CS5604, Information Retrieval, Fall 2016

# Collection Management (Tweets) Final Presentation

Faiz Abidi Mitch Wagner Shuangfei Fan

December 1, 2016
Virginia Tech @ Blacksburg, VA
Professor: Dr. Edward Fox

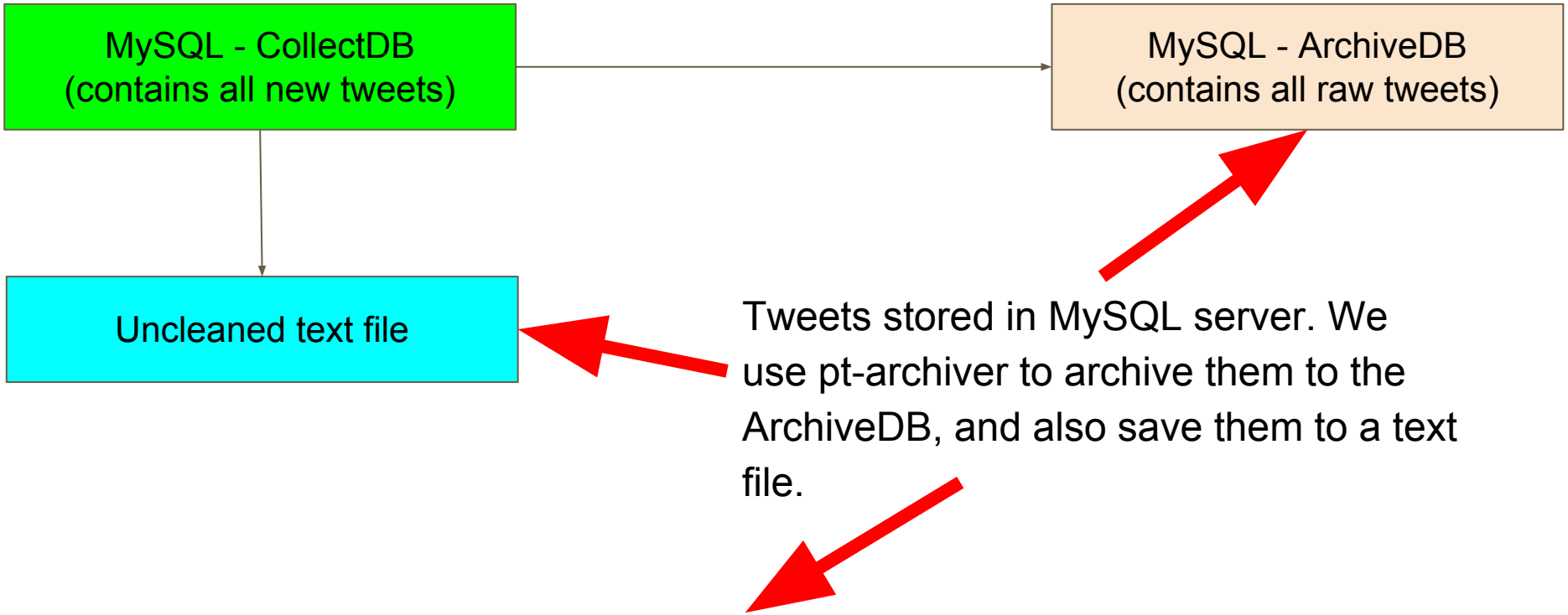
# Additions regarding tweet updates

| | | Before | Now |
|---|---|---|---|
| MySQL to HDFS | Mode of transfer | Batch mode | Incremental update |
| HDFS to HBase | | Batch mode | Incremental update |

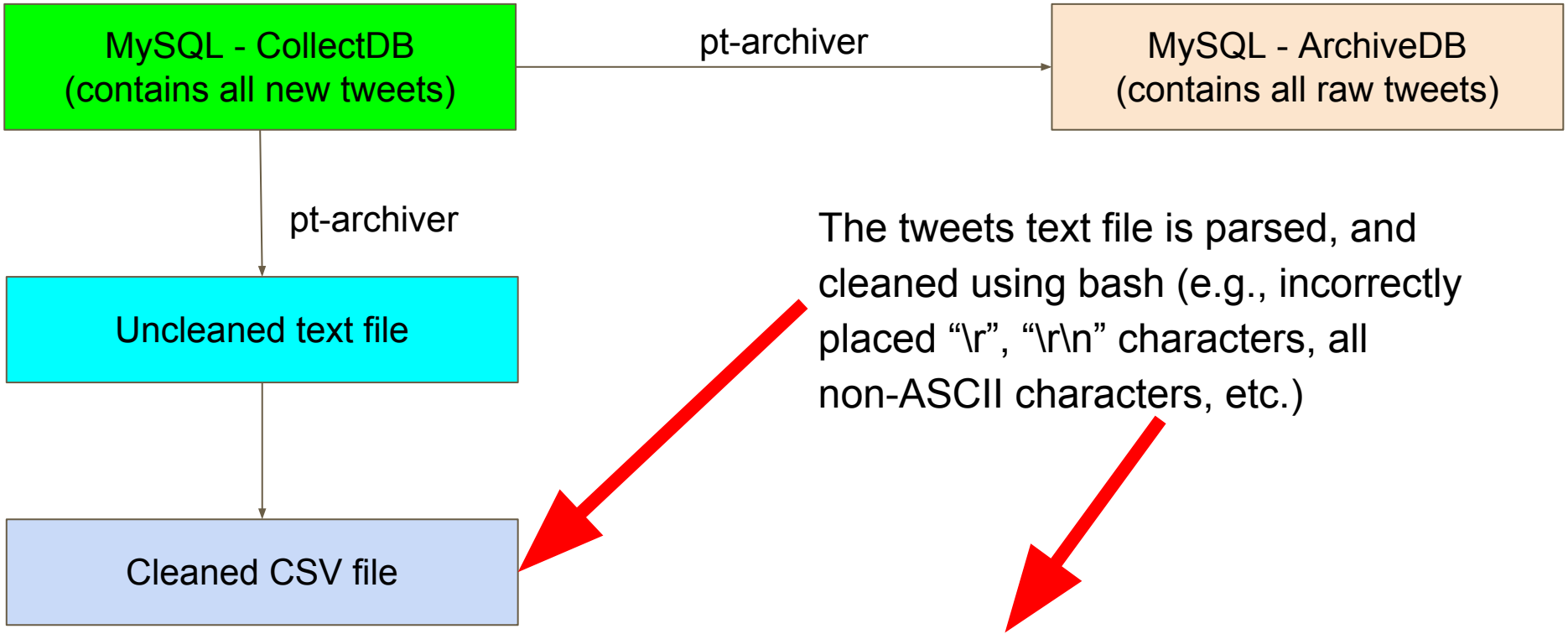
# What features did we improve?

| What was done before? | How did we improve it? |
| --- | --- |
| Limited amount of tweet parsing. | We are extracting a lot more fields now as per different teams' requirements. |
| Social network based on users as nodes, and links using mentions and re-tweets. Only one kind of node, with little emphasis on importance value. | Three kinds of nodes - users, tweets, and URLs. We are using the Twitter API to calculate an importance value for the users and the tweets, and taking the number of occurrences of a URL in a tweet collection as an indication of its importance within that collection. |

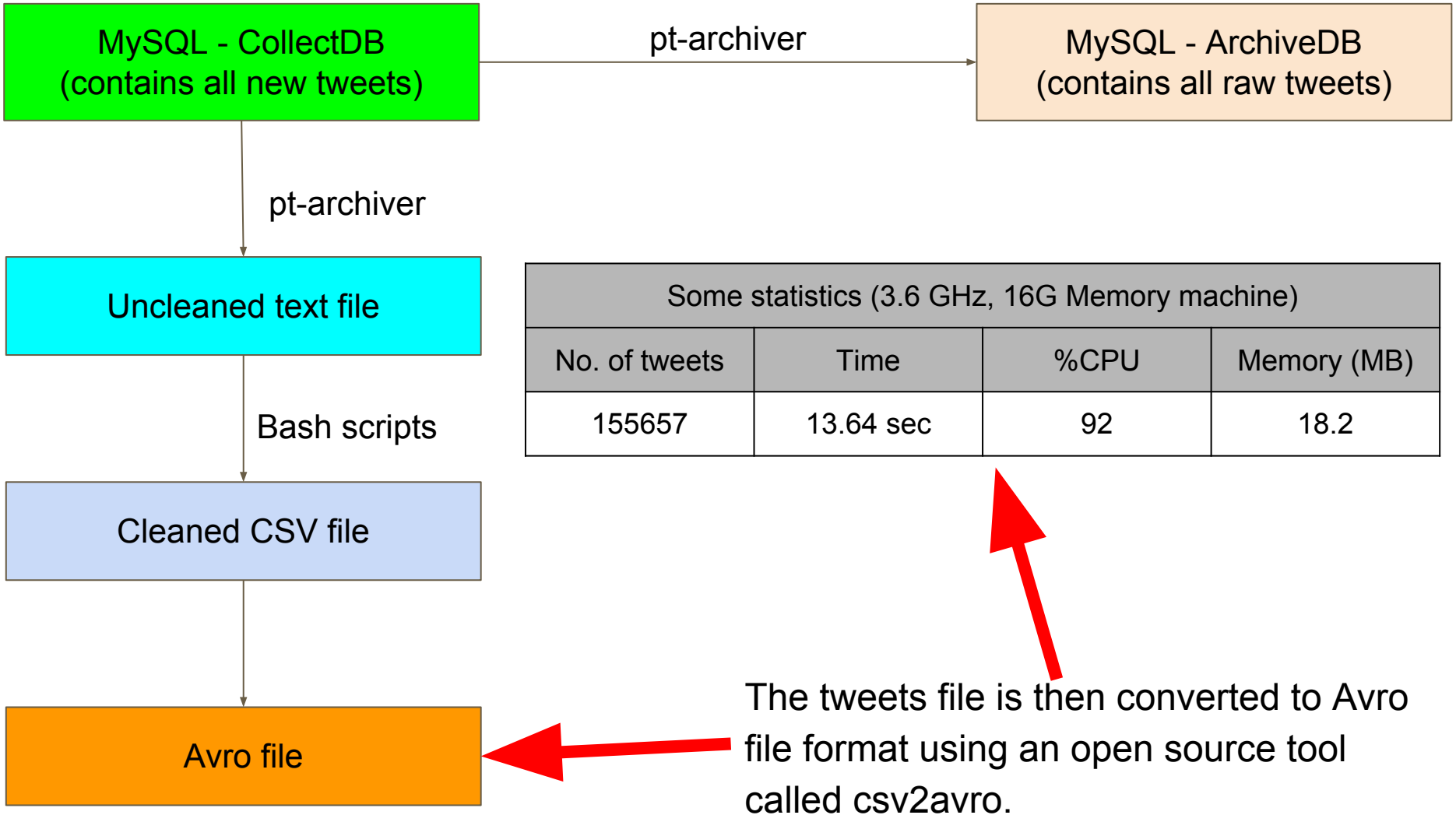
# Incremental Update From MySQL to HDFS



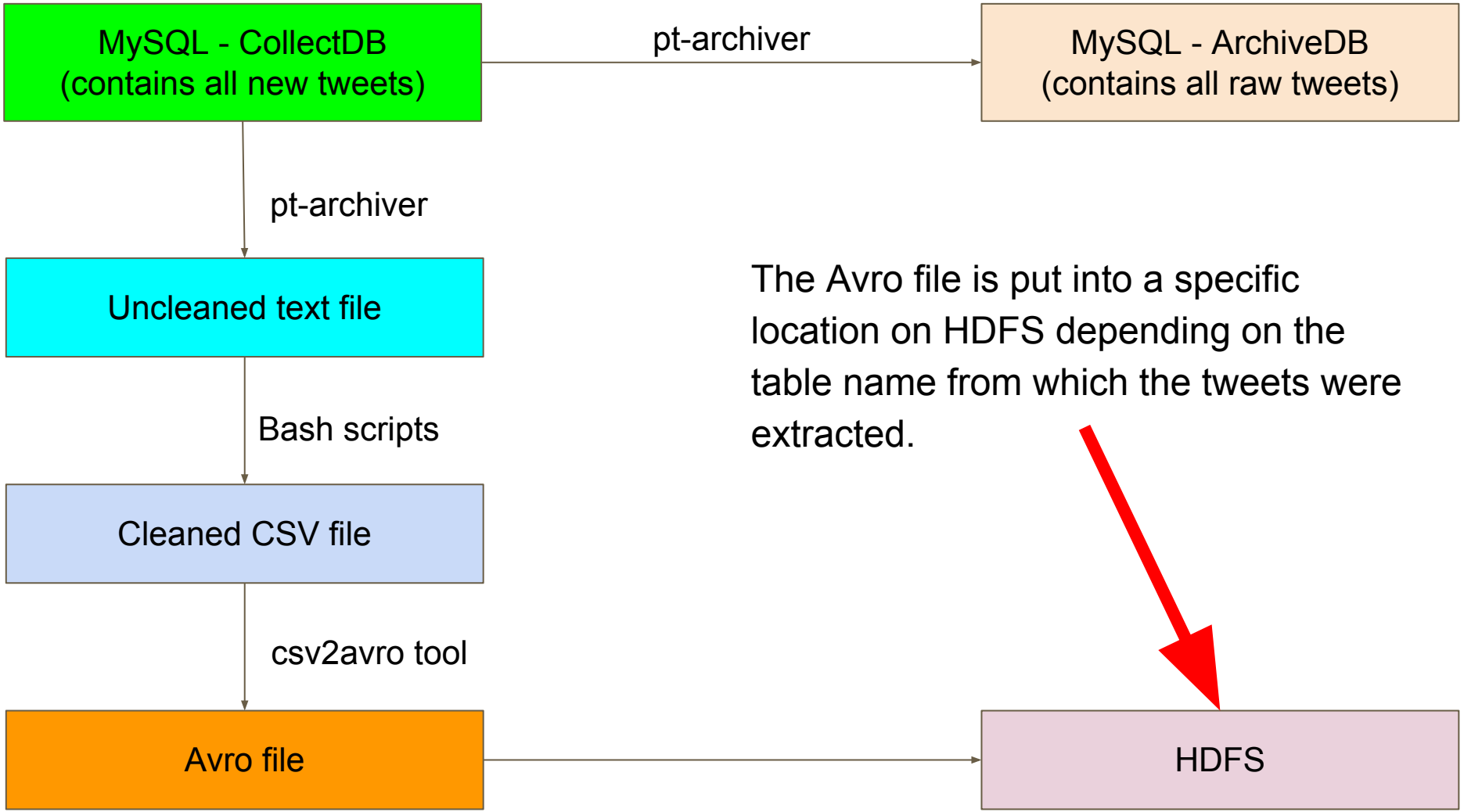

| Some statistics (3.6 GHz, 16G Memory machine) | | | |
|---|---|---|---|
| No. of tweets | Time | %CPU | Memory (MB) |
| 155657 | 1 min 35 sec | 29 | 19.7 |

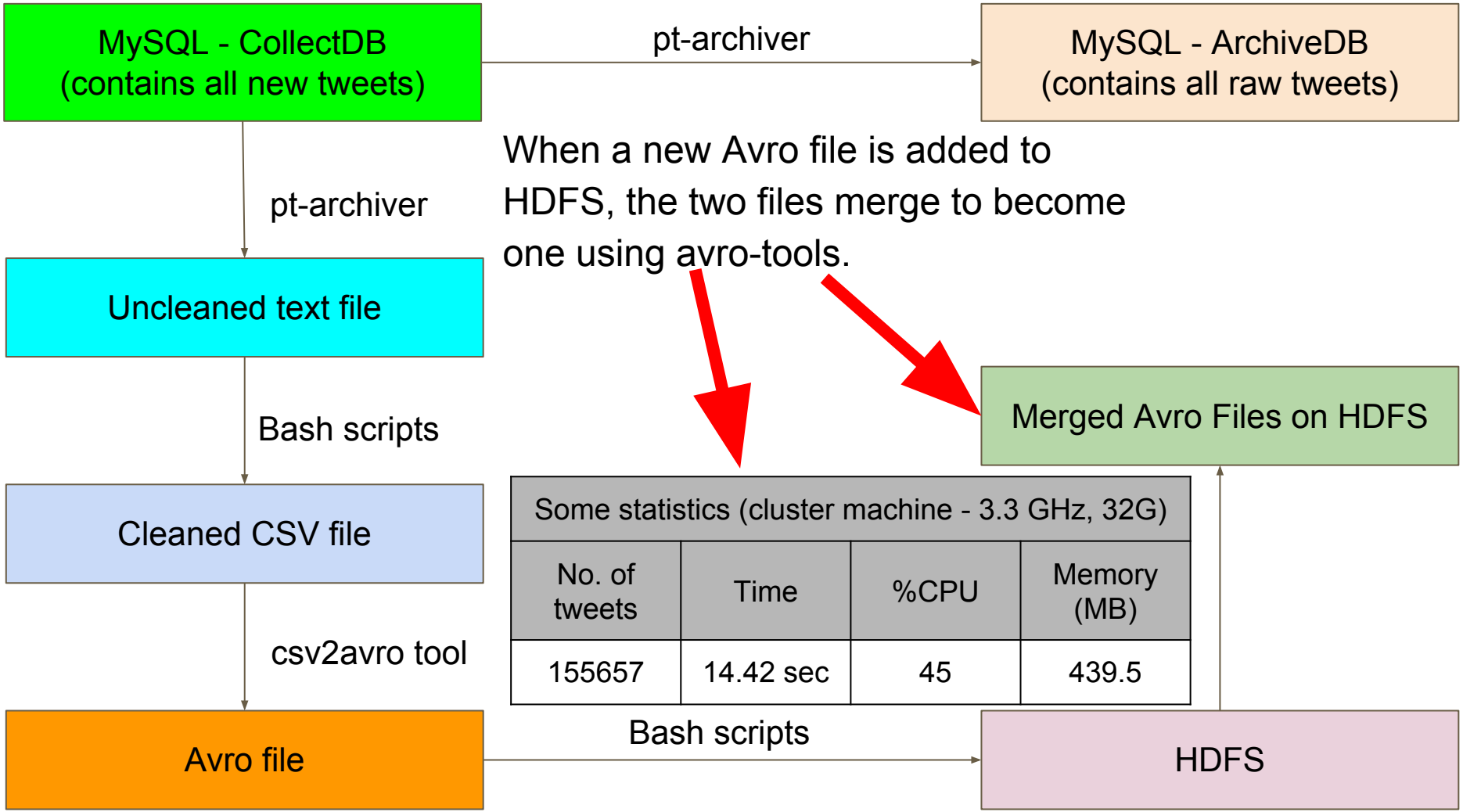



| Some statistics (3.6 GHz, 16G Memory machine) | | | |
|---|---|---|---|
| No. of tweets | Time | %CPU | Memory (MB) |
| 155657 | 7.89 sec | 57 | 169.9 |

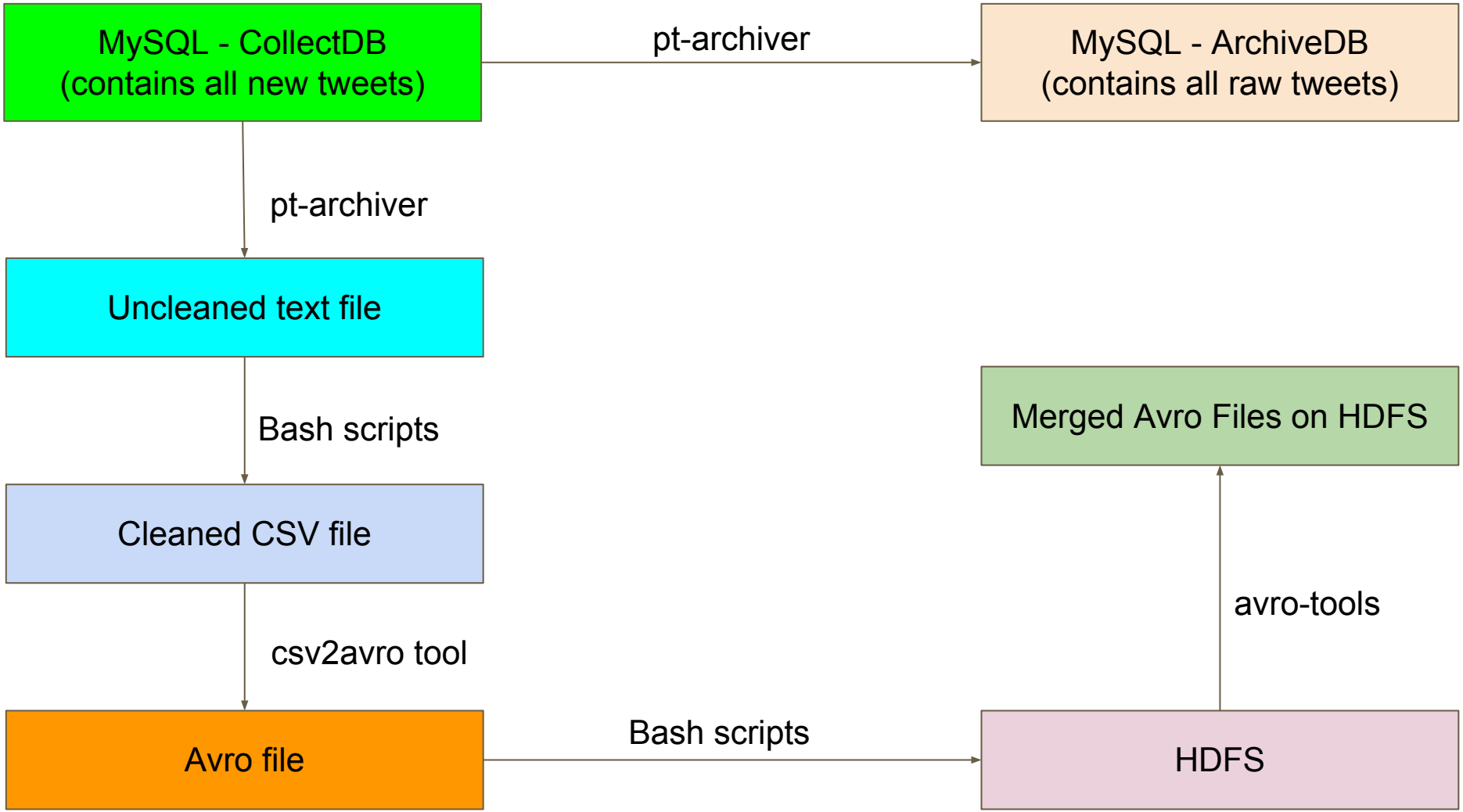



| Some statistics (3.6 GHz, 16G Memory machine) | | | |
|---|---|---|---|
| No. of tweets | Time | %CPU | Memory (MB) |
| 155657 | 13.64 sec | 92 | 18.2 |


MySQL - CollectDB
(contains all new tweets)
pt-archiver
MySQL - ArchiveDB
(contains all raw tweets)
pt-archiver
Uncleaned text file
Bash scripts
Cleaned CSV file
csv2avro tool
Avro file
HDFS
The Avro file is put into a specific location on HDFS depending on the table name from which the tweets were extracted.

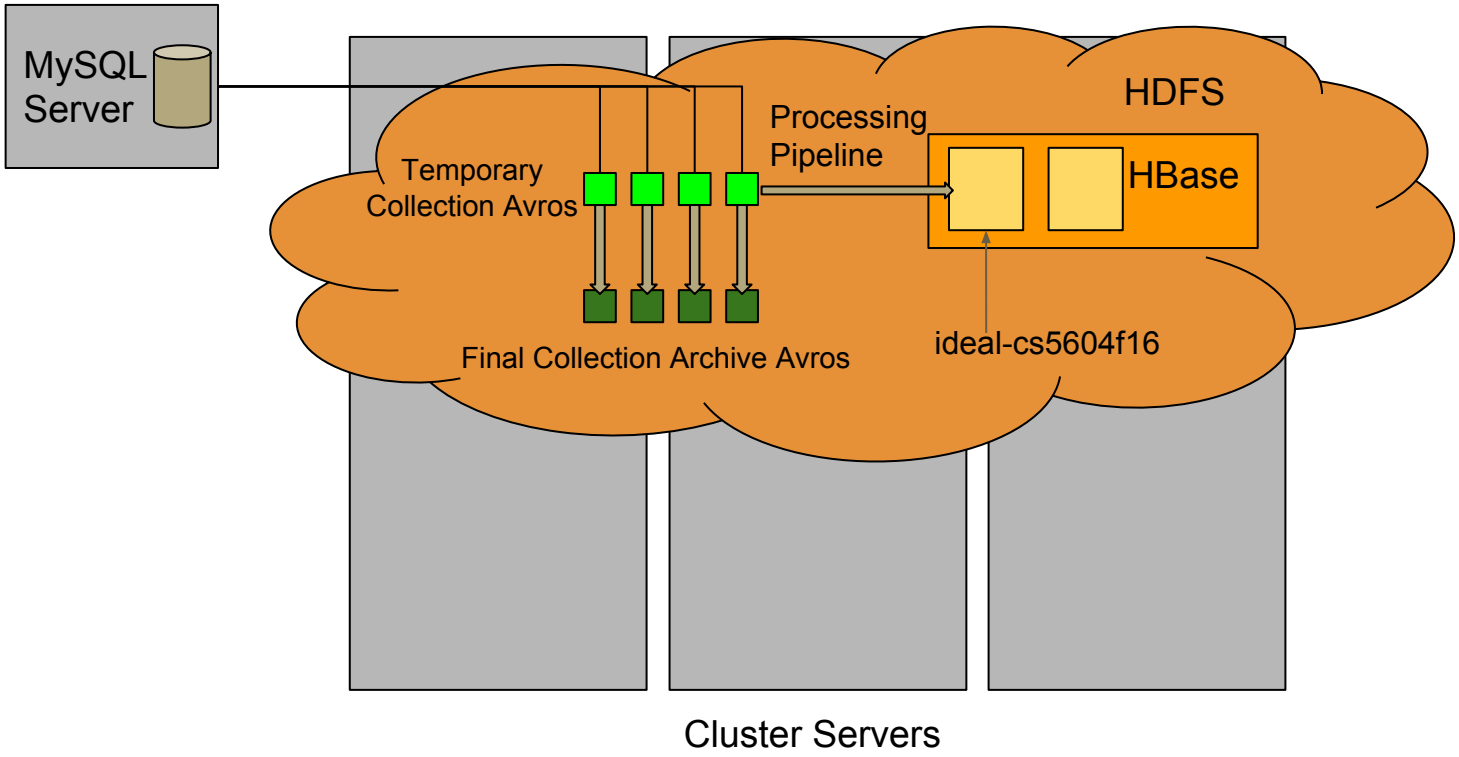



| Some statistics (cluster machine - 3.3 GHz, 32G) | | | |
|---|---|---|---|
| No. of tweets | Time | %CPU | Memory (MB) |
| 155657 | 14.42 sec | 45 | 439.5 |

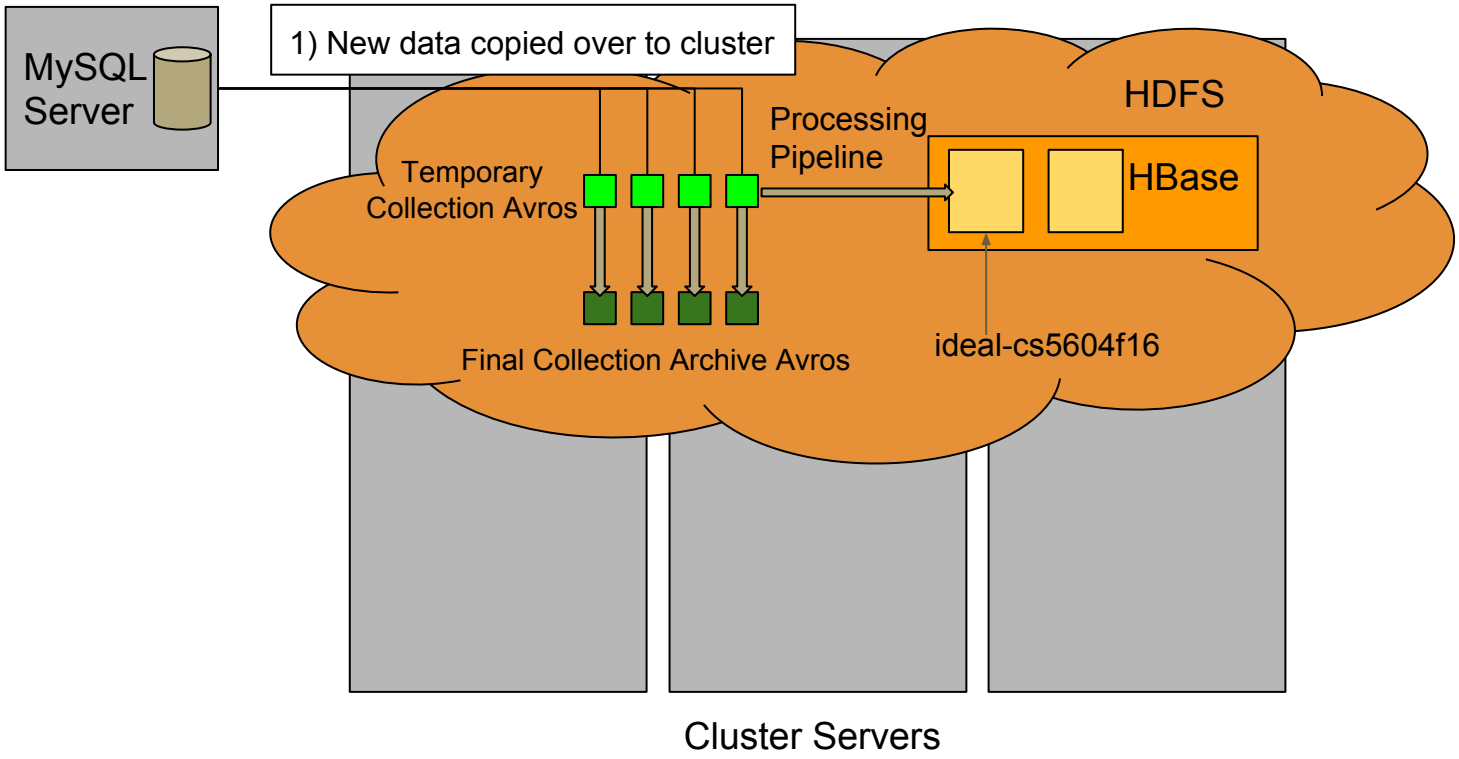

MySQL - CollectDB
(contains all new tweets)
pt-archiver
MySQL - ArchiveDB
(contains all raw tweets)
pt-archiver
Uncleaned text file
Bash scripts
Cleaned CSV file
csv2avro tool
Avro file
Bash scripts
HDFS
avro-tools
Merged Avro Files on HDFS

# Incremental Update from HDFS to HBase + Tweet Processing

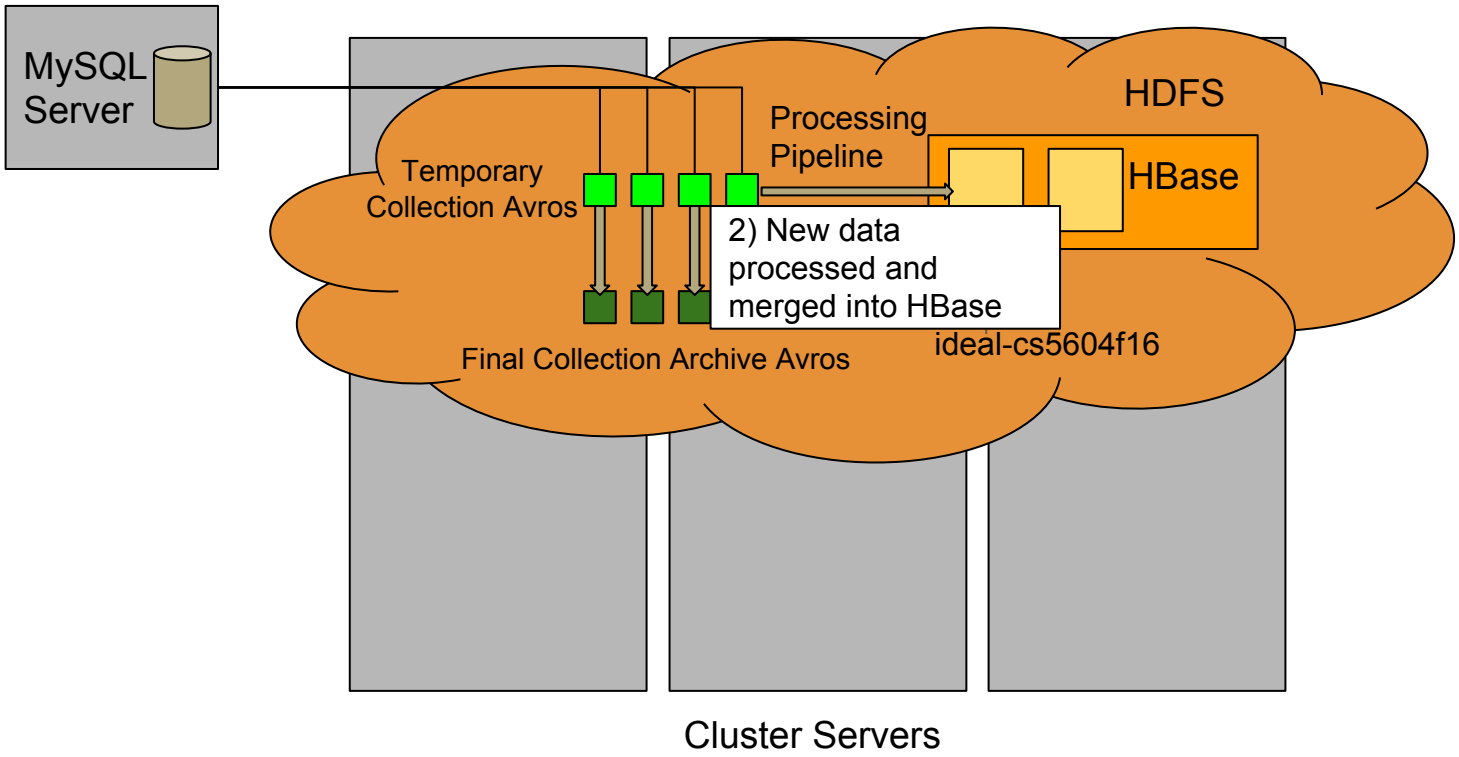
# Tweet Loading Pipeline



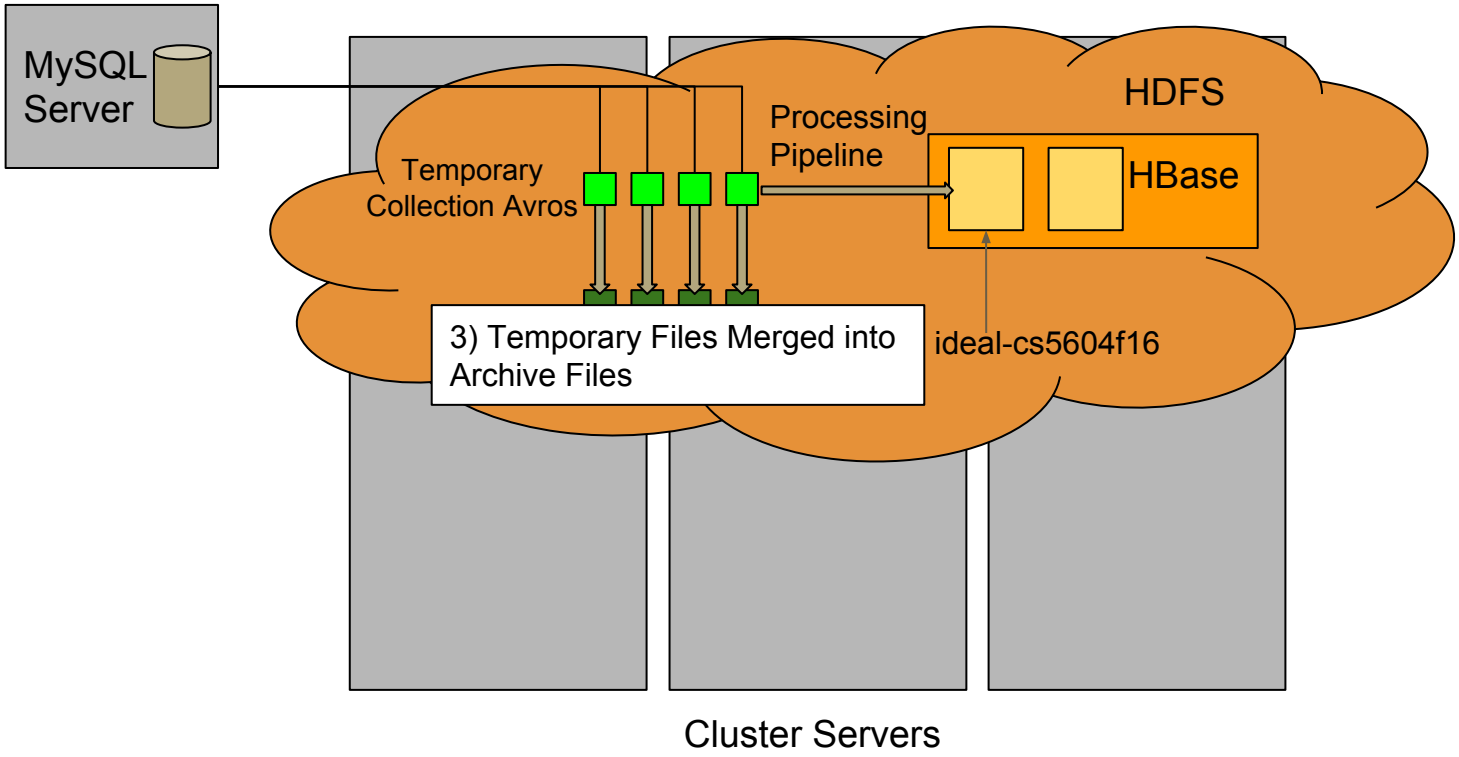
# Tweet Loading Pipeline



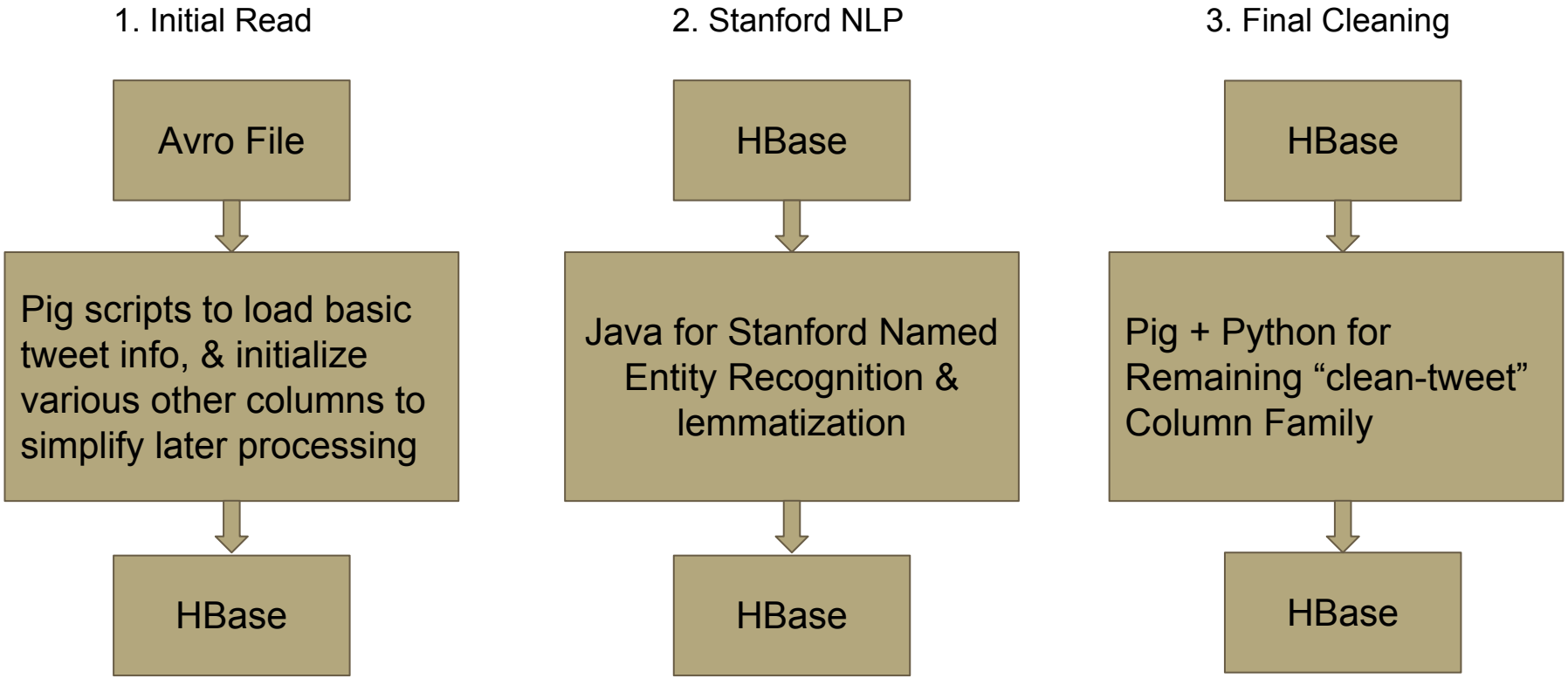
# Tweet Loading Pipeline



# Tweet Loading Pipeline



# Tweet Processing Pipeline



# Running Time Test

**Collection**: 312 (Water Main Break)

**Number of Tweets:** 155657

**Initial Read:** ~ 2 minutes

**Lemmatization:** ~33 minutes

**Cleaning Step:** ~27 minutes

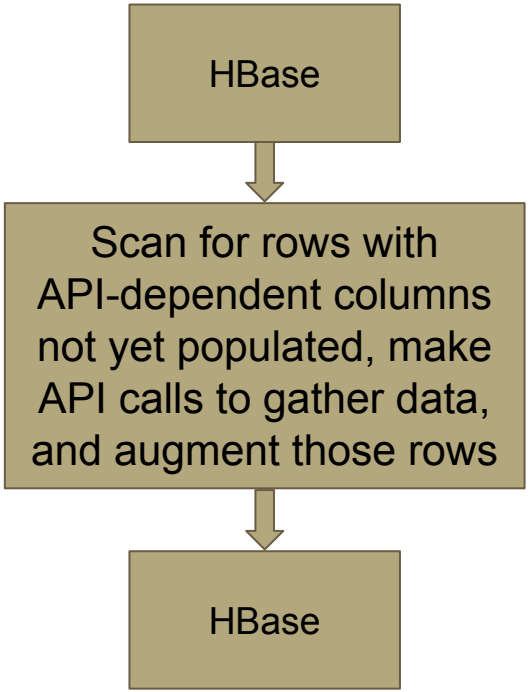
---

**Total time:** 1 hour

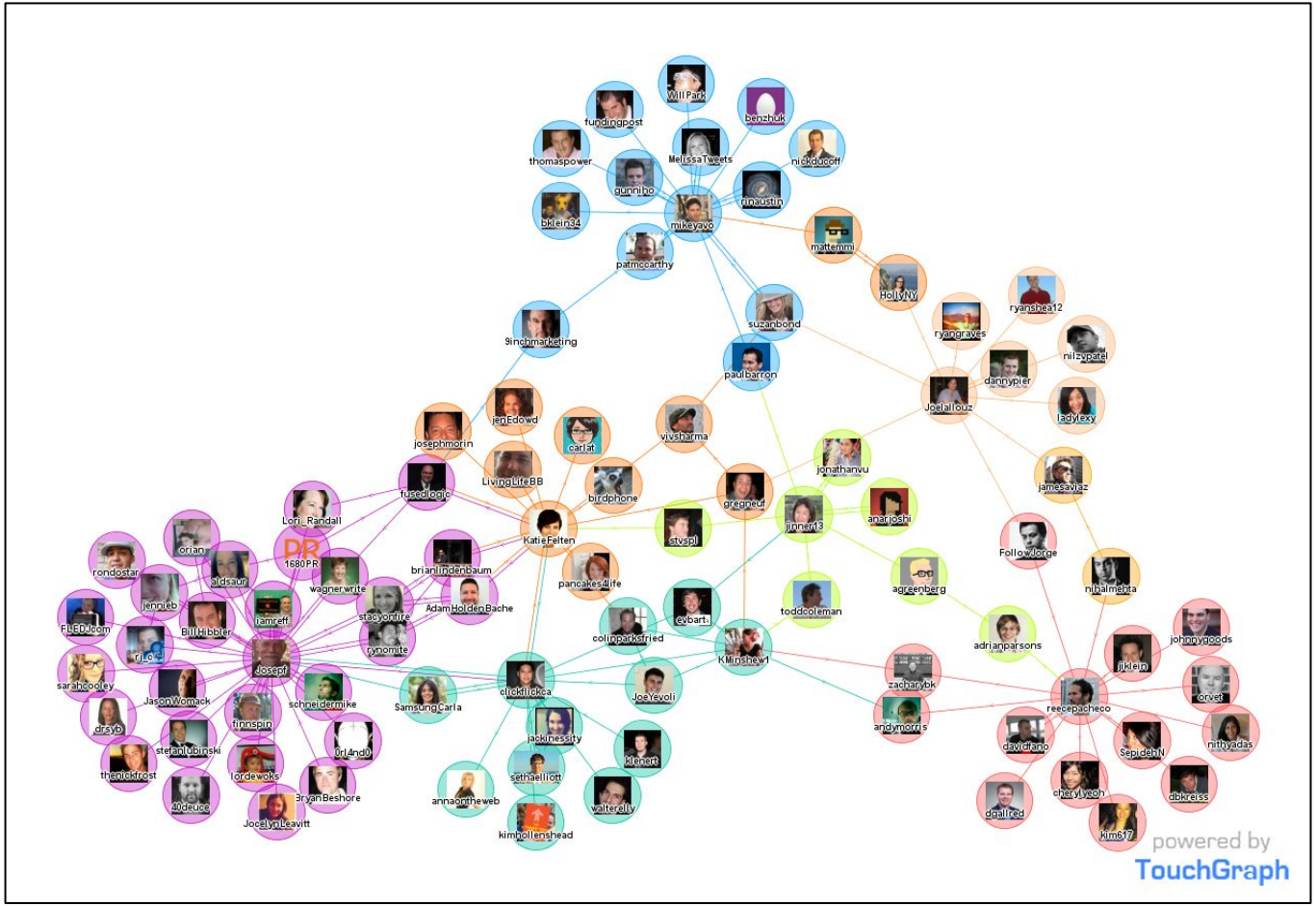
# Asynchronous Updates

Two clean-tweet columns are better suited for asynchronous updates:

- URL Extraction (Twitter has best information on URLs in tweets, rate-limited)
- Google Geolocation (rate-limited)



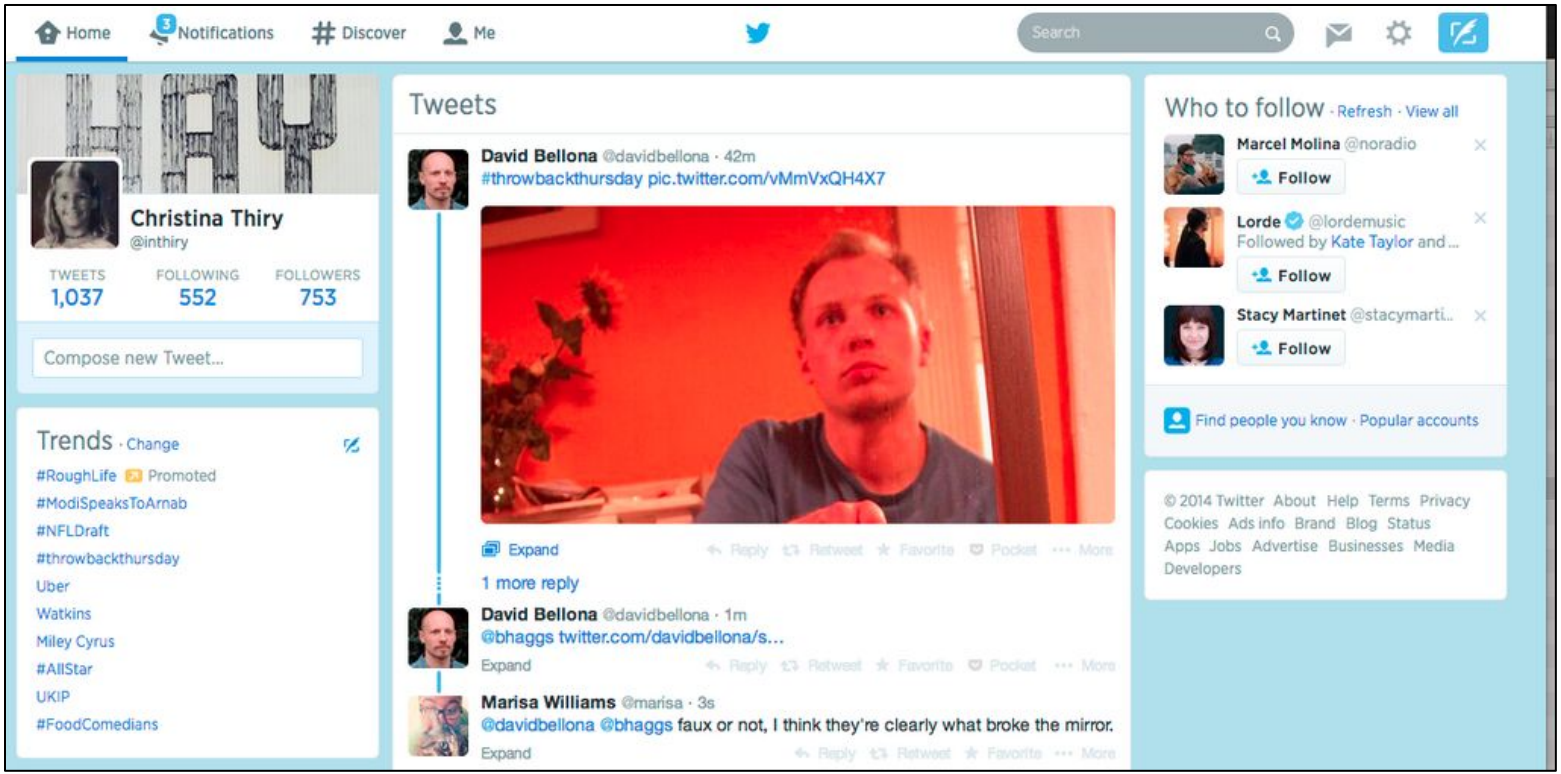
# Social Network




**Build a social network based on the tweet collection**

Credit: http://www.touchgraph.com

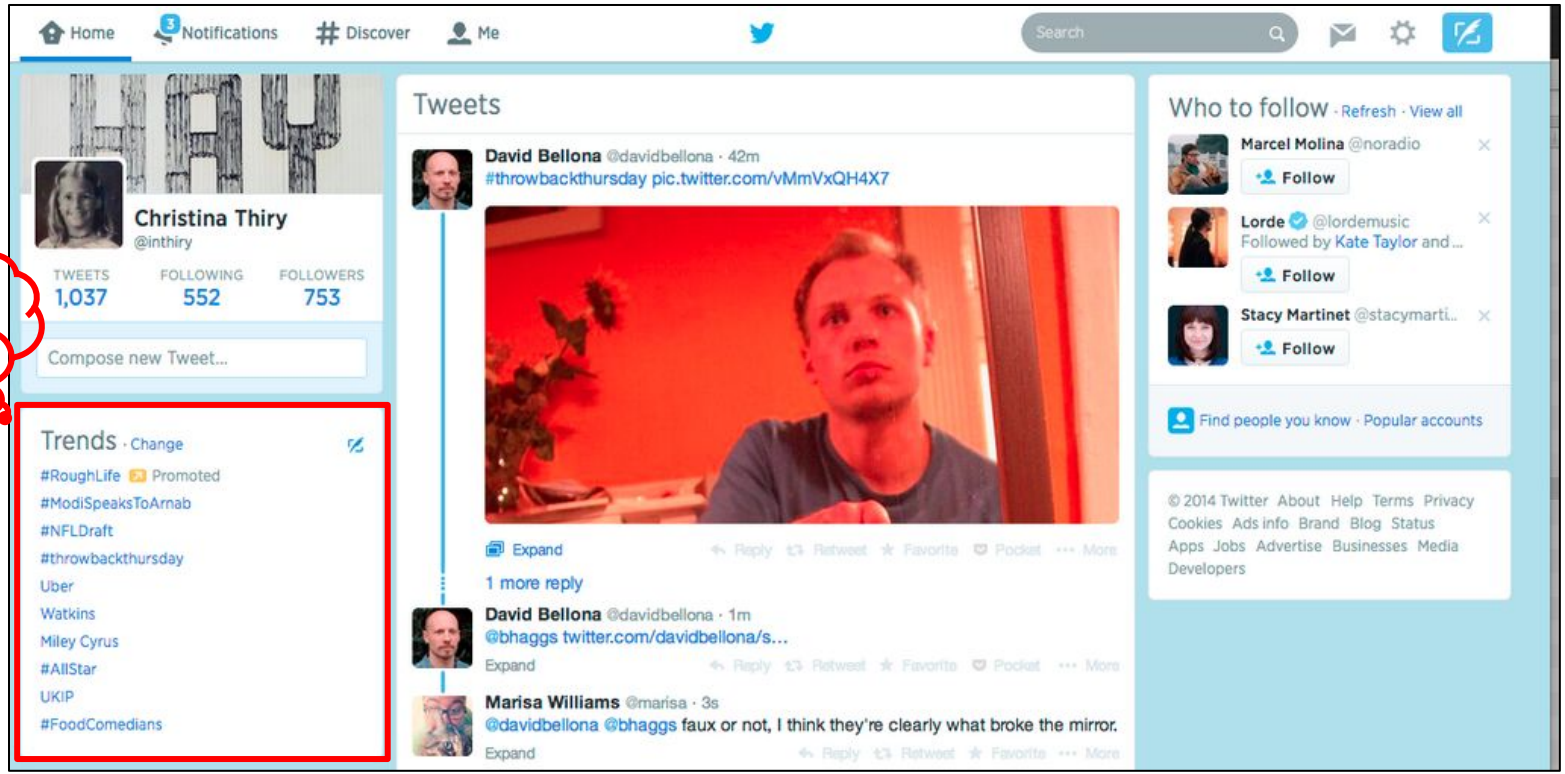
# Objective




**Rank the nodes for social network based recommendations**

Credit: http://thenextweb.com/twitter/2014/05/09/twitter-updates-web-layout-third-column-content-recommendation/

# Objective




Hot topics

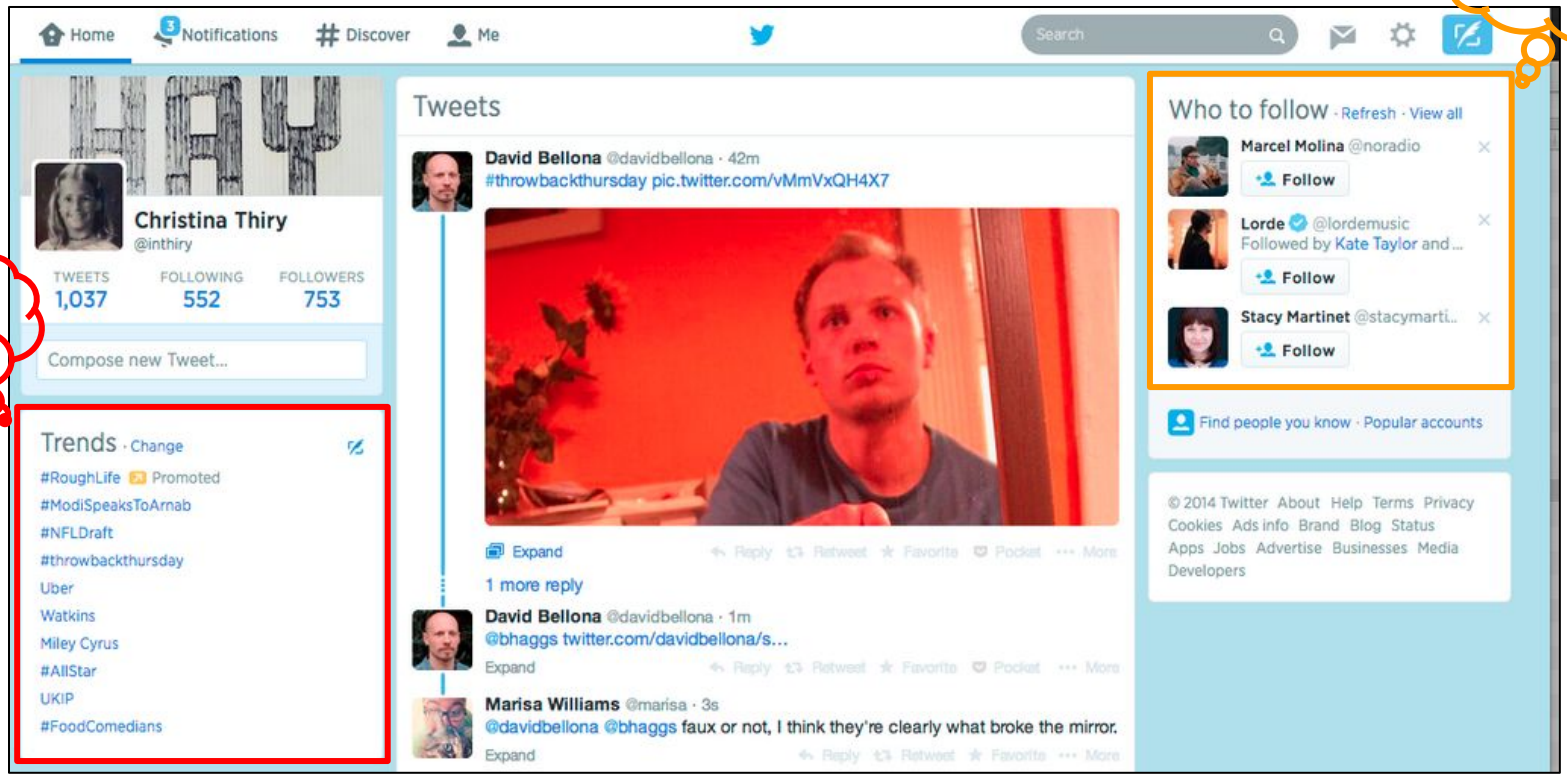
**Rank the nodes for social network based recommendations**

Credit: http://thenextweb.com/twitter/2014/05/09/twitter-updates-web-layout-third-column-content-recommendation/

# Objective

Popular people

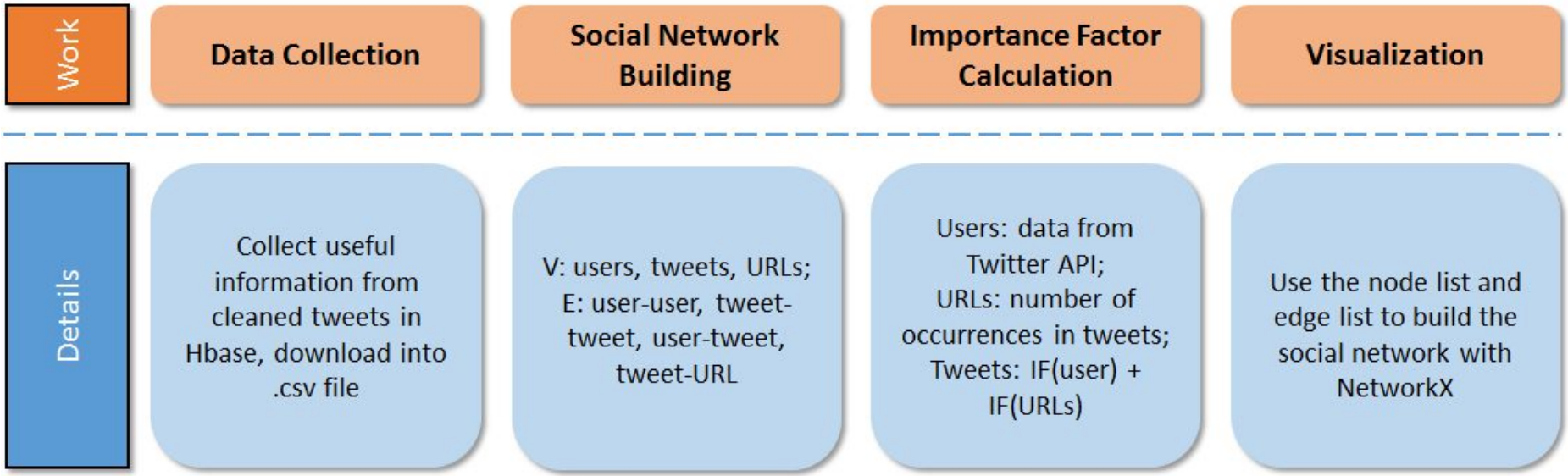
Hot topics




**Rank the nodes for social network based recommendations**

Credit: http://thenextweb.com/twitter/2014/05/09/twitter-updates-web-layout-third-column-content-recommendation/

# Pipeline

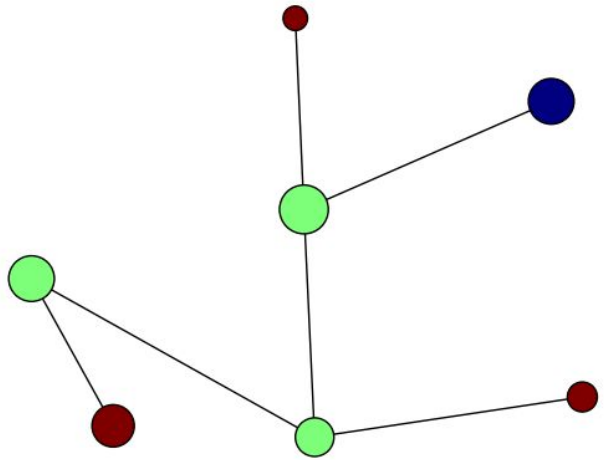

Work
Data Collection
Social Network Building
Importance Factor Calculation
Visualization
Details
Collect useful information from cleaned tweets in Hbase, download into .csv file
V: users, tweets, URLs; E: user-user, tweet-tweet, user-tweet, tweet-URL
Users: data from Twitter API; URLs: number of occurrences in tweets; Tweets: IF(user) + IF(URLs)
Use the node list and edge list to build the social network with NetworkX

# Previous work

- The S16 team built a social network G(V, E) where:
- Nodes (V): Users
- Edges (E): Edges between users according to RTs and mentions (@)
- Importance factor (IP): For edges (count)

# Nodes

| Nodes | Color |
| --- | --- |
| Users | Red |
| URLs | Blue |
| Tweets | Green |



# Edges

| Edges | Sources |
| --- | --- |
| User - User | Retweet (RT) , Mention (@) |
| Tweet - Tweet | Retweet (RT) |
| User - Tweet | If User posts the tweet |
| Tweet - URL | If the tweet includes the URL |

# Importance Factor

| Nodes | Importance Factor (IF) | Methods |
|---|---|---|
| Users | #followers, #friends, #statuses, #favorites, #listed (Twitter API) | IF(user) = 0.25 * #followers + 0.25 * #friends + 0.15 * #statuses + 0.25 * #favorites + 0.1* #listed |
| URLS | Number of occurrences of the URL in the tweet collection | IF(URLs) = #occurrences of a given URL / total number of URLs in the collection |
| Tweets | Importance factor of the tweeter and importance factor of URLs in the tweet | IF(Tweet) = .70 * IF(Users) + .30 * IF(URLs) |

# Visualization

- Tools
  - Python (NetworkX)

- Statistics
  - Number of tweets: 300
    - Collection z_3
  - Twitter API imposes size constraints
    - (180 queries every 15 minutes)

- Nodes
  - 300 tweet nodes
  - 158 user nodes
  - 110 URL nodes

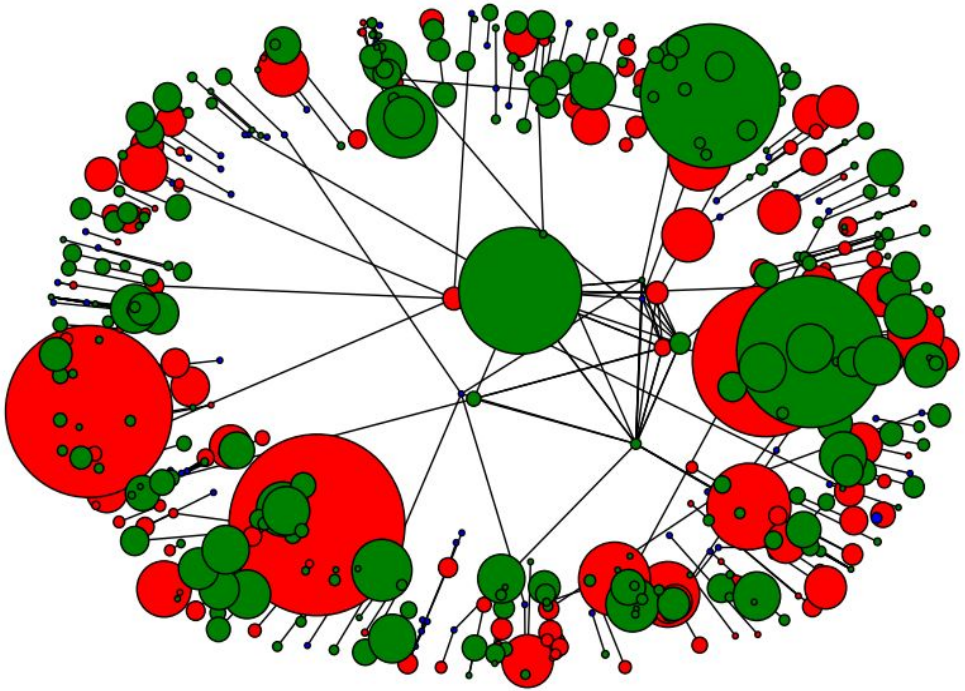
- Edges
  - 73 user-user edges
  - 54 tweet-tweet edges
  - 300 user-tweet edges
  - 140 tweet-URL edges

# Visualization

Green: tweets

Red: users

Blue: URLs

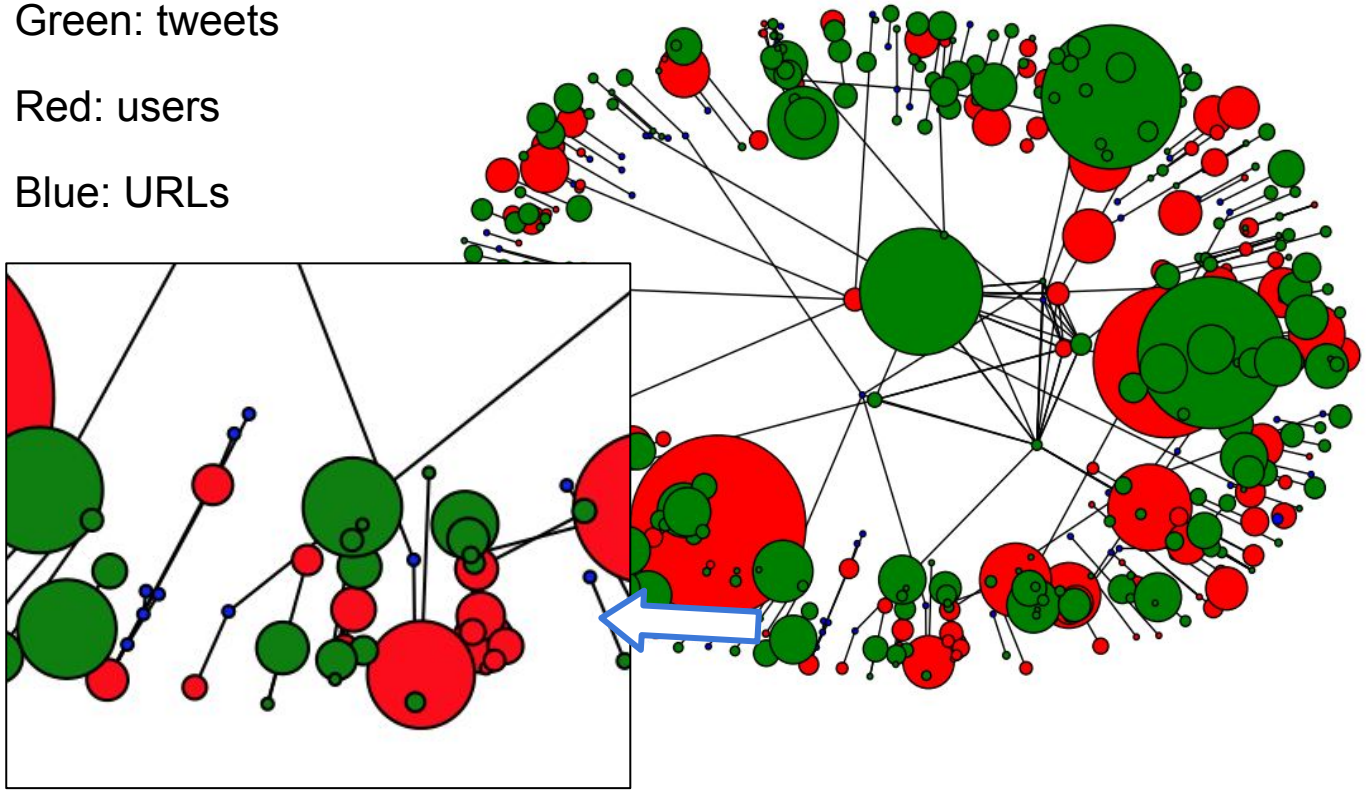


# Visualization

Green: tweets

Red: users

Blue: URLs



# Summary & Future Work

- We have delivered a robust ETL pipeline for moving tweets

- Can store and process thousands of tweets quickly
  - Flexible scripts accommodate large or small volumes of tweets

- In the future:
  - Do not remove comma, and double quotes from the text file of tweets
  - Develop asynchronous scripts to enhance tweets via API calls
  - Rigorous speed tests/processing pipeline optimization (including schema)
  - More extensive plan for handling profanity
  - Add hashtags to social network

# Challenges Faced

- Incomplete documentation from the previous semester
  - Schema
- Unfamiliarity with HBase, Pig, Twitter, Stanford NER
- Large, pre-existing system to understand
- Working in groups
  - Meeting time that works for all
  - Difficult to divide work based on our varying expertise
  - Dilemma to work together, or individually on parts of the project

# As a Learning Experience

- Exposure to different technologies
  - HBase + Hadoop Framework
  - Pig
  - Stanford NLP
  - Regex

- Concepts:
  - Extract, Transform, Load (ETL) Pipeline
  - NoSQL databases
  - Text parsing
  - Communication & synchronization between teams

- Overall
  - Divide responsibilities
  - Work iteratively
  - Ask questions

# Acknowledgement


- IDEAL: NSF IIS-1319578
- GETAR: NSF IIS-1619028
- Dr. Edward A. Fox
- GRA: Sunshin Lee

# References


1. Percona, “Percona - the database performance experts.” https://www.percona.com/, 2016.
2. “csv2avro - Convert CSV files to Avro .” https://github.com/sspinc/csv2avro, 2016.
3. A. A. Hagberg, D. A. Schult, and P. J. Swart, “Exploring network structure, dynamics, and function using NetworkX,” in Proceedings of the 7th Python in Science Conference (SciPy2008), (Pasadena, CA USA), pp. 11–15, Aug. 2008.
4. “CMT Team’s Codebase on GitHub.” https://github.com/mitchwagner/CMT, 2016.
5. “Touch Graph.” http://www.touchgraph.com/news, 2016.
6. N. Garun, “Twitter updates its Web layout with a third column for content recommendation.” http://thenextweb.com/twitter/2014/05/09/ twitter-updates-web-layout-third-column-content-recommendation/, 2014.