HIVE

Sundharakumar KB

Department of Computer Science and Engineering
School of Engineering

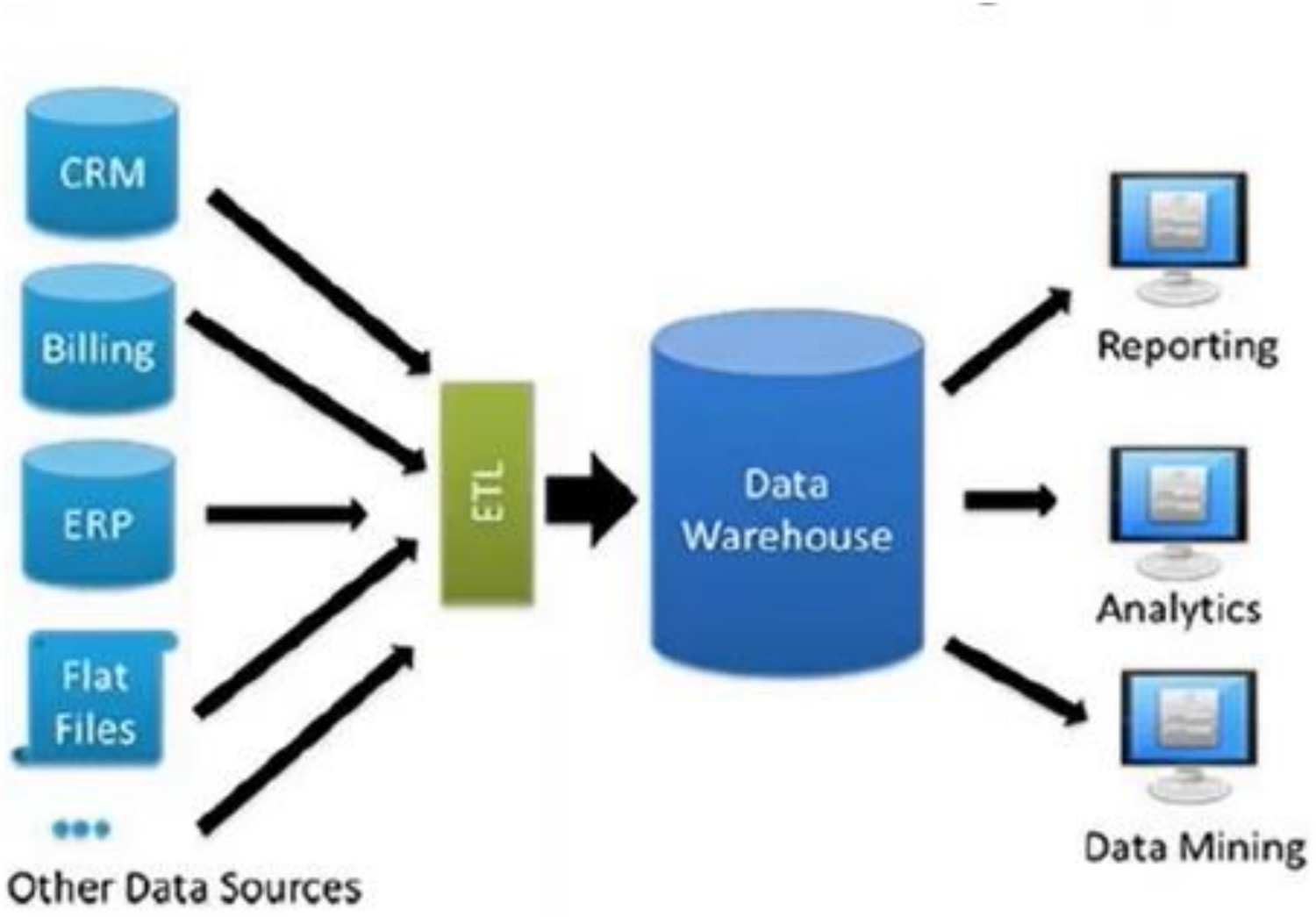
Shiv Nadar University Chennai

SHIV NADAR
UNIVERSITY
CHENNAI

HIVE

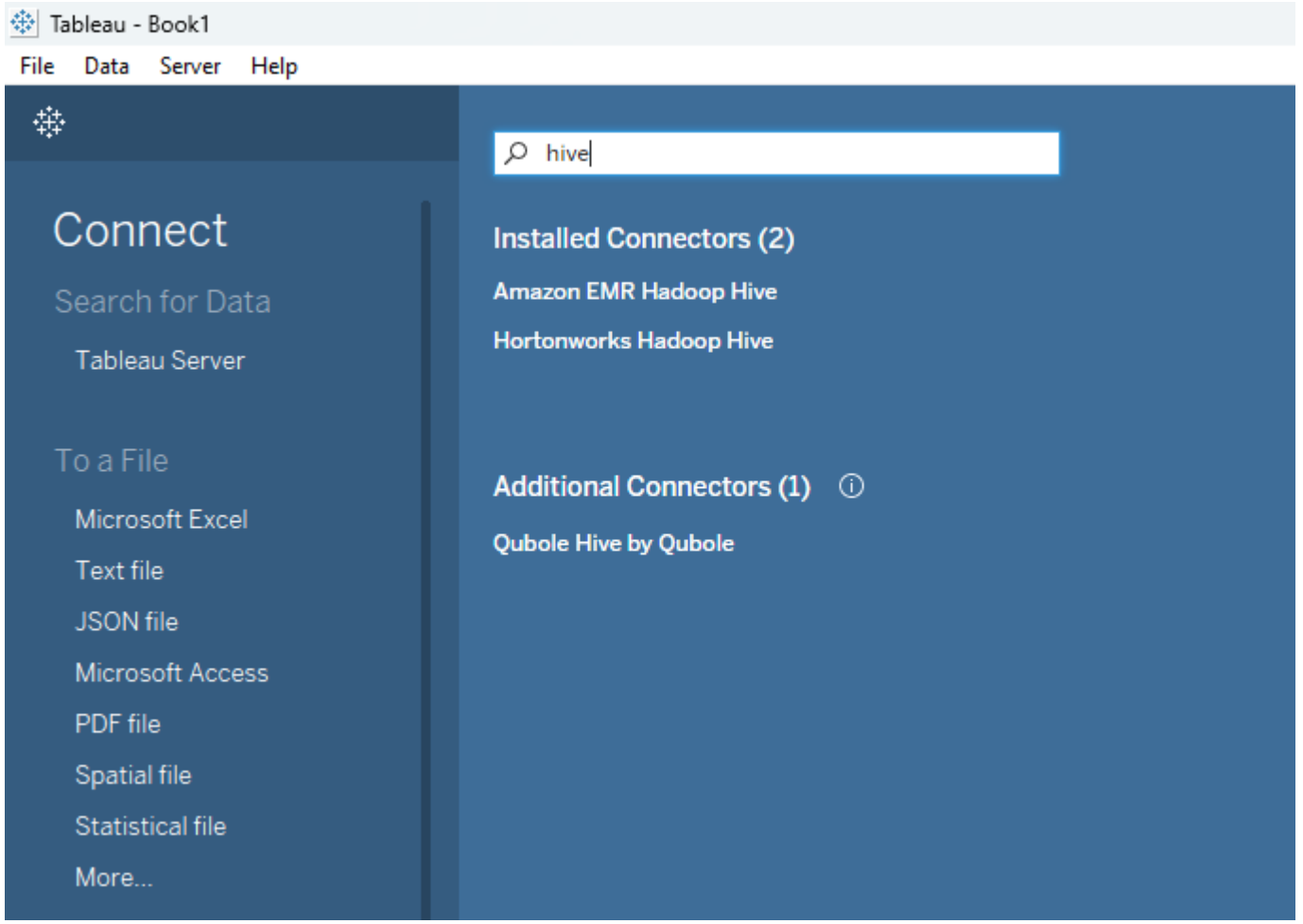
- Hive is a data warehouse infrastructure based on Hadoop framework
- Suitable for data summarization, analysis and querying.
- Uses SQL like syntax called as HQL – Hive Query Language.
- Hive uses mapreduce and HDFS for processing and storage/retrieval.

HIVE


CRM
Billing
ERP
Flat Files
Other Data Sources
ETL
Data Warehouse
Reporting
Analytics
Data Mining

Advantages of HIVE

- Can be used as ETL tool.
- Can handle large datasets.
- SQL(filters, joins, groupby) on top of map and reduce.
- Provides integration support to BI tools.



Where not to use Hive

- Don’t use hive if:

- The data size doesn’t cross GBs.
- Creating schema from the data is not possible
- RDBMS can solve the problem, don’t invest time in Hive.
- For low latency applications and for low response time.

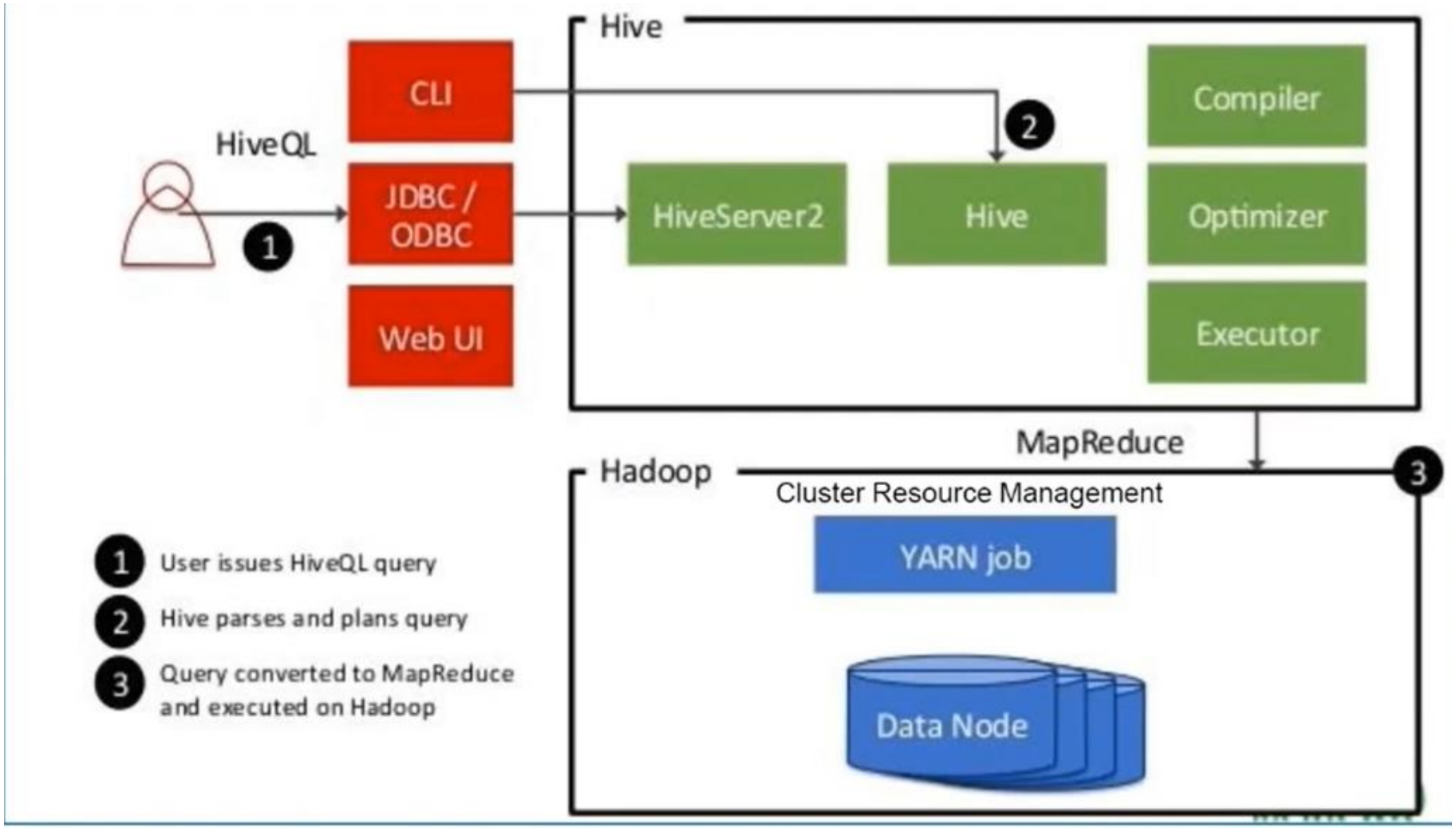
Similarity with SQL & Differences

- **Similarity:** HiveQL is very similar in syntax to SQL. It is based on SQL-92 framework and acts as SQL for DFS.

- **Difference:** the query is executed on Hadoop infra (cluster) than a traditional RDBMS.
- This allows hive to scale and handle huge datasets.
- Internal execution of hive queries is a series of automatically created map-reduce tasks.

- All hive queries are converted to mapreduce.. Why not straightaway write MR codes?
- To do this, internals of Hadoop framework is required. However, people with SQL knowledge can simply write hive queries to get the results.

Hive Architecture



Hive Architecture

- By default, Hive uses embedded metastore; this means the metadata is stored in the built-in Derby database.

- You can configure your own database as a metadata storage for Hive – eg: MySQL.

- Data in hive is organised into three categories:
  - Tables
  - Partitions
  - Buckets

Sample Queries

- Show databases;
- Show tables;
- Create database sample;
- Use sample;
- Create table emp (emp int, name string, city string)

ROW FORMAT delimited fields terminated by “,” LINES TERMINATED BY “\n” STORED AS TEXTFILE;

Sample Queries

- Insert into emp (sno, name, city) values (1, ‘sample’, ‘Chennai’);

- Load data local inpath ‘/home/Sundharakumar/Desktop/sampletxns.txt’) into table emp;

- Desc emp;
- Select * from emp; -> no mapreduce
- Select count(*) from emp; -> mapreduce

- Load data inpath ‘/samplehive/sampletxns.txt’) into table emp;

Sample Queries

```
hive>  select * from emp where city in ('chennai');
OK
1        g        chennai
```

```
hive> select count(*) from emp;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive
 1.X releases.
Query ID = test_20210712194342_be07401d-73aa-4baa-b43d-a00c31e431a3
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1626090715853_0003, Tracking URL = http://localhost:8088/proxy/application_1626090715853_0003/
Kill Command = /home/test/hadoop-2.9.1/bin/hadoop job  -kill job_1626090715853_0003
```

Partitions

- Partitions:
  - Each table can be broken into partitions
  - Partitions determine distribution of data within subdirectories

- Static and Dynamic Partitioning

CREATE_TABLE Sales (sale_id INT, amount FLOAT)

PARTITIONED BY (country STRING)

Insert into sales partition (country=‘India’) select sale_id, amount from sales_nopart where country=‘India’

CREATE_TABLE Sales (sale_id INT, amount FLOAT)

PARTITIONED BY (country STRING)

- Set hive.exec.dynamic.partition.mode=nonstrict;

INSERT into sales partition (country) select sale_id, amount from sales_nopart where country='India';

Buckets

- Buckets
- Can speed up queries that involve sampling the data
  - Sampling works without bucketing, but Hive has to scan the entire dataset
- Use CLUSTERED BY when creating table
  - For sorted buckets, add SORTED BY
- Data can be divided into buckets
- Based on a hash function of the column
- **H(column) mod NumBuckets = bucket number**
- Each bucket is stored as a file in partition directory

- Set.hive.enforce.bucketing = true;