Week 3-2

MapReduce II

Big Data

Prof. Hwanjo Yu
POSTECH

Example: Word length histogram

Abridged Declaration of Independence

A Declaration By the Representatives of the United States of America, in General Congress Assembled.
When in the course of human events it becomes necessary for a people to advance from that subordination in which they have hitherto remained, and to assume among powers of the earth the equal and independent station to which th e laws of nature and of nature's god entitle them, a decent respect to the opinions of mankind requires that they shoul d declare the causes which impel them to the change.
We hold these truths to be self-evident; that all men are created equal and independent; that from that equal creatio n they derive rights inherent and inalienable, among which are the preservation of life, and liberty, and the pursuit of happiness; that to secure these ends, governments are instituted among men, deriving their just
power from the consent of the governed; that whenever any form of government shall become destructive of these en ds, it is the right of the people to alter or to abolish it, and to institute new government, laying it's foundation on such principles and organizing it's power in such form, as to them shall seem most likely to effect their safety and happine ss. Prudence indeed will dictate that governments long established should not be changed for light and transient caus es: and accordingly all experience hath shewn that mankind are more disposed to suffer while evils are sufferable, th an to right themselves by abolishing the forms to which they are accustomed. But when a long train of abuses and us urpations, begun at a distinguished period, and pursuing invariably the same object, evinces a design to reduce them t o arbitrary power, it is their right, it is their duty, to throw off such government and to provide new guards for future security. Such has been the patient sufferings of the colonies; and such is now the necessity which constrains them to expunge their former systems of government. the history of his present majesty is a history of unremitting injuries an d usurpations, among which no one fact stands single or solitary to contradict the uniform tenor of the rest, all of whi ch have in direct object the establishment of an absolute tyranny over these states. To prove this, let facts be submitte d to a candid world, for the truth of which we pledge a faith yet unsullied by falsehood.

How many “big”, “medium”, and “small” words are used?

Example: Word length histogram

Big = Yellow = 10+ letters

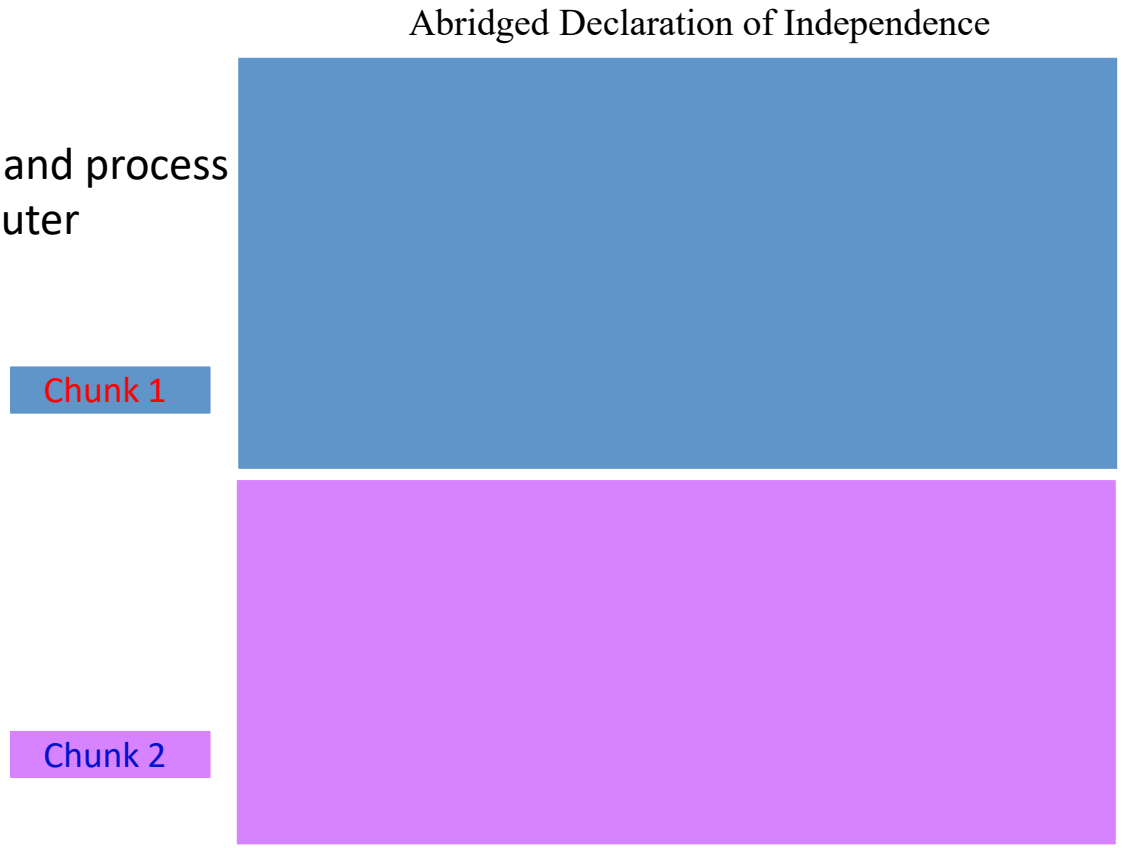
Medium = Red = 5..9 letters

Small = Blue = 2..4 letters

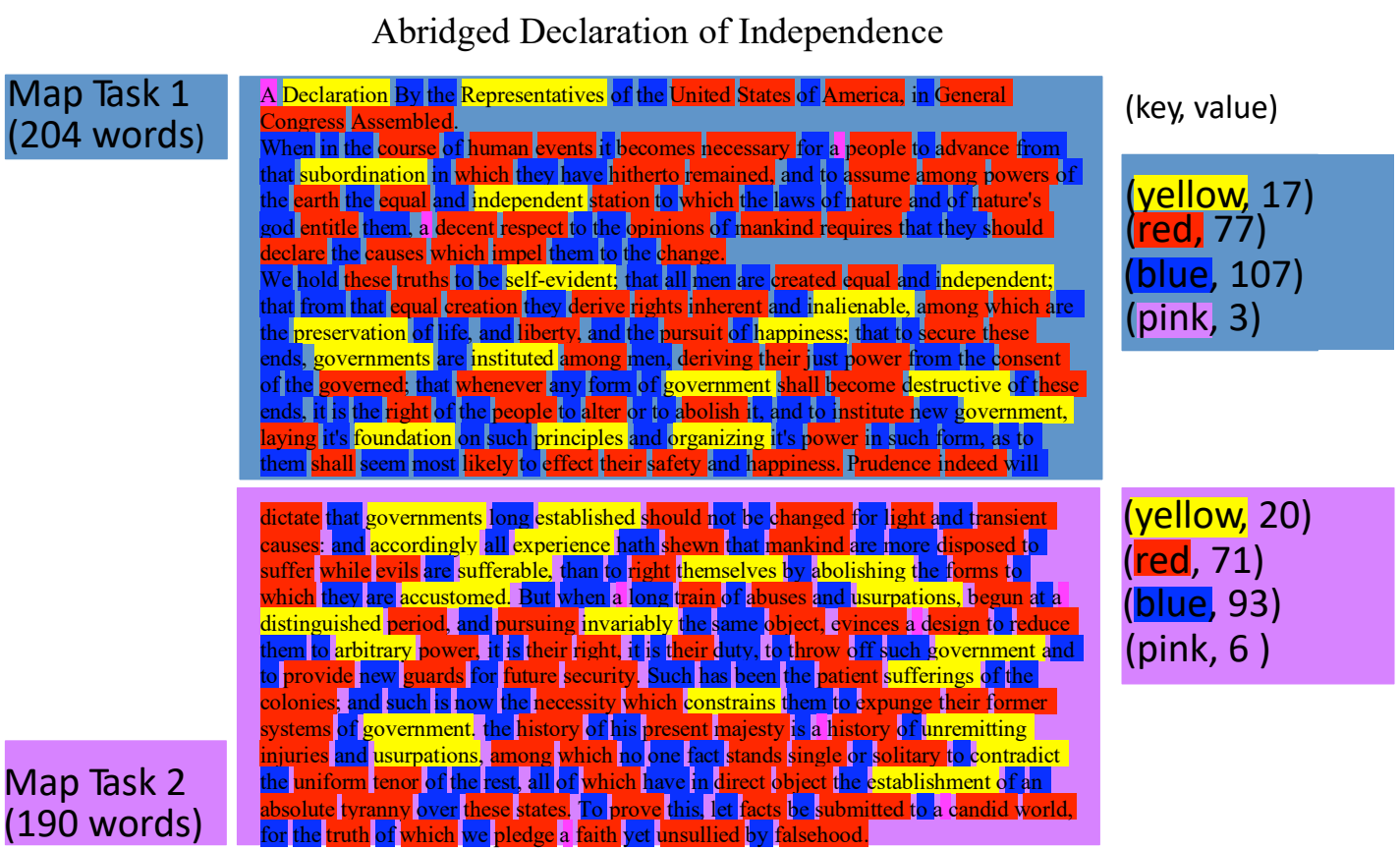
Tiny = Pink = 1 letter

Example: Word length histogram

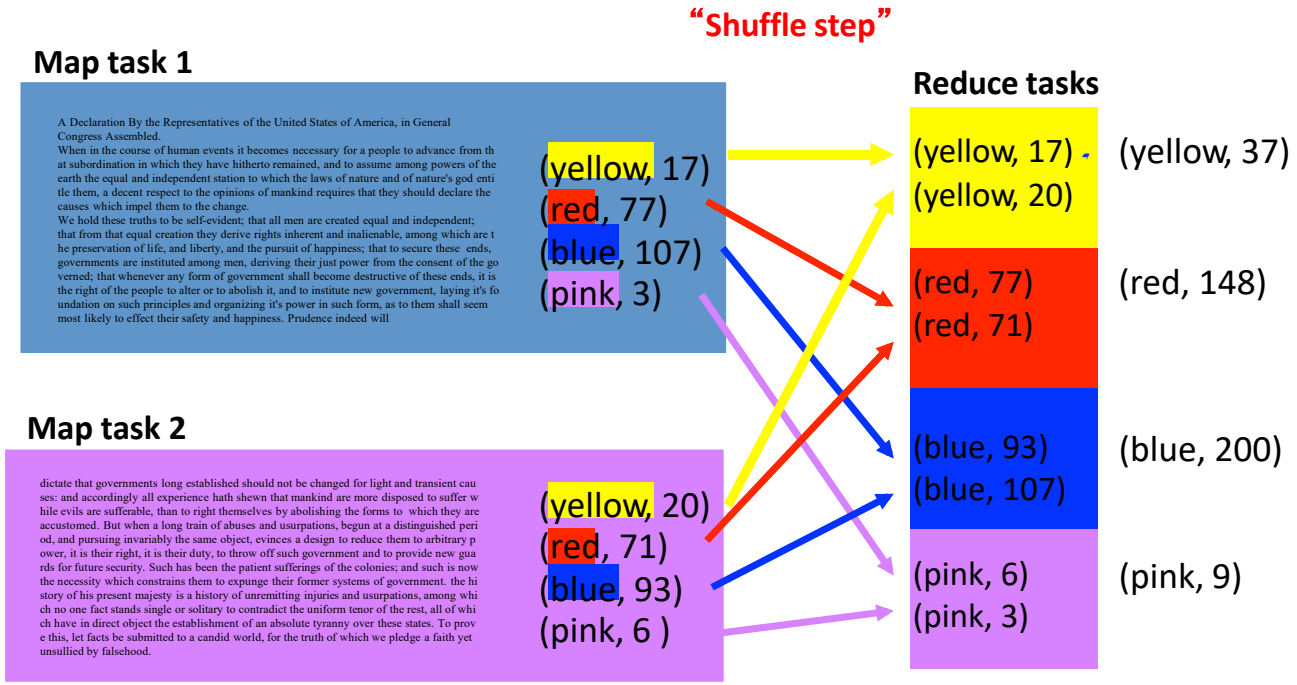
Split the document into chunks and process each chunk on a different computer



Example: Word length histogram



Example: Word length histogram



More examples: Build an inverted index

Input:

tweet1, (“I love pancakes for breakfast”)
tweet2, (“I dislike pancakes”)
tweet3, (“What should I eat for breakfast?”)
tweet4, (“I love to eat”)

Desired output:

“pancakes”, (tweet1, tweet2)
“breakfast”, (tweet1, tweet3)
“eat”, (tweet3, tweet4)
“love”, (tweet1, tweet4)
…

More examples: Relational join

Employee

Name	SSN
Sue	999999999
Tony	777777777

Assigned Departments

EmpSSN	DepName
999999999	Accounts
777777777	Sales
777777777	Marketing

Emplyee ⋈ Assigned Departments

Name	SSN	EmpSSN	DepName
Sue	999999999	999999999	Accounts
Tony	777777777	777777777	Sales
Tony	777777777	777777777	Marketing

Relational join in MapReduce: Before Map phase

Employee

Name	SSN
Sue	999999999
Tony	777777777

Assigned Departments

EmpSSN	DepName
999999999	Accounts
777777777	Sales
777777777	Marketing

Key idea: Lump all the tuples together into one dataset

Employee, Sue, 999999999
Employee, Tony, 777777777
Department, 999999999, Accounts
Department, 777777777, Sales
Department, 777777777, Marketing

What is this for?

Relational join in MapReduce: Map phase

Employee, Sue, 999999999
Employee, Tony, 777777777
Department, 999999999, Accounts
Department, 777777777, Sales
Department, 777777777, Marketing

key=999999999, value=(Employee, Sue, 999999999)
key=777777777, value=(Employee, Tony, 777777777)
key=999999999, value=(Department, 999999999, Accounts)
key=777777777, value=(Department, 777777777, Sales)
key=777777777, value=(Department, 777777777, Marketing)

why do we use this as the key?

Relational join in MapReduce: Reduce phase

key=999999999, values=[(Employee, Sue, 999999999),
(Department, 999999999, Accounts)]

Sue, 999999999, 999999999, Accounts

key=777777777, values=[(Employee, Tony, 777777777),
(Department, 777777777, Sales),
(Department, 777777777, Marketing)]

Tony, 777777777, 777777777, Sales
Tony, 777777777, 777777777, Marketing

Relational join in MapReduce, again

Order(orderid, account, date)

1, aaa, d1
2, aaa, d2
3, bbb, d3

LineItem(orderid, itemid, qty)

1, 10, 1
1, 20, 3
2, 10, 5
2, 50, 100
3, 20, 1

Map

tagged with relation name

Order

1, aaa, d1 → 1 : “Order”, (1,aaa,d1)
2, aaa, d2 → 2 : “Order”, (2,aaa,d2)
3, bbb, d3 → 3 : “Order”, (3,bbb,d3)

Line

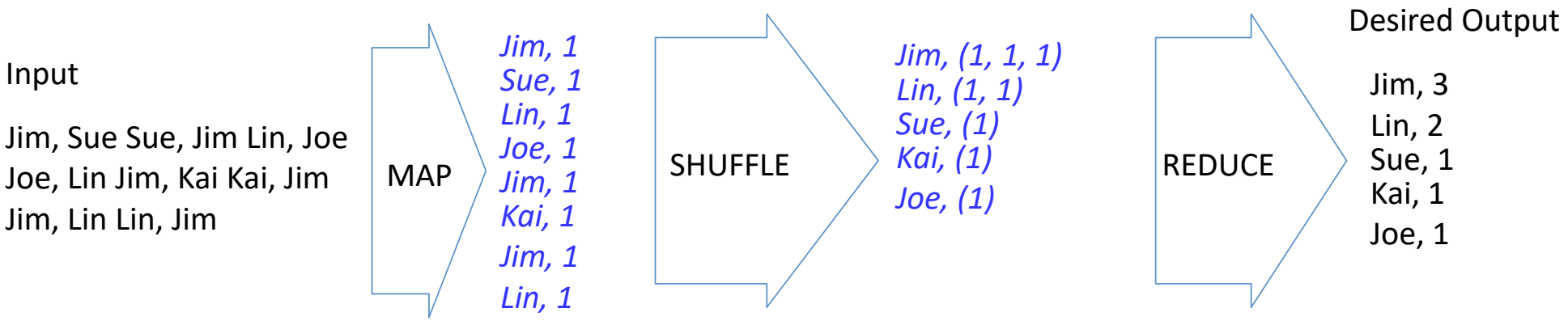
1, 10, 1 → 1 : “Line”, (1, 10, 1)
1, 20, 3 → 1 : “Line”, (1, 20, 3)
2, 10, 5 → 2 : “Line”, (2, 10, 5)
2, 50, 100 → 2 : “Line”, (2, 50, 100)
3, 20, 1 → 3 : “Line”, (3, 20, 1)

Reducer for key 1

“Order”, (1,aaa,d1)
“Line”, (1, 10, 1)
“Line”, (1, 20, 3)

(1, aaa, d1, 1, 10, 1)
(1, aaa, d1, 1, 20, 3)

Simple social network analysis: Count friends



Matrix multiplication

$$\begin{vmatrix} 1 & 3 & 4 & -2 \\ 6 & 2 & -3 & 1 \end{vmatrix} \times \begin{vmatrix} 1 & -2 \\ 4 & 3 \\ -3 & -2 \\ 0 & 4 \end{vmatrix} = \begin{vmatrix} 1 & -9 \\ 23 & 4 \end{vmatrix}$$

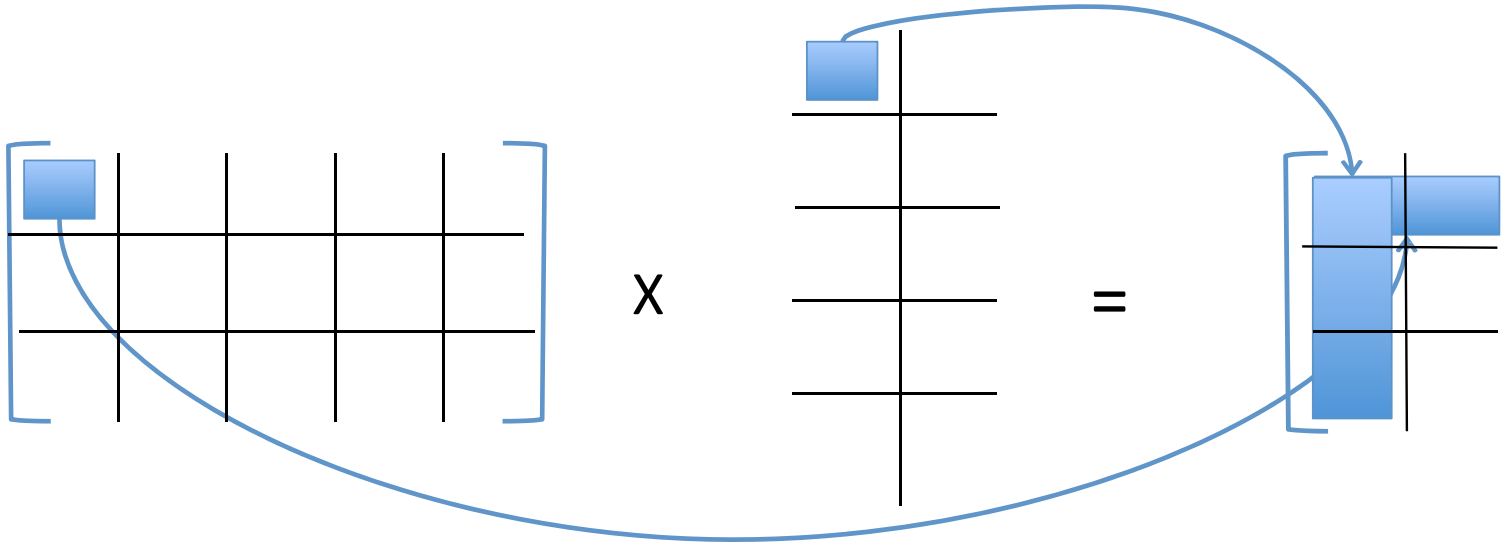
Matrix multiply in MapReduce

- In the map phase:
  - for each element (i,j) of A, emit ((i,k), A[i,j]) for k in 1..N
  - for each element (j,k) of B, emit ((i,k), B[j,k]) for i in 1..L
- In the reduce phase, emit
  - key = (i,k)
  - value = Sum_j (A[i,j] * B[j,k])

Notation

C = A X B
A has dimensions L,M
B has dimensions M,N

Matrix multiply in MapReduce



- One reducer per output cell
- Each reducer computes Sum_j (A[i,j] * B[j,k])

Taxonomy of parallel architectures

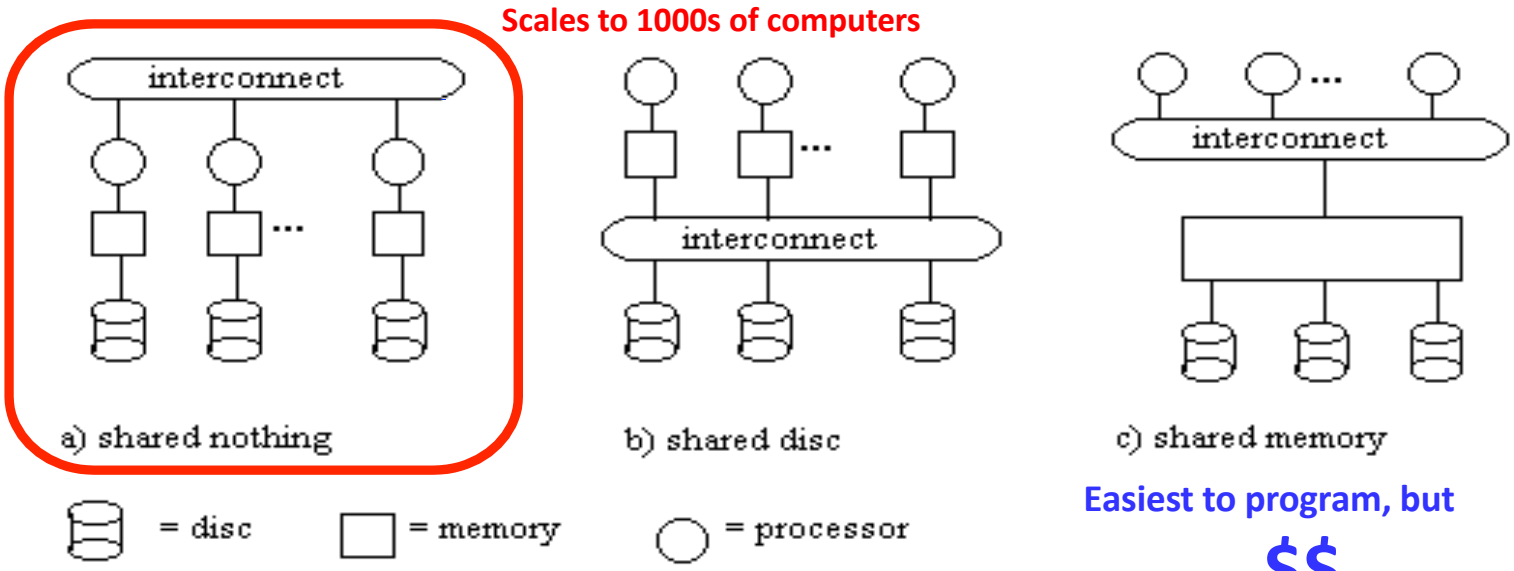


Fig. 3.1 Logical multi-processor database designs (diagram after [DEWI92])

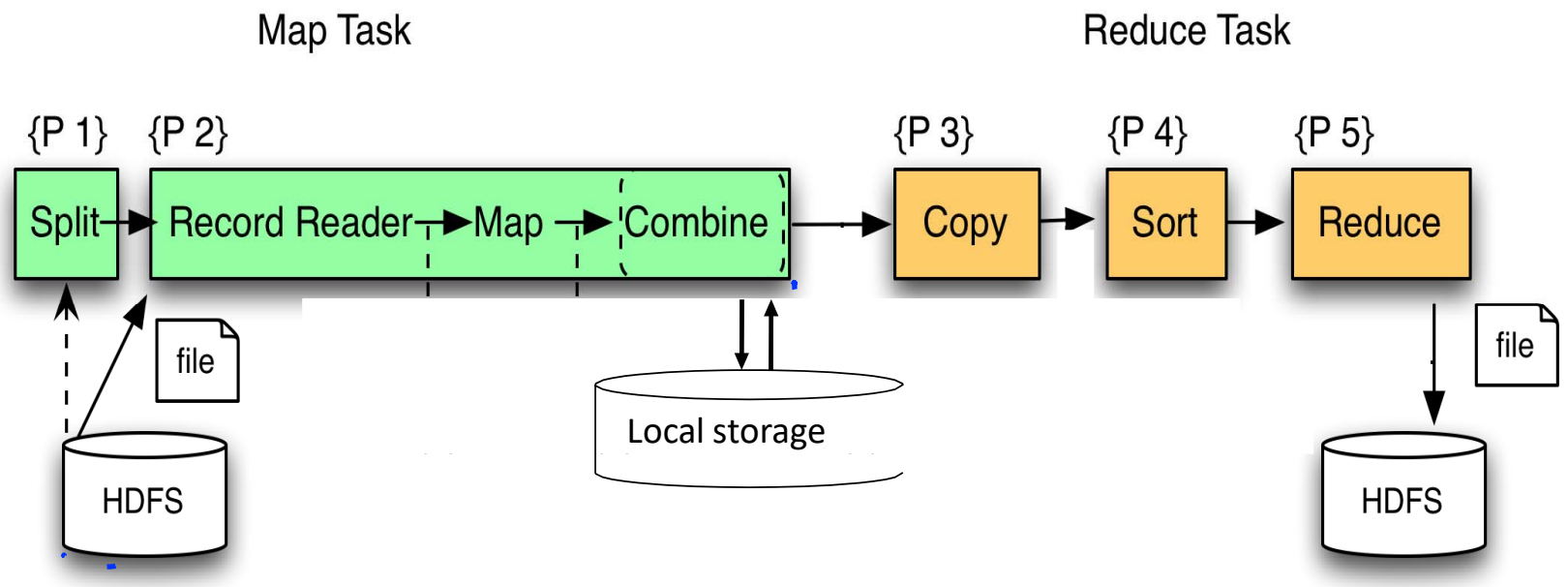
Cluster computing

- Large number of commodity servers, connected by commodity network
- Rack: holds a small number of servers
- Data center: holds many racks
- Massive parallelism:
  - 100s, 1000s, or 10,000s servers
- Failure:
  - If mean-time-between-failure is 1 year,
  - then, 10,000 servers have one failure per hour

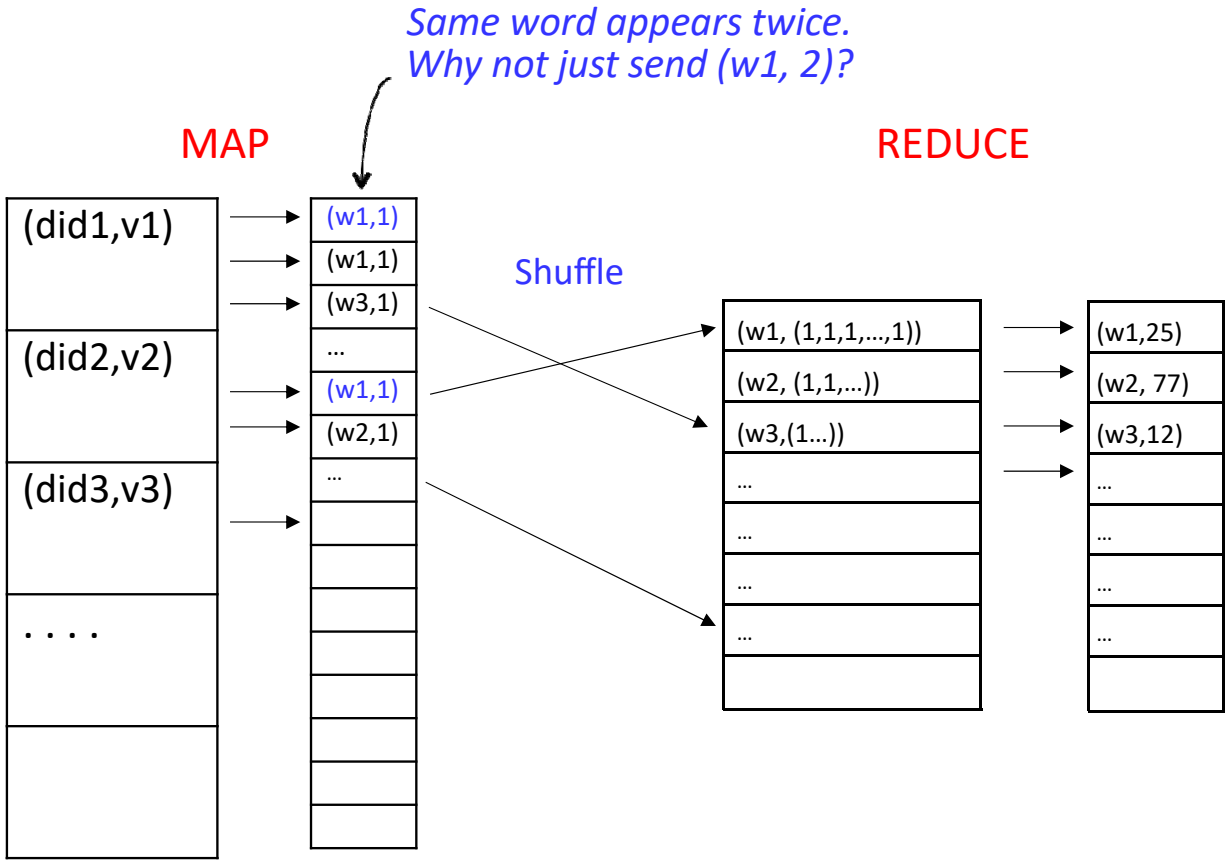
Distributed file system(DFS)

- For very large files: TBs, PBs
- Each file is partitioned into chunks, typically 64MB
- Each chunk is replicated several times (>=3) on different racks for fault tolerance
- Implementations:
  - Google's DFS: GFS, proprietary
  - Hadoop's DFS: HDFS, open source

MapReduce phases



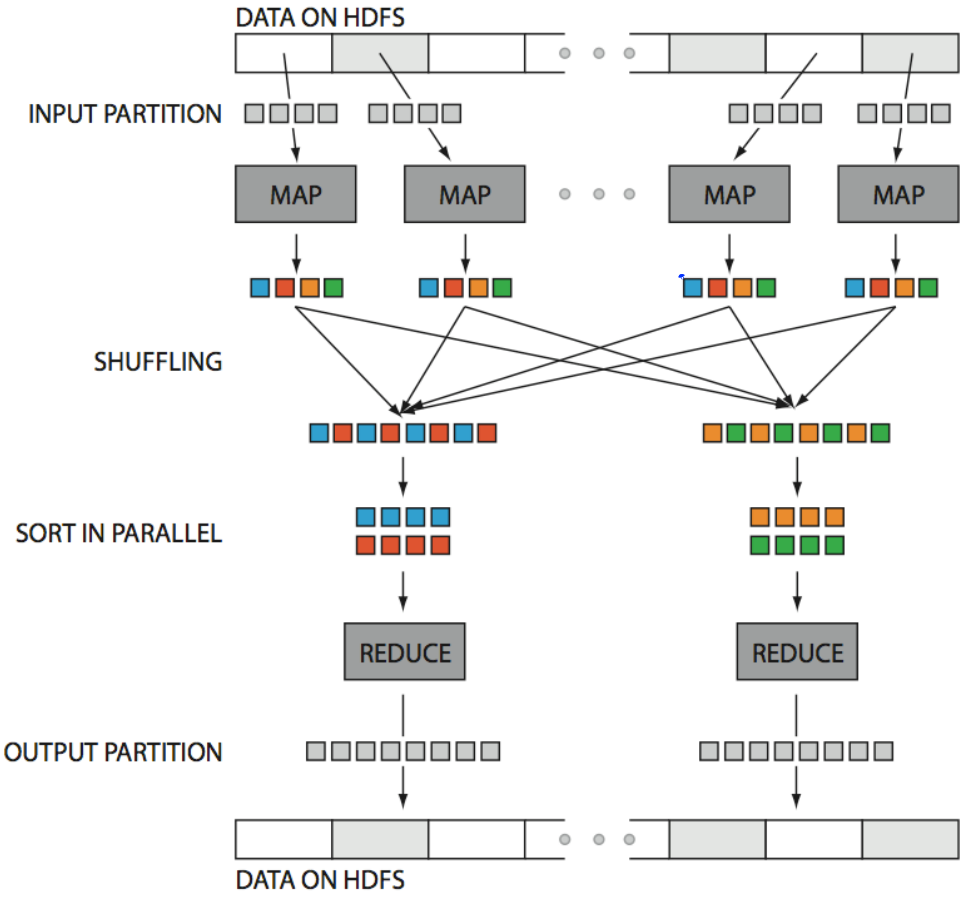
MapReduce phases



Adding a combine after Map before Reduce

```
map(String in_key, String in_value):
  // in_key: document name
  // in_value: document contents
  For each word w in in_value:
        Emit(w,1);

combine(String intermediate_key, Iterator intermediate_values):
  // intermediate_key: word
  // intermediate_values: ???
  Int result = 0;
  For each v in intermediate_values:
     result += v;
  Emit(intermediate_key, result);
```


DATA ON HDFS
INPUT PARTITION
MAP
MAP
MAP
MAP
SHUFFLING
SORT IN PARALLEL
REDUCE
REDUCE
OUTPUT PARTITION
DATA ON HDFS

Large-scale data processing

- Many tasks process big data, produce big data

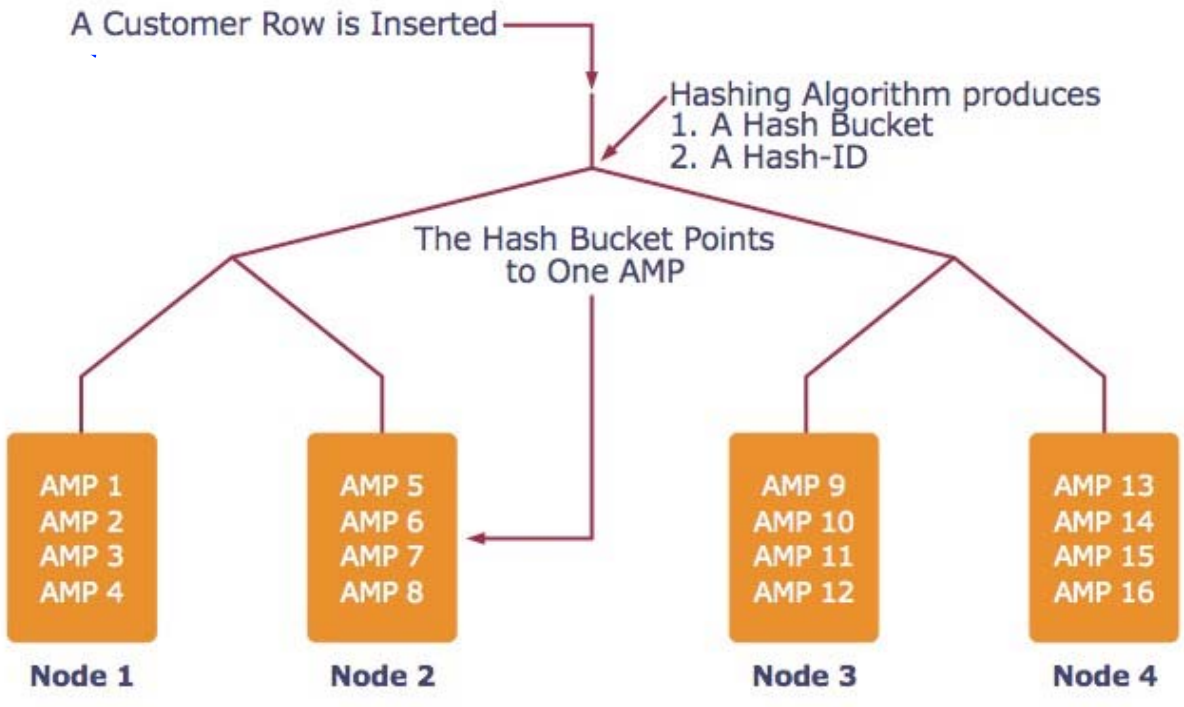
- Want to use hundreds or thousands of CPUs
  - ... but this needs to be easy
  - Parallel databases exist, but they are expensive, difficult to set up, and do not necessarily scale to hundreds of nodes.

- MapReduce is a *lightweight* framework, providing:
  - Automatic parallelization and distribution
  - Fault-tolerance
  - Status and monitoring

Two notions of parallel query processing

- “Distributed Query”
  - Rewrite the query as a union of subqueries
  - Workers communicate through standard interfaces, so compatible with federated, heterogeneous, or distributed databases
- “Parallel Query”
  - Each operator is implemented with a parallel algorithm

Parallel Query Example: Teradata

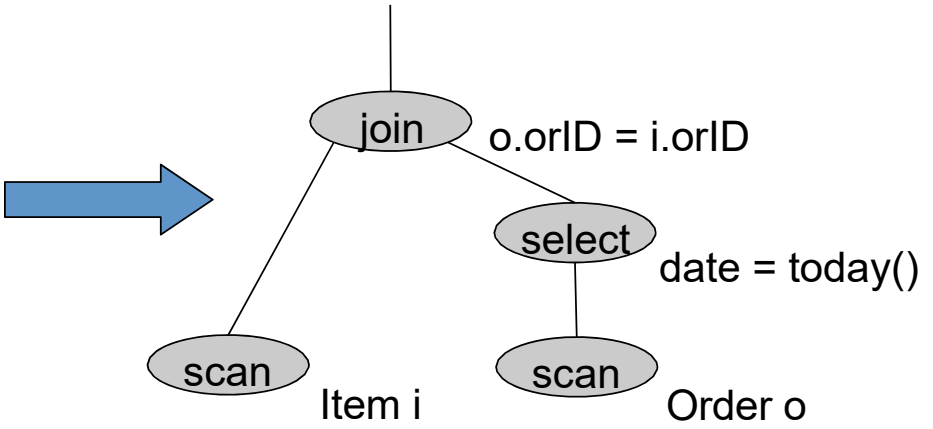


AMP = unit of parallelism

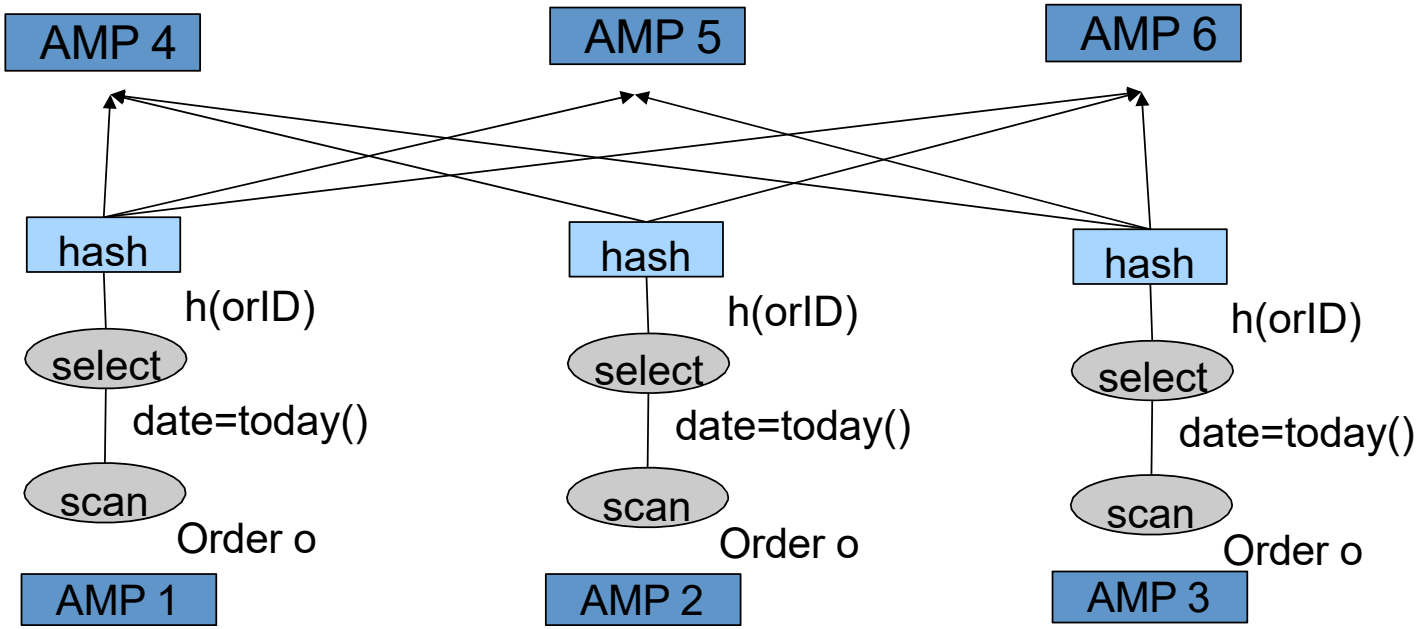
Key Idea: Declarative Languages

Find all orders from today, along with the items ordered

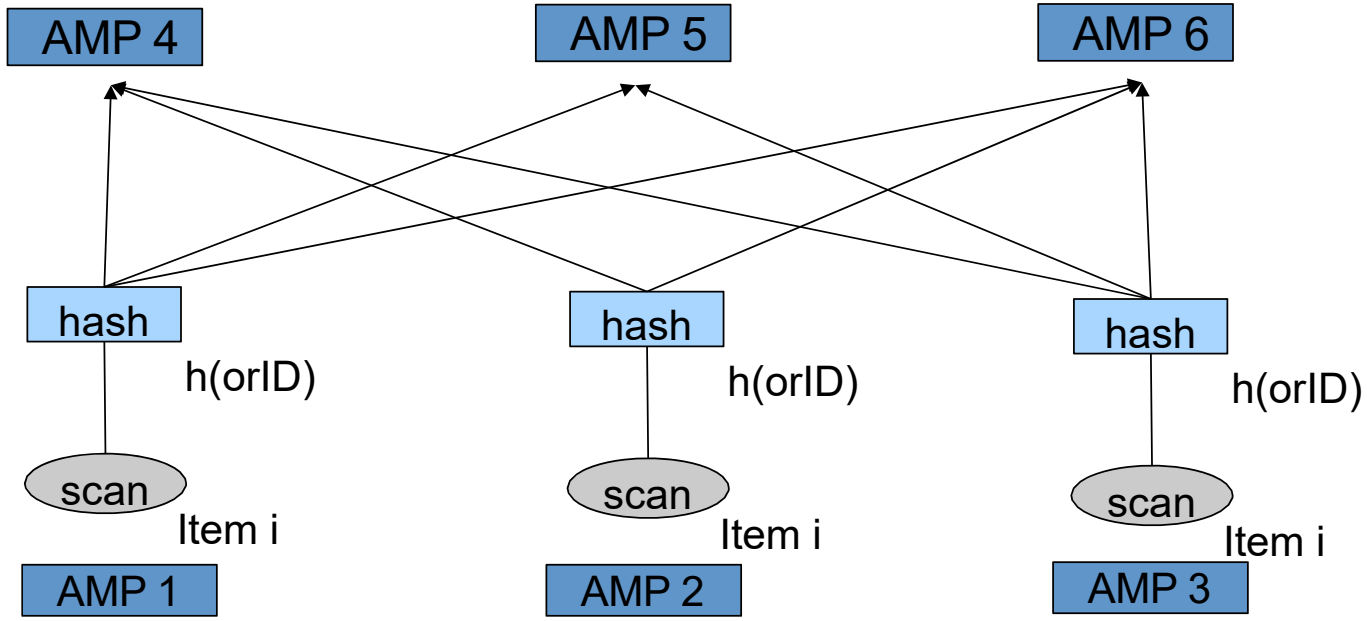
```
SELECT *
  FROM Order o, Item i
 WHERE o.orID = i.orID
   AND o.date = today()
```



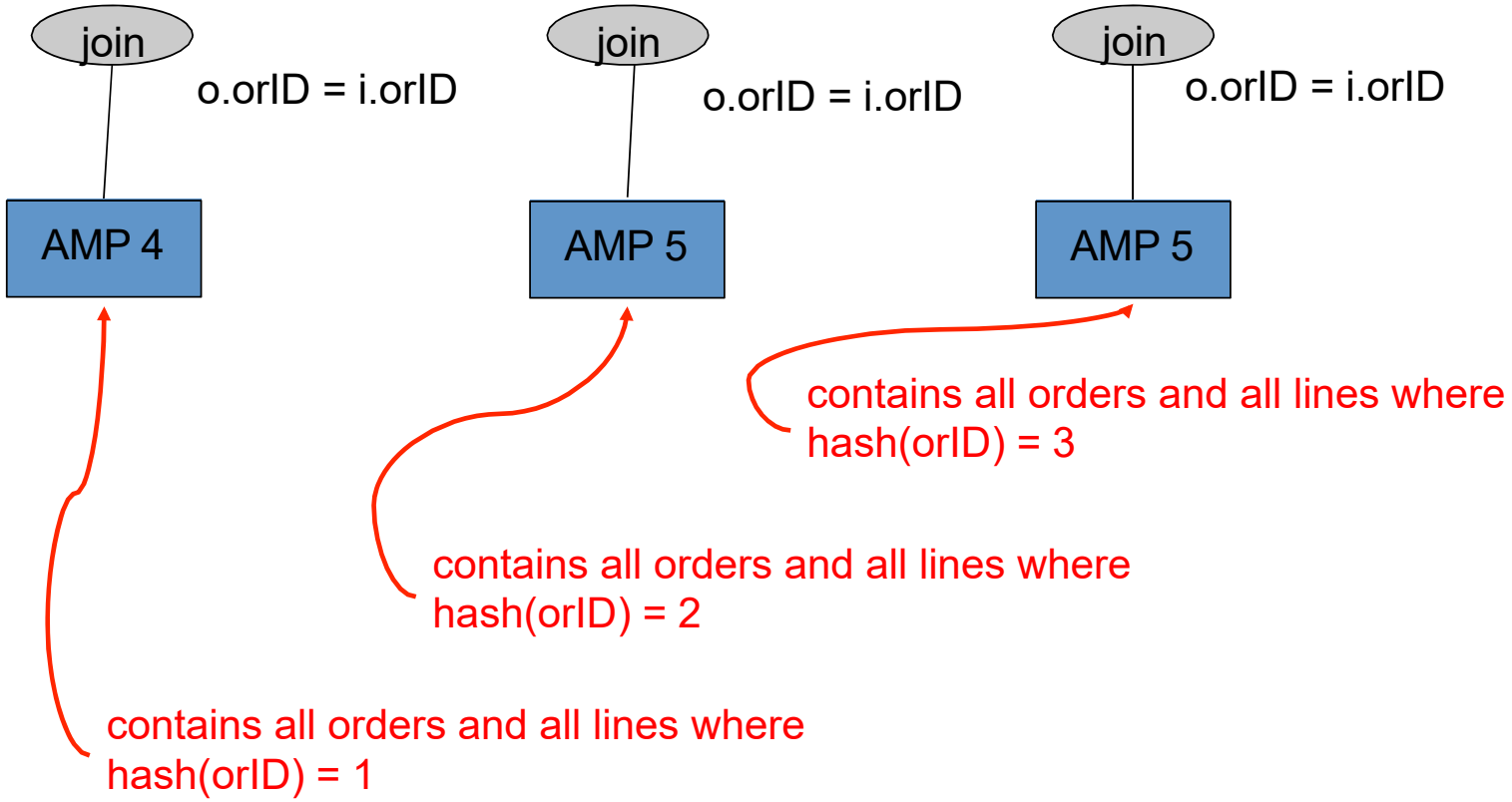
Example System: Teradata



Example System: Teradata



Example System: Teradata



MapReduce Extensions and Contemporaries

- Pig (Yahoo, available open source)
  - Relational Algebra over Hadoop
- HIVE (Facebook, available open source)
  - SQL over Hadoop
- Impala (Cloudera)
  - SQL over HDFS; uses some HIVE code
- Dryad (Microsoft, sadly not available)
  - Relational Algebra