20/12/2023

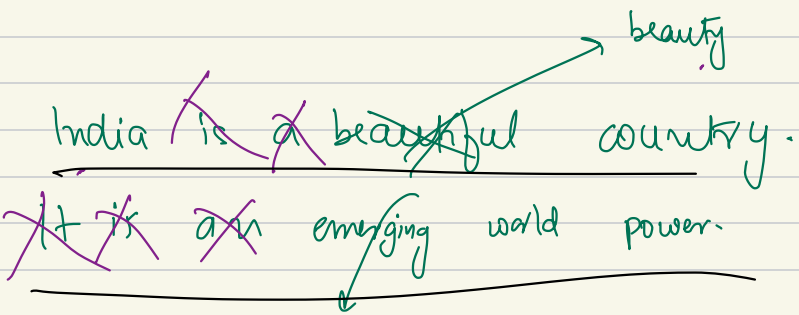
Inverted Index – Apache Lucene

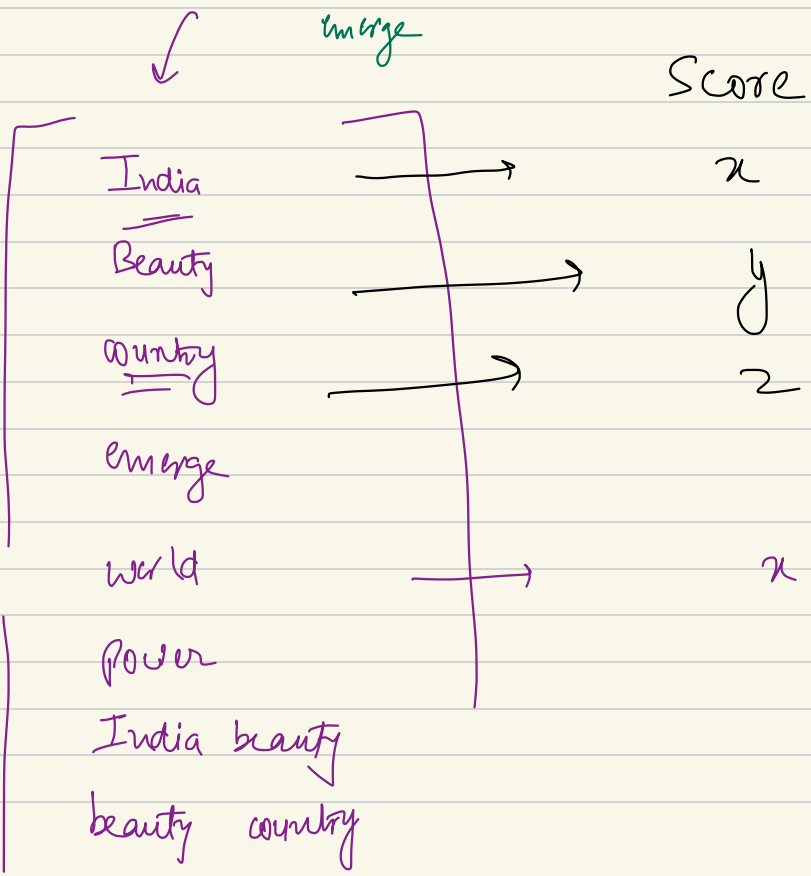
+scalability

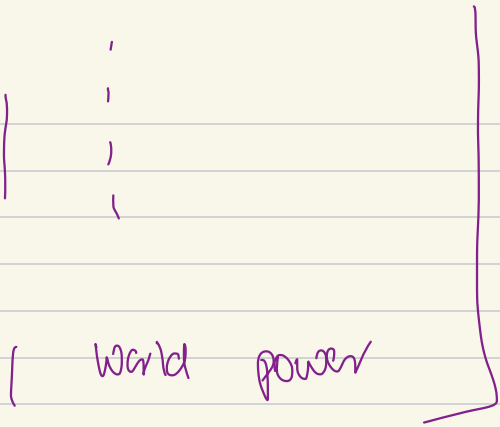
Elastic Search

doc: A

India is a beautiful country.
It is an emerging world power.



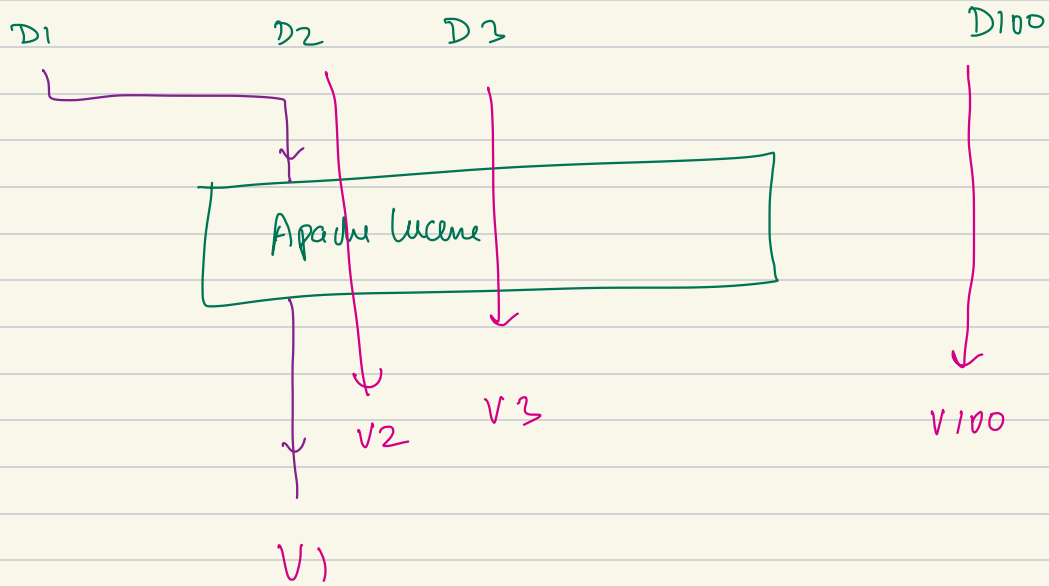




raw document → Apache Lucene

raw document + vector representation of that document

100 documents that I want to index



D1, V1

D2, V2

D3, V3

D4, D4

D100, V100

Inverted Index

Superset of key ← union of all vectors

A POST: India is a beautiful country

B POST: Canada India relationship is at a new low

C POST: Canada Engineering colleges might a show.

D POST: India lost the CWC

POST: I love my job. I love my life

ex. Unigrams.

Inverted Index

India – A (0.4), B (0.2); D (0.4)

beauty – A (0.3)

country – A (0.3)

Canada – B (0.4); C (0.3)

relation – B ( )

new – B

low – B

engineer – C

college – C

sham – C

loose –

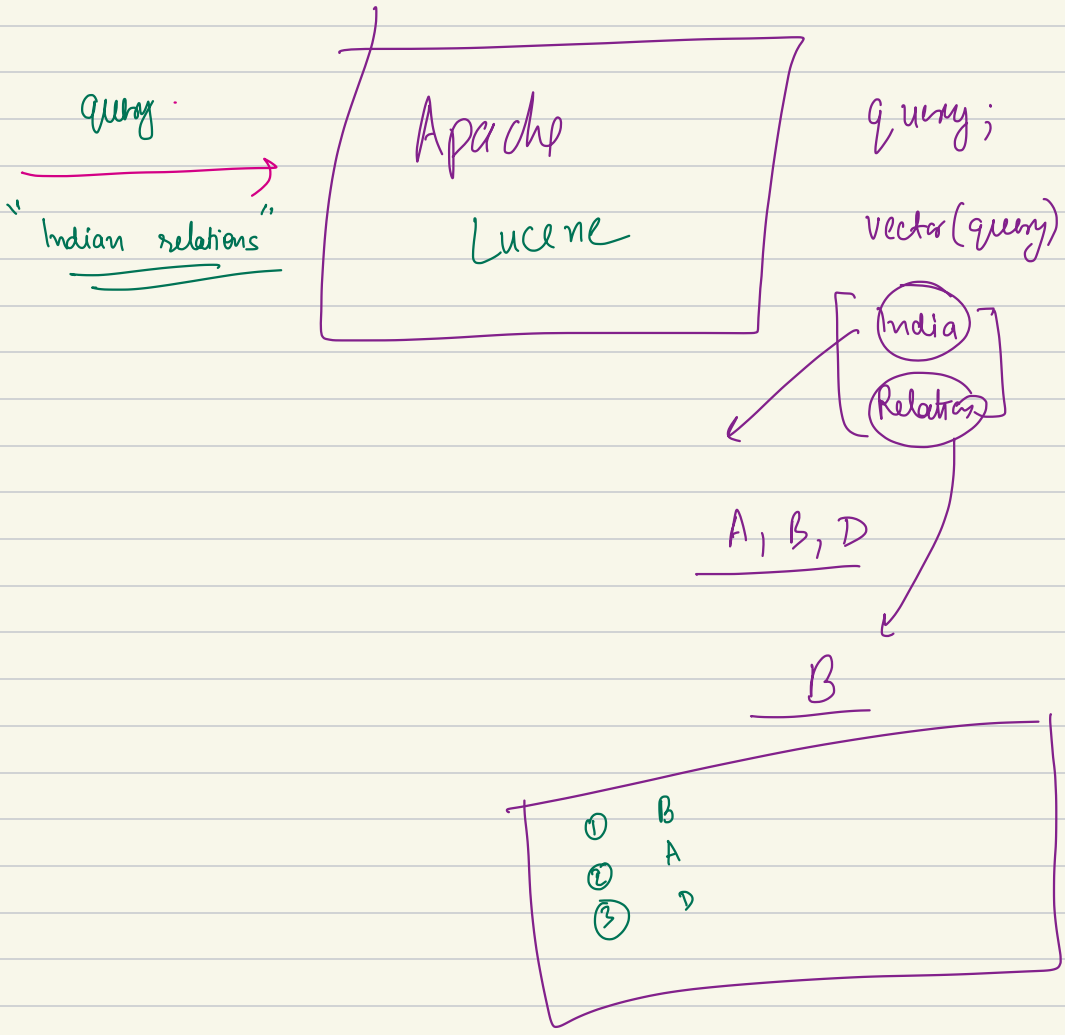
CWC –

love –

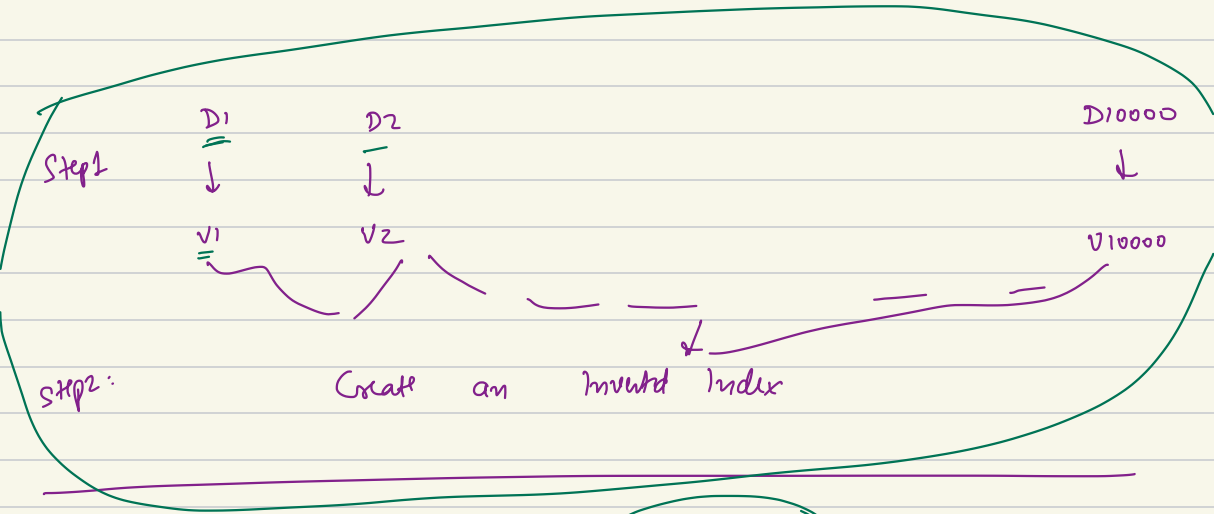
job –

my –

life –


query :
"Indian relations"
Apache
Lucene
query ;
vector(query)
India
Relations
A, B, D
B
1
2
3
B
A
D

example: 10k documents at Google



query → vector(query)

Search on Inverted Index

[list of documents]

[Sorted list of documents]

Apache Lucene is great. It does great NLP, fast query processing ☺

But Lucene is NOT scalable.

Elastic Search

– Lucene made scalable ☺

(3) Design Tradeoffs

(1) Availability >>> Consistency

☺ ☺

(2) ✓ Low latency for getting Search Results

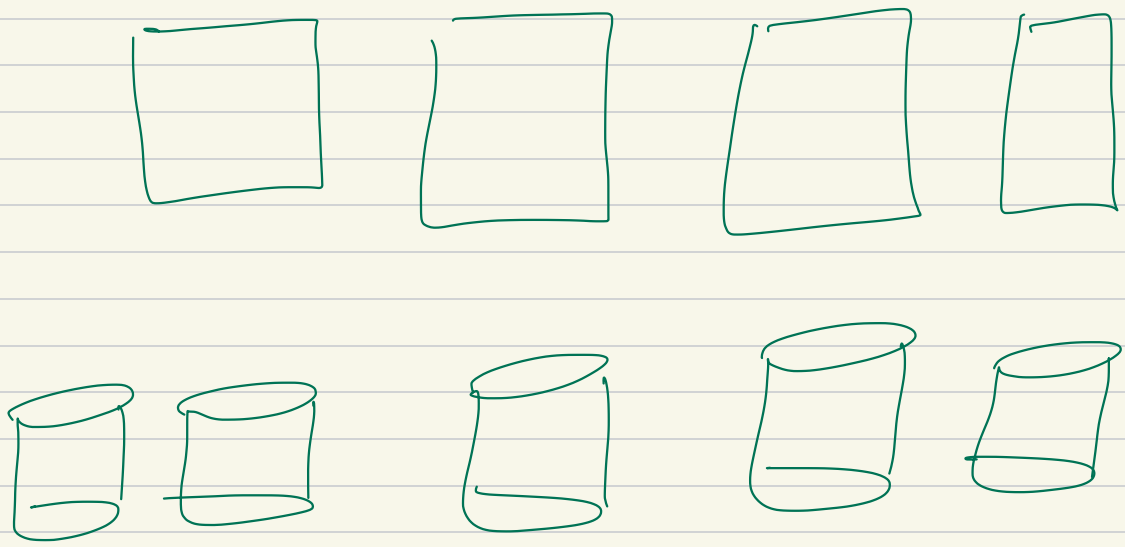
☆ Low Read latency

✓ We are fine with medium-high latency for writes...

(4) Design deep dive

(1) . compute resources.

(2) . storage.



① Sharding is definitely needed

② Replication as well.

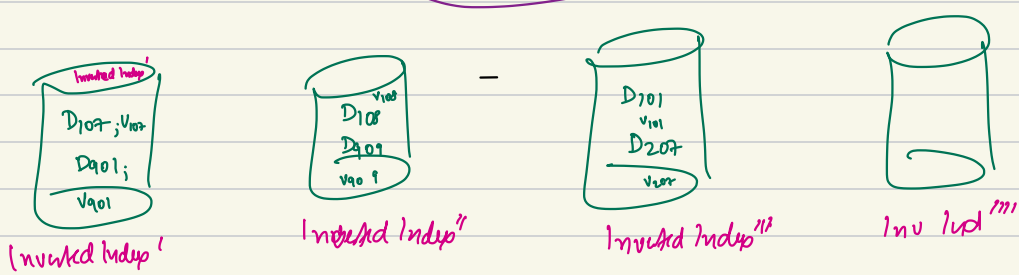
data you're storing

① ✓ documents $D_i \rightarrow 10B$

② ✓ vector $V_i \rightarrow 10B$

③ Inverted Index $\rightarrow 1$

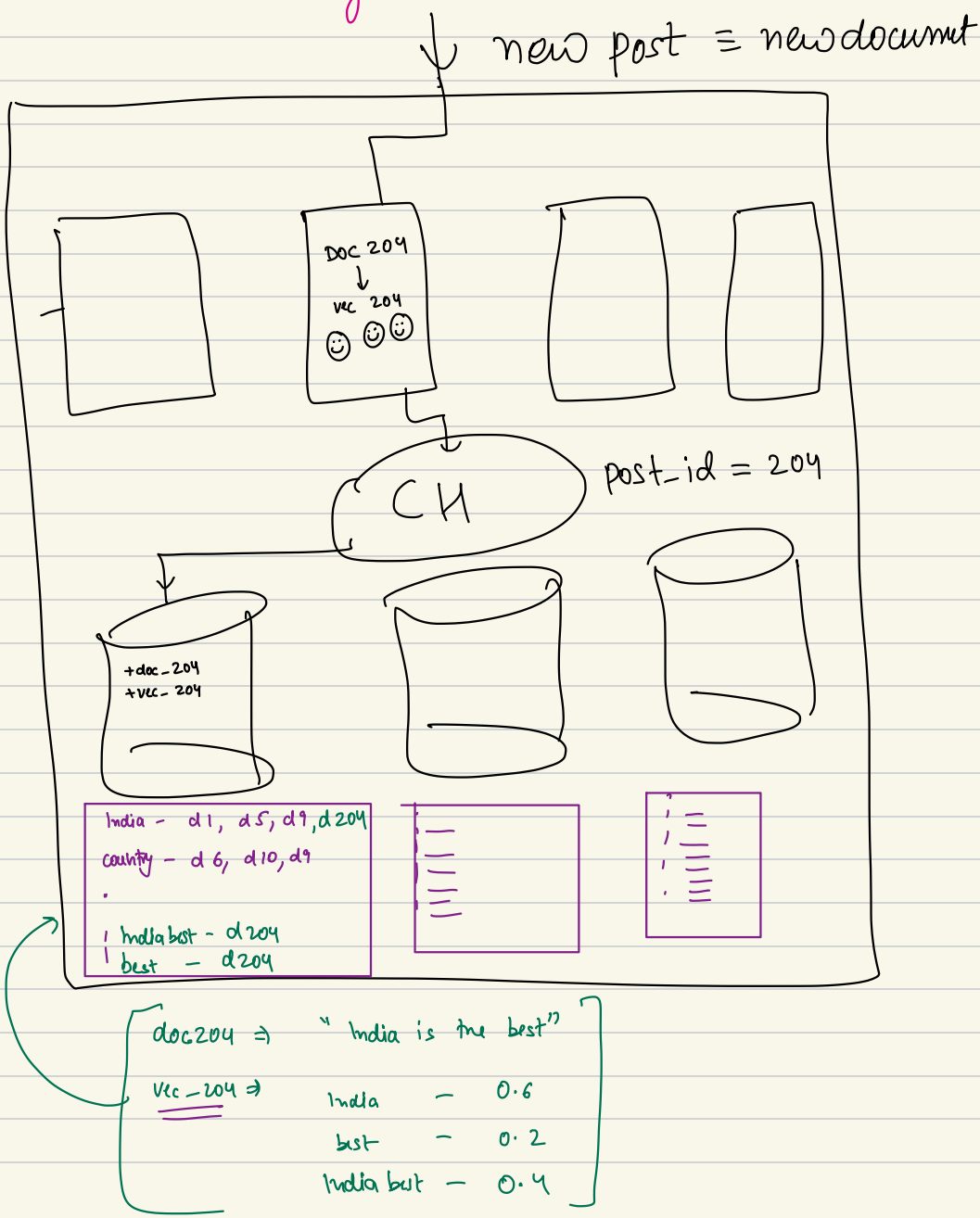
CN $\longrightarrow$ doc_id



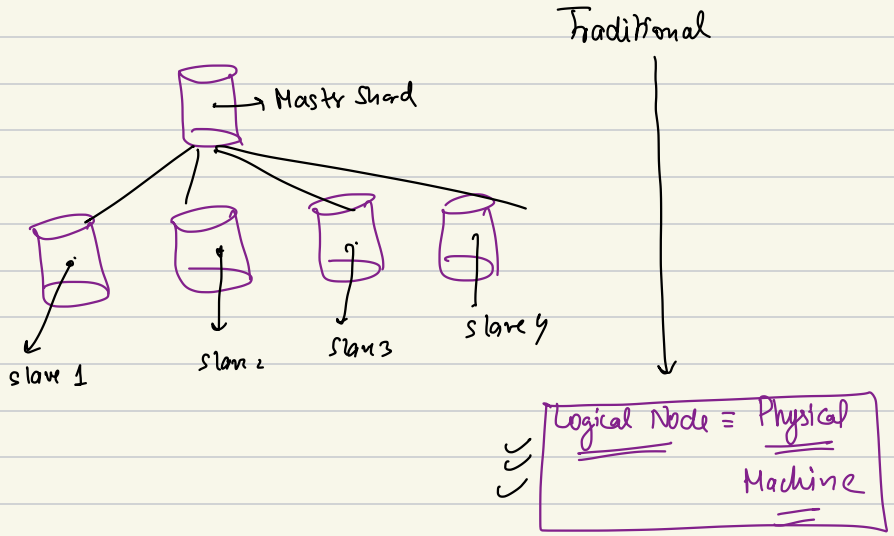
This means that we now have a distributed inverted indy

WRITE FLOW (a document ingestion)

ingest a new document

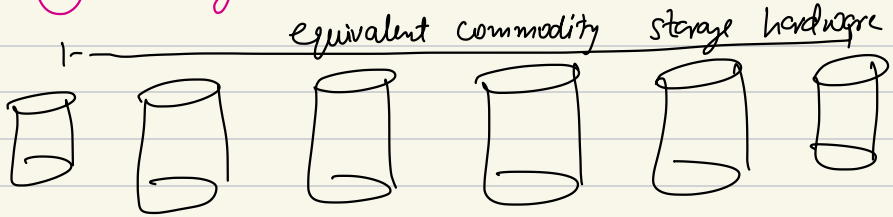


Logical Shards and Physical Nodes / Machines



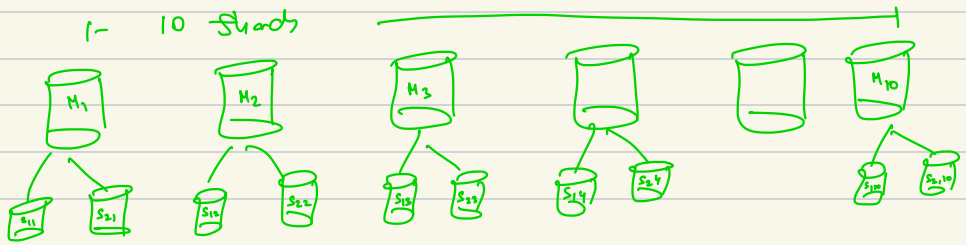
2 concepts a play.

1. Physical Machine



2. Logical Idea of shard → Master slave etc

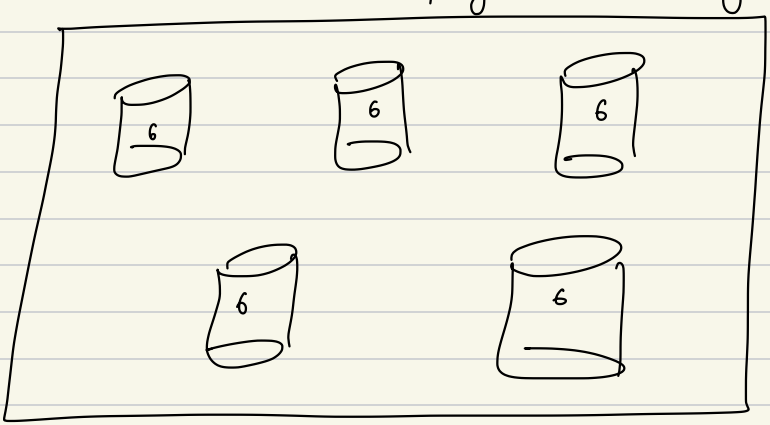
10 shards, each with 2 slaves

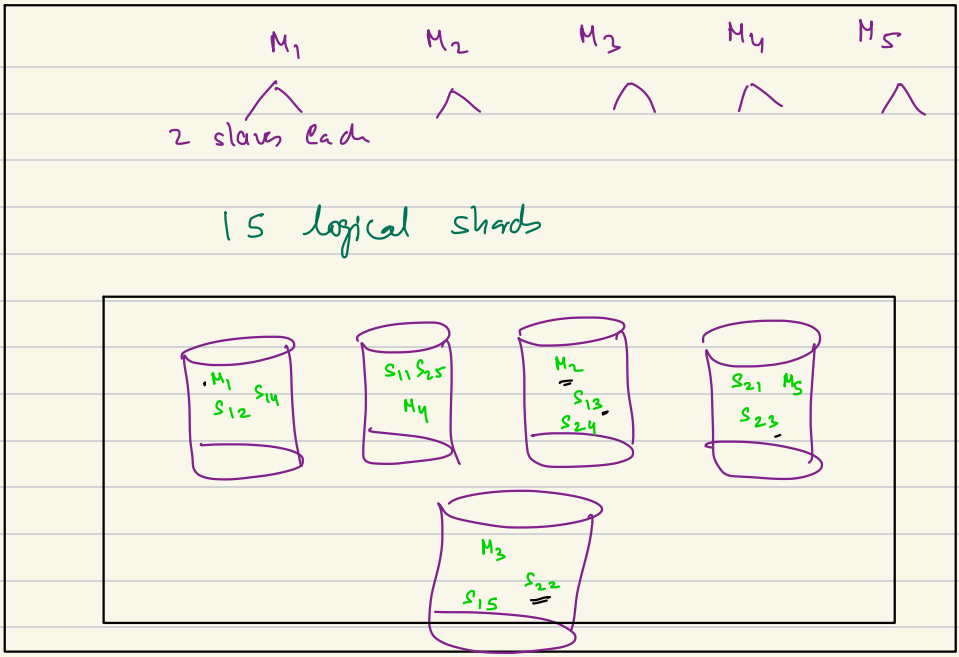


30 logical machines

M_1	M_2	M_3	--	M_{10}
S_{11}	S_{21}			
S_{12}	S_{22}			
S_{13}	S_{23}			
!				
$S_{1,10}$	$S_{2,10}$			

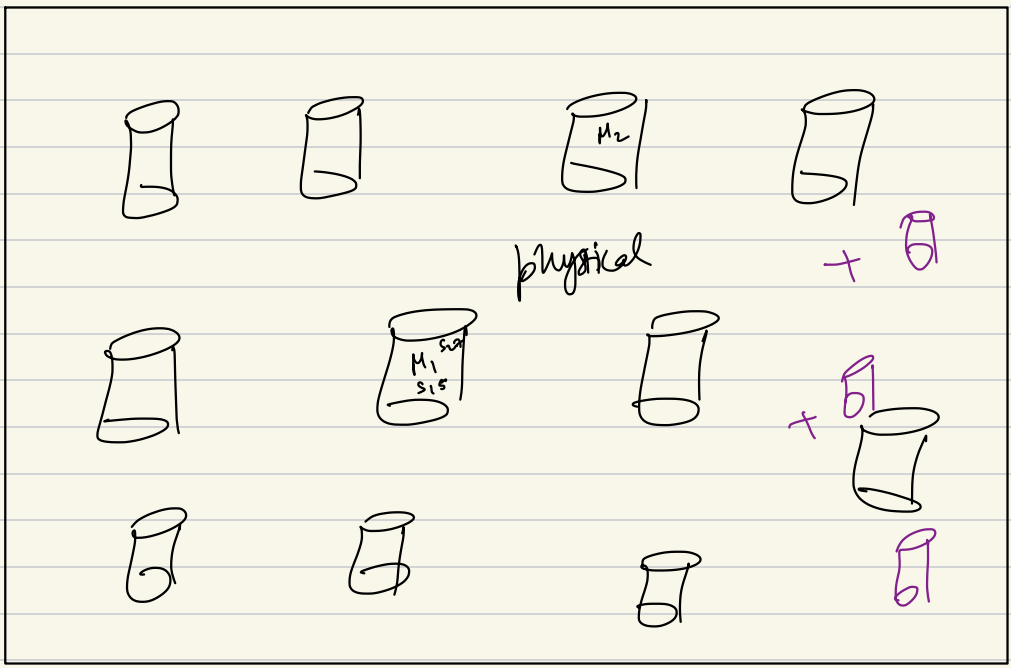
physical cluster of 5 machines




M1
M2
M3
M4
M5
2 slaves each
15 logical shards
M1
S12
S14
S11 S25
M4
M2
S13
S24
S21 M5
S23
M3
S15
S22

Elastic Search exploit the above idea

100 machines in ES cluster → + 10 physical machines



ES

- ✓ 100 physical machines
- ✓ 70 Master machines / shards
  - 3 slaves

280 logical shards

logical → 280

280 logical
10 physical

280 logical
110 physical

We have used diffent logical shard and physical machine concept.

Because every logical shard has some documents allocated and the respective inverted index.

We don't want to create/distroy our logical shards

as creating | destroying them would mean

re allocation of documents into new mapping

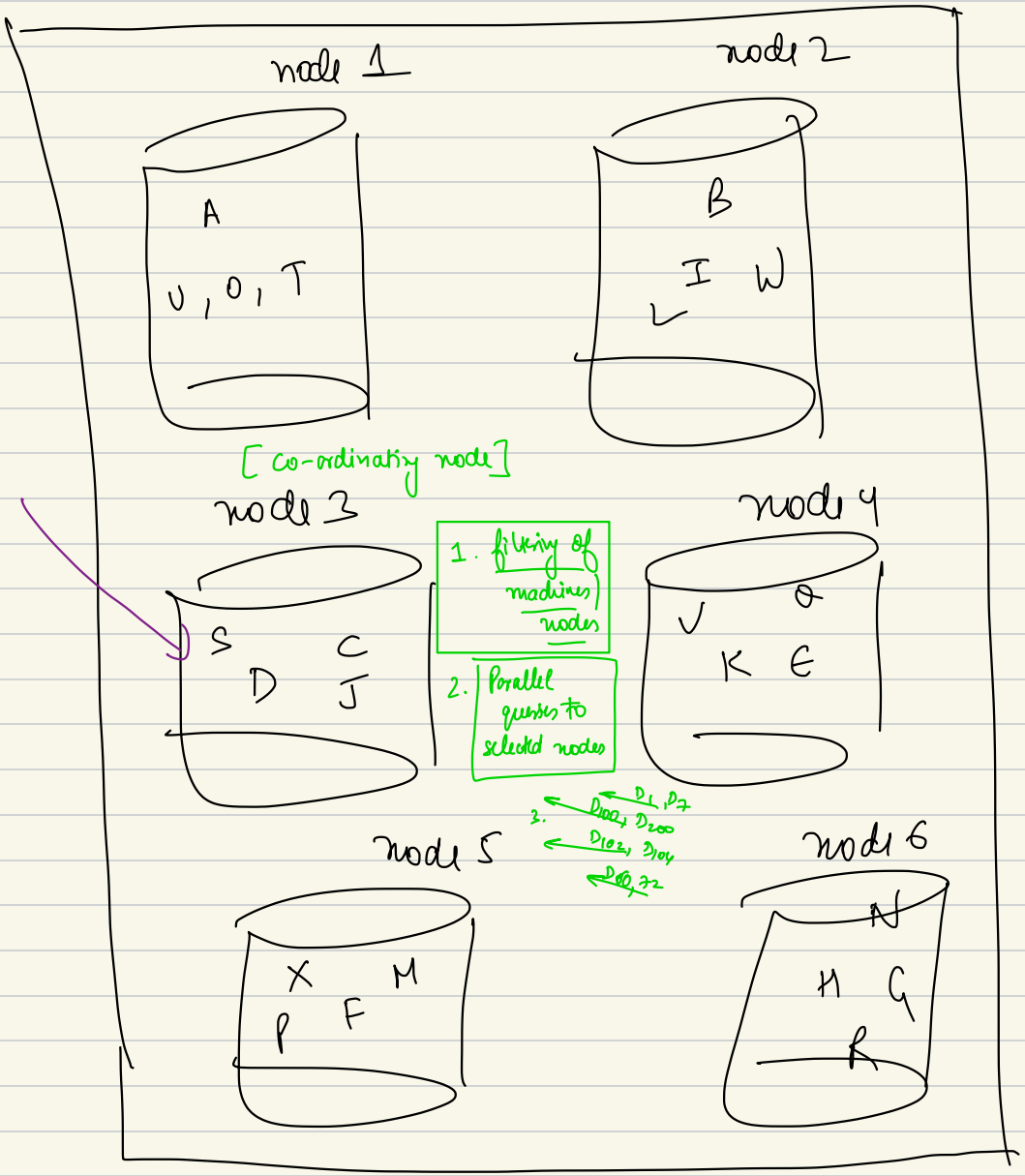
re creation of inverted indices

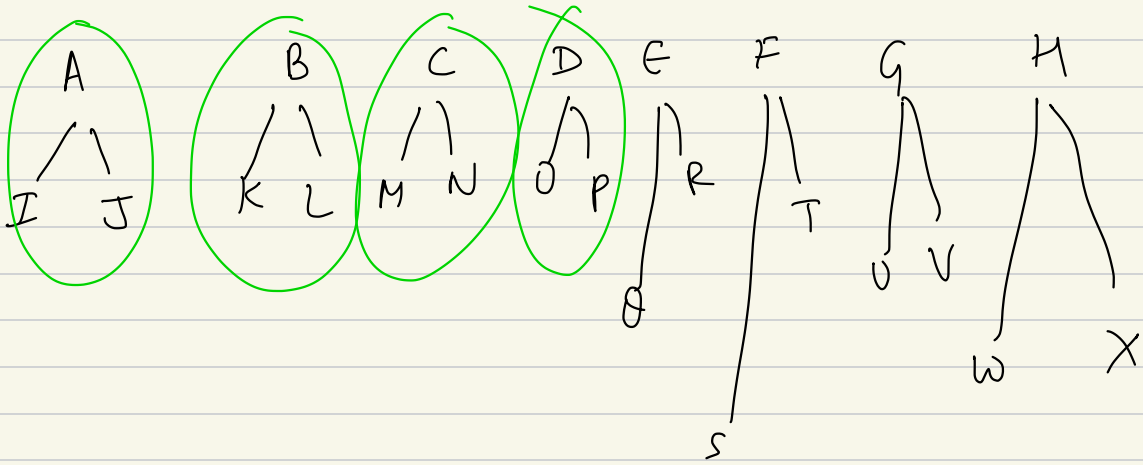
This is time consuming....

So we want to avoid this.

Rather, we can add new physical machines, keeping the logical shards as it is, so that we can pick up entire logical shards and put them to new machines without having the need to recreate indices.

Read Flow


node 1
A
U, O, T
node 2
B
I W
L
[co-ordinating node]
node 3
S
C
D
J
1. fetching of machines nodes
2. Parallel queries to selected nodes
node 4
V
Q
K E
3.
D1, D7
D100, D200
D102, D104
D80, 72
node 5
X M
P F
node 6
N
H G
R



8 masters

25 bins / master

24 logical shards

query → computation → raw query + vector(query)

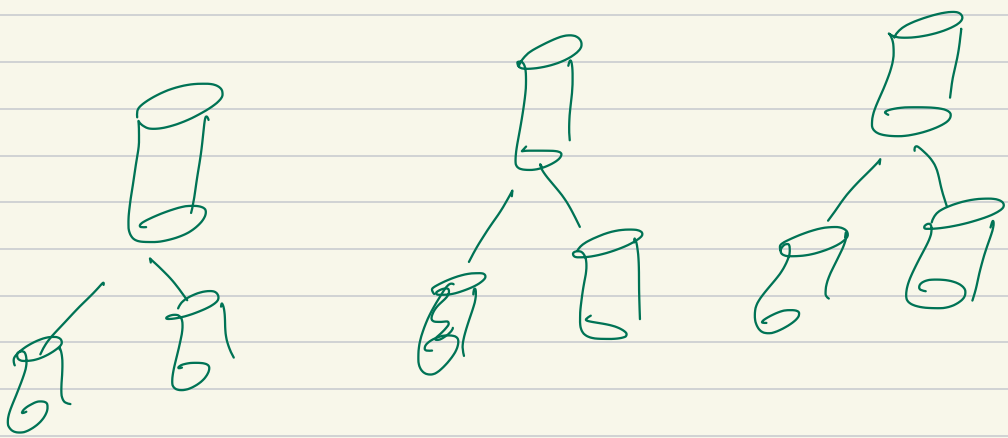
Search engine for Forbes.com

forbes.com/sports/cricket/indvspak-cwc-match.

forbes.com/sports/cricket/ind vs aus – cwc 2023-final

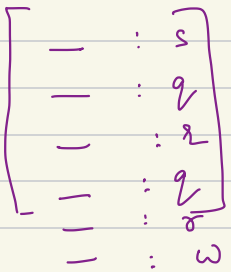
forbes.com/sports/tennis/wimbledon – records

forbes.com/cars/tesla-latest-model



Co-ordinating Node

① query processing → vector of query



→ land at a co-ordinating node

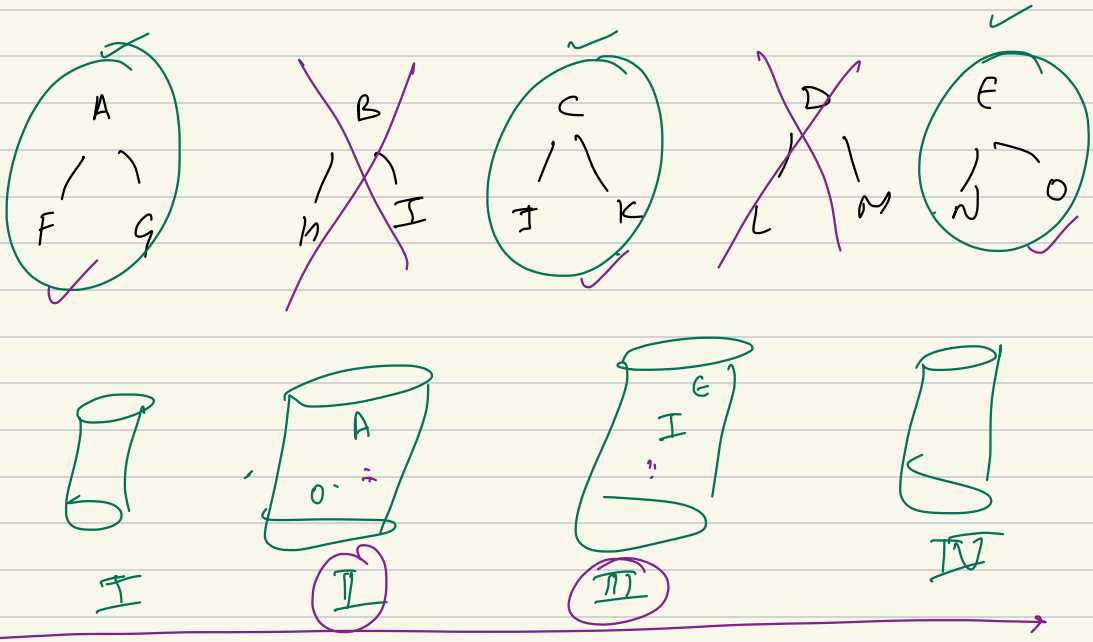
② [optional]

Filter out some master shards as important shards based on what your query contains.

[Step ② is only possible if the sharding of documents happened in a logical clustered way / topic based partitioning]

③ Out of the non-rejected shards, the coordinating node will decide a list of physical machines which have

Representation from each of selected M-S logical shards.

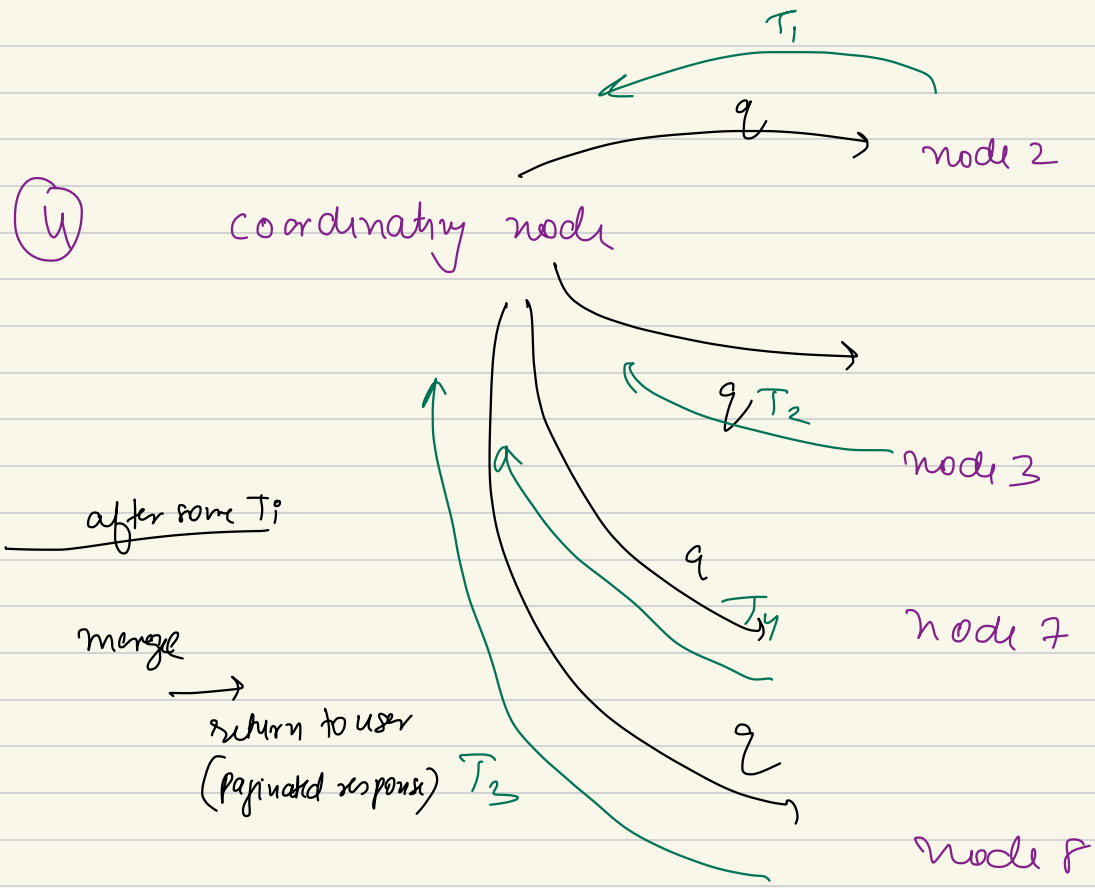


You will select such a list so that number of physical machines you piggyback on is minimum

+

load of machines is low

In the shard selection, the coordinatory node will identify some physical nodes (least number of machines) and parallely send queries to selected machines.



Parallel Processing of Query

Timeout

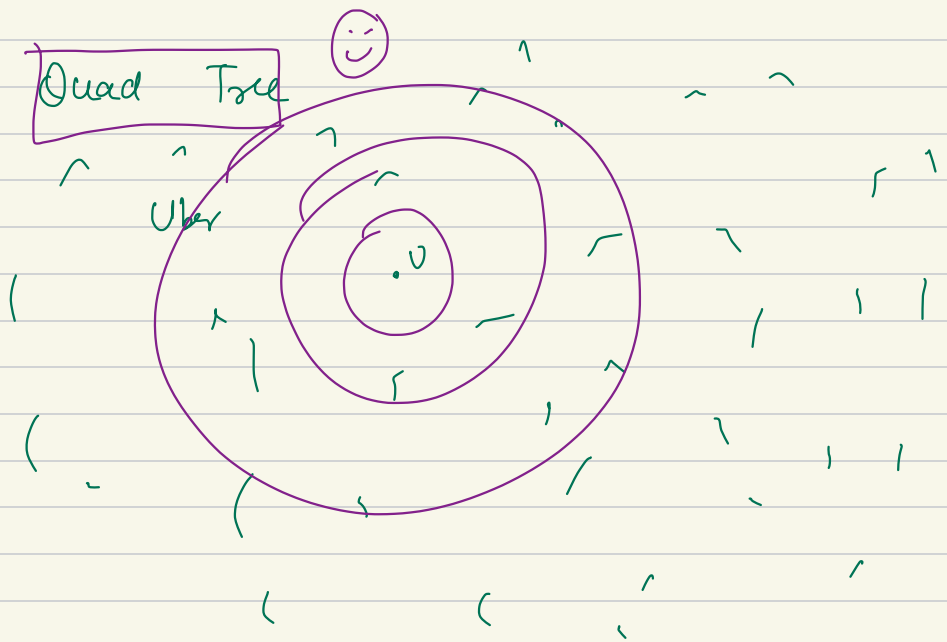
(combine + rank)

↓

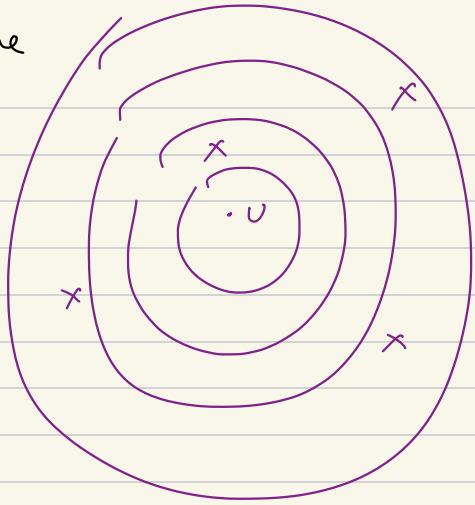
User

(I) S3 → File/Blob Store ≡ HDFS

(II) Quad Tree



barber shops near me



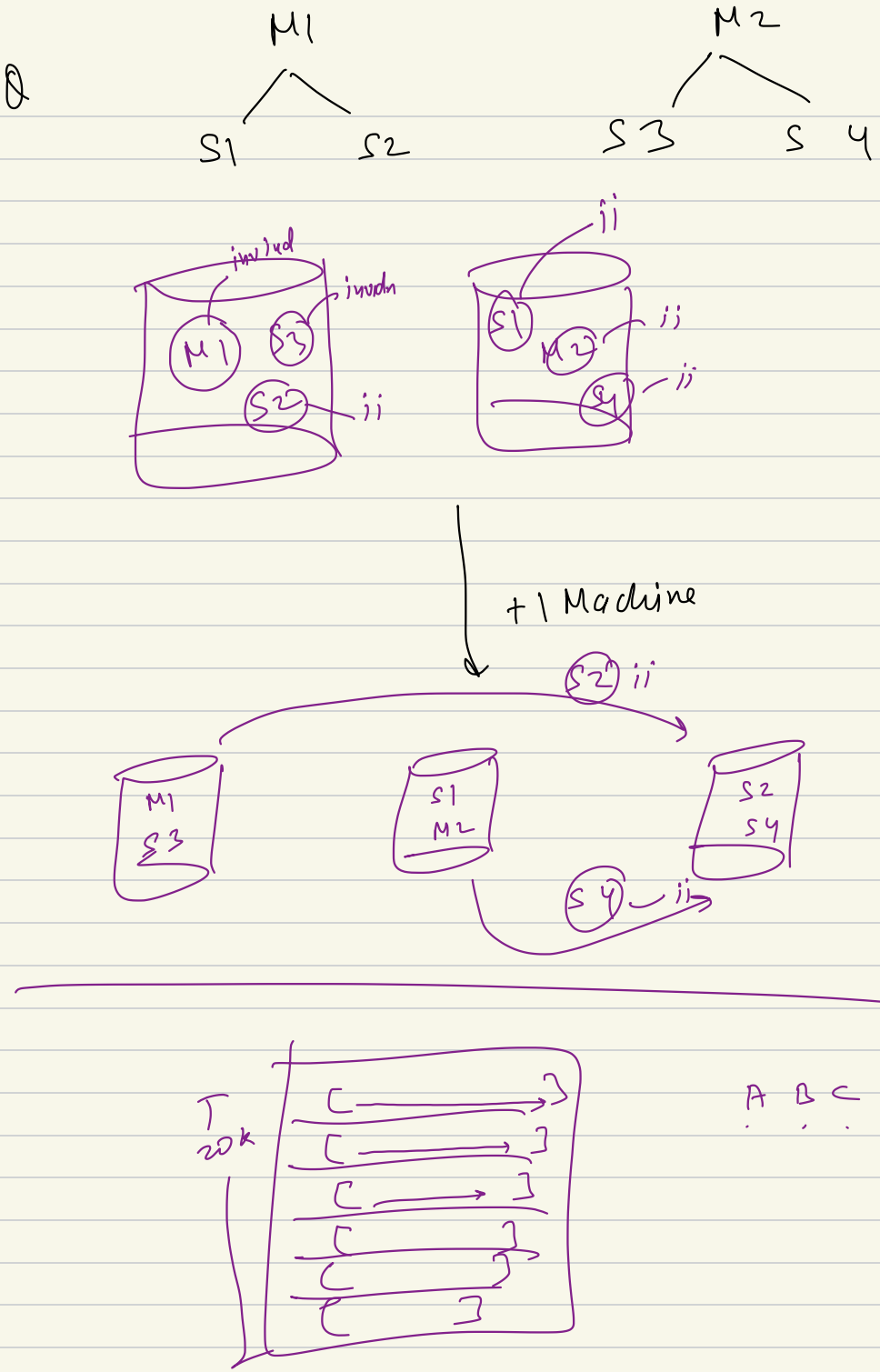
Questions

(1)

	Score
India	0.4
Country	
india best	
india pm	
Modi	0.5
Narendra Modi	
Australia	0.1

NLP Techniques

1. Frequency of word
2. Tf-Idf
3. 


M1
S1
S2
M2
S3
S 4
invalid
invalid
ii
ii
ii
ii
+1 Machine
S2
ii
M1
S3
S1
M2
S2
S4
S4
ii
T
20k
A B C