

Google MapReduce

DS 5110: Big Data Systems (Spring 2023)

Lecture 3b

Yue Cheng



UNIVERSITY
of VIRGINIA

WC.

Applications



Scalable computing engines

MR.

Scalable storage systems

GFS.



Datacenter infrastructure



The big picture (motivation)

- Datasets are **too big** to process using a single computer

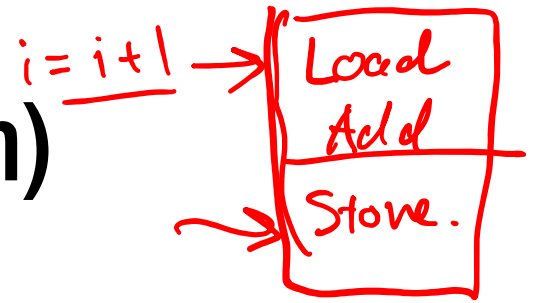
The big picture (motivation)

- Datasets are **too big** to process using a single computer
- Good parallel processing engines are **rare** (back then in the late 90s)

Decomposition

MPI:
Message Passing Interface.

The big picture (motivation)



- Datasets are **too big** to process using a single computer
- Good parallel processing engines are **rare** (back then in the late 90s)

Atomicity.

expressive.

- Want a parallel processing framework that:
 - is general (works for many problems)
 - is easy to use (no locks, no need to explicitly handle communication, no race conditions)
 - can automatically parallelize tasks
 - can automatically handle machine failures

Context (Google circa 2000)

- Starting to deal with **massive** datasets
- But also addicted to cheap, unreliable hardware
 - Young company, expensive hardware not practical
- Only a few expert programmers can write distributed programs to process them
 - Scale so large jobs can complete before failures



Context (Google circa 2000)

- Starting to deal with **massive** datasets
- But also addicted to cheap, unreliable hardware
 - Young company, expensive hardware not practical
- Only a few expert programmers can write distributed programs to process them
 - Scale so large jobs can complete before failures

→ **Key question:** how can every Google engineer be imbued with the ability to write **parallel, scalable, distributed, fault-tolerant** code?

- **Solution:** **abstract out** the redundant parts

→ **Restriction:** relies on job semantics, so restricts which problems it works for

Application: Word Count

cmd tools.

→ cat data.txt

| tr -s '[:punct:][:space:]' '\n'
| sort | uniq -c

→ SELECT count(word), word FROM data
GROUP BY word

Deal with multiple files?

Deal with multiple files?

1. Compute word counts from individual files

Deal with multiple files?

1. Compute word counts from individual files
2. Then merge intermediate output

Deal with multiple files?

1. Compute word counts from individual files
2. Then merge intermediate output
3. Compute word count on merged outputs

What if the data is too big to fit in one computer?

What if the data is too big to fit in one computer?

BSP.

1. In parallel, send to worker:
 - Compute word counts from individual files
 - Collect results, wait until all finished

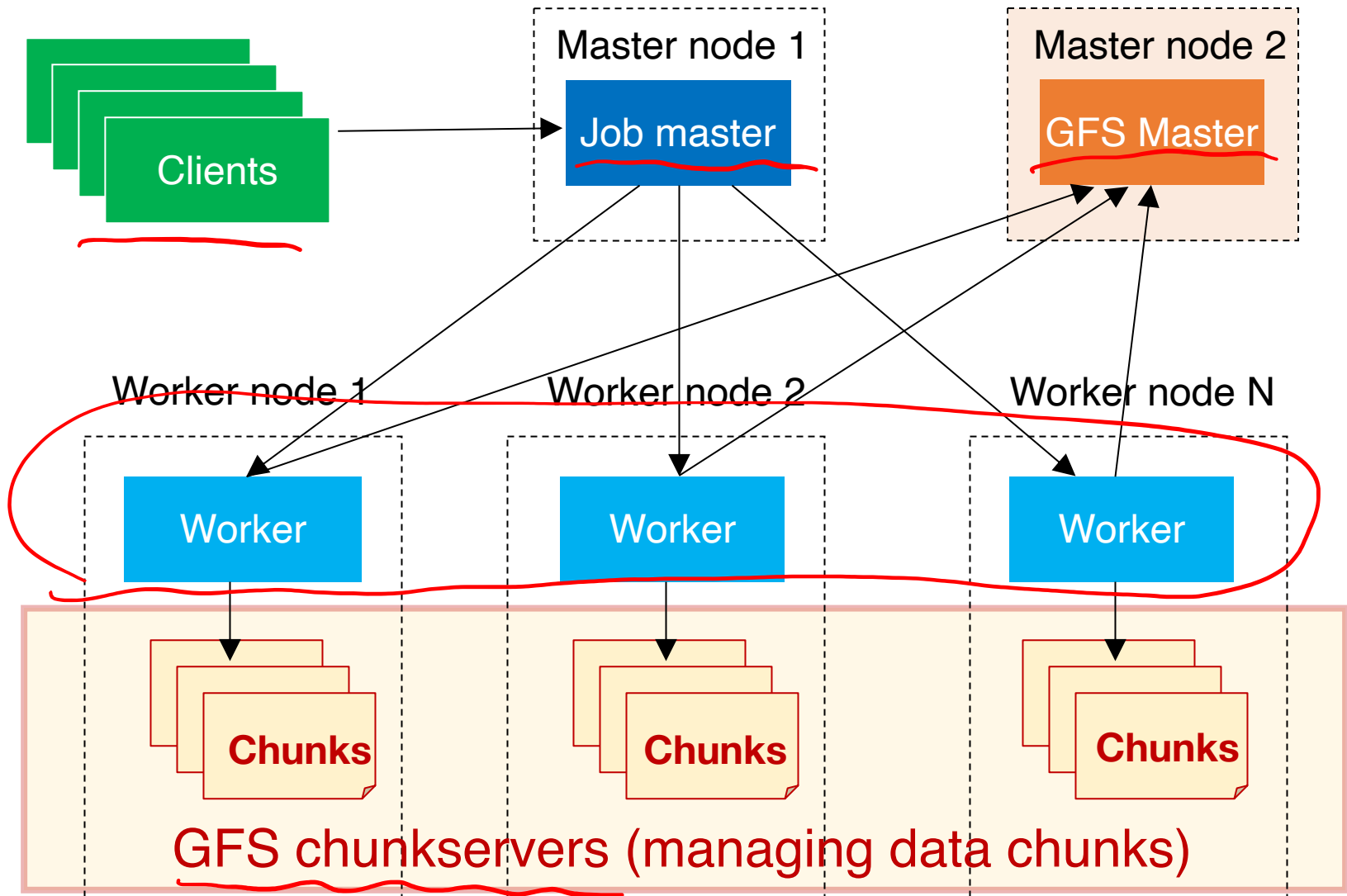
What if the data is too big to fit in one computer?

1. In parallel, send to worker:
 - Compute word counts from individual files
 - Collect results, wait until all finished
2. Then merge intermediate output

What if the data is too big to fit in one computer?

1. In parallel, send to worker:
 - Compute word counts from individual files
 - Collect results, wait until all finished
2. Then merge intermediate output
3. Compute word count on merged intermediates

MapReduce+GFS: Put everything together



MapReduce: Programming interface

- map(k1, v1) → list(k2, v2)
 - Apply function to (k1, v1) pair and produce set of intermediate pairs (k2, v2)
 - ↓
Intermediate Step → Shuffle.
↓
- reduce(k2, list(v2)) → list(k3, v3)
 - Apply aggregation (reduce) function to values
 - Output results

MapReduce: Word Count

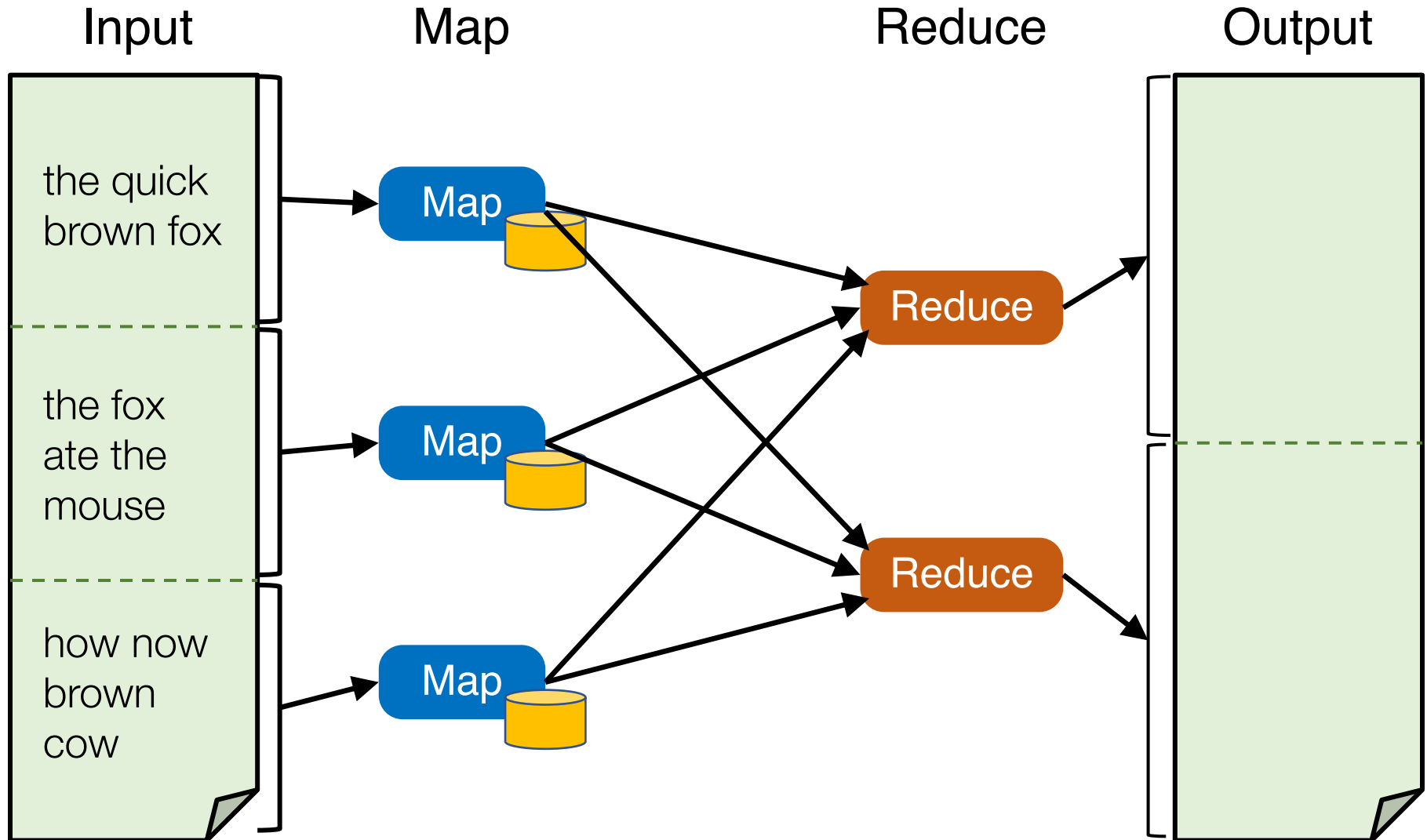
Ln. Line str.

```
→ map(key, value):  
    for each word w in value:  
        EmitIntermediate(w, "1");  
Shuffle. w List("1", "1", "1", ...)
```

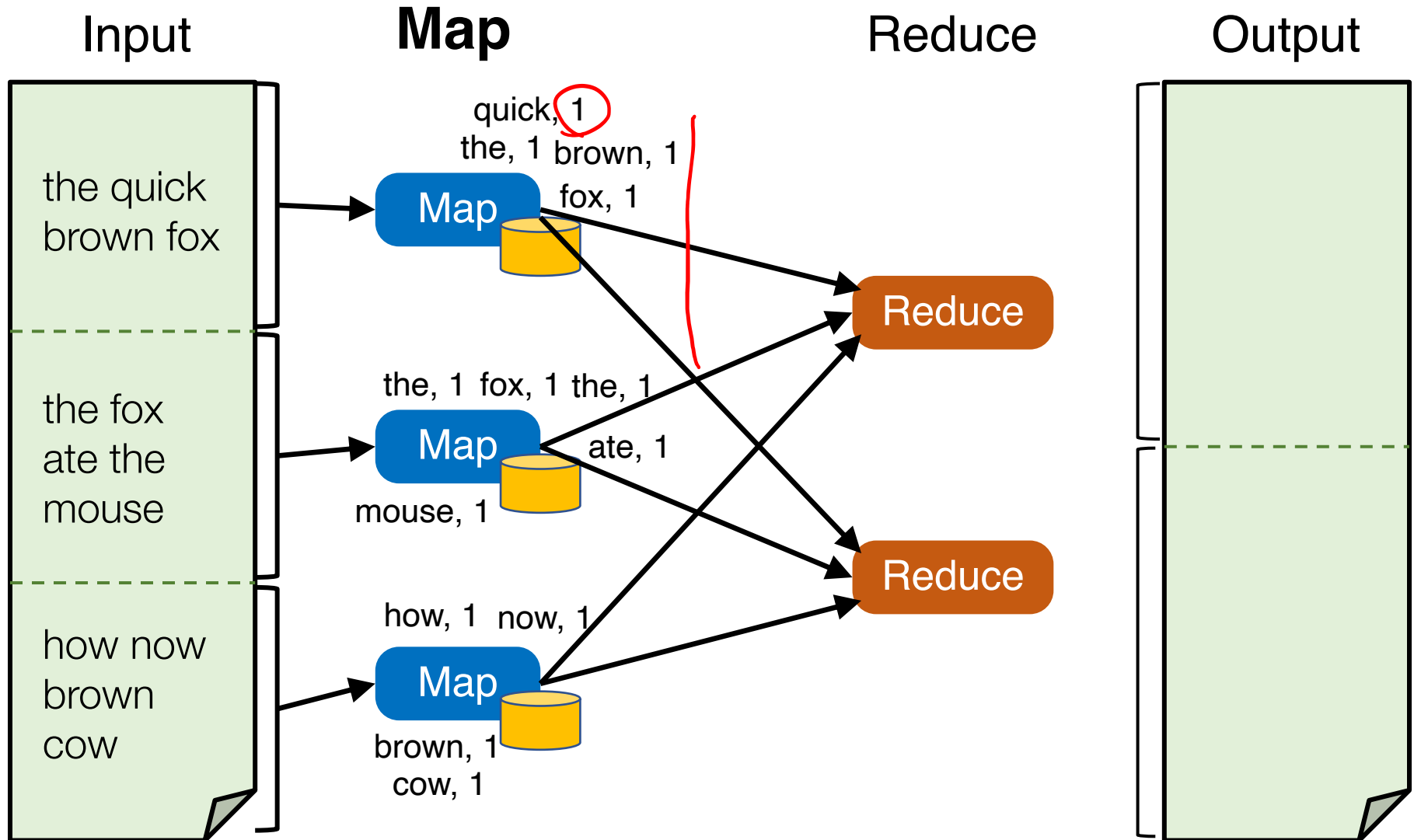
```
→ reduce(key, values):  
    int result = 0;  
    for each v in values:  
        result += ParseInt(v);  
    Emit(AsString(result));
```

Word Count execution

GFS.



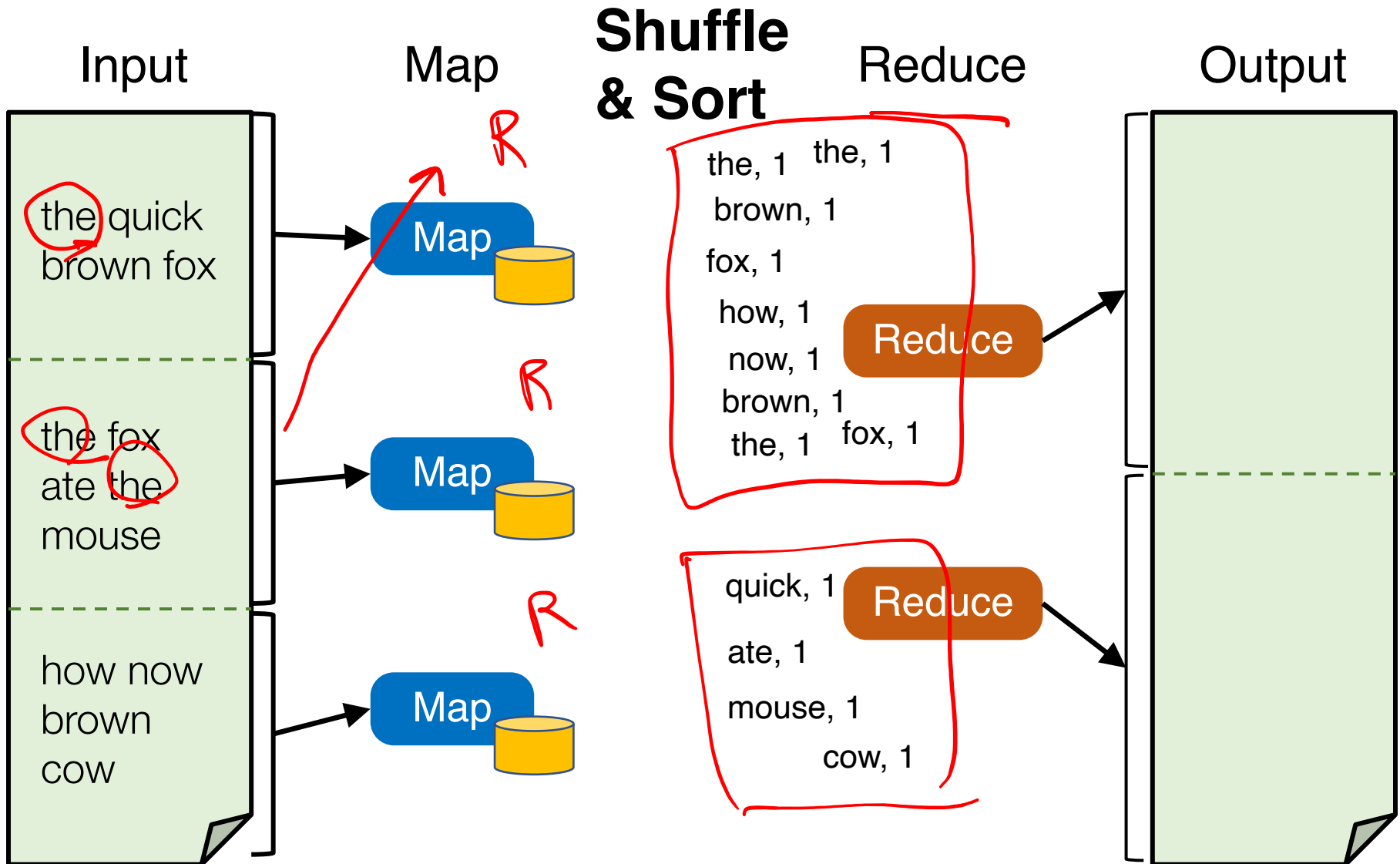
Word Count execution



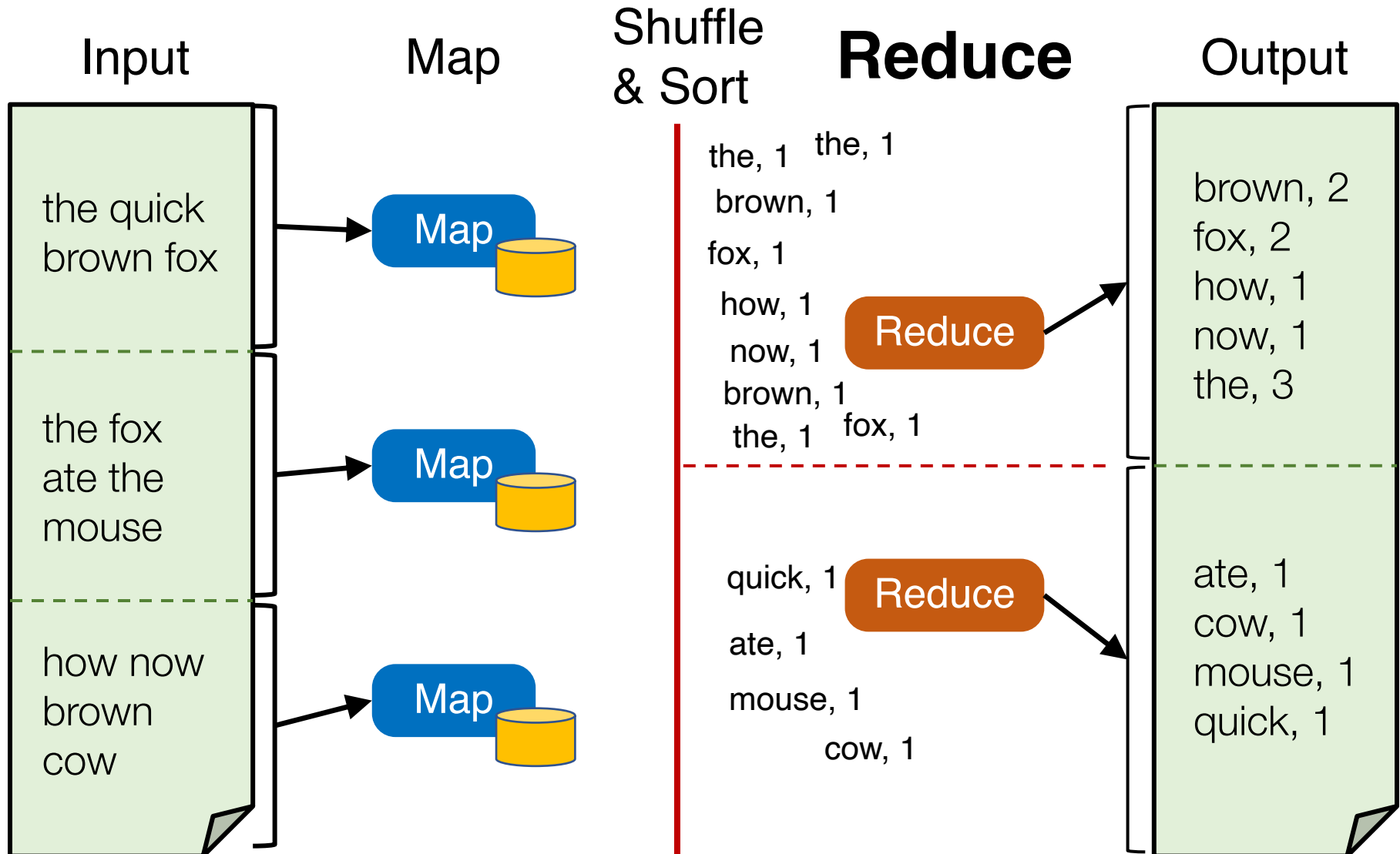
Word Count execution

hash func.

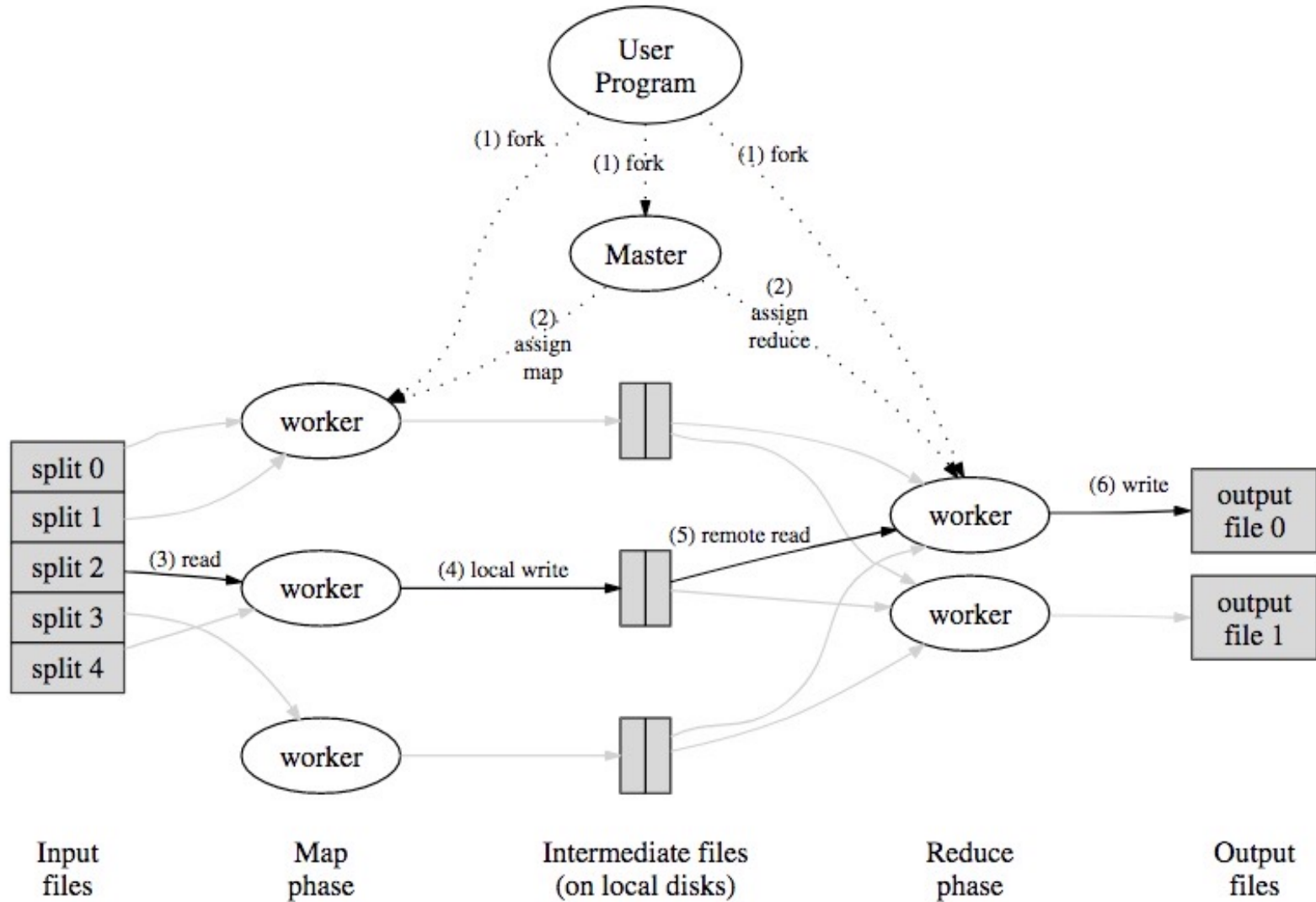
All-to-All.



Word Count execution



MapReduce data flows in paper

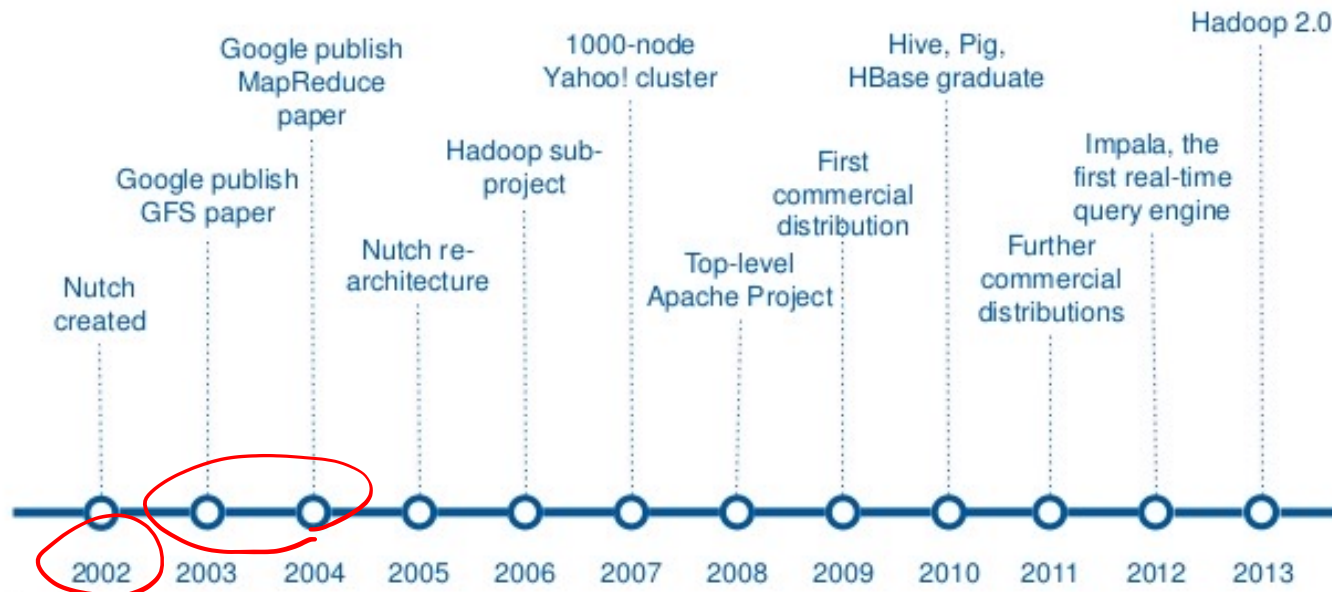


How it started: Apache Hadoop

- An open-source implementation of Google's MapReduce framework
 - Hadoop MapReduce atop Hadoop Distributed File System (HDFS)



A Brief History of Hadoop



How it's going ...

DATA & AI LANDSCAPE 2019

INFRASTRUCTURE

HADOOP ON-PREMISE
 cloudera Hortonworks
 MAPR Pivotal
 IBM InfoSphere
 jethro

HADOOP IN THE CLOUD
 AWS Microsoft Azure
 Google Cloud
 SAP Cloud Platform
 IBM InfoSphere BigInsights
 ORACLE
 HADOPIE CAZENA

STREAMING / IN-MEMORY
 Amazon Kinesis
 SAP Cloud Platform
 ORACLE
 confluent
 stream hazelcast GridGain
 GIGASPACE Wallaroo PASDATA Ix

ANALYTICS & MACHINE INTELLIGENCE

DATA ANALYST PLATFORMS
 Microsoft pentaho alteryx
 Digital Reasoning QUAVUS AYASDI
 ATTIVO Datameer incorta
 interlana MODE ENDOR
 sisu switchboard Starburst

DATA SCIENCE PLATFORMS
 IBM databricks dataiku
 DOMINO rapidminer TIBCO
 ANACONDA SAS Altair
 KNIME MathWorks

APPLICATIONS - ENTERPRISE

SALES
 CHORUS INSIDESALES.COM peopleai
 conversica clari avso tactai
 fusejmachines Cleantek

MARKETING - B2B
 RADIUS App Annie
 EVERSTING Lattice
 tubular JENGGAGIO
 KNOTCH mpro

MARKETING - B2C
 Bloomreach ACTIONIQ BLUECORE
 CONTENTSQUARE TEALUMIN
 Amplero ampertly QUANTIGO
 Simon Lyfika PERSADO
 remesh

CUSTOMER EXPERIENCE / SERVICE
 ZETA MEDALLIA SurveyMonkey
 CLARABRIDGE zendesk Kustomer
 mycom DMN LIVESTREAM Gainsight
 HEAP Amplitude Watson Assistant
 DigitalGenius ASAPP ada AUTOMAT ahni
 CarDesk metorm

ENTERPRISE PRODUCTIVITY
 slack ORACLE
 GURU lumiatu
 DIFFBOT clara
 talla Kasisto

NoSQL DATABASES
 Google Cloud AWS ORACLE Microsoft Azure
 mongoDB MarkLogic
 Couchbase DATASTAX
 redislabs REEROSPIKE
 AlazaoDB SCYLLA

NewSQL DATABASES
 SAP Clustrix Pivotal
 MEMSQL Influxdata
 Cockroach Labs
 VOLTA splice
 paradigms

GRAPH DBs
 Amazon Neptune
 IBM ORACLE
 GraphDB
 GraphSense
 GraphAware

MPP DBs
 TERADATA VERTICA
 IBM Data Warehouse Systems
 Exasol
 dremio Yellowbrick

CLOUD EDW
 AWS Google Cloud
 Pivotal
 snowflake
 Infoworks

SERVERLESS
 AWS Google Cloud
 Pivotal
 snowflake
 Infoworks

BI PLATFORMS
 looker databricks
 Domo ARCADIA DATA ThoughtSpot
 ATSCALE BIENICE Qlik
 GoodData Informatica
 MicroStrategy Keen IO

VISUALIZATION
 Tableau Power BI
 SAP Google Cloud
 celonis
 zepl Periscope Data
 CHARTIO

MACHINE LEARNING
 Amazon SageMaker
 H2O
 DataRobot gamaiion
 VIZENIZE ELEMENT
 deeppersonal

HUMAN CAPITAL
 pymentics
 hiit
 Ailyo Textio
 Workday Stella
 entelo
 incommo
 boomy

LEGAL
 RAVEL
 DISCO Q
 JUDICATA
 IRONLAD
 PRESENTION
 ROSS

REGTECH & COMPLIANCE
 Regi
 text IQ
 PARTNERSHIPS
 CROSSBEM
 DATA REPUBLIC

FINANCE
 Anaplan
 ZUORO
 TRADESHIFF
 SCALE FACTOR
 bukeeper
 pilot

BACK OFFICE AUTOMATION & RPA
 UiPath
 blueprism
 VIDAO
 Workfusion Workato
 RPAWORKS
 ALKYRI

SECURITY
 TANIUM
 CODE42 CyberCloud
 VECTRA
 pindrop
 CodeSecure
 bitbitgloss
 sponcoglossion
 BLUEHEXAGON Semble
 CYLANCE
 DARKTRACE ANOMALI
 DATAVISOR
 feedzai
 SenteralOne
 SecurityScorecard
 SECURE
 BITSIGHT
 JAS SK
 FORTER
 riskrecon
 JAS SK
 Zscaler
 StackPath
 Illumio

DATA TRANSFORMATION
 talend pentaho
 alteryx TRIFACTA
 tmnr Paxata
 StreamSets UNIFI

DATA INTEGRATION
 SAP Data Services Informatica
 Hadoop Talend
 Inphigic
 Segment
 ZALONI
 Infoworks
 Snowflake
 MANTL

DATA GOVERNANCE
 IBM ORACLE
 collibra
 Alation
 OKERA
 MANTA
 dataworld

MGMT / MONITORING
 AWS New Relic octio
 rubrik APPDYNAMICS
 dynatrace WAVEFOOT
 Signalix
 splunk
 zurnal
 uniplex
 OKERA
 MANTA
 dataworld

COMPUTER VISION
 Microsoft Azure
 Amazon Rekognition
 clarifai
 EVER AI
 neurologix
 xUBIQUITY
 YRUBIA
 YRUBIA
 YRUBIA

HORIZONTAL AI
 IBM Watson Cortana
 Affective
 Humentia
 nurologix
 YRUBIA
 YRUBIA
 YRUBIA

SPEECH & NLP
 Google Cloud
 Amazon Transcribe
 narrative science
 Movell
 Soundhound Inc.
 cogito slips
 SHARPLING

ADVERTISING
 AppNexus
 Oracle MOAT
 distillery
 TAPAI
 Oppir

EDUCATION
 Knewton
 Clever
 edclara
 kidaptive
 PANORAMA
 Kknow
 gradescope

REAL ESTATE
 Redfin
 Opener
 VTS
 CREDIFI
 GEOPHY
 COMPSTAK
 STACEMAKER

GOVT
 OPENOVY
 mark43
 LiveStories
 Passport
 SmartProcure
 STREETIMAGINE
 QUESTADIT

INTELLIGENCE
 Palantir
 Dataminr
 Quid
 PRIMER
 FORGE

FINANCE - INVESTING
 KENSH
 ADAPTEK
 MINDEN
 ISENTIUM
 AORZOR
 PowerPack
 PAGAYA

FINANCE - LENDING
 ondeck
 Affirm
 JIANPULAI
 TALAI
 LEARBANC
 100CREDIT
 Transcendence
 MoneyLion
 cire

INSURANCE
 Insomnio
 Anomali
 CYLESCO
 SHIFT Technology
 ROOT
 ZESTY AI
 CAPE

STORAGE
 AWS Google Cloud
 Microsoft Azure
 PURE STORAGE
 ALLUXIO
 HIRIBESTORAGE
 Datto
 COHEITY

CLUSTER SVCS
 Amazon EC2
 IBM Cloudfoundry
 PAGERDUTY
 SIGM
 CRYOCLOUD

DATA GENERATION & LABELLING
 Amazon Mechanical Turk
 workwork
 oppen
 HIVE
 Labelbox
 Lionbridge

AI OPS
 ALGORITHMIA
 Vertaai
 datom
 datation
 Weights & Biases
 Determined AI
 fidder

GPU DBs & CLOUD
 Kinetic
 ORACLE
 PG-Stream
 FLOYDHUB

HARDWARE
 Google TPU
 ARM
 MYTHIC
 Ahabox
 WAVE
 CRYMIA
 DEFNIX

SEARCH
 Elasticsearch ORACLE
 Algolia COVEO
 Ludworks ATTIVO
 swifttype EXALAND
 alphaspense MAANA
 omnius SINEQUA

LOG ANALYTICS
 Splunk
 sumologic
 solarsinds
 TIMBER
 HObano
 logz.io

SOCIAL ANALYTICS
 Hootsuite
 NETBASE
 synthesize
 simple reach
 bitly
 SimilarWeb

WEB / MOBILE / COMMERCE ANALYTICS
 Google Analytics
 mixpanel AMPUTITUDE
 Airtable RESCI
 SIGOPT
 granify
 CUSTORA

HEALTHCARE
 Flatiron
 METABOTA
 3DMED
 TEMPUS
 notable
 entelic
 Qventus
 inmovacor

LIFE SCIENCES
 Benchling
 BenevolentAI
 WUOLINECO
 TIBCO
 CIBIEN
 PHAROS
 CITRINE
 AEMETIC
 DOWON

TRANSPORTATION
 Uber Tesla
 CLEARPATH
 drive ai
 nauto
 NIO
 nextradyne
 think INRIX

AGRICULTURE
 FARMERS EDGE
 Granular
 JOHN DEERE
 BLUEBERIDGE
 FarmLogs
 TARMANIS
 GAMAYA
 Electric

COMMERCE
 Instacart
 FAIRFX
 STITCH FIX
 HEURTEL
 FARMERS EDGE
 TARMANIS
 GAMAYA
 Electric

INDUSTRIAL
 AVEVA SIEMENS
 PREDIX UPTAKE
 SCORTEX
 TACHYUS
 SOIERN
 BOXEVER
 SPOLK

CROSS-INFRASTRUCTURE/ANALYTICS

AWS Google Cloud Microsoft IBM SAP Hewlett Packard Enterprise SAS 1010DATA vmware TIBCO TERADATA ORACLE NetApp syncsort MAPR cloudera

OPEN SOURCE

FRAMEWORKS
 Spark
 Flink YARN TEZ
 HADOPIE HELIX

QUERY / DATA FLOW
 Spark SQL presto
 SLMDATA
 GraphQL

DATA ACCESS & DATABASES
 Cassandra mongoDB redis
 Cockroach Labs
 druid
 ScioDB
 etcd Kong

ORCHESTRATION & MGMT
 talend
 Apache Airflow
 Apache Beam
 etcd Kong

STREAMING & MESSAGING
 Spark
 Flink
 kafka stream
 Apache RocketMQ

STAT TOOLS & LANGUAGES
 Scala
 R
 Julia
 SPOly

AI OPS & INFRA
 mflow
 kubeflow
 DVC
 SELDON
 PyTorch

AI / MACHINE LEARNING / DEEP LEARNING
 TensorFlow Keras
 theano
 VELES
 DMSUM
 FeatureFu

SEARCH
 Elasticsearch
 Solr
 kibana
 fluentbit
 fluentd
 Grafana

LOGGING & MONITORING
 kibana
 Prometheus
 fluentbit
 fluentd
 Grafana

VISUALIZATION
 matplotlib
 TensorBoard
 seaborn
 Folium

COLLABORATION
 BeakerX
 Jupyter
 ANACONDA

SECURITY
 Apache Ranger
 KNOX
 Sentry
 SCURTYLO

DATA SOURCES & APIs

HEALTH
 Apple VALIDIC
 practicefusion
 fitbit GARMIN
 HUMAN APP
 kinsio

IOT
 GE Digital
 UPTAKE
 thingworx
 helium
 samsara

FINANCIAL & ECONOMIC DATA
 Bloomberg THOMSON REUTERS
 DOW JONES
 S&P CAPITAL IQ
 CBINSIGHTS
 PLAIID
 GEstimize
 PREMISE
 Quandl
 Stockworks
 xignite
 Thinknum
 earnest
 predata

AIR / SPACE / SEA
 Orbital Insight planet
 AIRBOTICS
 inspire
 Understudy
 WINDWARD
 MarineTraffic

PEOPLE / ENTITIES
 acxiom experian
 EPSON
 InsideView
 Crimson Hexagon
 Quantcast
 BASIS
 SAFEGRAPH

LOCATION INTELLIGENCE
 FOURSQUARE
 sense360
 PlaceIQ
 CAR2
 esri
 factual
 Mapillary
 StreetView
 cuebiq
 Radar
 iagritillio

OTHER
 DATA GOV
 IMAGENET
 CRUX
 iagritillio

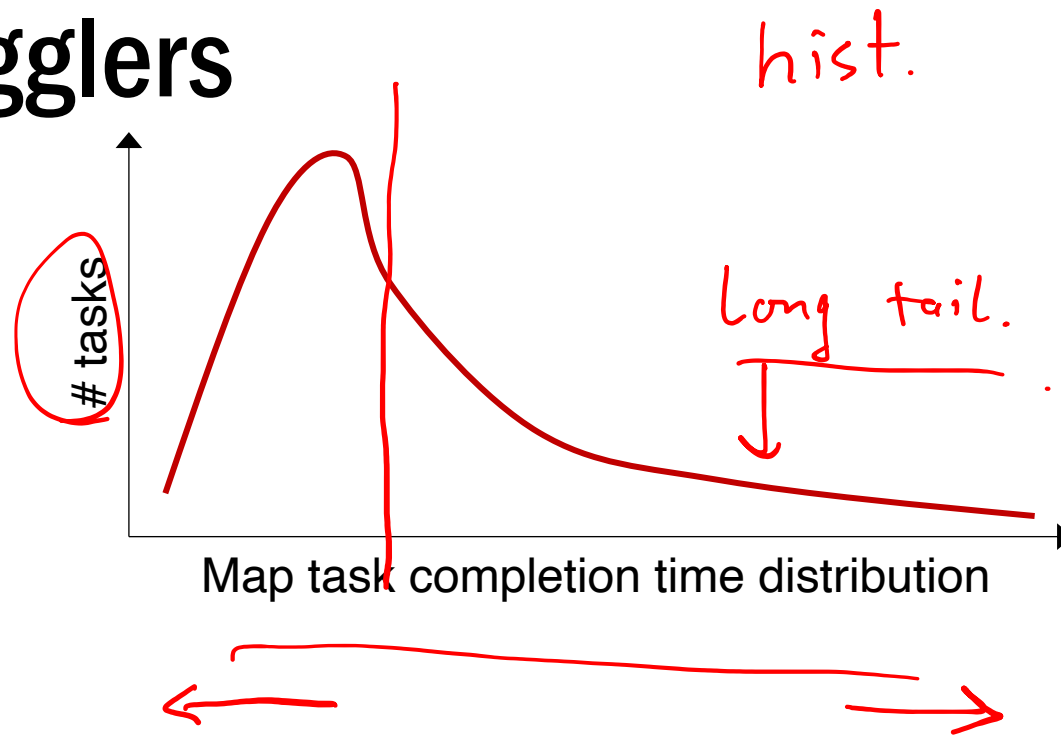
DATA RESOURCES

DATA SERVICES
 OPERA
 DATA SCIENCE
 fractal
 kaggle
 DataKind
 innoplexus

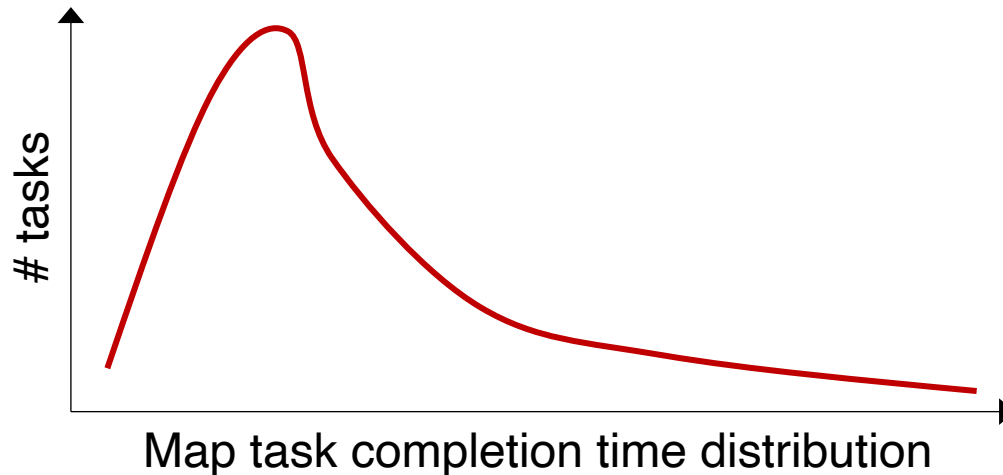
INCUBATORS & SCHOOLS
 PLURALSIGHT
 DataCamp
 INSIGHT
 The Data Incubator
 METIS

RESEARCH
 OpenAI facebook research
 MIRI
 VECTOR INSTITUTE
 AIZ
 ALLEN INSTITUTE FOR ARTIFICIAL INTELLIGENCE

Stragglers



Stragglers



$$x \cdot 1 = x$$

$$x \cdot 0 = 0$$

- **Tail execution time** means some workers (always) finish late

Idempotence.

- Q: How can MR work around this?
 - Hint: its approach to **fault-tolerance** provides the right tool

Resilience against stragglers

*speculative
exe.*

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

More design

Job

- Master failure

- Locality

- Task granularity

M R

computers. \rightarrow LB

GFS usage at Google

- 200+ clusters
- Many clusters of 1000s of machines
- Pools of 1000s of clients
- 4+ PB filesystems
- 40 GB/s read/write load
 - In the presence of frequent hardware failures

* Jeff Dean, LADIS 2009

MapReduce usage statistics over time

	Aug, '04	Mar, '06	Sep, '07	Sep, '09
Number of jobs	29K	171K	2,217K	3,467K
Average completion time (secs)	634	874	395	475
Machine years used	217	2,002	11,081	25,562
Input data read (TB)	3,288	52,254	403,152	544,130
Intermediate data (TB)	758	6,743	34,774	90,120
Output data written (TB)	193	2,970	14,018	57,520
Average worker machines	157	268	394	488

* Jeff Dean, LADIS 2009

MapReduce discussion

What will likely serve as a performance bottleneck for Google's MapReduce used back in 2004 (or even earlier)? CPU? Memory? Disk? Network? Anything else?

MapReduce discussion

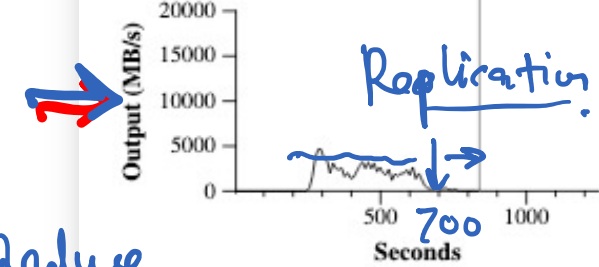
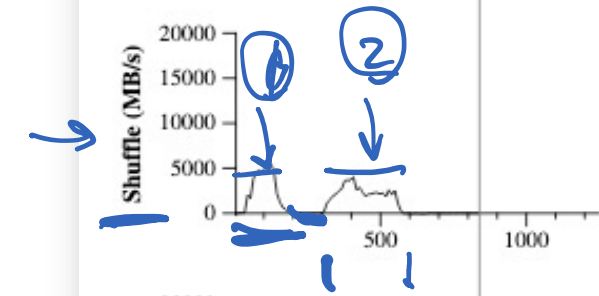
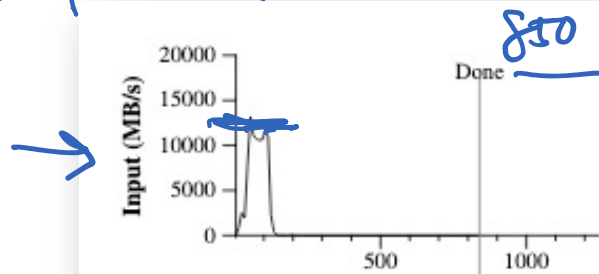
What will likely serve as a performance bottleneck for Google's MapReduce used back in 2004 (or even earlier)? CPU? Memory? Disk? Network?
Anything else?

12.5 MB / s.

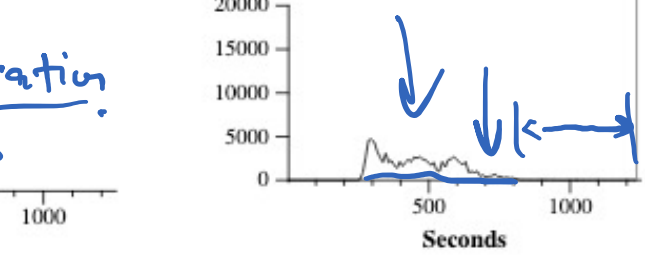
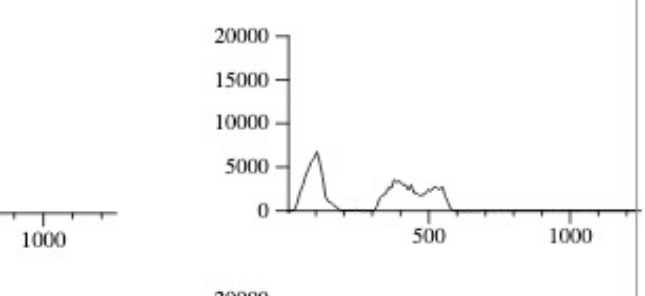
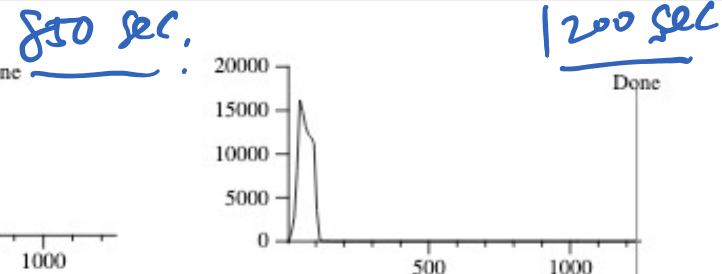
How does MapReduce reduce the effect of slow network?

MapReduce discussion

Map. 13 GB/s.

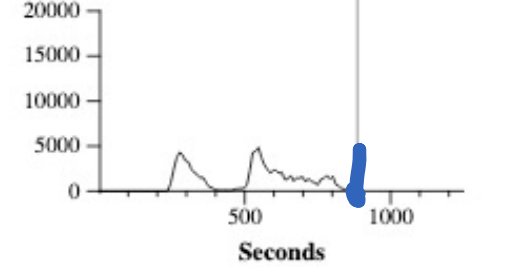
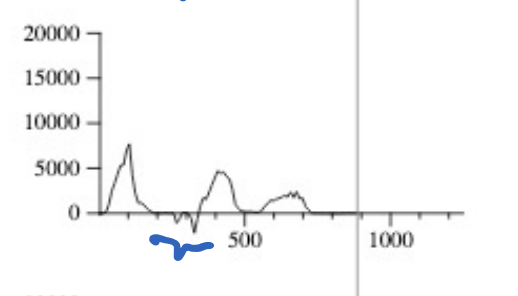
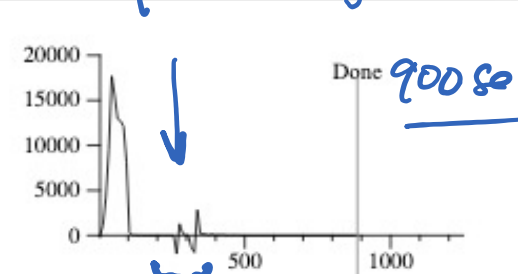


(a) Normal execution



(b) No backup tasks

Failures of 200 Workers.



(c) 200 tasks killed

Reduce

MapReduce discussion

Consider a log analytics job where you perform log-based debugging. You want to extract the timestamp info of all entries that match a keyword and then calculate the count of all matched entries:

1. Filter the entries with the keyword;
2. Calculate the count of all matched entries

What are the main shortcomings of using MapReduce to support such pipeline-like applications?

Next step

- Look out for
 - • Project suggestion doc
 - Fill the team composition form
 - Project bid and team composition due by Feb 24
- Next week: Apache Spark