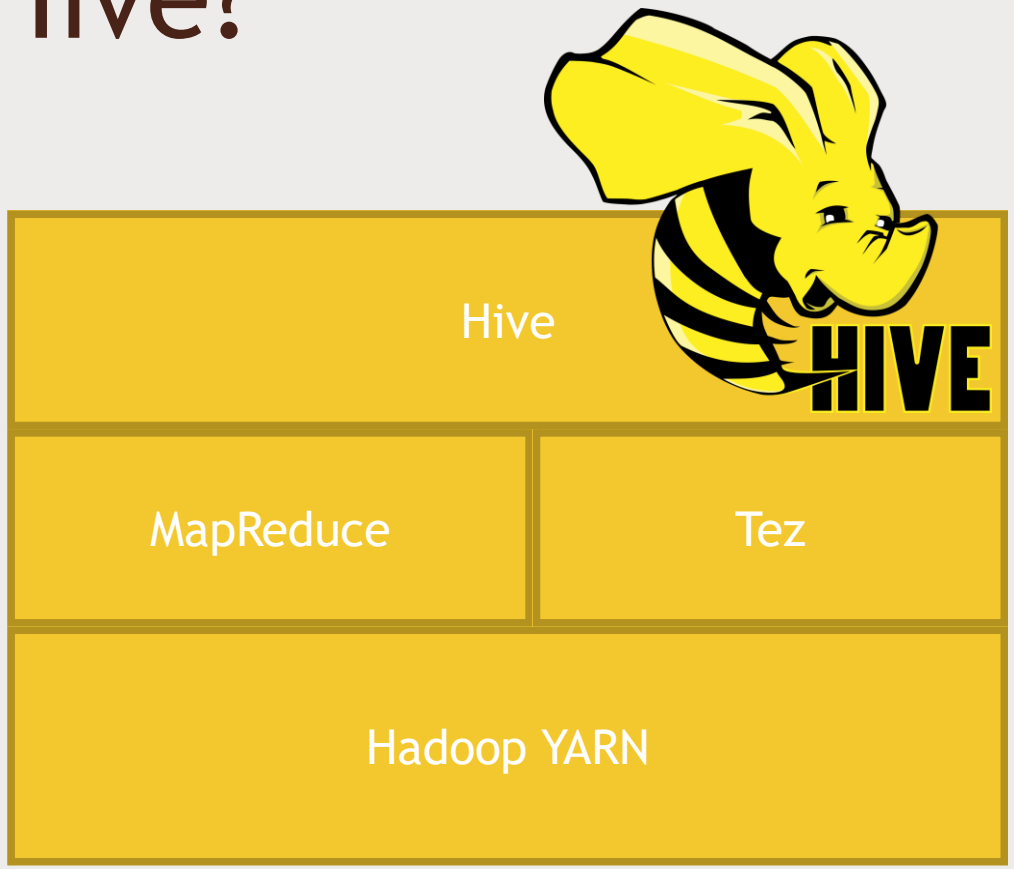
HIVE

Distributing SQL queries with Hadoop

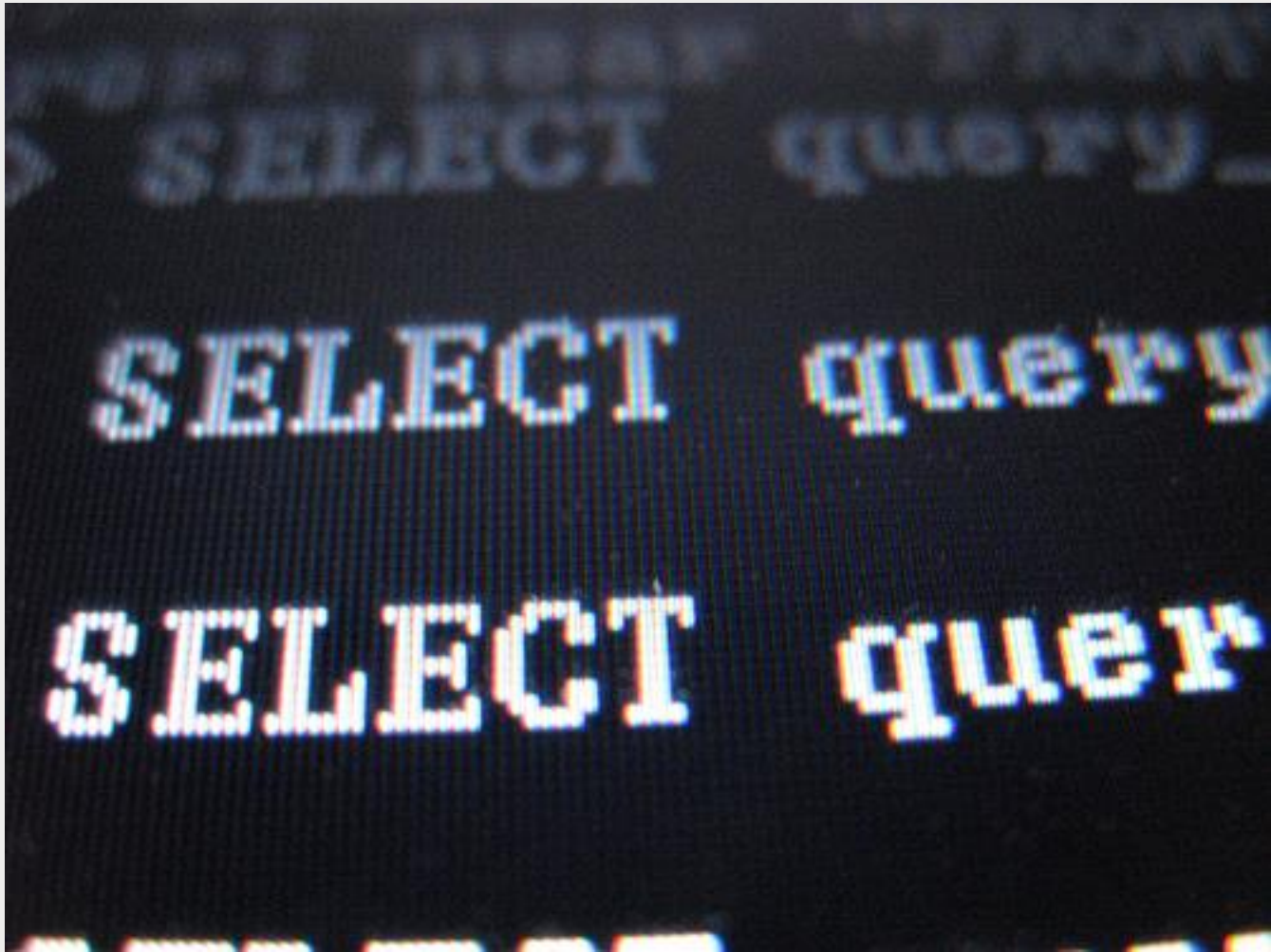
What is Hive?



Translates SQL queries to MapReduce or Tez jobs on your cluster!

Why Hive?

- Uses familiar SQL syntax (HiveQL)
- Interactive
- Scalable – works with “big data” on a cluster
  - *Really most appropriate for data warehouse applications*
- Easy OLAP queries – WAY easier than writing MapReduce in Java
- Highly optimized
- Highly extensible
  - *User defined functions*
  - *Thrift server*
  - *JDBC / ODBC driver*



Why not Hive?

- High latency – not appropriate for OLTP
- Stores data de-normalized
- SQL is limited in what it can do
  - *Pig, Spark allows more complex stuff*
- No transactions
- No record-level updates, inserts, deletes

Online Transaction Processing

HiveQL

- Pretty much MySQL with some extensions
- For example: views
  - *Can store results of a query into a "view", which subsequent queries can use as a table*
- Allows you to specify how structured data is stored and partitioned

Let’s just dive into an example.

HOW HIVE WORKS

Schema On Read

- Hive maintains a “metastore” that imparts a structure you define on the unstructured data that is stored on HDFS etc.

```
CREATE TABLE ratings (
    userID INT,
    movieID   INT,
    rating INT,
    time   INT)
ROW FORMAT DELIMTED
FIELDS TERMINATED BY ’\t’
STORED AS TEXTFILE;

LOAD DATA LOCAL INPATH ‘${env:HOME}/ml-100k/u.data’
OVERWRITE INTO TABLE ratings;
```

Where is the data?

- LOAD DATA
  - *MOVES data from a distributed filesystem into Hive*
- LOAD DATA LOCAL
  - *COPIES data from your local filesystem into Hive*
- Managed vs. External tables

```
CREATE EXTERNAL TABLE IF NOT EXISTS ratings (
      userID    INT,
      movieID   INT,
      rating    INT,
      time      INT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
LOCATION '/data/ml-100k/u.data';
```

1. external table is a good way to share data with others.
2. if you drop the table, the original file will still be there.

Partitioning

- You can store your data in partitioned subdirectories
  - *Huge optimization if your queries are only on certain partitions*

```
CREATE TABLE customers (
      name     STRING,
      address  STRUCT<street:STRING, city:STRING, state:STRING, zip:INT>
)
PARTITIONED BY (country STRING);
```

```
…/customers/country=CA/
…/customers/country=GB/
```

Ways to use Hive

- Interactive via hive> prompt / Command line interface (CLI)
- Saved query files
  - *hive -f /somepath/queries.hql*
- Through Ambari / Hue
- Through JDBC/ODBC server
- Through Thrift service
  - *But remember, Hive is not suitable for OLTP*
- Via Oozie


HIVE

HIVE CHALLENGE

Find the movie with the highest average rating

- Hint: AVG() can be used on aggregated data, like COUNT() does.
- Extra credit: only consider movies with more than 10 ratings