Learning from Big Data: Architecture Options

Andy Feng (@afeng76, afeng@apache.org)
Distinguished Architect, Yahoo
Feb. 28, 2014

Who am I?

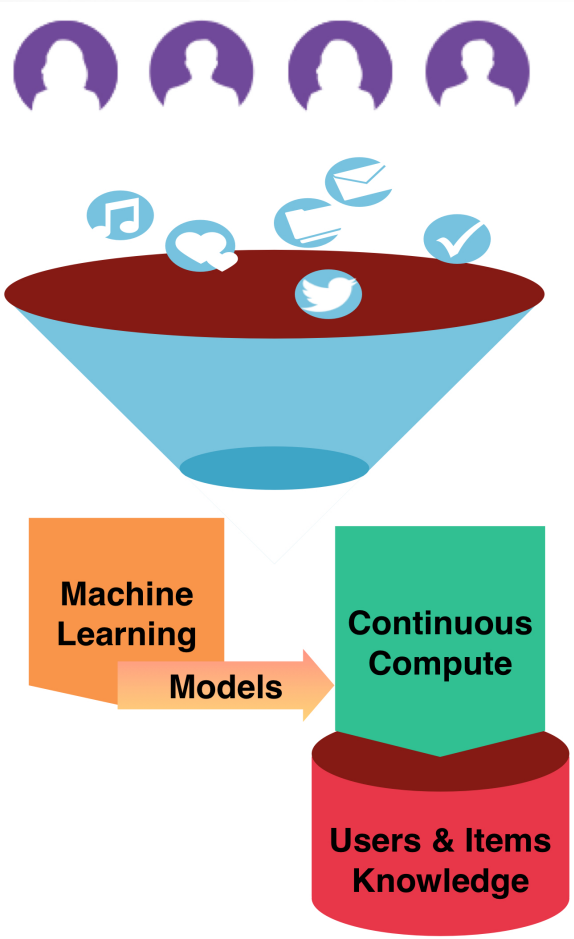
- Current
  - Distinguished Architect, Yahoo Hadoop Group
  - Committer, Apache Storm
  - Contributor, Apache Spark
- Past
  - Online advertisement
  - Personalization
  - Cloud services
  - NoSQL

YAHOO!

Agenda

1. Machine Learning
   - Use Cases
   - Libraries
2. Big-Data Technologies
   - Hadoop, Spark and Storm
3. Design Patterns
   - Batch/incremental learning
   - Real-time learning
   - Hybrid learning

Yahoo Machine Learning



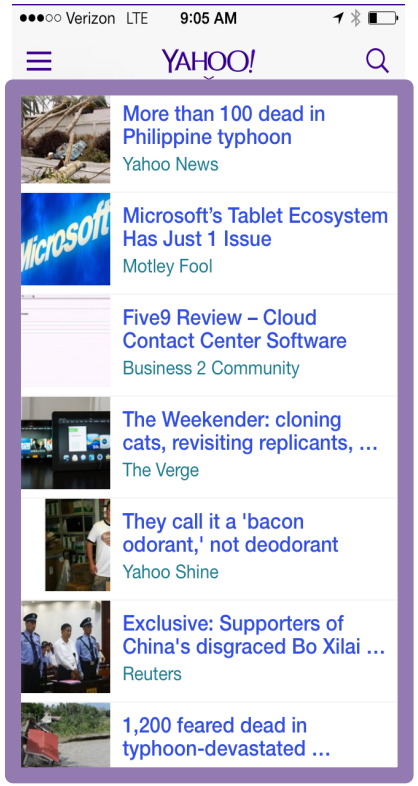
- Will a user enjoy a new article?
- What vertical contents are relevant to a user search?
- What products should we recommend a user?
- Is this email a spam?
- Is this ad click fraud?
- Was this photo taken at a beach?
- Find all potential users to be targeted by an ad campaign
- …

YAHOO!

Use Case 1: Personalized Homepage

http://www.yahoo.com

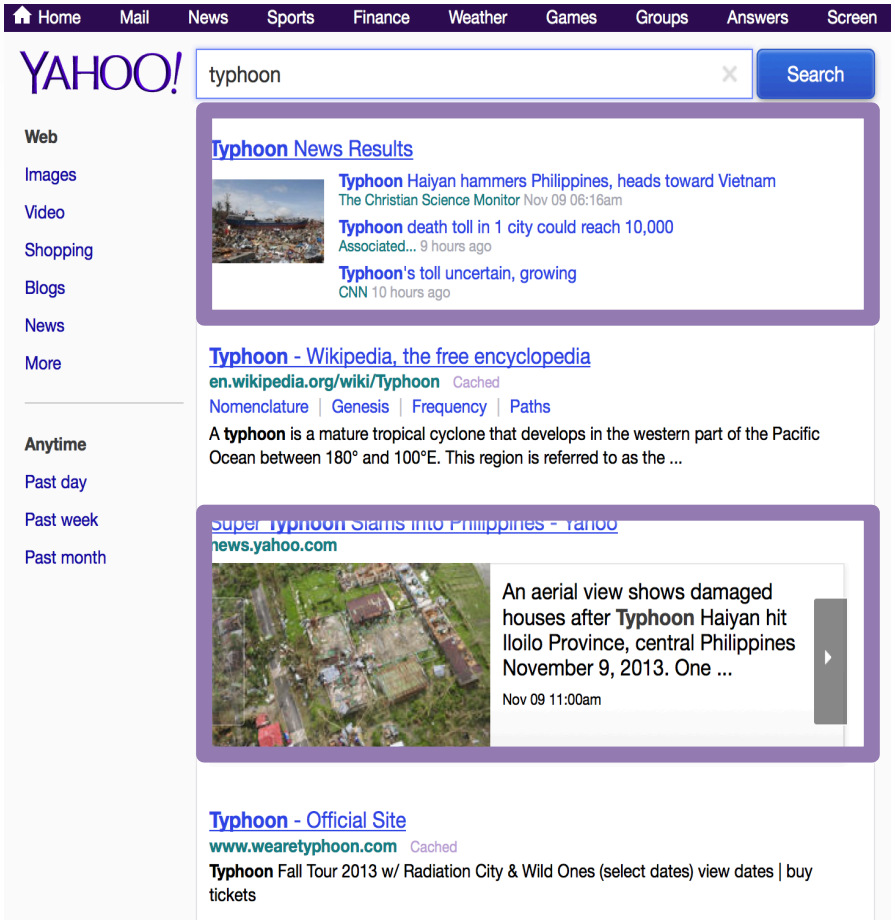
Mobile



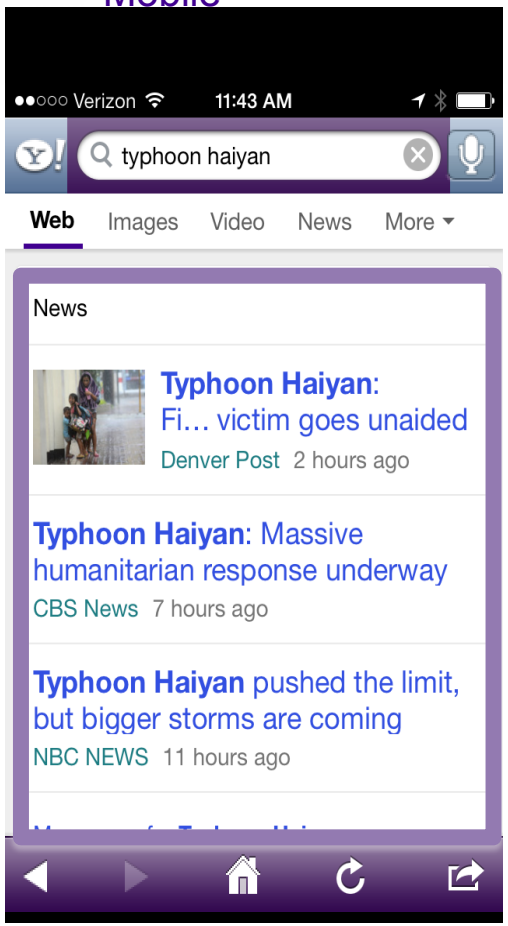
YAHOO!

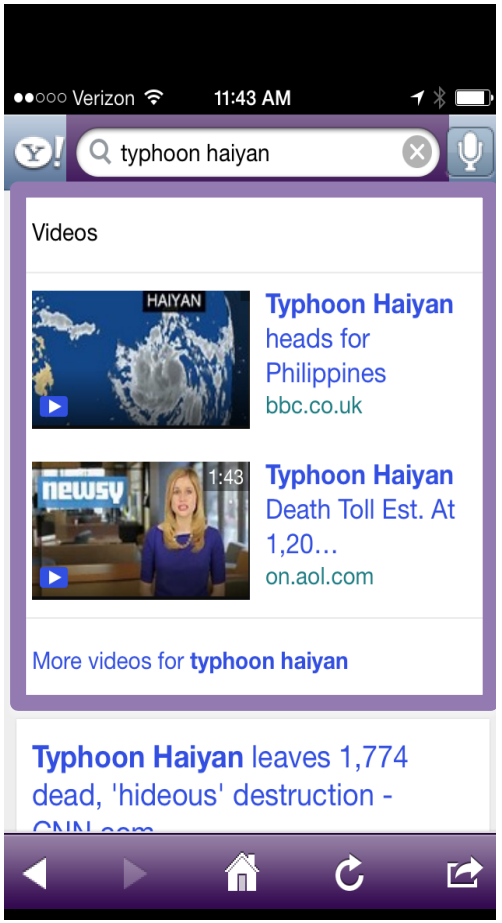
Use Case 2: Improved WEB Search w/ Vertical Content

http://search.yahoo.com

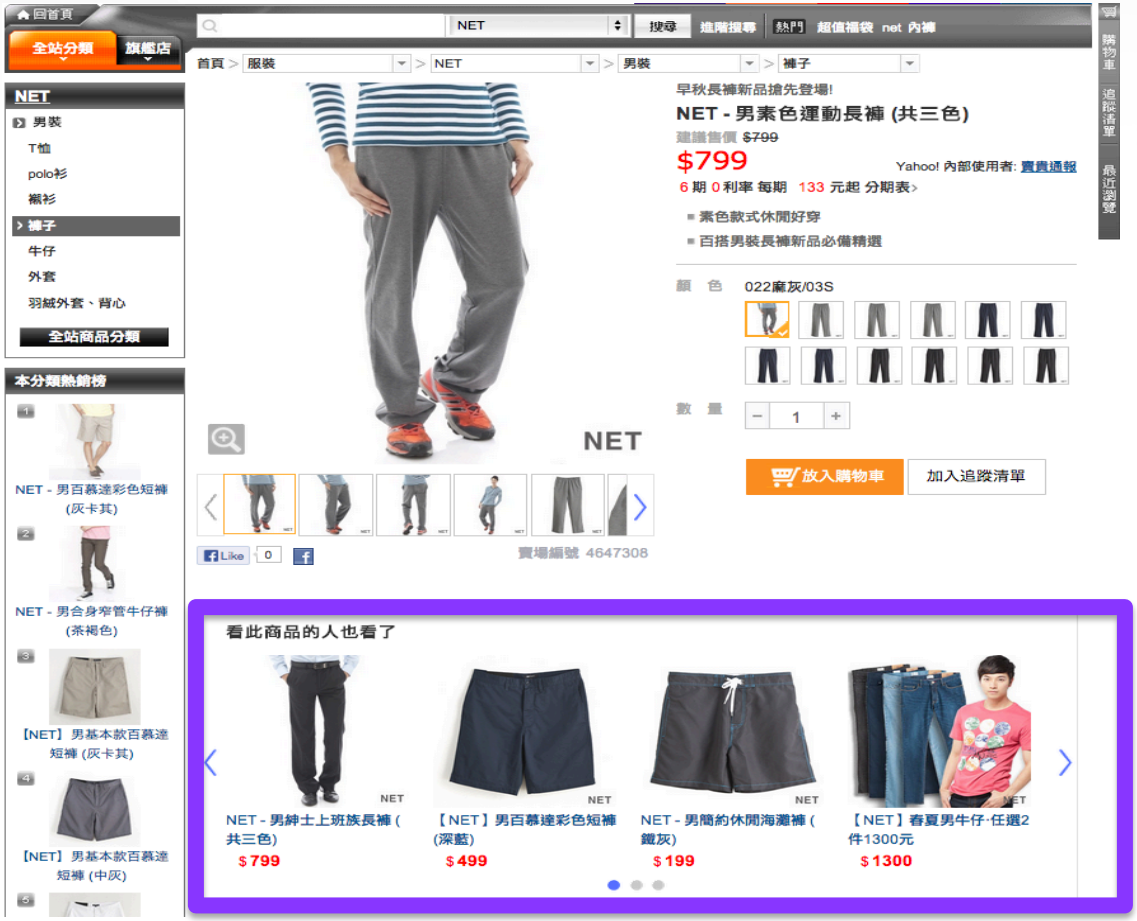


Mobile





Use Case 3: E-Commerce



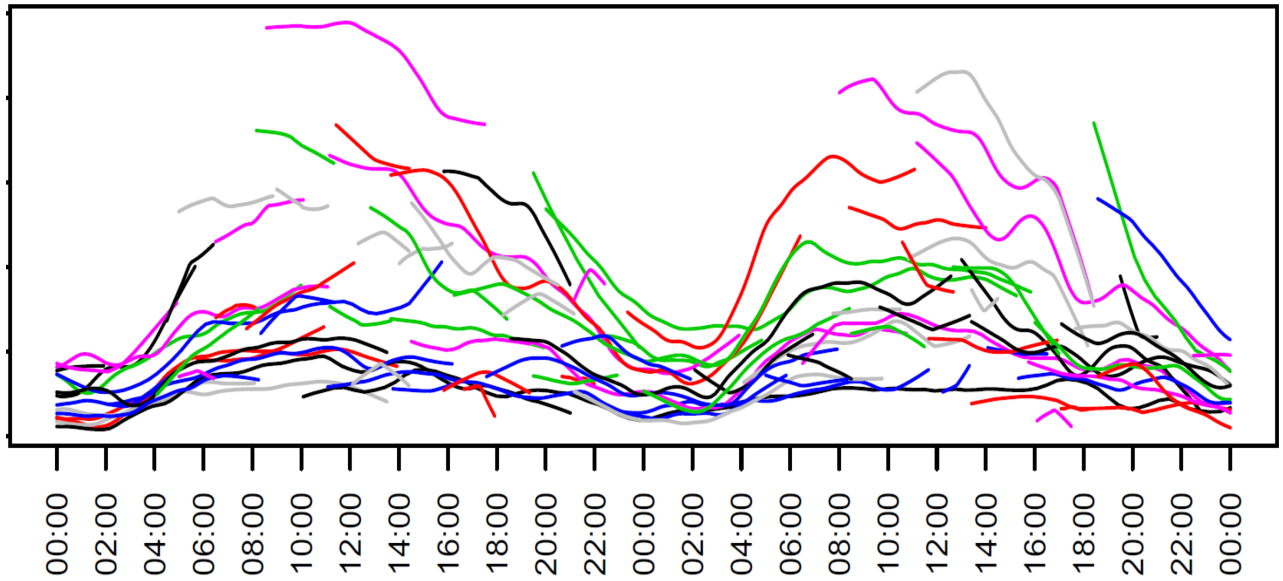
YAHOO!

Challenge: Scale

- 150 PB data on Yahoo Hadoop clusters
- Science projects need to be executed on Hadoop clusters directly
  - ✳Avoid latency caused by data movement
  - ✳Leverage existing 35,000 servers for computation

Challenge: Speed

- Ex. Item CTR in Yahoo homepage Today Module
  - * Short Lifetimes
  - * Temporal effect
  - * Break news

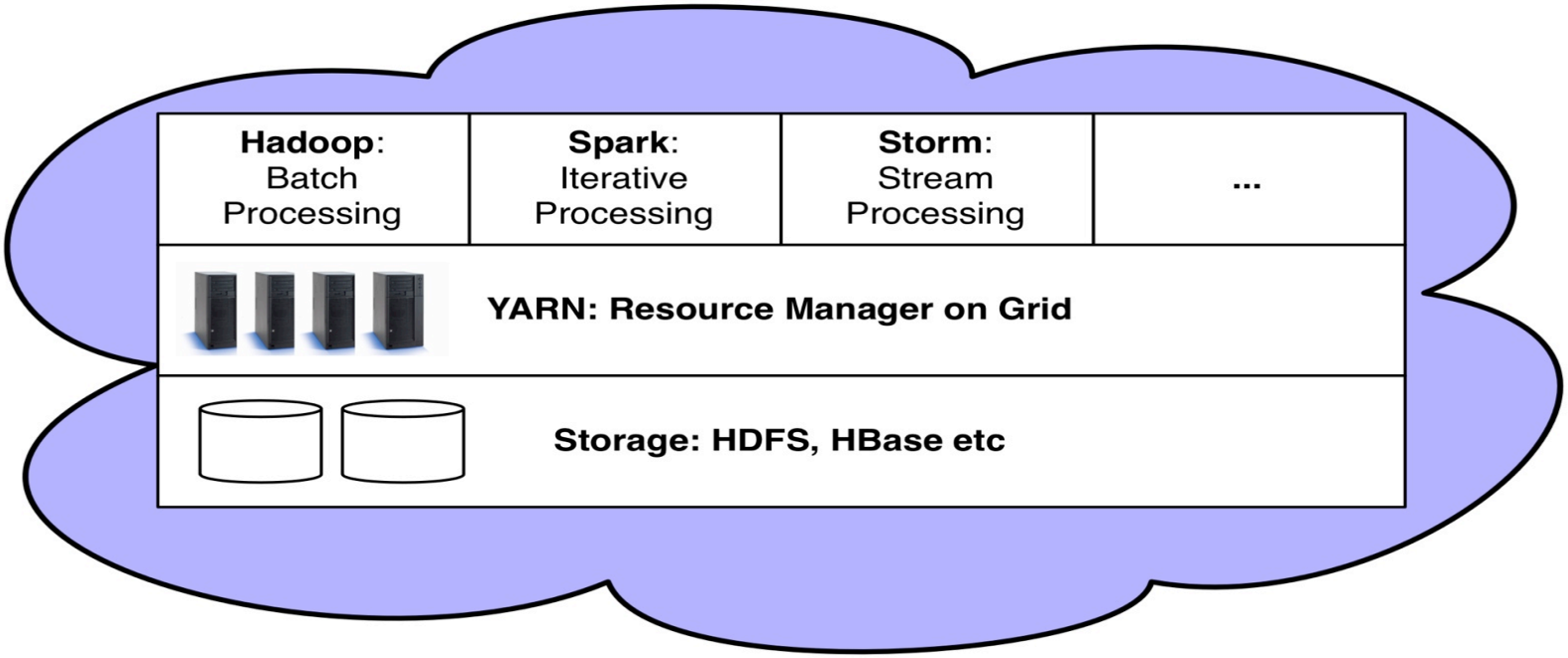


✳Update user/item knowledge continuously

✳Enable item discovery within minutes

YAHOO!

Yahoo Big-Data Platform

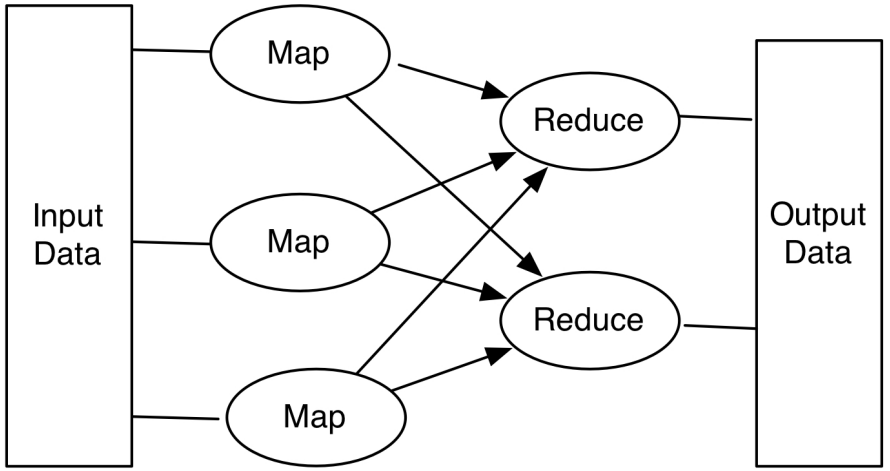


YAHOO!

Apache Hadoop

http://wiki.apache.org/hadoop/

- Popular framework for running applications on large cluster built of commodity hardware
- Implement Map/Reduce and Tez with YARN resource manager
- Provides HDFS to store data on the compute nodes
- Designed for very high throughput and reliability

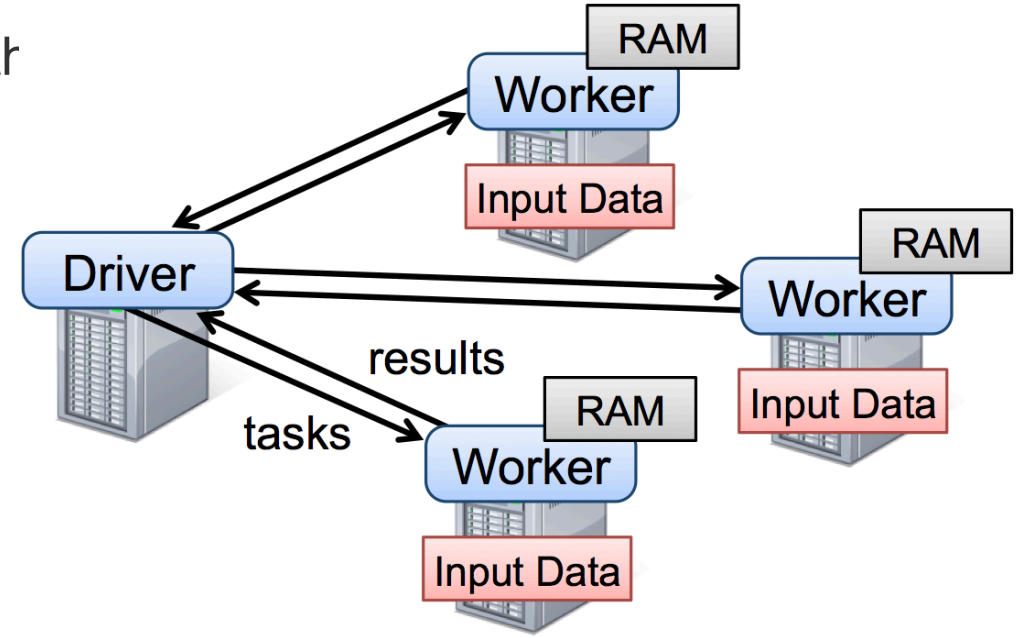


YAHOO!

Apache Spark

http://spark.incubator.apache.org

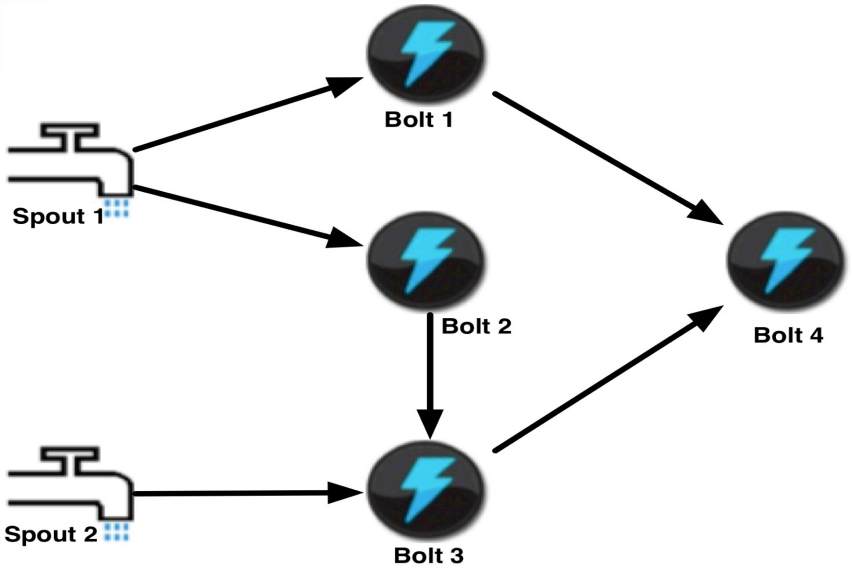
- Fast and expressive cluster computing system compatible with Apache Hadoop
- Support general execution DAGs
  - Include iterative programming
- Resilient distributed datasets (RDDs)
  - In-memory storage



YAHOO!

Apache Storm

http://storm.incubator.apache.org



- "Hadoop for Realtime"
  - a distributed, fault-tolerant, and high-performance realtime computation system that provides strong guarantees on the processing of data
- Emerging standard
  - Initially developed and deployed at BackType in 2011
  - Open sourced via github.com in Sept. 2011
  - Apache incubation in Sept. 2013, and Storm 0.9.1-incubating released in Feb 2014
  - 9 committers, 50 contributors, 12,000 readers

Hadoop vs. Spark vs. Storm

- Hadoop
  - Batch computing at data location; Excellent throughput
  - Massively scalable
- Spark
  - Batch computing at data location; Excellent throughput
  - Enable iterative applications
  - RDDs w/ caching option
- Storm
  - Realtime computing; Low-latency

Sample ML Libraries

- *Vowpal Wabbit*
  - A fast out-of-core learning system
  - Several optimization algorithms available including sparse gradient descent (GD), BFGS and several loss functions
- *LibSVM*
  - *A popular* library for Support Vector Machines
  - Support vector classification & regression

6 Design Patterns

- Batch/incremental learning
  1. ML in Hadoop Reducers only (Hadoop Streaming)
  2. ML in Hadoop Mappers only (Hadoop Streaming)
  3 Spark Pipe to external ML processes
  4 Spark native ML
- Real-time learning
  5. Storm Bolt w/ ML in JNI
- Hybrid learning
  6. ML in Hadoop/Spark and Storm

1. ML in Hadoop Reducers

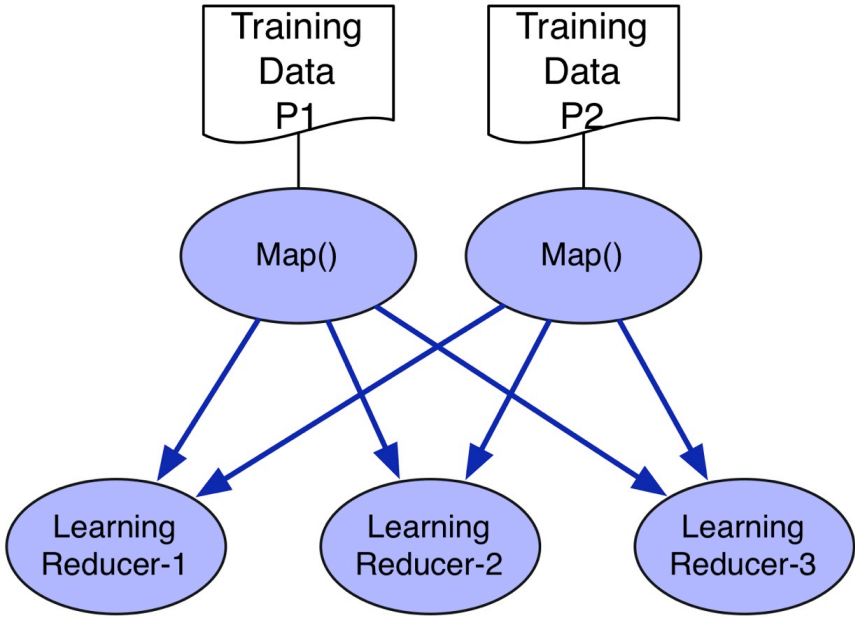
- **Basic Requirements**
  - 100's - 1000's models
  - Training data for each model could be loaded into a single machine
- **Solution: 1 reducer per model**
  - hadoop jar hadoop-streaming.jar

    -Dmapreduce.job.reduces=$num_models

    -reducer "vw --passes 20 --cache_file …"
  - hadoop jar lib/hadoop-streaming.jar

    -D mapreduce.job.reduces=$num_models

    -reducer "svm_train_reducer.py …"



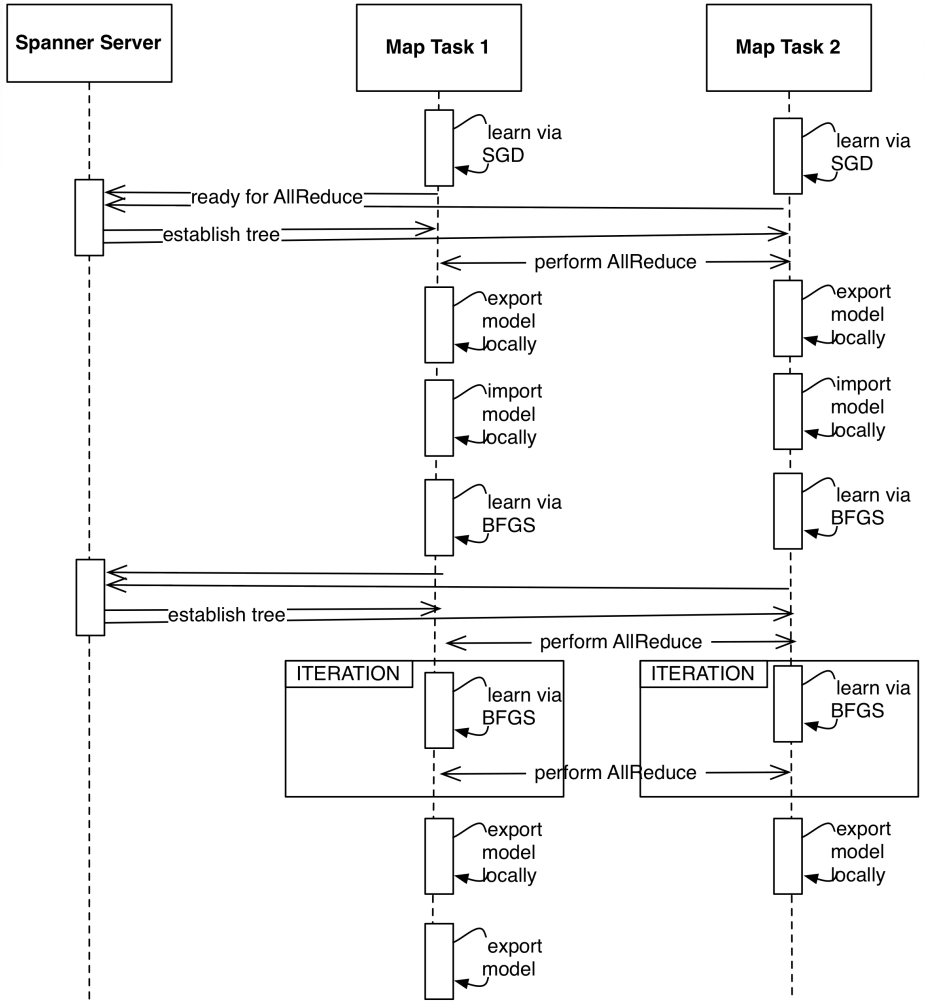
YAHOO!

2. ML in Hadoop Mappers

- **Basic Requirements**
  - Small # of models to be trained
  - Training data are too large to be loaded into a single machine

- **Solution: Mappers + MPI AllReduce**
  1. spanning_tree
  2. hadoop jar hadoop-streaming.jar

  -input $training_data -output $model_loc
  -Dmapreduce.job.maps=$num_mappers
  -mapper "runvw.sh $model_location
  $span_server $num_mappers"
  -reducer NONE

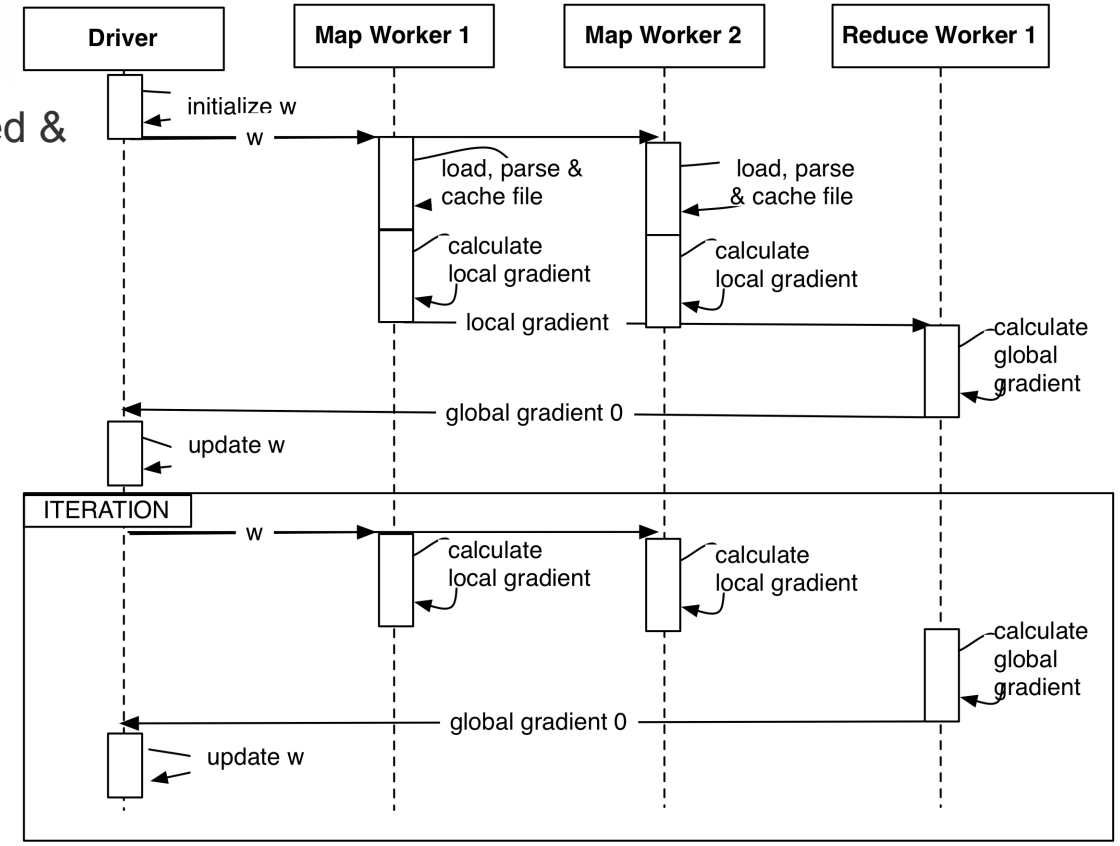


3. Spark Pipe to External ML Processes

- **Basic Requirements**
  - Small # of models built via distributed & iterative computing
  - Hadoop's job launch overhead is significant

- **Solution: Spark RDD Cache & Pipe**

```
val data= ctx.textFile(...).map(…).cache()
var w=data.pipe(GD_cmd).reduce(Avg)
for (i <- 1 to ITERATIONS) {
   val gradient = data.pipe(BFGS_cmd)
            .reduce((a,b) => a+b)
    w -= gradient
}
```



YAHOO!

4. Spark Native ML

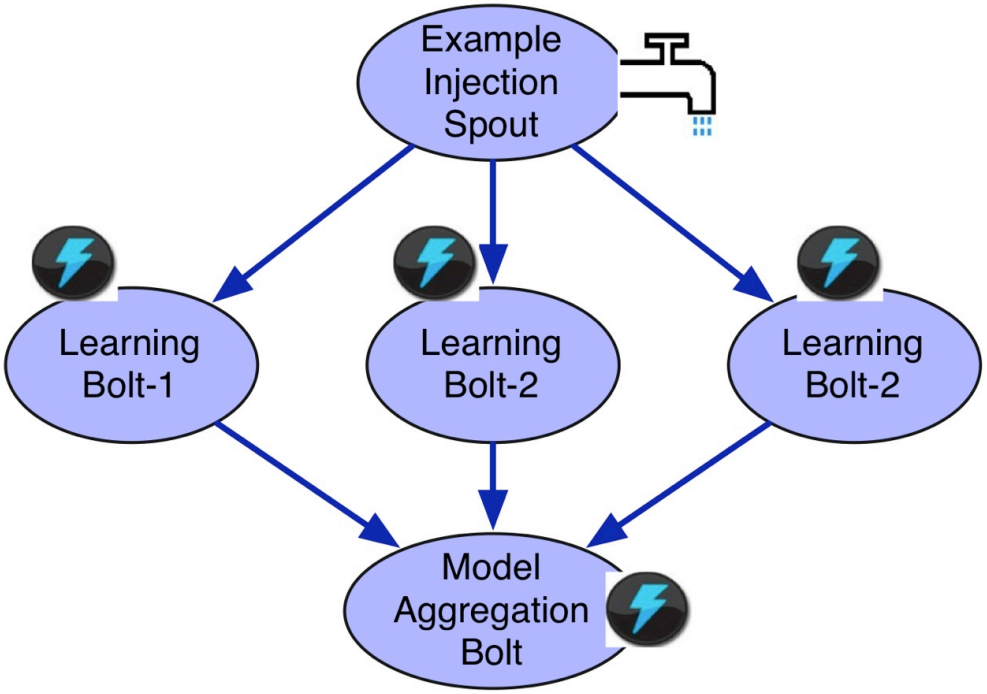
- **Basic Requirements**
  - Don’t need advanced ML features as in VW or LibSVM

- **Sample Solution – Spark or MlLib**
  - Spark
    - Yahoo E-Commerce: 30 LOC Spark program for collaborative filtering
  - MlLib (http://spark.incubator.apache.org/docs/0.9.0/mllib-guide.html)
    - Binary classification … logistic regression; Support Vector Machines
    - Linear regression … Linear regression; Ridge Regression; Lasso
    - collaborative filtering … Alternating Least Squares (ALS) matrix factorizaton
    - Clustering … K-means clustering

5. Realtime ML in Storm Bolts

- **Basic Requirements**
  - Freshness of ML model is critical
- **Sample Solution**

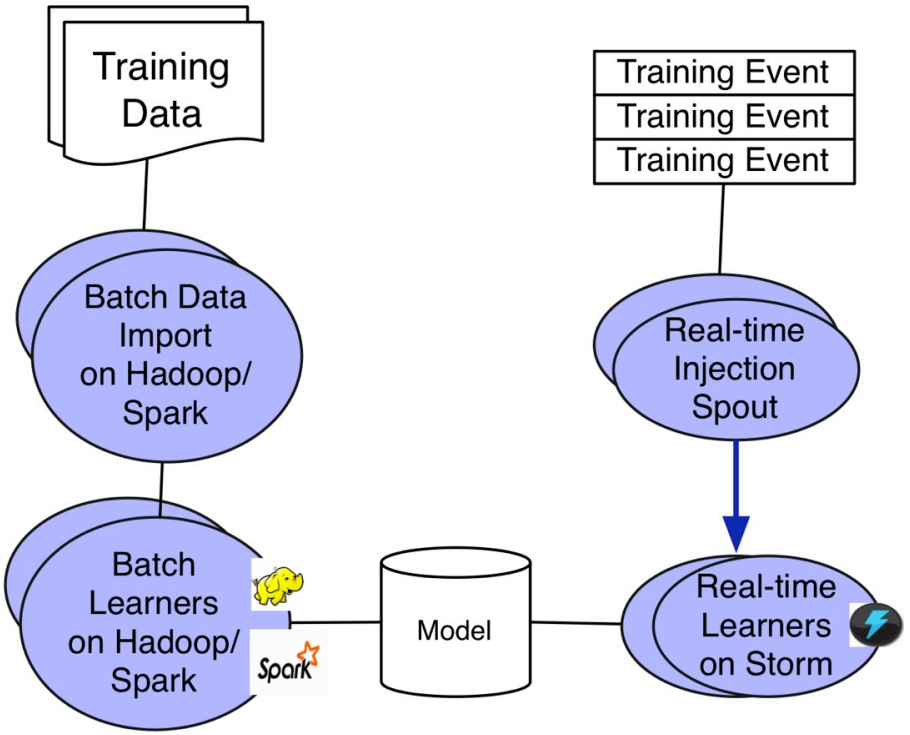
```
public class TrainingBolt extends BaseBasicBolt  {
  Model model;
  public void prepare(Map conf, TopologyContext ctx) {
    System.loadLibrary("VW");
    model =VW.init(conf);
   }
   public void execute(Tuple input, OutputCollector colle
    Instance example = input. getValue(0);
    model.learn(example);
    if (Time since last export)  collector.emit(model);
  }
}
```



YAHOO!

6. Hybrid Learning

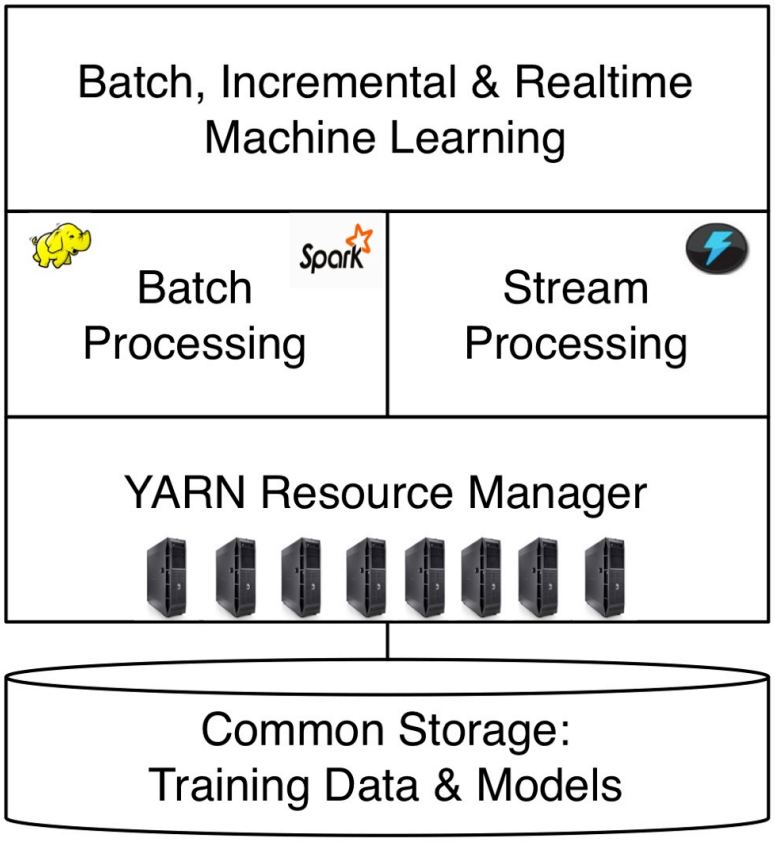
- **Basic Requirements**
  - Boostrape models via batch learning from large datasets
  - Update models via realtime learning from latest events
- **Sample Solution**
  - ML in Hadoop + Storm
  - ML in Spark + Storm



YAHOO!

Conclusion

- Hadoop, Spark and Storm empower variety of design patterns for machine learning from big-data.
- We expect active research and development of new ML capabilities on these big-data platform.



YAHOO!

We are Hiring!

Email: afeng@yahoo-inc.com

http://careers.yahoo.com

YAHOO!