Hashtag Lifecycle
Ryan Boccabella and Kim Ngo
Original Plan

- Real Time Twitter Data
- MapReduce
  - Hashtags
  - Geo-Tag Location
  - Combine into Heat Map
Road Block

- Real Time Twitter Data
- Map Reduce
  - Hashtags
  - Geo-Tag Location
  - Combine into Heat Map

Twitter API Limits
Most users opt out from providing info
Revised Plans and Goals

1. Study Twitter Hashtag Lifecycle

2. Compare Sequential vs MapReduce performance time
1. Hashtag Lifecycle

Overview
People
Publications
Demos
Datasets

We put here several datasets being used in some of our demos. Please cite the source of these datasets as the 'Apollo Project, Department of Computer Science and Engineering, University of Notre Dame' when using these datasets in your publications, demos, or presentations.

Please email dwang5 at nd dot edu to contact for collaborations or request username/password to download these datasets. Also, if you have a request for data about an on-going event, please suggest via email as well.

Crimea Unrest
Around 1.9 million tweets were collected from February 19, 2014 to April 9, 2014. Download

Syria Tactical Weapon
Around 205,000 tweets were collected from August 22, 2013 to August 31, 2013. Download

Boston Marathon Bombings
Around 543,000 tweets were collected from April 15, 2013 to April 22, 2013. Download

Hurricane Sandy
Around 904,000 tweets were collected from October 27, 2012 to November 18, 2012. Download
"text": "RT @GuyCodes: Prayers go our to all of the victims of the Boston marathon explosion, especially this little guy. #prayforboston http://t.co/ejoDBqfi0d",
"profile_image_url": "http://a0.twimg.com/profile_images/3412256835/d5ae2611fececd3d98a1b47a16a8dbbd_normal.jpeg",
"from_user": "ginoo_xD",
"from_user_id": 464971414,
"geo": null,
"id": 323906397597216768,
"iso_language_code": "en",
"from_user_id_str": "464971414",
"created_at": "Mon, 15 Apr 2013 21:11:17 +0000",
"source": "&lt;a href=&quot;http://twitter.com/download/iphone&quot;&gt;Twitter for iPhone&lt;/a&gt;",
"id_str": "323906397597216768",
"from_user_name": "YoursTruly\u2693",
"profile_image_url_https": "https://si0.twimg.com/profile_images/3412256835/d5ae2611fececd3d98a1b47a16a8dbbd_normal.jpeg",
"metadata": {"result_type": "recent"}
"text": "RT @GuyCodes: Prayers go our to all of the victims of the Boston marathon explosion, especially this little guy. #prayforboston http://t.co/ejoDBqfi0d",
"profile_image_url": "http://a0.twimg.com/profile_images/3412256835/d5ae2611fececd3d98a1b47a16a8dbbd_normal.jpeg",
"from_user": "ginoo__xD",
"from_user_id": 464971414,
"geo": null,
"id": 323906397597216768,
"iso_language_code": "en",
"from_user_name": "Yours Truly\u2693",
"created_at": "Mon, 15 Apr 2013 21:11:17 +0000",
"source": "&lt;a href='http://twitter.com/download/iphone'&gt;Twitter for iPhone&lt;/a&gt;"
Hadoop MapReduce

- MapReduce:
  - Similar to Word Count
  - Map: Emit (hashtag, time)
  - Reduce: Emit (hashtag, list[pair(time, count)])
MapReduce

Tweet → (Hashtag, Time) → Hashtag, Time, Count
Boston Bombing 4/15/13 - 4/22/13

Lifetime of #BostonMarathon
Boston Bombing

# of hashtag occurrences vs Time From 0 Hour (Hours)
Crimea Unrest

2/19/14 - 4/9/14
2. Sequential vs MapReduce

- **Sequential:** C++
  - Read stdin
  - Hashmap `<hashtag, vector<time, count>>`

- **MapReduce:** Python
  - Map: Emit (hashtag, time)
  - Reduce: Emit (hashtag, list[pair(time, count)])
## Performance Results

<table>
<thead>
<tr>
<th>Size</th>
<th>Sequential Time</th>
<th>MapReduce Time</th>
<th>Speedup*</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 KB</td>
<td>0.01 sec</td>
<td>15 sec</td>
<td>0.00066</td>
</tr>
<tr>
<td>500 KB</td>
<td>0.04 sec</td>
<td>18 sec</td>
<td>0.00222</td>
</tr>
<tr>
<td>1 MB</td>
<td>0.08 sec</td>
<td>18 sec</td>
<td>0.00444</td>
</tr>
<tr>
<td>100 MB</td>
<td>8.38 sec</td>
<td>21 sec</td>
<td>0.39905</td>
</tr>
<tr>
<td>500 MB</td>
<td>40.11 sec</td>
<td>21 sec</td>
<td>1.91</td>
</tr>
<tr>
<td>1 GB</td>
<td>1 min 17.53 sec</td>
<td>21 sec</td>
<td>3.69</td>
</tr>
<tr>
<td>4 GB</td>
<td>5 min 2.75 sec</td>
<td>24 sec</td>
<td>12.61</td>
</tr>
<tr>
<td>8 GB</td>
<td>10 min 6.07 sec</td>
<td>33 sec</td>
<td>18.36</td>
</tr>
<tr>
<td>850 GB</td>
<td>17 hr 53 min 14.94 sec**</td>
<td>24 min 52 sec</td>
<td>64394.94</td>
</tr>
</tbody>
</table>

* Speedup = Sequential / Parallel

** (10mins 6.07 sec) x 850/8 = 17 hr 53 min 14.94 sec
Performance Results

- Sequential
- MapReduce

Execution Time (seconds) vs. File Size (GB)

Execution Time (seconds) vs. File Size (KB)
What We’ve Done

● Created infrastructure for lifecycle analysis
● Shown high versatility, it’s easy to:
  ○ Plot
  ○ Run statistical analyses
● Explored MapReduce overhead time
What We Can Do

- Analyze trending hashtag characteristics
  - Relevance to an event
  - Word length