

Week 1

Data integration is one leading cause leading to the bigness

Video 1 of data.

Summary of

Big data modeling and Management

→ Recall why big data modelling and management is essential in preparing to gain insights from your data.

→ Summarize different kinds of data models.

→ Describe streaming data and the different challenges it presents.

→ Explain the diff. betⁿ a DBMS and a BDMS.

↓
(Big Data Management System)

• Data modeling tells you:-

→ How your data is structured

→ What operations can be done on the data

→ What constraints apply to the data

• Database management Systems :-

→ typically handle many low-level details of data storage manipulation, retrieval, transactional updates, failure and security.

→ Relieve a user to focus on high level operations like querying and analysis.

MongoDB :- semi structured data management system.

Aerospike :- key value store

spark :- most popular big data engines

04

2016

Friday

March

064-302 • WK 10

06	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
07	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31							
08	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31														
09	22	23	24	25	26	27	28	29	30	31																					
10	29																														

Relational data model has been implemented in traditional database systems.

Different data models

9:00 Relational Data

→ where data look like tables

10:00 ↘ They are being refreshly implemented in modern data systems over Hadoop and Spark.

11:00 Semi-structured Data

→ Document data, XML and JSON ⇒ This can be embedded another

12:00 data element and hence can often be modeled as tree.

1:00 Graph Data

→ 3:00 social Networks, email networks.

nodes = entities
edges = Relations
btⁿ such entities

* The operations performed on graph data includes transerving the network

4:00 Text Data

→ Articles, reports

The text data is much more unstructured bcoz an entire data item like new article can be just a text string

primary form :- text

6:00 • Streaming Data :- An infinite flow of data coming from a data source

→ sensor data from instruments

→ stock price data

It will need different kind of storage system. memory in chunks = windows

NOTES

April 2016							05	May 2016						
W	T	F	S	S			Wk	M	T	W	T	F	S	S
1	2	3					18/23	30	31					
4	5	6	7	8	9	10	19	2	3	4	5	6	7	8
11	12	13	14	15	16	17	20	9	10	11	12	13	14	15
18	19	20	21	22	23	24	21	16	17	18	19	20	21	22
25	26	27	28	29	30		22	23	24	25	26	27	28	29

2016
Tuesday
March

08

068-298 • WK 11

→ Data rates vary - can be too fast and too large to store.

→ often processed in memory

→ Many need to be processed immediately

① Inform whenever 3 tech stocks go by by 3% within a 30 second span.

② Used for event detection and prediction.

• BDMS :- Designed for parallel and distributed processing

→ Data-partitioned parallelism

- May not always guarantee consistency for every update

→ More likely to guarantee eventual consistency.

→ often built-on Hadoop

→ Offer map reduce style computation

→ Utilize replication natively offered by HDFS

Video 2 :- Week 1

Why is Big data Processing Different?

• Summarize the requirements of programming models for big data and why you should care about them.

• Explain how the challenges of big data related to its variety, volume and velocity affects its processing

NOTES

Requirements of Big data systems :-

9.00

10.00

11.00

12.00

1.00

2.00

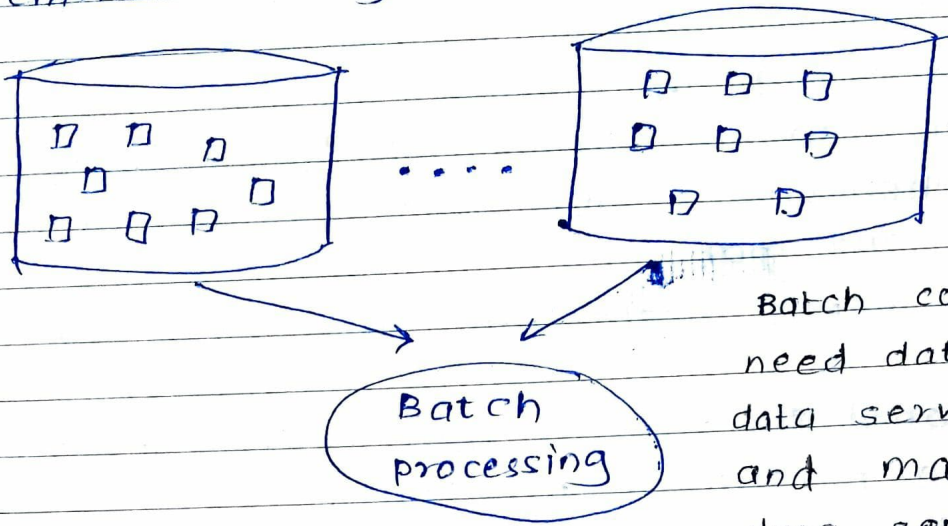
3.00

4.00

5.00

6.00

7.00

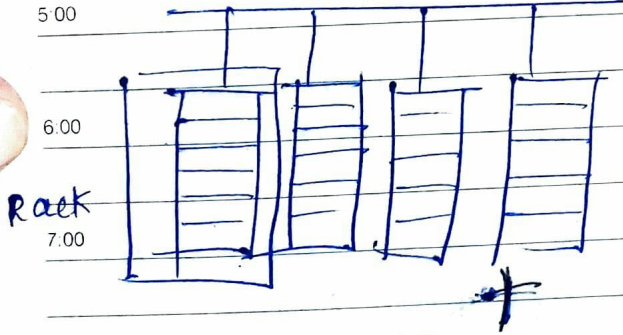


Batch computations that need data from multiple data servers need to access and maintain use of data separat which might end up being quite slow and costly.

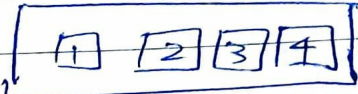
↑
↓
scalability complexity

Network

⇒ Data-parallel scalability

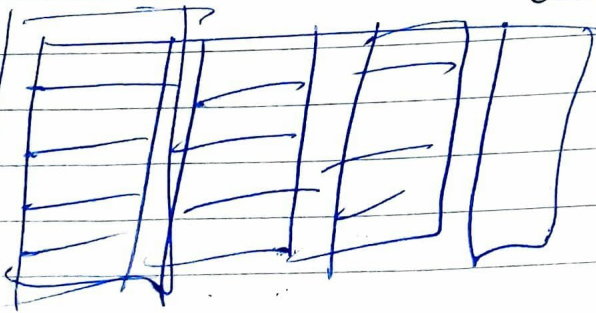


data



compute

NOTES



2016

Friday

July

183-183 • WK 27

Programming model

= abstractions

06

June 2016

07

July

wk	M	T	W	T	F	S	S
23			1	2	3	4	5
24	6	7	8	9	10	11	12
25	13	14	15	16	17	18	19
26	20	21	22	23	24	25	26
27	28	29	30				

wk	M	T	W	T	F
27					1
28	4	5	6	7	8
29	11	12	13	14	15
30	18	19	20	21	22
31	25	26	27	28	29

Runtime
Libraries

+

Programming
Languages

* Requirements of Big Data Systems.

1. Support Big Data operations

→ Split volⁿ of data

→ Access data fast

→ Distribute computations to nodes.

2. Handle Fault Tolerance

→ Replicate data partitions

→ Recover files when needed.

3. Enable Adding More Racks.

4. Optimized and extensible for many data type.
Document, table, key-value, graph,

5. Enable both streaming and batch processing

→ Low latency processing of streaming data.

→ Accurate processing of all avail. data.

volⁿ

→ scalable batch processing

Velocity

→ stream processing

variety

→ Extensible data storage, access
and integration.

Big data Integration and Processing

Week 1 part 2

What is Data Retrieval?

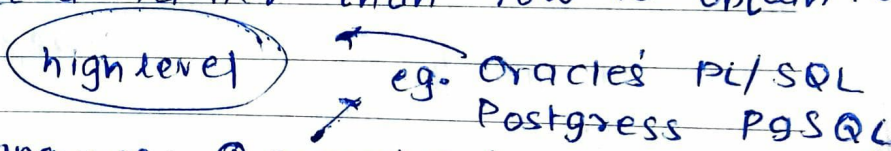
- Define a query language
- Write simple queries using SQL
- Express simple queries using MongoDB
- Write simple queries using Aerospike
- Explain how large-scale data can be processed by data-partitioned parallelism.

Data retrieval:- The way in which the desired data is specified and retrieved from a data store.

- Our focus:- How to specify a data request
 - For static and streaming data
 - The internal mechanism of data retrieval
 - For large and streaming data.

Query language:- A language to specify the data items you need.

- Query language is declarative
- Specify what you need rather than how to obtain it
- SQL

Database programming language:- 
① procedural programming language
② Embeds query operator.

SQL:- The standard for structured data
→ Oracle's SQL to Spark SQL

Select - Project Queries in the Large

→ Large Tables can be partitioned

- many partitioning schemes
- Range partitioning on primary key

A Tuple is a sequence of immutable python objects. Tuples are sequences, just like lists. The difference between tuples and lists are, the tuples cannot be changed unlike lists and tuples use parentheses, whereas lists use square brackets. Creating a tuple is as simple as putting different comma-separated values.

* Local and Global Indexing :-

→ What if a machine does not have any data for the query attributes?

- Index structures :-
 - ① Given value, return records
 - ② Several solⁿ

→ Use local index on each machine.

→ Use a machine index for each value.

→ Use a combined index in a global index server.

NOTES

data retrieval :- The way in which the described data is

specified and retrieved from a data store

Our focus :- How to specify a data request

→ For static and streaming data

→ The internal mechanism of data retrieval

→ For large and streaming data

→ The internal mechanism of data retrieval

→ For large and streaming data

What is a Query Language?

→ A language to specify the data items you need.

→ A query language is declarative.

→ Specify what you need rather than how to obtain

• SQL (structure Query language)

→ Database programming language

• Procedural programming language

• Embeds query operations.

SQL :- Oracle's SQL to Spark SQL.

SELECT name  o/p attributes)

FROM  Tables) to use

WHERE codⁿ

09

2016
Wednesday
March

069-297 • WK 11

02 February 2016						
Wk	M	T	W	T	F	S
06	1	2	3	4	5	6
07	8	9	10	11	12	13
08	15	16	17	18	19	20
09	22	23	24	25	26	27
10	29					

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29

Local and Global Indexing :-

→ What if a machine does not have any data for the query attributes?

→ Index structures

- Given value, return records

11:00 • several colⁿ

→ Use local index on each machine

12:00 → Use a machine index for each value.

→ Use a combined index in a global index

1:00 Servers

2:00 Querying Two Relations :-

→ often we need to combine two relations for queries.

- ~~eq~~

4:00

Join in a Distributed Setting :-

5:00 • Semijoin :- A semijoin from R to S on attribute is used to reduce the data transmission cost.

6:00

- Computing steps :-

7:00 • Project R on attribute A and call it (R_{CA}) the Drinkers column

- ship this projection (a semijoin projection) from the site of R to site of S.

NOTES

Reduce S to S' by eliminating tuples whose attribute A are not matching any value in R_{CA}