“Hadoop”: A Distributed Architecture, FileSystem, & MapReduce

Stony Brook University
CSE545 - Spring 2019
Classical Data Mining
Classical Data Mining
Classical Data Mining
Classical Data Mining
IO Bounded

Reading a word from disk versus main memory: $10^5$ slower!

Reading many contiguously stored words is faster per word, but fast modern disks still only reach 150MB/s for sequential reads.
IO Bounded

Reading a word from disk versus main memory: $10^5$ slower!

Reading many contiguously stored words is faster per word, but fast modern disks still only reach 150MB/s for sequential reads.

IO Bound: biggest performance bottleneck is reading / writing to disk.

starts around 100 GBs: $\sim$10 minutes just to read

200 TBs: $\sim$20,000 minutes = 13 days
Classical Big Data Analysis

**Classical focus:** efficient use of disk. e.g. Apache Lucene / Solr

**Classical limitation:** Still bounded when needing to process all of a large file.
IO Bound

How to solve?
Distributed Architecture (Cluster)
Distributed Architecture (Cluster)

In reality, modern setups often have multiple cpus and disks per server, but we will model as if one machine per cpu-disk pair.
Distributed Architecture (Cluster)
Challenges for IO Cluster Computing

1. Nodes fail
   1 in 1000 nodes fail a day

2. Network is a bottleneck
   Typically 1-10 Gb/s throughput

3. Traditional distributed programming is often ad-hoc and complicated
Challenges for IO Cluster Computing

1. Nodes fail
   1 in 1000 nodes fail a day
   **Duplicate Data**

2. Network is a bottleneck
   Typically 1-10 Gb/s throughput
   **Bring computation to nodes, rather than data to nodes.**

3. Traditional distributed programming is often ad-hoc and complicated
   **Stipulate a programming system that can easily be distributed**
Challenges for IO Cluster Computing

1. Nodes fail
   1 in 1000 nodes fail a day
   **Duplicate Data**

2. Network is a bottleneck
   Typically 1-10 Gb/s throughput
   **Bring computation to nodes, rather than data to nodes.**

3. Traditional distributed programming is often ad-hoc and complicated
   **Stipulate a programming system that can easily be distributed**

**MapReduce Accomplishes**
Distributed File System

The effectiveness of MapReduce is in part simply due to use of a distributed filesystem!
Characteristics for Big Data Tasks

Large files (i.e. >100 GB to TBs)

Reads are most common

No need to update in place (append preferred)
Distributed File System

(e.g. Apache HadoopDFS, GoogleFS, EMRFS)

C, D: Two different files

(chunk server 1) (chunk server 2) (chunk server 3) (chunk server n)

(Leskovec at al., 2014; http://www.mmds.org/)
Distributed File System

e.g. Apache HadoopDFS, GoogleFS, EMRFS

C, D: Two different files

(chunk server 1)  (chunk server 2)  (chunk server 3)  (chunk server n)

(Leskovec at al., 2014; http://www.mmds.org/)
Distributed File System

(e.g. Apache HadoopDFS, GoogleFS, EMRFS)

C, D: Two different files

(chunk server 1)  (chunk server 2)  (chunk server 3)  (chunk server n)

(Leskovec at al., 2014; http://www.mmds.org/)
Components of a Distributed File System

Chunk servers (on Data Nodes)

- File is split into contiguous chunks
- Typically each chunk is 16-64MB
- Each chunk replicated (usually 2x or 3x)
- Try to keep replicas in different racks

(Leskovec at al., 2014; http://www.mmds.org/)
Components of a Distributed File System

**Chunk servers (on Data Nodes)**
- File is split into contiguous chunks
- Typically each chunk is 16-64MB
- Each chunk replicated (usually 2x or 3x)
- Try to keep replicas in different racks

**Name node (aka master node)**
- Stores metadata about where files are stored
- Might be replicated or distributed across data nodes.

**Client library for file access**
- Talks to master to find chunk servers
- Connects directly to chunk servers to access data

(Leskovec at al., 2014; http://www.mmds.org/)
Challenges for IO Cluster Computing

1. Nodes fail
   1 in 1000 nodes fail a day
   \textbf{Duplicate Data (Distributed FS)}

2. Network is a bottleneck
   Typically 1-10 Gb/s throughput
   \textit{Bring computation to nodes, rather than data to nodes.}

3. Traditional distributed programming is often ad-hoc and complicated
   \textit{Stipulate a programming system that can easily be distributed}
What is MapReduce?

*noun* 1 - A style of programming

input chunks => **map tasks** | **group_by keys** | **reduce tasks** => output

“|” is the linux “pipe” symbol: passes stdout from first process to stdin of next.

E.g. counting words:

    tokenize(document) | sort | uniq -C
What is MapReduce?

**noun.1 - A style of programming**

input chunks => map tasks | group_by keys | reduce tasks => output

“|” is the Linux “pipe” symbol: passes stdout from first process to stdin of next.

E.g. counting words:

```
    tokenize(document) | sort | uniq -C
```

**noun.2 - A system that distributes MapReduce style programs across a distributed file-system.**

(e.g. Google’s internal “MapReduce” or apache.hadoop.mapreduce with hdfs)
What is MapReduce?
What is MapReduce?

- **Input chunks** → **Map**
  - Extract what you care about.
  - line => (k, v)

- **Key-value pairs** (k, v) → **Reduce tasks**
  - Group by keys
  - Keys with all their values (k, [v, w,...]) → **Combined output**
What is MapReduce?

Map

extract what you care about.

sort and shuffle

many \((k, v) \Rightarrow (k, [v_1, v_2]), \ldots\)

combined output

Group by keys

Reduce tasks
What is MapReduce?

Map:
- extract what you care about.

Reduce:
- aggregate, summarize

Input chunks → sort and shuffle → Keys with all their values (k, [v, w, ...]) → Group by keys → Combined output
What is MapReduce?

(Leskovec et al., 2014; http://www.mmds.org/)
The Map Step

Input key-value pairs

Intermediate key-value pairs

(map)

... (Leskovec et al., 2014; http://www.mmds.org/)
The Sort / Group By Step

(Leskovec at al., 2014; http://www.mmds.org/)
The Reduce Step

Key-value groups

Output key-value pairs

reduce

reduce

...  ...

k v

k v

(Leskovec at al., 2014; http://www.mmds.org/)
What is MapReduce?

(Leskovec at al., 2014; http://www.mmds.org/)
What is MapReduce?

Map: (k, v) -> (k’, v’)*
(Written by programmer)

Group by key: (k’, v1’), (k2’, v2’), ... -> (k1’, (v1’, v’, ...)),
(system handles) (k2’, (v1’, v’, ...)), ...

Reduce: (k’, (v1’, v’, ...)) -> (k’, v”)*
(Written by programmer)
Example: Word Count

tokenize(document) | sort | uniq -C
Example: Word Count

tokenize(document) | sort | uniq -C

Map: extract what you care about.

sort and shuffle

Reduce: aggregate, summarize
Example: Word Count

The crew of the space shuttle Endeavor recently returned to Earth as ambassadors, harbingers of a new era of space exploration. Scientists at NASA are saying that the recent assembly of the Dextre bot is the first step in a long-term space-based man/machine partnership.

"The work we're doing now -- the robotics we're doing -- is what we're going to need ..."
The crew of the space shuttle Endeavor recently returned to Earth as ambassadors, harbingers of a new era of space exploration. Scientists at NASA are saying that the recent assembly of the Dextre bot is the first step in a long-term space-based man/machine partnership. "The work we're doing now -- the robotics we're doing -- is what we're going to need ....................

**MAP:**
Read input and produces a set of key-value pairs

(The, 1)
(crew, 1)
(of, 1)
(the, 1)
(space, 1)
(shuttle, 1)
(Endeavor, 1)
(recently, 1)

Big document (key, value)
The crew of the space shuttle Endeavor recently returned to Earth as ambassadors, harbingers of a new era of space exploration. Scientists at NASA are saying that the recent assembly of the Dextre bot is the first step in a long-term space-based man/machine partnership. "The work we're doing now -- the robotics we're doing - - is what we're going to need ..................
The crew of the space shuttle Endeavor recently returned to Earth as ambassadors, harbingers of a new era of space exploration. Scientists at NASA are saying that the recent assembly of the Dextre bot is the first step in a long-term space-based man/machine partnership. "The work we're doing now -- the robotics we're doing -- is what we're going to need .......
The crew of the space shuttle Endeavor recently returned to Earth as ambassadors, harbingers of a new era of space exploration. Scientists at NASA are saying that the recent assembly of the Dextre bot is the first step in a long-term space-based man/machine partnership. "The work we're doing now -- the robotics we're doing -- is what we're going to need ..."
Example: Word Count

@abstractmethod
def map(k, v):
    pass

@abstractmethod
def reduce(k, vs):
    pass
Example: Word Count (version 1)

```python
def map(k, v):
    for w in tokenize(v):
        yield (w,1)

def reduce(k, vs):
    return len(vs)
```
Example: Word Count (version 2)

def map(k, v):
    counts = dict()
    for w in tokenize(v):
        try:
            counts[w] += 1
        except KeyError:
            counts[w] = 1
    for item in counts.items():
        yield item

def reduce(k, vs):
    return sum(vs)
Challenges for IO Cluster Computing

1. Nodes fail
   1 in 1000 nodes fail a day
   **Duplicate Data (Distributed FS)**

2. Network is a bottleneck
   Typically 1-10 Gb/s throughput **(Sort & Shuffle)**
   Bring computation to nodes, rather than data to nodes.

3. Traditional distributed programming is often ad-hoc and complicated
   **Stipulate a programming system that can easily be distributed**
Challenges for IO Cluster Computing

1. Nodes fail
   1 in 1000 nodes fail a day
   Duplicate Data (Distributed FS)

2. Network is a bottleneck
   Typically 1-10 Gb/s throughput (Sort & Shuffle)
   Bring computation to nodes, rather than data to nodes.

3. Traditional distributed programming is often ad-hoc and complicated
   (Simply requires Mapper and Reducer)
   Stipulate a programming system that can easily be distributed