Course NDBI040: **Big Data Management and NoSQL Databases**

Practice 03:

Column-Family Stores: Cassandra

Martin Svoboda

1. 12. 2015

Faculty of Mathematics and Physics, Charles University in Prague

Outline

- Column-family stores
  - **Apache Cassandra**
- Assignment 3

Apache Cassandra Tutorial

Column-Family Store

Installation

- **Downloads**
  - http://cassandra.apache.org/download/
    - Debian
  - http://www.planetcassandra.org/cassandra/
    - Windows, Mac OS X, Ubuntu, …
- **Installation**

Clients

- **Cassandra CLI Utility**
  - cassandra-cli
- **Cassandra CQL Shell**
  - cqlsh
- **Cassandra OpsCenter**
  - Cluster monitoring and management
  - Web application
    - http://localhost:8888/opscenter/index.html

Connection

- **SSH and SFTP access**
  - **nosql.ms.mff.cuni.cz:42222**
  - Login and password: as usual
- Tools
  - PuTTY and WinSCP
- Clients
  - **`/home/svoboda/cassandra/apache-cassandra-2.0.3/bin/`**
    - **`./cqlsh`**

Keyspaces

- Create a new keyspace
  - ```
    CREATE KEYSPACE name
    WITH replication =
      {'class' : 'SimpleStrategy',
       'replication_factor' : 1};
    ```
    - Use your login name as a name for your keyspace
- List all keyspaces
  - ```
    DESCRIBE KEYSPACES;
    ```
  - ```
    SELECT * FROM system.schema_keyspaces;
    ```
- Keyspace usage
  - ```
    USE name;
    ```

Tables

- Create a table for users
  - ```
    CREATE TABLE users1 (
       id int,
       fname text,
       lname text,
       email text,
       PRIMARY KEY (id)
    );
    ```
- View table definition
  - ```
    DESCRIBE TABLE users1;
    ```

Tables

- Insert new users
  - ```
    INSERT INTO users1
      (id, fname, lname, email)
    VALUES (1, 'Irena', 'Holubova',
      'holubova@ksi.mff.cuni.cz');
    ```
  - …
- Browse existing users
  - ```
    SELECT * FROM users1;
    ```
  - ```
    SELECT * FROM users1 WHERE (id = 1);
    ```
  - ```
    SELECT * FROM users1
    WHERE (id IN (1, 2, 3));
    ```

Tables

- Find particular users
  - ```
    SELECT * FROM users1
    WHERE (lname = 'Holubova');
    ```
- Create a new index
  - ```
    CREATE INDEX ON users1 (lname);
    ```
- Find particular users
  - ```
    SELECT * FROM users1
    WHERE
       (lname = 'Holubova') AND
       (fname = 'Irena')
    ALLOW FILTERING;
    ```
    ```

Tables

- Update and remove users
  - ```
    UPDATE users1
    SET email = 'irena.holubova@mff.cuni.cz'
    WHERE (id = 1);
    ```
  - ```
    DELETE FROM users1 WHERE (id = 1);
    ```

Collections

- Create a new table for users
  - ```
    CREATE TABLE users2 (
       id int,
       fname text,
       lname text,
       emails set<text>,
       cities list<text>,
       phones map<text, text>,
       PRIMARY KEY (id)
    );
    ```
    ```

Collections

- Insert new users
  - ```
    INSERT INTO users2
      (id, fname, lname, emails, cities, phones)
    VALUES (
      1,
      'Irena',
      'Holubova',
      {'holubova@ksi.mff.cuni.cz'},
      ['Prague', 'Zlin'],
      {'work' : '+420 251 554 316'}
    );
    ```
  - …
    ```

Collections

- Update existing users
  - ```
    UPDATE users2
    SET
      emails =
        emails + {'irena.holubova@mff.cuni.cz'},
      cities = cities - ['Zlin'],
      phones['work'] = '+420 951 554 316'
    WHERE (id = 1);
    ```
  - ```
    DELETE
      cities[0],
      phones['work']
    FROM users2 WHERE (id = 1);
    ```

Comments

- Single line
  - `-- comment`
  - `// comment`
- Multiple lines
  - `/* comment */`

References

- Cassandra 3.0
  - http://docs.datastax.com/en/cassandra/3.0/cassandra/cassandraAbout.html
- **Cassandra 2.0**
  - http://docs.datastax.com/en/cassandra/2.0/cassandra/gettingStartedCassandraIntro.html
- CQL 3.3 (Cassandra 2.2 and later)
  - http://docs.datastax.com/en/cql/3.3/cql/cqlIntro.html
- **CQL 3.1** (Cassandra 2.0 and 2.1)
  - http://docs.datastax.com/en/cql/3.1/cql/cql_intro_c.html

Assignment 3

Column-Family Stores: Apache Cassandra

Assignment 3

- **Create a non-trivial real-world application** that would work with Apache Cassandra
  - Choose any database domain
    - E.g. cinema tickets, flight booking, study system, …
  - Create a commented script with CQL statements
- **Submit your solution** by e-mail
  - svoboda@ksi.mff.cuni.cz
  - Deadline: 14. 12. 2015

Assignment 3

- **Requirements**
  - Create a keyspace with at least 3 tables
  - Insert at least 5 rows into each of them (on average)
  - Use all three types of collections
    - Set, list, map
  - Define at least 2 indices
  - Perform at least 5 updates and 5 deletions
    - Involve non-trivial operations on collections as well
  - Express at least 5 meaningful SELECT queries
    - Use WHERE, ORDER BY, LIMIT and ALLOW FILTERING clauses