Details

Saturday, 20 June 2015 12:17 pm

1. Nikhil - Source folder structure definition
2. Nikhil - Twitter feed:
   a. Get it into flat text file
      i. Each line is json tweet object with data mentioned in point c.
   b. One file per product (hashtag)
   c. Tweet, likes on tweet, hashtag, geo location, gender, celebrity flag
   d. Get initial data. Use it to generate the system, collect data every day multiple time and add to mapper input.
3. Ajith - Mapper: Create indico.io index:
   a. Manually import data into HDFS
   b. Scan text
   c. Calculate indico.io happiness index
   d. Save in format
      i. Key: name of product
      ii. Value: Class: Tweet, HappinessIndex
      iii. Use Java object to write data - context.write(twitterKey, twitterValue)
4. *Dehasish - Mapper: Map Happiness Index to Happy, Neutral, Not Happy based on range.* ***(For later)***
   and associated ***Reducer*** *- Happy 5, NotHappy 4*
   0.1, 0.3, 0.5, 0.9
   Mapper*:* <NotHappy,1>
   NotHappy, 1
   Neutral, 1
   Happy, 1
   Reducer:
   NotHappy,2
   Neutral, 1
   Happy, 1

5. *Rushabh - Combiner: Group all happy, neutral and not happy.* ***(For later)***
6. Sugesh- Reducer: Output: One json per product

```
{
	"Product":"iPhone6",
	"Happy":60
	"TopHappyTweets":[
		"So happy with this",
		"Happy",
	],
	"NotHappy":20
	"TopNotHappyTweets:[
		"So happy with this",
		"Happy",
        ]
}
```

7. Jagadeesh - Web interface: (Put it in PaaS <Value Add>)
    a. No text search (In progress)
    b. List or combo box
    c. VS list
    d. UI Page
        i. Index - Happy/Not happy
        ii. Selection checkbox
        iii. Tweet list
        iv. Pie chart (or any other)
8. Rushabh - Publish to Paas Amazon/Azure (Nikhil)
9. Nikhil/Sugesh - Project Documentation
10. Jagadeesh - Push files to HDFS using Apache Flume (Additional)