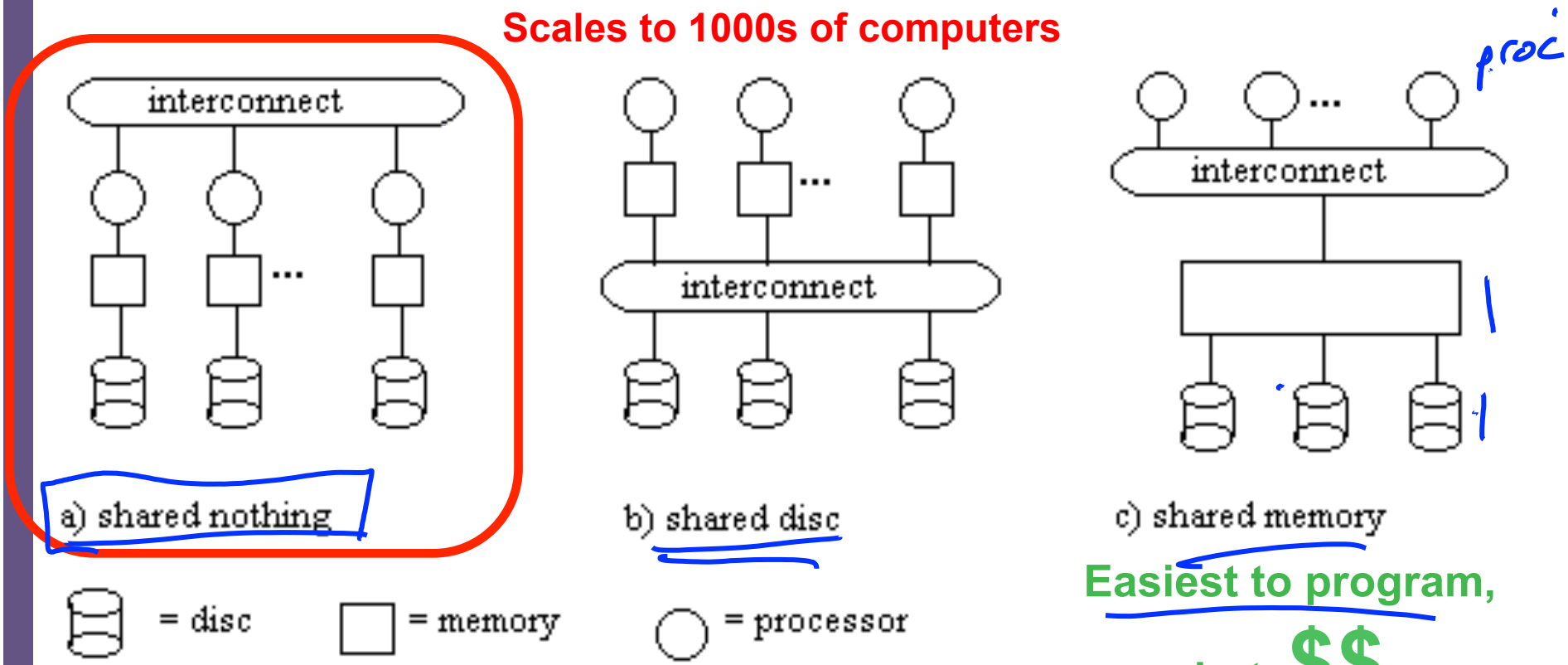
Taxonomy of Parallel Architectures

Scales to 1000s of computers



Easiest to program,

but $$

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

Cluster Computing

- Large number of commodity servers, connected by high speed, commodity network
- Rack: holds a small number of servers
- Data center: holds many racks

Cluster Computing

- Massive parallelism:
  - 100s, or 1000s, or 10000s servers
  - Many hours
- Failure:
  - If ~~medium~~ on-time-between-failure is 1 year
  - Then 10000 servers have one failure / hour

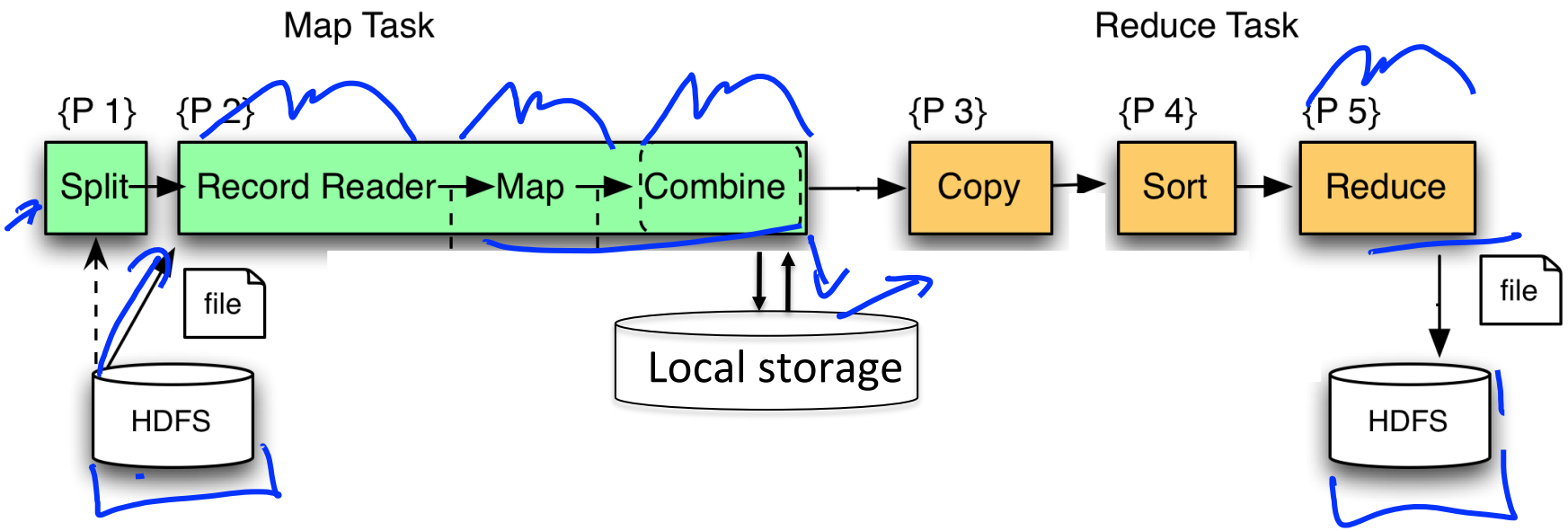
Distributed File System (DFS)

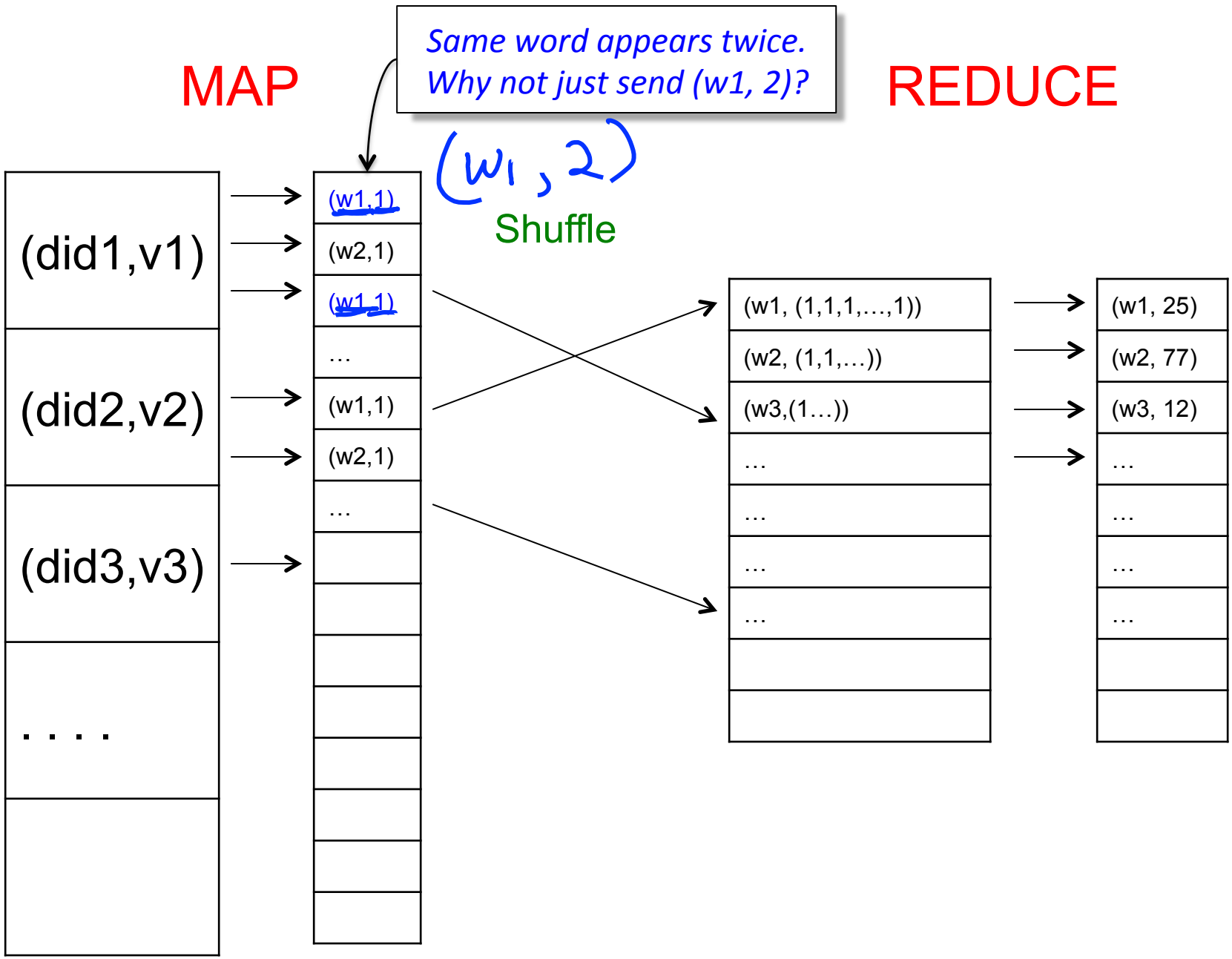
- For very large files: TBs, PBs
- Each file is partitioned into *chunks*, typically 64MB
- Each chunk is replicated several times (≥3), on different racks, for fault tolerance
- Implementations:
  - Google's DFS: GFS, proprietary
  - Hadoop's DFS: HDFS, open source

Implementation

- There is one master node
- Master partitions input file into *M splits*, by key
- Master assigns *workers* (=servers) to the *M map tasks*, keeps track of their progress
- Workers write their output to local disk, partition into *R regions*
- Master assigns workers to the *R reduce tasks*
- Reduce workers read regions from the map workers' local disks

MapReduce Phases




MAP
Same word appears twice.
Why not just send (w1, 2)?
REDUCE
(w1, 2)
Shuffle
(did1,v1)
(did2,v2)
(did3,v3)
. . . .
(w1,1)
(w2,1)
(w1,1)
…
(w1,1)
(w2,1)
…
(w1, (1,1,1,…,1))
(w2, (1,1,…))
(w3,(1…))
…
…
…
…
(w1, 25)
(w2, 77)
(w3, 12)
…
…
…
…

Adding a combiner

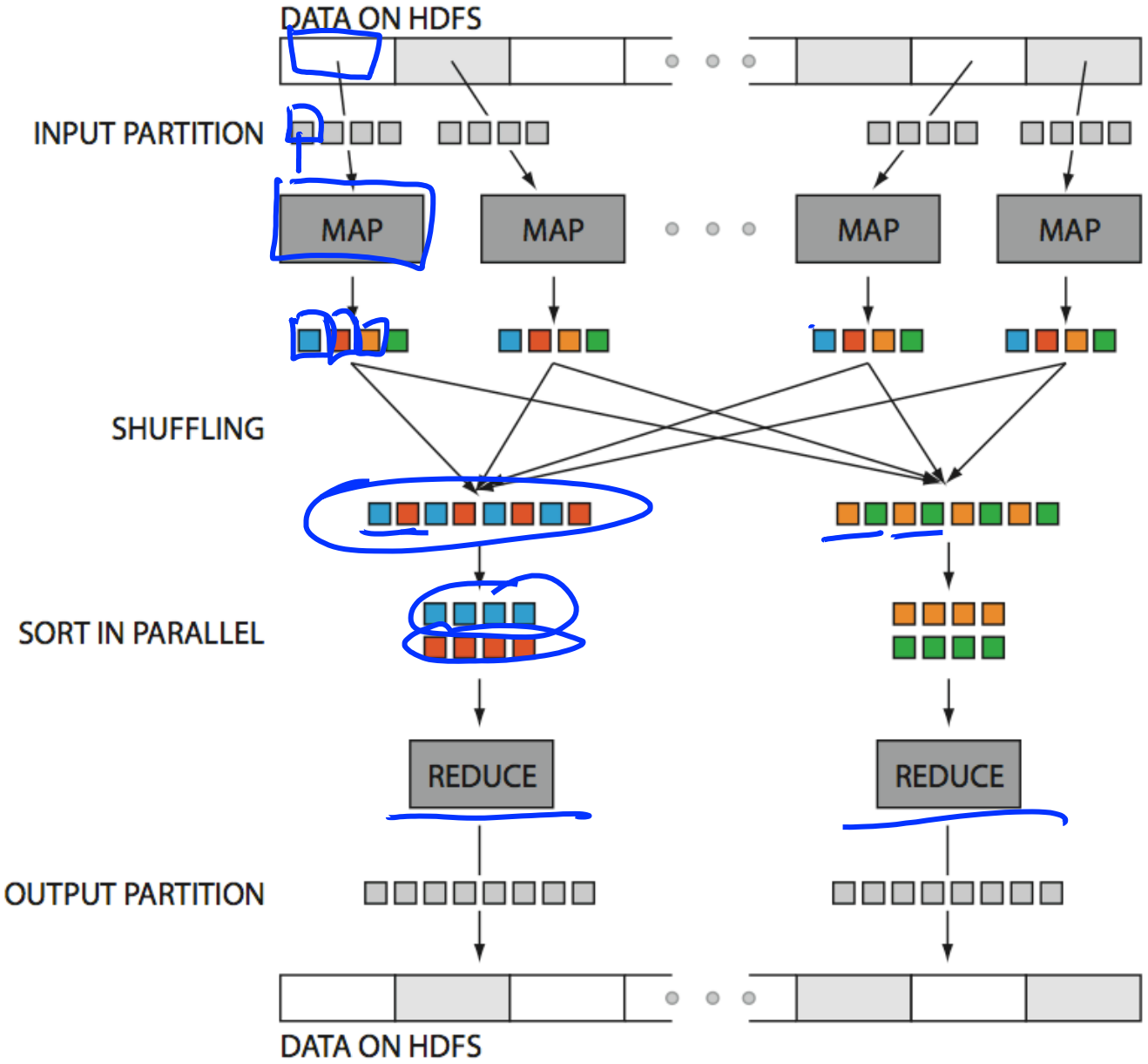
```
map(String input_key, String input_value):
  // input_key: document name
  // input_value: document contents
  for each word w in input_value:
      EmitIntermediate(w, 1);

combine(String intermediate_key, Iterator intermediate_values)
  returns (intermediate_key, intermediate_value)

reduce(String intermediate_key, Iterator intermediate_values):
  // output_key: word
  // output_values: ????
  int result = 0;
  for each v in intermediate_values:
      result += v;
  Emit(result);
```

slide source: Google, Inc.

Hadoop in One Slide



src: Huy Vo, NYU Poly