

# MapReduce: Simplified data processing on large clusters

Jeffrey Dean and Sanjay Ghemawat

Google, Inc.

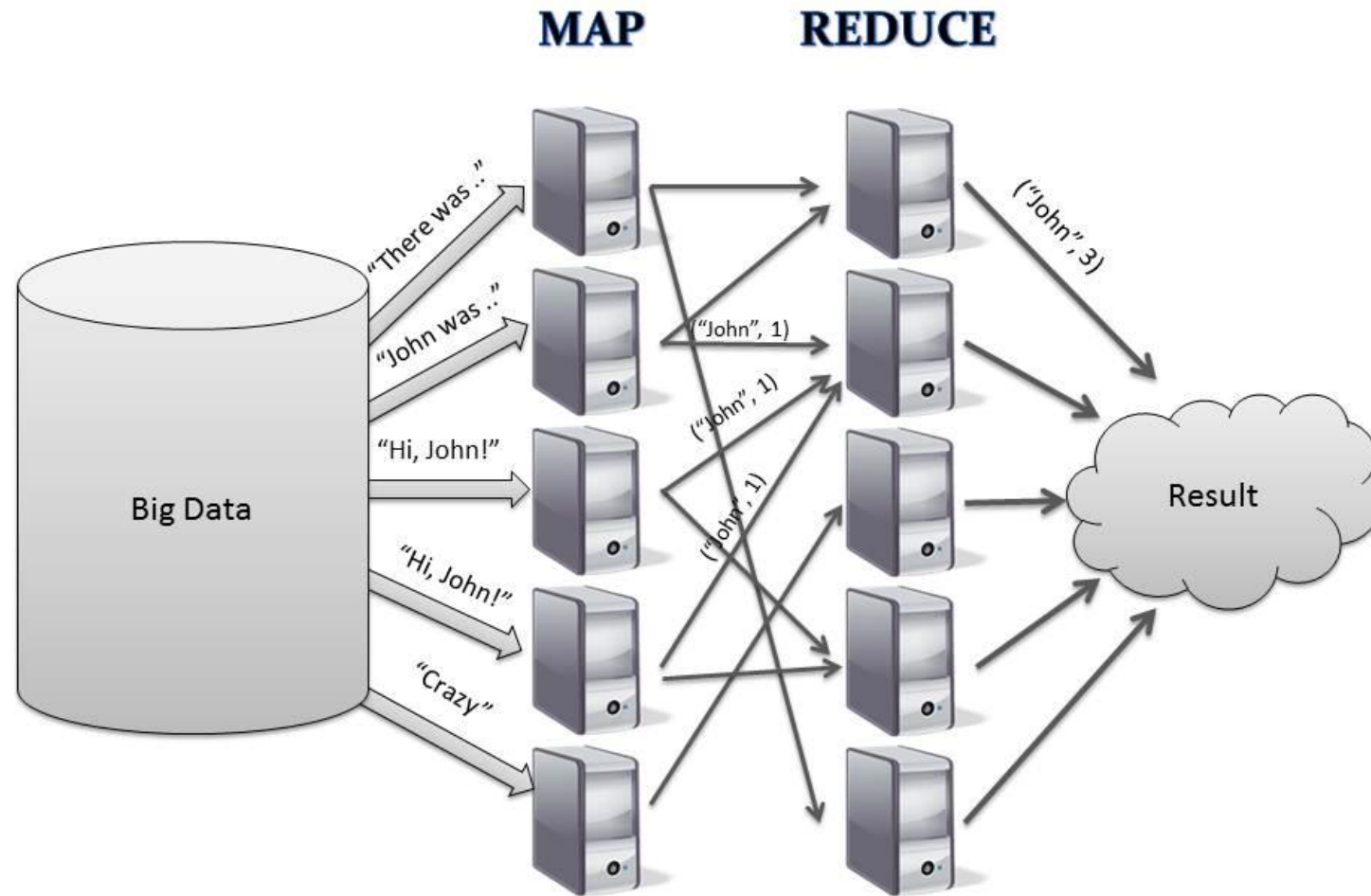
2004

Diego Gonzalez, Harold Achicanoy

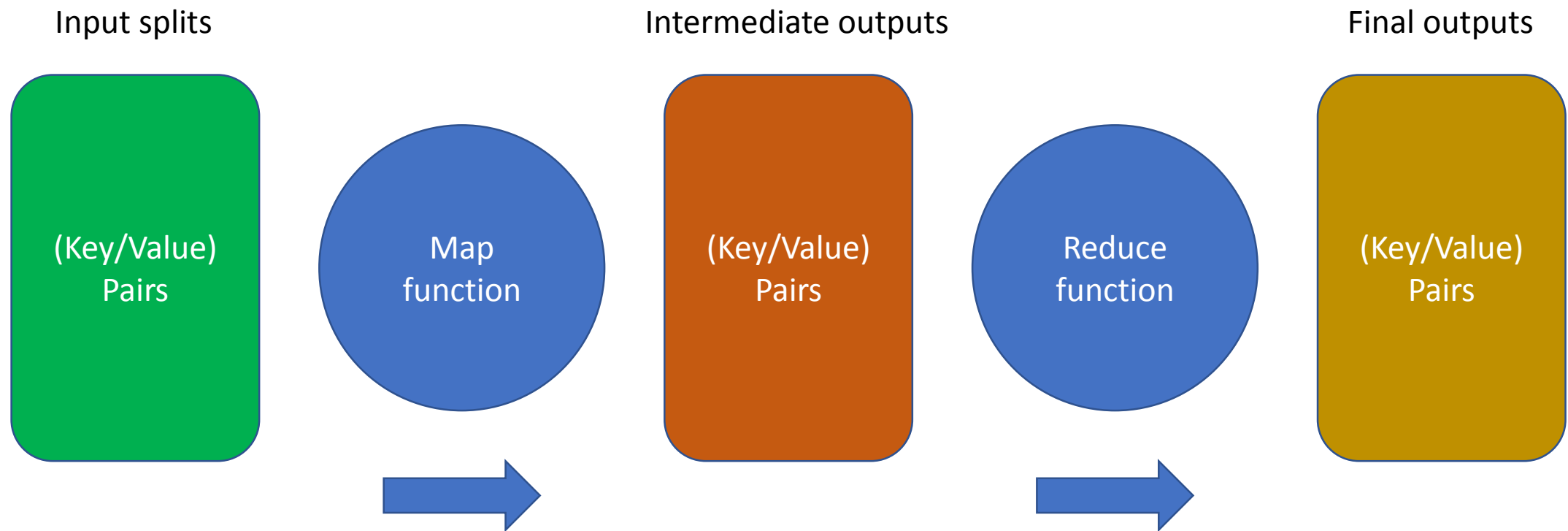
# Outline

- MapReduce
- Programming model
- Implementation
- Performance
- Applications and conclusions

# MapReduce

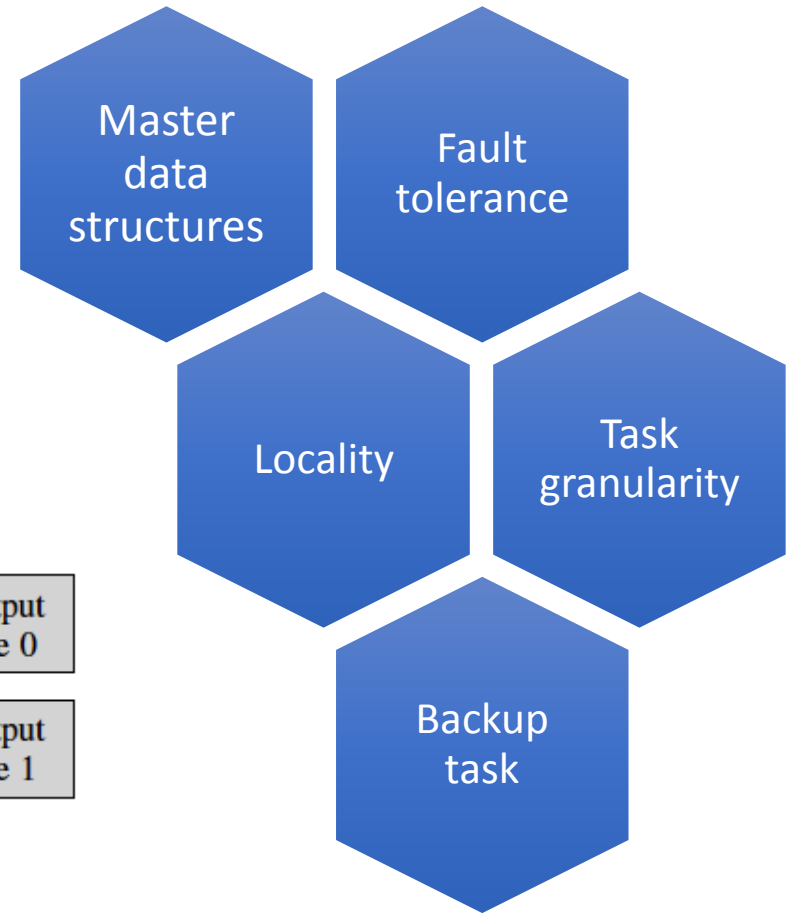
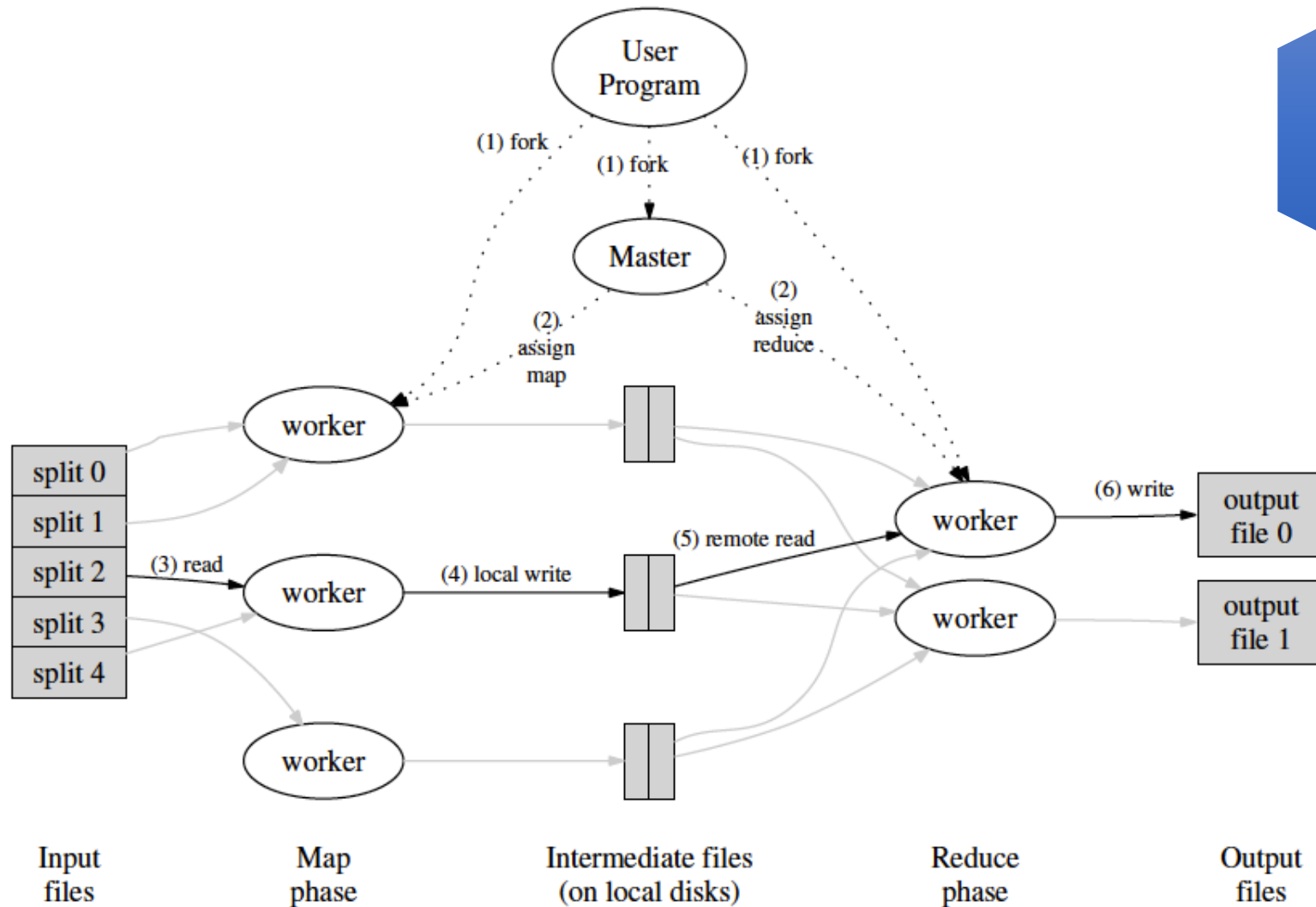


# Programming model



**Results:** Produces a set of merged output values (usually just one)

# Implementation



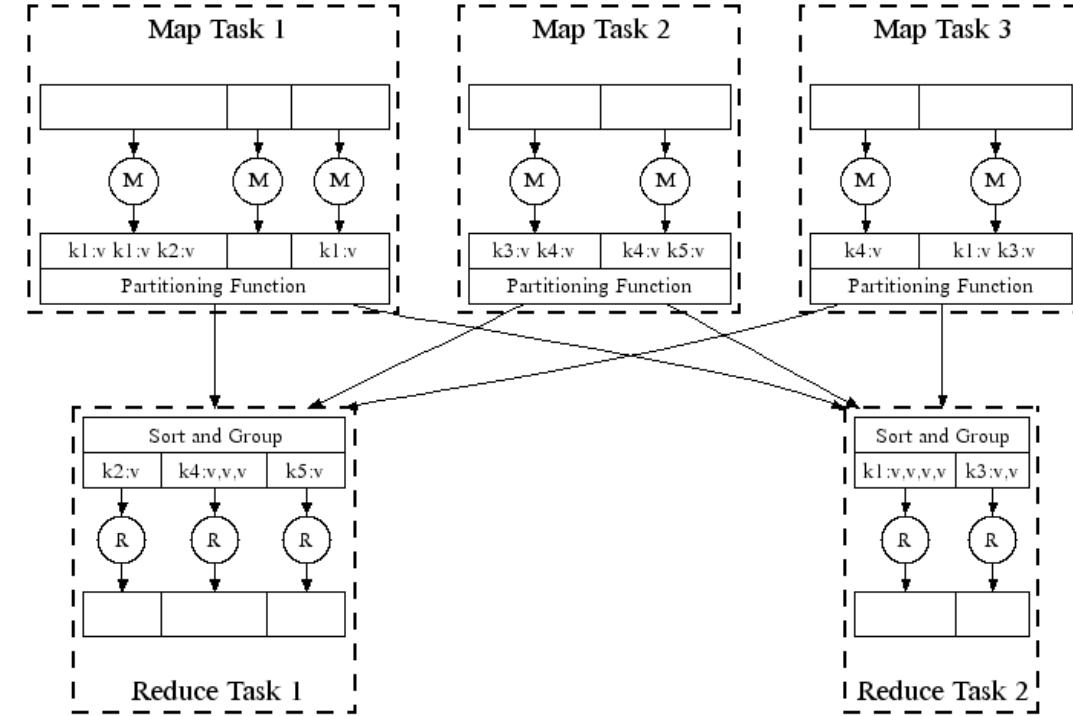
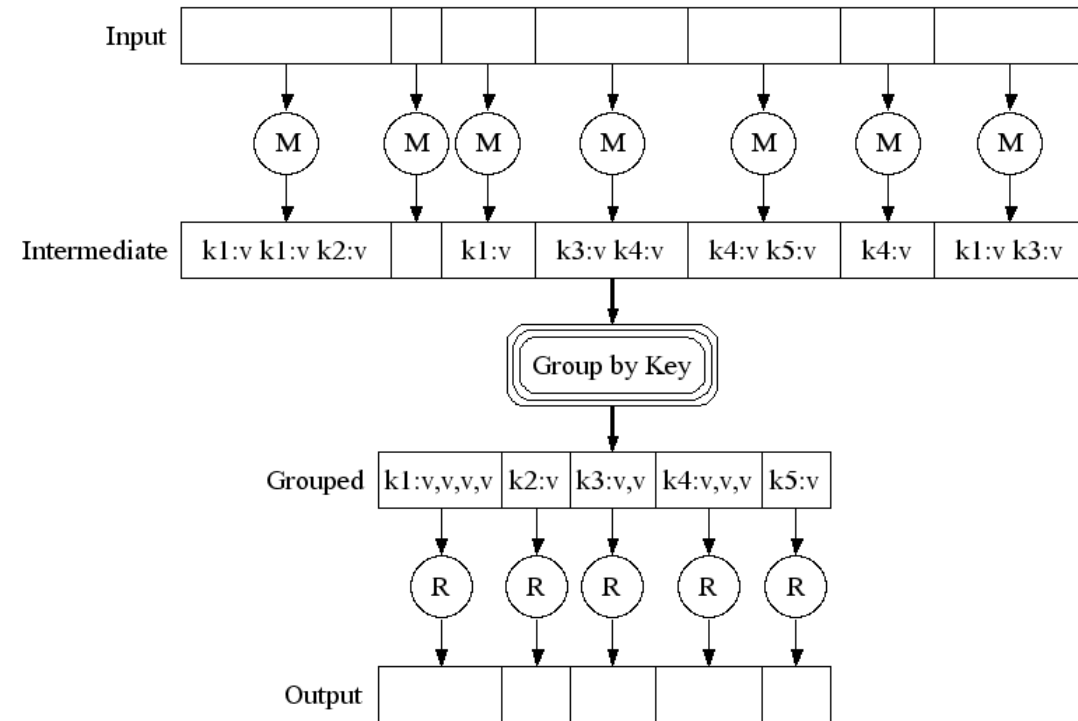
# Refinements

- **Partitioning function:** This tends to result in fairly well-balanced partitions.
- **Ordering guarantees:** to support efficient random access lookups by key
- **Combiner function:** Partial combining significantly speeds up certain classes of MapReduce operations.
- **Input and output types:** To produce data in different formats and it is easy for user code to add support for new output types.

# Refinements

- **Skipping bad records:** it detects which records cause deterministic crashes and skips these records in order to make forward progress.
- **Local execution:** To help facilitate debugging, profiling, and small-scale testing.
- **Status information:** it shows the progress of the computation.
- **Counters:** Users have found the counter facility useful for sanity checking the behavior of MapReduce operations.

# Refinements





# Performance

- 1 TB of processed data
- 1800 machines
- Each machine had two 2GHz Intel Xeon processors with Hyper-Threading enabled, 4GB of memory, two 160GB IDE disks, and a gigabit Ethernet link.
- Grep: The grep program scans through  $10^{10}$  100-byte records, searching for a relatively rare three-character pattern (the pattern occurs in 92,337 records).
- Sort: The sort program sorts  $10^{10}$  100-byte records (approximately 1 terabyte of data).

# Applications

- Large-scale machine learning problems
- Clustering problems for the Google News and Froogle products
- Extraction of data used to produce reports of popular queries
- Extraction of properties of web pages for new experiments and products

# Conclusions

MapReduce provides:

- Automatic parallelization and distribution
- Fault-tolerance
- I/O scheduling
- Status and monitoring