

Map Reduce

Google : GFS, MapReduce Chubby
Storage compute R. Coord.

Jeff Dean Sanjay Ghemawat.

Why Distributed Computing

Sort $\frac{1TB}{60MB/s}$ $\frac{250MB-500MB/s}{\sim 1 Day}$ $2 \times \frac{10^3}{10^6} \frac{P}{G}$
Google > 100 billion webpages. $10^9 M$
 $10^2 \times 10^9 * 20KB = 2 \times 10^{12} KB$

Solution? 2000 Machines

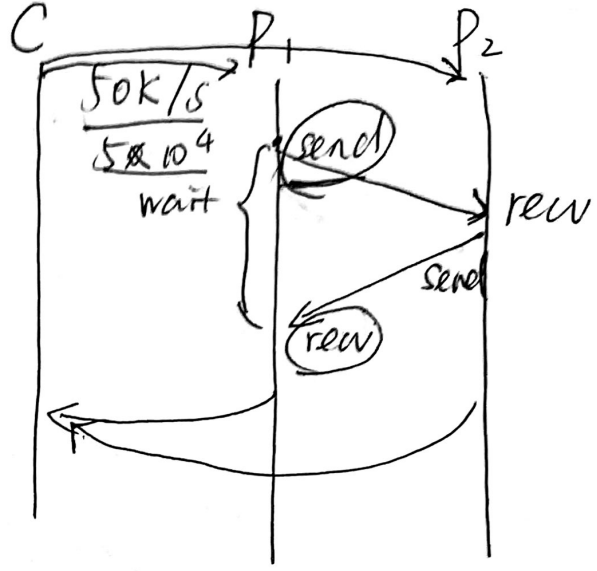
Data centers. { Temp. Cloud computing
Viber - - - Amazon EC2
Linode.

Assumptions $1 \rightarrow 2000 \times$

D.C. / Programming

$$2 \text{ GHz} \quad \frac{2 \times 10^9}{20} = 10^8$$

1. send recv / slow + complex
2. Coord.
3. Debug.
4. Optimize.
5. Failure



Before MapReduce World.

1. Systems Reserch Comm
D.C. = multi-thread ~~com~~ program

D.S.M



2. HPC Fortran

MPI / OpenMP.

commodity

non-com - - -

3. No Fault Tolerance.

Map Reduce.

Programming framework / model
input / output.

no persistence.

As a ~~service~~ ^{process}

As a library

main

MySQL

Level / Rocks / Ber

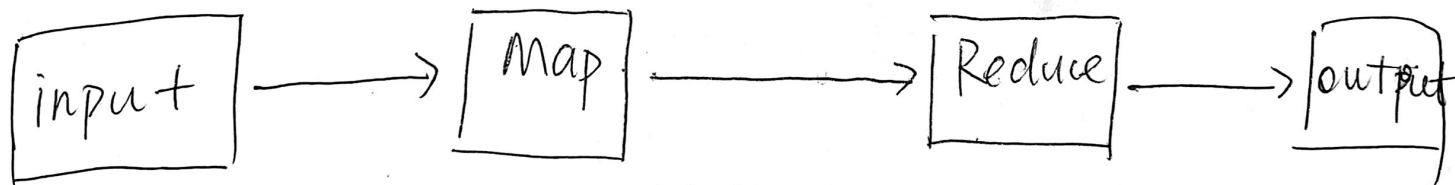
Dynamic Plugin C++
Link

Map Reduce

map
reduce

Auto parallel.
load balance.
locality failures.

Design / Concept.

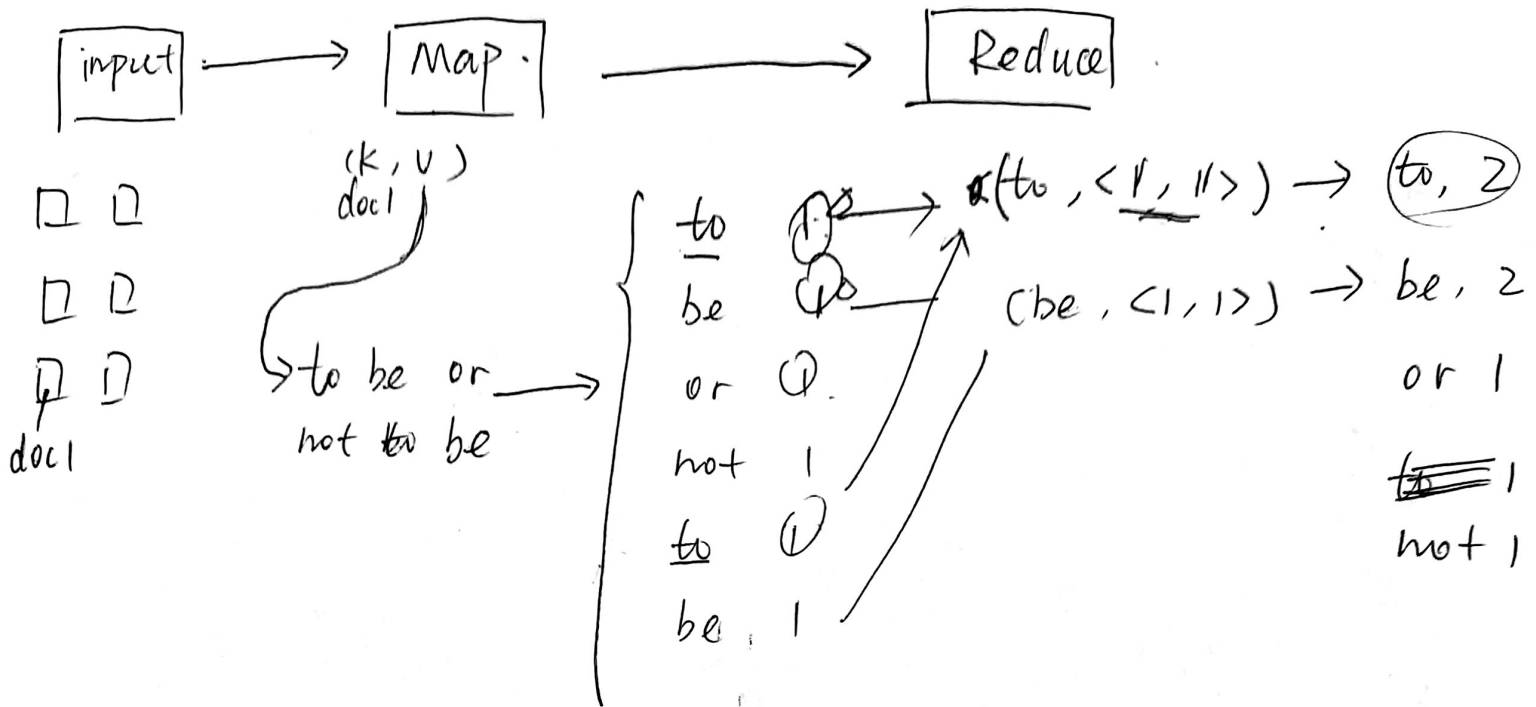


$(k, v) \rightarrow \langle (k', v'), * \rangle$

UDF (map)

$k', \langle v', * \rangle \rightarrow k', v''$
UDF reduce

E.g. Word Count



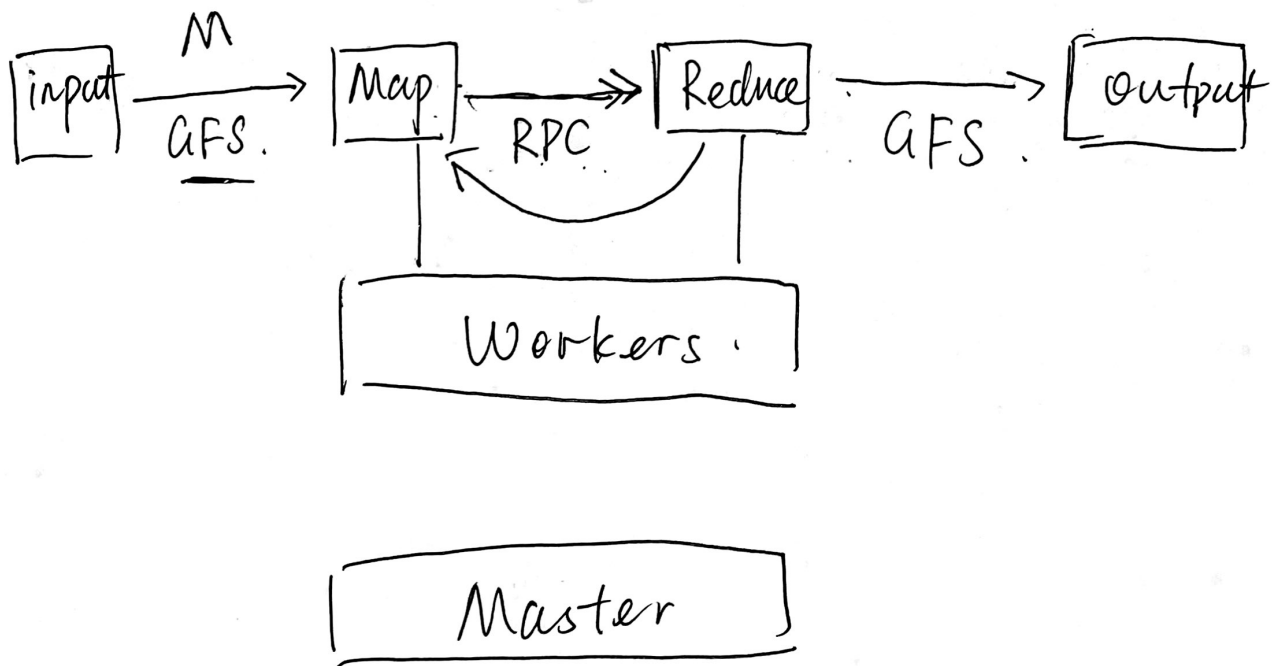
More Application

1. sorting

2. grep

3. reversed links

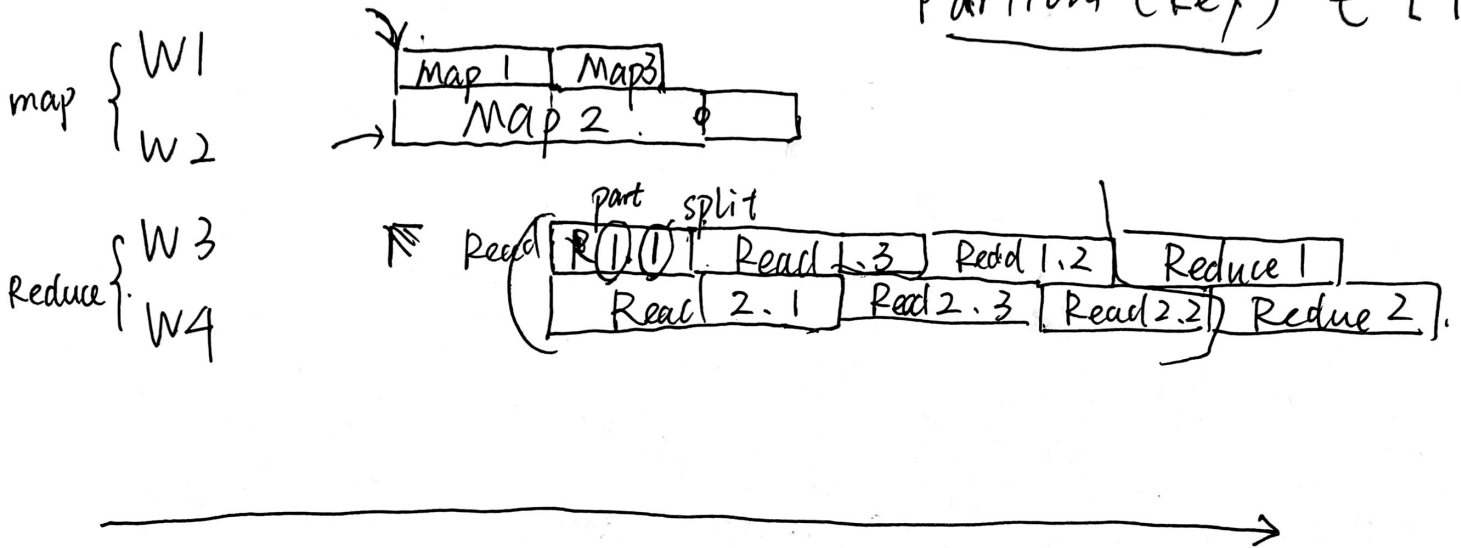
Map Reduce Distributed Imple.



Load Balancing / Pipelining.

$$M = 3 \quad R = 2.$$

Partitions (key) $\in [1..R]$



Fault Tolerance.

Failures.

Networks.

Servers.

Permanet.

M.R.

Master .

Abort .

Seldom : Disk

Worker . Failure . : Re-execute .

Map : ~~to~~ ✓

Reduce : Atomic GFS functions .

Slow worker = failed worker