Hadoop - Distributed Computing for the Masses

Casey Stella

November 7, 2012

Table of Contents

Hi, I'm Casey

- Education
  - B.S. in Computer Science from University of Louisiana at Monroe
  - M.S. in Mathematics from Texas A&M University with an emphasis in Theoretical Computer Science

Hi, I'm Casey

- Education
  - B.S. in Computer Science from University of Louisiana at Monroe
  - M.S. in Mathematics from Texas A&M University with an emphasis in Theoretical Computer Science
- I tend to work with "Big Data"
  - I just joined Hortonworks as a Systems Architect
  - I just left the high performance indexing team at Explorys, a medical informatics startup based in the Cleveland Clinic
  - I was a Research Geophysicist in the Oil Industry doing signal processing
  - I built a VOIP network oriented toward WoW gamers

Distributed Computing

- Distributed computing turns out to be hard

Distributed Computing

- Distributed computing turns out to be hard
  - Needs to be fast

Distributed Computing

- Distributed computing turns out to be hard
  - Needs to be fast
  - Needs to be scalable

Distributed Computing

- Distributed computing turns out to be hard
  - Needs to be fast
  - Needs to be scalable
  - Needs to be fault-tolerant

Distributed Computing

- Distributed computing turns out to be hard
  - Needs to be fast
  - Needs to be scalable
  - Needs to be fault-tolerant
- Message Passing Systems
  - Idea: build your own systems from a set of communication primitives
  - Often designed specifically to solve one very big, very nasty problem

Distributed Computing

- Distributed computing turns out to be hard
  - Needs to be fast
  - Needs to be scalable
  - Needs to be fault-tolerant
- Message Passing Systems
  - Idea: build your own systems from a set of communication primitives
  - Often designed specifically to solve one very big, very nasty problem
  - Pro: very flexible and high performance

Distributed Computing

- Distributed computing turns out to be hard
  - Needs to be fast
  - Needs to be scalable
  - Needs to be fault-tolerant
- Message Passing Systems
  - Idea: build your own systems from a set of communication primitives
  - Often designed specifically to solve one very big, very nasty problem
  - Pro: very flexible and high performance
  - Con: can be very difficult to maintain, write and understand. Often designed for massive parallel computers.

Map-Reduce

- A programming model for processing large data sets in batch.

Map-Reduce

- A programming model for processing large data sets in batch.
- Designed to execute on a cluster of commodity hardware.

Map-Reduce

- A programming model for processing large data sets in batch.
- Designed to execute on a cluster of commodity hardware.
- Let tasks fail and be able to retry.

Map-Reduce

- A programming model for processing large data sets in batch.
- Designed to execute on a cluster of commodity hardware.
- Let tasks fail and be able to retry.
- Bring the code to the data, rather than the data to the code.

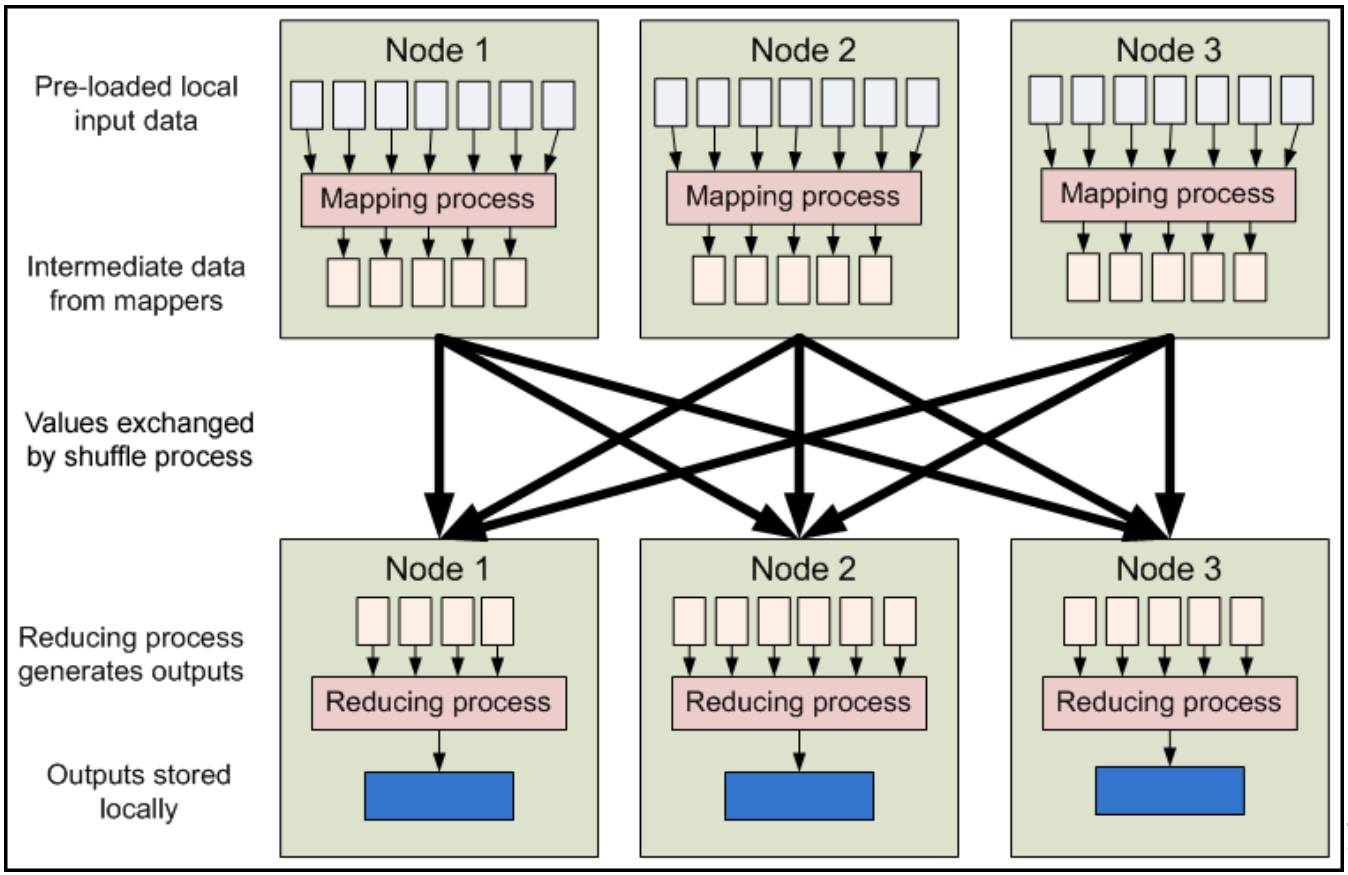
Map-Reduce

- A programming model for processing large data sets in batch.
- Designed to execute on a cluster of commodity hardware.
- Let tasks fail and be able to retry.
- Bring the code to the data, rather than the data to the code.
- Limit communication by allowing only a certain set of operations in a flow.

Map-Reduce

- A programming model for processing large data sets in batch.
- Designed to execute on a cluster of commodity hardware.
- Let tasks fail and be able to retry.
- Bring the code to the data, rather than the data to the code.
- Limit communication by allowing only a certain set of operations in a flow.
- Inspired by functional programming

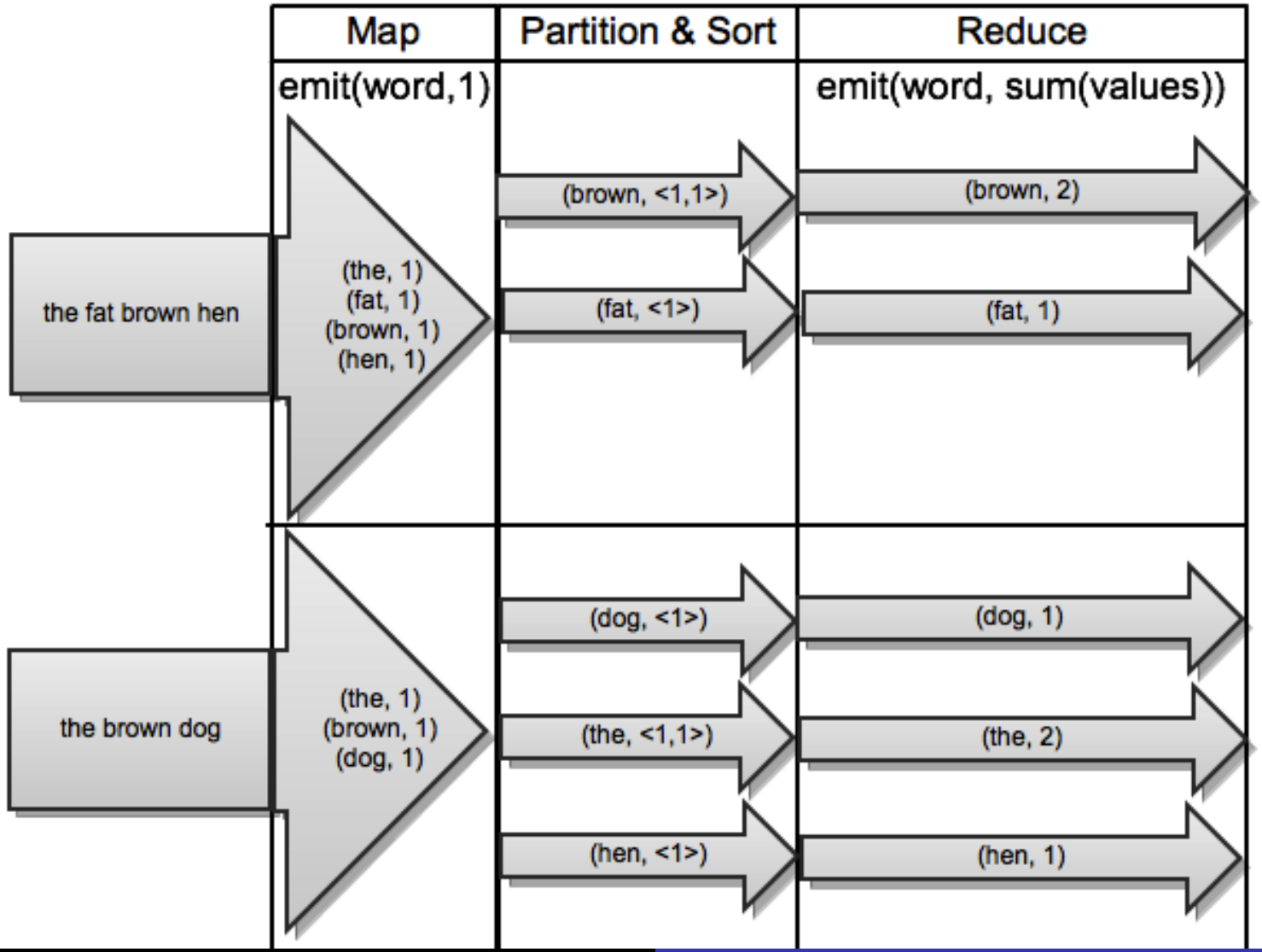
Data Flow



[1]

[1] http://developer.yahoo.com/hadoop/tutorial/module4.html

Word Count - The Canonical Example



Caveats

- All problems do not fit well (or at all) within this model.

Caveats

- All problems do not fit well (or at all) within this model.
- This model is not suitable for real-time processing of data.

Caveats

- All problems do not fit well (or at all) within this model.
- This model is not suitable for real-time processing of data.
- While it can linearly scale out in relation to your input data, if you choose the number of keys you emit from your mappers improperly, you could create a non-linear scale-out.

Caveats

- All problems do not fit well (or at all) within this model.
- This model is not suitable for real-time processing of data.
- While it can linearly scale out in relation to your input data, if you choose the number of keys you emit from your mappers improperly, you could create a non-linear scale-out.
- Easier than rolling your own solution, but can be tricky to fit your problem to it.

Apache Hadoop – History

- Hadoop was derived from Google's MapReduce and Google File System (GFS) papers.

Apache Hadoop – History

- Hadoop was derived from Google's MapReduce and Google File System (GFS) papers.
- Open source, Apache licensed project created by Doug Cutting and Michael J. Cafarella.

Apache Hadoop – History

- Hadoop was derived from Google's MapReduce and Google File System (GFS) papers.
- Open source, Apache licensed project created by Doug Cutting and Michael J. Cafarella.
- Named after Doug's son's toy elephant.

Apache Hadoop – History

- Hadoop was derived from Google's MapReduce and Google File System (GFS) papers.
- Open source, Apache licensed project created by Doug Cutting and Michael J. Cafarella.
- Named after Doug's son's toy elephant.
- It was originally developed to support distribution for the Nutch search engine project.

Apache Hadoop – History

- Hadoop was derived from Google's MapReduce and Google File System (GFS) papers.
- Open source, Apache licensed project created by Doug Cutting and Michael J. Cafarella.
- Named after Doug's son's toy elephant.
- It was originally developed to support distribution for the Nutch search engine project.
- Has grown to have a whole ecosystem around it: HBase, HDFS, etc.

The Components

- Zookeeper - Configuration, synchronization and naming registry for distributed systems
- HDFS - An append-only distributed filesystem inspired by Google File System
- Hadoop Map-Reduce - The Map-Reduce infrastructure
- HBase - A column-store NoSQL database inspired by Google BigTable

Usage

- Core part of Netflix's analytics stack
- Facebook is a heavy user of map-reduce and HBase runs Facebook's messaging platform
- Yahoo! continues to be a heavy user
- O'Reilly Strata and Hadoop World merged this year. Over 2500 attendees.

Conclusion

- Thanks for your attention
- Follow me on twitter @*casey_stella*
- Find me at
  - `http://caseystella.com`
  - `https://github.com/cestella`