Scalable Machine Learning at Yahoo

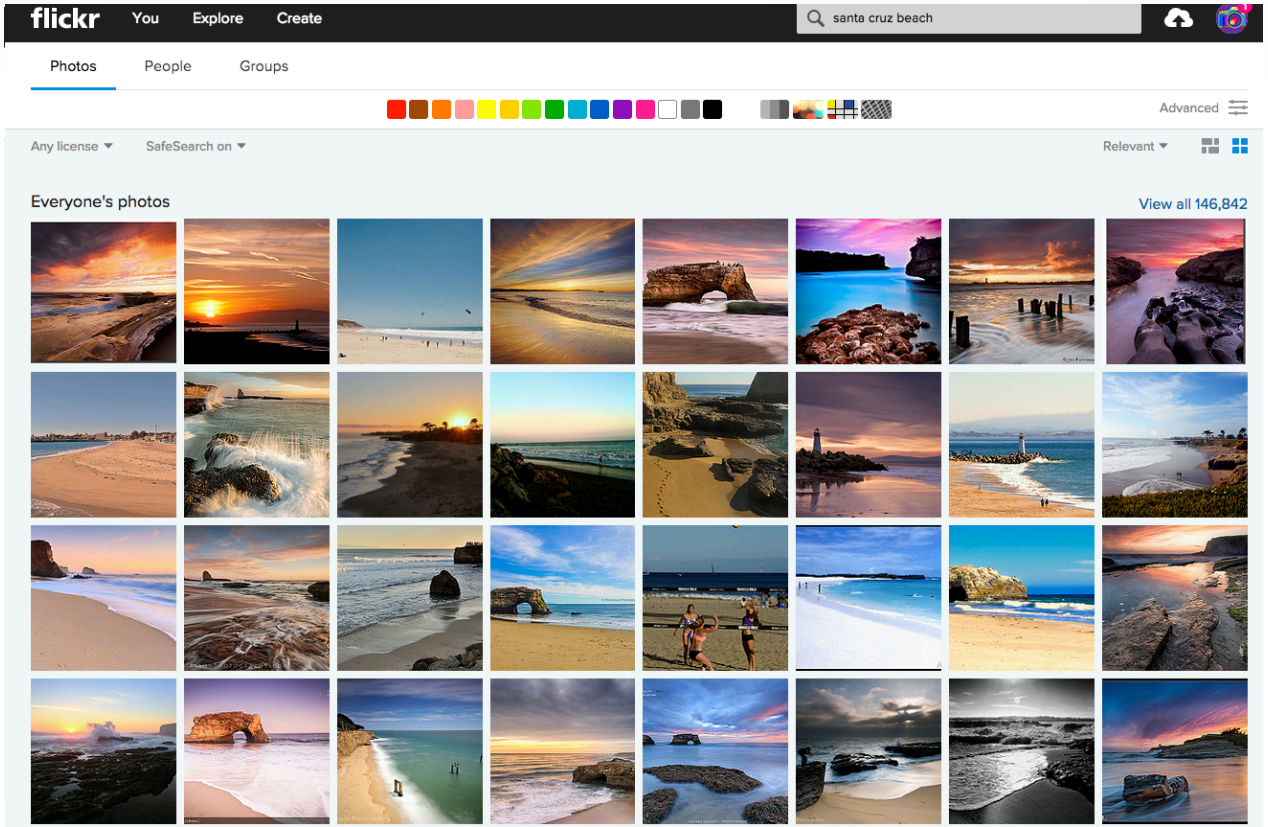
Andy Feng

VP Architecture, Yahoo

Agenda

- Use Cases
- Challenge
- Our Approach
- Algorithms
  1. Logistic Regression
  2. Word Embedding
  3. Deep Learning

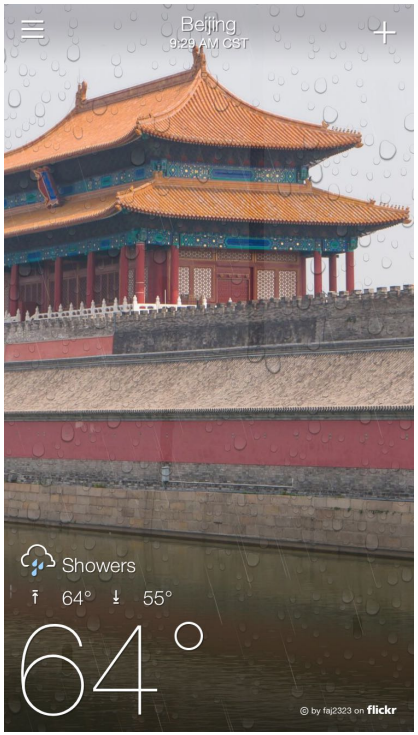
ML User Case: Flickr



- 1TB free storage
- 10 billion photos
  - 7.5 million per day
  - No tags for most photos
- Search: private/public
  - Computed tag
- Scene detection
  - Beach, Sunset, …
- Object recognition
  - Lighthouse, Sea arch, …

YAHOO!

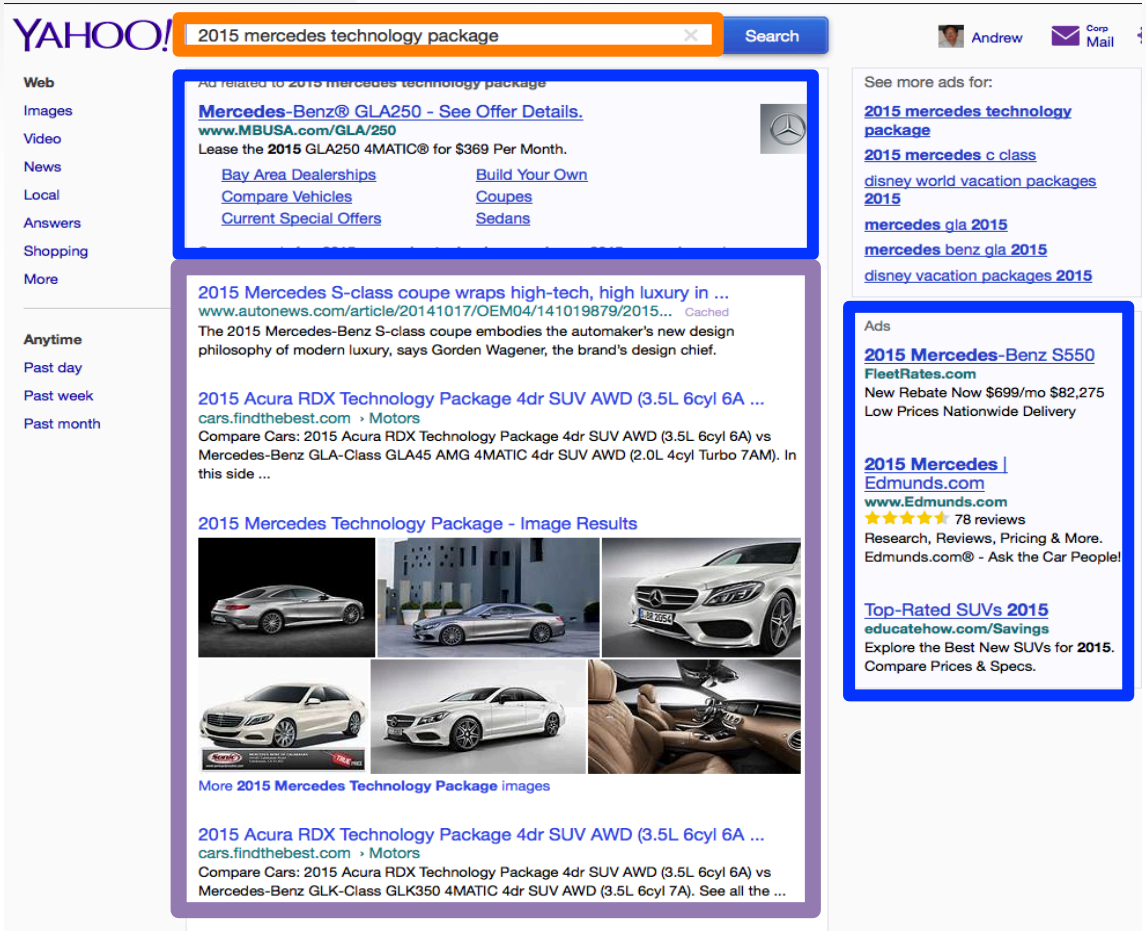
ML Use Case: Yahoo Weather



- Beauty
  - Computational assessed
- Relevant scene
  - Location
  - Time
  - Mostly cloudy
  - Partly cloudy
  - Shower
  - …

YAHOO!

ML Use Case: Search



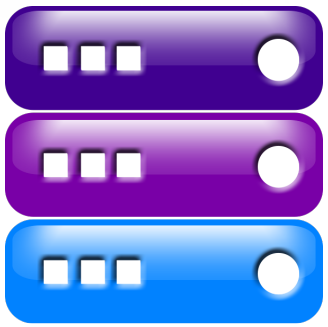
- Intention understanding
- Content ranking
- Query-Ads matching
- Ad click prediction

YAHOO!

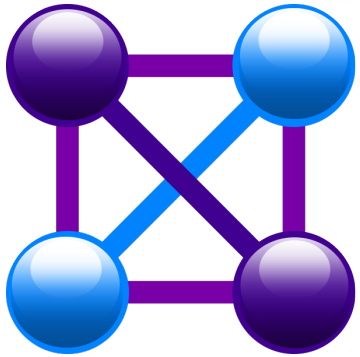
ML Challenge: Scale

1. Massive amount of examples
   - Naïve solutions take days/weeks
2. Billions of features
   - Model exceeds memory limits of 1 computer
3. Variety of algorithms
   - Different solutions required for scale-up

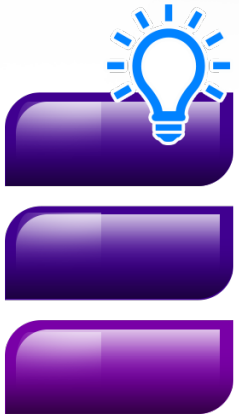
Massive Hadoop at Yahoo



600 PB
HDFS



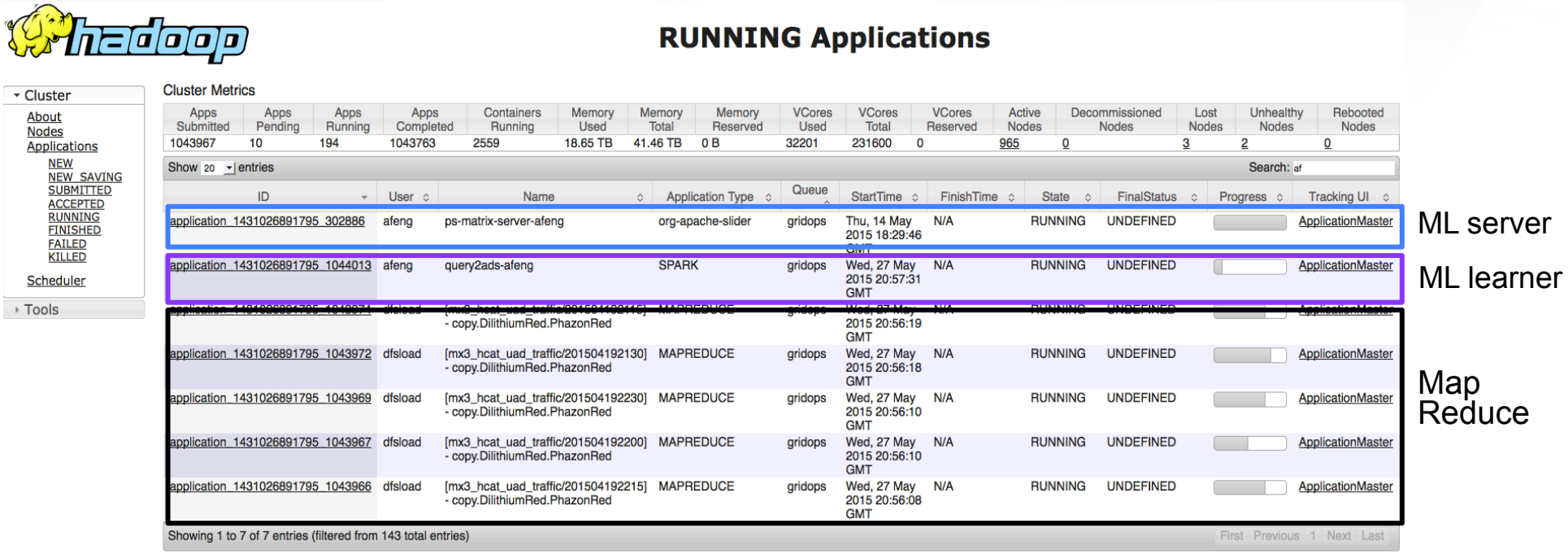
40K
Computers



MACHINE
LEARNING

YAHOO!

Scalable ML in Action



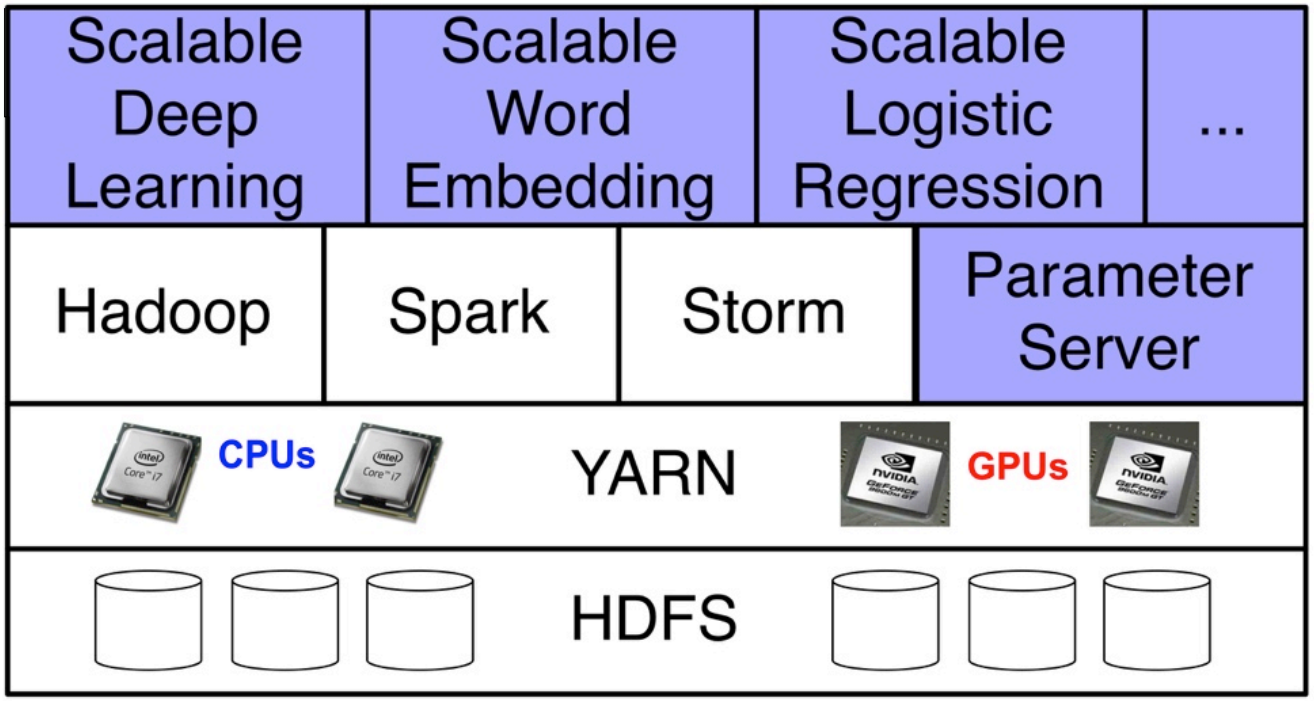
ML server

ML learner

Map Reduce

YAHOO!

Architecture for Scalable ML



Examples of ML Algorithms

1. Logistic Regression
   - Problem: Model size
   - Solution: Spark + Parameter Server
2. Ad-Query Vectors
   - Problem: Model size + Training latency
   - Solution: Spark + Parameter Server
3. Deep Learning
   - Problem: Training latency
   - Solution: Spark + GPU + MPI

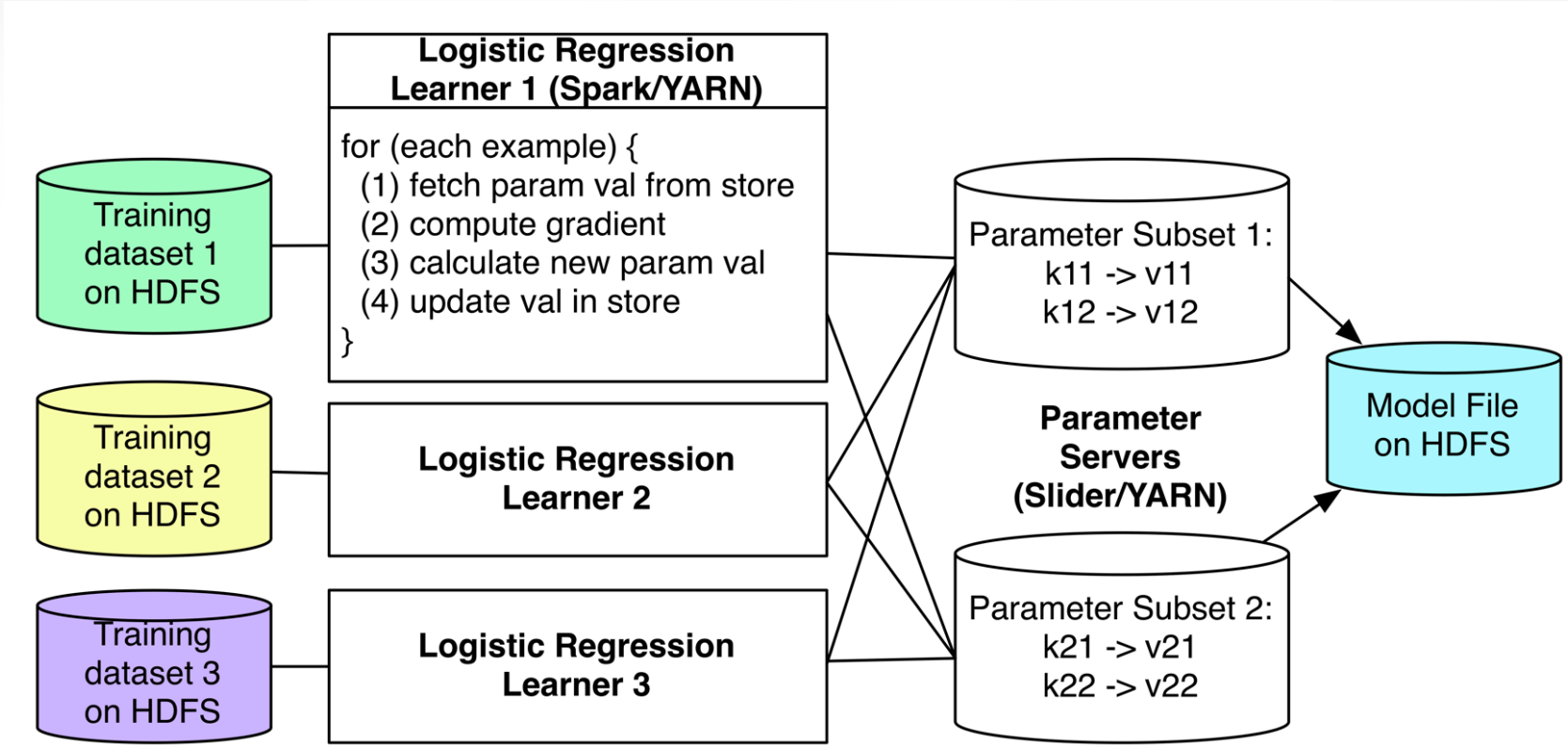
Algorithm 1: Logistic Regression

$$p(X_1, X_2, \ldots, X_p) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p)}}$$

When |β| > 100B,

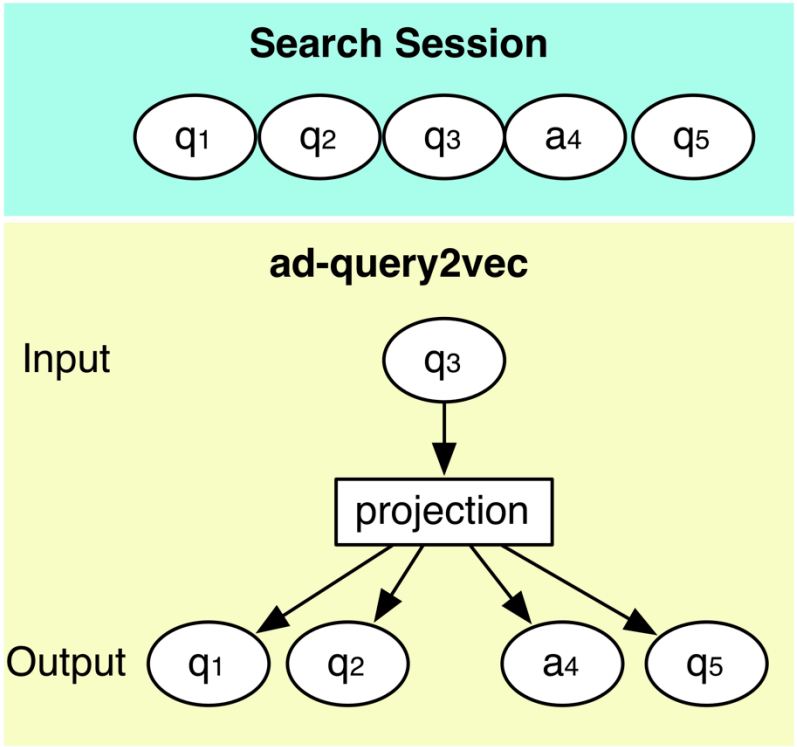
- 100 Billion * 16 Bytes = 1.6 TB
- β exceeds memory limit of 1 computer

Logistic Regression: 1000x Scale-up



YAHOO!

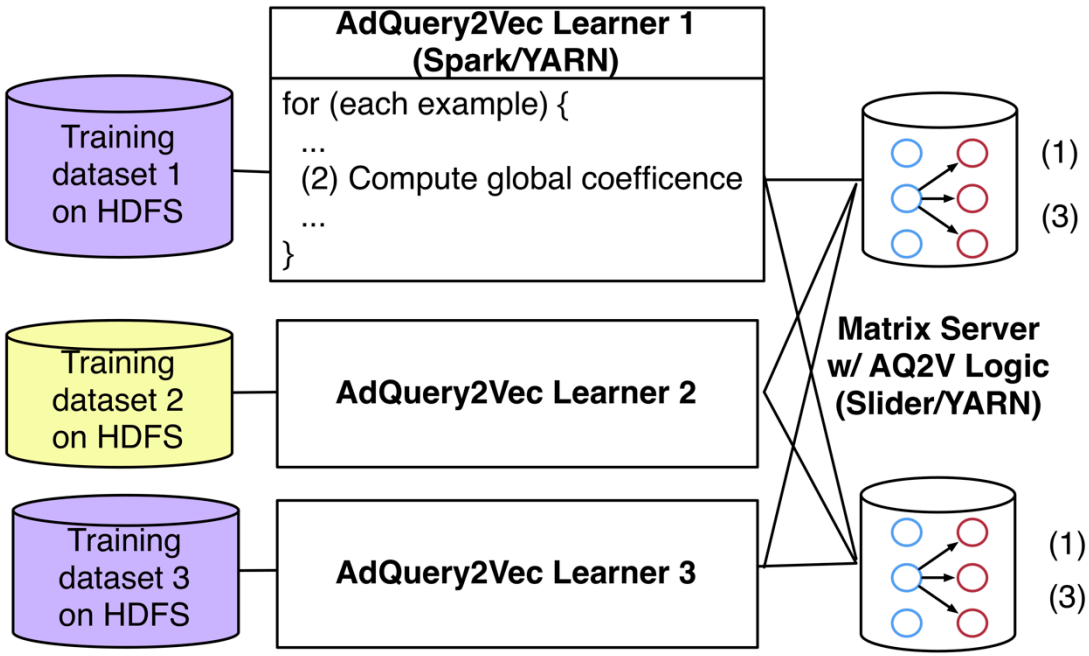
Algorithm 2: Ad-Query Vectors



- Vector: numeric representation of queries/ads
  - Vector("san jose weather") ≈ Vector("weather 95113") ≈ Vector(ad123)
- Model size
  - 1 Billion* 300 dimensions = 2.4TB
- Vector computation (X*Y, aX+Y)
  - Took weeks for small datasets

* Yahoo Labs: http://bit.ly/1G3f6L2

Ad-Query2Vec: 100x Speed/Scale-up



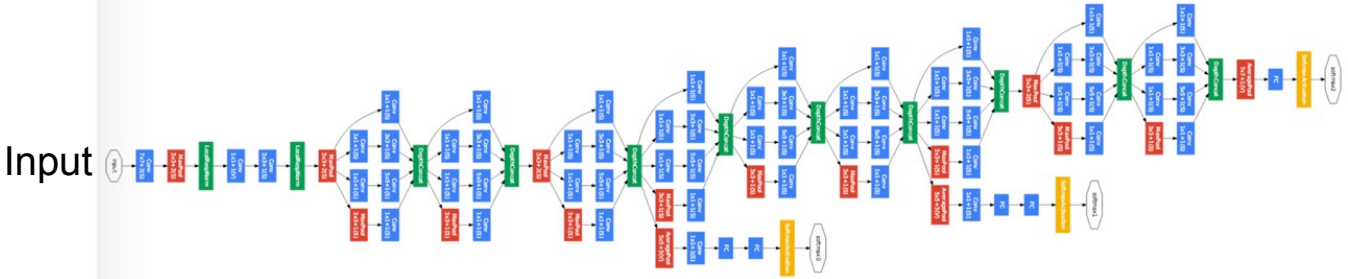
- Computation on servers
  - (1) Negative sampling
  - (1) Compute gradient: X*Y
  - (3) Adjust vectors: Y=aX+Y
- Daily training enabled
  - weeks ➔ hours

YAHOO!

Algorithm 3: Deep Learning

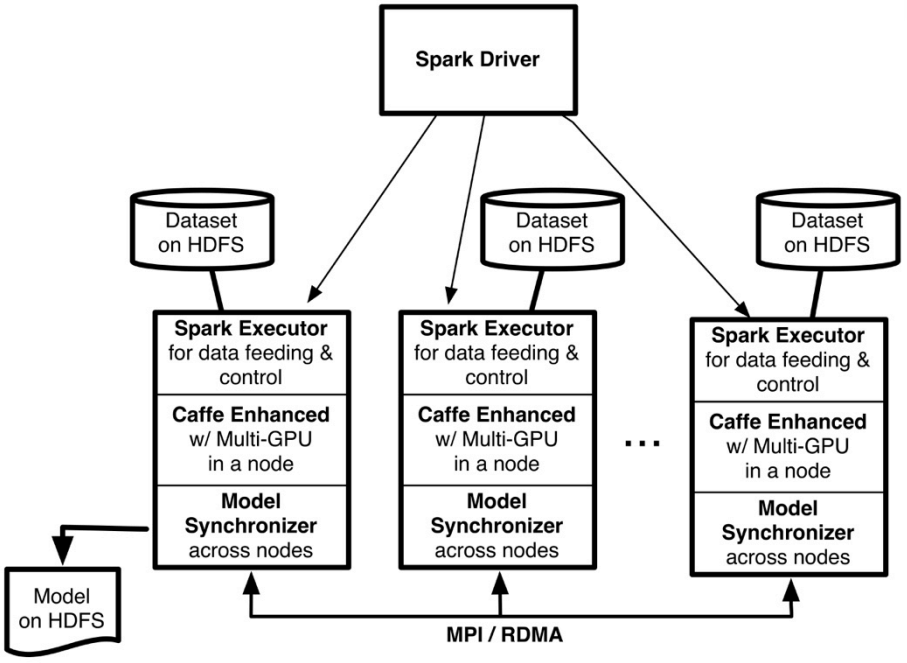
- ILSVRC classification
  - Classify images to one of 1,000 categories
  - Dataset
    - 1.2 million images for training
    - 50,000 images for validation

- GoogLeNet: 2014 Winner
  - 22 layer deep network
    - Convolution, Pooling,…, Loss
  - Trained on Google DistBelief cluster (16,000 CPU cores)
- Training latency
  - **47 days** for 1 K40c GPU to achieve 88.9% top-5 accuracy http://bit.ly/1FtFMa4



YAHOO!

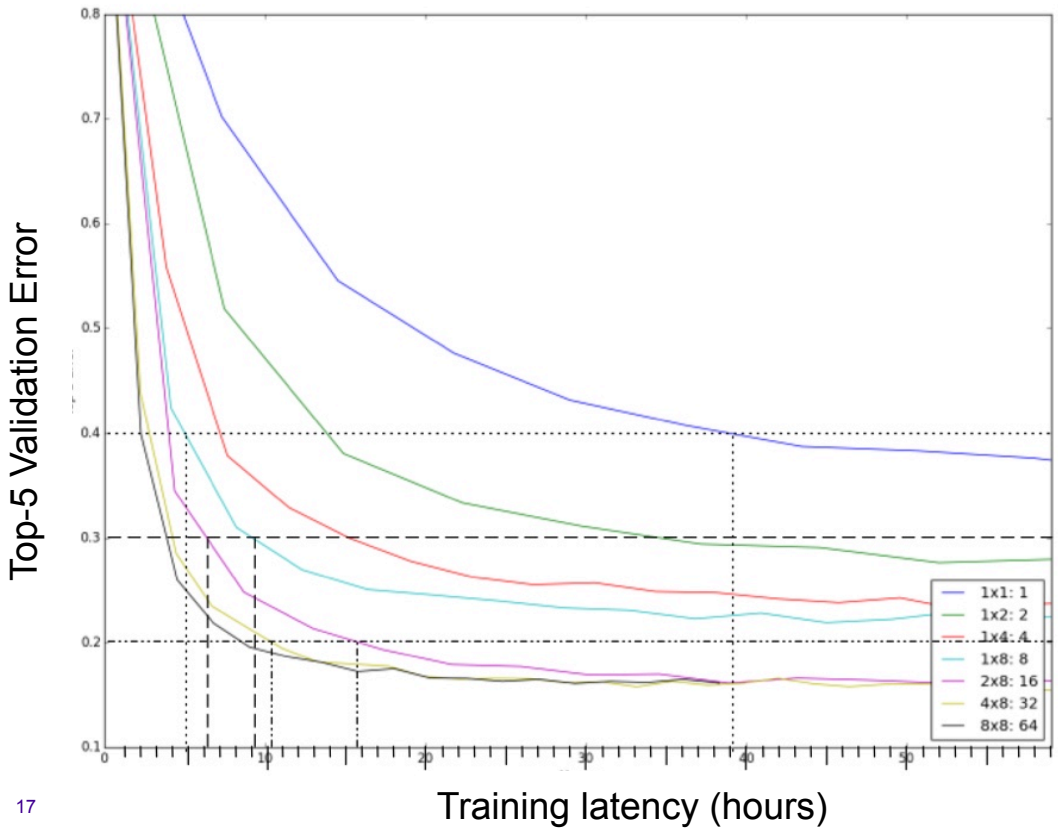
Deep Learning: Caffe-on-Spark with GPU



- GPU + Infiniband
  - 10x computing speedup over CPU
  - 10x network speedup over Ethernet
- Caffe-on-Spark
  - Apache Spark (spark.apache.org)
    - open source for cluster computing
  - Caffe (caffe.berkeleyvision.org)
    - Open source for deep learning

YAHOO!

Deep Learning: 19x Speedup



- 1x1 GPU
  - 39 hours … 60% top-5 accuracy
- 4x8 GPUs
  - 10 hours … 80% top-5 accuracy
  - 19x speedup estimated over 1x1 GPU
    - We are working on larger speedup

YAHOO!

Summary

- Machine learning is critical for Yahoo business
  - Search, advertisement, Flickr, weather, recommendation, security, …
- Scalable ML platforms built on Hadoop & Spark
  - Heterogeneous cluster: CPU + GPU
  - Approximate computing

Additional Info

- Scalable machine learning
  - http://bit.ly/1NW3GvO
  - http://bit.ly/1VedY0f
- Flickr use case
  - http://bit.ly/1KEytuV
  - http://bit.ly/1KIDfof
- Yahoo Hadoop blog
  - http://yahoohadoop.tumblr.com/

Thanks! We are hiring. bigdata@yahoo-inc.com

YAHOO!