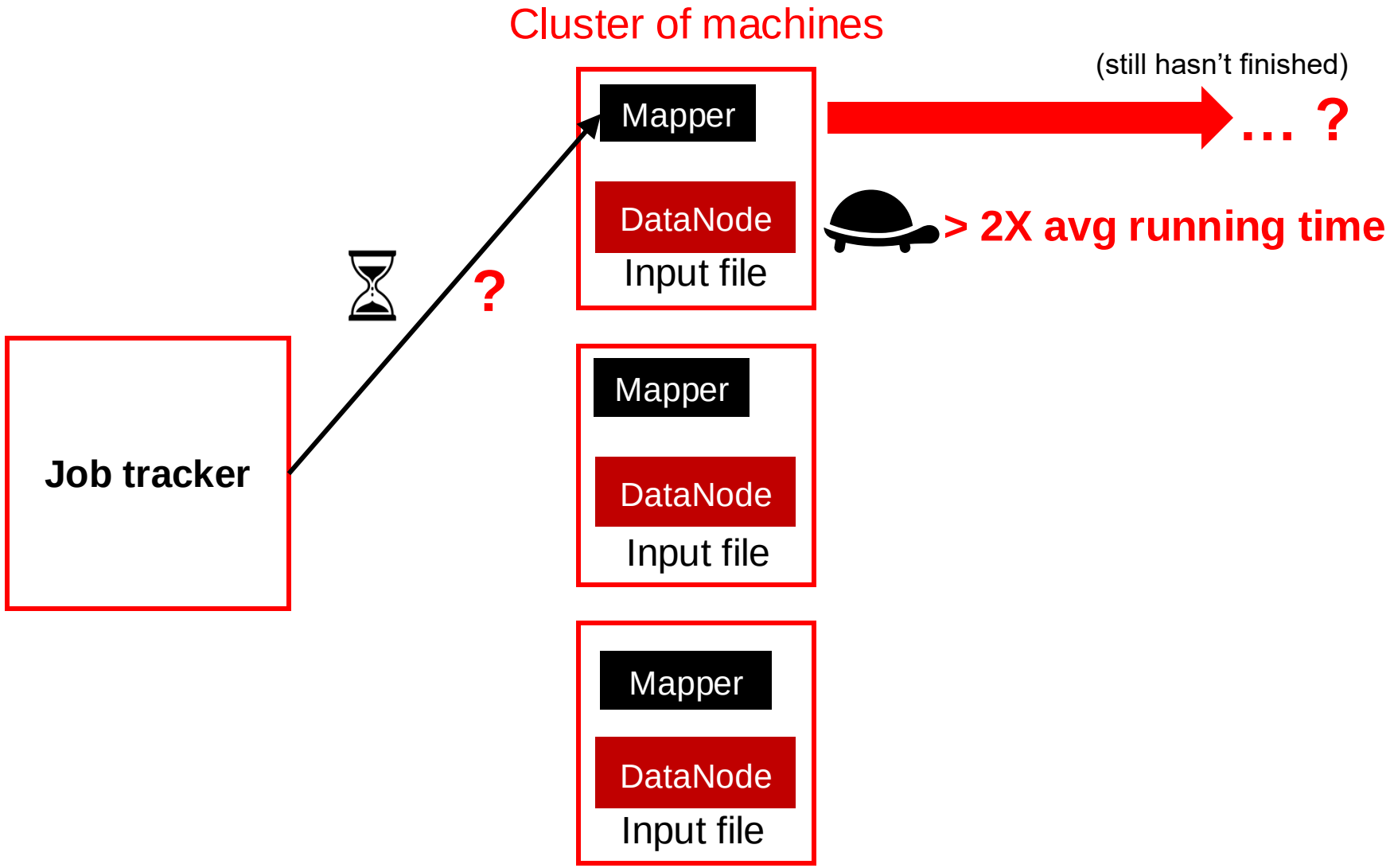
Q: How can MapReduce work around this?

- Hint: its approach to **fault-tolerance** provides the right tool

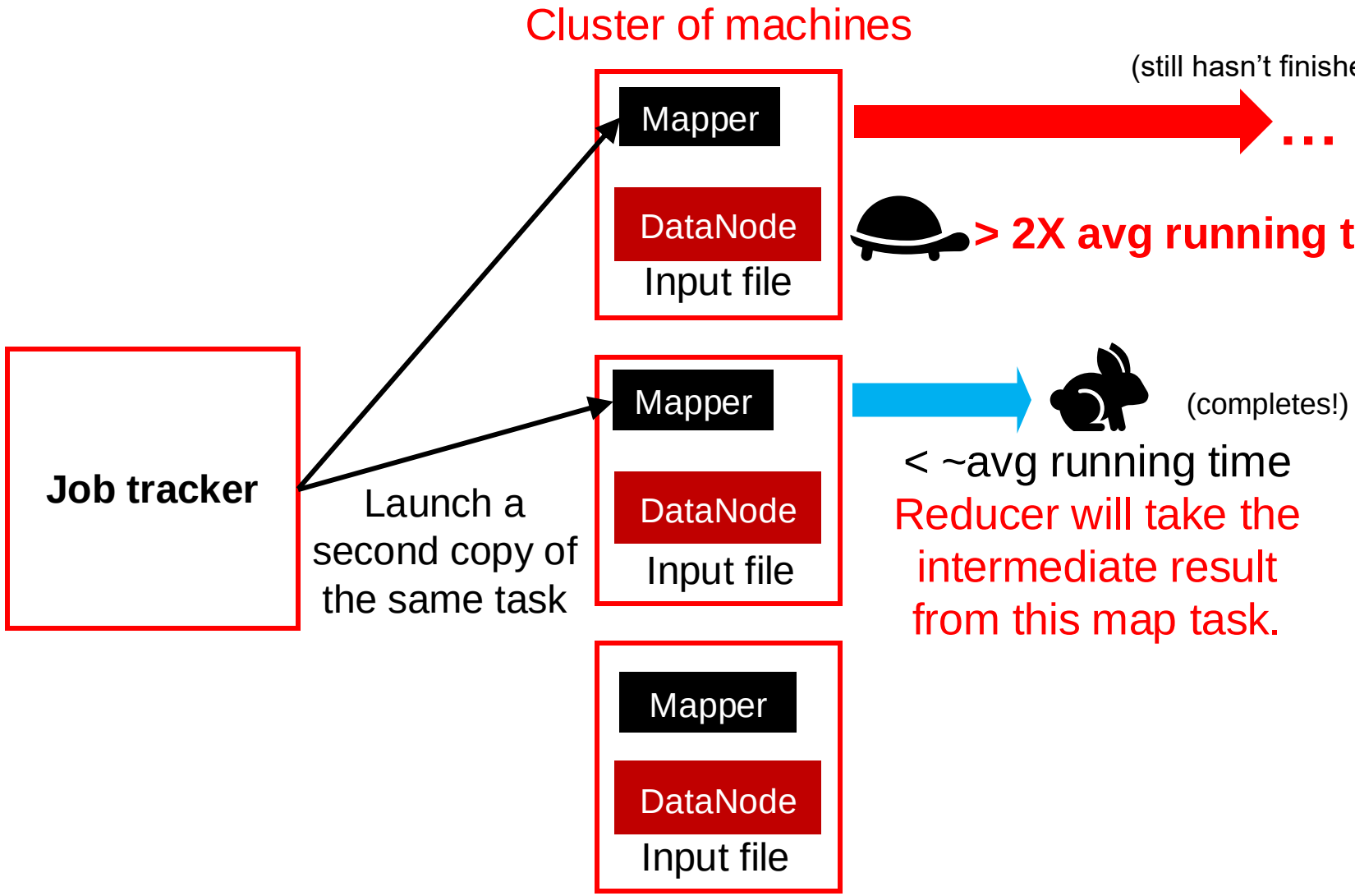
Resilience against stragglers?

- If a task is going slowly (i.e., straggler):
  - Launch second copy of task (**backup task**) on another node
  - Take the output of whichever finishes first

Resilience against stragglers



Resilience against stragglers



Would backup tasks cause correctness issue in MapReduce jobs?

Discussion: MapReduce eval (paper)