

# Big Data: Data Analysis Boot Camp Hadoop and R

Chuck Cartledge, PhD

21 January 2018

# Table of contents (1 of 1)

## 1 Intro.

## 2 Basics

- Hadoop Distributed File System (hdfs)
- Map/Reduce computing model

## 3 Hands-on

- Word count

- Airports and travel

## 4 Q & A

## 5 Conclusion

## 6 References

## 7 Files

# What are we going to cover?

- 1 Look at the Hadoop map-reduce programming model
- 2 Pick apart the “classic” map-reduce word count program
- 3 Look at how the map-reduce model can be used with complex keys



# The Hadoop Distributed File System (HDFS)

*“The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets. HDFS relaxes a few POSIX requirements to enable streaming access to file system data. HDFS was originally built as infrastructure for the Apache Nutch web search engine project. HDFS is now an Apache Hadoop subproject.”*

A. Staff [2]

## HDFS Assumptions and Goals[2]

**Hardware Failure** Hardware failure is the norm rather than the exception.

**Streaming Data Access** Applications that run on HDFS need streaming access to their data sets.

**Large Data Sets** Applications that run on HDFS have large data sets. A typical file in HDFS is gigabytes to terabytes in size.

**Simple Coherency Model** HDFS applications need a write-once-read-many access model for files.

**Moving Computation is Cheaper than Moving Data** A computation requested by an application is much more efficient if it is executed near its data.

**Portability Across Heterogeneous Hardware and Software Platforms** HDFS has been designed to be easily portable from one platform to another.

## HDFS Implementations[3]

**Hardware Failure** Redundant copies of the data are kept by the system.

**Streaming Data Access** Applications that run on HDFS need streaming access to their data sets. Programs read and write data from and to STDIN and STDOUT.

**Large Data Sets** An HDFS data file is “chuncked” to minimize total program execution time.

**Simple Coherency Model** HDFS trades off some POSIX requirements for performance, so some operations may behave differently than you expect them to.

**Moving Computation is Cheaper than Moving Data** map() are copied to the data and the results are copied to the reducer functions.

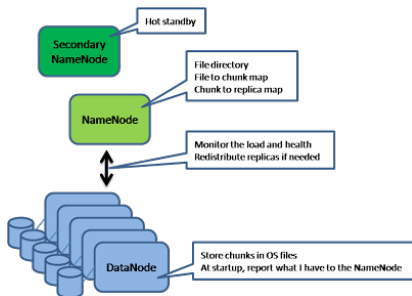
**Portability Across Heterogeneous Hardware and Software Platforms** Systems IAW standards gain market share.



# HDFS terminology

## Some terms:

- namenode** manages the filesystem namespace. It maintains the filesystem tree and the metadata for all the files and directories in the tree.
- client** accesses the filesystem on behalf of the user by communicating with the namenode and datanodes. The client presents a POSIX-like filesystem interface.
- datanode** are the workhorses of the filesystem. They store and retrieve blocks when they are told to (by clients or the namenode).



Our applications are clients, and the mysteries of the name and data nodes are hidden from us.

Image from [1].

# Same image.

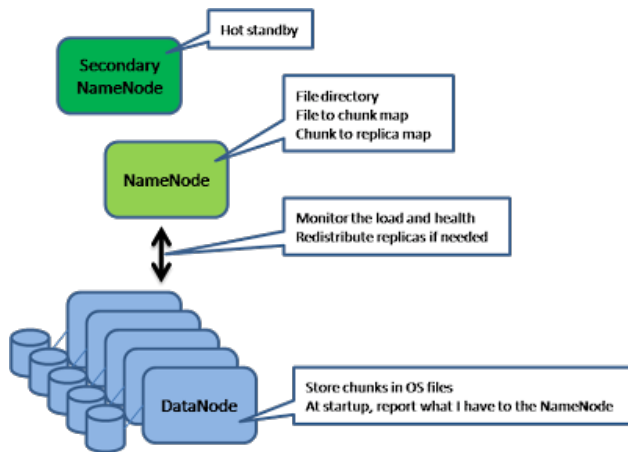


Image from [1].



## Map reduce model from 50,000 foot view.

A simple and powerful model:

- 1 A line of data is presented to a “mapper” function.
- 2 The “mapper” outputs 0 or more key and value tuples per presented input line
- 3 Hadoop sorts and merges all keys and values so that there is one key with one or more values
- 4 The “reducer” processes each key and associated values to the output

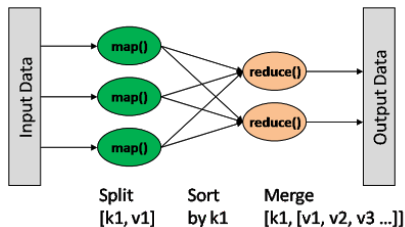


Image from [1].

Same image.

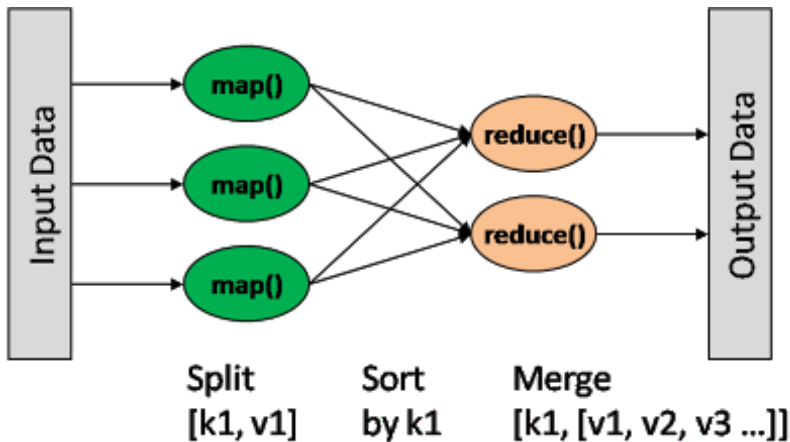


Image from [1].



## A lower level view

There are a lot of processes and coordination happening behind the scenes. The client submits a job to Hadoop, mapper functions are copied to the data, key values are sorted, then presented to the reducers, and output is written. Much of this activity can be monitored at port 8787.

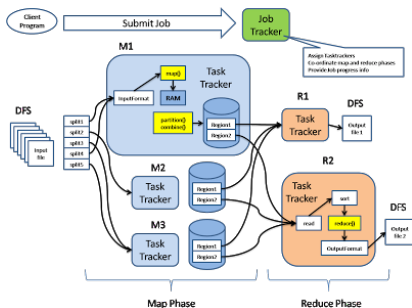


Image from [1].



# Same image.

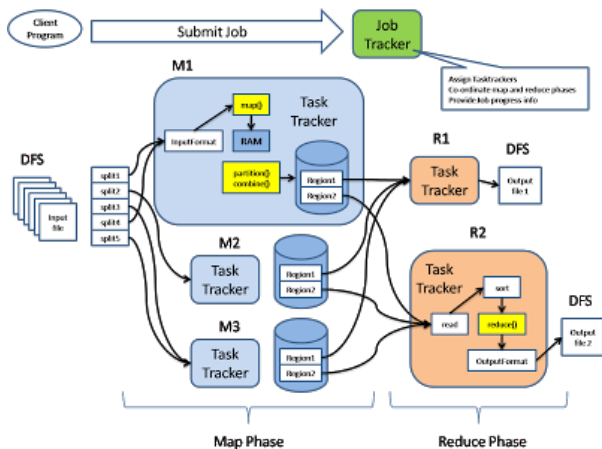
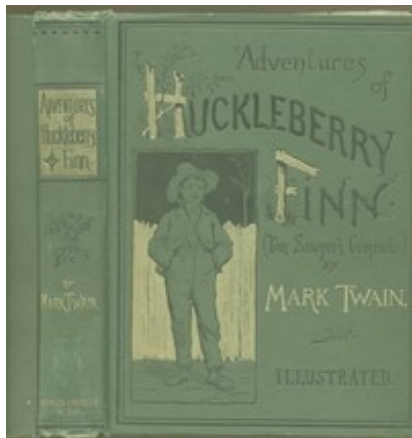


Image from [1].

## Classic word count program

The program is in the attached file (Hadoop word count). We'll:

- 1 Set some environment variables for Hadoop
- 2 Load necessary R libraries
- 3 Download and save the text file
- 4 Do some HDFS housekeeping
- 5 Define and execute the map-reduce job
- 6 See where the results ended up



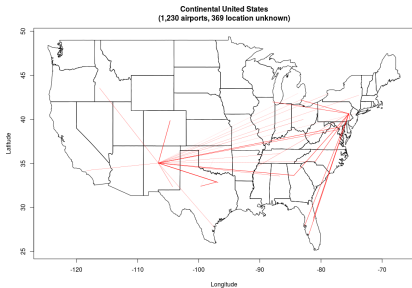
## Ways to modify the word count program.

- Remove all “words” that are in fact a space
- Remove all “stop” words
- Remove all words that are numbers
- Stem all words
- Process a different text file
- Create a histogram of the first  $n$  most common words
- Estimate the “reading” level of the processed text
- Create a word cloud in the shape of something associated with the text

# Looking at air traffic between US domestic airports (attached Airport route exploration)

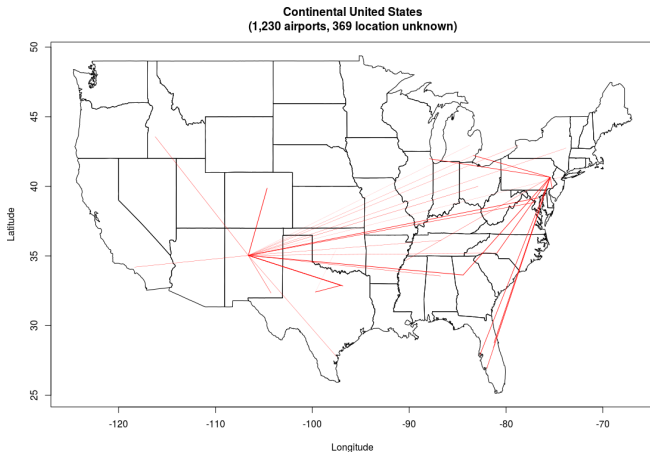
Mashing data from different sources.

- Use the US Government Bureau of Transportation Statistics to get route data
- Use the OpenFlights to find airport latitude and longitude
- Use Hadoop map/reduce model to create a pivot table
- Plot results



Attached file.

# Same image.



Attached file.







# BTS Airlines and Airports page

OST-R | BTS | TranStats - Opera

OST-R | BTS | Trans: X

www.transtats.bts.gov/Tables.asp

United States Department of Transportation

Ask-A-Librarian | A-Z Index

## Bureau of Transportation Statistics

Explore Topics and Geography | Browse Statistical Products and Data | Learn About BTS and Our Work | Newsroom

OST-R > BTS

**TranStats**

Search this site:

[Advanced Search](#)

**Resources**

- [Database Directory](#)
- [Glossary](#)
- [Upcoming Releases](#)
- [Data Release History](#)

**Data Finder**

**By Mode**

- Aviation
- Maritime
- Highway
- Transit
- Rail
- Pipeline
- Bike/Pedestrian
- Other

**By Subject**

- Safety

**Database Name: Air Carrier Statistics (Form 41 Traffic)- U.S. Carriers**

[Database Profile](#)

All Rows Shown

Table Name	Description
T-100 Domestic Market (U.S. Carriers)	<b>Note:</b> Over time both the code and the name of a carrier may change and the same code or name may be assumed by a different airline. To ensure that you are analyzing data from the same airline, TranStats provides four airline-specific variables that identify one and only one carrier or its entity: Airline ID (AirlineID), Unique Carrier Code (UniqueCarrier), Unique Carrier Name (UniqueCarrierName), and Unique Entity (UniqueCarrierEntity). A unique airline (carrier) is defined as one holding and reporting under the same DOT certificate regardless of its Code, Name, or holding company/corporation. US Airways and America West started to report combined on-time data in January 2006 and combined traffic and financial data in October 2007 following their 2005 merger announcement. Delta and Northwest began reporting jointly in January 2010 following their 2008 merger announcement. Continental Micronesia was combined into Continental Airlines in December 2010 and joint reporting began in January 2011. Atlantic Southeast and ExpressJet began reporting jointly in January 2012. United and Continental began reporting jointly in January 2012 following their 2010 merger announcement. Endeavor (SE) operated as Penair prior to August 2013. Envoy (MQ) operated as American Eagle prior to April 2014. Southwest (WN) and AirTran (FL) began reporting jointly in January 2015 following their 2011 merger announcement. American (AA) and US Airways (US) began reporting jointly as AA in July 2015 following their 2013 merger announcement.
T-100 Domestic Market (U.S. Carriers)	This table contains domestic market data reported by U.S. air carriers, including carrier, origin, destination, and service class for enplaned passengers, freight and mail when both origin and destination airports are located within the boundaries of the United States and its territories. <a href="#">Table Profile</a> <a href="#">Carrier Release Status</a> <a href="#">Download</a>
T-100 Domestic Segment (U.S. Carriers)	This table contains domestic non-stop segment data reported by U.S. air carriers, including carrier, origin, destination, aircraft type and service class for transported passengers, freight and mail, available capacity, scheduled departures, departures performed, aircraft hours, and load factor when both origin and destination airports are located within the boundaries of the United States and its territories. <a href="#">Table Profile</a> <a href="#">Carrier Release Status</a> <a href="#">Download</a>

https:

//www.transtats.bts.gov/Tables.asp?DB\_ID=110&DB\_Name=

Air%20Carrier%20Statistics%20%28Form%2041%20Traffic%

# BTS Domestic Data Selection page

The screenshot shows the 'DL SelectFields.asp' page on the Bureau of Transportation Statistics website. The page is titled 'All Carriers - 1-398 Domestic Market (U.S. Carriers)'. It includes a search bar and filters for Geography, Year, and Period. A table of fields is displayed, with columns for Field Name, Description, and a 'Select' checkbox. The fields listed are:

Field Name	Description	Select
<input checked="" type="checkbox"/> Passengers	On-Flight Market Passengers Exploded	
<input checked="" type="checkbox"/> Freight	On-Flight Market Freight Exploded (weights)	
<input checked="" type="checkbox"/> Mail	On-Flight Market Mail Exploded (weights)	
<input type="checkbox"/> Distance	Distance between airports (miles)	
<b>Carrier</b>		
<input checked="" type="checkbox"/> UniqueCarrier	Unique Carrier Code. When the same code has been used by multiple carriers, a numeric suffix is used for multiple carriers, for example, AA, AA(1), AA(2). Use this field for analysis across all carrier codes.	<a href="#">Get Lookup Table</a>
<input type="checkbox"/> Arrived	An identification number assigned by US DOT to identify unique arrival carriers. A unique arrival carrier ID is defined as a holding and reporting under the same DOT certificate regardless of the Code, Name, or holding company/corporation.	<a href="#">Get Lookup Table</a>
<input type="checkbox"/> UniqueCarrierName	Unique Carrier Name. When the same carrier has been used by multiple carriers, a numeric suffix is used for multiple carriers, for example, Air Caribbean, Air Caribbean (1).	<a href="#">Get Lookup Table</a>
<input type="checkbox"/> UniqueCarrierCity	Unique Entity for a Carrier's Operation Region.	<a href="#">Get Lookup Table</a>
<input type="checkbox"/> CarrierGroup	Carrier's Operation Region. Carriers Report Data by Operation Region.	<a href="#">Get Lookup Table</a>
<input type="checkbox"/> Carrier	Code assigned by IATA and commonly known to identify a carrier. As the same code may have been assigned to different carriers over time, the code is not always unique. For analysis, use the Unique Carrier Code.	<a href="#">Get Lookup Table</a>
<input type="checkbox"/> CarrierName	Carrier Name	<a href="#">Get Lookup Table</a>
<input type="checkbox"/> CarrierGroup	Carrier Group Code. Used in Legacy Analysis.	<a href="#">Get Lookup Table</a>
<input type="checkbox"/> CarrierGroupShort	Carrier Group Name	<a href="#">Get Lookup Table</a>
<b>Origin</b>		
<input type="checkbox"/> OriginAirportID	Origin Airport. Airport ID. An identification number assigned by US DOT to identify a	<a href="#">Get Lookup Table</a>

[https://www.transtats.bts.gov/DL\\_SelectFields.asp](https://www.transtats.bts.gov/DL_SelectFields.asp)

# OpenFlights home page

OpenFlights: Airport and airline data - Opera

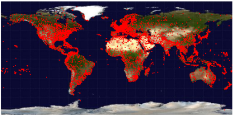
OpenFlights: Airpor x +

openflights.org | data.html

Airport, airline and route data

Navigation: [Airport](#) | [Airline](#) | [Route](#) | [Schedule](#) | [Other](#) | [License](#)

Airport database



(click to enlarge)

As of January 2017, the OpenFlights Airports Database contains over 10,000 airports, train stations and ferry terminals spanning the globe, as shown in the map above. Each entry contains the following information:

**Airport ID** Unique OpenFlights identifier for this airport.

**Name** Name of airport. May or may not contain the city name.

**City** Main city served by airport. May be spelled differently from **Name**.

**Country** Country or territory where airport is located. See [countries.dat](#) to cross-reference to ISO 3166-1 codes.

**IATA** 3-letter IATA code. Null if not assigned/unknown.

**ICAO** 4-letter ICAO code. Null if not assigned.

**Latitude** Decimal degrees, usually to six significant digits. Negative in South, positive in North.

**Longitude** Decimal degrees, usually to six significant digits. Negative in West, positive in East.

**Altitude** In feet.

**Timezone** Hours offset from UTC. Fractional hours are expressed as decimals, eg. India in 5.5.

**DST** Daylight savings time. One of E (Europe), A (US/Canada), S (South America), O (Australia), Z (New Zealand), N (None) or U (Unknown). See also [Date\\_Tzdb](#).

**Tr database** Timezone in "I" (Iceland format), eg. "America/Los\_Angeles".

**Time zone**

**Type** Type of the airport. Value "airport" for air terminals, "station" for train stations, "port" for ferry terminals and "unknown" if not known. In airports.csv only type=airport is included.

**Source** Source of this data. "OurAirports" for data sourced from [OurAirports](#), "Legacy" for old data not matched to OurAirports (mostly DAFIF), "User" for unverified user contributions. In airports.csv, only source=OurAirports is included.

The data is UTF-8 (Unicode) encoded.

Note: Rules for daylight savings time change from year to year and from country to country. The current data is an approximation for 2009, built on a country level. Most airports in DST-less regions in countries that generally observe DST (eg. AL, HI in the USA, NT, QL in Australia, parts of Canada) are marked incorrectly.

Sample entries

Aflstate

**FAST, FAIR, AND HASSLE-FREE CLAIM**

or your money back only from Aflstate

**QUOTE NOW**

<https://openflights.org/data.html>

## Lessons learned about `keyval()`

Some “interesting” things about the `keyval()` function:

- 1 The last call wins. If your processing creates a collection of key value pairs, the last `keyval()` call is the data passed to `reduce()`.
- 2 `keyval()` is vectorized. There can be more than one key or value passed to the function.

To pass more than one key value combination, use:

```
keyval(c(...), c(...))
```

Be aware that the shorter argument will be recycled as necessary to match the longer argument.

Execute `keyval` at the R prompt to see code.

## Ways to modify the airport program.

- Change the lines between airports to great circle routes
- Reduce the number of routes to those that carry the greatest weight
- See the difference between cargo and passenger routes
- Modify the routes to show source and destination
- Identify the most common carriers by weight
- Identify the most frequent carriers
- Compute net weight exchange between airports (find sources and sinks)
- If the data is for US domestic routes, why are there links to Chile
- Expand the list of airport locations to remove all unknown locations

## Q & A time.

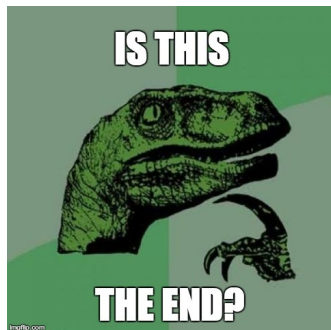
Q: How many Oregonians does it take to screw in a light bulb?

A: Three. One to screw in the light bulb and two to fend off all those Californians trying to share the experience.



## What have we covered?

- Gained an understanding of how R interfaces with the Hadoop map-reduce programming model
- “Played” with a word count program
- Looked at things that airlines carry between airports and how to display that data



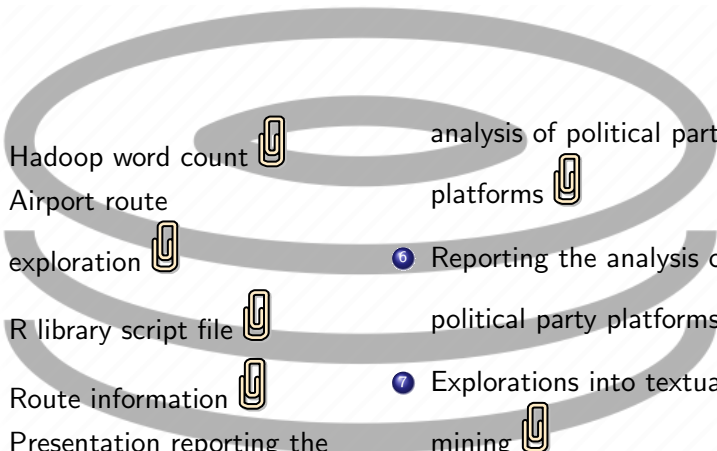







Next: BDAR Chapter 5, RDBMSs and R



## References (1 of 1)

- [1] Ricky Ho, [How Hadoop Map/Reduce works](#), 2008.
- [2] Apache Staff, [HDFS Architecture Guide](https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html), [https://hadoop.apache.org/docs/r1.2.1/hdfs\\_design.html](https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html), 2017.
- [3] Tom White, [Hadoop: The Definitive Guide, 4th Edition](#), O'Reilly Media, Inc., 2015.

## Files of interest

- 
- 1 Hadoop word count  analysis of political party
  - 2 Airport route  platforms 
  - 3 R library script file  Reporting the analysis of
  - 4 Route information  political party platforms 
  - 5 Presentation reporting the  6 Explorations into textual
  - 7 mining 