# Hive

Igor Yakushin
ivy2@uchicago.edu

# Hive: introduction

- Hive provides Hadoop with SQL access to data
- Hive server, sitting on top of HDFS and MapReduce, accepts connections from various Hive clients via, for example, ODBC, JDBC drivers and converts SQL into MapReduce jobs
- Hive supports a large subset of SQL including joins
- One can create indexes to improve performance
- When creating a table, one specifies where the data is stored: textfile, HBase table or any other of numerous data formats supported by Hadoop and residing on HDFS

# Hive: example

```
$ hive
hive> set hive.cli.print.current.db=true;
hive> CREATE DATABASE my1;
hive> USE my1;
hive (my1)> CREATE TABLE t1(W STRING, N INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
hive (my1)> LOAD DATA INPATH 'words.csv' INTO TABLE t1;
hive (my1)> select count(*) from t1;
hive (my1)> select max(N) from t1;
```

- To avoid collision, everybody should use a separate database named by username
- Interactive usage is only good for experimentation, don't use it for homework. Instead, prepare a sql script, for example, `my.sql`, and execute it as follows:

  ```
  hive -f my.sql > out.txt 2> err.txt
  ```

  and submit three files as homework: `my.sql`, `out.txt` and `err.txt`.
- You can also run sql on hive database as follows:

  ```
  hive --database my1 -e 'select count(*) from t1;'
  ```