

Using Big Data Tools when there are Holes in the Data

Tidewater Big Data Enthusiasts
Chuck Cartledge
Developer

July 26, 2016 at 9:34pm

Contents

List of Figures	i
List of Tables	ii
1 Introduction	1
1.1 Hbase origins	1
1.2 What's ahead	1
2 HBase compared to Relational Database Systems	2
3 Approach	5
4 Results	7
4.1 IMDb explorations	7
4.2 NDBC explorations	17
5 Conclusion	28
A Selected stations	30
B Misc. files	35
C References	37

List of Figures

1	A columnar database showing column families.	3
2	HBase ecosystem.	6
3	Internet Movie Database home page.	9
4	IMDb Alternate Interfaces home page.	10
5	IMDb Alternate Interfaces FTP home page.	11
6	A partial dump of the IMDb movies listing.	12
7	A partial dump of the IMDb crazy credits listing.	13
8	A partial dump of the IMDb director listing.	14
9	Histogram of crazy credits per movies.	15
10	Histogram of directors per movie.	16
11	National Data Buoy Center (NDBC) home page.	18
12	NDBC real time data home page.	19
13	NDBC real time data download page.	20
14	Sample data from station 41013.	21
15	Sample station location data.	22
16	Buoy reporting status worldwide.	23
17	Buoy reporting status near continental USA.	24
18	Buoy reporting status near Virginia.	25
19	Station 41013 wind file.	26
20	Buoy reporting status worldwide (bad data).	27

List of Tables

1	Example of a RDBMS table.	4
2	Example of a columnar database.	4
3	A collection of interesting stations.	30

1 Introduction

We'll explore the world of columnar databases. Databases that have rows and columns, but the intersection of a row and a column can have 0 or more values. The values can be versioned, timestamped for automatic deletion, and other neat features. We'll look at HBase (one of many databases built on top of Hadoop), to explore some of the data in the Internet Movie Database. HBase is used by Adobe, LinkedIn, Netflix, Spotify, and others.

1.1 Hbase origins

In 2008, Google published a paper describing their “BigTable” technology, explaining in detail its internal structure, benefits, and limitations. BigTable is the underlying technology in many of Google's applications.

“A Bigtable is a sparse, distributed, persistent multidimensional sorted map. The map is indexed by a row key, column key, and a timestamp; each value in the map is an uninterpreted array of bytes.”

Chang, et al. [2]

BigTable was created to overcome some the limitations of the Google File System, and the Map Reduce technologies. HBase is a NoSQL database that primarily works on top of Hadoop. HBase is based on the BigTable storage architecture. HBase inherits the storage design from the column-oriented databases and the data access design from the keyvalue store databases where a key-based access to a specific cell of data is provided[3].

BigTable uses row-keys to locate data. The row-keys are arbitrary strings. Each read or write on a row is atomic, regardless of the number of columns in the row. Row-keys are kept in lexicographic order. Column keys are grouped into “column families”, and data is accessed using “family:qualifier” syntax. Where column names must be printable, byt qualifiers can be arbitrary strings. Data in a row is timestamped, where default to current time in microseconds, or a value set by the user. A garbage collection operation runs in the background to automatically delete old data.

1.2 What's ahead

We'll look at two different sources of data in different manners to see how HBase can be used. First we look at the Internet Movie Database (IMDb) to answer a couple of basic questions about movies:

1. How many “crazy credits” are there per movie, and
2. How many directors there are per movie.

Crazy credits are credits added to a movie that are outside the “normal” set of credits, as determined by the IMDb. An example of a crazy credit from the 1983 movie Scarface is:

“Enjoy yourself, every day above ground is a good day.” ANONYMOUS, MIAMI 1981

Secondly we will look at real-time data from the National Data Buoy Center (NDBC)¹. The NDBC monitors and makes available and water measurements from each of the approximately 1,000 floating and stationary buoys it monitors. We will download the buoy measurements and buoy location to see which buoys are active.

We will use Python to collect data from different sources, update data an HBase data base, and present the results of our analysis.

2 HBase compared to Relational Database Systems

A Relational Database Management System (RDBMS) is a row oriented system. Meaning that each entry in a table has the same number of columns (see Figure 1). Data in RDBMS is created, reported, updated, and deleted (CRUD operations) by using Structured Query Language (SQL) commands. By contrast, HBase is a columnar, or column oriented database. Data in a column is stored together, and every “row” may have a different number of columns. Expanding the data from the previous example (see Table 1) with additional (and sometimes) missing data, we can get to a columnar database structure (see Table 2).

In general, it is very expensive to add new columns to a RDBMS table. It is very inexpensive to add a new column to a columnar database, or to change the type or number of elements in a columnar cell. Sometimes a column in a columnar database is called a “column family” in order to make it explicit that more than one value can be “stored” in a cell.

¹<http://www.ndbc.noaa.gov/data/realtime2/>

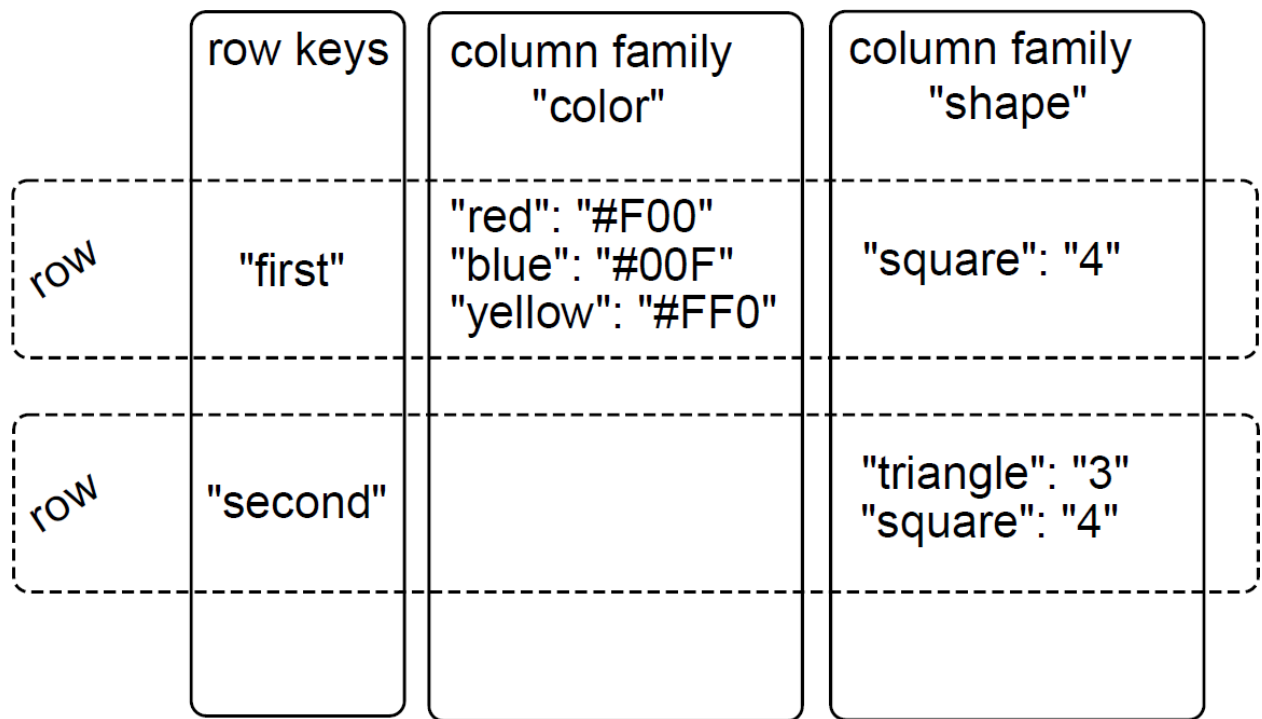


Figure 1: A columnar database showing column families. Image from [4] showing two column families, where different (possibly missing values) in different cells.

Table 1: Example of a RDBMS table. Example data taken from [6]. Each row and column intersection (a cell) has the same type of data. In some cases the data may be NULL, but the cell remains.

Employee_ID	FirstName	LastName	Email	Department_ID
1000	Arun	Jayaram	arun@softwaredeveloper.com	100
1001	Manoj	Shankar	manof@softwaredeveloper.com	100
1002	Syam	Sundar	syam@softwaredeveloper.com	102

Table 2: Example of a columnar database. Each row may have a different number of columns, and the contents of each cell (the intersection of a row and column) may have a different number and type of data elements.

Row key	Name	Contact	Dept	Hobby
1000	“First:Arun” , “Last:Jayaram”	“Email:arun@softwaredeveloper.com” , “mobile:555-123-4567”	“primary:100” , “secondary:103”	
1001	“First:Manoj, “Last:Shankar	“Email:manof@softwaredeveloper.com”	“primary:100”	“main:foosball”
1002	“First:Syam, “Last:Sundar	“Email:syam@softwaredeveloper.com” , “land:555-123-4568”	“primary:102”	“main:paintBall” , “sec- ondary:football”

3 Approach

We were interested in exploring a full up HBase installation, so that meant installing and setting up the following pieces of software:

1. Apache Hadoop - is an open-source software framework for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware[7].
2. Hadoop Distributed File System (HDFS) - a distributed file system designed to run on commodity hardware. It has many similarities with existing distributed file systems[5].
3. Thrift server - Thrift is a software framework that allows you to create cross-language bindings. In the context of HBase, Java is the only first-class citizen. However, the HBase Thrift interface allows other languages to access HBase over Thrift by connecting to a Thrift server that interfaces with the Java client[1].
4. HBase a column-oriented database that prides itself on consistency and scaling out[4].

The relationship between these pieces of software can be thought of as a stack (see Figure 2).

We installed these versions of software for this exploration:

1. Hadoop - 2.7.2
2. HBase - 1.1.0
3. Thrift - 0.9.1
4. Python - 2.7.12
5. Java - openjdk 1.8.0_91

Hadoop, Hbase, and Thrift are Java applications, and as such are subject to the limitations imposed by the Java Virtual Machine (JVM). These include the number of JVMs that can be running simultaneously within the RAM installed on the host motherboard. If the collective memory JVM requirements exceed the available RAM, then the host operating system will start using swap space on the drive and slow performance considerably.

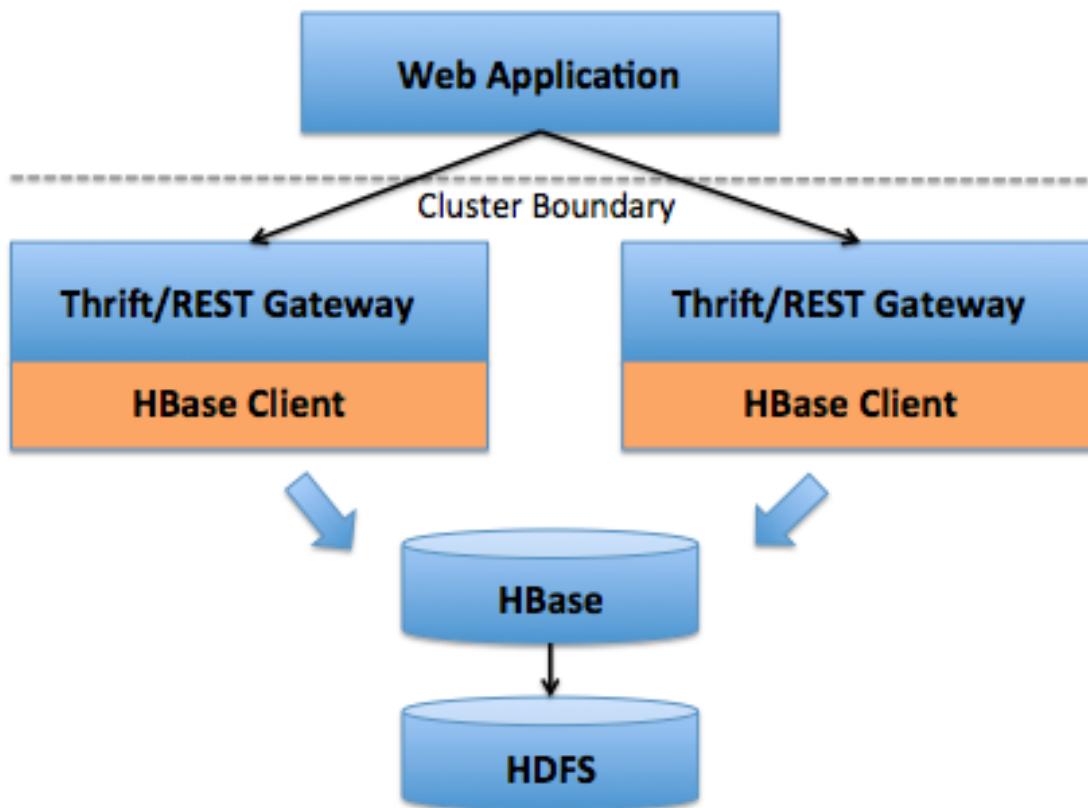


Figure 2: HBase ecosystem. Image from [1].

4 Results

We present the results of our explorations into the IMDb and NDBC.

4.1 IMDb explorations

Explorations of the IMDb depends on three data files that can be downloaded from the IMDb (see Figure 3)²³. Raw IMDb data files are available by following links in the IMDb Alternate Interfaces page (see Figures 4 and 5). The files we are interested in are:

1. `movies.list.gz` - a formatted list of all movies, games, and television shows (see Figure 6). Each line (between the header and trailer) is one video entry. Television and video games have leading characters to distinguish the type of entry. The most reliable way to identify movies, is to compare the last field with the next to last field. The last field is a year, and if the next to last is the same year only with parentheses around it, then the entry is a year. Otherwise it something else and we don't care about it.
2. `crazy-credits.list.gz` - a loosely formatted listing of crazy credits in movies, games, and television shows (see Figure 7). An entry starts with a line with a hash mark (#) as the first character, and continues until an empty line. A movie entry has a year bracketed by parentheses as the last field. Each crazy credit has a hyphen as the first character in a line, and a credit may span more than one line.
3. `directors.list.gz` - formatted list of directors for movies, games, and television shows (see Figure 8). The director's file is more complex than the other IMDb files we process. A director's efforts are bracketed by blank lines. The first line contains the director's name, tab separated from the first effort. Each effort after that is offset from the left by some number of tabs to make the printed output look nice. Each credit field that has a year bracketed by parentheses is a movie.

Algorithmically this is how we processed and explored the IMDb database:

1. Created an empty table with two column families: *credit*, and *director*. We were able to create the column families at the beginning because we knew in advance the columns we were interested in. We could just have easily created the column when we added the first credit or director entry.
2. Scanned the `movies.list.gz` file for all movies and added them as row keys.

²<http://www.imdb.com/>

³<ftp://ftp.fu-berlin.de/pub/misc/movies/database/>

3. Scanned the `directors.list.gz` file for movie directors and when found, updated count of directors for that movie. If a movie was found in the director's file that was not in the movies file, then added the director entry would automatically create a new entry in the database.
4. Scanned the `crazy-credits.list.gz` file for movie credits (similar in concept as the processing for the director's file).

At the end of this processing, the database has a "table" indexed by movie name where each "row" may, or may not have an entry for the number of credits an directors.

5. Create a histogram of the number of crazy credits per movie (see Figure 9). Of the 675,271 movies extracted from the IMDb data files, there were 12,844 that had some sort of crazy credit.
6. Create a histogram of the number of directors per movie (see Figure 10). Of the 675,271 movies extracted from the IMDb data files, there were 714,789 directors. Close examination of the histogram shows that there were a few movies that did not have any directors listed, and a relative handful (less than 40,000) that had more than one director. While that number may seem high, it is only about 6% of all movies had more than one director.

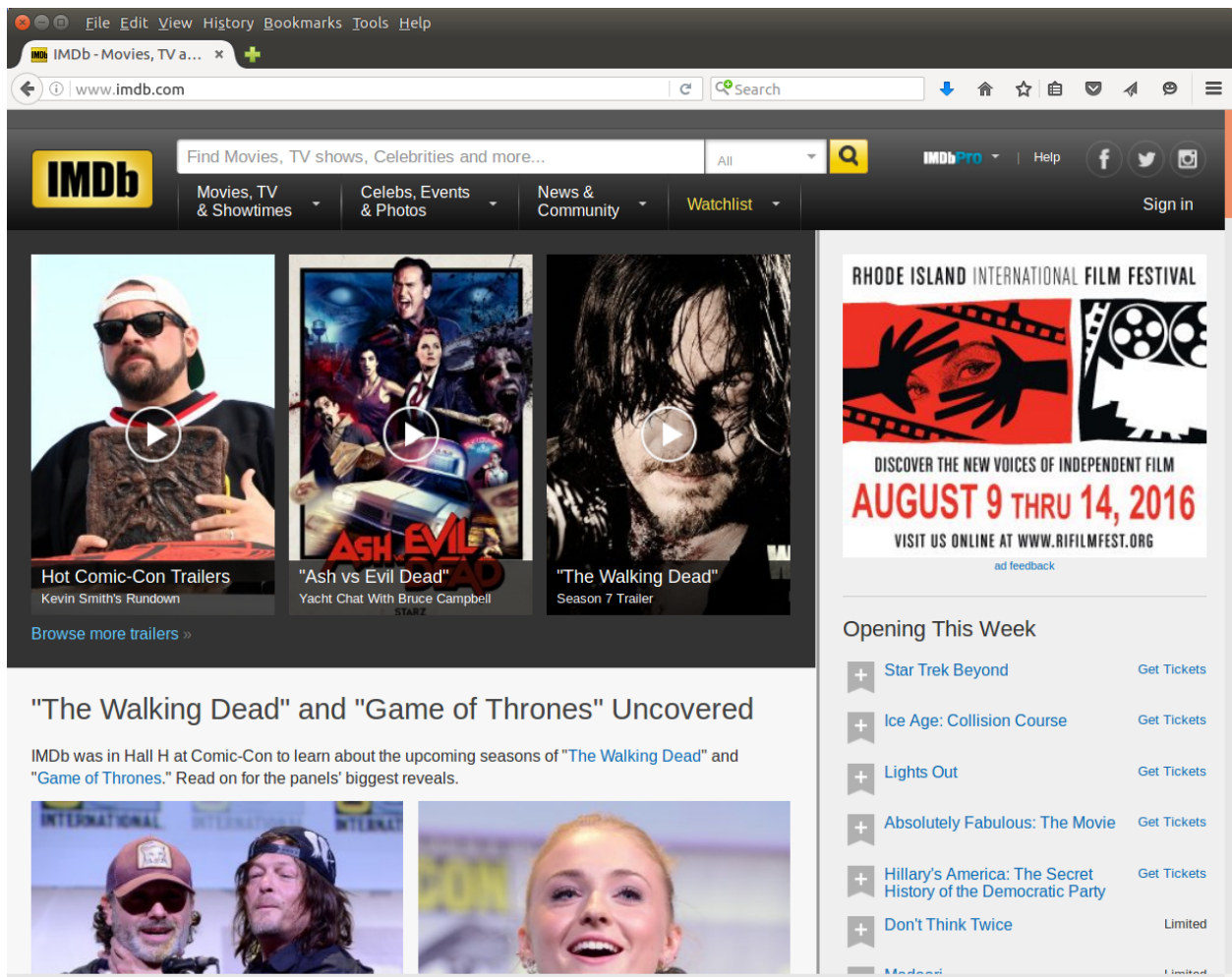


Figure 3: Internet Movie Database home page. This image was captured 23 July 2016 from <http://www.imdb.com/>. Files can be downloaded following links in the IMDb alternate interfaces page.

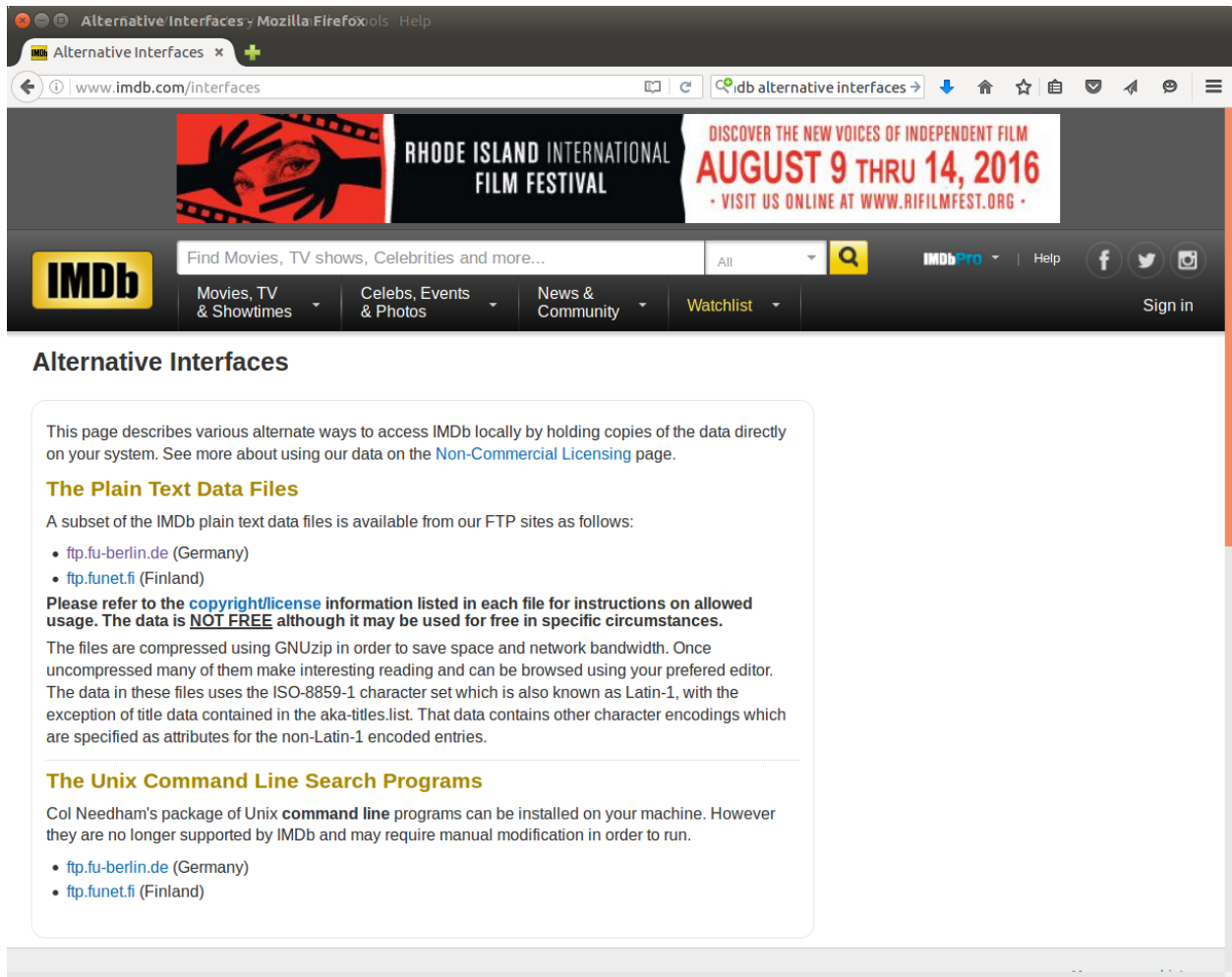


Figure 4: IMDb Alternate Interfaces home page. Raw IMDb files can be found by following links on this page:<http://www.imdb.com/interfaces>

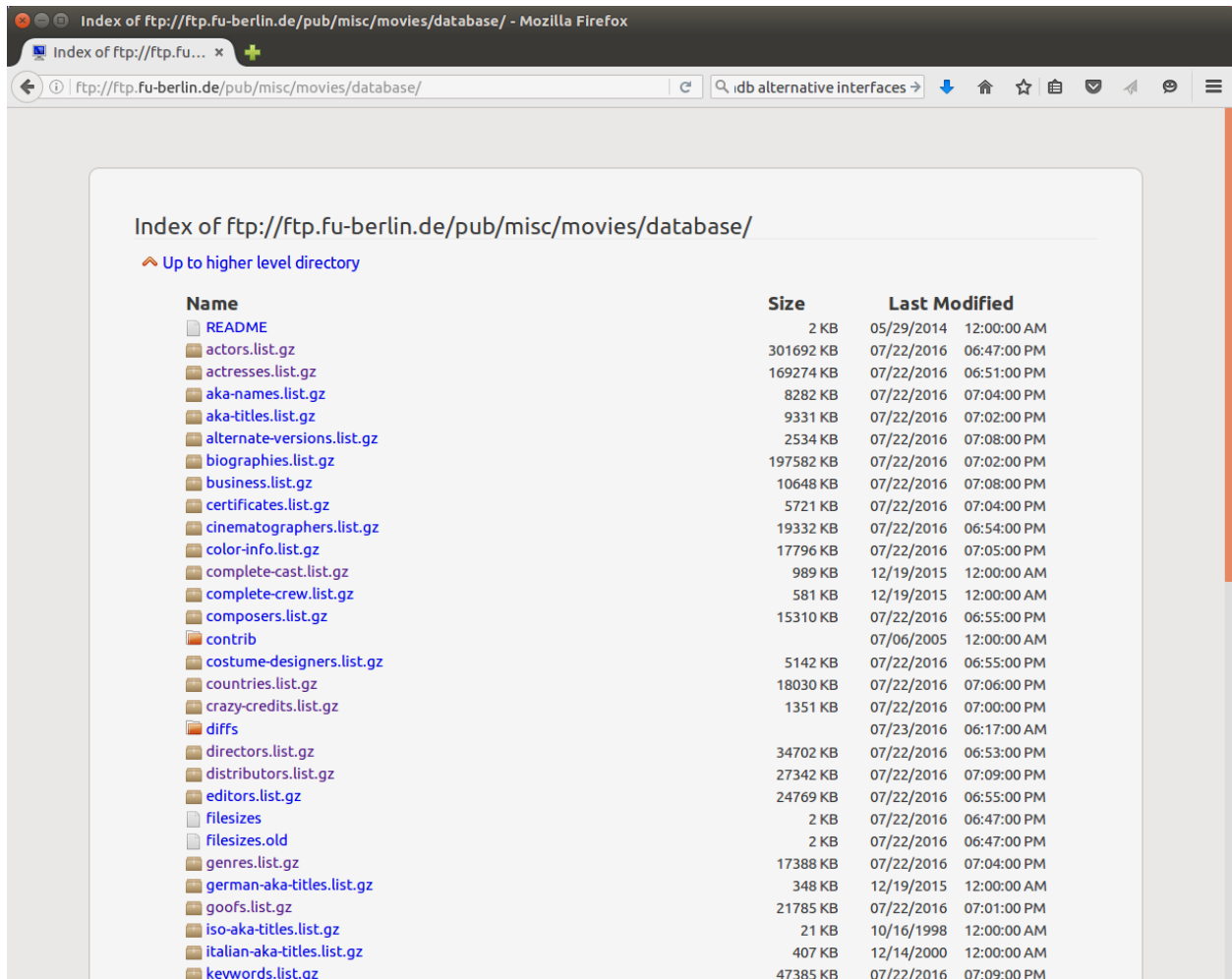


Figure 5: IMDb Alternate Interfaces FTP home page. Raw IMDb files can be downloaded from this page: <ftp://ftp.fu-berlin.de/pub/misc/movies/database/>

MOVIES LIST

=====

```

"!Next?" (1994) ██████████ 1994-1995
"#1 Single" (2006) ██████████ 2006-????
"#1 Single" (2006) {Cats and Dogs (#1.4)} ██████████ 2006
"#1 Single" (2006) {Finishing a Chapter (#1.5)} ██████████ 2006
"#1 Single" (2006) {Is the Grass Greener? (#1.1)} ██████████ 2006
"#1 Single" (2006) {Stay (#1.8)} ██████████ 2006
"#1 Single" (2006) {The Rules of Dating (#1.3)} ██████████ 2006
"#1 Single" (2006) {Timing Is Everything (#1.7)} ██████████ 2006
"#1 Single" (2006) {Window Shopping (#1.2)} ██████████ 2006
"#1 Single" (2006) {Wingman (#1.6)} ██████████ 2006
"#1MinuteNightmare" (2014) ██████████ 2014-????
"#1MinuteNightmare" (2014) {My Sweet Valentine (#1.5)} ██████████ 2014-????
"#30Nods" (2014) ██████████ 2014-2015
"#7DaysLater" (2013) ██████████ 2013-????
"#7DaysLater" (2013) {Apocalypse (#1.2)} ██████████ 2013
"#7DaysLater" (2013) {Cowboys (#1.5)} ██████████ 2013
"#7DaysLater" (2013) {Drama Queen (#1.1)} ██████████ 2013
"#7DaysLater" (2013) {Haunted House (#1.6)} ██████████ 2013
"#7DaysLater" (2013) {Portrait (#1.4)} ██████████ 2013
"#7DaysLater" (2013) {Zombies (#1.3)} ██████████ 2013
"#Adulthood" (????) ██████████ 2014-????
"#ATown" (2014) ██████████ 2014-????
"#ATown" (2014) {Best Friends Day (#1.10)} ██████████ 2014
"#ATown" (2014) {Chicks in Pink, Vomit in a Sink (#1.6)} ██████████ 2014
"#ATown" (2014) {Dunzo (#1.9)} ██████████ 2014
"#ATown" (2014) {IMPROVments (#1.4)} ██████████ 2014
"#ATown" (2014) {Jobs & Juice (#1.2)} ██████████ 2014
"#ATown" (2014) {Kayaking Adventure (#1.5)} ██████████ 2014
"#ATown" (2014) {Pilot (#1.1)} ██████████ 2014
"#ATown" (2014) {So Fucked Up (#1.7)} ██████████ 2014
"#ATown" (2014) {The Breakup Party (#1.8)} ██████████ 2014
"#ATown" (2014) {The Greenbelt (#1.3)} ██████████ 2014
"#AwkwardMornings" (2014) ██████████ 2014-????
"#AwkwardMornings" (2014) {Best Friends (#1.4)} ██████████ 2014
"#AwkwardMornings" (2014) {Boyfriend (#1.2)} ██████████ 2014

```

Figure 6: A partial dump of the IMDb movies listing. The colored areas represent some number of tabs in the line. The number of tabs varies from line to line and are used to ensure that the year field displays neatly on the “paper.”

CRAZY CREDITS

=====

```
# "'Allo 'Allo!" (1982) {Prisoners of War (#4.1)}
- Opening credits prologue: STALAG LUFT IV NORMANDY, FRANCE 1941

# "'Orrible" (2001)
- Episode 1.4 ("May the Best Man Win") uses the Buzzcocks' "Ever Fallen In
  Love" as its end theme.
- Episode 1.8 ("New Best Friend") features Johnny Vaughan and Ricky Grover
  singing "Up Where We Belong" as its end theme.

# "1 ret og 2 vrang" (1969) {(1970-03-16)}
- This episode was broadcast with the end credit roll missing.

# "12 oz. Mouse" (2005) {Auraphull (#2.6)}
- This episode was written by "No One."

# "2 Stupid Dogs" (1993) {A Quarter/Egg/Red (#1.6)}
- John Kricfalusi was credited for supplying "Tidbits of Poor Taste."

# "2 Stupid Dogs" (1993) {Family Values/Platypus/Red Strikes Back (#1.10)}
- John Kricfalusi was credited for supplying "Tidbits of Poor Taste."

# "2 Stupid Dogs" (1993) {Stunt Dogs/Doctor O/Return of Red (#1.11)}
- John Kricfalusi was credited for supplying "Tidbits of Poor Taste."

# "21 Jump Street" (1987) {Back from the Future (#4.15)}
- At the end of the episode, instead of the normal credits showing various
  high schools, we are treated to outtakes from over the series.

# "21 Jump Street" (1987) {Don't Stretch the Rainbow (#2.7)}
- qv##nm0455052##'s "I Have a Dream..." speech is played over the closing credits in pla

# "21 Jump Street" (1987) {Gotta Finish the Riff (#1.6)}
- This is the only time in the show's history where the episode title is shown prior to

# "24" (2001)
- Each episode of the show opens with a title screen and Kiefer Sutherland's
  voice-over saying "The following takes place between (hour) and (hour)"
```

Figure 7: A partial dump of the IMDb crazy credits listing.

KEY:

"xxxxx" = a television series
"xxxxx" (mini) = a television mini-series
(TV) = TV movie, or made for cable movie
(V) = made for video movie (this category does NOT include TV
episodes repackaged for video, guest appearances in
variety/comedy specials released on video, or
self-help/physical fitness videos)
(VG) = video game

THE DIRECTORS LIST
=====

Name	Titles
Özkul, Ahmet Salih	Ii (2013)
'Abd Al-Hamid, Ja'far	A Two Hour Delay (2001) Badgeless sur la Croisette (2012) Just Outside the Frame: The Profilmic Event and Beyond (2008) Mesocafe (2009) {{SUSPENDED}} Mesocafé (2011)
'D.J'Arlia, Domenic	She'll Never Know (2012)
'Dada' Pecori, Diego	Adam (????) (attached) Cantarella (2011) Makhno Beer (2010)
'Kid Niagara' Kallet, Harry	Drug Demon Romance (2012) (co-director)
'Kusare, Mak (I)	Baby Beautiful (2013/II) Comrade (2008)
'Kusare, Mak (II)	A Play Called a Temple Made of Clay (2014)
'Legend' Spivey, Larry	The Crime City Diaries: Entry 1 - Crooked (2012)
'Noble Julz'Hamilton, Ulia	Church Hurt (2015)
's Gravesande, Ad	"Het gat van Nederland" (1972)
's-Gravesande, Pjotr	Editors: The Back Room (Festival Edition) (2005) (V) Go Back to the Zoo: Live at Paradiso 2011 (2011) (V) Lucie Silvas: Live in Amsterdam (2007) (V) Milow: Maybe Next Year - Live in Amsterdam (2009) (V) (uncredite sd)

Figure 8: A partial dump of the IMDb director listing. The colored areas are tab characters.

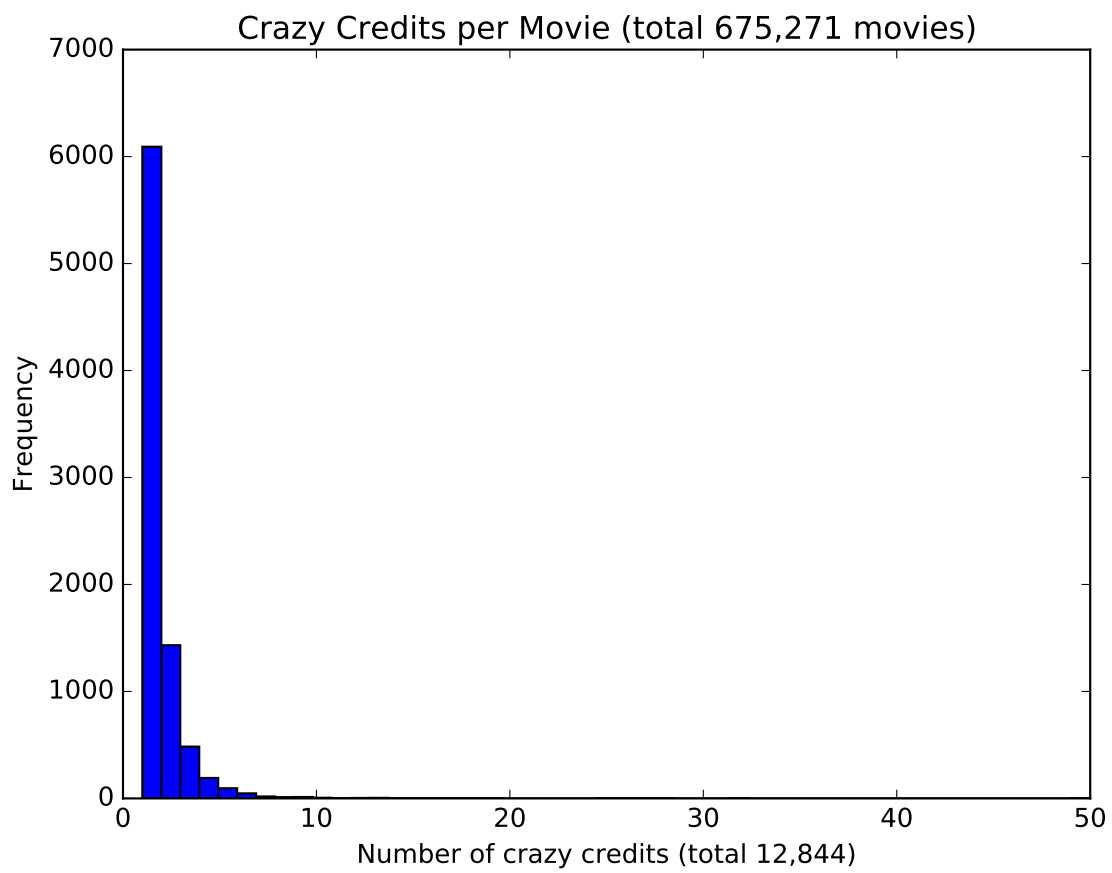


Figure 9: Histogram of crazy credits per movies.

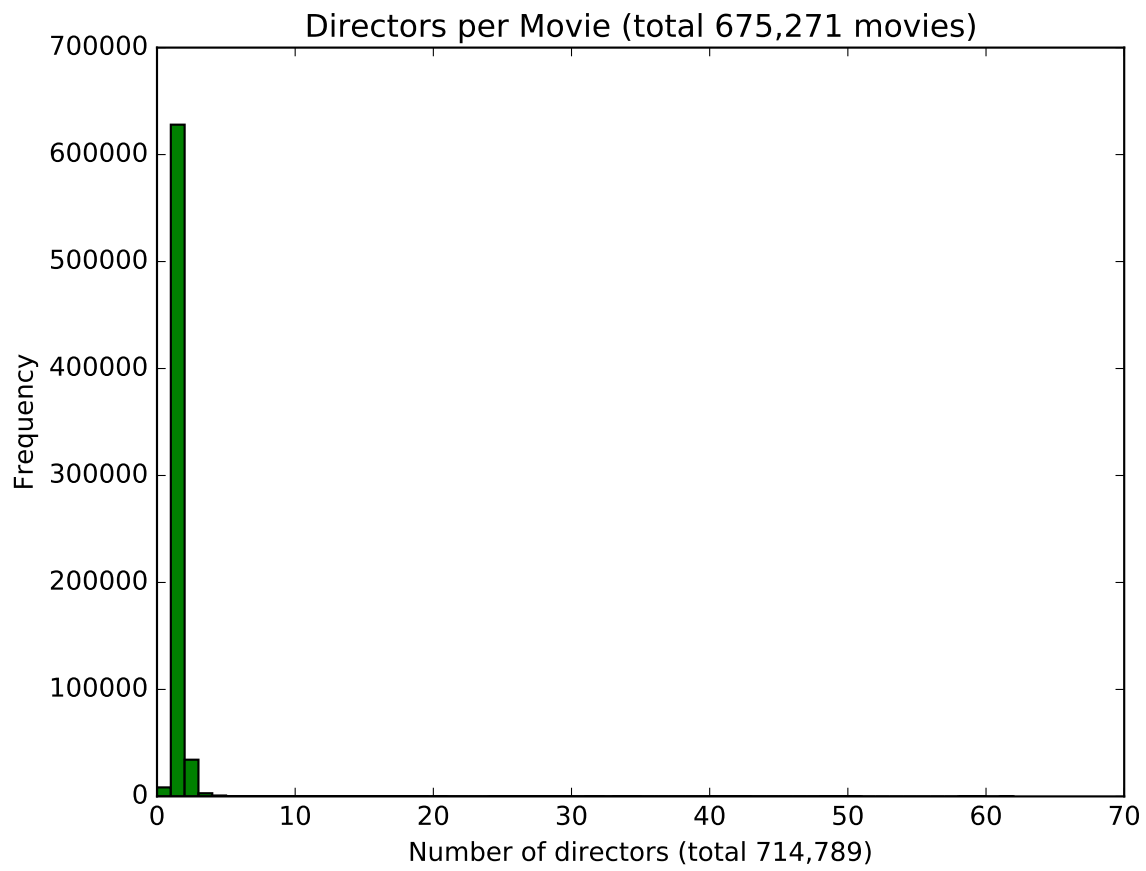


Figure 10: Histogram of directors per movie.

4.2 NDBC explorations

We explored the NDBC (see Figure 11) buoy database looking to combine a collection of technologies with interesting data. For this exploration, we are looking at real-time weather data from the buoys that the NDBC monitors (see Figures 12 and 13). The technologies that we combined were:

1. Web scraping - where we download a raw HTML page, extract parts of it that interest us, and use those parts as input to the rest of the system.
2. HBase autoversioning - where we configure the HBase database to manage autoversioning data automatically for use. This will allow us to update the database with new data without having to worry about managing the old data.
3. Python plotting capabilities - where we report the status of the latest data from the buoys on a geographic plot, and look at the weather data reported by selected buoys as a function of time.

Algorithmically, this is how we explored the NDBC database:

1. We decided if we were going to query the NDBC for live data, or to download all the interesting data in mass to local storage. The difference between the two approaches is the time to access new data files. There are numerous programs and applications that are optimized for downloading lots of files in a fast and efficient manner. Python is not one of these programs. Getting live data from the NDBC is fraught with all the normal problems associated with accessing data from the Internet, and can be a challenge.
2. Once the source of new data (local, or NDBC) has been identified, then meteorological data for each of the stations is parsed from the station's associated data file (see Figure 14), and put into the database. Each data file can have up to 36 line entries (1 per hour), to that a short amount of historical data is available should the user desire the data. The database is configured to handle 40 versions (also known as updates) automatically. By default the user is always given the latest data. Previous versions are available by requesting them.
3. All stations in the database have their position (latitude and longitude) updated from a static file available from the NDBC (see Figure 15). There are stations in the database that are not location database, and stations in the location database that are not reporting.
4. The location of each station is color coded onto a geographic display, at three different resolutions (see Figures 16 through 18). A station whose data is less than 1 hour old (station data is reported hourly) is colored green. Data that is more than one hour old is considered stale, and colored red.

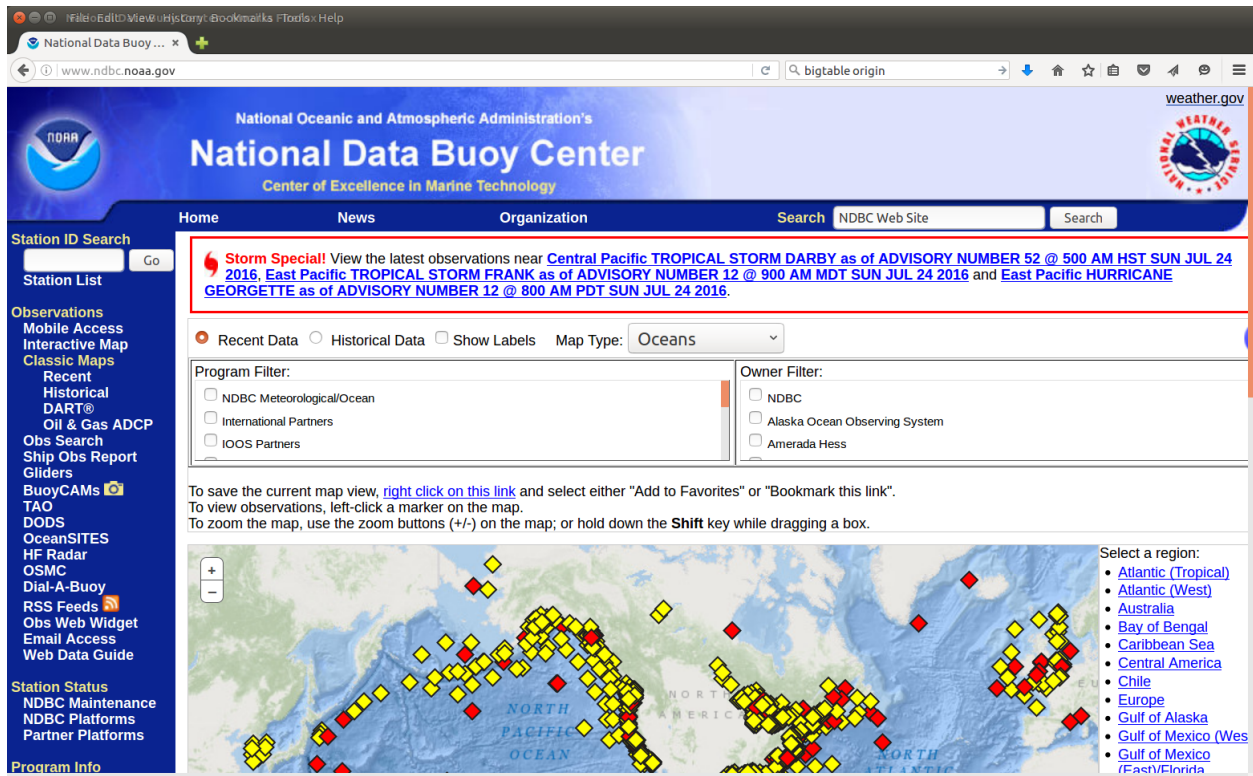


Figure 11: National Data Buoy Center (NDBC) home page. From <http://www.ndbc.noaa.gov/>

5. A selected station's wind data is plotted on polar plot to show how the data changes over time (see Figure 19).

Geographic plotting of the stations was not as straight forward as it should have been (see Figure 20). Apparently you have to explicitly close each plot, even though you are creating a new basemap each time.

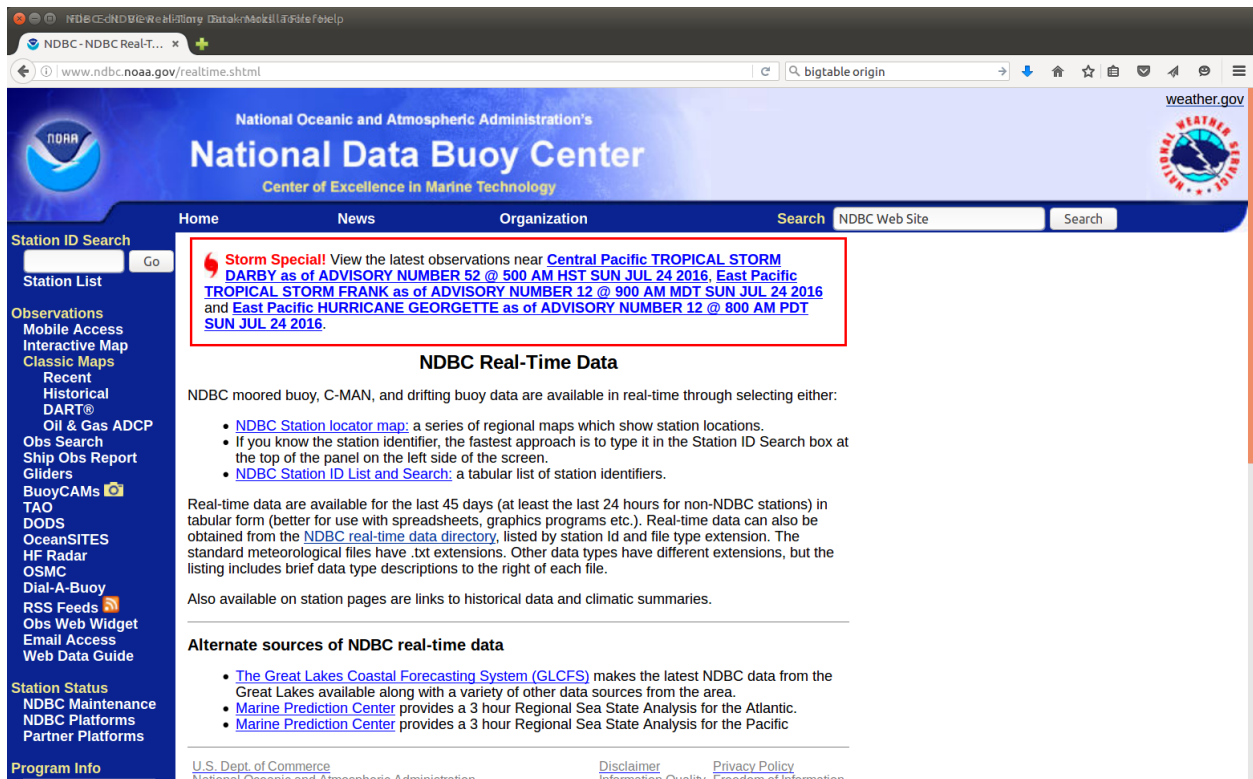


Figure 12: NDBC real time data home page. From <http://www.ndbc.noaa.gov/realtime.shtml>

Name	Last modified	Size	Description
Parent Directory		-	
32ST0.ocean	24-Jul-2016 10:23	86K	Oceanographic Data
32ST0.srad	24-Jul-2016 10:23	41K	Solar Radiation Data
32ST0.txt	24-Jul-2016 10:21	100K	Standard Meteorological Data
41NT0.ocean	24-Jul-2016 10:23	86K	Oceanographic Data
41NT0.srad	24-Jul-2016 10:23	41K	Solar Radiation Data
41NT0.txt	24-Jul-2016 10:21	100K	Standard Meteorological Data
420TP.adcp	31-Dec-2003 09:28	56K	Acoustic Doppler Current Profiler Data
420TP.cwind	28-Aug-2005 14:22	61K	Continuous Winds Data
420TP.data_spec	28-Aug-2005 14:01	180K	Raw Spectral Wave Data
420TP.hkp	28-Aug-2005 14:01	18K	Housekeeping Data
420TP.ocean	03-Mar-2004 10:02	73K	Oceanographic Data
420TP.spec	30-Jun-2004 15:09	5.9K	Spectral Wave Summary Data
420TP.swdir	31-Dec-2003 09:26	698	Spectral Wave Data (alpha1)
420TP.swdir2	31-Dec-2003 09:27	695	Spectral Wave Data (alpha2)
420TP.swr1	31-Dec-2003 09:27	697	Spectral Wave Data (r1)
420TP.swr2	31-Dec-2003 09:28	697	Spectral Wave Data (r2)
420TP.txt	29-Apr-2009 12:54	24K	Standard Meteorological Data
42539.data_spec	24-Jul-2016 11:08	715K	Raw Spectral Wave Data

Figure 13: NDBC real time data download page. From <http://www.ndbc.noaa.gov/data/realtime2/>

#YY	MM	DD	hh	mm	WDIR	WSPD	GST	WVHT	DPD	APD	MWD	PRES	ATMP	WTMP	DEWP	VIS	PTDY	TIDE
#yr	mo	dy	hr	mn	degT	m/s	m/s	m	sec	sec	degT	hPa	degC	degC	degC	nmi	hPa	ft
2016	07	22	17	30	130	1.0	2.0	MM	MM	MM	MM	1019.5	28.6	29.9	21.7	MM	MM	MM
2016	07	22	17	20	120	1.0	2.0	MM	MM	MM	MM	1019.6	28.7	29.9	22.0	MM	MM	MM
2016	07	22	17	10	80	1.0	2.0	MM	MM	MM	MM	1019.8	28.5	29.9	21.7	MM	MM	MM
2016	07	22	17	00	100	1.0	3.0	MM	MM	MM	MM	1019.6	28.6	29.8	21.9	MM	+0.5	MM
2016	07	22	16	50	80	1.0	2.0	0.8	9	6.8	116	1019.5	28.6	29.8	21.9	MM	MM	MM
2016	07	22	16	40	90	1.0	3.0	MM	MM	MM	MM	1019.6	28.6	29.7	21.7	MM	MM	MM
2016	07	22	16	30	80	2.0	3.0	MM	MM	MM	MM	1019.5	28.4	29.7	21.4	MM	MM	MM
2016	07	22	16	20	70	2.0	3.0	MM	MM	MM	MM	1019.6	28.4	29.7	21.8	MM	MM	MM
2016	07	22	16	10	70	2.0	3.0	MM	MM	MM	MM	1019.5	28.4	29.7	21.7	MM	MM	MM
2016	07	22	16	00	50	2.0	3.0	MM	MM	MM	MM	1019.5	28.4	29.7	21.5	MM	+0.7	MM
2016	07	22	15	50	50	2.0	3.0	0.8	10	6.9	117	1019.5	28.4	29.7	21.5	MM	MM	MM
2016	07	22	15	40	40	1.0	2.0	MM	MM	MM	MM	1019.5	28.4	29.7	21.3	MM	MM	MM
2016	07	22	15	30	10	1.0	2.0	MM	MM	MM	MM	1019.8	28.6	29.6	21.0	MM	MM	MM
2016	07	22	15	20	30	1.0	2.0	MM	MM	MM	MM	1020.0	28.5	29.6	21.1	MM	MM	MM
2016	07	22	15	10	40	2.0	3.0	MM	MM	MM	MM	1019.9	28.4	29.6	21.1	MM	MM	MM
2016	07	22	15	00	40	3.0	4.0	MM	MM	MM	MM	1019.6	28.2	29.6	21.1	MM	+0.6	MM
2016	07	22	14	50	40	3.0	4.0	0.8	9	6.8	114	1019.5	28.2	29.6	21.3	MM	MM	MM
2016	07	22	14	40	40	3.0	4.0	MM	MM	MM	MM	1019.3	28.2	29.6	21.5	MM	MM	MM
2016	07	22	14	30	50	3.0	4.0	MM	MM	MM	MM	1019.4	28.2	29.6	21.8	MM	MM	MM
2016	07	22	14	20	40	3.0	4.0	MM	MM	MM	MM	1019.3	28.1	29.6	21.7	MM	MM	MM
2016	07	22	14	10	40	3.0	4.0	MM	MM	MM	MM	1019.1	28.1	29.5	21.7	MM	MM	MM

Figure 14: Sample data from station 41013. All data columns are white-space delimited. Missing data is indicated by the “MM”.

```

# STATION_ID | OWNER | TTYPE | HULL | NAME | PAYLOAD | LOCATION | TIMEZONE | FORECAST | NOTE
#
00922|DU|Slocum Glider||OTN201 - 4800922||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|E| |
00923|DU|Slocum Glider||OTN200 - 4800923||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|E| |
01500|R|Spray Glider||SP031 - 3801500||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|?| |
01502|UA|Slocum Glider||Penobscot - 4801502||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|E| |
01503|WH|Slocum Glider||Saul - 4801503||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|E| |
01504|UM|Slocum Glider||Blue - 4801504||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|E| |
01505|RU|Slocum Glider||RU28 - 4801505||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|E| |
01506|RU|Slocum Glider||RU22 - 4801506||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|E| |
01507|RU|Slocum Glider||RU23 - 4801507||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|E| |
01508|UD|Slocum Glider||OTIS - 4801508||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|E| |
01510|CS|Slocum Glider||Salacia - 4801510||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|E| |
01511|S|Slocum Glider||Modena - 4801511||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|E| |
01517|WH|Slocum Glider||WHOI_406 - 4801517||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|E| |
01518|RU|Slocum Glider||RU30 - 4801518||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|E| |
01519|UA|Slocum Glider||Unit - 4801519||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|E| |
01521|R|Spray Glider||SP011 - 4801521||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|?| |
01522|R|Spray Glider||SP018 - 4801522||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|?| |
01523|R|Spray Glider||SP025 - 4801523||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|?| |
01524|R|Spray Glider||SP028 - 4801524||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|?| |
01526|R|Spray Glider||SP048 - 4801526||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|?| |
01531|R|Spray Glider||SP407 - 4801531||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|?| |
01532|R|Spray Glider||SP020 - 4801532||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|?| |
01534|R|Spray Glider||SP030 - 4801534||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|?| |
01535|R|Spray Glider||SP052 - 4801535||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|?| |
01536|R|Spray Glider||SP063 - 4801536||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|?| |
01537|RU|Slocum Glider||ru07 - 4801537||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)| |
01538|R|Spray Glider||SP043 - 4801538||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|?| |
01909|R|Spray Glider||SCRIPPS Glider - 4801909||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|?| |
01910|R|Spray Glider||SCRIPPS Glider - 4801910||30.000 N 90.000 W (30&#176;0'0" N 90&#176;0'0" W)|?| |
0y2w3|CG|Weather Station||Sturgeon Bay CG Station, WI||44.794 N 87.313 W (44&#176;47'39" N 87&#176;18'48" W)|C|
13001|PR|Atlas Buoy||PM-595|NE Extension||12.000 N 23.000 W (12&#176;0'0" N 23&#176;0'0" W)| |
13002|PR|Atlas Buoy||NE Extension||21.000 N 23.000 W (21&#176;0'0" N 23&#176;0'0" W)| |
13008|PR|Atlas Buoy||PM-531|Reggae||15.000 N 38.000 W (15&#176;0'0" N 38&#176;0'0" W)| |
13009|PR|Atlas Buoy||PM-533|Lambada||8.000 N 38.000 W (8&#176;0'0" N 38&#176;0'0" W)| |
13010|PR|Atlas Buoy||PM-590|Soul||0.000 N 0.000 E (0&#176;0'0" N 0&#176;0'0" E)| |
15001|PR|Atlas Buoy||PM-597|Gavotte||10.000 S 10.000 W (10&#176;0'0" S 10&#176;0'0" W)| |
15002|PR|Atlas Buoy||PM-591|Java||0.000 N 10.000 W (0&#176;0'0" N 10&#176;0'0" W)| |
15006|PR|Atlas Buoy||PM-593|Valse||6.000 S 10.000 W (6&#176;0'0" S 10&#176;0'0" W)| |
15007|PR|Atlas Buoy||PM-594|Fugate||10.000 S 10.000 W (10&#176;0'0" S 10&#176;0'0" W)| |

```

Figure 15: Sample station location data. Fields are delimited by the pipe symbol (—), and may be empty. The station ID is the first field. Station location is reported in two different formats in the same field. The first is degree decimal degree followed by the hemisphere indicator. The second format is suitable for HTML presentation as degree (with the degree symbol), minute, second, followed by the hemisphere indicator.

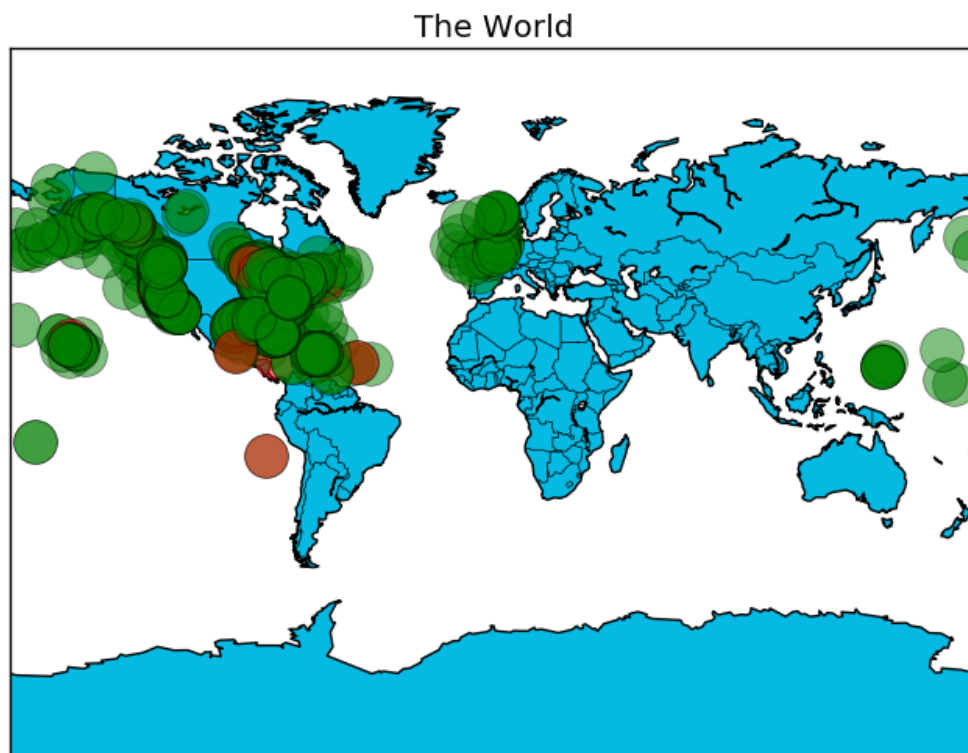


Figure 16: Buoy reporting status worldwide. A station whose data is less than 1 hour old (station data is reported hourly) is colored green. Data that is more than one hour old is considered stale, and colored red.

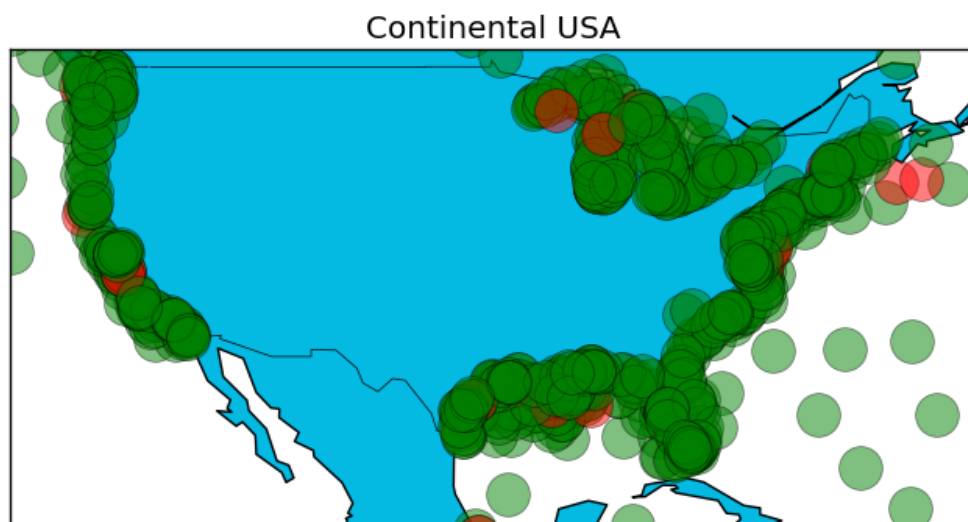


Figure 17: Buoy reporting status near continental USA. A station whose data is less than 1 hour old (station data is reported hourly) is colored green. Data that is more than one hour old is considered stale, and colored red.

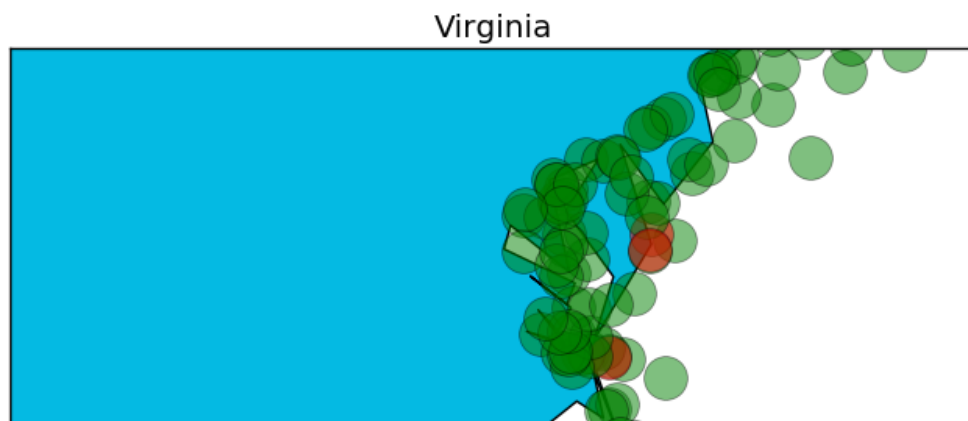


Figure 18: Buoy reporting status near Virginia. A station whose data is less than 1 hour old (station data is reported hourly) is colored green. Data that is more than one hour old is considered stale, and colored red.

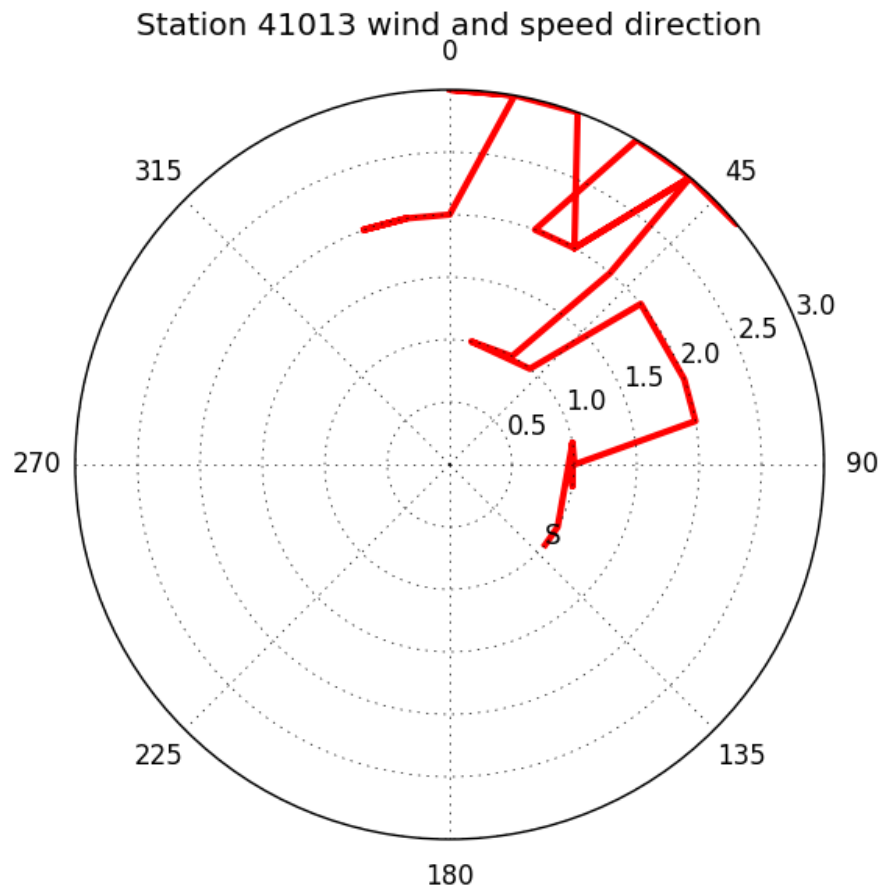


Figure 19: Station 41013 wind file. The "S" marks the oldest data.

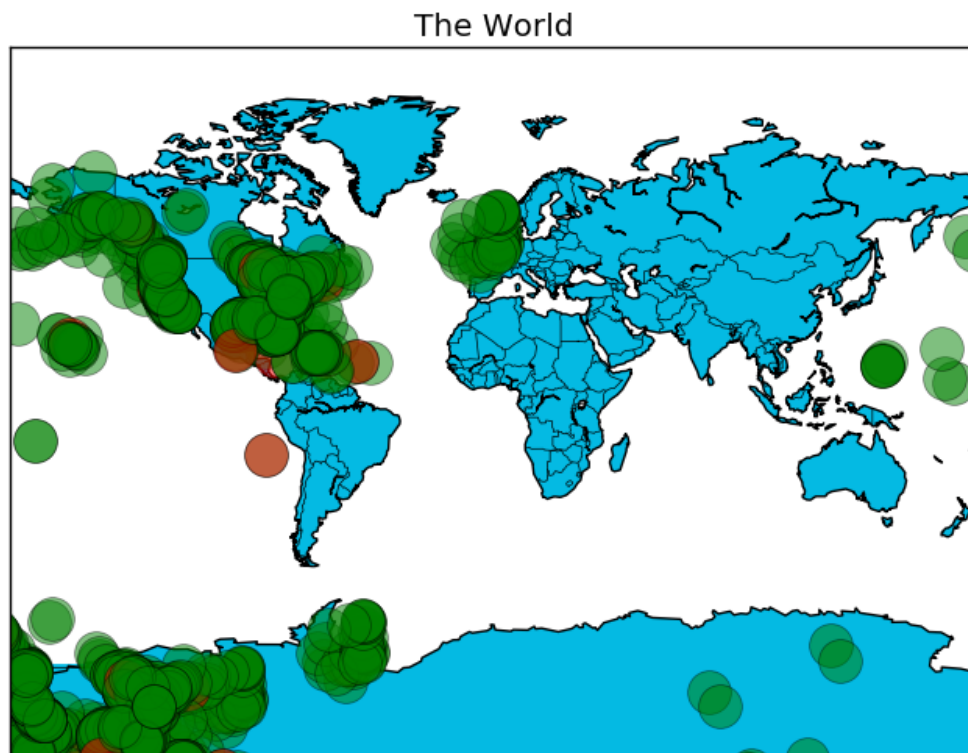


Figure 20: Buoy reporting status worldwide (bad data). The collection of stations in the lower left is wrong. It appears that the the data was “left over” from previous plots. Apparently you have to explicitly close each plot, even though you are creating a new basemap each time.

5 Conclusion

“On the performance front, HBase is meant to scale out. If you have huge amounts of data, measured in many gigabytes or terabytes, HBase may be for you. HBase is rack-aware, replicating data within and between datacenter racks so that node failures can be handled gracefully and quickly.”

Redmond and Wilson [4]

“Although HBase is designed to scale out, it doesnt scale down. . . . Additionally, HBase is almost never deployed alone. Rather, its part of an ecosystem of scale-ready pieces. These include Hadoop (an implementation of Googles MapReduce), the Hadoop distributed file system (HDFS), and Zookeeper (a headless service that aids internode coordination). This ecosystem is both a strength and a weakness; it simultaneously affords a great deal of architectural sturdiness but also encumbers the administrator with the burden of maintaining it.”

Redmond and Wilson [4]

“Any problem in computer science can be solved with one additional layer of indirection. But that usually will create another problem.” - David Wheeler

“Any program in computer science can be sped up by removing one layer of indirection.”
- Anonymous

Parsing and summarizing information from the Internet Movie Database went smoothly. The database key-value was the name of the movie, and the number of crazy credits and directors per movie were updated easily and quickly. The histograms of crazy credits versus movies, and directors versus movies, showed that of the 675,271 movies in the database there were 12,844 crazy credits. In about 50% of the time, a movie would have more than one crazy credit, if it had any. The directors histogram showed that there were holes in the database, because some of the movies did not credit a director.

Accessing, parsing, and interpreting the National Data Buoy Center (NDBC) data was more challenging. The original design and implementation was to have the program access live data on the Internet, process the data, and update the database. The data is updated in “real-time” about once an hour on the hour. The original implementation would take almost an hour to complete its processing due to Internet connection speeds, and HBase processing speed. The implementation was changed to scrape the NDBC site to identify all available files, and then download those files in parallel to local storage. Ten download processes at a time were started, reducing the download time to approximately 6 minutes. Ten processes were chosen arbitrarily, and not based on any sort of evaluation process. The HBase database was configured to handle 40 updates per row-key, so the 36 updates in each NDBC station report could be handled in their entirety and the database would ensure only the last 40 updates were available. Parsing the machine generated data files was straight




forward. Displaying the data revealed that a Python plot must be closed before new data can be displayed.

HBase is a columnar database. That is, it works well where the data has holes, where the type of data to be grouped into a common “key-value” row is unknown, and where the data and groupings may change over time. Because it is built on top of Hadoop, it should be able to scale up and out easily. The test environment was a single node Hadoop installation, so we weren’t able to test this capability.

A Selected stations



A collection of “interesting” station types gleaned from the NDBC station location file.

Table 3: A collection of interesting stations.

Name	Image	Explanation
10-meter discus buoy		Weather buoys are instruments which collect weather and ocean data within the world’s oceans, as well as aid during emergency response to chemical spills, legal proceedings, and engineering design. Moored buoys have been in use since 1951, while drifting buoys have been used since 1979. Moored buoys are connected with the ocean bottom using either chains, nylon, or buoyant polypropylene.
2.5-meter ODAS buoy		Canadian Ocean Data Acquisition System (ODAS)
Atlas Buoy		Design of the relatively inexpensive ATLAS (Autonomous Temperature Line Acquisition System) mooring was initiated by PMEL’s Engineering Development Division (EDD) in 1984. By the mid-1990’s, a reengineering effort was underway to modernize the ATLAS.


(Continued on the next page.)

Table 3. (Continued from the previous page.)

Name	Image	Explanation
Bottom Mounted ADCP		<p>An Acoustic Doppler Current Profiler, or Acoustic Doppler Profiler, is often referred to with the acronym ADCP. Scientists use the instrument to measure how fast water is moving across an entire water column. An ADCP anchored to the seafloor can measure current speed not just at the bottom, but also at equal intervals all the way up to the surface.</p>
Canadian NO-MAD buoy		<p>The AXYS NOMAD is a unique aluminum environmental monitoring buoy designed for deployments in extreme conditions. The NOMAD (Navy Oceanographic Meteorological Automatic Device) is a modified version of the 6m hull originally designed in the 1940s for the U.S. Navys offshore data collection program. It has been operating in Canadas Weather Buoy network for over 25 years and commonly experiences winter storms and hurricanes with wave heights approaching 20m.</p>



(Continued on the next page.)

Table 3. (Continued from the previous page.)

Name	Image	Explanation
Seaglider		<p>Seaglider is an autonomous underwater vehicle (AUV) or underwater glider developed for continuous, long term measurement of oceanographic parameters. Rather than an electrically driven propeller, the vehicle uses small changes in buoyancy and wings to achieve forward motion. The system's pitch and roll are controlled using adjustable ballast (the vehicle battery).</p>
Slocum Glider		<p>The Slocum Glider is a uniquely mobile network component capable of moving to specific locations and depths and occupying controlled spatial and temporal grids. Driven in a sawtooth vertical profile by variable buoyancy, the glider moves both horizontally and vertically. The long-range and duration capabilities of Slocum gliders make them ideally suited for subsurface sampling at the regional scale. Carrying a wide variety of sensors, they can be programmed to patrol for weeks at a time, surfacing to transmit their data to shore while downloading new instructions at regular intervals, realizing a substantial cost savings compared to traditional surface ships.</p>



(Continued on the next page.)

Table 3. (Continued from the previous page.)

Name	Image	Explanation
Spray Glider		<p>Spray gliders are robotic submarines that navigate underwater without a human crew onboard and without cables connecting them to research vessels at the sea surface. Spray gliders are among a class of ocean instruments known as autonomous underwater vehicles, or AUVs. These gliders carry a variety of sensors, and are programmed by researchers to go where they are needed to do research. They are used to take vertical profiles of data, giving scientists a clearer understanding of the temperature, salinity, and turbidity of specific areas of the oceans. These measurements are then used to determine and understand ocean circulation and its role and influence on the global climate</p>
STB - SAIC Tsunami Buoy		<p>The Science Applications International Corporation (SAIC) Tsunami Buoy (STB) is an enhanced version of the NOAA Deep-ocean Assessment and Reporting of Tsunami (DART) system.</p>

(Continued on the next page.)


Table 3. (Continued from the previous page.)


Name	Image	Explanation
TABS II		<p>Texas Automated Buoy System (TABS) In August, 1994, The State of Texas General Land Office (GLO) directed the Geochemical and Environmental Research Group (GERG) of Texas A&M University to implement a program that provides real-time observations of surface currents and water temperature at selected locations along the Texas coast. The Texas Automated Buoy System (TABS) became operational in April 1995.</p>
Waverider Buoy		<p>...in combination with designs for very low power electronics, resulted in the Waverider buoys.</p>


(Last page.)


B Misc. files

The files used to create all these figures are attached to this report. They are:

1. `startAll.sh`  - a bash shell script used to:
 - (a) Start the Hadoop DFS daemons, the namenode and datanodes via `start-dfs.sh`
 - (b) Start ResourceManager daemon and NodeManager daemon via `start-yarn.sh`
 - (c) Start the HBase server via `start-hbase.sh`

2. `stopAll.sh`  - a bash shell script used to:
 - (a) Stop the HBase server via `stop-hbase.sh`
 - (b) Stop ResourceManager daemon and NodeManager daemon via `stop-yarn.sh`
 - (c) Stop the Hadoop DFS daemons, the namenode and datanodes via `stop-dfs.sh`

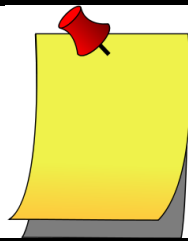
3. `buoy.py`  - a Python script that accesses the National Data Buoy Center for “real-time” data. The program:
 - (a) Downloads data reports from all buoys and sensors tracked by the data center.
 - (b) Parses the reports to get wind and temperature data.
 - (c) Plots the buoy position on various geographic displays.
 - (d) Data is persisted in an HBase database.

4. `imdb.py`  - a Python script that accesses the data from the Internet Movie Database (via downloaded zip files). The program:
 - (a) Parses selected files (list of videos, list of directors, list of crazy credits).
 - (b) Stores data in an HBase database.
 - (c) Extracts data from the database and creates histograms of interest.

Movie data comes from the Internet Movie Database (IMDb)⁴. Buoy data comes from the National Data Buoy Center⁵.

⁴<http://www.imdb.com/>

⁵<http://www.ndbc.noaa.gov/data/realtime2/>



The start and stop bash shell scripts ensure that all daemons (servers and services) are started and stopped in the correct order.

C References

- [1] Jesse Anderson, *How-to: Use the hbase thrift interface, part 1*, <http://blog.cloudera.com/blog/2013/09/how-to-use-the-hbase-thrift-interface-part-1/>, 2013.
- [2] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C Hsieh, Deborah A Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E Gruber, *Bigtable: A distributed storage system for structured data*, ACM Transactions on Computer Systems (TOCS) **26** (2008), no. 2, 4.
- [3] Nishant Garg, *Hbase essentials*, Packt Publishing Ltd, 2014.
- [4] Eric Redmond and Jim R Wilson, *Seven databases in seven weeks*, Pragmatic Bookshelf, 2012.
- [5] Apache Staff, *Hdfs architecture guide*, https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html, 2016.
- [6] Career Staff, *Introduction to rdbms*, <http://www.careerbless.com/db/rdbms/c1/rdbms.php>, 2016.
- [7] Wikipedia Staff, *Apache hadoop*, https://en.wikipedia.org/wiki/Apache_Hadoop, 2016.