Data-Intensive Information Processing Applications — Session #2

Hadoop: Nuts and Bolts

Jimmy Lin
University of Maryland

Tuesday, February 2, 2010

Hadoop Programming

- Remember “strong Java programming” as pre-requisite?
- But this course is *not* about programming!
  - Focus on “thinking at scale” and algorithm design
  - We’ll expect you to pick up Hadoop (quickly) along the way
- How do I learn Hadoop?
  - This session: brief overview
  - White’s book
  - RTFM, RTFC(!)

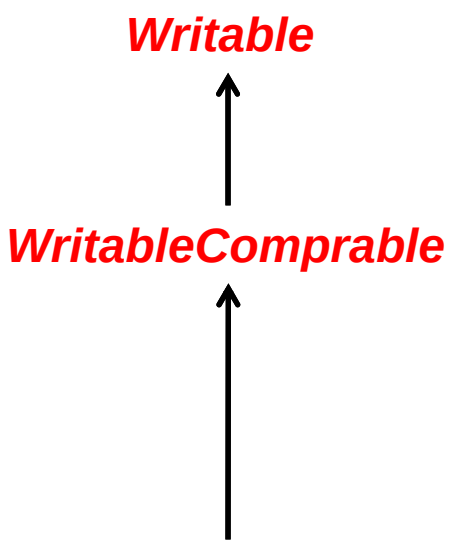
Source: Wikipedia (Mahout)

Basic Hadoop API*

- Mapper
  - void map(K1 key, V1 value, OutputCollector<K2, V2> output, Reporter reporter)
  - void configure(JobConf job)
  - void close() throws IOException
- Reducer/Combiner
  - void reduce(K2 key, Iterator<V2> values, OutputCollector<K3,V3> output, Reporter reporter)
  - void configure(JobConf job)
  - void close() throws IOException
- Partitioner
  - void getPartition(K2 key, V2 value, int numPartitions)

*Note: forthcoming API changes…

Data Types in Hadoop



Writable — Defines a de/serialization protocol. Every data type in Hadoop is a Writable.

WritableComprable — Defines a sort order. All keys must be of this type (but not values).

IntWritable
LongWritable
Text
…

Concrete classes for different data types.

SequenceFiles — Binary encoded of a sequence of key/value pairs

“Hello World”: Word Count

```
Map(String docid, String text):
    for each word w in text:
        Emit(w, 1);

Reduce(String term, Iterator<Int> values):
    int sum = 0;
    for each v in values:
        sum += v;
        Emit(term, value);
```

Three Gotchas

- Avoid object creation, at all costs
- Execution framework reuses value in reducer
- Passing parameters into mappers and reducers
  - DistributedCache for larger (static) data

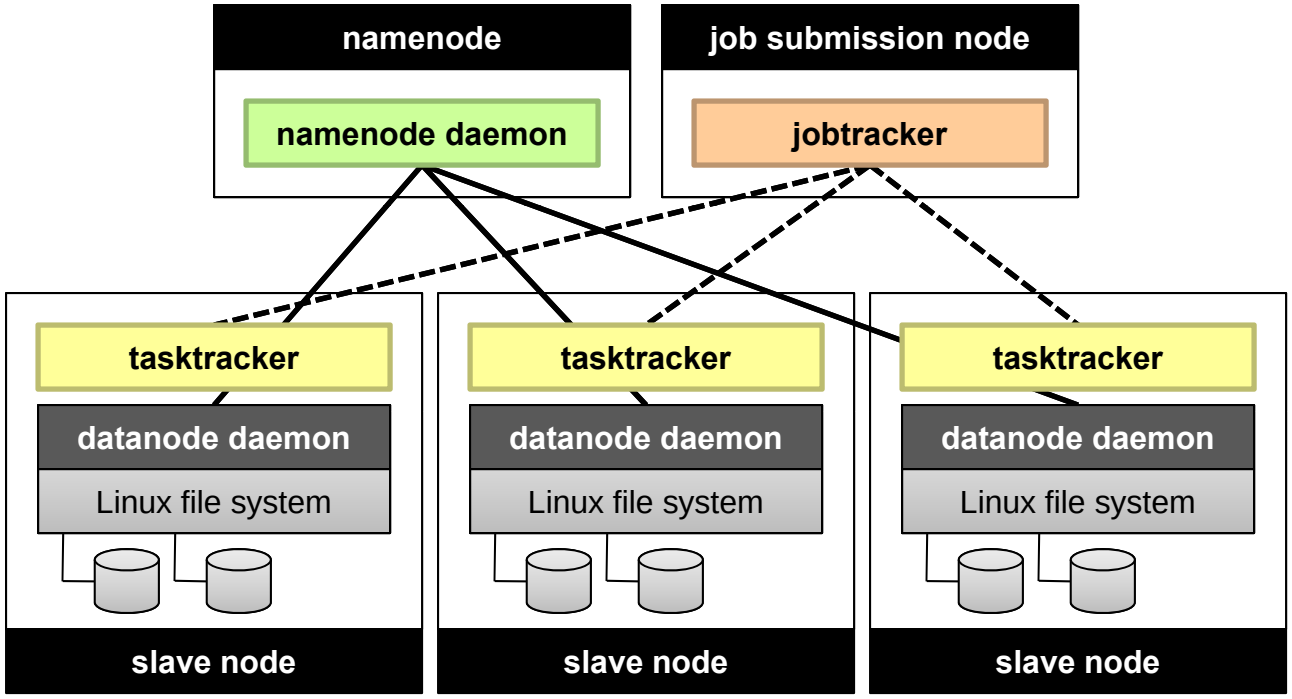
Complex Data Types in Hadoop

- How do you implement complex data types?
- The easiest way:
  - Encoded it as Text, e.g., (a, b) = “a:b”
  - Use regular expressions to parse and extract data
  - Works, but pretty hack-ish
- The hard way:
  - Define a custom implementation of WritableComprable
  - Must implement: readFields, write, compareTo
  - Computationally efficient, but slow for rapid prototyping
- Alternatives:
  - Cloud[9] offers two other choices: Tuple and JSON
  - (Actually, not that useful in practice)

Basic Cluster Components

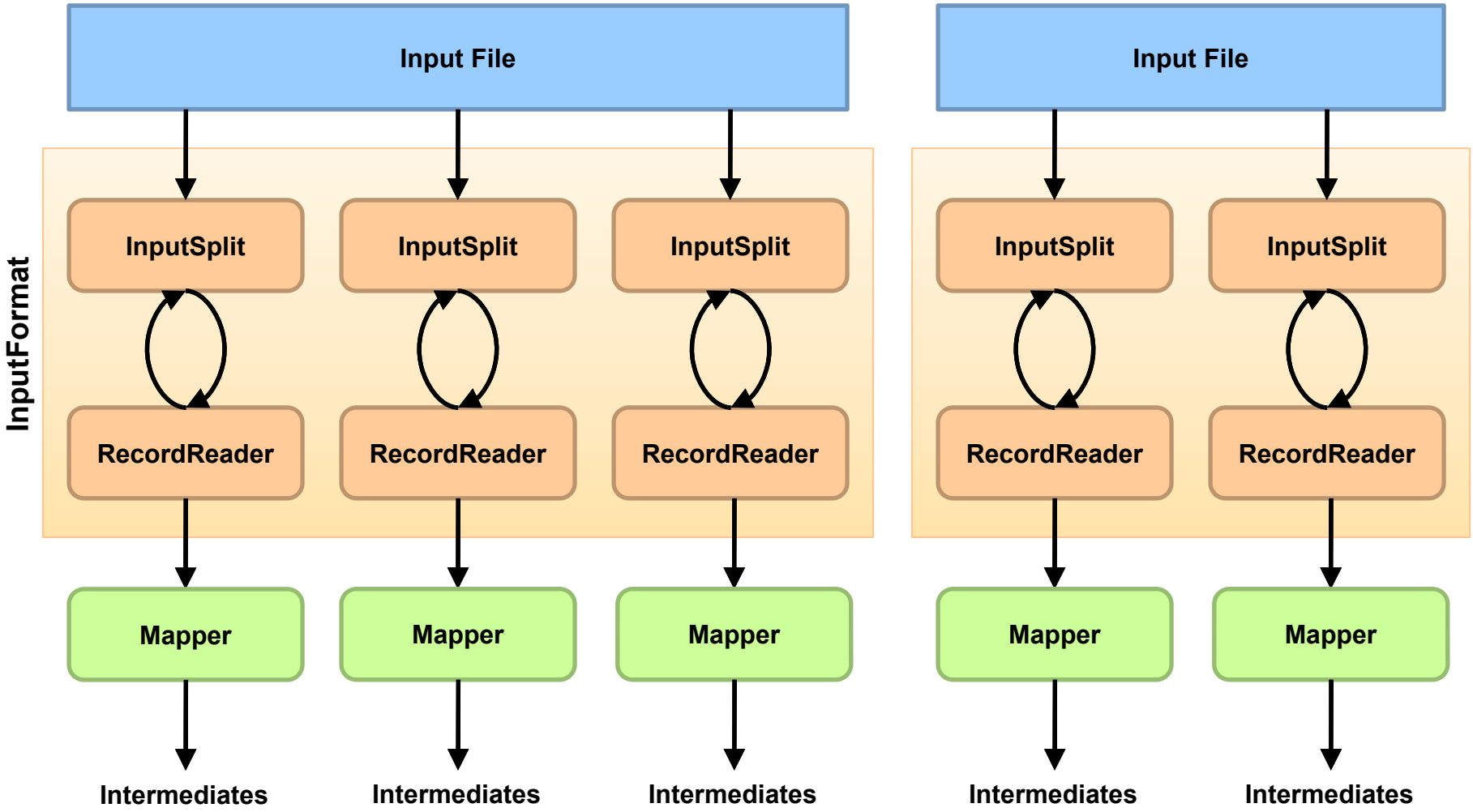
- One of each:
  - Namenode (NN)
  - Jobtracker (JT)
- Set of each per slave machine:
  - Tasktracker (TT)
  - Datanode (DN)

Putting everything together...

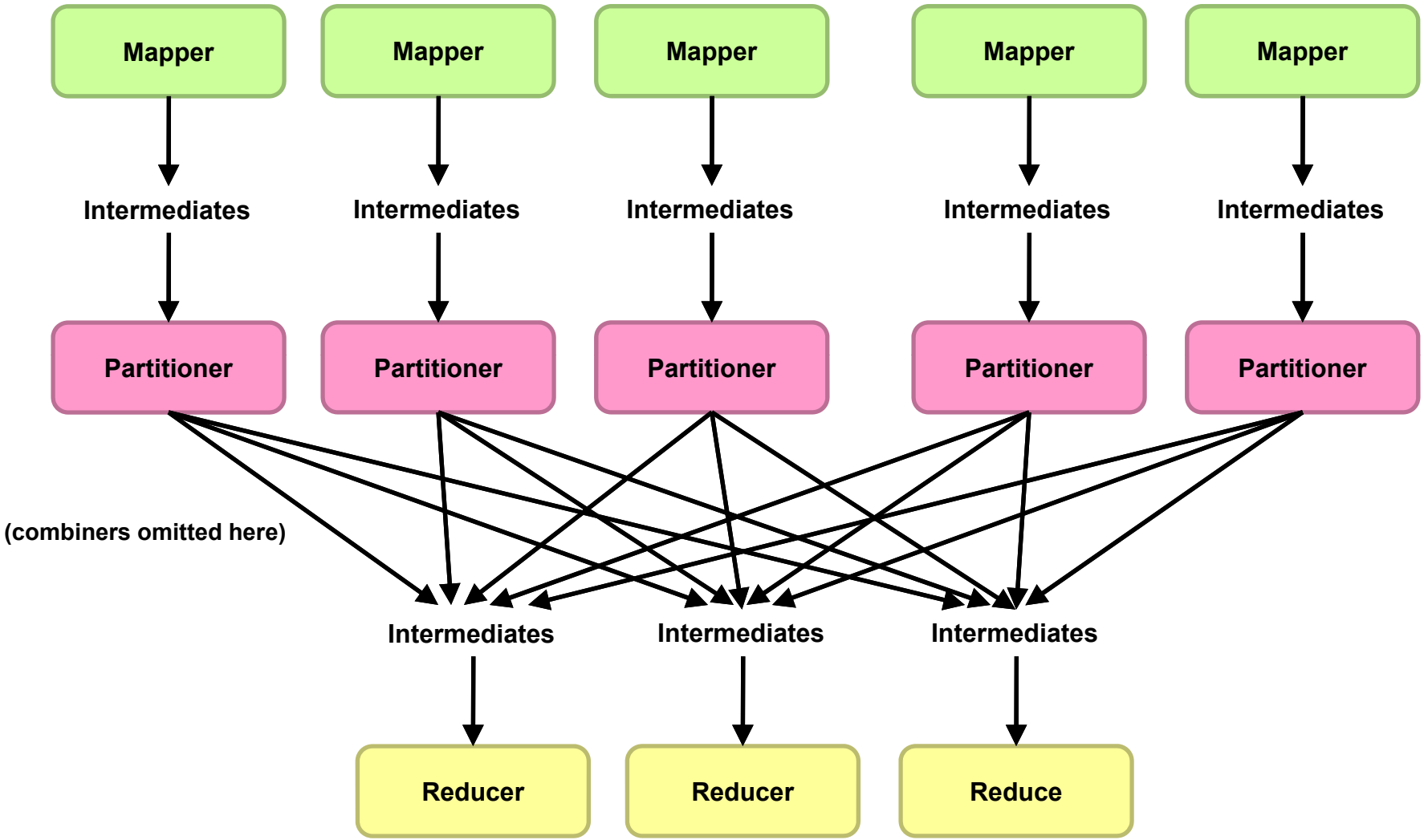


Anatomy of a Job

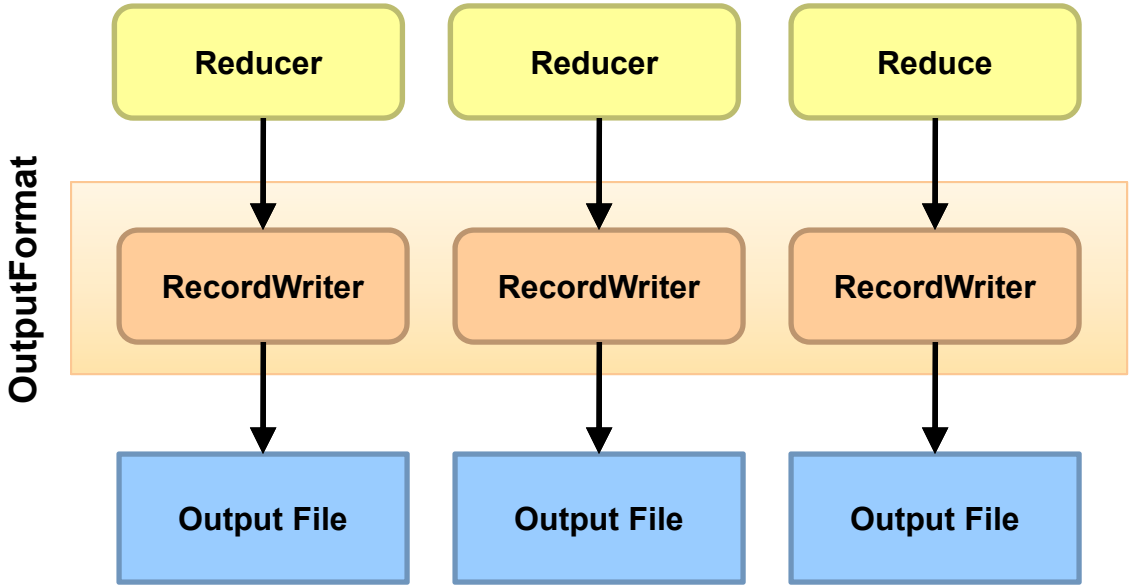
- MapReduce program in Hadoop = Hadoop job
  - Jobs are divided into map and reduce tasks
  - An instance of running a task is called a task attempt
  - Multiple jobs can be composed into a workflow
- Job submission process
  - Client (i.e., driver program) creates a job, configures it, and submits it to job tracker
  - JobClient computes input splits (on client end)
  - Job data (jar, configuration XML) are sent to JobTracker
  - JobTracker puts job data in shared location, enqueues tasks
  - TaskTrackers poll for tasks
  - Off to the races…



Source: redrawn from a slide by Cloduera, cc-licensed



Source: redrawn from a slide by Cloduera, cc-licensed



Source: redrawn from a slide by Cloduera, cc-licensed

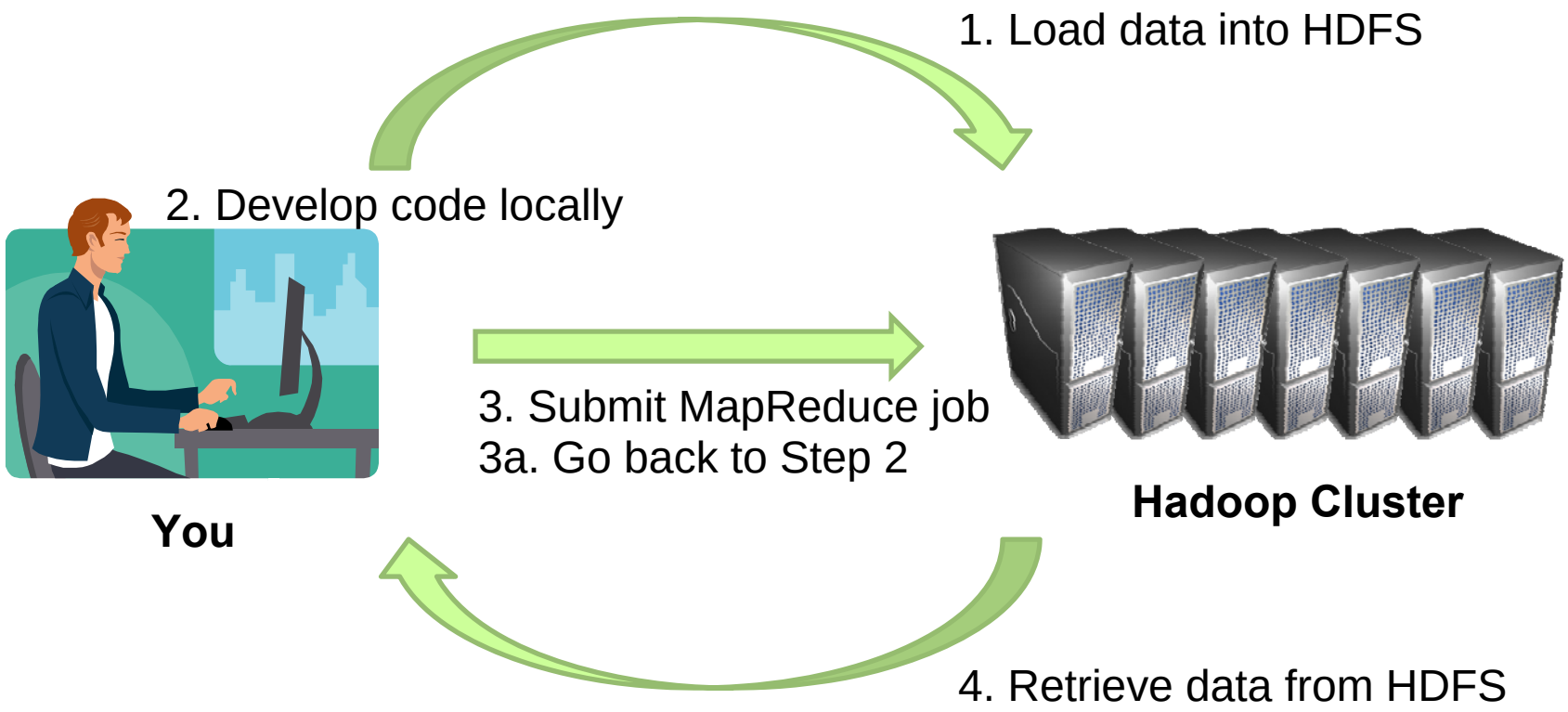
Input and Output

- InputFormat:
  - TextInputFormat
  - KeyValueTextInputFormat
  - SequenceFileInputFormat
  - …
- OutputFormat:
  - TextOutputFormat
  - SequenceFileOutputFormat
  - …

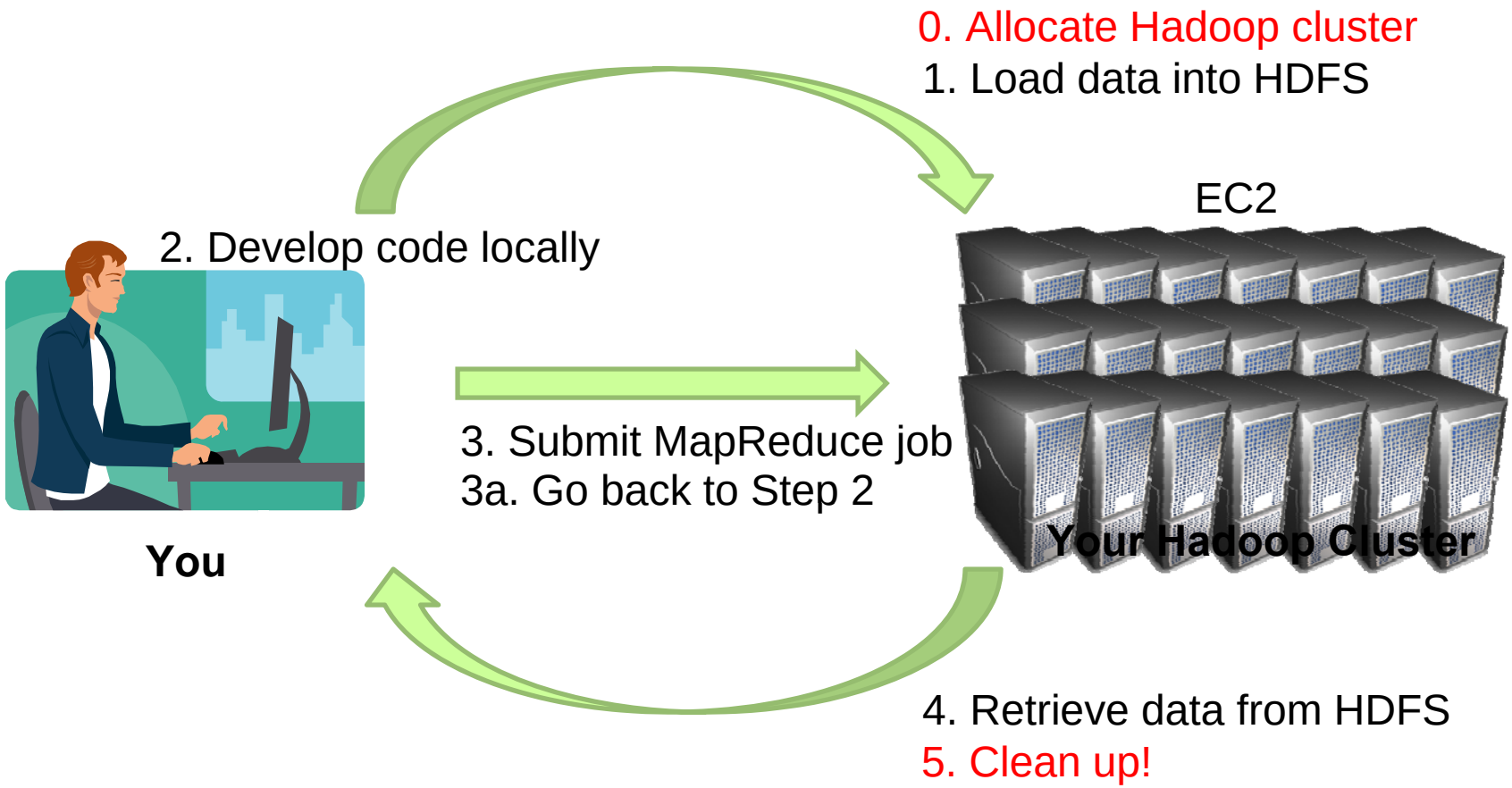
Shuffle and Sort in Hadoop

- Probably the most complex aspect of MapReduce!
- Map side
  - Map outputs are buffered in memory in a circular buffer
  - When buffer reaches threshold, contents are “spilled” to disk
  - Spills merged in a single, partitioned file (sorted within each partition): combiner runs here
- Reduce side
  - First, map outputs are copied over to reducer machine
  - “Sort” is a multi-pass merge of map outputs (happens in memory and on disk): combiner runs here
  - Final merge pass goes directly into reducer

Hadoop Workflow

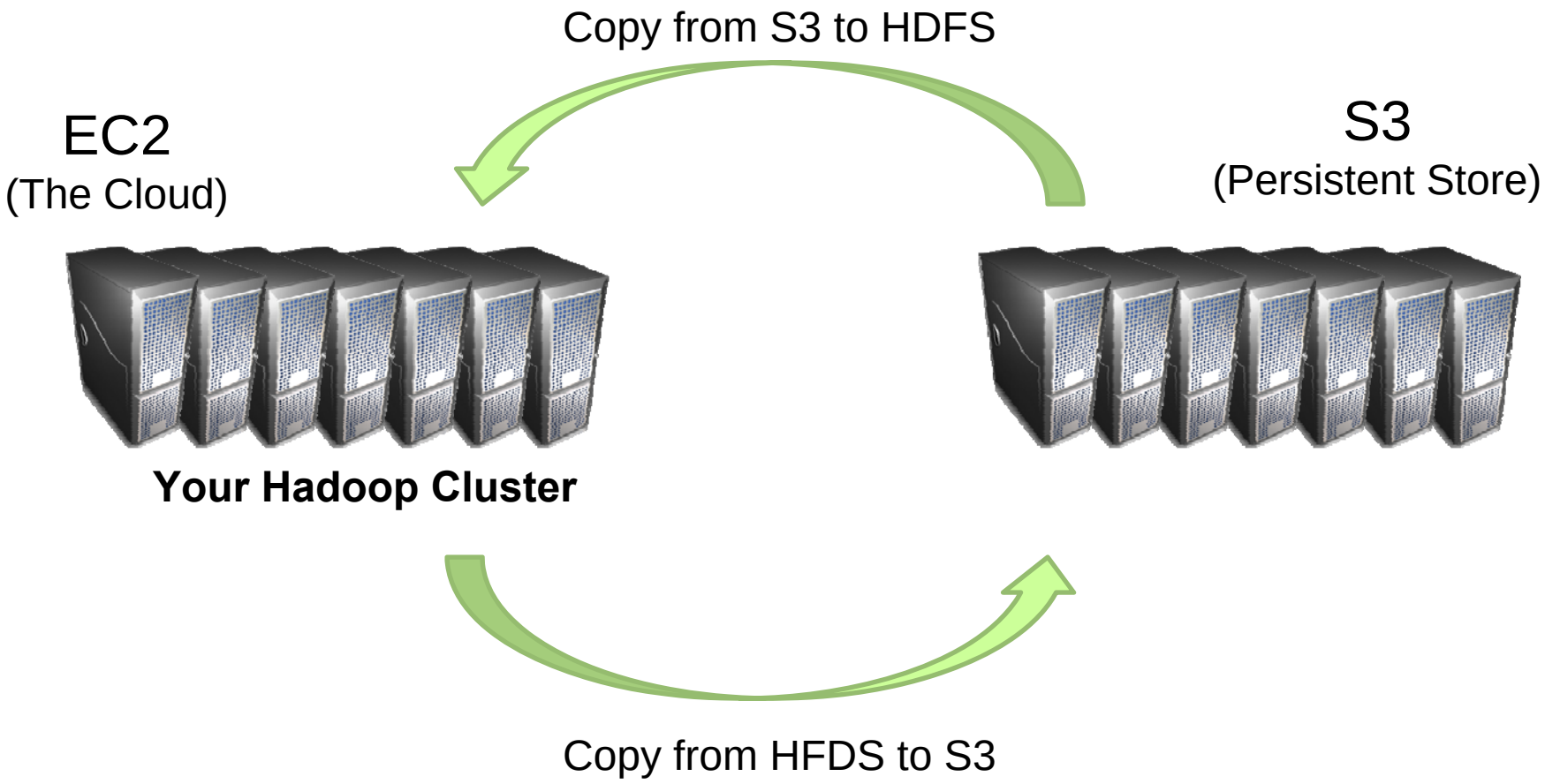


On Amazon: With EC2



Uh oh. Where did the data go?

On Amazon: EC2 and S3



Debugging Hadoop

- First, take a deep breath
- Start small, start locally
- Strategies
  - Learn to use the webapp
  - Where does println go?
  - Don’t use println, use logging
  - Throw RuntimeExceptions

Recap

- Hadoop data types
- Anatomy of a Hadoop job
- Hadoop jobs, end to end
- Software development workflow

Source: Wikipedia (Japanese rock garden)