

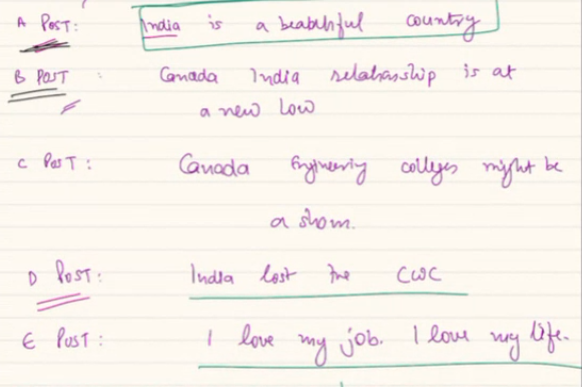
S3: file storage, to store big file, image, video. We use blog, file store. AWS has similar product called AWS S3.

We first discuss elastic search remaining topic..

**Elastic Search:**

Apache Luecene a product helps to build inverted index. If I have input as a document, I do stop word, steming, tokenization. This is for 1 document. U can do this for all document. Each doc can be converted to vector. If I have 100 / 1 mill documents. Doc = linkedin me post. Here each doc represent LL post. In google search each doc are web page that google has indexed. For facebook, doc = posts made on FB.

Now I have a doc which is paragraph of content, based on that I will create vector of this paragraph. A vector of a paragraph will look like this:



When I convert, I will get keys like, india, Canada, beautiful etc..

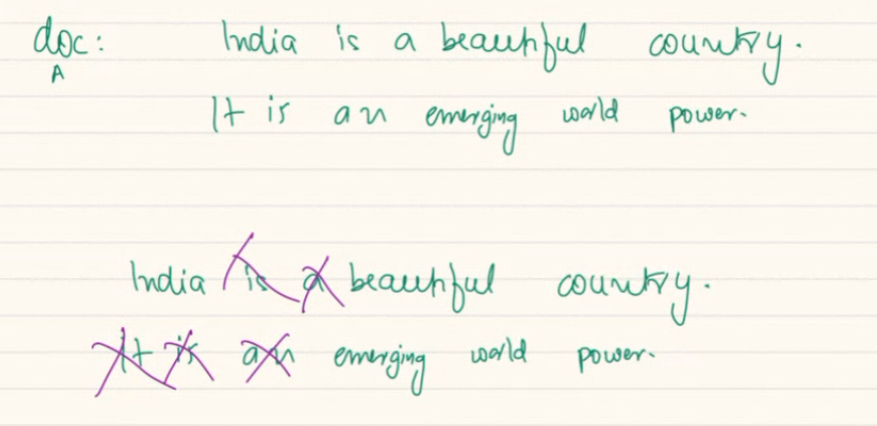
Same way we get BI-grams..

After tokenization whatever token you have will be token of my vector.

Lets say I have a document which contains:

Doc A: India is a beautiful country, it is an emerging world power

Using Lucene will convert this doc to vector.. 1st it will remove stop word..



We have some terms which are important.

Next will STEM them, convert them to base form, after stemming I get

Beautiful = beauty

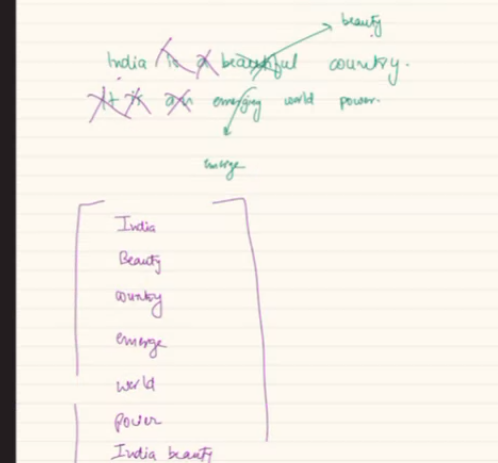
Emerging = emerge.

After that I get 1 or collection of 2 words or 3 words. Then I convert to vvector of tokens,.

Tokens are: India, beauty, emerge, world, power, beauty country, world power, then may be 3 words = tri-gram

The more combination you want to generation , you will have perform/ throw more and more computation.

You can decide whatever type of n gram you wantto created.



Again eachof these vector will get NLP SCORE. each word might come at diff location of the

If a paragraph, probably first 10% and last 5% are more importat. The way writer write some aspect are more important. Also some word are repeated more times.

Also how important a term is based on corpouse..

If I have million doc and each of them they have “the”.. that info is present in my doc. Common in every doc. So the imprtantnce of this is very low. As its aspected in everywhere.

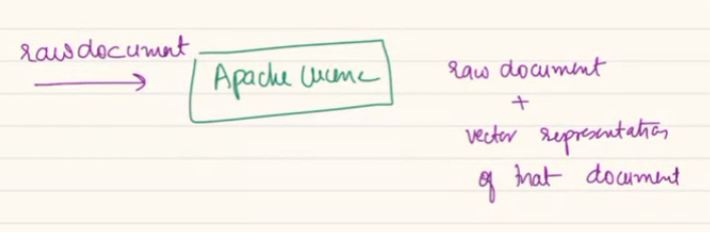
Same way India will not present in all document. So India is important for my document. Rare word. My doc is depended on this word.

TF-IDF, noun-very, which part of doc it is in.. freq of the word…… each of the word in the vector can be given a score… all of these APACE LUCENE can do for me..

Highest score word are more important…



If I have APACHE LUCENE, I input a raw document inside APACHE LUCENE, I will get a raw document, alongside with raw doc, I willalso have vector representation of that DOCUMENT.



SPLUNK is also similar.

keys of vector are something to recognize the context of documents ?

when build , I try to convert English to some mathematical formula..

LL = large language model

Lucene can create a representational model of it..

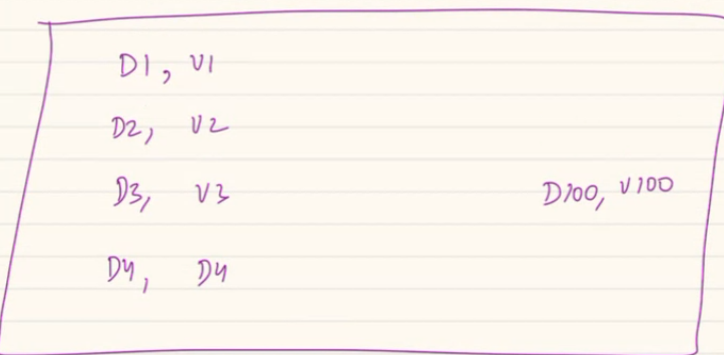
IF I want to build a system where I want to search across million documents. Say 100 doc I have, that I want to Index.

D1, D2, D3……………………………………………….D100.

We will use Lucene, where we will input each documents, Lucene will give me vector representation of these documentation.

D1🡪 v1 vector representation.

So I have a corpses of these doc and I also have vector representation of all these documents.



Lucene will create Inverted INDEX..

When I have doc and VR. I f I can combine all vector together. I will get a superset of possible keys.. 7 of the doc will have India, country will be present. When I take union of each of these vector I will get longer superset of keys which I get after performing union of all keys.

When I union all doc I will get a inverted index.. like below

Lets say we are doing unigram only:

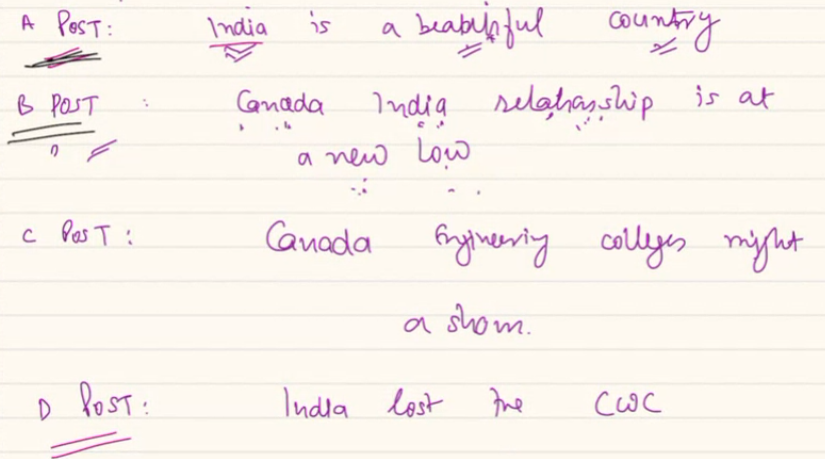
From doc-1: India, beauty, country,

Doc-2: Canada, relation, new, low

Doc-3: engineer, college, sham,

Doc -4: lost, cricket world cup.

Doc-5: love, my job, life..

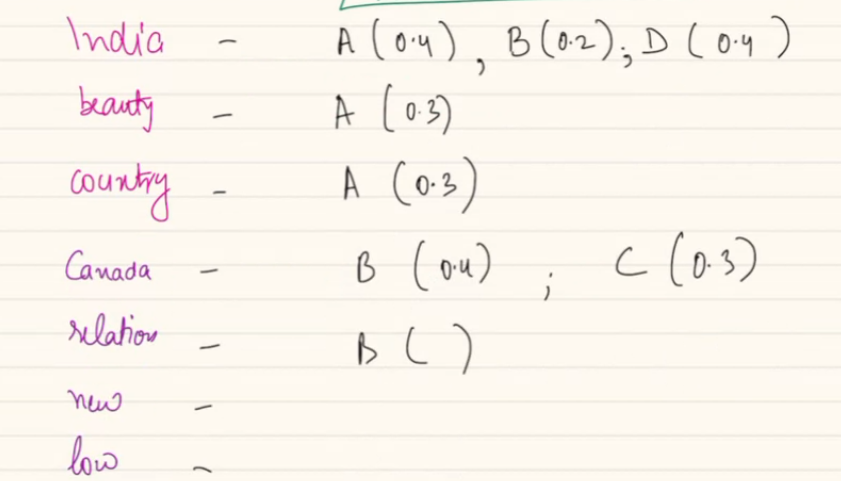


These are all the elements inside the inverted index…..

We have combined all vector we got from all doc… Now I write like a glossary of a book… at the end of a book, which tells you which all page no these words come iside. Same way.. india will come under 3 documents..

India: A, B, D in all these document. Anlso the score of each documents.. SCORE falls between (o🡪1)

India: A(0.4), B (0.2), D (0.4).



Like combining hashmaps and creating a new one..

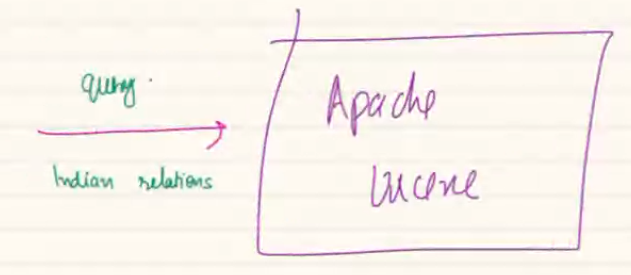
Here using all doc combined will created 1 inverted index. Represent collection. Vector represent 1 documet.

Many signal will influence the score. 1 means it has all significance. 0 means 0 significance.

here you did only unigrams right. but what about bigrams,trigrams etc? we will do bi, tri etc.

Once all these are done, when you want to search in LUCEN. Lucene is like BLACKBOX. When its done, I wantto send a query so lucene can tell…

Query is : “Indian relations”



Query will gets converted into vector form.. look like [india, relation]

Look at each term of vector, .. I

India: relevant at A, B< D

Relation: B is relavant.

Based on threshold will combine all these doc. Get list of doc in ranked manner.

B,A,D example.

I get a ranked list from a query..

Power of inverted index, by Lucene.

Extending this idea google search .

**Google**:

Say google Has 10,000 doc.

Use Lucene, for each doc create vector representation. Create inverted index.

When a query comes at me, I will convert to vector representation using NLP. Then perform a search not in all doc but in the inverted index…

Searching a Hashmap is very easy.. we just check if key exists we just get the document.

Base don search we get list of doc. Based on score, we get sorted list of doc. And give to user..

Here we have done some preprocessing, based on that every query will be answered by a HASHMAP, rather than text search.. answered a lot fats. As search happen in inverted index.

All NLP model there has the accuracy, precision, recall.. but keep improving the NLP results.

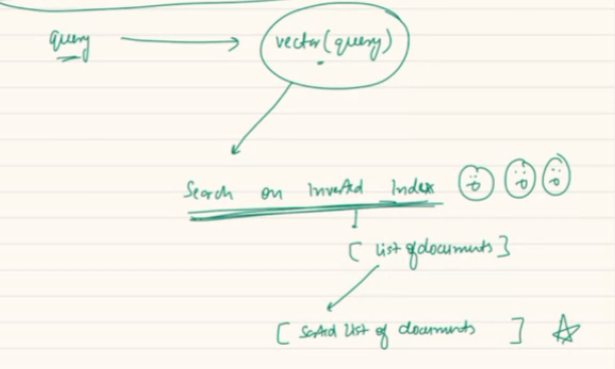
**Lucene will preprocess when??**

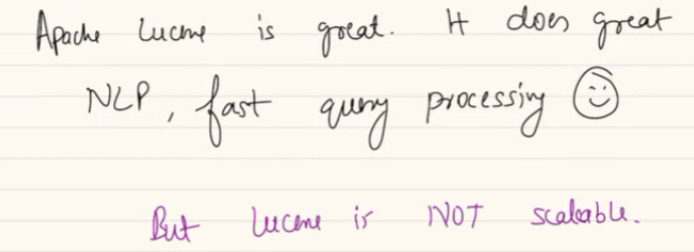
At the very first time, when I start LUCENE, I will have to batch processing say I have 10,000 document, I will do batch process. There after immediately I will create new vector when a new doc. Then I merge, add a new key or take existing key and add new value.

When new doc, convert into vector, merge, Ranking gets updated.

Hashmap are fast, as underlying DS is array, when we have token will they stored in array?

LL based or array based implementation. problem is how you build the HashMap lib.. use best technique.



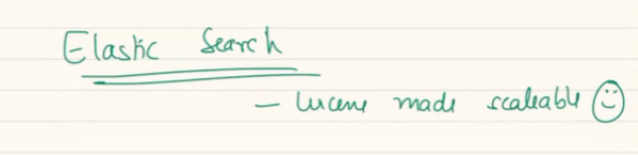


Lucene is a single machine setup, not meant to scale horizontal.. now will look at ELASTIC SEARCH.. nothing but LUCENE made SCALABLE.

**ELASTIC SEARCH:**

Elastic search is a product, has many offering, you can host or managed ES by AWS. Aws hosted elastic search, cost you monel.

You can use own, which is licensed.. less. But education me free. Different offering…



In google will have billions of documents..

I would prefer Availablity over consistency..

What kind of latency I want? Low, high, super low… = low latency, I cant keep on waiting but can take some seconds, usually the results are paginated, give result in paginated manner..

LOW LATENCY for getting search result. Its Read . low read latency. Can writing a doc take some time. We are fine with medium-high latency for writes.

PACELC says out latency can be low..design trade off..

**4. design deep Dive:**

Understand the APIs yourself.

IN ES like system you require compute resources because you want to convert a doc into vector, a query in vector, update,

Storage resourses: store raw doc, term vectors(vector represent of doc, ), store..

We have some machnes which will do compute for me.

Some for store document and term vector. Sharding is needed. All document cant fit in one machine.

Replication: we replicate so data never lost, DB fault tolerant, support higher frequesncy of reads, in master-slave replication I can support higher read. So in ES I want to have replication as well. It sread heavey, lot of data, so need replication.

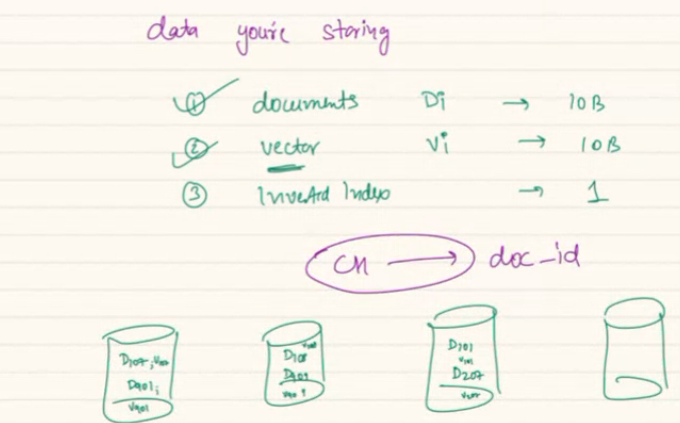
Sharding: what all data we are storing? Documents(Di), vector (Vi, 10B vector for 10B doc), Inverted index only one we will store.

How will we shard? We wont chop , devide doc into pieces, we want each doc in one single machine. I can use doc id as shard-id for every doc I can put in one shard.. if I store 107 in shard-A. CH puts lot of doc in a shard, same shard can have many doc.

CH can take doc id as hashing key..



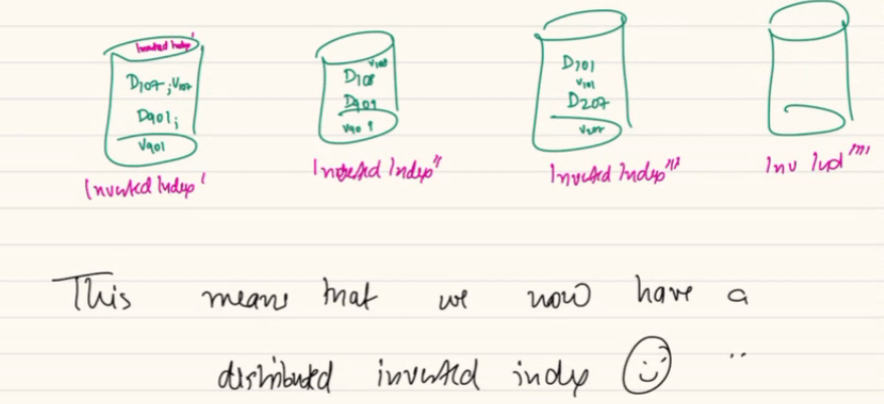
Should I send corresponding vector to the same shard.. so document 107 k sath vector 107 will also be here.. where you send doc you will send its vector also in the same machine. We will distribute the doc and vector by keeping D-V together.



We have one Inverted Index, i cant put entire inverted index in 1 machine.. we can shard the II across machine. Whatever II I can have for these set of documents that II can be here..

Same way whatever II we create because of D101, 207 we keep that II there.

II will be sharded in same manner, one II is devided across diff shards..



Now inverted index is no longer a II, it’s a distributed Inverted index.

Example:

From outside we geta new post, this post is a new document. When I have doc 204, wi will perform all steps so I have vector 204 is created. Based on post ID, doc\_id, I determine which shard it will be going to. Use post id to go to correct shard. In that shard I add document 204. I will also add the vector 204 in the DB.

Each doc is analogous to a post in LL, or page in Google. Vector will eb bigger may be comparable size even include bigram, trigram. Each machine will have inverted index.

Say India is at = d1, d5, d9

Country: d6, d10, d9.

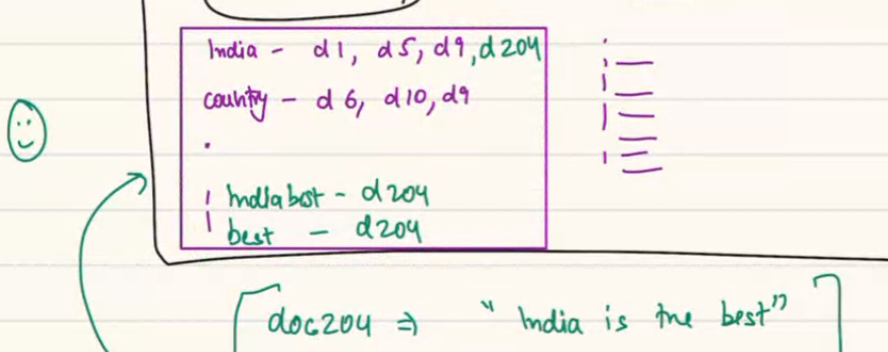
Say I have a doc204 coming, app server me you do computation. it has india is the best.. vector 204 will look like: india, best, india best….

Each of them will have some score.

Once I have doc. the II of that shard will egt updated.

If best not there, I add a new key add document 204.

This is how you ingest a doc here.



Other shard can have india, I wont change there, as that shard don’t have the document204..

ES = lucene on steroids,

here doc editing is not possiple...only new doc injetion rt? Not a use case.. if they update 204.. think about it.. go to corrct machine, remove doc, and remove the entry of II.. so update will also work.

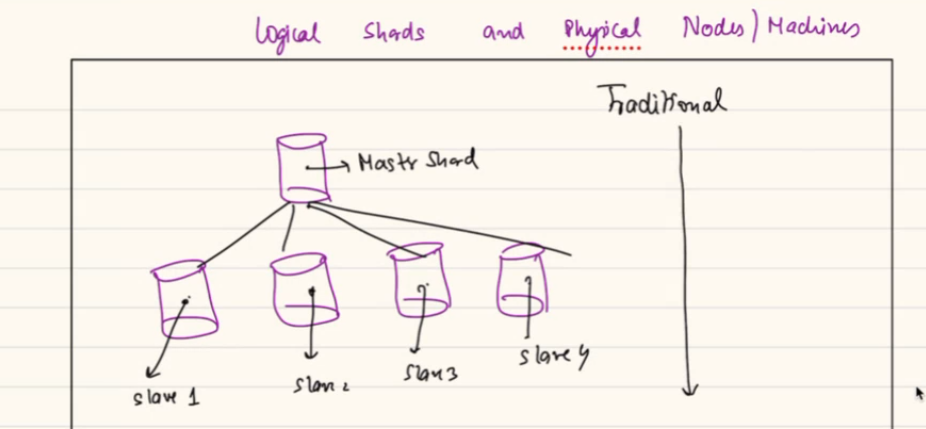
if document updated then vector will also need to be updatd right? Yes

**New Concept:**

Idea of Logical Shards and Physical Node/Machines. He shard a record class..

There are physical machines.

In traditional sense we understand these are physical machine. One act as master as master SHARD. Other acts a s slave-1, 2, 3,4



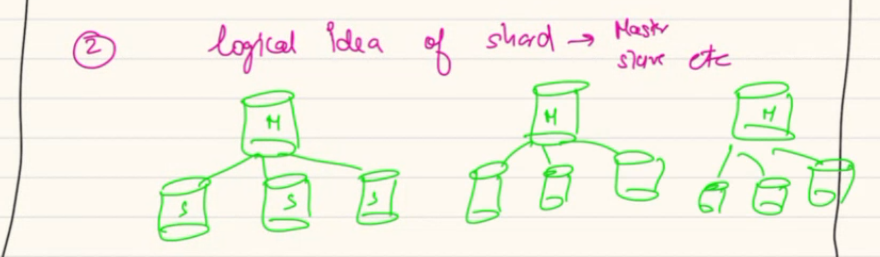
In traditional sense…one machine is going to have master copy, other machine will be independent slave.. logical node = same as physical machine..

There are 2 different concepts at play.

1. A physical machine, go and but/rent a physical machine… it can be master or salve but each of these are equivalent commodity hardware. Whatever you host that’s different.. this machine can be plugged in ..these are tangible machine you can touch and feel.
2. Other concept is LOGICAL IDEA OF SHARD, shard can be a master or slave… one actual machine can have lot of virtual machine running on top of that. Or container on it. So 1 single physical hardware can support logical instance running on that physical machine.

Traditional sense 1 to 1 relation.

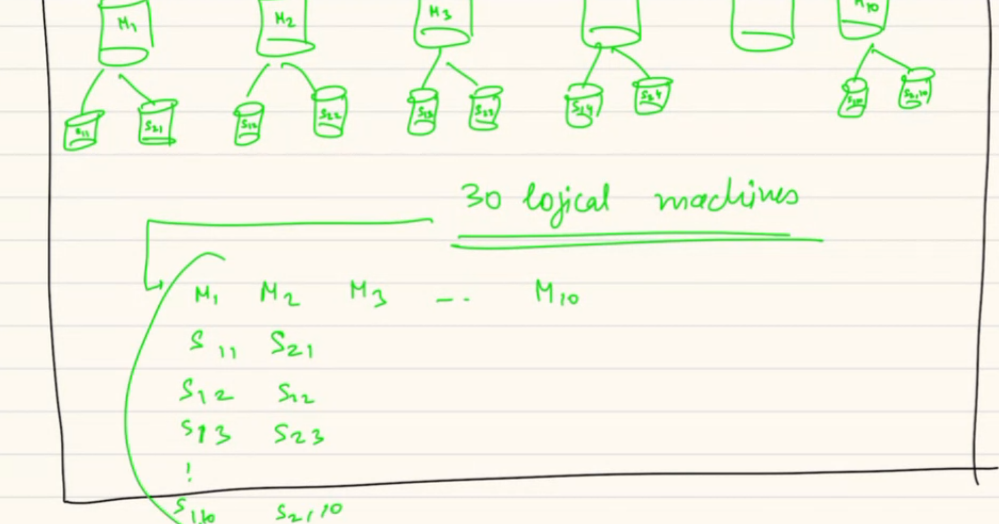
In nontraditional we have diff logical and physical entity.



I have eco system, has 10 shards.. CH ring will have 10 entries. For each of these shards I have 2 slaves.. so I have used 30 logical machines..

Master 1- 10 M1, M2, M3… M10

Slave will be : S11, S21, Slave12, S22.. slave 1 of master 2, slave 2 of master 2.



If every logical machine corresponds to 1 physical machine I wil need 30 dedicated physical machine..

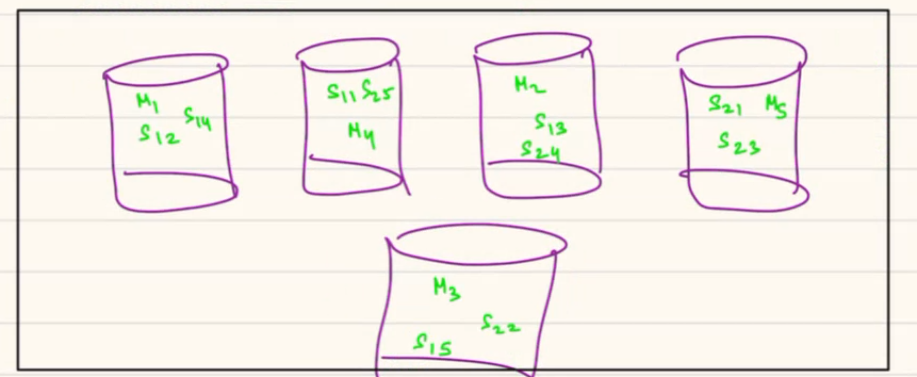
Reality is even if I give u a cluster of 5 physical machines.. I can still do it. As in a way each physical will run 6 logical shards… u can spawn shards as you want.. 5 machione me I am running 30 logical machines.

When u have too many logical shard in single machine, context switching and management will take time. Also cant have lot many logical shards.

Fail over strength I got that was because of physical machines.

But there might be 100 of logical machine spread across slightly less physical machine. It sfine.

Lets say I have 5 masters, for each I have 2 slaves each. So I have 15 logical shards.. in my actual cluster I have 5 physical machines… but total 15 logical instance runs across 5 of them..



This way I can make sure this master, that slave run here.. managed by zookeeper. Each of PM is node and each of them wil have logical master slave will be running.

Allocation strategy can be anything…. Here I try to reduce no of co-relation among cluster.

If 2nd machine down. As master is somewhere elase or another slave I somewhere.. so my allocation strategy reduced the probability of ……..

There are many strategy…… use imagination… for shards to nodes. Availability zone, duplicate Zone, etc

He will share the orchestration class video…

xcvbg

so as soon as i add new physical machines the logical enitty will be moved automatically by the system or i wil have to do some configuration or Zookeeper will work here

**Read flow:**