

# Cloud Computing and Big Data

# Hadoop Adjuncts and Extras

Oxford University  
Software Engineering  
Programme  
July 2018



© Paul Fremantle 2015. This work is licensed under a Creative Commons  
Attribution-NonCommercial-ShareAlike 4.0 International License  
See <http://creativecommons.org/licenses/by-nc-sa/4.0/>

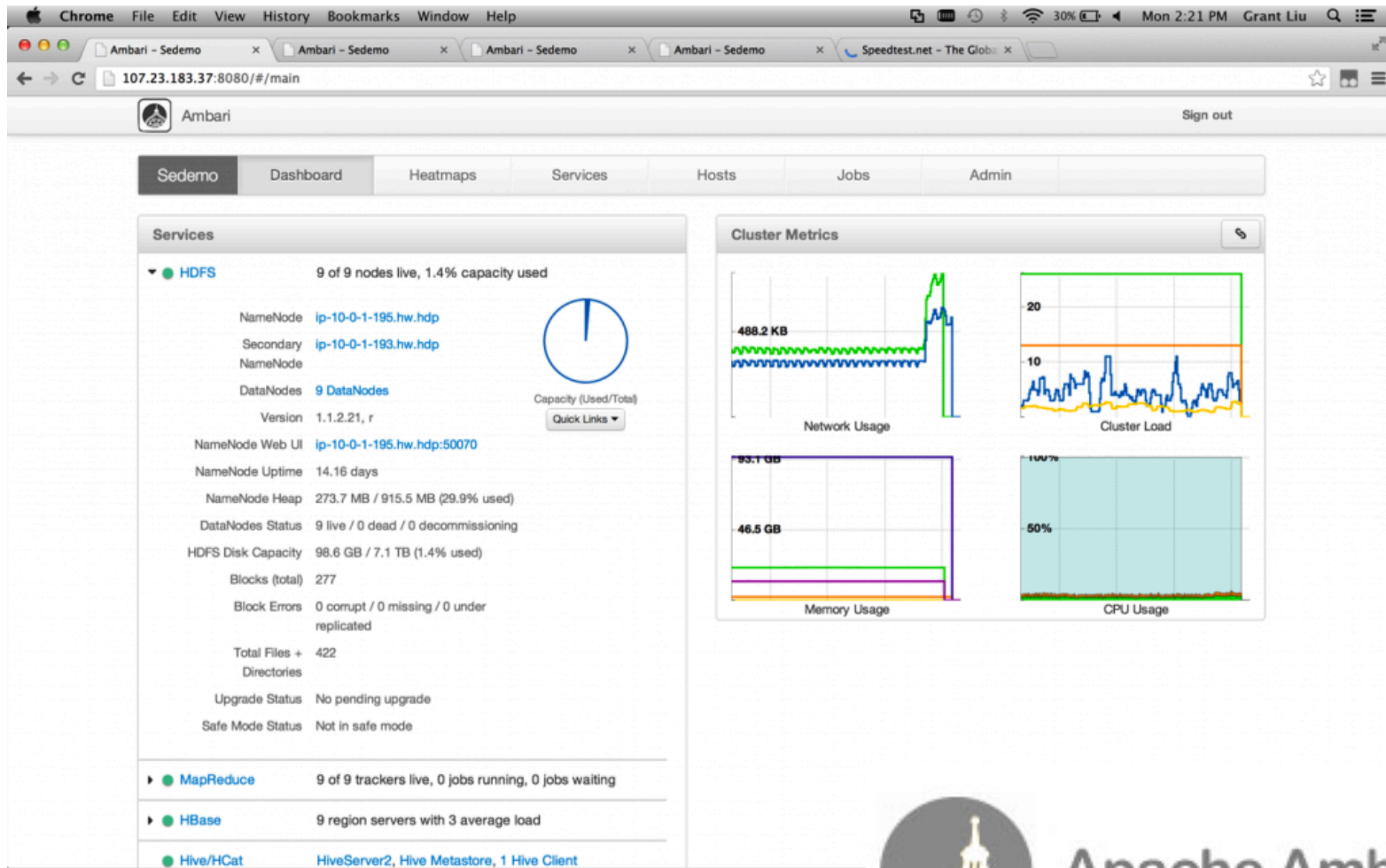
# Hadoop related projects at Apache

- Ambari
  - Web based monitoring for Hadoop
- HBase
  - Scalable, Distributed database
- Hive
  - SQL query language for Map Reduce
- Pig
  - Dataflow and execution language for parallel execution
- Mahout
  - Machine Learning
- Chukwa
  - Log collection and processing on top of Hadoop
- Spark
  - Large scale data processing on top of YARN or Mesos
- Sqoop
  - Transfer of data into Hadoop from traditional databases
- Tez
  - Going beyond Map Reduce on top of YARN



# Apache Ambari

<http://ambari.apache.org>



Apache Ambari



# Apache HBase

- More in the NoSQL section



# Apache Hive

<http://hive.apache.org>



- Just like SQL except it generates Map Reduce jobs
- Works on Hadoop and Spark
  - Part of SparkSQL
- Includes DDL (Data Definition Language) as well as SQL
- Makes many processing tasks very simple



# Hive example

```
CREATE TABLE page_view(viewTime INT, userid BIGINT,  
                        page_url STRING, referrer_url STRING,  
                        ip STRING COMMENT 'IP Address of the User')  
COMMENT 'This is the page view table'  
PARTITIONED BY(dt STRING, country STRING)  
STORED AS SEQUENCEFILE;
```

```
LOAD DATA LOCAL INPATH /tmp/pv_2008-06-08_us.txt INTO TABLE page_view  
PARTITION(date='2008-06-08', country='US')
```

```
INSERT OVERWRITE TABLE xyz_com_page_views  
SELECT page_views.*  
FROM page_views  
WHERE page_views.date >= '2008-03-01' AND page_views.date <=  
'2008-03-31' AND  
      page_views.referrer_url like '%xyz.com';
```



# Apache Pig

<http://pig.apache.org>

- Pig is a language for parsing, sorting, and working with data from HDFS
- Pig scripts are runnable on Hadoop MapReduce
- Very effective approach



# Pig Latin example

```
raw = LOAD 'excite.log' USING PigStorage('\t') AS (user, time, query);
clean1 = FILTER raw BY org.apache.pig.tutorial.NonURLDetector(query);
clean2 = FOREACH clean1 GENERATE user, time,
org.apache.pig.tutorial.ToLower(query) as query;
houred = FOREACH clean2 GENERATE user, org.apache.pig.tutorial.ExtractHour(time)
as hour, query;
ngramed1 = FOREACH houred GENERATE user, hour,
flatten(org.apache.pig.tutorial.NGramGenerator(query)) as ngram;
ngramed2 = DISTINCT ngramed1;
hour_frequency1 = GROUP ngramed2 BY (ngram, hour);
hour_frequency2 = FOREACH hour_frequency1 GENERATE flatten($0), COUNT($1) as
count;
uniq_frequency1 = GROUP hour_frequency2 BY group::ngram;
uniq_frequency2 = FOREACH uniq_frequency1 GENERATE flatten($0),
flatten(org.apache.pig.tutorial.ScoreGenerator($1));
uniq_frequency3 = FOREACH uniq_frequency2 GENERATE $1 as hour, $0 as ngram, $2
as score, $3 as count, $4 as mean;
filtered_uniq_frequency = FILTER uniq_frequency3 BY score > 2.0;
ordered_uniq_frequency = ORDER filtered_uniq_frequency BY (hour, score);
STORE ordered_uniq_frequency INTO '/tmp/tutorial-results' USING PigStorage();
```





# mahout

/mə'haʊt/ 

*noun*

(in South and SE Asia) a person who works with and rides an elephant.



# Apache Mahout

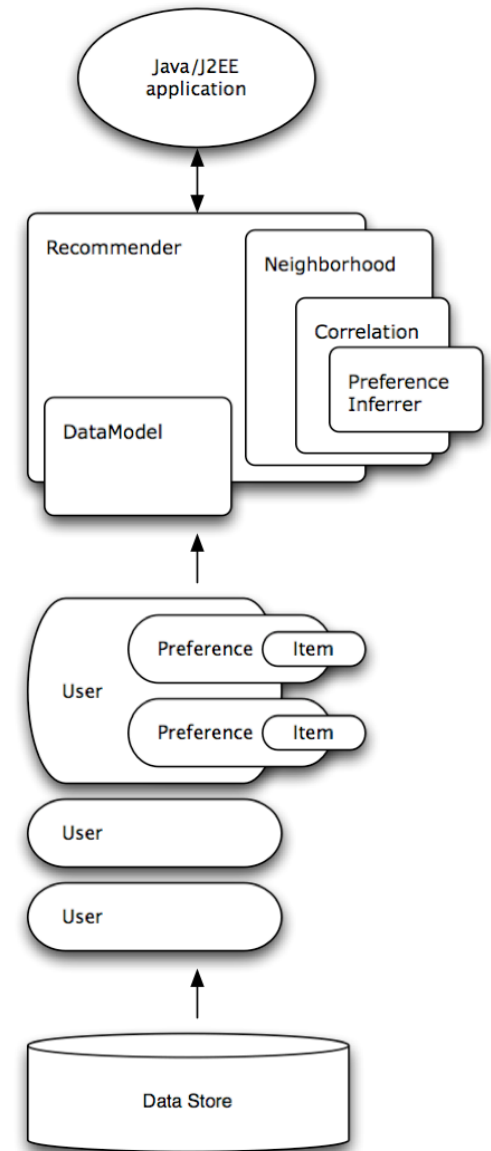
<http://mahout.apache.org>

- A system for creating **scalable machine learning** and data mining systems
  - Clustering
  - Classification
  - Recommendation
  - Frequent ItemSet



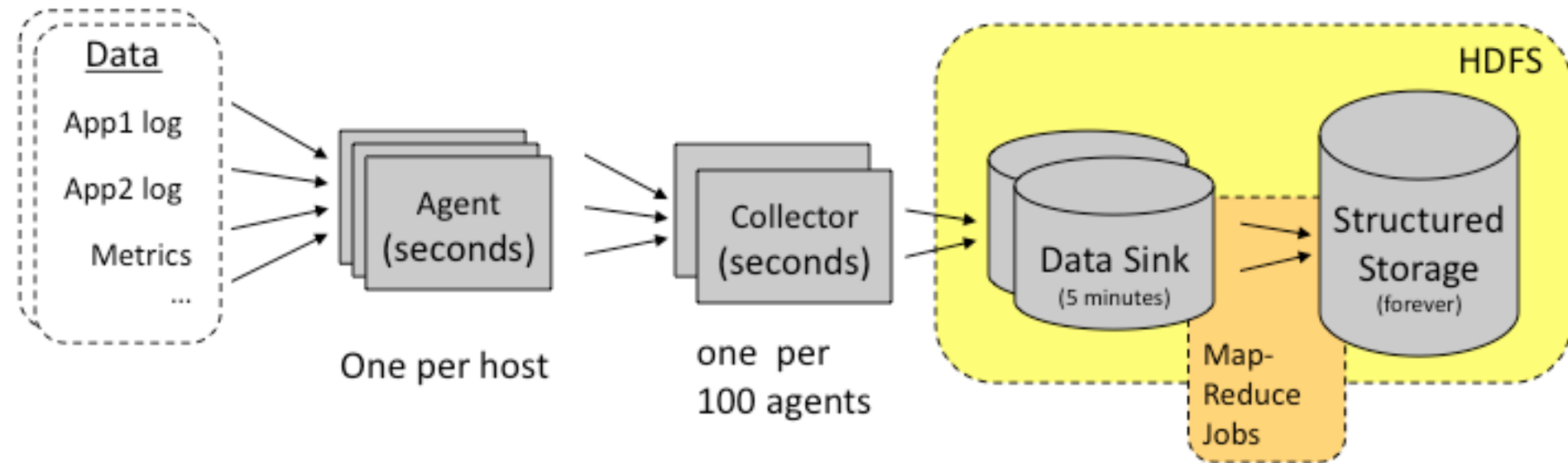
# e.g. Recommender

- Takes users preferences
  - “Likes”
- Estimates preferences for other items
- For example, which books you might like to read next



# Apache Chukwa

<http://chukwa.apache.org>



# Spark

- Much more later!



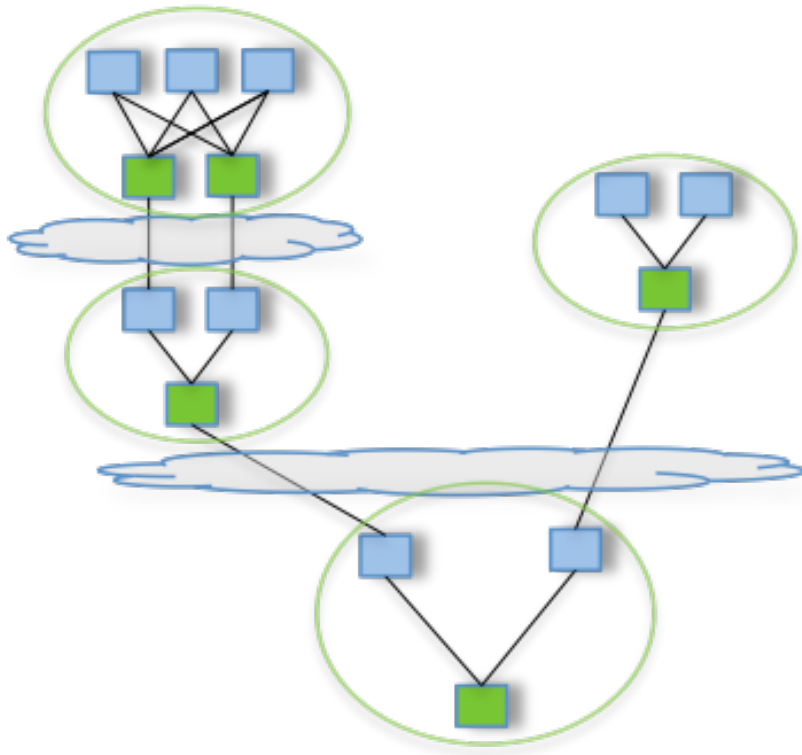
# Apache Tez

<http://tez.apache.org>

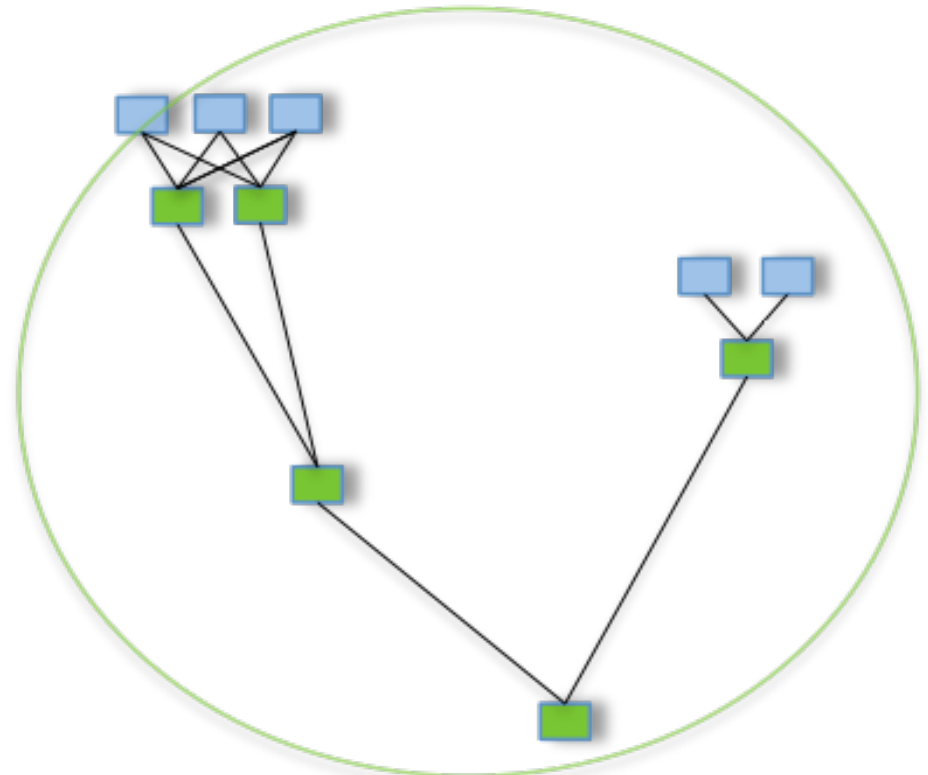
- Support for complex Directed Acyclic Graphs (DAGs) on top of YARN
  - Supports in memory jobs
  - Simplifies work that would previously be in multiple MR jobs
- Designed to support Pig and Hive



# Tez



Pig/Hive - MR



Pig/Hive - Tez

# Apache Avro

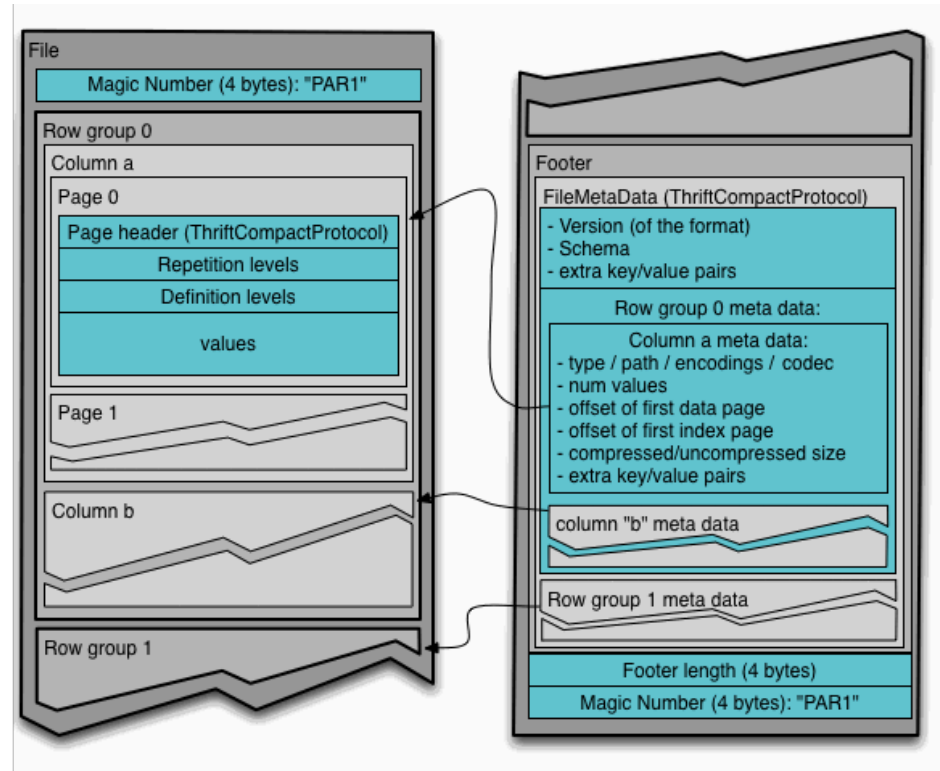
- A compact data storage and transmission system
  - Uses schemas of data to ensure it can be read by the receiver
  - Supports dynamic typing
- Used by RPC or data collection systems
  - Fast binary protocols
- Also supports storage
  - Hence used by many Big Data apps including Hadoop and Spark





# Apache Parquet

- Apache Parquet is a columnar data storage model
  - Works with Hadoop, Spark and many others
  - Efficient storage of data
  - Based on another Google system called Dremel



# Questions?



© Paul Fremantle 2015. This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License  
See <http://creativecommons.org/licenses/by-nc-sa/4.0/>