Guest Lecture

Daniel Dao & Nick Buroojy
WHAT IS CIVITAS LEARNING

Civitas Learning

- Mid-sized startup
- Data driven company
- Education
“We partner with forward-thinking colleges and universities, harnessing the power of insight and action analytics to help a million more students learn well and finish strong.” – The Million More Mission
WHAT WE DO

• Work with institutions to provide insights through various applications
  • Inspire
Inspire for Faculty Demo
WHAT I DO

• My role in the company
• How my work is broken down
  • Product
    • Dev managers, PSMs, engineers
  • Frontend
    • Work with HTML/CSS/ReactJS
  • Backend
    • Writing APIs
    • Working with models
    • Writing SQL
    • Optimizing performance
    • Writing tests
HOW I USE DATABASES

elasticsearch

PostgreSQL

Amazon Redshift
Nick Buroojy

• Graduated from Carnegie Mellon
  • Bachelors in Computer Science

• Software Engineering
  • I've been working in Software for about 6 years
  • I've been at Civitas for three years
  • I’ve worked at Apple, Google, Civitas
Goals

At the end of this lecture, you will be able to:

• Describe the process Civitas uses to manipulate data.

• Describe the differences between column and row oriented data stores

• Explain how Redshift uses distributed compute for query performance

• Describe the use of the data layout options DIST_KEY and SORT_KEY
Extract

• As long as the data is in the tables, there are export commands that can simply dump the data to a file.
SELECT
  SPBPERS.SPBPERS_PIDM AS raw_person_id,
  SPBPERS.SPBPERS_BIRTH_DATE AS raw_birth_dt,
  SPBPERS.SPBPERS_DEAD_DATE AS raw_death_dt,
  SPBPERS.SPBPERS_SEX AS raw_gender,
  null AS raw_primary_language,
  null AS raw_country_of_origin
FROM
  src_banner_saturn.spbpers
Load

Flat file: Plain Text file that is non-hierarchical, usually in the form of CSV, or TSV. Each row represents one row in the database.
Data Flow

1. Web Server
2. Secure File Transfer Protocol
3. AWS
4. Red Shift
5. Postgres
Redshift Performance

- Columnar data storage
- Distributed data storage
- DIST_KEY
- SORT_KEY
- Parallel query execution
- COPY / UNLOAD
Columnar data storage

Row-oriented data store example:

<table>
<thead>
<tr>
<th>SSN</th>
<th>Name</th>
<th>Age</th>
<th>Addr</th>
<th>City</th>
<th>St</th>
</tr>
</thead>
<tbody>
<tr>
<td>101259797</td>
<td>SMITH</td>
<td>88</td>
<td>899 FIRST ST</td>
<td>JUNO</td>
<td>AL</td>
</tr>
<tr>
<td>892375862</td>
<td>CHIN</td>
<td>37</td>
<td>16137 MAIN ST</td>
<td>POMONA</td>
<td>CA</td>
</tr>
<tr>
<td>318370701</td>
<td>HANDU</td>
<td>12</td>
<td>42 JUNE ST</td>
<td>CHICAGO</td>
<td>IL</td>
</tr>
</tbody>
</table>

Source: [docs.aws.amazon.com](docs.aws.amazon.com)
Columnar data storage

Column-oriented data store example:

<table>
<thead>
<tr>
<th>SSN</th>
<th>Name</th>
<th>Age</th>
<th>Addr</th>
<th>City</th>
<th>St</th>
</tr>
</thead>
<tbody>
<tr>
<td>101259797</td>
<td>SMITH</td>
<td>88</td>
<td>899 FIRST ST</td>
<td>JUNO</td>
<td>AL</td>
</tr>
<tr>
<td>892375862</td>
<td>CHIN</td>
<td>37</td>
<td>16137 MAIN ST</td>
<td>POMONA</td>
<td>CA</td>
</tr>
<tr>
<td>318370701</td>
<td>HANDU</td>
<td>12</td>
<td>42 JUNE ST</td>
<td>CHICAGO</td>
<td>IL</td>
</tr>
</tbody>
</table>

Source: docs.aws.amazon.com
Distributed data storage

• Why?
• DB constraints
  • Disk
  • CPU
  • Network
Partial aggregations

- **SUM**

\[ 10 + 21 + 15 + 6 \]

136

3567

168

6
Partial aggregations

• COUNT

```
3 + 4 + 3 + 1
```

```
1 3 6
3 5 6 7
1 6 8
3 4 3 1
6
```
Partial aggregations

• AVG = SUM / COUNT
Partial aggregations

• Redshift can distribute
  • AVG
  • SUM
  • COUNT
  • MAX
  • MIN
  • STDDEV
  • ...

• More challenging (slower)
  • COUNT DISTINCT
  • ORDER BY x LIMIT n
DIST KEY

• Allows Redshift user to specify which records are on the same node
• Used to keep balanced
• Used for join locality
  • Can perform a join without “shuffling”. That is, sending data between nodes.
SORT KEY

• Orders of storage for records
• Allows queries to skip ranges
• Allows for faster joins (merge vs. hash)
• Faster ORDER BY queries
PRIMARY KEY

• Redshift doesn’t enforce primary keys or foreign keys
• Primary key must be non-null and unique
• Used by query optimizer
• Civitas checks our keys after building each table
  • \( \text{COUNT}(pk) = \text{COUNT}(\ast) = \text{COUNT}(\text{DISTINCT } pk) \)
COPY

- Loads flat file data from bulk storage (S3) into Redshift
- Each node loads some parts of the data
- Master doesn’t touch the data, and is not a bottleneck
- Unload: opposite direction. Redshift -> S3
Summary

• Process Civitas uses to manipulate data.
• Columnar data layout
• Distributed query aggregations
• Data layout options

• Careers at Civitas Learning
Questions?