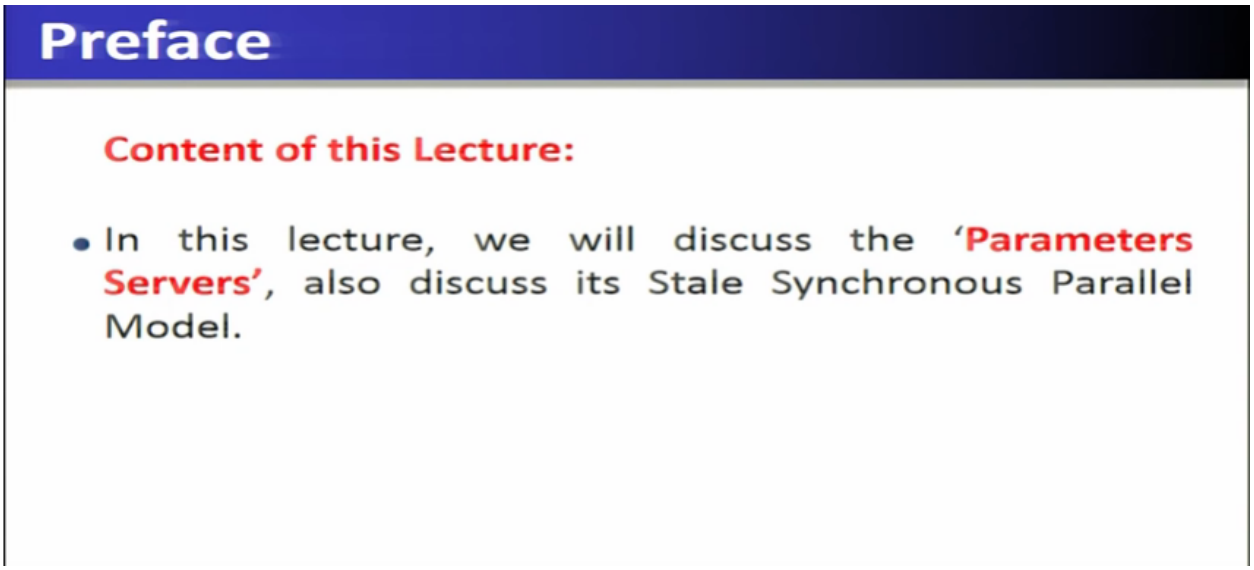
Lecture - 30
Parameter Servers

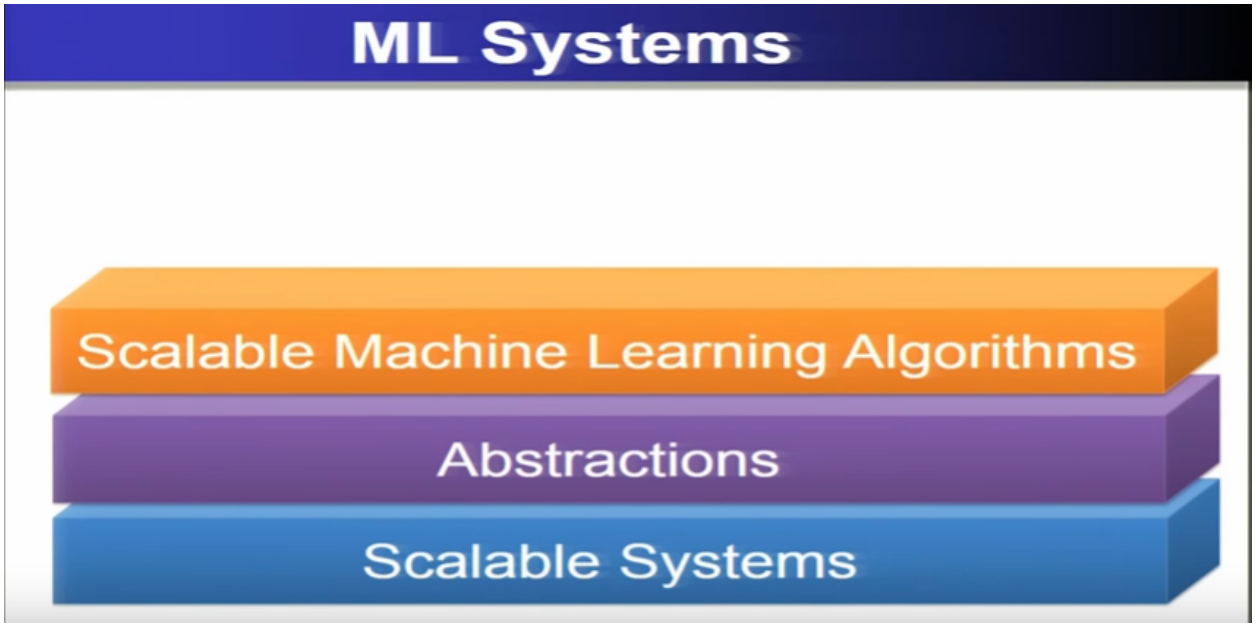
Parameter servers.

Refer Slide Time :(0:17)



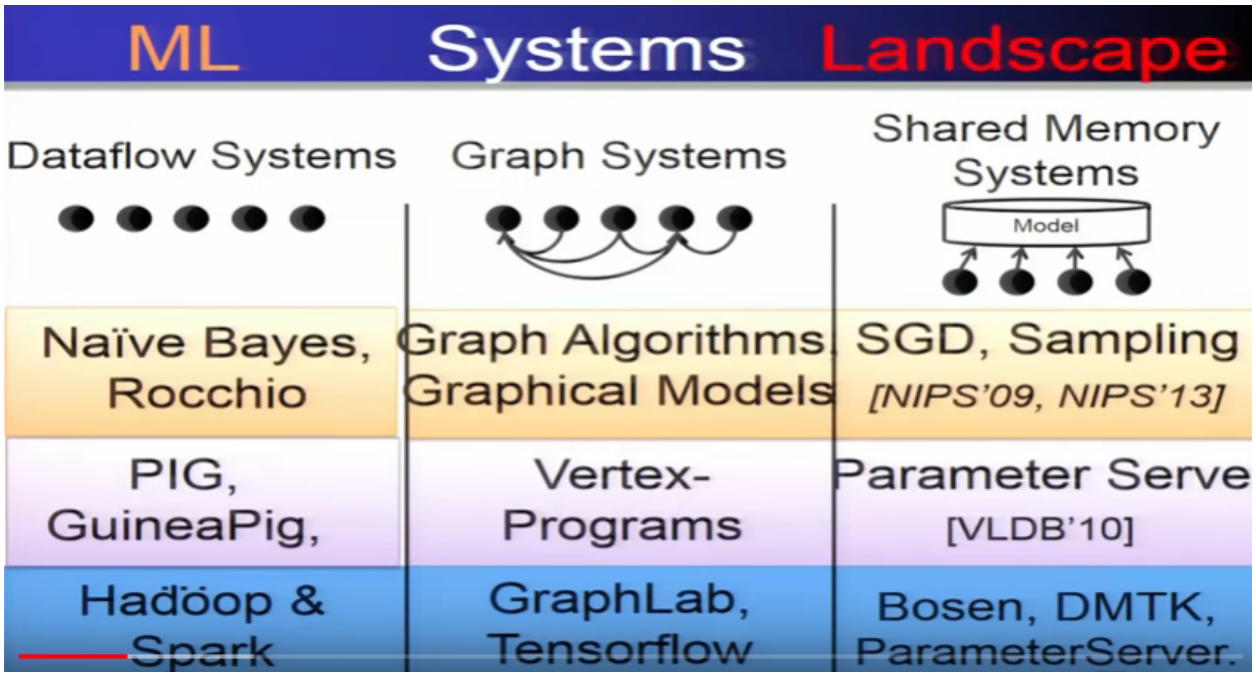
Preface, content of this lecture: In this lecture, we will discuss the parameter servers. And also, discuss the stale synchronous parallel model.

Refer Slide Time :(0:27)



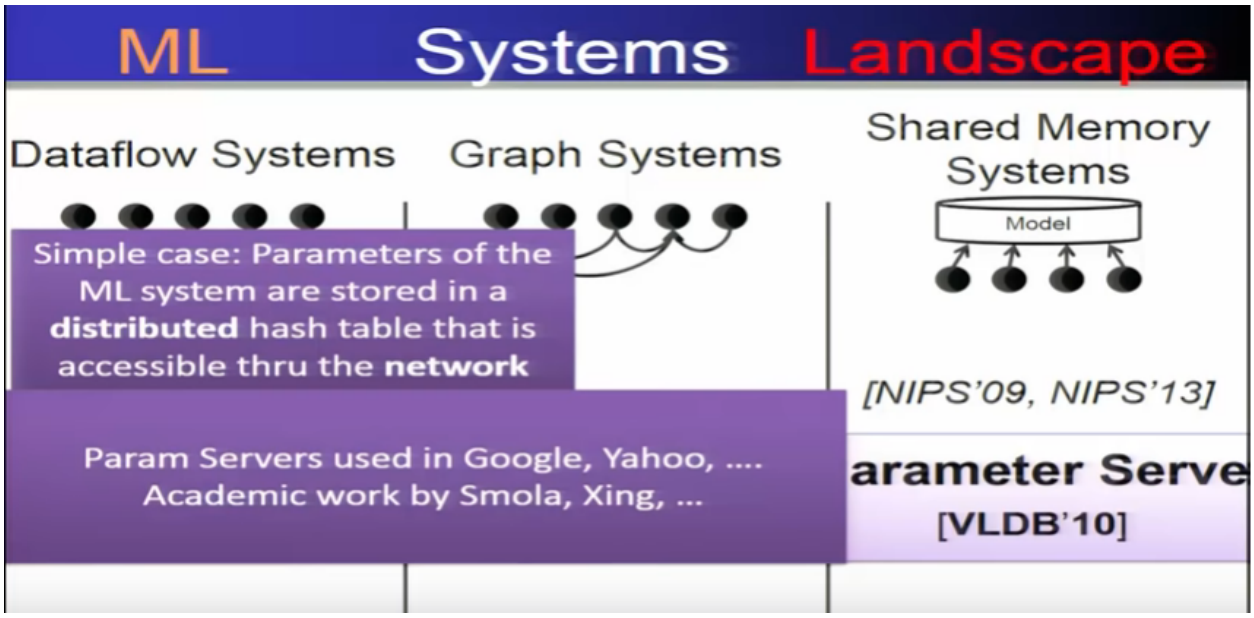
In machine learning systems, especially the scalable machine learning algorithms that we are seeing for, big data computing, requires the abstractions, for the scalable, building scalable systems.

Refer Slide Time :(0:48)



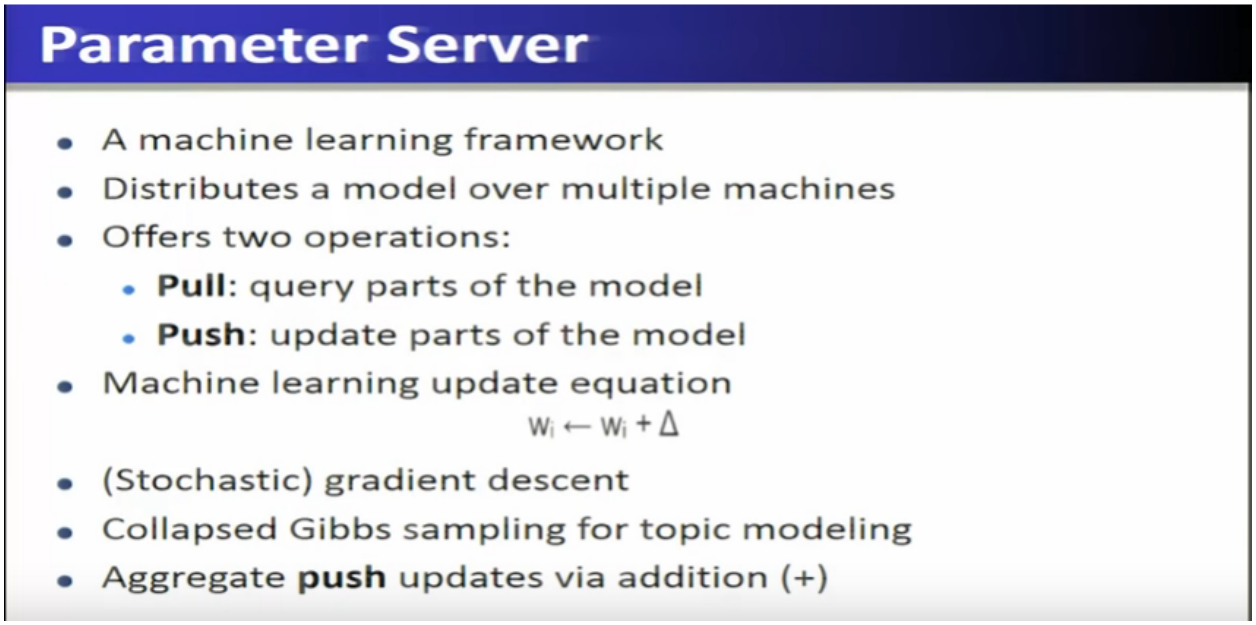
And here, we can see that, there are three different the machine learning systems landscape we have to see that, there are data flow systems which are, built over Hadoop and stack platforms, using the graph system algorithms and shared memory systems. And these machine learning algorithms, which are nothing but supported as the dataflow systems, such as naive Bayes and requires an abstraction from the underlying systems like, Hadoop and spark and for graph systems it requires abstractions from, the graph lab and tensor flow. And similarly, for shared memory systems and these are Bosen and DMT K, also requires the parameter servers.

Refer Slide Time :(1:51)



So, what we will see here is that, in, in a simple case, the parameters of machine learning are stored in a distributed hash table that is accessible, through the, through the network. Which is there in the shared memory systems? So, parameter servers which are used in Google, Yahoo and also, is used as an academic work by Smola and Xing.

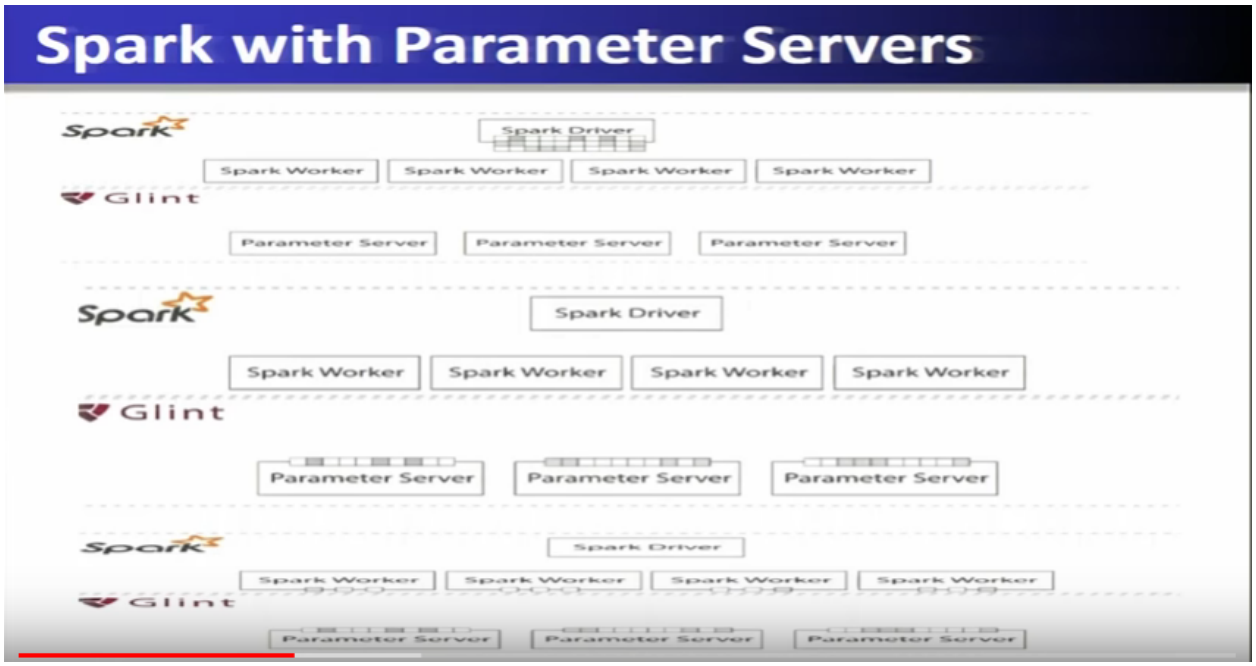
Refer Slide Time :(2:20)



And now, we have to see the parameter server and its how it is going to be used as, the framework for machine learning. So, to build the scalable machine learning systems, so skill machine learning framework requires, the use of parameter server and it distributes, a model over multiple machines and offers two different operations, one is called, 'Pull', that is the query part of the model. And the, 'Push',

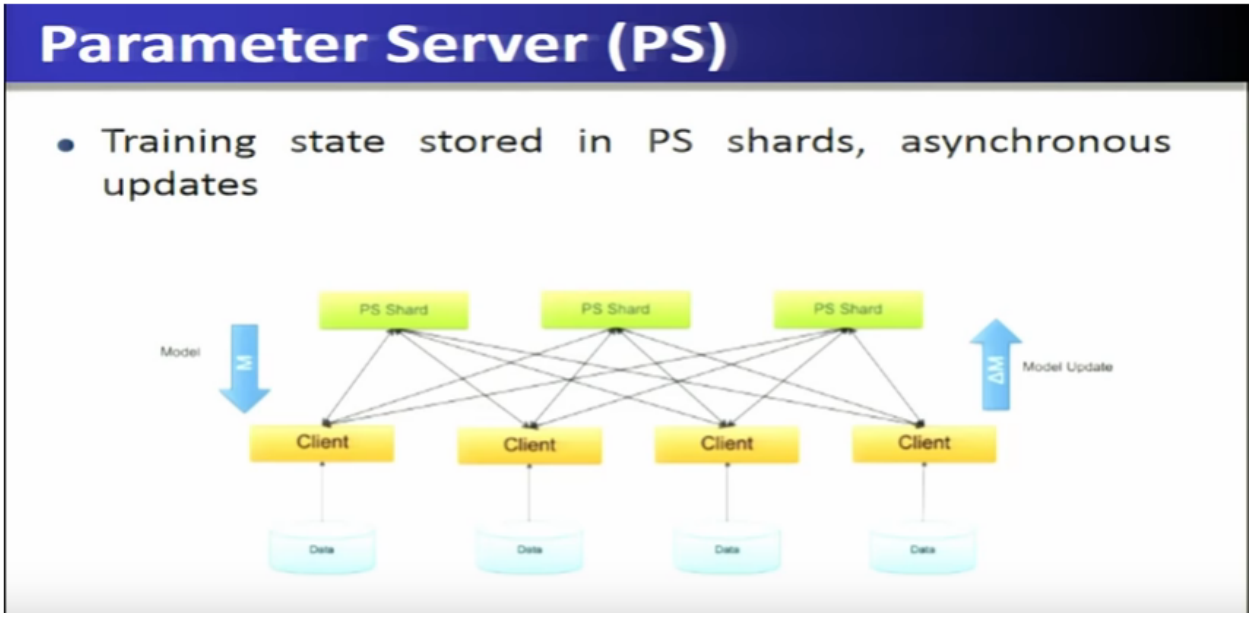
means, update part of the model. So, in any machine learning update equation, which is shown here a, wi which is updated WI plus little delta is, to be handled into the model, by push that is in the form of plus operation. So, in a stochastic gradient descent and all in a collapsed Gibbs sampling for a topic modeling, this aggregate that is the push updates is done via the addition operation. So, whenever this addition operation is there, so these parameters are to be pushed into the model and that is handled using the parameter server.

Refer Slide Time :(3:41)



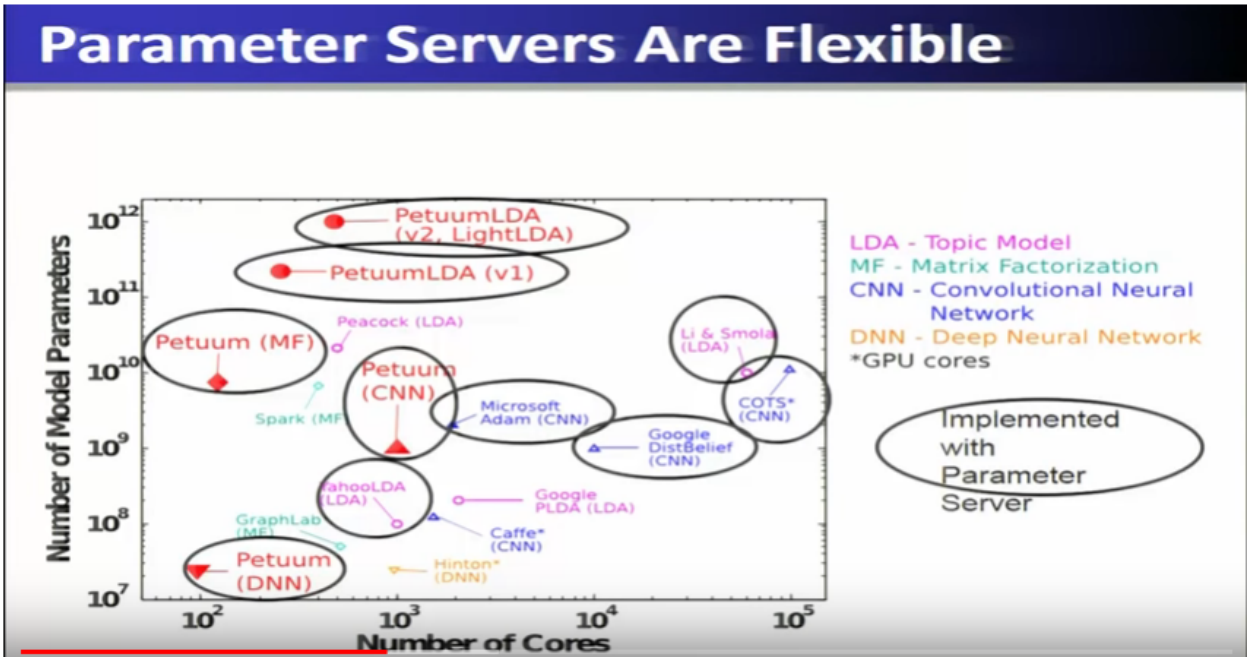
Now, this spark machine learning is done with the parameter servers. So, here in this particular scenario, we see that spark driver and spark workers, the together work and support this parameter server notion. Similarly, this kind of parameter server notion is also supported in the glint and in the glint, we can see there is a spark, there is a parameter servers. And in a spark there is a spark driver and different spark workers are there. So, we can see here in this particular example that, that these parameter servers are either maintained in the glint, as the queues, as the distributed queue and in the spark, as a and Driver worker combination.

Refer Slide Time :(4:38)



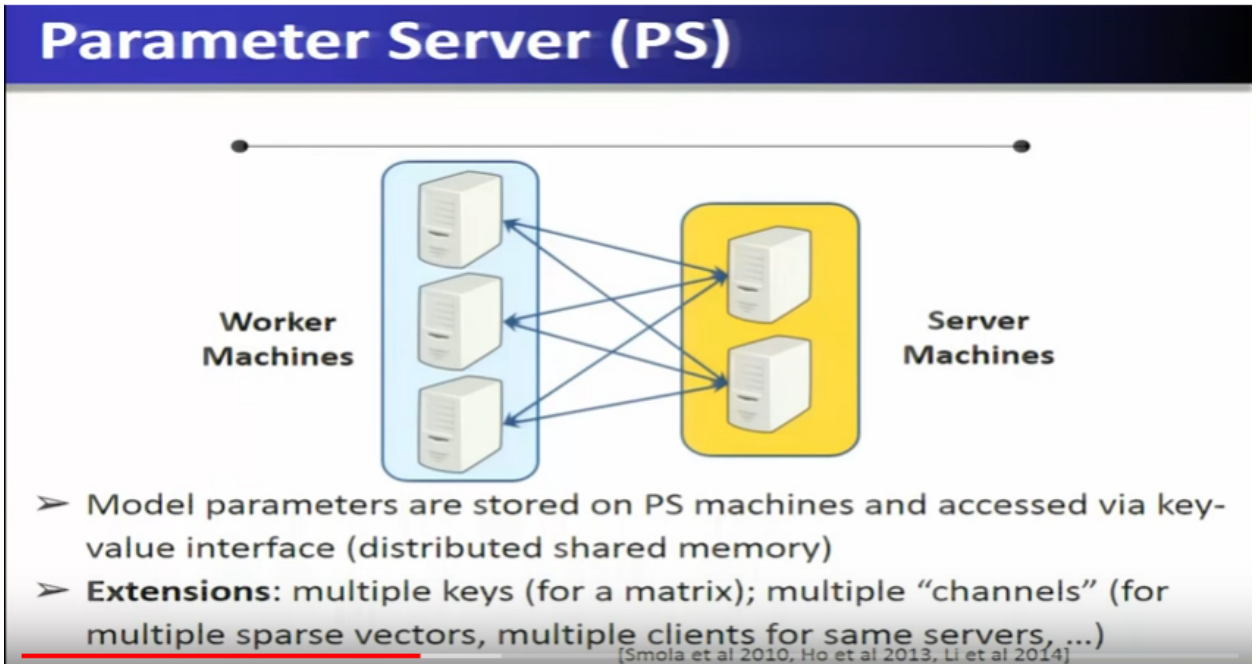
So, parameter server is used in the training phase also, so in the training state is stored in the parameter shards and asynchronous updates are being handled .So, this is the example, figure which shows that, the, the models are stored as a part of parameters shards, parameter server charts and whenever, there is an update. So, it has to be updated at all the shards which is, which is managing the parameter server. So, the parameter servers are maintained in this big data scenario that is through the, through the clusters that we are going to understand here, in this part of the discussion.

Refer Slide Time :(5:21)



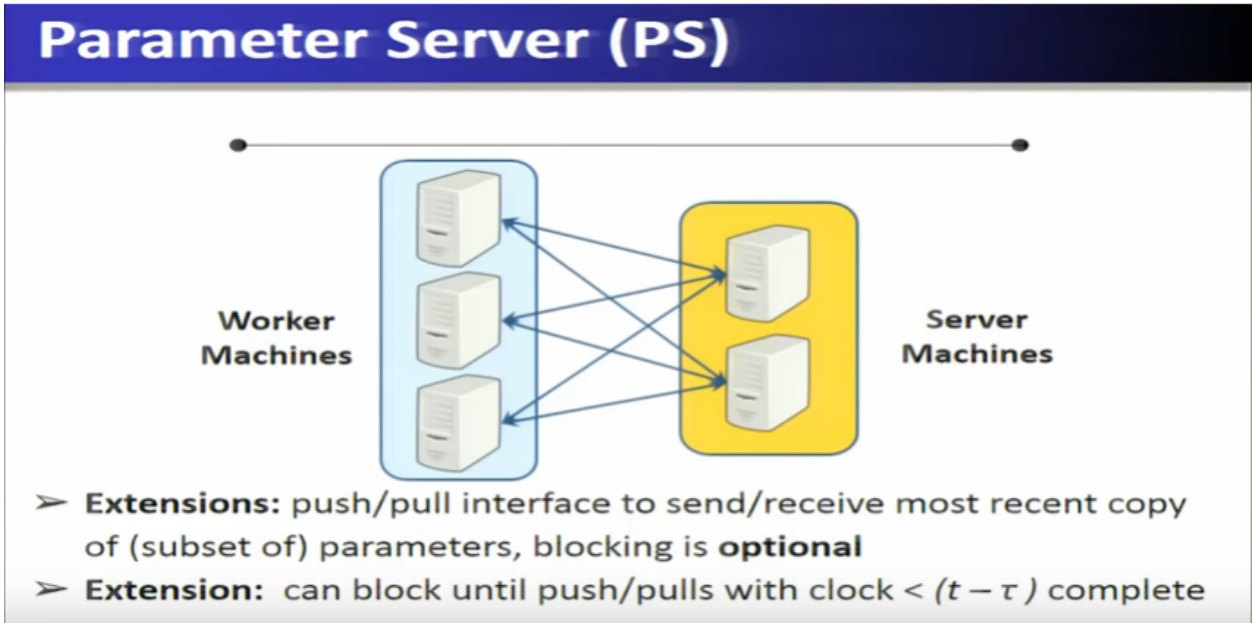
So, parameter servers are flexible. And here, the number of, number of model parameters which is being supported is quite large and number of cores, which are there which supports this, parameter server is shown, for different machine learning landscape. For example, topic modeling and matrix factorization and CNN that is convolutional neural networks and DNN deep neural networks, all requires flexible parameter servers and huge amount of parameters are complex parameter servers are being managed, efficiently therefore it is shown here, in this example figures.

Refer Slide Time :(6:07)



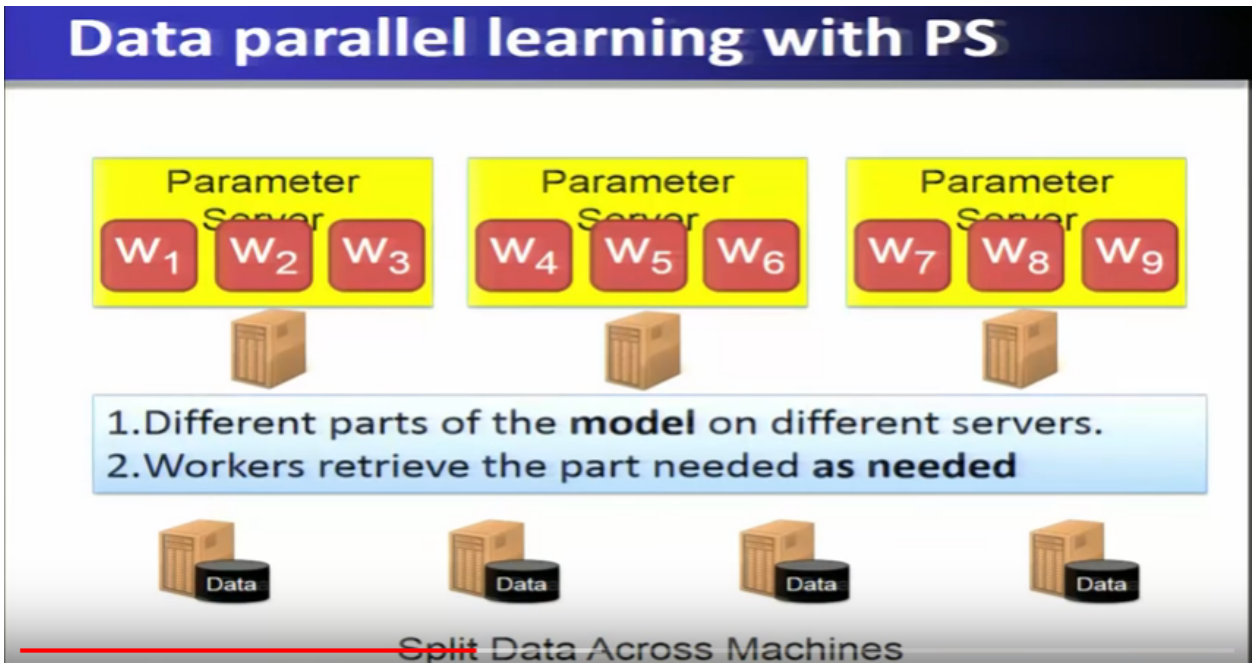
So, let us understand about the parameter servers. So, parameter server now comprises in its spark, prices of the server machines and the worker machines and it models the parameter servers, as are stored in the parameter server machines and accessed via the key/value, interface that is in the distributed, as per the distributed shared memory. So, extensions of it are multiple keys are available, for matrix operations and multiple channels for, multiple sparse vectors and multiple clients for the Sims for the same servers.

Refer Slide Time :(6:47)



So, the extensions which are available here in the parameter server is, in the form of push-pull interface, to send or receive most recent copy of the subset of the parameters and the blocking is optional. And the extension, can block until the push pulls with a, with a clock which is bounded by t minus rho is completed.

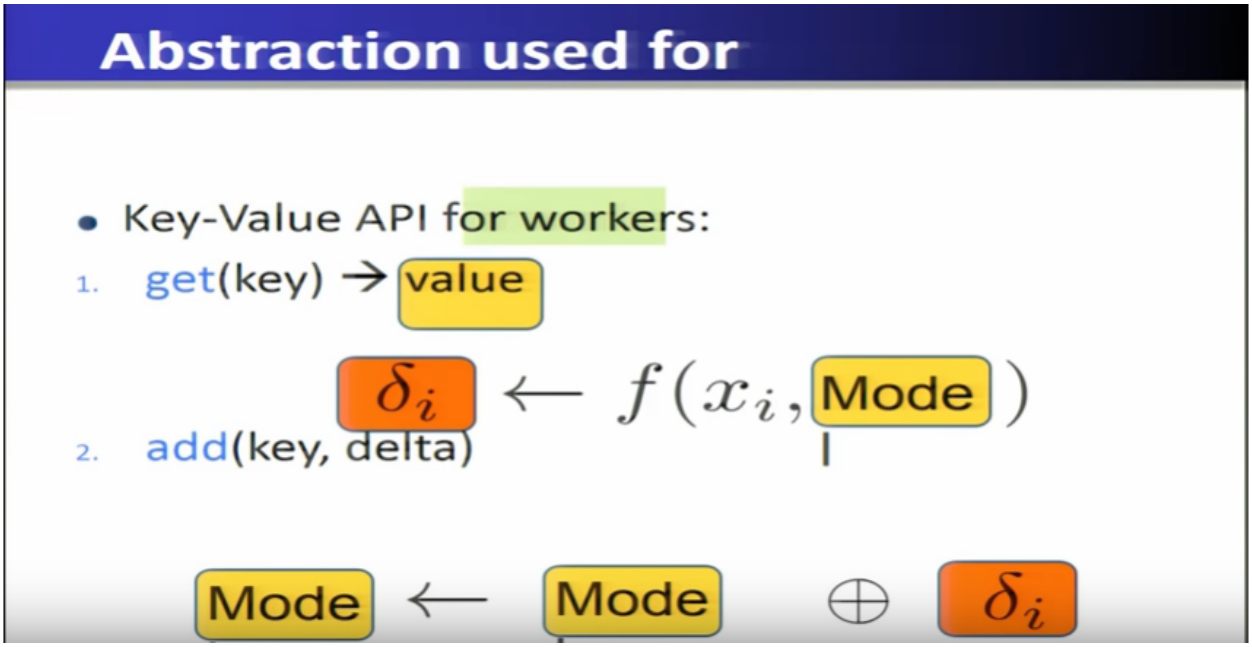
Refer Slide Time :(7:14)



So, let us see the data parallel learning with the parameter server. And here, the parameter servers are shown in this particular picture that they are, that they are being exchanged, through a different part of the

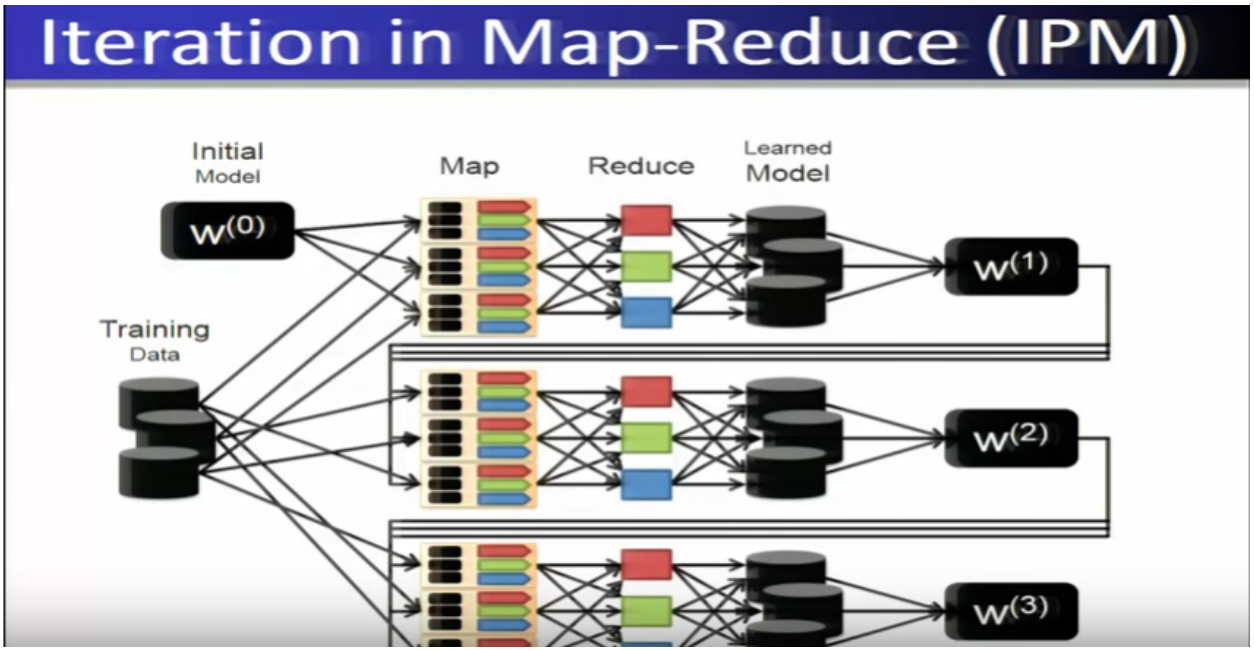
model is available, on different servers and the workers, they retrieve the part of the model, as it is needed.

Refer Slide Time :(7:39)



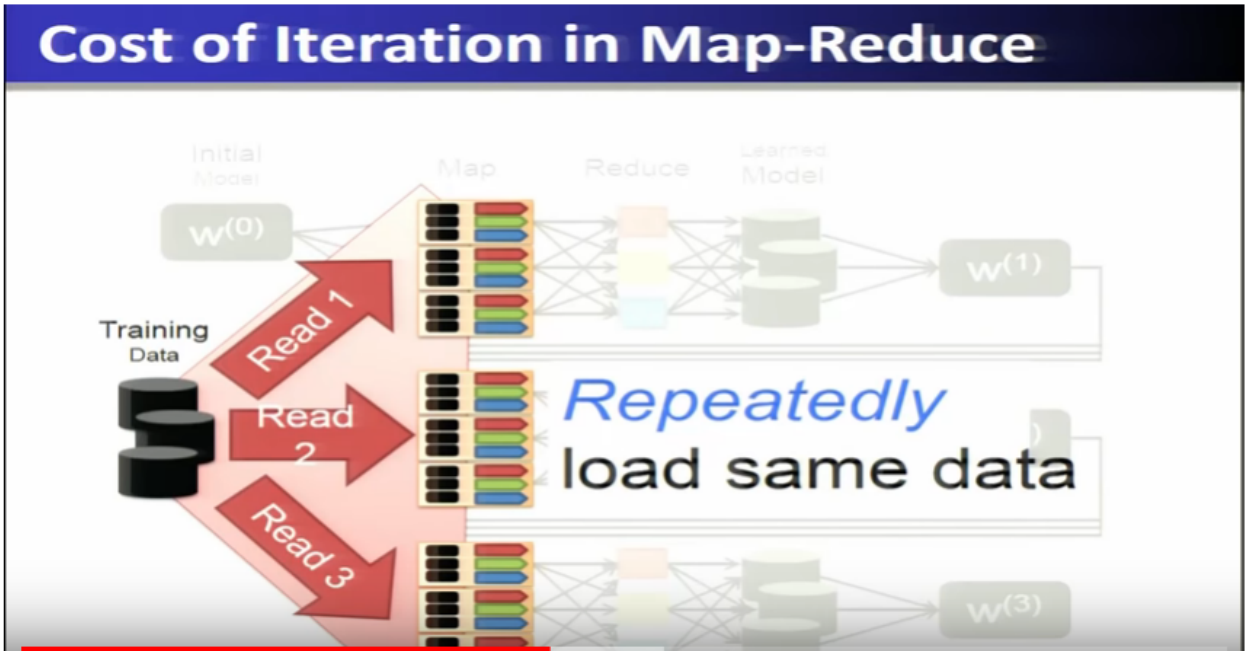
So therefore, the abstractions are provided which are internally implemented, as the data parallel operations in the parameter server, such as the key value API is for the workers that is they can get the key. And also, that means they may get the model values and by get key operation. Similarly, I had key with the wither Delta, to be updated in the parameter server and this particular addition can be performed here, using add operation.

Refer Slide Time :(8:18)



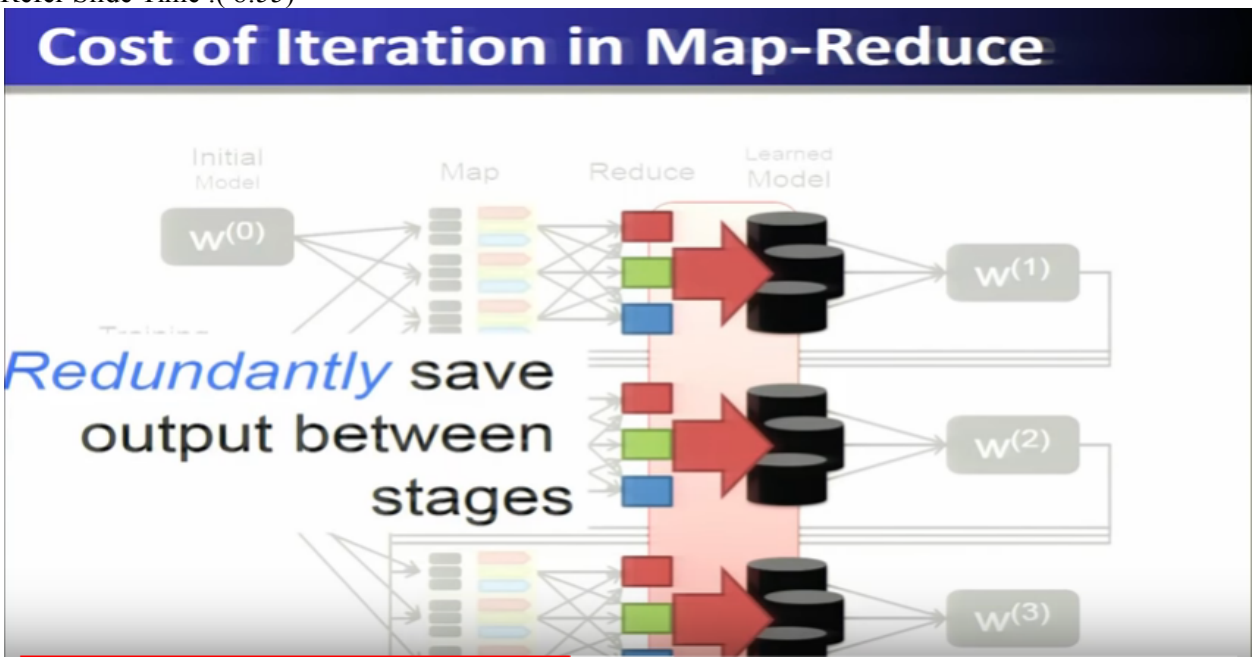
So, let us see how the iterations in a Map Reduce is supported, using the parameter server. So, this particular iterations in the Map Reduce is often required, to implement the scalable machine learning and this requires the training data, to be provided

Refer Slide Time :(8:43)



to the different map phase and which are to be read, in to different splits or in the form of the shots.

Refer Slide Time :(8:55)

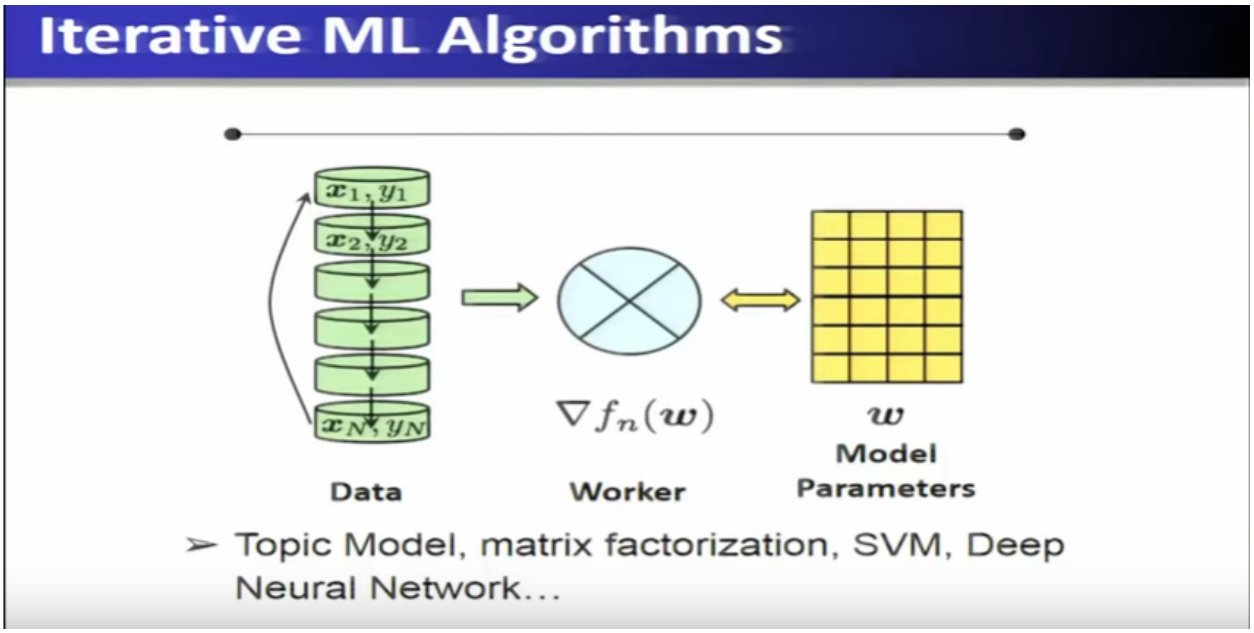


And then, they will redundantly save the output between these stages, in the Map Reduce function and therefore, increase the cost of this iterations of the Map Reduce.

Refer Slide Time :(9:10)

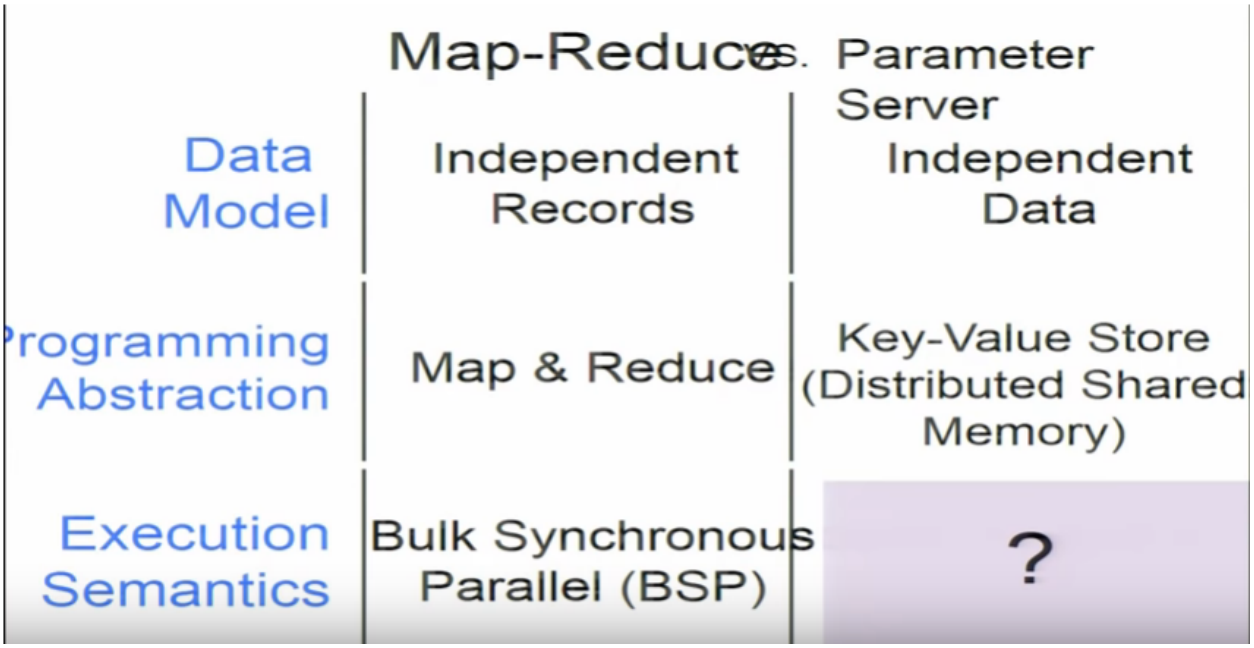
And therefore the parameter servers are very much needed, to make this machine learning, scalable machine learning every support. So, this particular parameter servers requires, still synchronous parallel model we will understand about them. So here, we have to see these model parameters are stored on the, on the parameter server machines and, and accessed via the key/value, interface using distributed memory.

Refer Slide Time :(9:42)



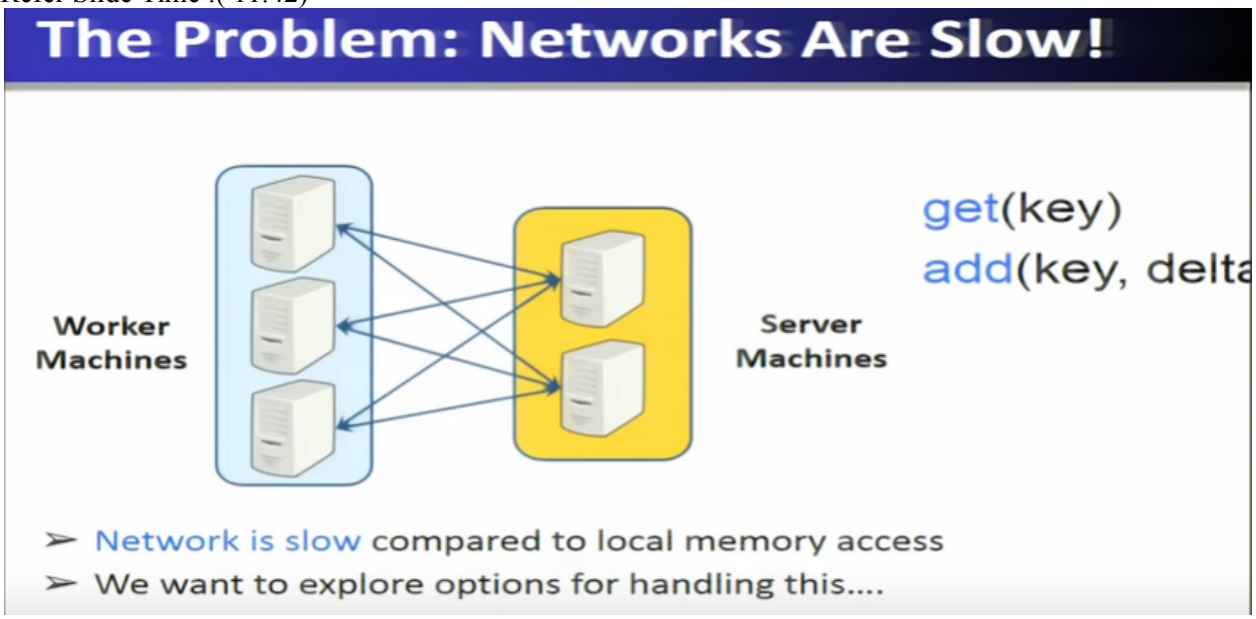
So, iterative machine learning, if we see the, the workload which comprises of these parameter transformations that is the data is transformed and by the worker nodes and in between, they are accessing the model parameters. And for example, in the topic modeling and matrix factorization, support vector machine and deep neural networks, they all are iterative machine learning algorithms, wherein the workers, requires these parameters to be updated, parameters means model parameters are to be, updated. So therefore, a parameter server has to maintain this model parameters. And the all the workers will now communicate with the model parameters server and these values are, either read or that means, they will either get or they will put these values on this particular parameter server.

Refer Slide Time :(10:52)



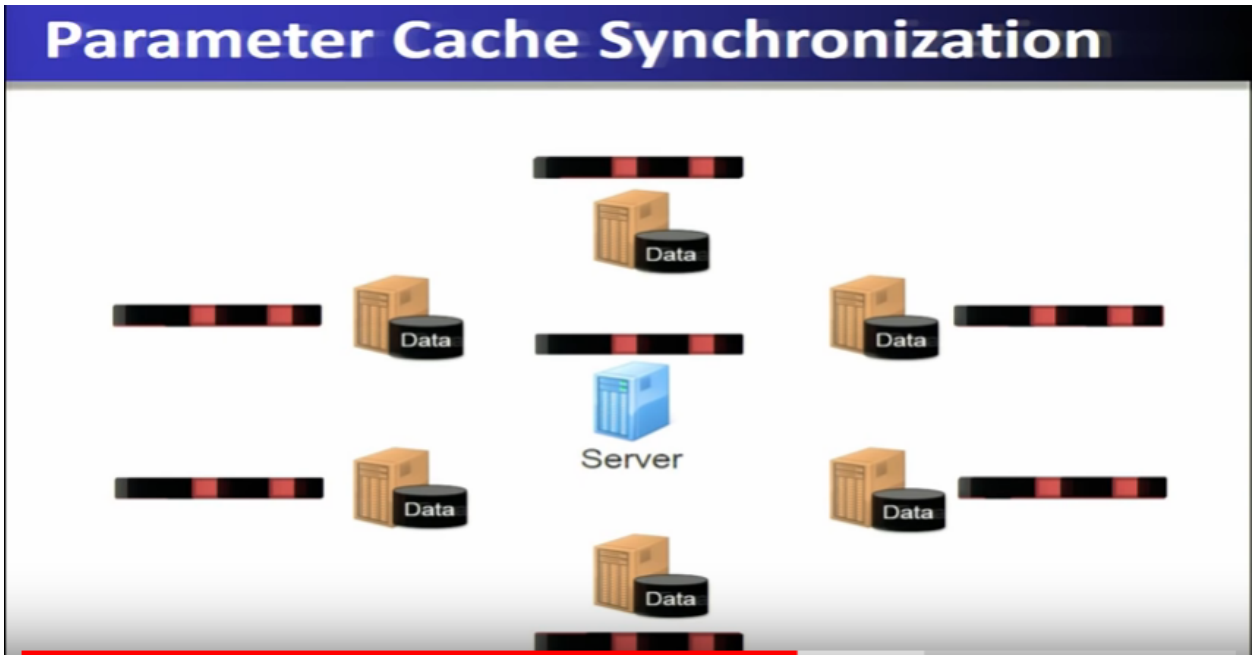
So, here we can see that in typical Map Reduce model, where the data model is the independent records and the programming abstraction is in the form of Map Reduce and the execution semantics, which is used as bulk synchronous parallel, in the Map Reduce to support the map, to support the iterative machine learning algorithms. Now, if we compare this with the parameter server here, the data model in the parameter server is also independent records and the programming abstraction is not the Map Reduce, but it is simply the key value store that is in the form of a distributed shared memory to be provided. Now, what is the execution semantics is it, the bulk synchronous parallel or some other synchronous that we are going to see.

Refer Slide Time :(11:42)



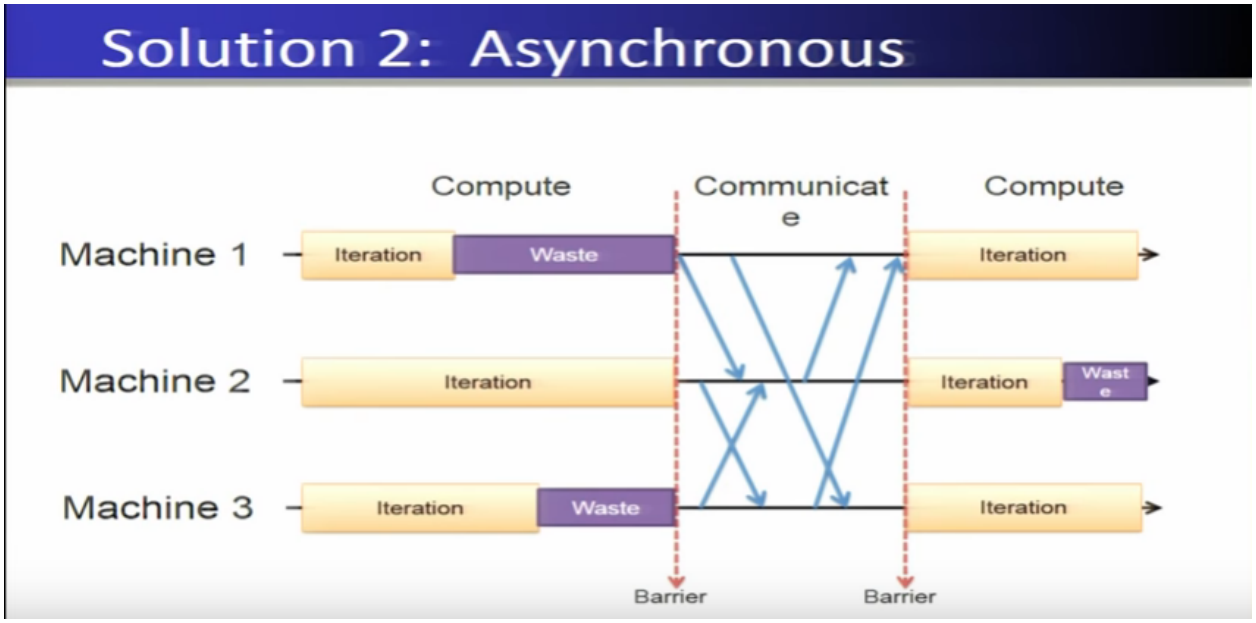
Now, the problem here is that, the networks are slow. That means, when these parameter servers which are maintaining the parameters, of the model and using gate and adding the keys are the operations, which are to be performed on the network, there becomes quite slow. As the network is slow compared to the local memory axis. So, we want to explore the options for handling, this with the help of the caching.

Refer Slide Time :(12:09)



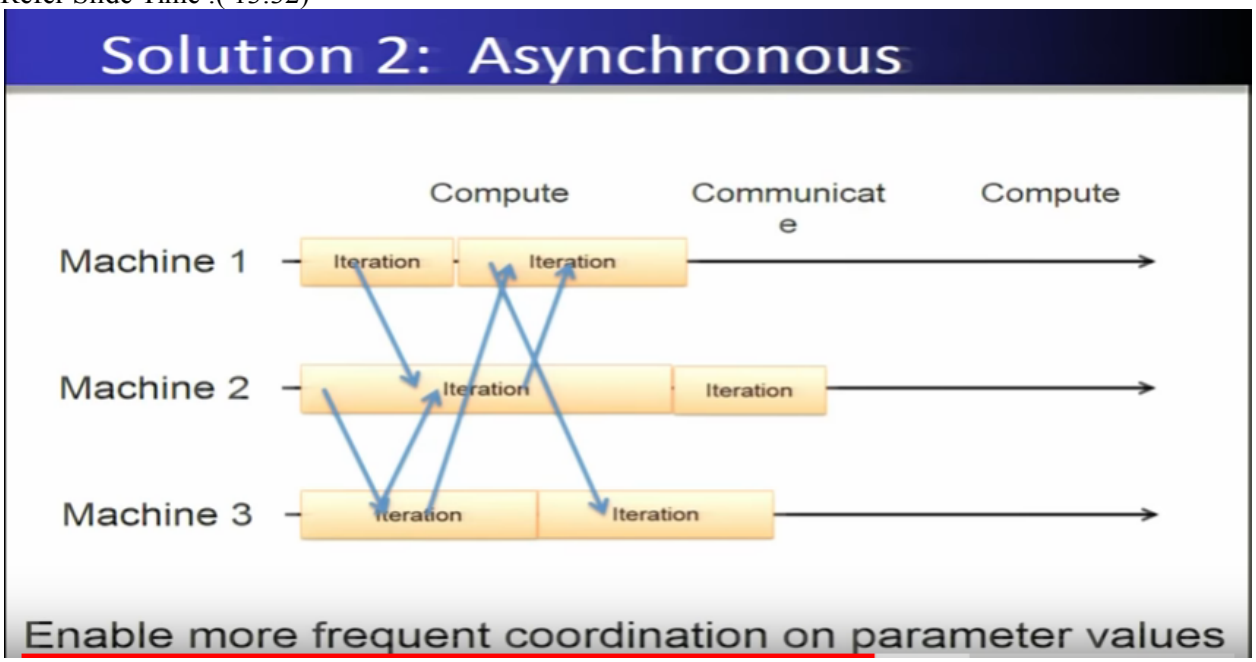
Now, the caching will introduce a problem that is to be handled using, the cache synchronization and here we can see that, the, the, the cache which is maintained in and whenever sparse changes, to the model is happening and repeat is done through the cache. Then, these parameter servers requires this particular cache to be synchronized that is called, 'Cache Synchronization'. So, having implemented the cache synchronization, in the parameter servers the problem of accessing, the parameter server, across network also, can be become efficient if the access is done through the cache.

Refer Slide Time :(12:57)



The second solution, to implement the efficient, access of the parameter server on the network is, about the asynchronous operations. So here, the machine one, machine two, machine three, they will perform another computation, they will iterate and now they have to wait, till others, other iteration they complete their, their iterations. And then, they will interchange their communication and then, the computation can go on.

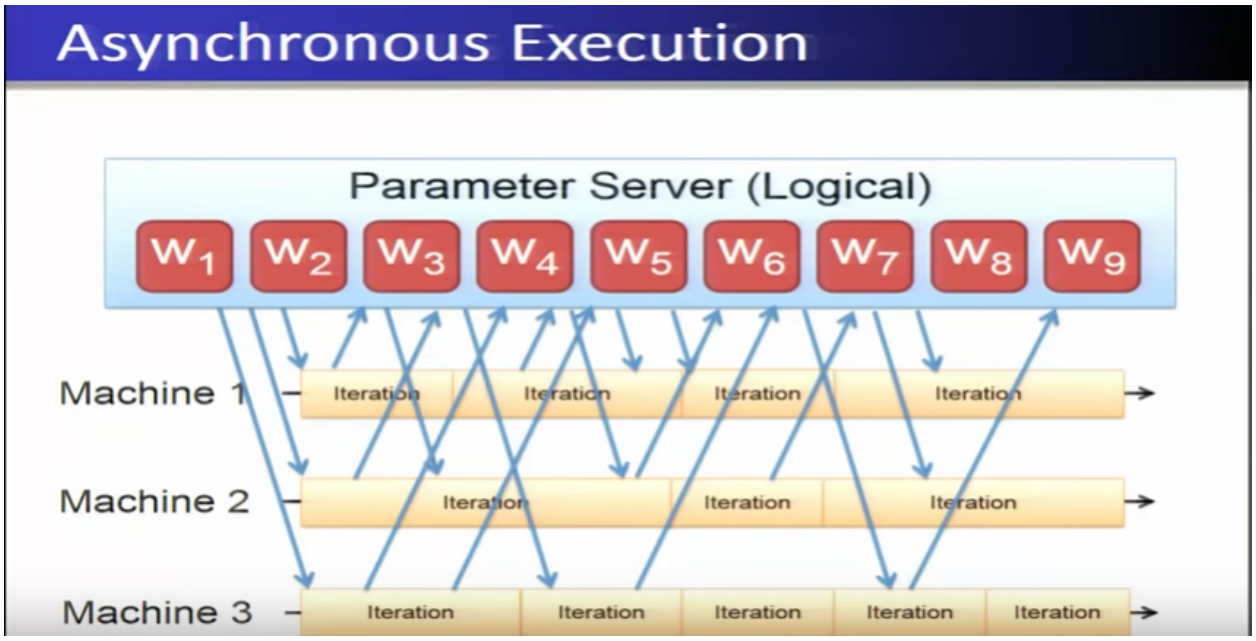
Refer Slide Time :(13:32)



,

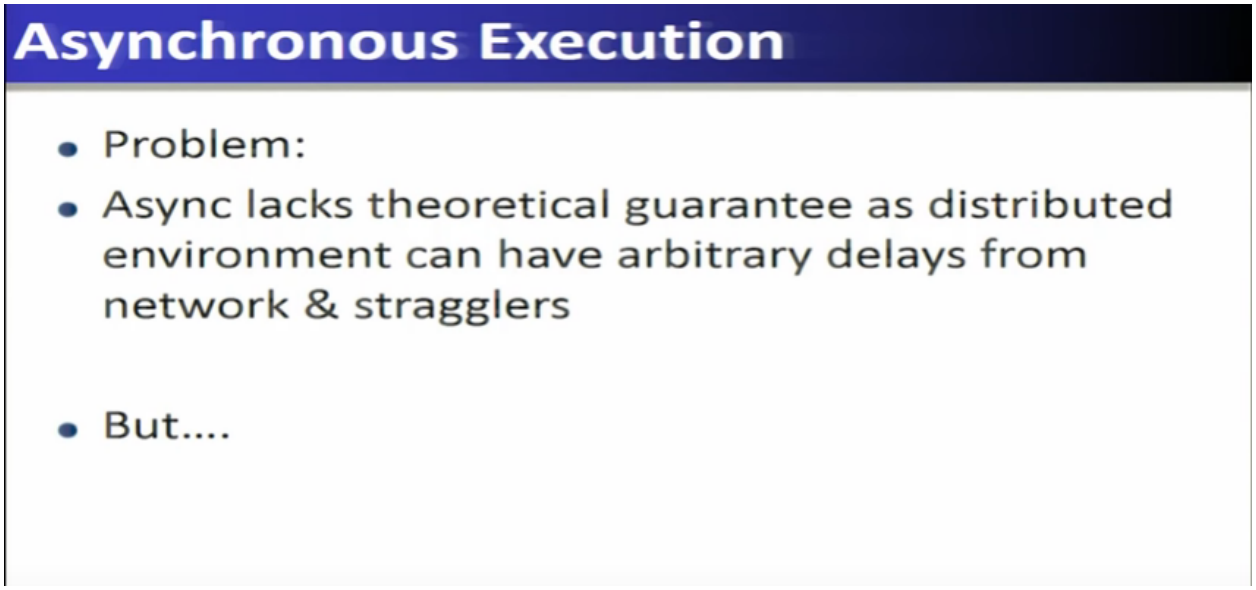
So, the in every iteration we see that, there will be a barrier, after the machine finishes the iteration and start the communication and before, all the communication completes, so that the next iteration will begin and these are the barriers and if we collapse these barriers, now we can introduce a much efficient way, of implementing this a synchronous, asynchronous here in this case. So, enable the more frequent coordination on, the parameter servers.

Refer Slide Time :(14:15)



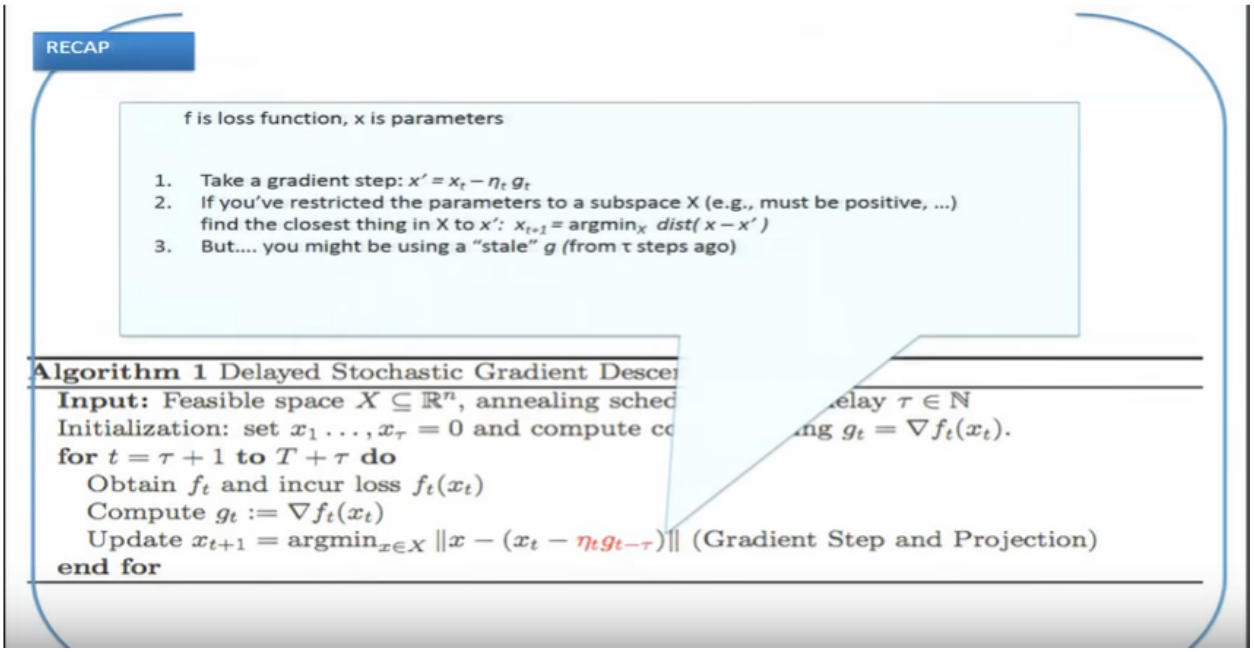
So, if we support this asynchronous execution, environment with the parameter servers, then the, the scenario will look like this that, the efficient, the, the operations in the parameter server will become more efficient. For example, here the machine one, machine two, machine three, they access different parameters W 1 and W 2 and W 3, from a different parameter servers, which are shown logically W 1, W 2, W 3, they are different parameter servers. And they keep on, accessing by different machines and they internally they communicate, the results back to this particular server.

Refer Slide Time :(14:58)



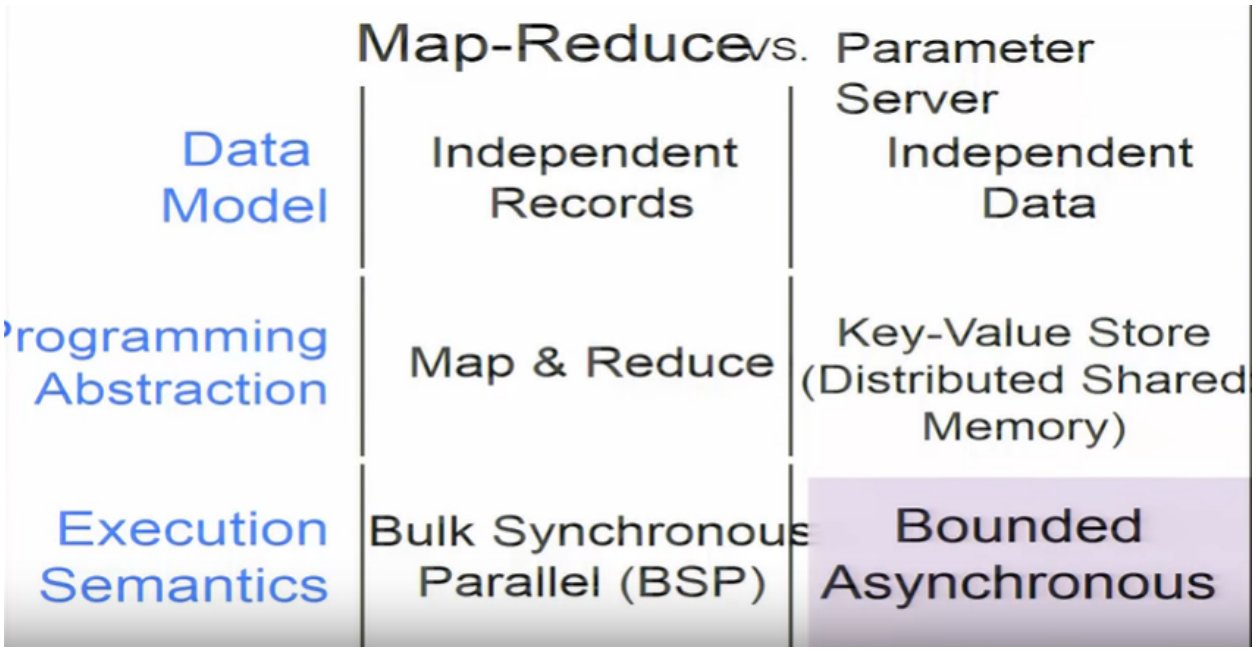
So, this asynchronous operations are, if it is there then, let us see the problems, in this asynchronous execution. So, asynchronous execution lacks the theoretical guarantees, as the distributive environment can have, arbitrary delays from the network and stragglers.

Refer Slide Time :(15:16)



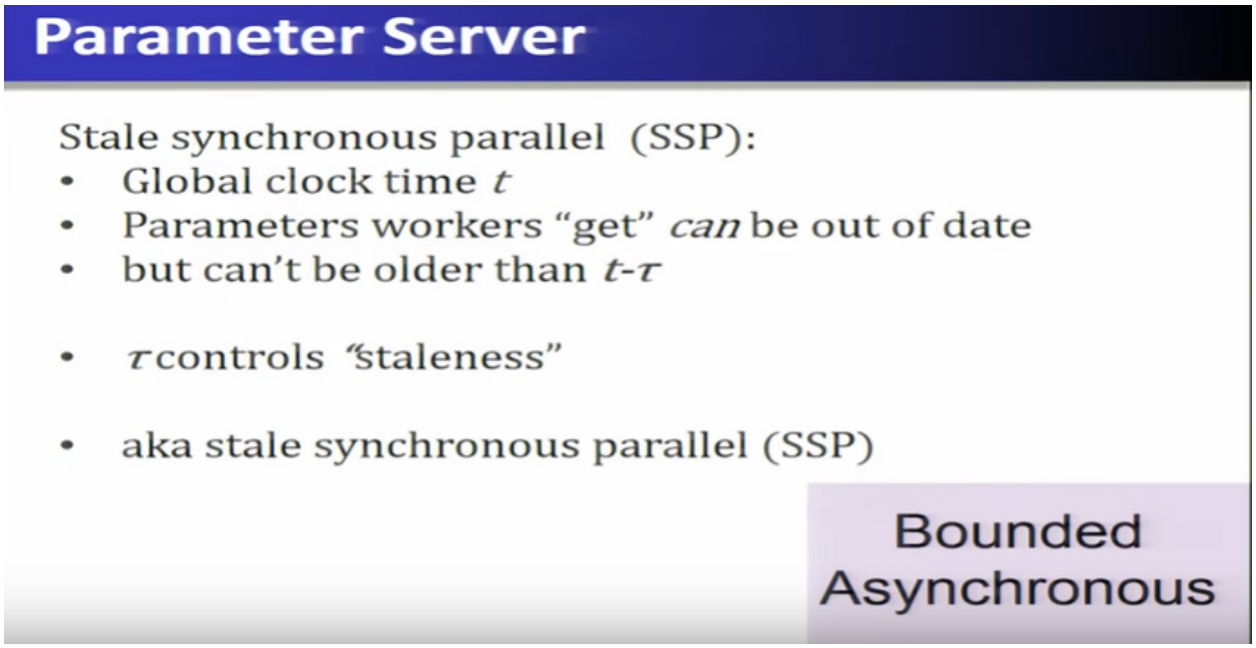
So therefore, what we see here is that, if we see a particular implementation of the scalable machine learning algorithm. Let us say that, we say the delayed stochastic gradient descent algorithm, here we will see that, F is a function loss and X is a parameter, which is being computed in this algorithm and which required to be updated. And this particular updation.

Refer Slide Time :(15:51)



Let us see what problem will create, during the using asynchronous communication. And therefore, we require to make about execution semantics and decided so that, this particular a synchronous, operation should not be a bounded, unbounded. So, it has to be a bounded synchronous operation, which has to be supported.

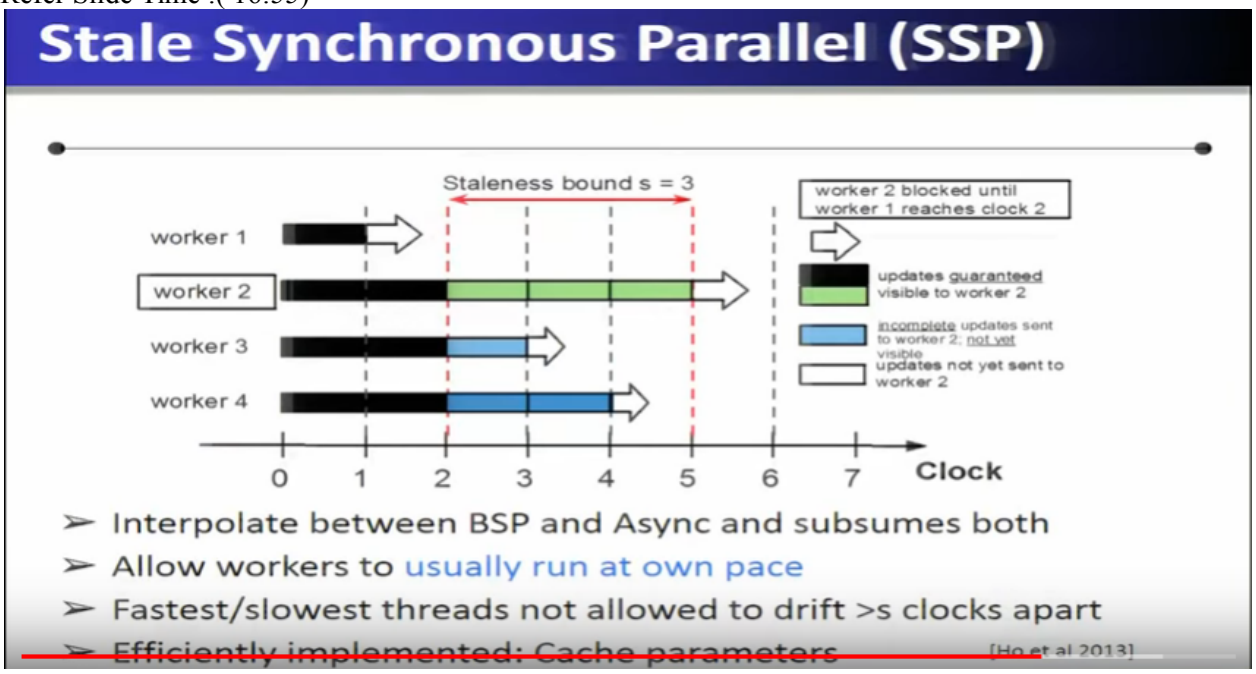
Refer Slide Time :(16:14)



So, the parameter servers, will use the model which is called, 'Stale Synchronous Parallel'. Model where in the global time is let us say, T and the parameter workers, will get and can be out of the date, but

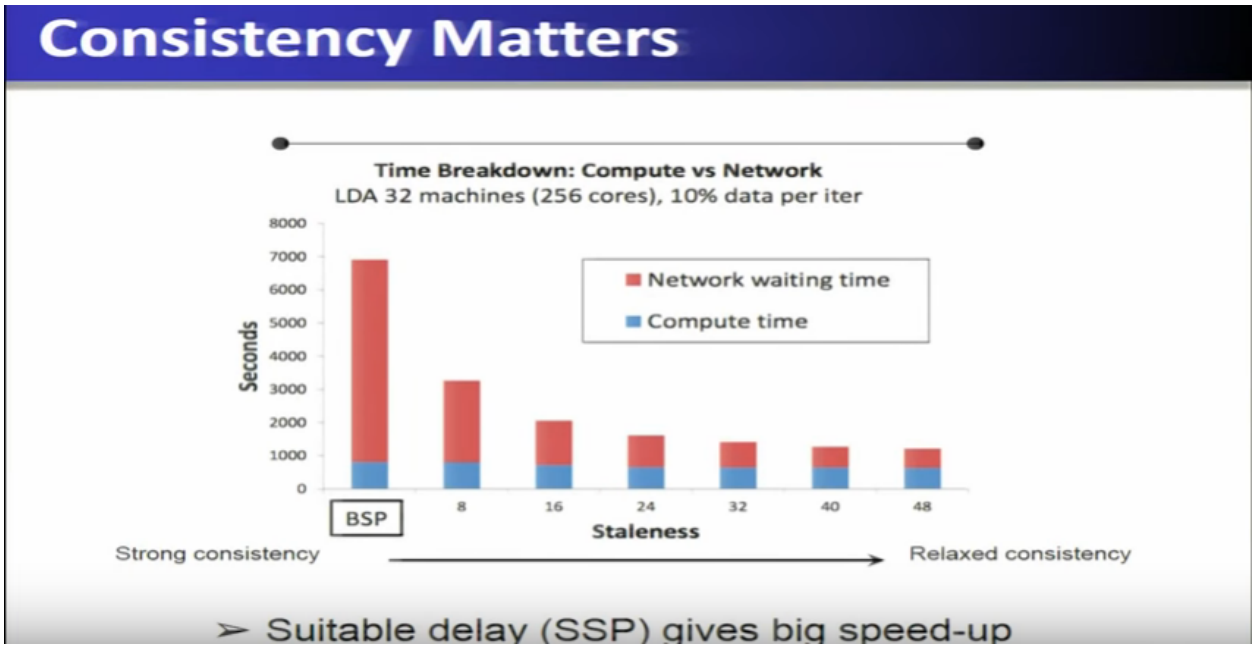
cannot be older than t minus tau. So, tau is the tolerance, which is introduced and therefore, it is called, 'Bounded Synchronous'. So, tau controls the Staleness, also called as,' Stale Synchronous Parallel'. So, stale synchronous parallel, execution semantics. We are going to see here.

Refer Slide Time :(16:55)



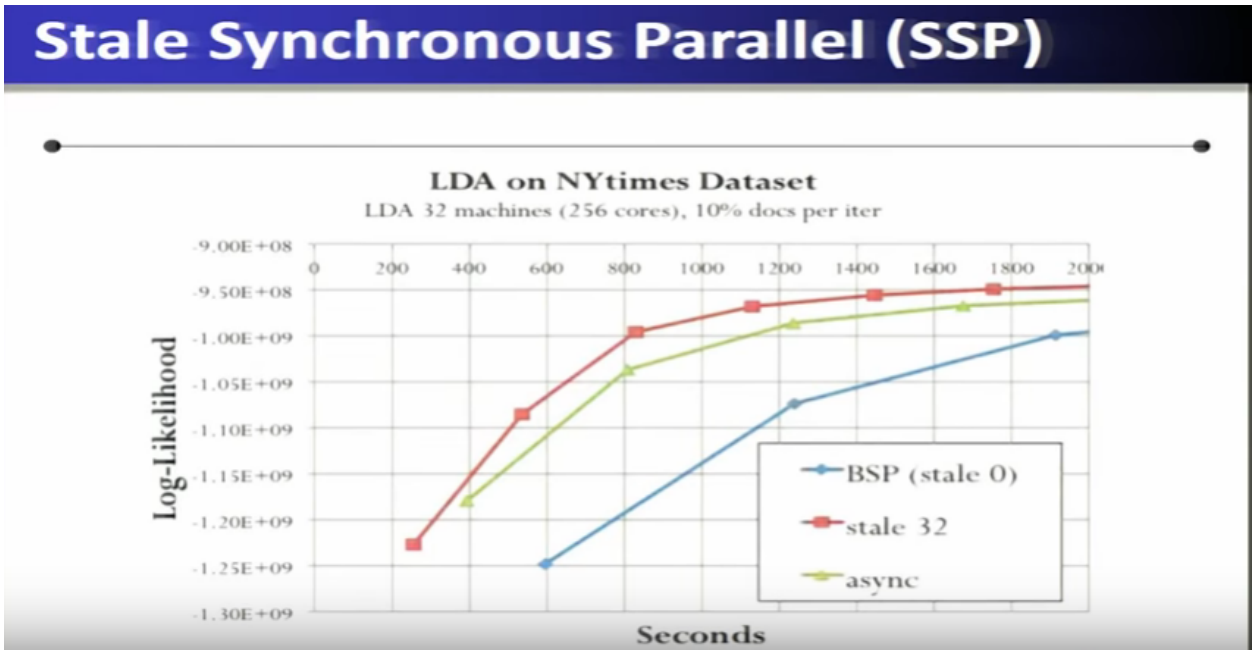
Which is being supported in the parameter servers? So, in the stale synchronous parallel, we see here is that, they will these are workers 1 ,2, 3 & 4. So, these workers now, they will be changing the parameters, as synchronously and the stillness is also, a bounded by s is equal to 3 that means, now the here we can see here is that, the worker 2 will be blocked until, worker 1 reaches the clock of, clock of 2. And so that means, the updates will be guaranteed and this is black means, updates are guaranteed and green means it is visible to the, worker 2 and blue will indicate that it is in complete. And incomplete updates sent to the worker 2 and not yet visible, updates not yet sent to the worker too. So therefore, this particular slots which is divided and stainless bounds s is equal to three which is being calculated. So, this way the interoperate, interpolate between BSP and the as synchronous modes and they subsumes both. Allow the workers to usually run at their own pace. And the fastest, slowest threats not allowed to drift ,the clock UI that means more than, s clocks apart. So, efficiently it implements the cache parameters, in this particular manner.

Refer Slide Time :(18:42)



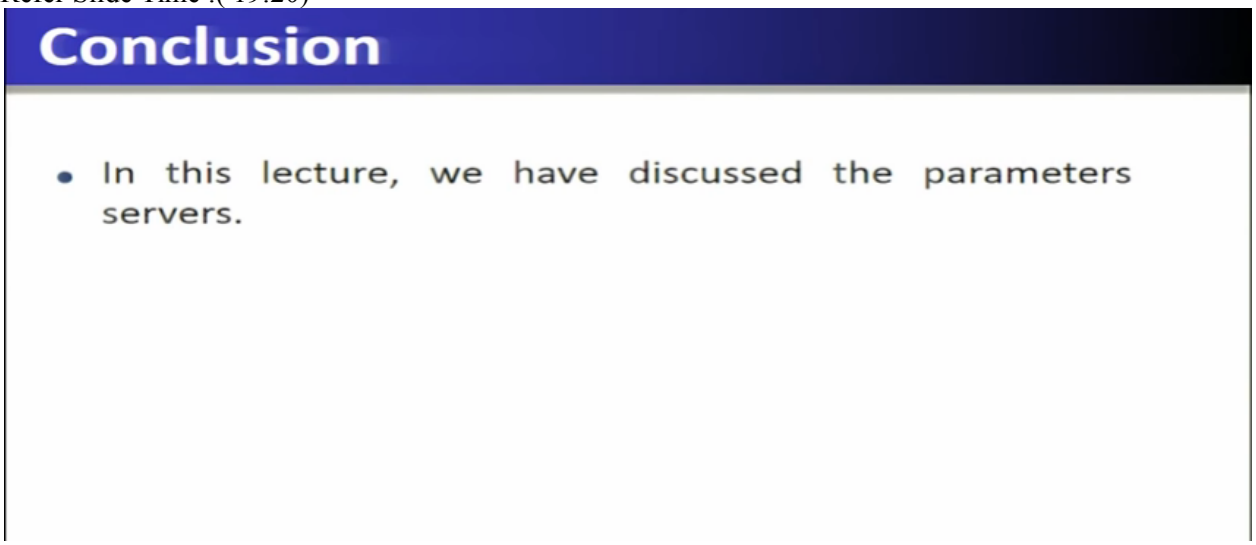
So, consistency here matters. Because, staleness has to be bounded by BSP, which is a strong consistency but we relaxed the consistency. And stainless staleness is bounded, so therefore, suitable delay will gives a bigger speed up here in this particular case.

Refer Slide Time :(19:08)



So, it stale synchronous parallel, if we introduce then we will see that, it is, it is giving the better result. And ensuring the problems of the asynchronous operations.

Refer Slide Time :(19:20)



So, in this lecture, we have discussed the parameter servers. And the is stale synchronous parallel model of operations, which is supported in the parameter server. Thank you.