MapReduce

Aiichiro Nakano

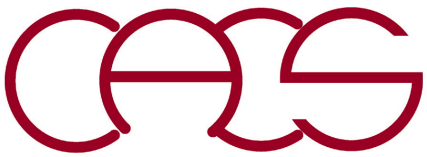
Collaboratory for Advanced Computing & Simulations
Dept. of Computer Science, Dept. of Physics & Astronomy,
Dept. of Chemical Engineering & Materials Science
Dept. of Biological Sciences
University of Southern California

Email: anakano@usc.edu

Amazon Elastic Computing Cloud (EC2)

or

Hadoop@USC-HPC



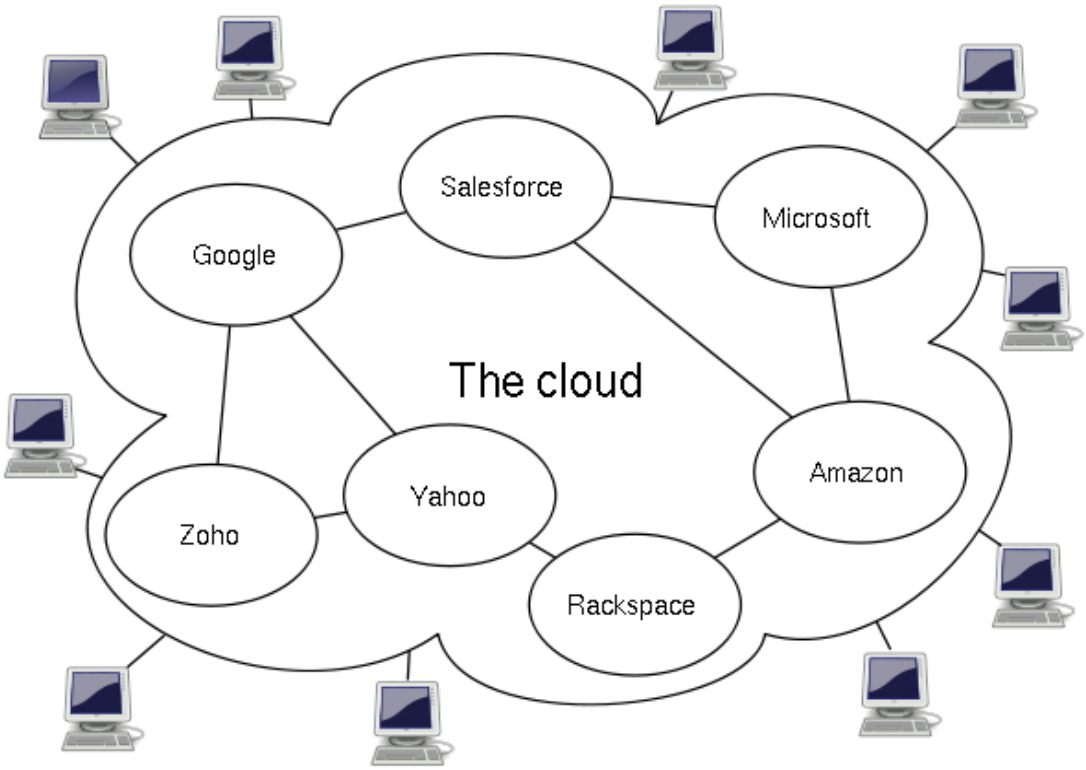
Above the Clouds: A Berkeley View of Cloud Computing

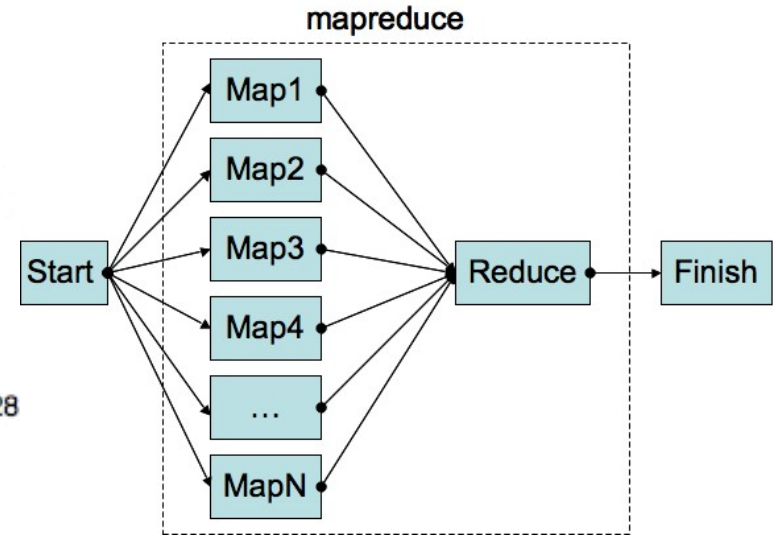
Michael Armbrust
Armando Fox
Rean Griffith
Anthony D. Joseph
Randy H. Katz
Andrew Konwinski
Gunho Lee
David A. Patterson
Ariel Rabkin
Ion Stoica
Matei Zaharia

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2009-28
http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28

February 10, 2009





MapReduce

- Parallel programming model for data-intensive applications on large clusters
  - > User just implements Map() and Reduce()
- Parallel computing framework
  - > Libraries take care of everything else
    - Parallelization
    - Fault tolerance
    - Data distribution
    - Load balancing
- Developed at Google

Functional Abstraction

- **Map and Reduce functions borrowed from functional programming languages**

  (Common LISP example)

  ```
  > (mapcar '1+ '(1 2 3 4)) ⇒ (2 3 4 5)
  > (reduce '+ '(1 2 3 4)) ⇒ 10
  ```
  cf. **MPI_Allreduce()**

- **Map()**
  - **Process a key/value pair to generate intermediate key/value pairs**
- **Reduce()**
  - **Merge all intermediate values associated with the same key**

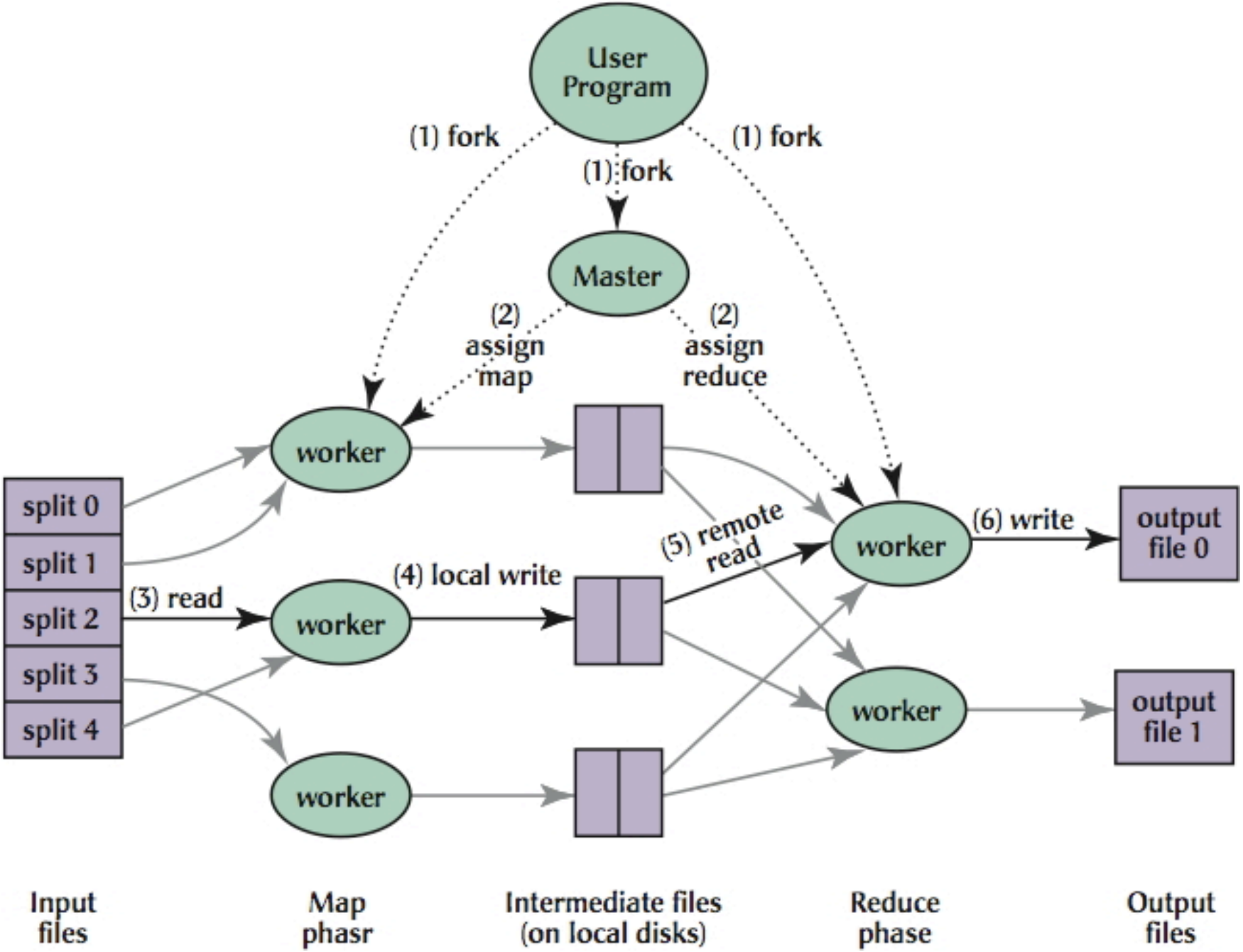
Example: Counting Words

- **Map()**
  - > **Input <filename, file text>**
  - > **Parses file and emits <word, count> pairs**
    - ***e.g.* <"hello", 1>**
- **Reduce()**
  - > **Sums values for the same key and emits <word, TotalCount>**
    - ***e.g.* <"hello", (3 5 2 7)> ⇒ <"hello", 17>**

```
map(String key, String value):
// key: document name
// value: document contents
for each word w in value:
  EmitIntermediate(w, "1");

reduce(String key, Iterator values):
// key: a word
// values: a list of counts
int result = 0;
for each v in values:
  result += ParseInt(v);
Emit(AsString(result));
```

Parallel Execution



MapReduce Resources

- **Hadoop implementation**
  `http://hadoop.apache.org`
- **MapReduce tutorial**
  `https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html`
- **Paper**
  J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Communications of the ACM* **51(1)**, 107 ('08)
- **Free account (Amazon Web Services)**
  `http://aws.amazon.com/free`

Cloud Supercomputing

TECHNOLOGY LAB / INFORMATION TECHNOLOGY

18 hours, $33K, and 156,314 cores: Amazon cloud HPC hits a "petaflop"

1.21 petaflops? Great scott!

by **Jon Brodkin** - Nov 12 2013, 11:00am PST

CLOUD SUPERCOMPUTING 54

USC chemistry professor Mark Thompson needed the cluster to design materials that might be well-suited to converting sunlight into solar energy.

"For any possible material, just figuring out how to synthesize it, purify it, and then analyze it typically takes a year of grad student time and hundreds of thousands of dollars in equipment, chemicals, and labor for that one molecule," Cycle Computing CEO Jason Stowe wrote in a blog post today.

Instead of doing that, Thompson uses simulation software made by Schrödinger. With that software running on Amazon, Thompson was able to simulate 205,000 molecules and do the equivalent of 2.3 million hours of science (counting the compute time for each core separately). The cluster ran only last week, so it's too early to find out what its impact on solar science will be. Still, from a computing standpoint, it's impressive.

National Strategic Computing Initiative

- **July 29, 2015: President Obama issued an executive order**

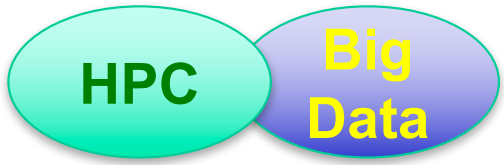
EXECUTIVE ORDER

- - - - - - -

CREATING A NATIONAL STRATEGIC COMPUTING INITIATIVE

BARACK OBAMA

By the authority vested in me as President by the Constitution and the laws of the United States of America, and to maximize benefits of high-performance computing (HPC) research, development, and deployment, it is hereby ordered as follows:



- **NSCI will merge exaflop/s (10^{18} floating-point operations per second) high performance computing (HPC) & exabyte (10^{18} bytes) "big data" to advance the frontier of sciences, economic growth, & national security**