

Scaling Machine Learning To Billions of Parameters

Badri Bhaskar, Erik Ordentlich
(joint with Andy Feng, Lee Yang, Peter Cnudde)
Yahoo, Inc.



SPARK SUMMIT 2016
DATA SCIENCE AND ENGINEERING AT SCALE

Outline

- Large scale machine learning (ML)
- Spark + Parameter Server
 - Architecture
 - Implementation
- Examples:
 - Distributed L-BFGS (Batch)
 - Distributed Word2vec (Sequential)
- Spark + Parameter Server on Hadoop Cluster



LARGE SCALE ML



Web Scale ML

Big Model

Billions of features

Big Data
Hundreds of billions of examples



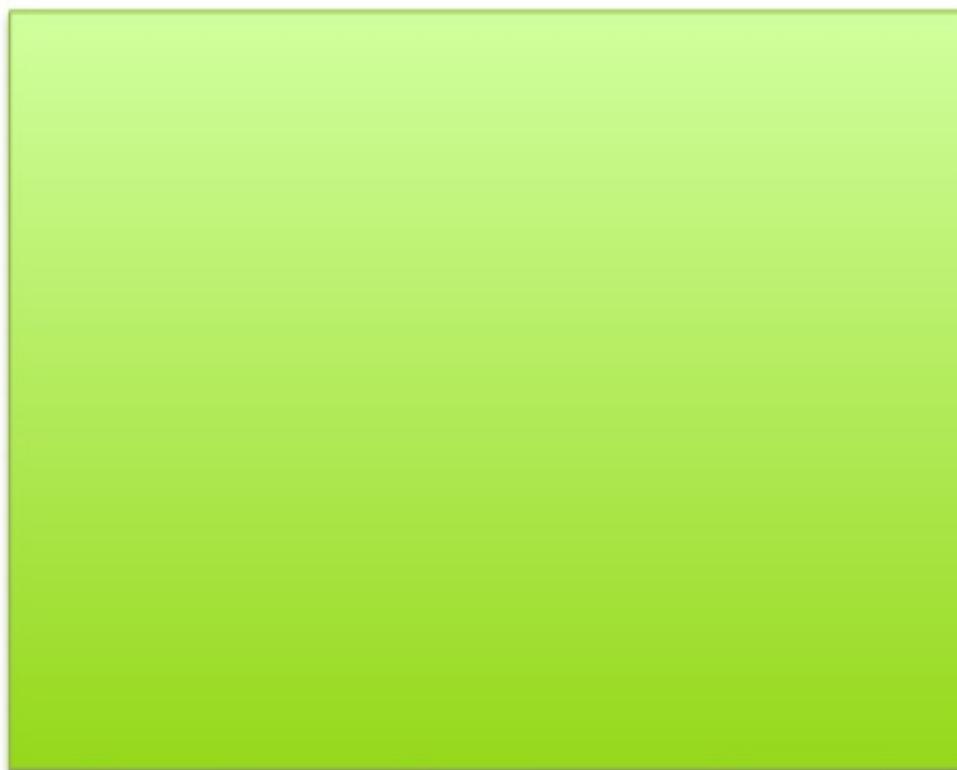
Ex: Yahoo word2vec - 120 billion parameters and 500 billion samples

Web Scale ML

Big Model

Billions of features

Big Data
Hundreds of billions of examples



Store

Store

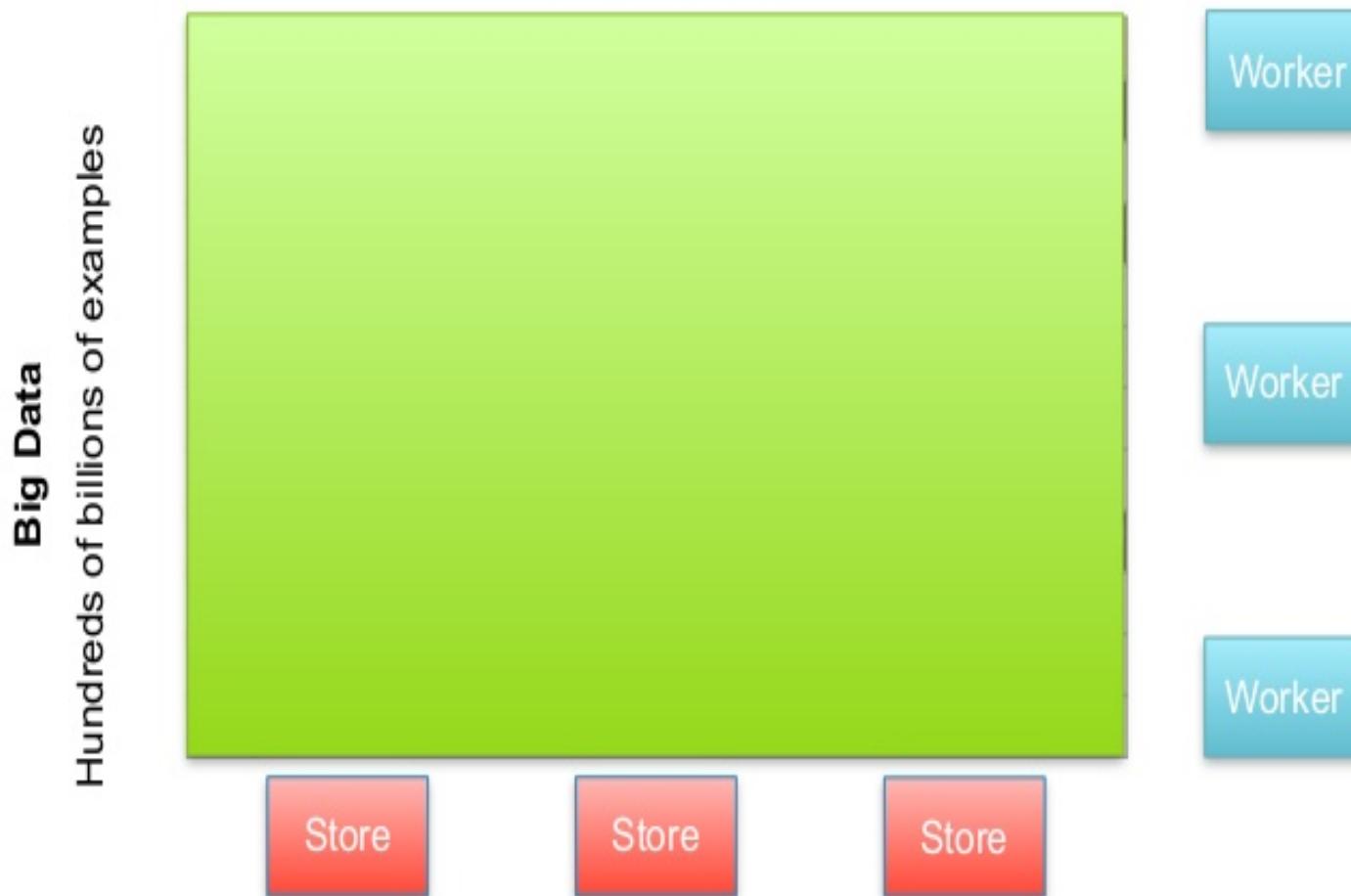
Store

Ex: Yahoo word2vec - 120 billion parameters and 500 billion samples

Web Scale ML

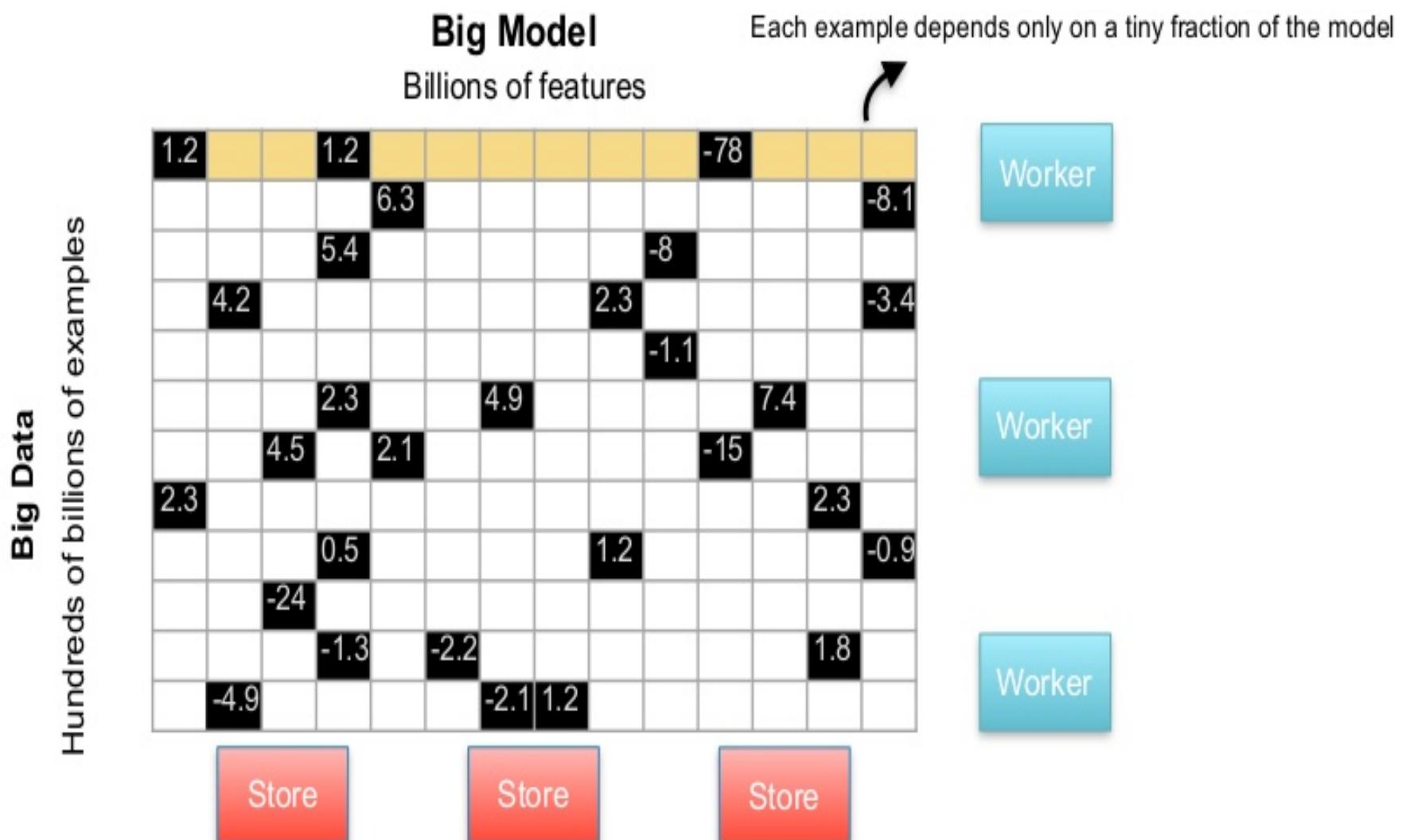
Big Model

Billions of features



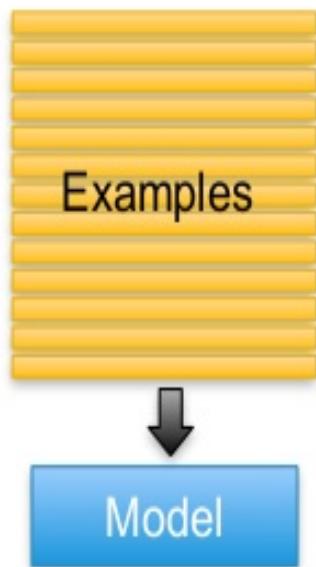
Ex: Yahoo word2vec - 120 billion parameters and 500 billion samples

Web Scale ML



Ex: Yahoo word2vec - 120 billion parameters and 500 billion samples

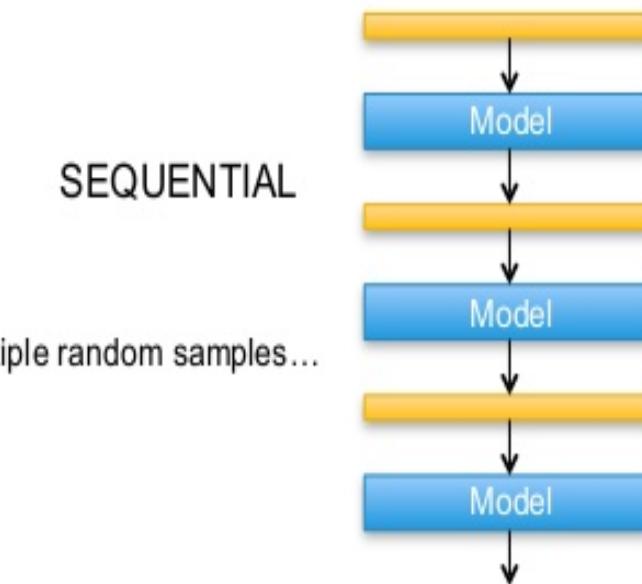
Two Optimization Strategies



BATCH

Multiple epochs...

Example: Gradient Descent, L-BFGS

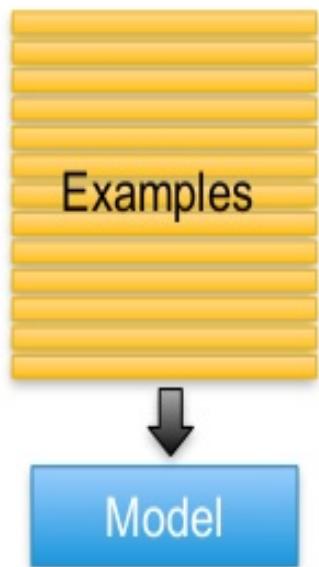


SEQUENTIAL

Multiple random samples...

Example: (Minibatch) stochastic gradient method,
perceptron

Two Optimization Strategies

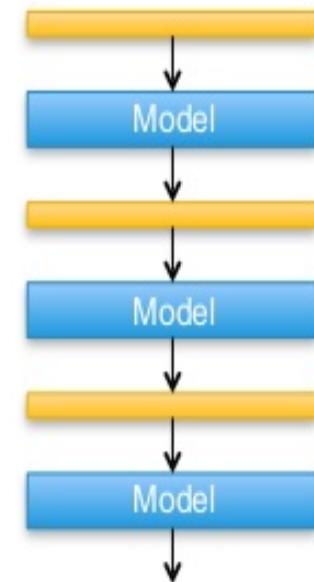


Example: Gradient Descent, L-BFGS

BATCH

SEQUENTIAL

Multiple random samples...

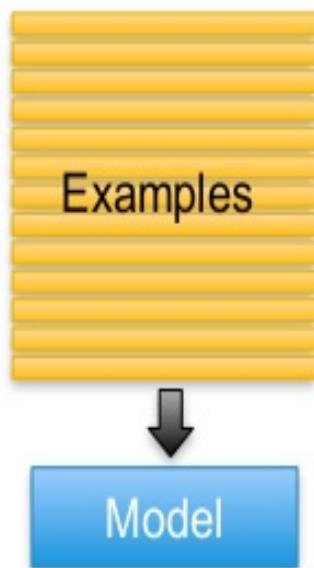


Example: (Minibatch) stochastic gradient method, perceptron

- Small number of model updates
- Accurate
- Each epoch may be expensive.
- *Easy to parallelize.*

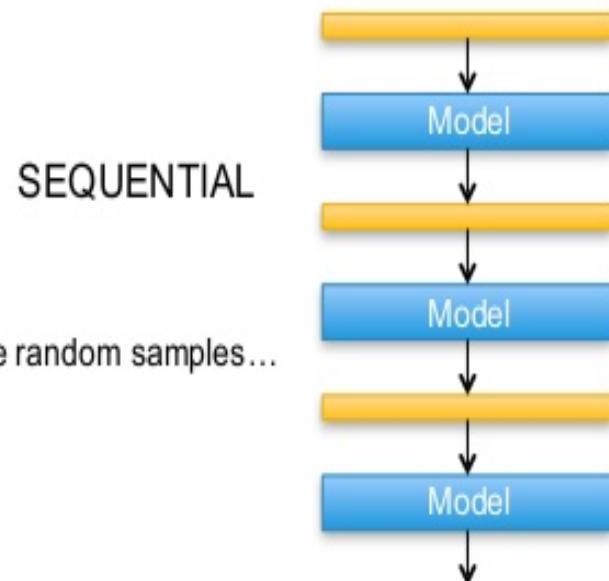


Two Optimization Strategies



Example: Gradient Descent, L-BFGS

- Small number of model updates
- Accurate
- Each epoch may be expensive.
- *Easy to parallelize.*



Example: (Minibatch) stochastic gradient method, perceptron

- Requires lots of model updates.
- Not as accurate, but often good enough
- A lot of progress in one pass* for big data.
- *Not trivial to parallelize.*



Requirements

Requirements

- ✓ Support both **batch** and **sequential** optimization

Requirements

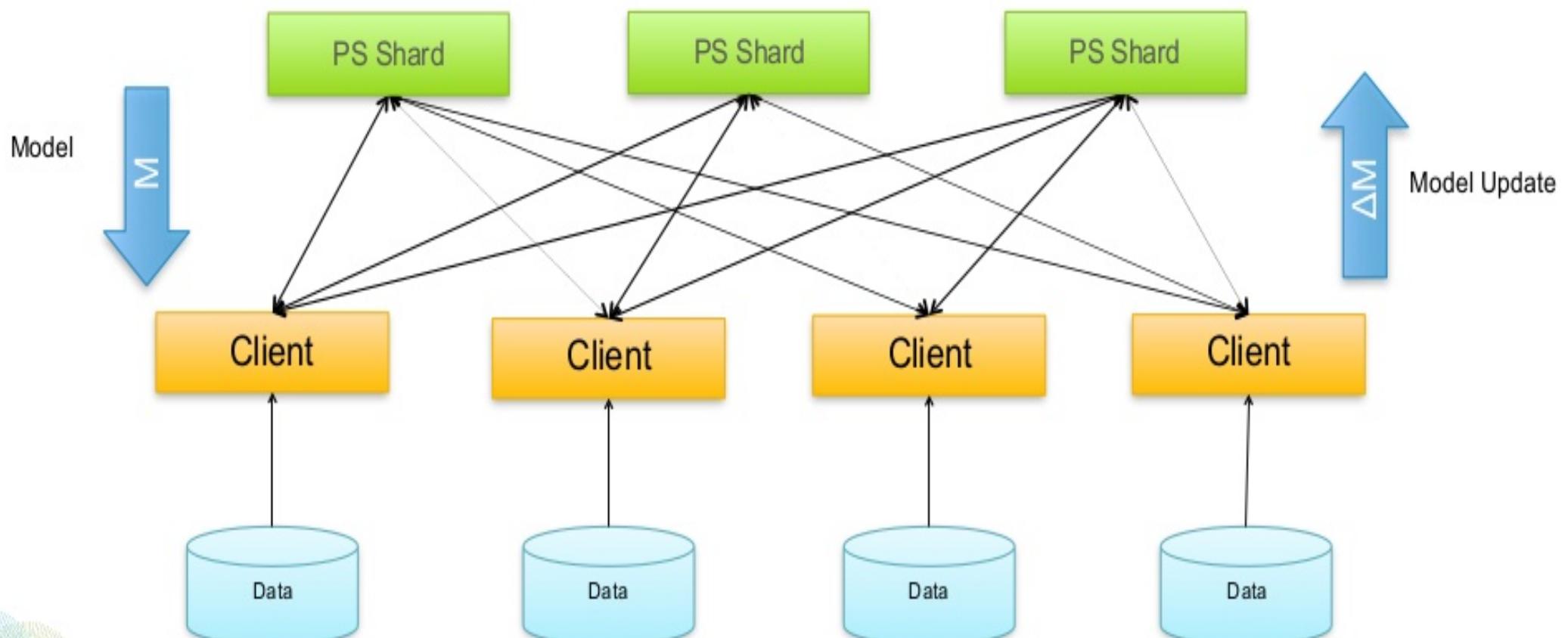
- ✓ Support both **batch** and **sequential** optimization
- ✓ **Sequential training:** Handle frequent updates to the model

Requirements

- ✓ Support both **batch** and **sequential** optimization
- ✓ **Sequential training:** Handle frequent updates to the model
- ✓ **Batch training:** 100+ passes each pass must be fast.

Parameter Server (PS)

Training state stored in PS shards, asynchronous updates

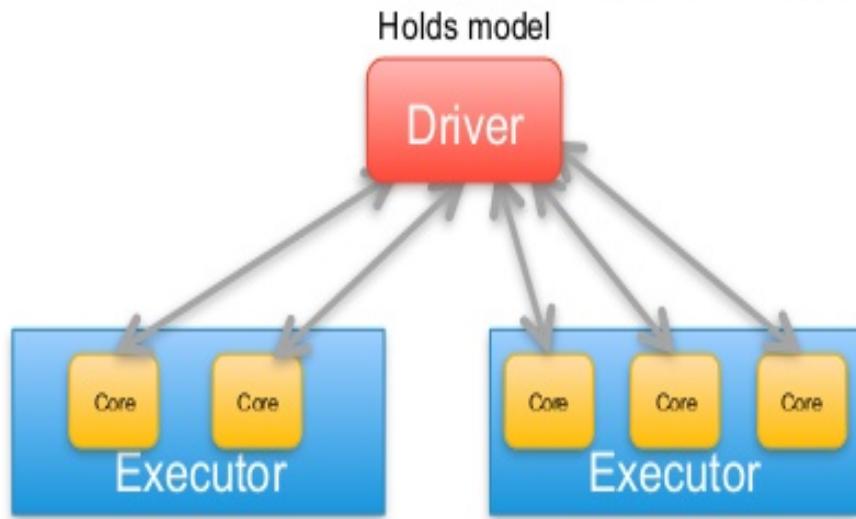


Early work: Yahoo LDA by Smola and Narayananurthy based on memcached (2010),

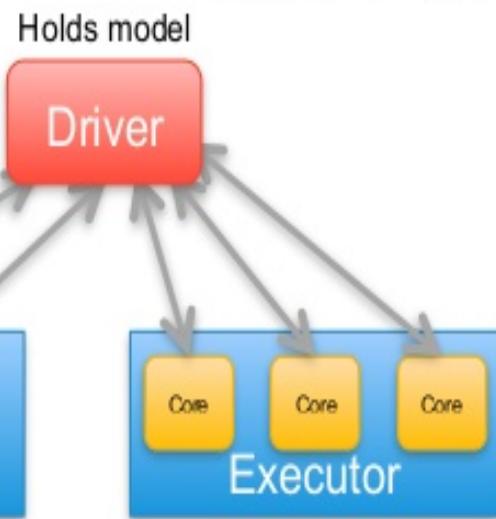
SPARK + PARAMETER SERVER



ML in Spark alone



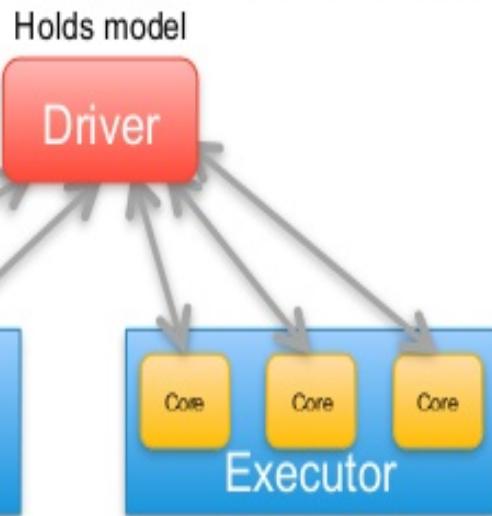
ML in Spark alone



```
def train(data: RDD[Example]) = {  
    while (not_converged) {  
        broadcast(model)  
        val cumGradient = data.sample().treeAggregate(...)  
        model.update(cumGradient)  
    }  
}
```

MLlib optimization

ML in Spark alone

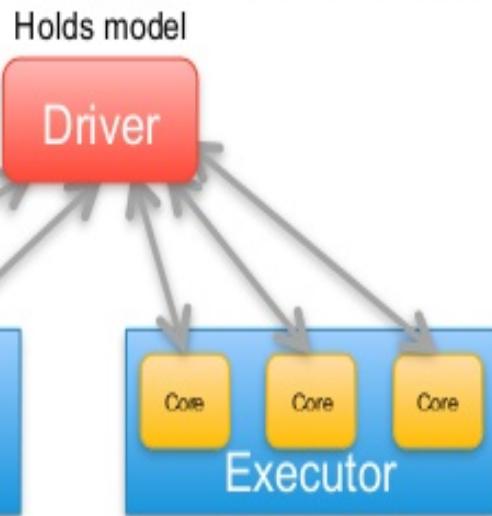


```
def train(data: RDD[Example]) = {  
    while (not_converged) {  
        broadcast(model)  
        val cumGradient = data.sample().treeAggregate(...)  
        model.update(cumGradient)  
    }  
}
```

MLlib optimization

- Sequential:
 - Driver-based communication limits frequency of model updates.
 - Large minibatch size limits model update frequency, convergence suffers.

ML in Spark alone

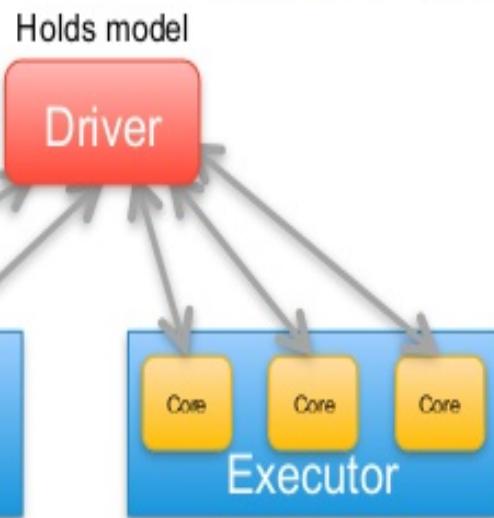


```
def train(data: RDD[Example]) = {  
    while (not_converged) {  
        broadcast(model)  
        val cumGradient = data.sample().treeAggregate(...)  
        model.update(cumGradient)  
    }  
}
```

MLlib optimization

- Sequential:
 - Driver-based communication limits frequency of model updates.
 - Large minibatch size limits model update frequency, convergence suffers.
- Batch:
 - Driver bandwidth can be a bottleneck
 - Synchronous stage wise processing limits throughput.

ML in Spark alone



```
def train(data: RDD[Example]) = {  
    while (not_converged) {  
        broadcast(model)  
        val cumGradient = data.sample().treeAggregate(...)  
        model.update(cumGradient)  
    }  
}
```

MLlib optimization

- Sequential:
 - Driver-based communication limits frequency of model updates.
 - Large minibatch size limits model update frequency, convergence suffers.
- Batch:
 - Driver bandwidth can be a bottleneck
 - Synchronous stage wise processing limits throughput.

PS Architecture circumvents both limitations...

Spark + Parameter Server

Spark + Parameter Server

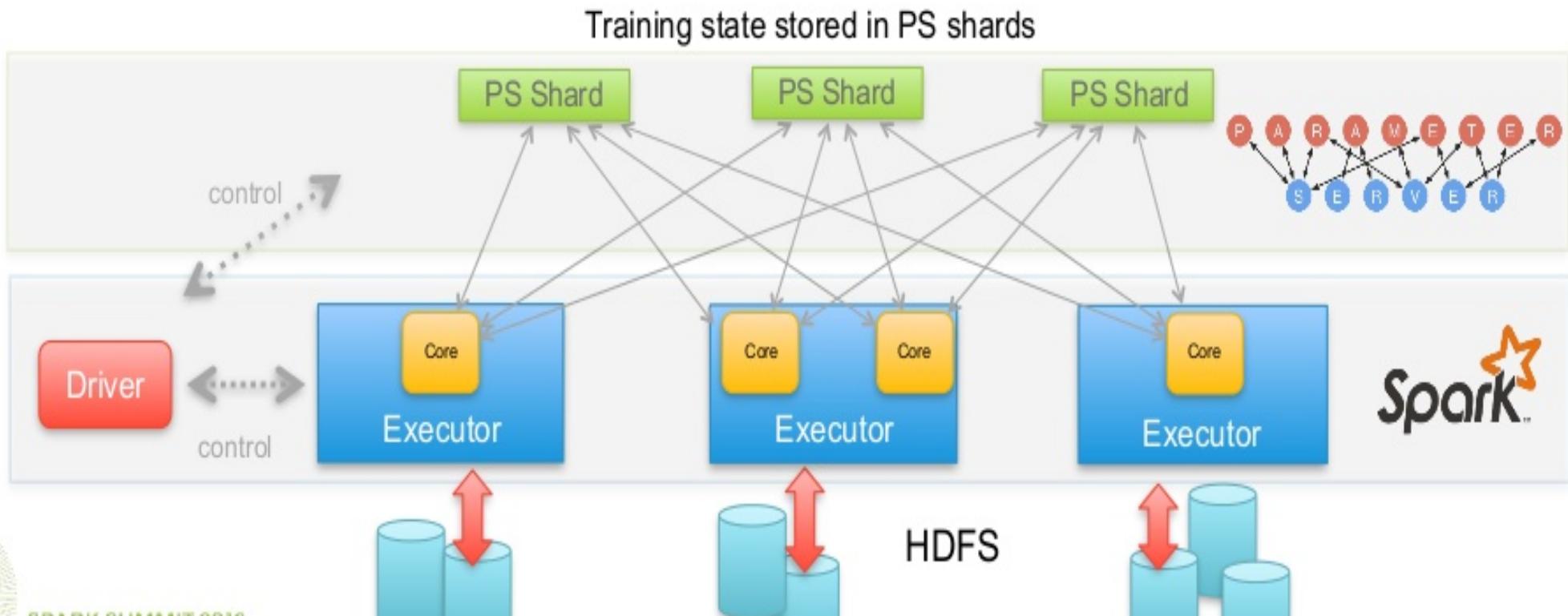
- Leverage Spark for HDFS I/O, distributed processing, fine-grained load balancing, failure recovery, in-memory operations

Spark + Parameter Server

- Leverage Spark for HDFS I/O, distributed processing, fine-grained load balancing, failure recovery, in-memory operations
- Use PS to sync models, incremental updates during training, or sometimes even some vector math.

Spark + Parameter Server

- Leverage Spark for HDFS I/O, distributed processing, fine-grained load balancing, failure recovery, in-memory operations
- Use PS to sync models, incremental updates during training, or sometimes even some vector math.



Yahoo PS

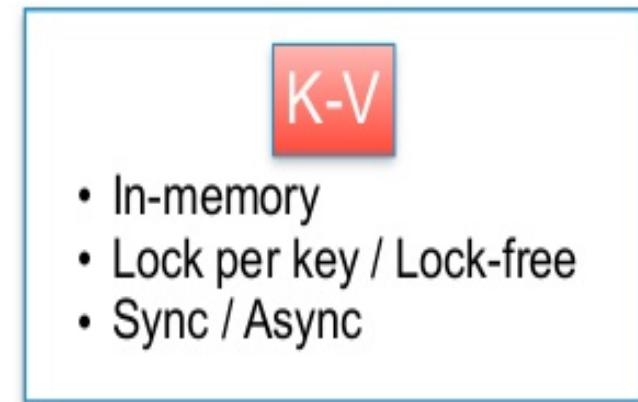
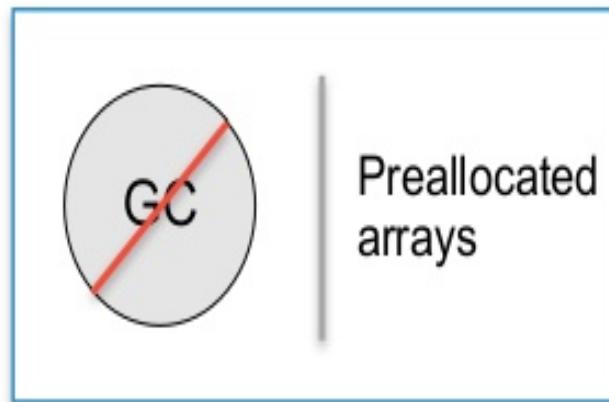
Yahoo PS



Yahoo PS



Yahoo PS



Yahoo PS



Server

Client API



Preallocated arrays



- In-memory
- Lock per key / Lock-free
- Sync / Async

9	13	5	2
1	11	7	6
3	7	4	1
6	0	7	10

- Column-partitioned
- Supports BLAS

Yahoo PS



Server

Client API

9	13	5	2
1	11	7	6
3	7	4	1
6	0	7	10

- Column-partitioned
- Supports BLAS



Preallocated arrays



HDFS

- Export Model
- Checkpoint



- In-memory
- Lock per key / Lock-free
- Sync / Async

Yahoo PS



Server

Client API

9	13	5	2
1	11	7	6
3	7	4	1
6	0	7	10

- Column-partitioned
- Supports BLAS



Preallocated arrays



HDFS

- Export Model
- Checkpoint



- In-memory
- Lock per key / Lock-free
- Sync / Async

UDF

- Client supplied aggregation
- Custom shard operations

Map PS API

```
trait MapClient[K,V] {  
    def get(key: K) : Future[V]  
    def put(key: K, value: V) : Future[Unit]  
  
    def multiGet(keys: Seq[K]) : Future[Map[K,V]]  
    def multiPut(keyValue: Seq[(K, V)]) : Future[Int]  
  
    def mapReduce[T,U](zero: U, mapFunc: T => U, reduceFunc: (U,U) => U) : Future[U]  
}
```

- Distributed key-value store abstraction
- Supports batched operations in addition to usual get and put
- Many operations return a future – you can operate asynchronously or block

Matrix PS API

```
trait MatrixClient extends MapClient[Int, Array[Float]] {  
    def dot(x: Int, y: Int): Float  
    def scal(row: Int, factor: Float) : Future[Unit]  
    def axpy(a: Float, x: Int, y: Int) : Future[Unit]  
    def copy(to: Int, from: Int) : Future[Unit]  
    ...  
  
    def increment(x: Int, indices: Array[Int], values: Array[Int]) : Future[Unit]  
    def fetch(x: Int, indices: Array[Int]) : Array[Float]  
}
```

- Vector math (BLAS style operations), in addition to everything Map API provides
- Increment and fetch sparse vectors (e.g., for gradient aggregation)
- We use other custom operations on shard (API not shown)

EXAMPLES



Sponsored Search Advertising

The screenshot shows a Yahoo search interface with the query "Apache Spark". The results are filtered under the "Web" category. The first result is a sponsored ad from Udemy for an online class titled "Apache Spark Online Class - Master Essentials Of Apache Spark". It includes a star rating of 4.5 and a link to "spark.udemy.com". Below the ad, there are two more sponsored links: "iOS App Development" and "Android App Development", both associated with "Top Web Development Class" and "Top Development Courses". The second result is the official Apache Spark website, which is described as a fast and general engine for big data processing. The third result is the Apache Spark entry on Wikipedia, described as an open source cluster computing framework. The fourth result is a link to Databricks' page on Apache Spark.

YAHOO! Apache Spark X Search

Web Images Video News More Anytime

Also try: apache spark tutorial, apache spark architecture

Ad related to: Apache Spark

Apache Spark Online Class - Master Essentials Of Apache Spark.
www.Udemy.com/Apache_Spark
4.5 ★★★★☆ rating for udemy.com
Master Essentials Of Apache Spark. Enroll Today & Save 20% Off!

iOS App Development Top Web Development Class
Android App Development Top Development Courses

Apache Spark - Official Site
spark.apache.org ▾
Apache Spark is a fast and general engine for big data processing, with built-in modules for streaming, SQL, machine learning and graph processing.

Apache Spark - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Apache_Spark ▾
Apache Spark is an open source cluster computing framework. Originally developed at the University of California, Berkeley's AMPLab, the Spark codebase was later ...

What is Apache Spark | Databricks
databricks.com/spark ▾
The team that created Apache Spark founded Databricks in 2013. Apache Spark is 100% open source, hosted at the vendor-independent Apache Software Foundation.

Sponsored Search Advertising

The screenshot shows a Yahoo search interface. The search bar contains "Apache Spark". Below the search bar, there are filters: "Web" (which is selected), "Images", "Video", "News", "More", and "Anytime". A "Search" button is located to the right of the search bar. Below the filters, a "Also try:" section lists "apache spark tutorial" and "apache spark architecture". An "Ad related to: Apache Spark" section displays an advertisement for an "Apache Spark Online Class - Master Essentials Of Apache Spark." It includes a link to "www.Udemy.com/Apache_Spark", a 4.5-star rating, and a promotion for "Master Essentials Of Apache Spark. Enroll Today & Save 20% Off!". Below the ad, there are two more links: "iOS App Development" and "Android App Development" under the heading "Top Web Development Class", and "Top Development Courses".

Apache Spark - Official Site

spark.apache.org ▾

Apache Spark is a fast and general engine for big data processing, with built-in modules for streaming, SQL, machine learning and graph processing.

Apache Spark - Wikipedia, the free encyclopedia

en.wikipedia.org/wiki/Apache_Spark ▾

Apache Spark is an open source cluster computing framework. Originally developed at the University of California, Berkeley's AMPLab, the Spark codebase was later ...

What is Apache Spark | Databricks

databricks.com/spark ▾

The team that created Apache Spark founded Databricks in 2013. Apache Spark is 100% open source, hosted at the vendor-independent Apache Software Foundation.

cost per impression = advertiser bid × click probability

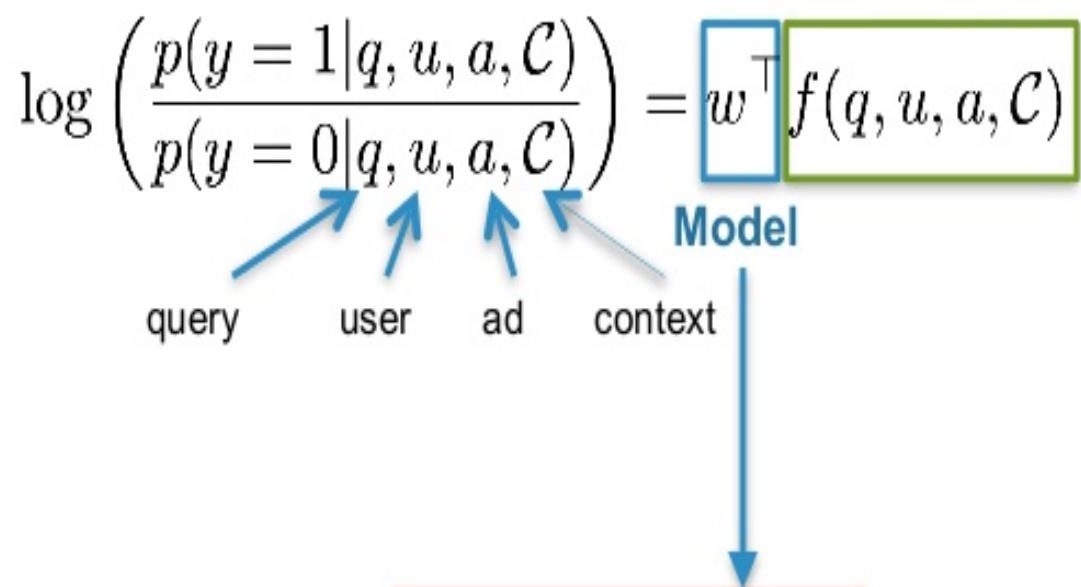


Sponsored Search Advertising

The screenshot shows a Yahoo search results page for the query "Apache Spark". The search bar at the top contains "Apache Spark". Below the search bar, there are filters: "Web" (which is selected), "Images", "Video", "News", "More", and "Anytime". A purple sidebar on the left lists "Also try: apache spark tutorial, apache spark architecture" and "Ad related to: Apache Spark". The main search results include:

- Apache Spark Online Class - Master Essentials Of Apache Spark.**
www.Udemy.com/Apache_Spark
4.5 ★★★★ rating for udemy.com
Master Essentials Of Apache Spark. Enroll Today & Save 20% Off.
iOS App Development Top Web Development Class
Android App Development Top Development Courses
- Apache Spark - Official Site**
spark.apache.org
Apache Spark is a fast and general engine for big data processing, with built-in modules for streaming, SQL, machine learning and graph processing.
- Apache Spark - Wikipedia, the free encyclopedia**
en.wikipedia.org/wiki/Apache_Spark
Apache Spark is an open source cluster computing framework. Originally developed at the University of California, Berkeley's AMPLab, the Spark codebase was later ...
- What is Apache Spark | Databricks**
databricks.com/spark
The team that created Apache Spark founded Databricks in 2013. Apache Spark is 100% open source, hosted at the vendor-independent Apache Software Foundation.

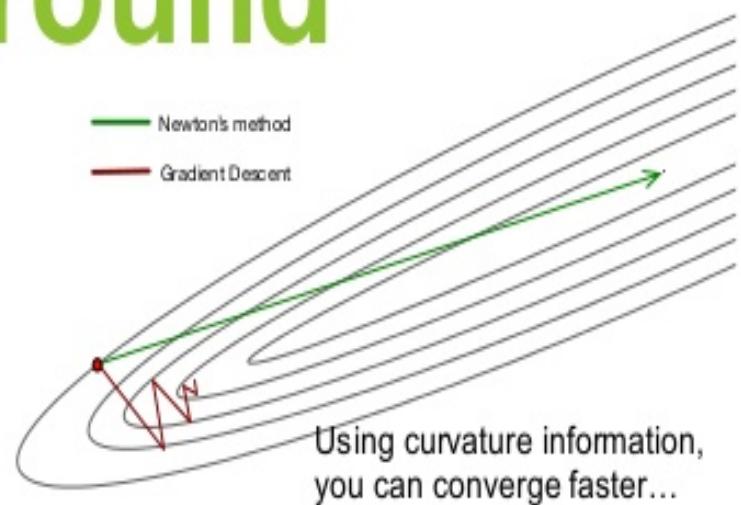
Example Click Model: (Logistic Regression) Features



cost per impression = advertiser bid × click probability

L-BFGS Background

L-BFGS Background



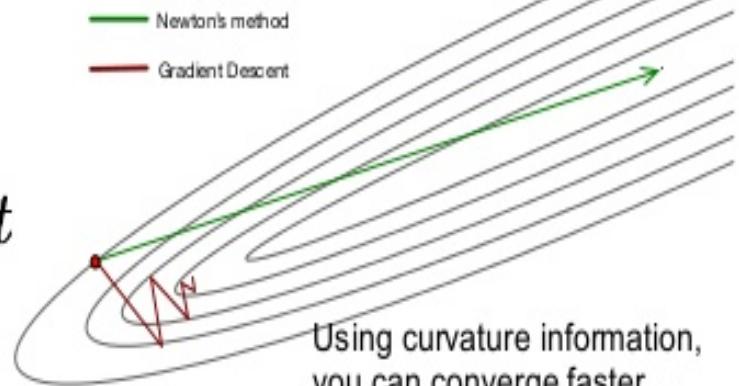
L-BFGS Background

Exact, impractical

$$w_{t+1} \leftarrow w_t - H_t^{-1} g_t \quad \text{in } \mathbb{R}^d$$

$d \times d$ matrix of partial derivatives

$\mathcal{O}(d^3)$ to invert!

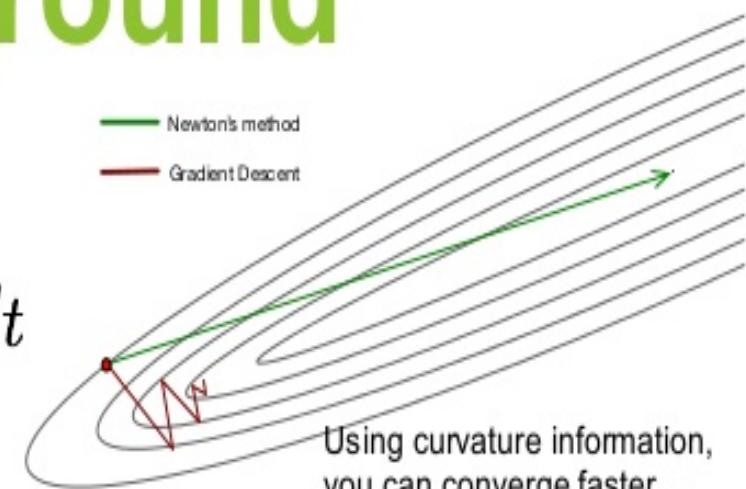


L-BFGS Background

Exact, impractical

$$w_{t+1} \leftarrow w_t - H_t^{-1} g_t \quad \text{in } \mathbb{R}^d$$

- Newton's method
- Gradient Descent



Using curvature information,
you can converge faster...

$d \times d$ matrix of partial derivatives

$\mathcal{O}(d^3)$ to invert!

Approximate, practical

$$w_{t+1} \leftarrow w_t - \gamma_t \tilde{H}_{\text{inv}}(g_t)$$

Step Size computation

- Needs to satisfy some technical (Wolfe) conditions
- Adaptively determined from data

Inverse Hessian Approximation

(based on history of L-previous gradients and model deltas)

L-BFGS Backtracking

Exact, impractical

$$w_{t+1} \leftarrow w_t - \gamma_t H^{-1}(g_t) g_t$$



Approximate, practical

$$w_{t+1} \leftarrow w_t - \gamma_t \tilde{H}_{\text{inv}}(g_t) g_t$$

Step Size computation

- Needs to satisfy some technical (Wolfe) conditions
- Adaptively determined from data

REQUIRE: State vectors $M = (\{s_i\}_{i=t-1}^{t-m}, \{y_i\}_{i=t-1}^{t-m})$

OUTPUT: Proposed search direction

function $H_{\text{inv}}(g_t)$

$$q \leftarrow g_t$$

for $i = t-1, t-2, \dots, t-m$ do

$$\alpha_i \leftarrow \rho_i s_i^\top q$$

$$q \leftarrow q - \alpha_i y_i$$

end for

$$\gamma_t \leftarrow s_{t-1}^\top y_{t-1} / y_{t-1}^\top y_t$$

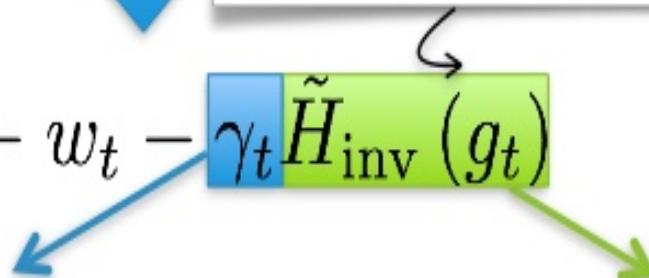
$$r \leftarrow \gamma_t q$$

for $i = t-m, t-m+1, \dots, t-1$ do

$$\beta \leftarrow \rho_i y_i^\top r$$

$$r \leftarrow r + s_i(\alpha_i - \beta)$$

end for



Inverse Hessian Approximation

(based on history of L-previous gradients and model deltas)

L-BFGS Backtracking

Exact, impractical

$$w_{t+1} \leftarrow w_t - \gamma_t H^{-1}(g_t) g_t$$



Approximate, practical

$$w_{t+1} \leftarrow w_t - \gamma_t \tilde{H}_{\text{inv}}(g_t) g_t$$

Step Size computation

- Needs to satisfy some technical (Wolfe) conditions
- Adaptively determined from data

REQUIRE: State vectors $M = (\{s_i\}_{i=t-1}^{t-m}, \{y_i\}_{i=t-1}^{t-m})$

OUTPUT: Proposed search direction

function $H_{\text{inv}}(g_t)$

$q \leftarrow g_t$

for $i = t-1, t-2, \dots, t-m$ do

$\alpha_i \leftarrow \rho_i s_i^\top q$

$q \leftarrow q - \alpha_i y_i$

end for

$\gamma_t \leftarrow s_{t-1}^\top y_{t-1} / y_{t-1}^\top y_t$

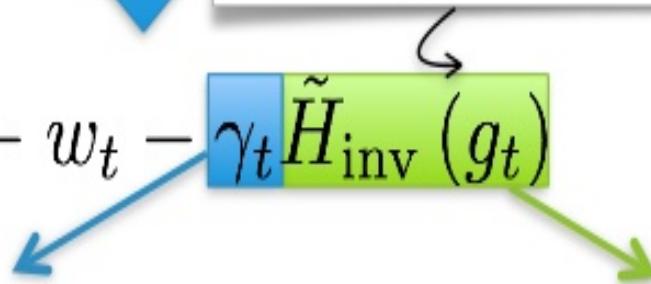
$r \leftarrow \gamma_t q$

for $i = t-m, t-m+1, \dots, t-1$ do

$\beta \leftarrow \rho_i y_i^\top r$

$r \leftarrow r + s_i(\alpha_i - \beta)$

end for



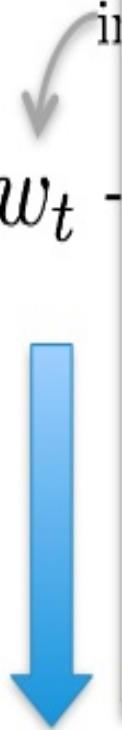
Inverse Hessian Approximation

(based on history of L-previous gradients and model deltas)

L-BFGS Backtracking

Exact, impractical

$$w_{t+1} \leftarrow w_t - \gamma_t H^{-1}(g_t)$$



Approximate, practical

$$w_{t+1} \leftarrow w_t - \gamma_t \tilde{H}_{\text{inv}}(g_t)$$

Step Size computation

- Needs to satisfy some technical (Wolfe) conditions
- Adaptively determined from data

REQUIRE: State vectors $M = (\{s_i\}_{i=t-1}^{t-m}, \{y_i\}_{i=t-1}^{t-m})$

OUTPUT: Proposed search direction

function $H_{\text{inv}}(g_t)$

$q \leftarrow g_t$

for $i = t-1, t-2, \dots, t-m$ do

$\alpha_i \leftarrow \rho_i s_i^\top q$

$q \leftarrow q - \alpha_i y_i$

end for

$\gamma_t \leftarrow s_{t-1}^\top y_{t-1} / y_{t-1}^\top y_t$

$r \leftarrow \gamma_t q$

for $i = t-m, t-m+1, \dots, t-1$ do

$\beta \leftarrow \rho_i y_i^\top r$

$r \leftarrow r + s_i(\alpha_i - \beta)$

end for

Vector Math

copy

dotprod

axpy ($y \leftarrow ax + y$)

dotprod

scal

axpy

scal

Inverse Hessian Approximation

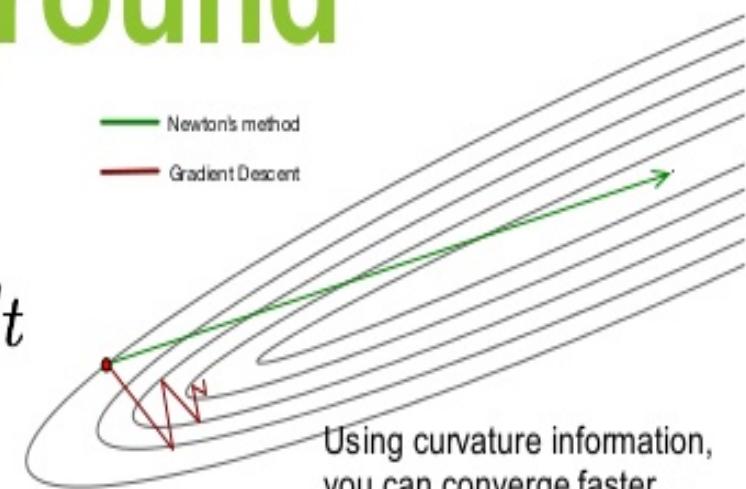
(based on history of L-previous gradients and model deltas)

L-BFGS Background

Exact, impractical

$$w_{t+1} \leftarrow w_t - H_t^{-1} g_t \quad \text{in } \mathbb{R}^d$$

- Newton's method
- Gradient Descent



Using curvature information,
you can converge faster...

$d \times d$ matrix of partial derivatives

$\mathcal{O}(d^3)$ to invert!

Approximate, practical

$$w_{t+1} \leftarrow w_t - \gamma_t \tilde{H}_{\text{inv}}(g_t)$$

Step Size computation

- Needs to satisfy some technical (Wolfe) conditions
- Adaptively determined from data

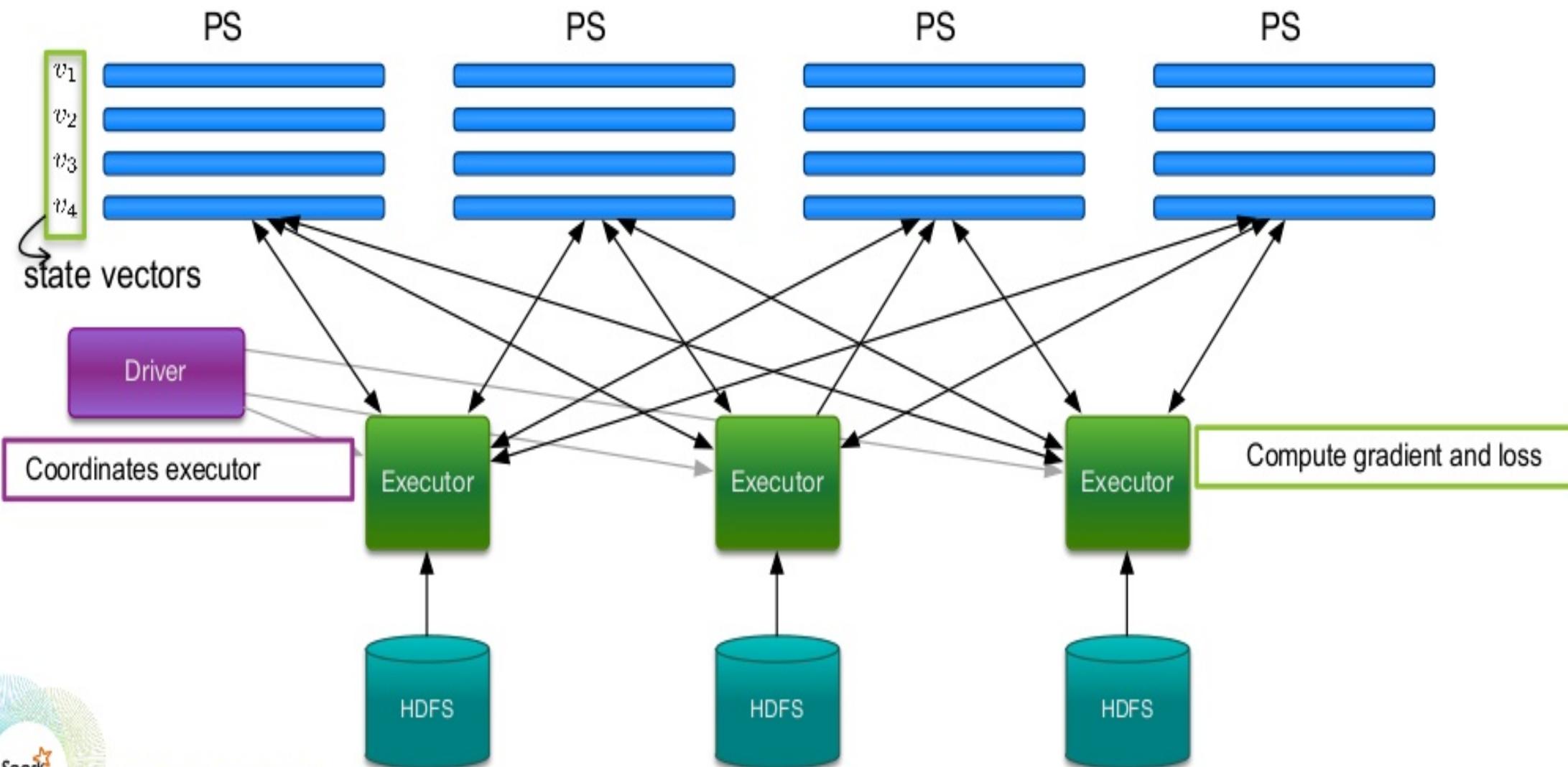
Inverse Hessian Approximation

(based on history of L-previous gradients and model deltas)

Distributed LBFGS*

Step 1: Compute and update Gradient

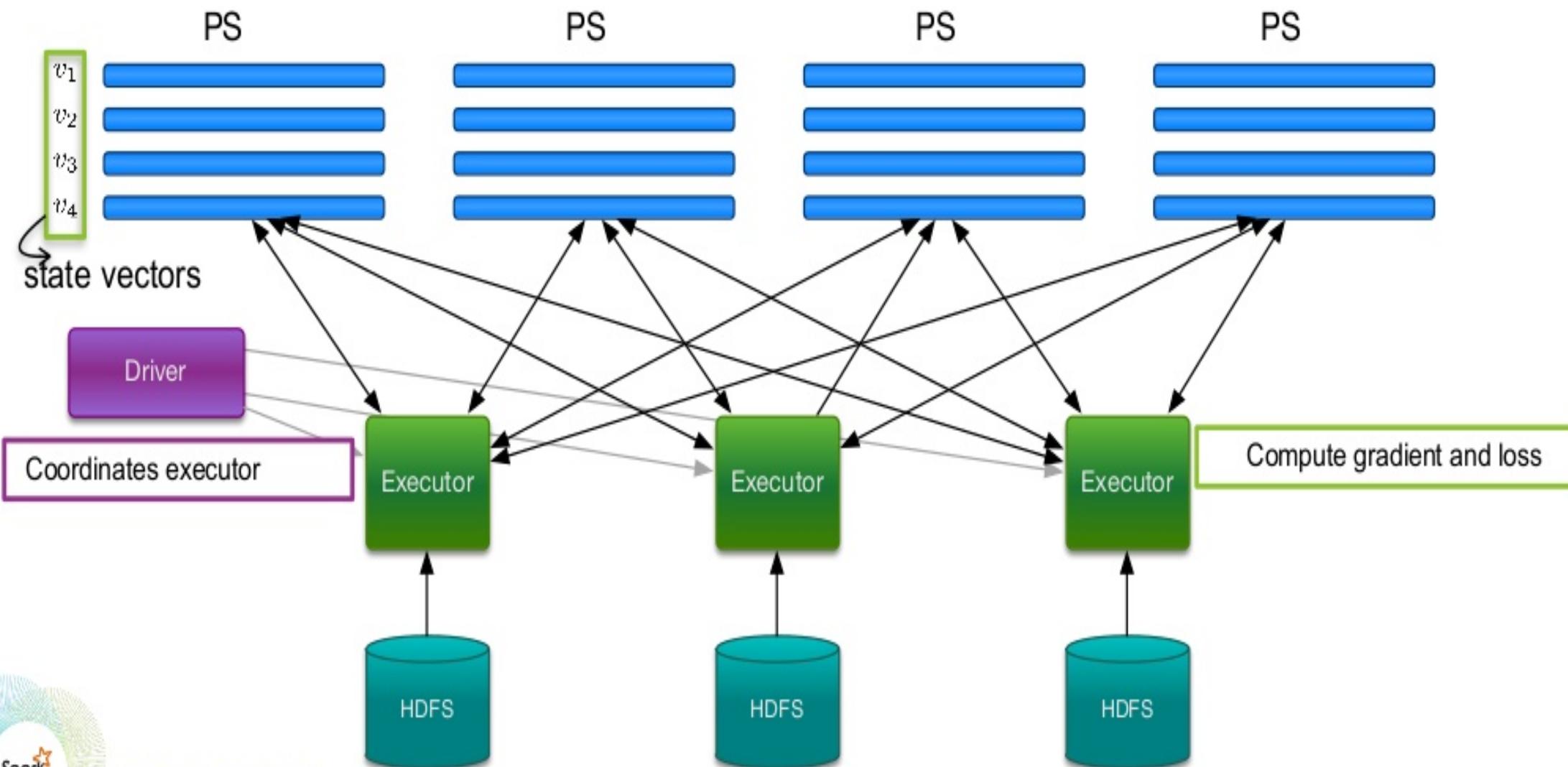
1. Incremental sparse gradient update
2. Fetch sparse portions of model



Distributed LBFGS*

Step 1: Compute and update Gradient

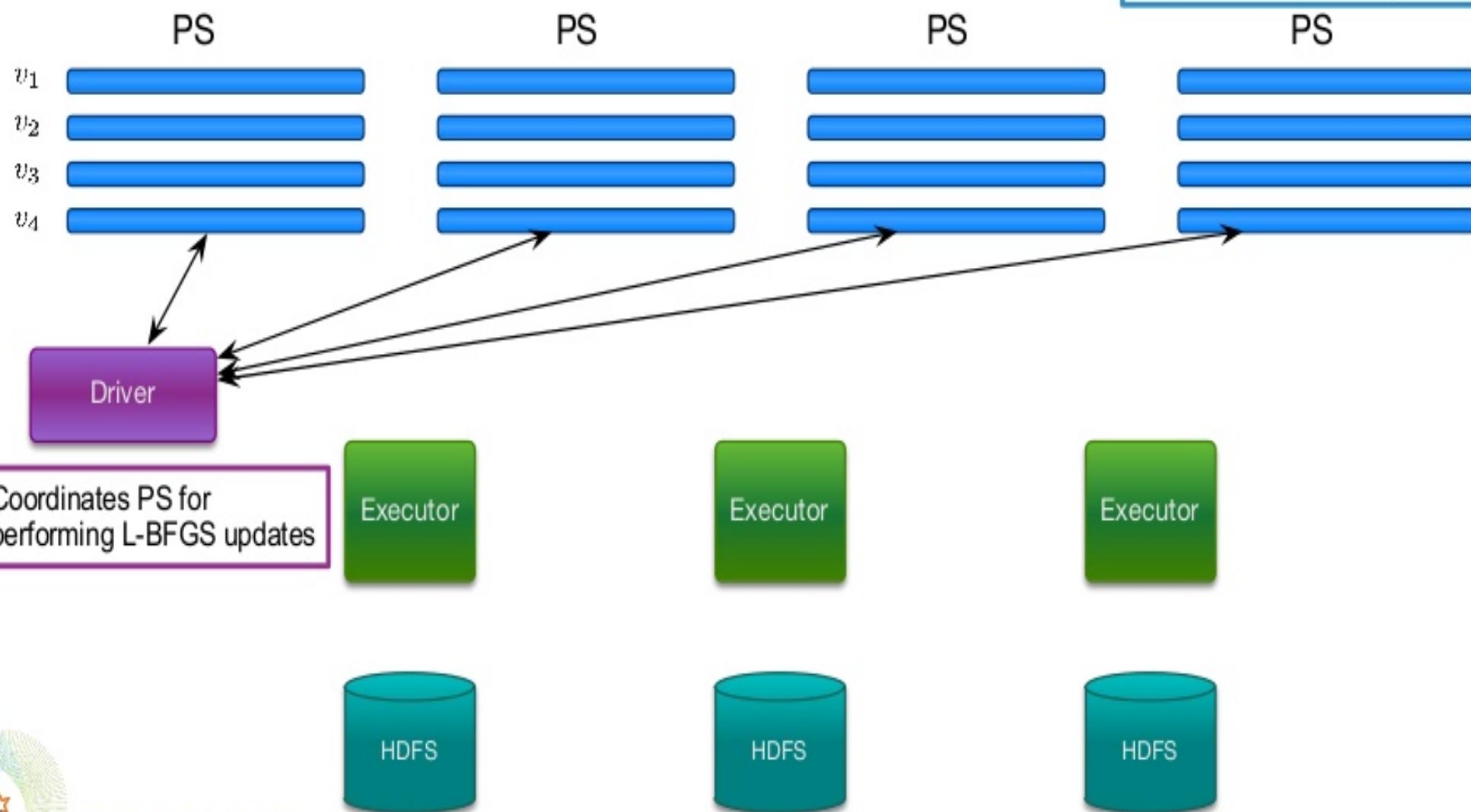
1. Incremental sparse gradient update
2. Fetch sparse portions of model



Distributed LBFGS

Step 2: Build inverse Hessian Approximation

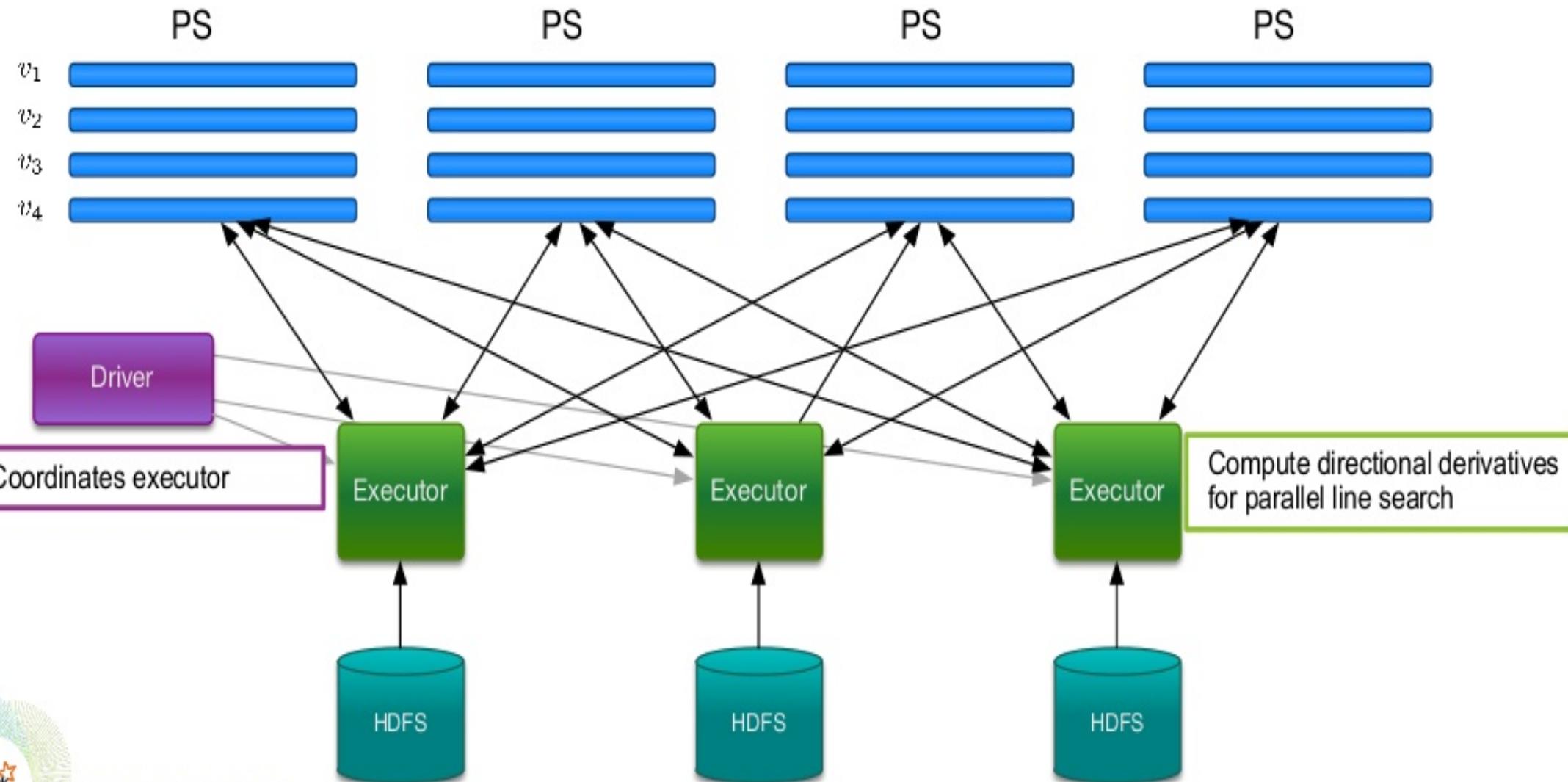
Actual L-BFGS updates
(BLAS vector math)



Distributed LBFGS

Step 3: Compute losses and directional derivatives

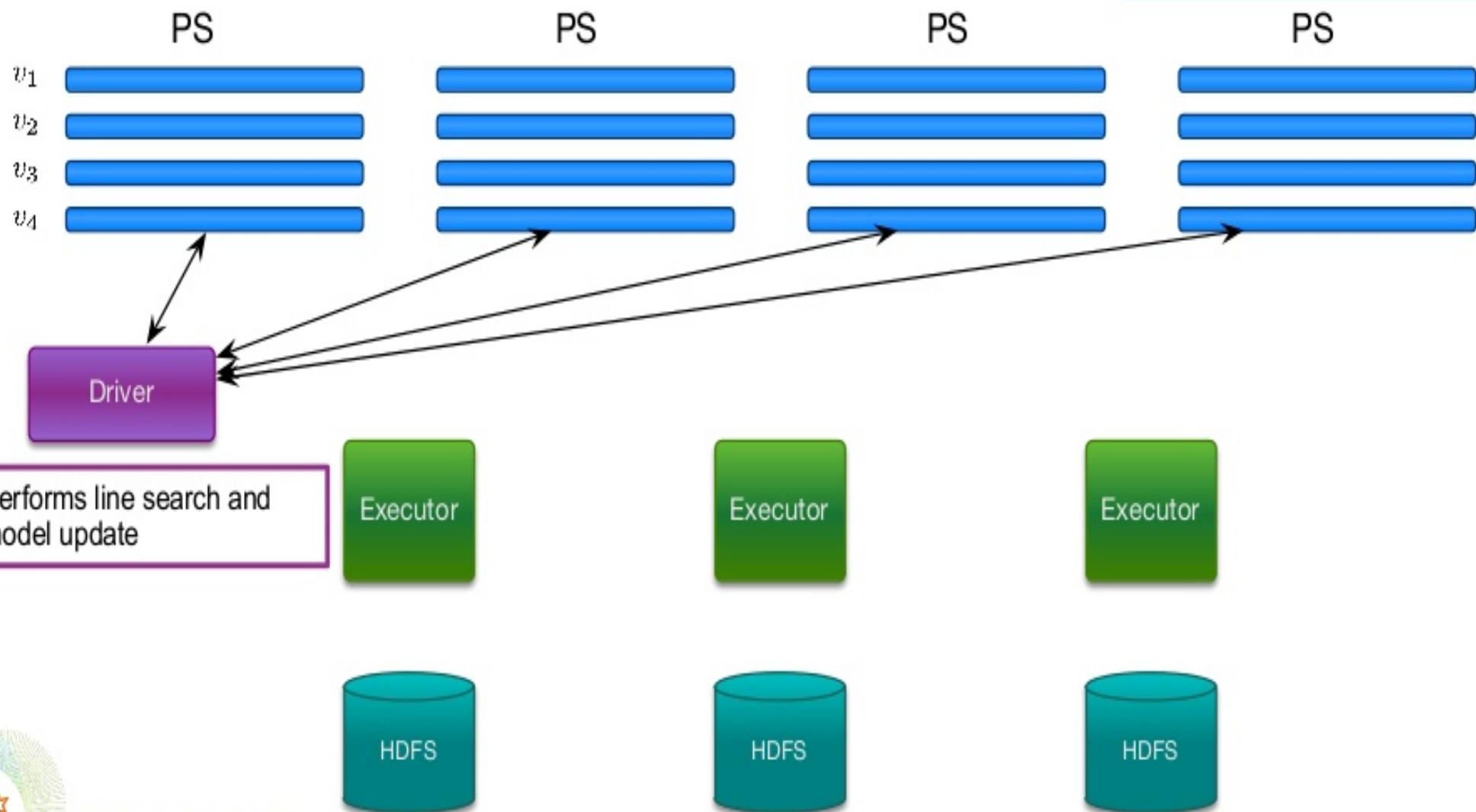
Fetch **sparse** portions of model



Distributed LBFGS

Step 4: Line search and model update

Model update (BLAS vector math)



Speedup tricks

Speedup tricks

- Intersperse communication and computation

Speedup tricks

- Intersperse communication and computation
- Quicker convergence
 - Parallel line search for step size
 - Curvature for initial Hessian approximation*



Speedup tricks

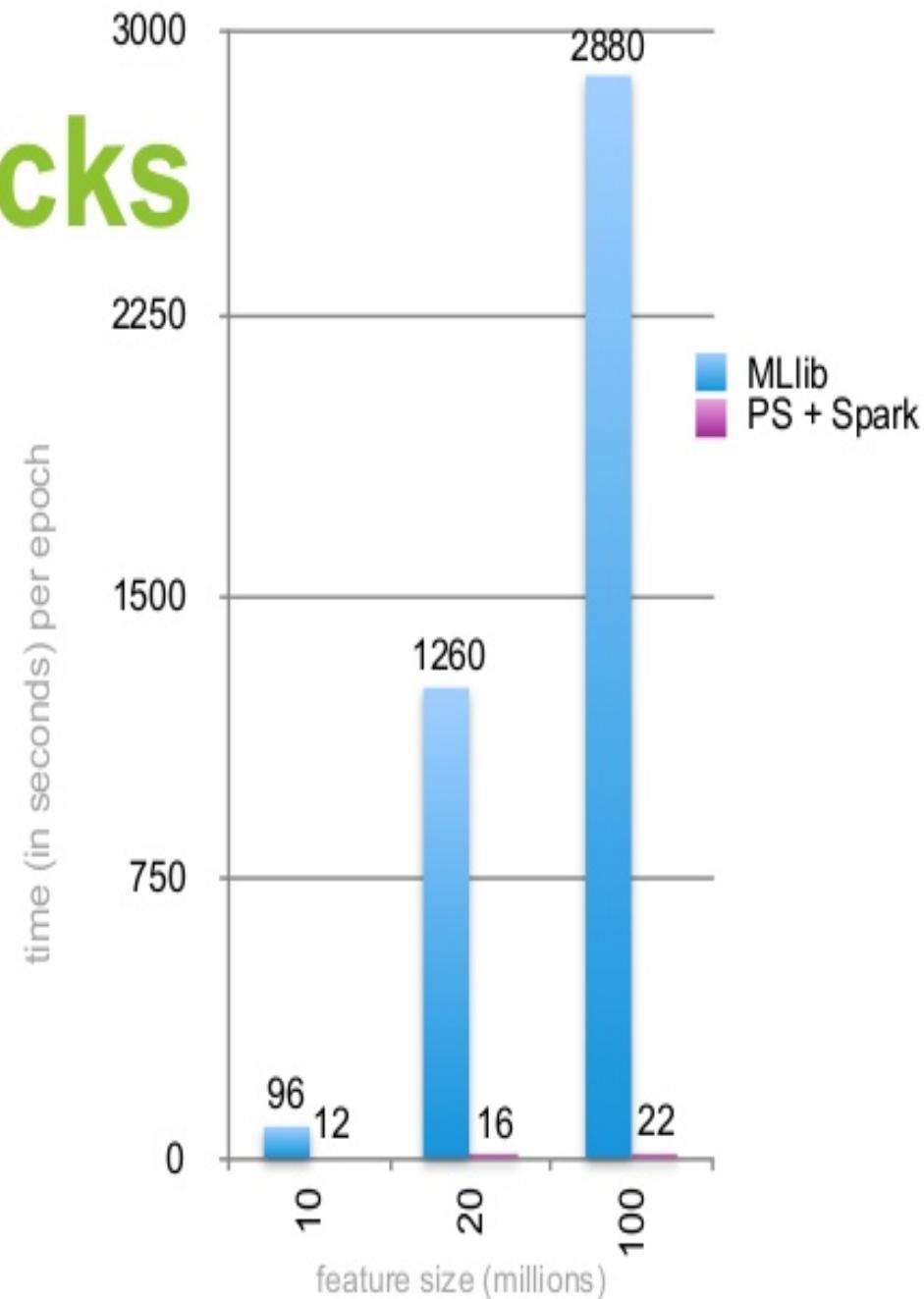
- Intersperse communication and computation
- Quicker convergence
 - Parallel line search for step size
 - Curvature for initial Hessian approximation*
- Network bandwidth reduction
 - Compressed integer arrays
 - Only store indices for binary data

Speedup tricks

- Intersperse communication and computation
- Quicker convergence
 - Parallel line search for step size
 - Curvature for initial Hessian approximation*
- Network bandwidth reduction
 - Compressed integer arrays
 - Only store indices for binary data
- Matrix math on minibatch

Speedup tricks

- Intersperse communication and computation
- Quicker convergence
 - Parallel line search for step size
 - Curvature for initial Hessian approximation*
- Network bandwidth reduction
 - Compressed integer arrays
 - Only store indices for binary data
- Matrix math on minibatch



1.6×10^8 examples, 100 executors, 10 cores

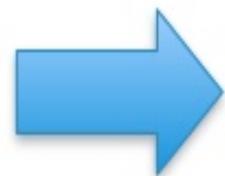
Word Embeddings

Word Embeddings



WIKIPEDIA
The Free Encyclopedia

Word Embeddings



$v(\text{paris}) = [0.13, -0.4, 0.22, \dots, -0.45]$
 $v(\text{lion}) = [-0.23, -0.1, 0.98, \dots, 0.65]$
 $v(\text{quark}) = [1.4, 0.32, -0.01, \dots, 0.023]$

⋮
⋮
⋮

WIKIPEDIA
The Free Encyclopedia



Word2vec

Word2vec

Distributed Representations of Words and Phrases and their Compositionality

Tomas Mikolov

Google Inc.

Mountain View

mikolov@google.com

Ilya Sutskever

Google Inc.

Mountain View

ilyasu@google.com

Kai Chen

Google Inc.

Mountain View

kai@google.com

Greg Corrado

Google Inc.

Mountain View

gcorrado@google.com

Jeffrey Dean

Google Inc.

Mountain View

jeff@google.com



Word2vec

Distributed Representations of Words and Phrases and their Compositionality

Tomas Mikolov

Google Inc.

Mountain View

mikolov@google.com

Ilya Sutskever

Google Inc.

Mountain View

ilyasu@google.com

Kai Chen

Google Inc.

Mountain View

kai@google.com

Greg Corrado

Google Inc.

Mountain View

gcorrado@google.com

Jeffrey Dean

Google Inc.

Mountain View

jeff@google.com

- new techniques to compute vector representations of words from corpus



Word2vec

Distributed Representations of Words and Phrases and their Compositionality

Tomas Mikolov
Google Inc.
Mountain View

mikolov@google.com

Ilya Sutskever
Google Inc.
Mountain View

ilyasu@google.com

Kai Chen
Google Inc.
Mountain View

kai@google.com

Greg Corrado
Google Inc.
Mountain View

gcorrado@google.com

Jeffrey Dean
Google Inc.
Mountain View

jeff@google.com

- new techniques to compute vector representations of words from corpus
- geometry of vectors captures word semantics

Word2vec

Word2vec

- Skipgram with negative sampling:

Word2vec

- Skipgram with negative sampling:
 - training set includes pairs of words and neighbors in corpus, along with randomly selected words for each neighbor

Word2vec

- Skipgram with negative sampling:
 - training set includes pairs of words and neighbors in corpus, along with randomly selected words for each neighbor
 - determine $w \rightarrow \mathbf{u}(w), \mathbf{v}(w)$ so that $\text{sigmoid}(\mathbf{u}(w) \bullet \mathbf{v}(w'))$ is close to (minimizes log loss) the probability that w' is a neighbor of w as opposed to a randomly selected word.

Word2vec

- Skipgram with negative sampling:
 - training set includes pairs of words and neighbors in corpus, along with randomly selected words for each neighbor
 - determine $w \rightarrow \mathbf{u}(w), \mathbf{v}(w)$ so that $\text{sigmoid}(\mathbf{u}(w) \bullet \mathbf{v}(w'))$ is close to (minimizes log loss) the probability that w' is a neighbor of w as opposed to a randomly selected word.
 - SGD involves computing many vector dot products e.g., $\mathbf{u}(w) \bullet \mathbf{v}(w')$ and vector linear combinations e.g., $\mathbf{u}(w) += \alpha \mathbf{v}(w')$.

Word2vec Application at Yahoo

- Example training data:

gas_cap_replacement_for_car

slc_679f037df54f5d9c41cab05bfae0926

gas_door_replacement_for_car

slc_466145af16a40717c84683db3f899d0a fuel_door_covers

adid_c_28540527225_285898621262

slc_348709d73214fdeb9782f8b71aff7b6e autozone_auto_parts

adid_b_3318310706_280452370893 auoto_zone

slc_8dcda5d20a2caa02b8b1d1c8ccbd36b

slc_58f979b6deb6f40c640f7ca8a177af2d

Distributed Word2vec

Distributed Word2vec

- Needed system to train 200 million 300 dimensional word2vec model using minibatch SGD

Distributed Word2vec

- Needed system to train 200 million 300 dimensional word2vec model using minibatch SGD
- Achieved in a high throughput and network efficient way using our matrix based PS server:

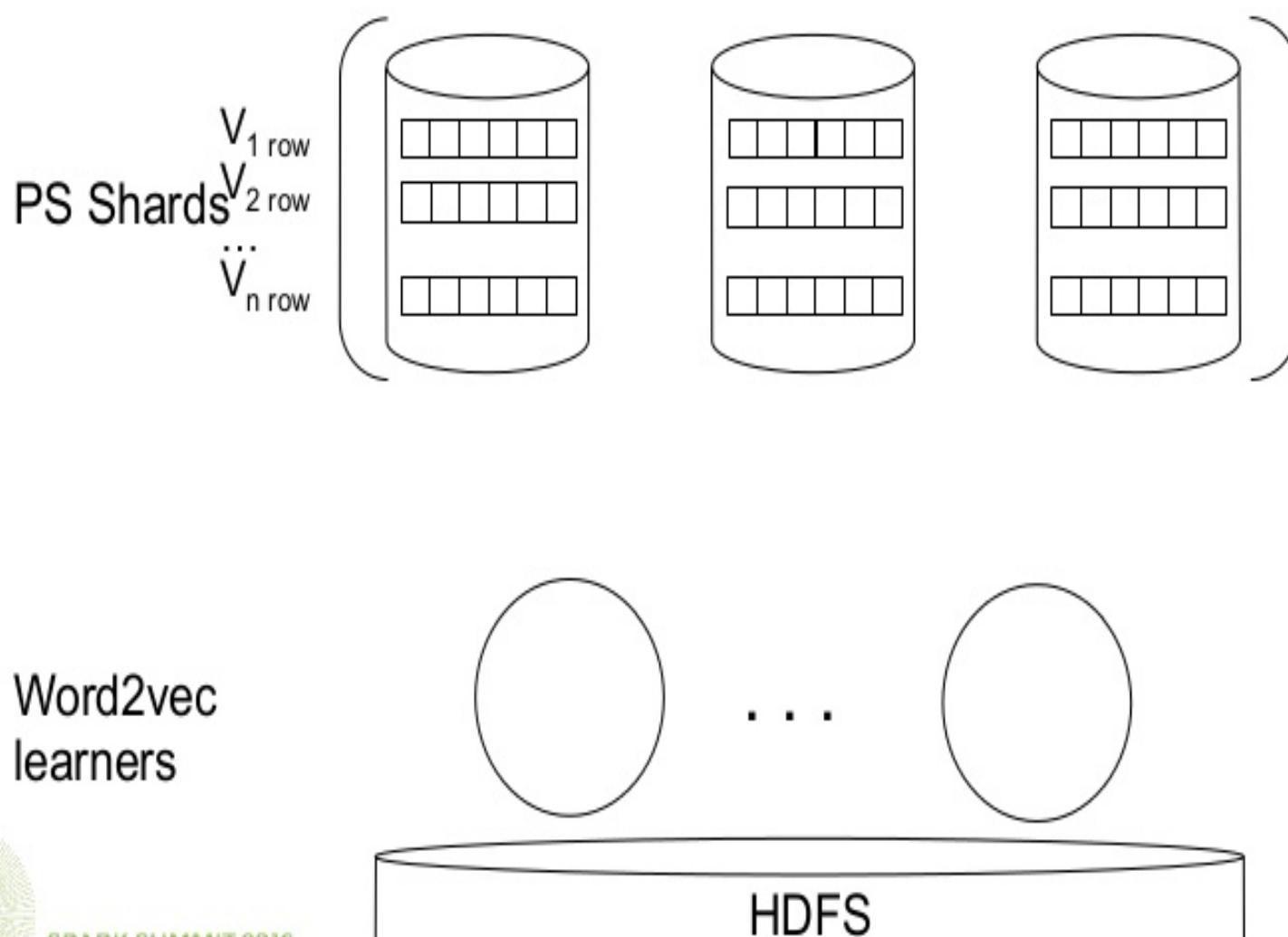
Distributed Word2vec

- Needed system to train 200 million 300 dimensional word2vec model using minibatch SGD
- Achieved in a high throughput and network efficient way using our matrix based PS server:
 - Vectors don't go over network.

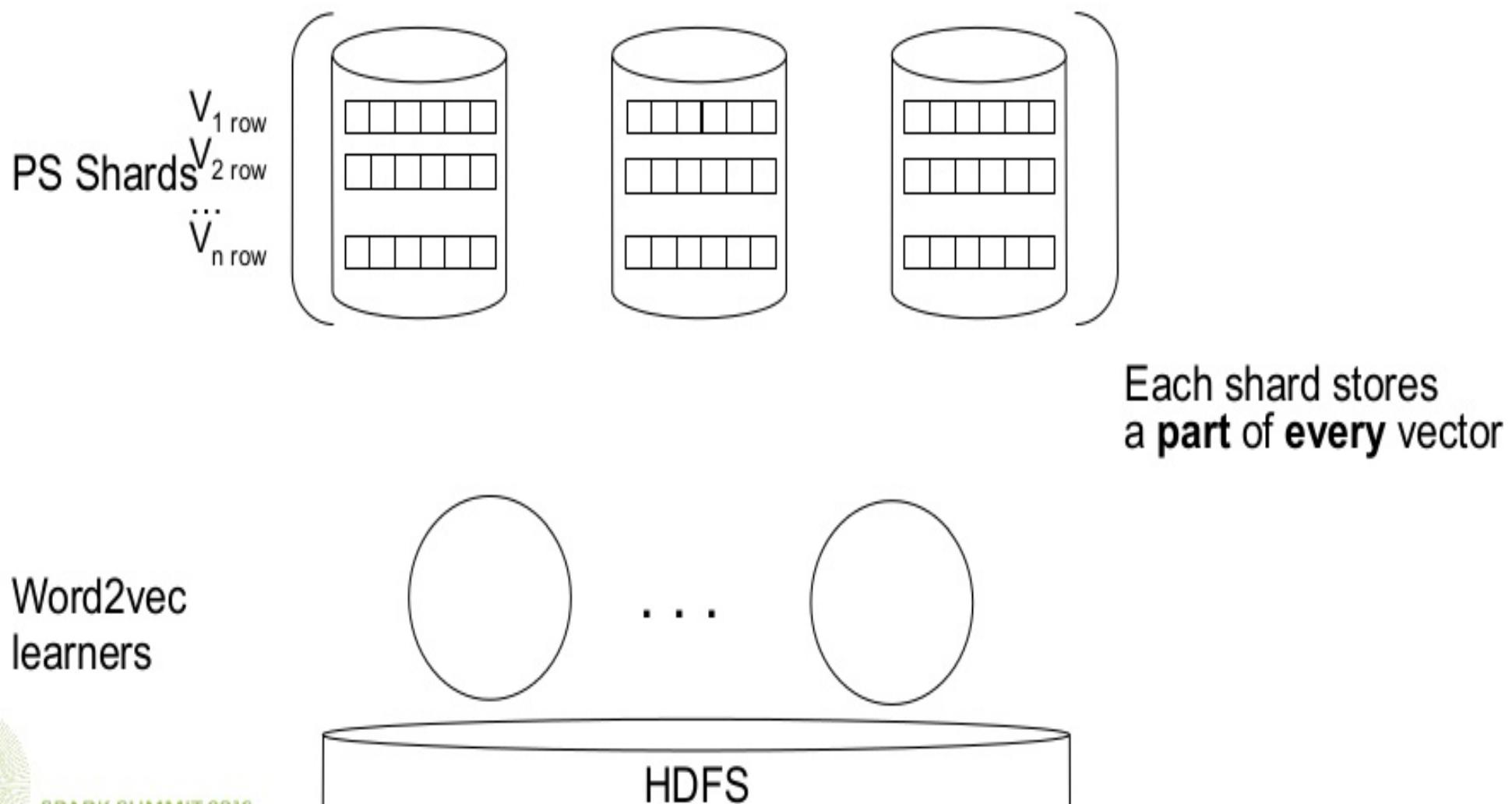
Distributed Word2vec

- Needed system to train 200 million 300 dimensional word2vec model using minibatch SGD
- Achieved in a high throughput and network efficient way using our matrix based PS server:
 - Vectors don't go over network.
 - Most compute on PS servers, with clients aggregating partial results from shards.

Distributed Word2vec

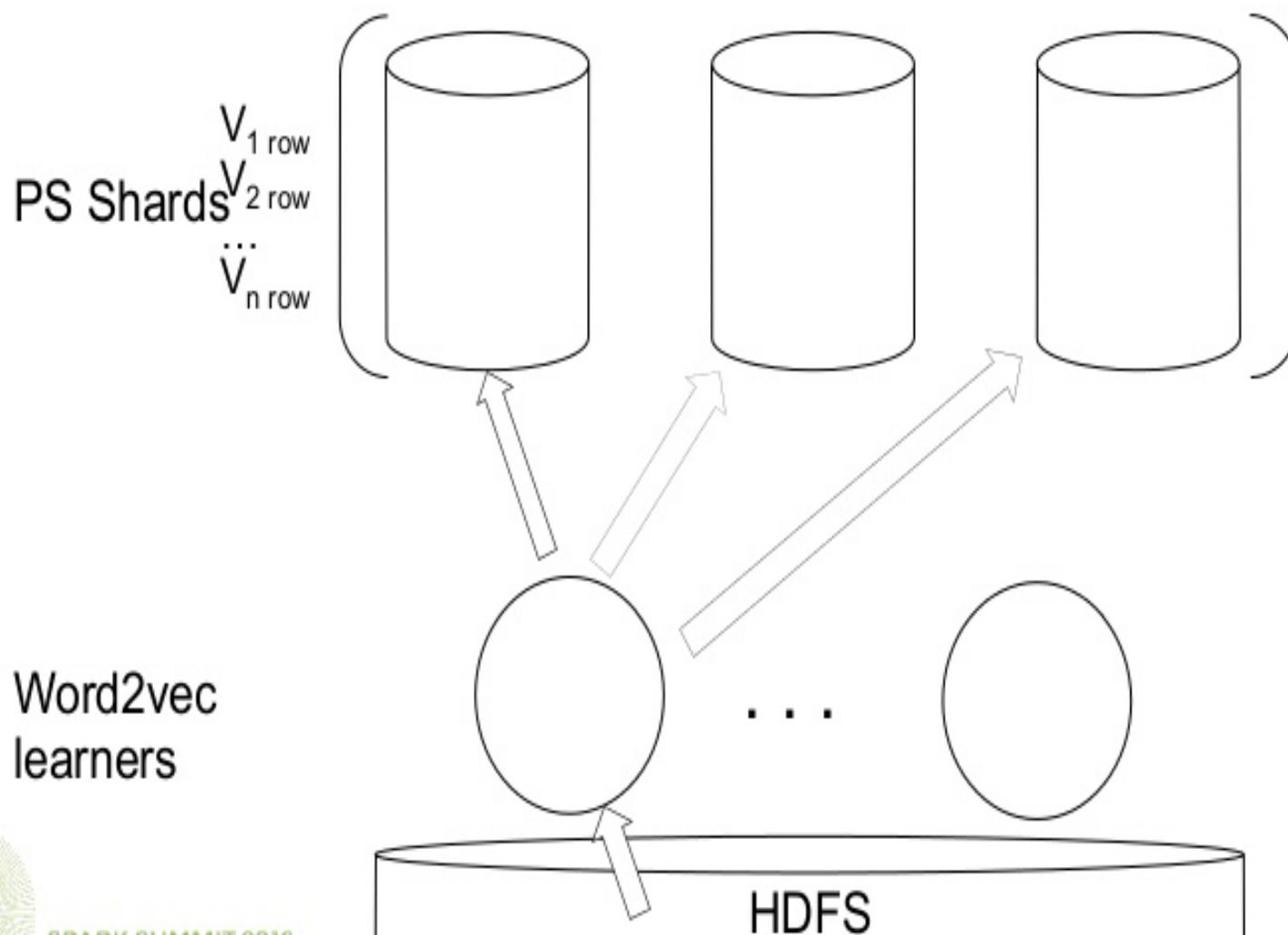


Distributed Word2vec



Distributed Word2vec

Send word indices
and seeds

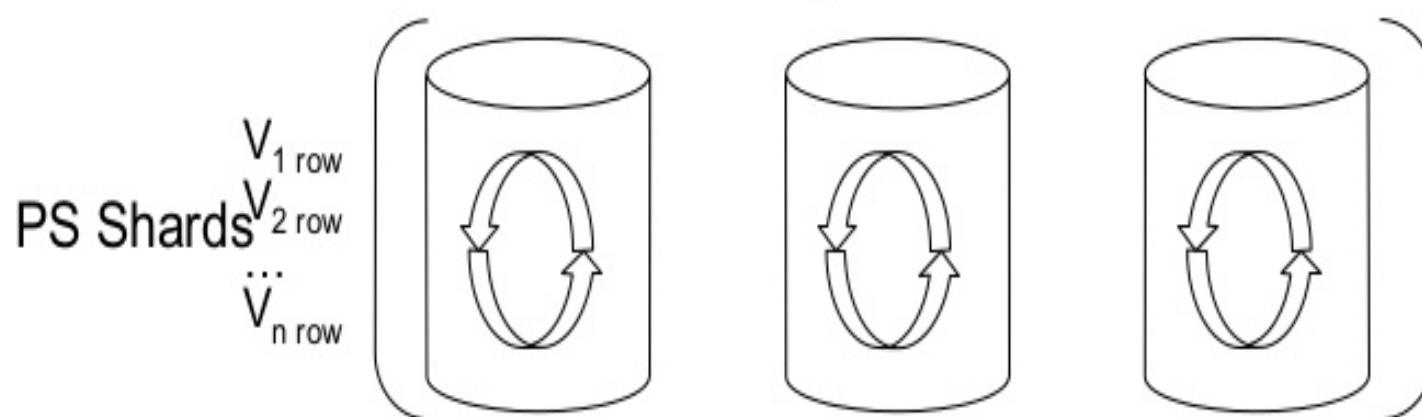


Word2vec
learners

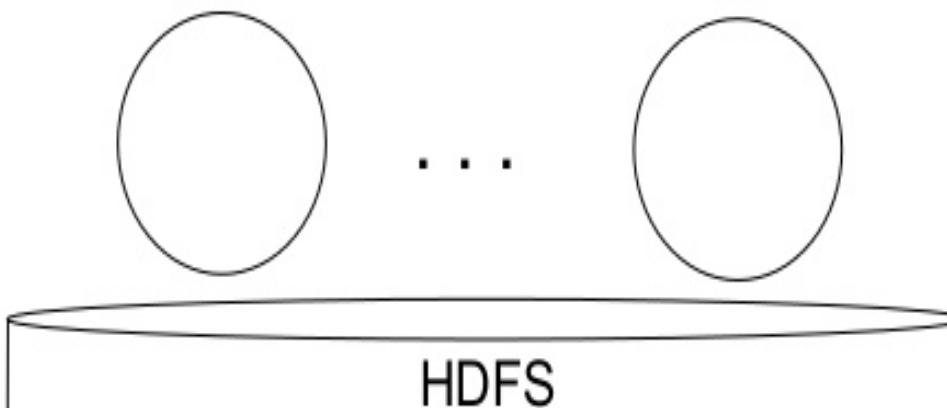
HDFS

Distributed Word2vec

Negative sampling,
compute $u \bullet v$

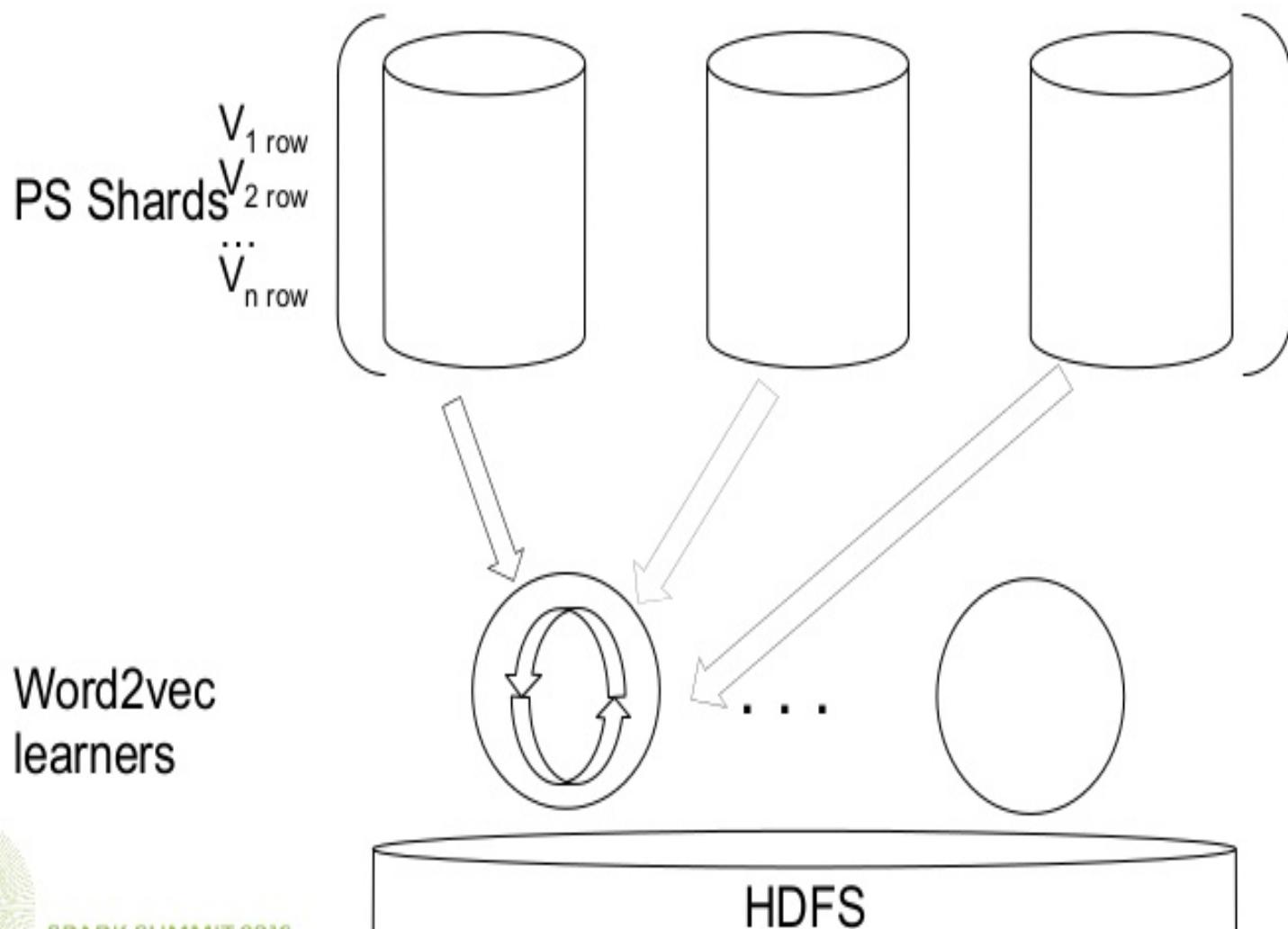


Word2vec
learners

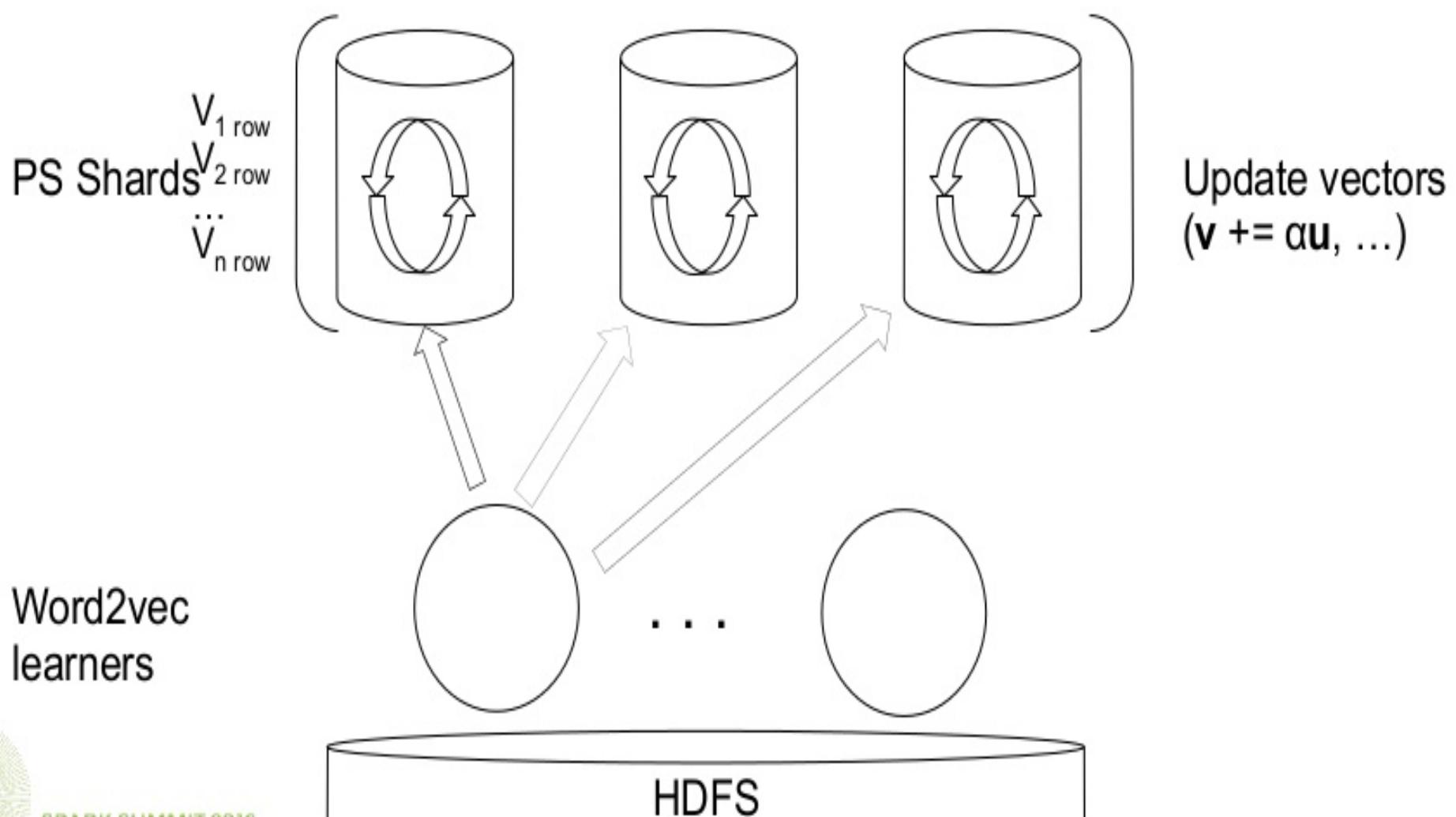


Distributed Word2vec

Aggregate results &
compute lin. comb. coefficients (e.g., α ...)



Distributed Word2vec



Distributed Word2vec

Distributed Word2vec

- Network lower by factor of #shards/dimension compared to conventional PS based system (1/20 to 1/100 for useful scenarios).

Distributed Word2vec

- Network lower by factor of #shards/dimension compared to conventional PS based system (1/20 to 1/100 for useful scenarios).
- Trains 200 million vocab, 55 billion word search session in 2.5 days.

Distributed Word2vec

- Network lower by factor of #shards/dimension compared to conventional PS based system (1/20 to 1/100 for useful scenarios).
- Trains 200 million vocab, 55 billion word search session in 2.5 days.
- In production for regular training in Yahoo search ad serving system.

Other Projects using Spark + PS

- Online learning on PS
 - Personalization as a Service
 - Sponsored Search
- Factorization Machines
 - Large scale user profiling

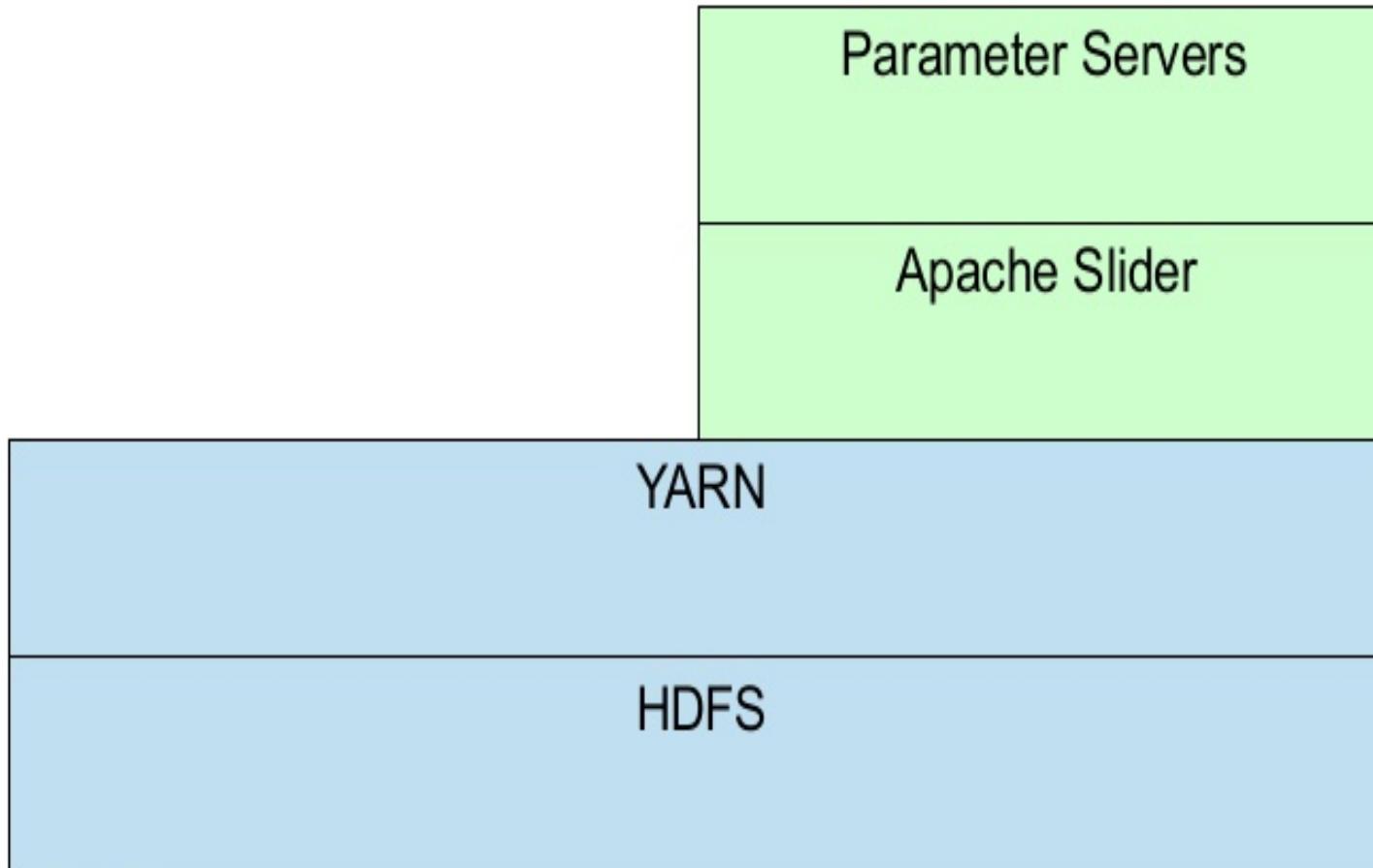
SPARK+PS ON HADOOP CLUSTER



Training Data on HDFS

HDFS

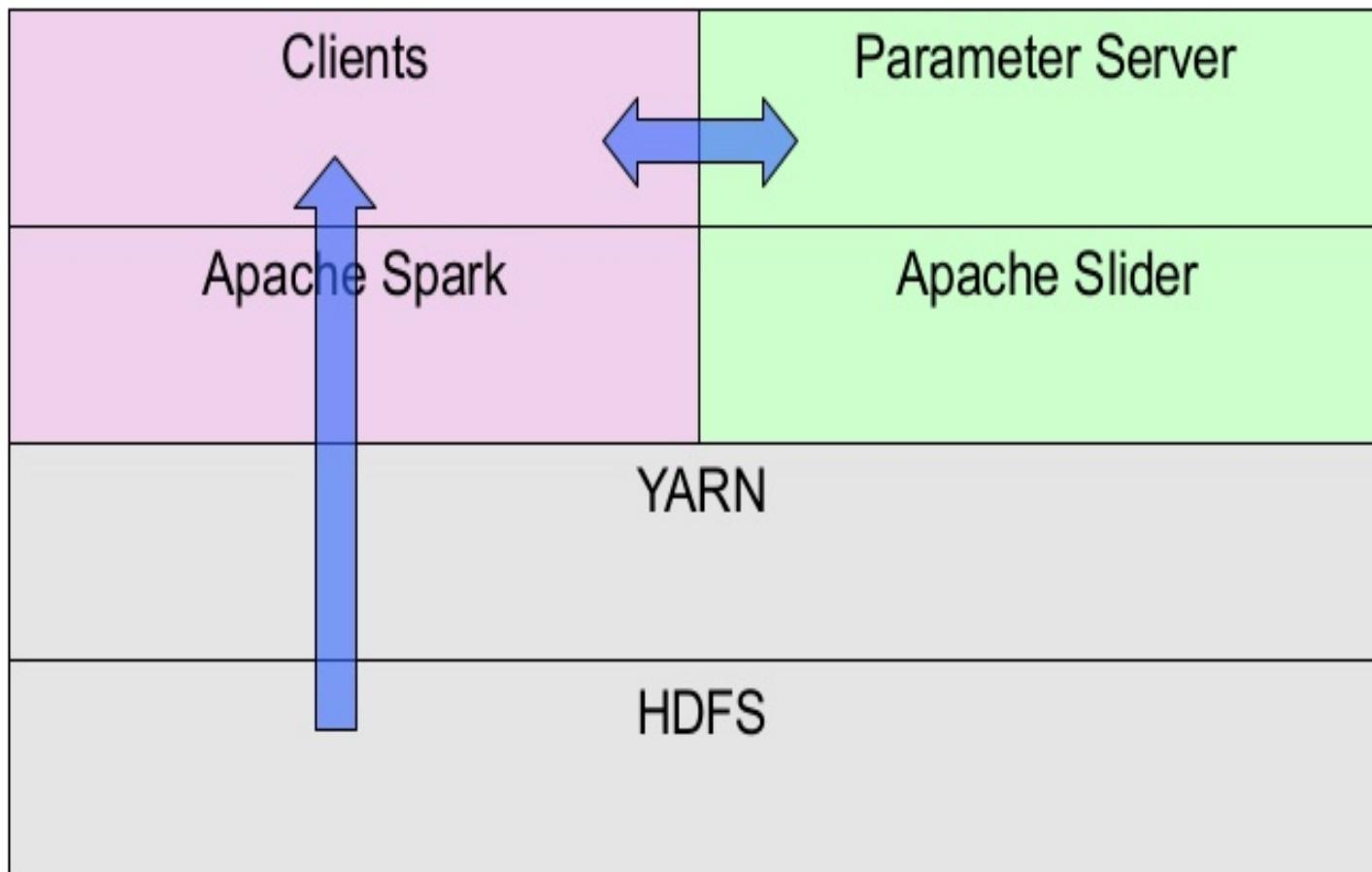
Launch PS Using Apache Slider on YARN



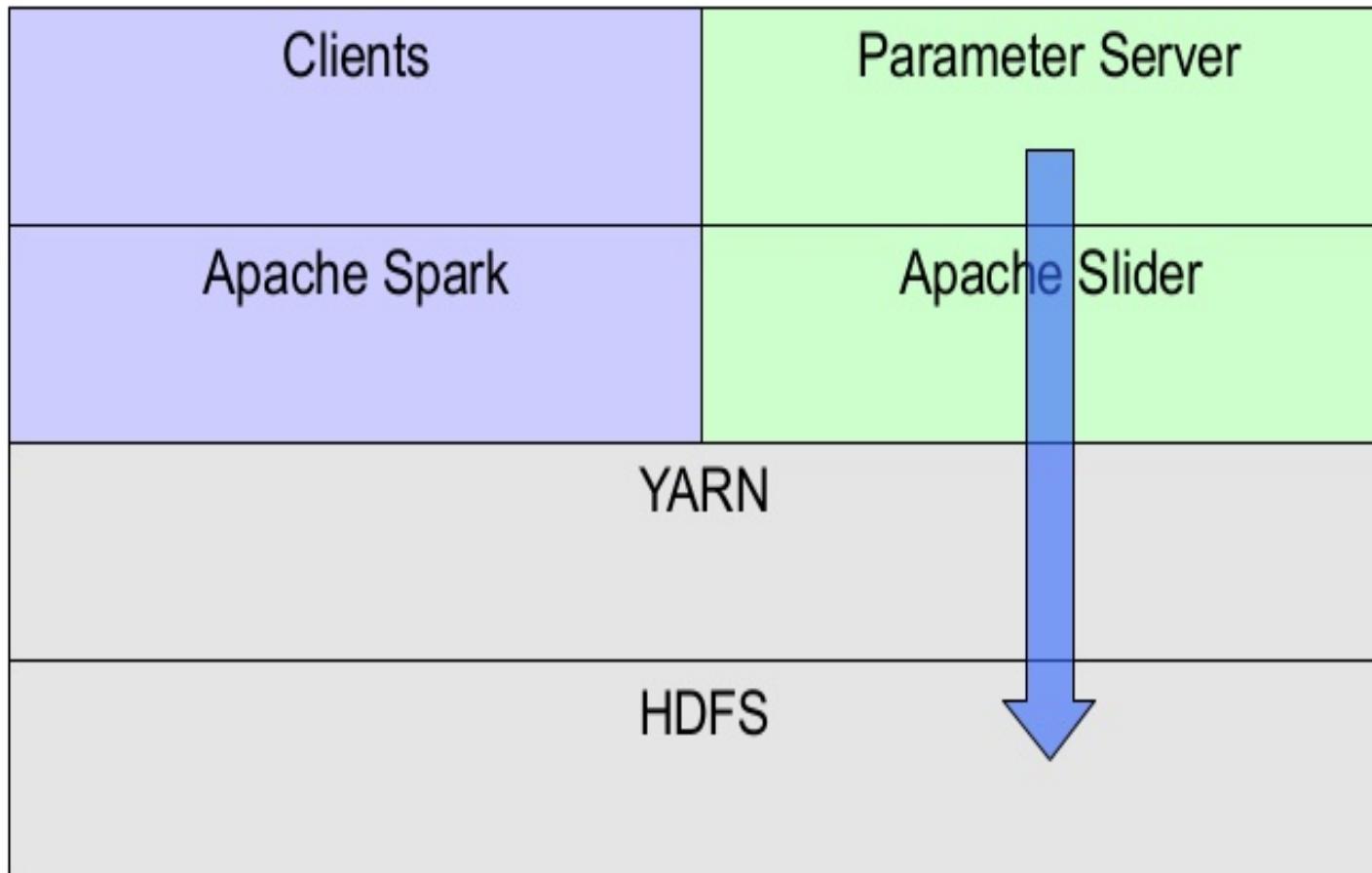
Launch Clients using Spark or Hadoop Streaming API

Clients	Parameter Servers
Apache Spark	Apache Slider
YARN	
HDFS	

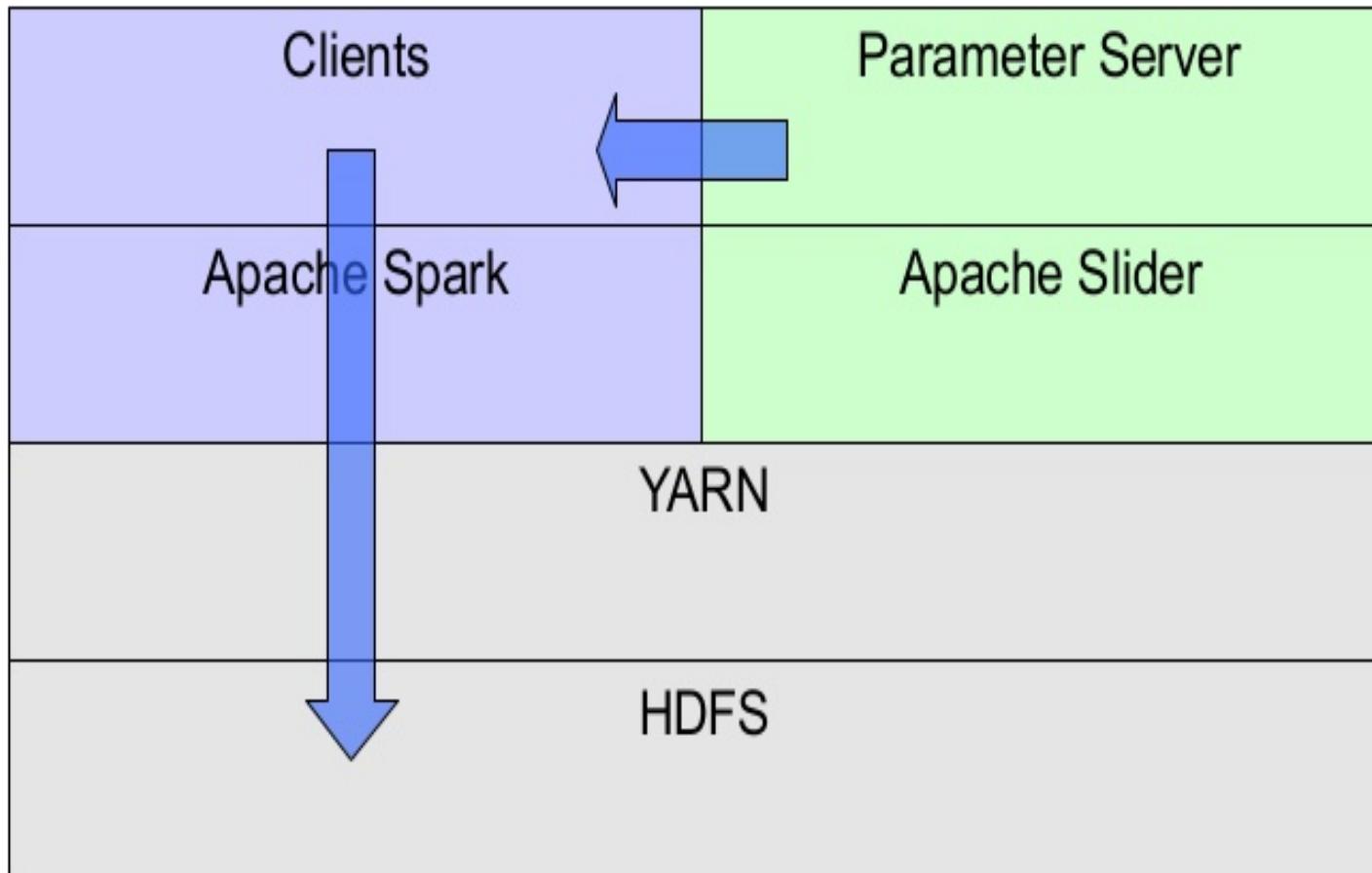
Training



Model Export



Model Export



Summary

- Parameter server indispensable for big models
- Spark + Parameter Server has proved to be very flexible platform for our large scale computing needs
- Direct computation on the parameter servers accelerate training for our use-cases

Thank you!

For more, contact bigdata@yahoo-inc.com.



SPARK SUMMIT 2016
DATA SCIENCE AND ENGINEERING AT SCALE