MySQL [SQL DB] Evaluation #29

Edit New issue

Closed **ppyordanov** opened this issue on Nov 1, 2014 · 6 comments

ppyordanov commented on Nov 1, 2014

The SQL database management system is going to be evaluated based on a number of different criteria. Results will be compared to the performance of a NoSQL alternative(for example, MongoDB or CouchDB). THe following tests are going to be carried out:

- Data Write:
  - insert around 60 000 records in the key DB table (*DataReading*)
  - single instance of type data reading insertion
  - updating a single instance of type data reading
- Data Read:
  - retrieve all records in *DataReading* table
  - retrieve a single instance of type data reading from the table
- Additional Tests:
  - delete a single record from the table

NoSQL (MongoDB) Evaluation details: #30

DBMS choice information: #27

Labels

enhancement

Milestone

Sensor Data Set

Assignee

ppyordanov

Notifications

Unsubscribe

You're receiving notifications because you modified the open/close state.

1 participant

Lock issue

ppyordanov added the **enhancement** label on Nov 1, 2014

ppyordanov self-assigned this on Nov 1, 2014

ppyordanov added this to the **Sensor Data Set** milestone on Nov 1, 2014

ppyordanov referenced this issue on Nov 1, 2014

NoSQL DBMS Evaluation #30 Closed

ppyordanov referenced this issue from a commit on Nov 2, 2014

`[READ_SAMPLE_DATA] loaded 5000 records in DataReading, data deseriali… ···` `e45d9f6`

ppyordanov referenced this issue from a commit on Nov 2, 2014

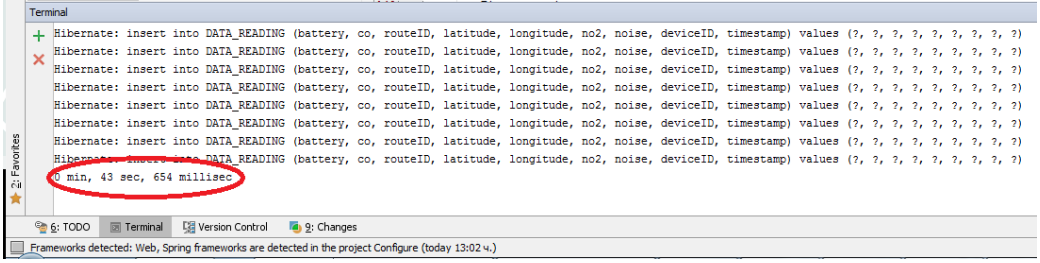
`[SQL] data insertion 70K records in DB, tests under way` `52c1181`

ppyordanov commented on Nov 2, 2014

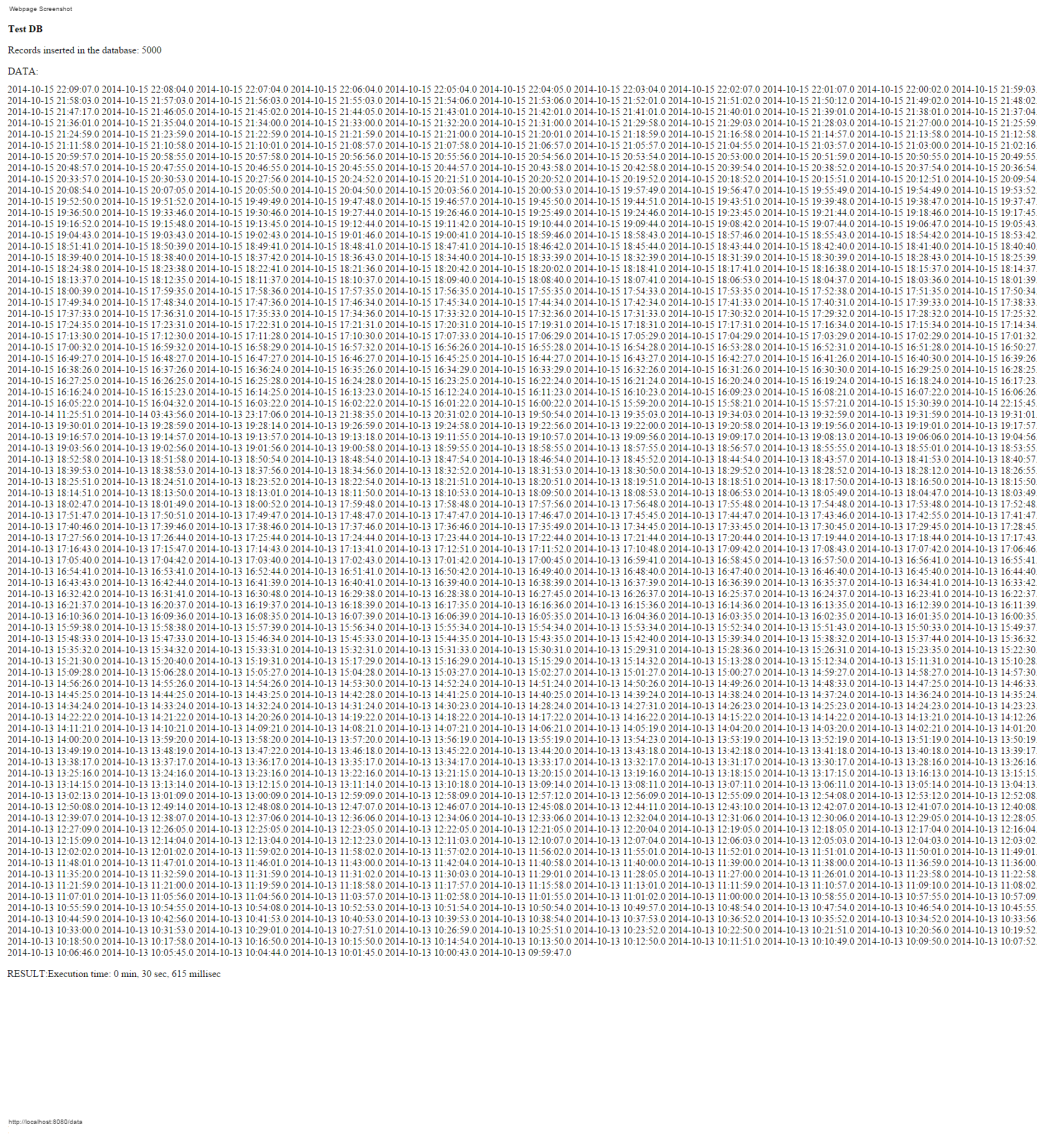
- Data deserialization from sample_data.json file now works. 5000 readings were inserted in the table for the first test:

```
[INFO] Started Jetty Server
Hibernate: insert into DEVICE (description, kit_version, mac_address, title) values (?, ?, ?, ?)
Readings loaded from fil: 5000
5000
Hibernate: insert into DATA_READING (battery, co, routeID, latitude, longitude, no2, noise, deviceID, timestamp) values (?, ?, ?, ?, ?, ?, ?, ?, ?)
```

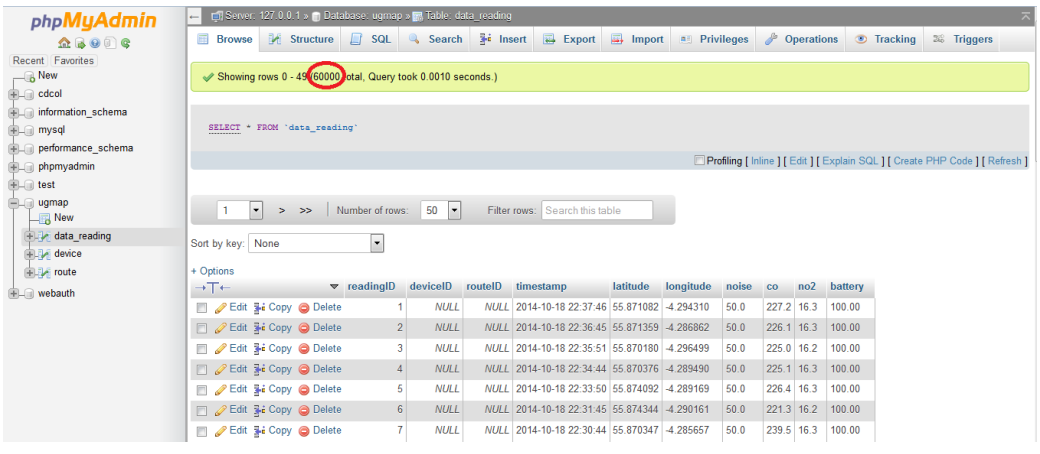
- Execution time is around 30-40 seconds, not bad at all , around ***6 ms per record***:



- Example output in the controller view on front end (again, some data has been removed for improved readability):

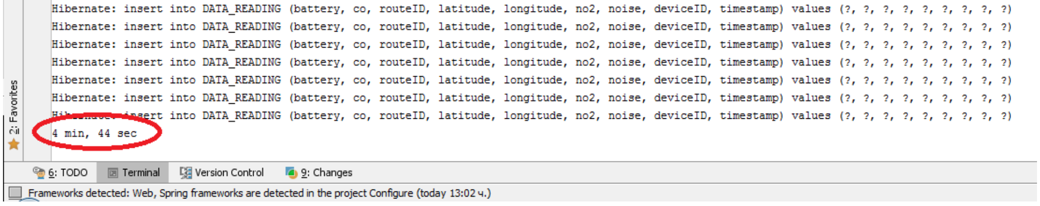


- For the complete benchmarking, 60, 000 data readings were inserted in the *DataReading* table using the Hibernate session manager and the data reading DAO (Data Access Object):



- This was even faster on average, taking ***4 min, 44 sec*** in total, which is ***4,73 milliseconds*** per insertion:

284 000 millisec / 60 000 insertions = 4,7333(33..) milliseconds



PY **ppyordanov** commented on Nov 2, 2014

OUTPUT /some data removed for readability/

```
Test DB
Records inserted in the database: 5000

DATA:

2014-10-13 10:45:55.0 2014-10-13 10:44:59.0 2014-10-13 10:42:56.0 2014-10-13 10:41:53.0 2014-10-13

RESULT:Execution time: 0 min, 30 sec, 615 millisec```
```

PY **ppyordanov** referenced this issue from a commit on Nov 2, 2014

PY [RESTRUCTURE] added additional benchmarking tests   692e1e7

PY **ppyordanov** commented on Nov 2, 2014

FINAL RESULTS /NO INDICES/

Two consecutive tests were carried out to ensure that the information is accurate as performance depends on system resources.

- *FULL TEST 1*:

Test DB

Records inserted in the database:

DATA: skipped

RESULT:

Insert single record:
Single insert in tables DEVICE and DATA_READING: Execution time: 0 min, 0 sec, 260 millisec

Insert real data (SIM):
Execution time: 4 min, 43 sec, 240134 millisec **Data size:**
12 * 5000

Update single record:
Execution time: 0 min, 0 sec, 70 millisec

Get single record:
Execution time: 0 min, 0 sec, 46 millisec

Delete single record:
Execution time: 0 min, 0 sec, 30 millisec

Get all records:
Execution time: 0 min, 5 sec, 72 millisec

TOTAL:

Execution time: 4 min, 48 sec, 240612 millisec

```
Test DB

Records inserted in the database:
```

```
DATA: skipped

RESULT:

Insert single record:
Single insert in tables DEVICE and DATA_READING: Execution time: 0 min, 0 sec, 260 millisec

Insert real data (SIM):
Execution time: 4 min, 43 sec, 240134 millisec Data size:
12 * 5000

Update single record:
Execution time: 0 min, 0 sec, 70 millisec

Get single record:
Execution time: 0 min, 0 sec, 46 millisec

Delete single record:
Execution time: 0 min, 0 sec, 30 millisec

Get all records:
Execution time: 0 min, 5 sec, 72 millisec

TOTAL:

Execution time: 4 min, 48 sec, 240612 millisec
```

- *FULL TEST 2*:

Test MySQL DB

DATA: skipped

RESULT:

Insert single record:
Single insert in tables DEVICE and DATA_READING: Execution time: 0 min, 0 sec, 246 millisec

Insert real data (SIM):
Execution time: 4 min, 9 sec, 240310 millisec **Data size:**
12 * 5000

Update single record:
Execution time: 0 min, 0 sec, 54 millisec

Get single record:
Execution time: 0 min, 0 sec, 25 millisec

Delete single record:
Execution time: 0 min, 0 sec, 17 millisec

Get all records:
Execution time: 0 min, 3 sec, 448 millisec

TOTAL:

Execution time: 4 min, 13 sec, 240101 millisec

```
Test MySQL DB

DATA: skipped

RESULT:

Insert single record:
Single insert in tables DEVICE and DATA_READING: Execution time: 0 min, 0 sec, 246 millisec

Insert real data (SIM):
Execution time: 4 min, 9 sec, 240310 millisec Data size:
12 * 5000
```

```
Update single record:
Execution time: 0 min, 0 sec, 54 millisec

Get single record:
Execution time: 0 min, 0 sec, 25 millisec

Delete single record:
Execution time: 0 min, 0 sec, 17 millisec

Get all records:
Execution time: 0 min, 3 sec, 448 millisec

TOTAL:

Execution time: 4 min, 13 sec, 240101 millisec
```

PY

ppyordanov commented on Nov 2, 2014

RESULTS

AVG_TOTAL = (288 + 253 sec)/2 = 270,5 sec

TOTAL_OPERATIONS ≈ 60 010

AVG_PER_OPERATION = AVG_TOTAL/TOTAL_OPERATIONS = 270 500 millisec / 60 010 op = ***4.507582 millisec/op***

PY

ppyordanov commented on Nov 2, 2014

- *FULL TEST 3: 5000 INSERTIONS*

Test MySQL DB

DATA: skipped

RESULT:

Insert single record:
Single insert in tables DEVICE and DATA_READING: Execution time: 0 min, 0 sec, 222 millisec

Insert real data (SIM):
Execution time: 0 min, 37 sec, 115 millisec **Data size:**
1 * 5000

Update single record:
Execution time: 0 min, 0 sec, 84 millisec

Get single record:
Execution time: 0 min, 0 sec, 31 millisec

Delete single record:
Execution time: 0 min, 0 sec, 90 millisec

Get all records:
Execution time: 0 min, 0 sec, 989 millisec

TOTAL:

Execution time: 0 min, 38 sec, 531 millisec

```
Test MySQL DB
```

```
MySQL

DATA: skipped

RESULT:

Insert single record:
Single insert in tables DEVICE and DATA_READING: Execution time: 0 min, 0 sec, 222 millisec

Insert real data (SIM):
Execution time: 0 min, 37 sec, 115 millisec Data size:
1 * 5000

Update single record:
Execution time: 0 min, 0 sec, 84 millisec

Get single record:
Execution time: 0 min, 0 sec, 31 millisec

Delete single record:
Execution time: 0 min, 0 sec, 90 millisec

Get all records:
Execution time: 0 min, 0 sec, 989 millisec

TOTAL:

Execution time: 0 min, 38 sec, 531 millisec
```

FULL TEST 4: 200 000 INSERTIONS

Test MySQL DB

DATA: skipped

RESULT:

Insert single record:
Single insert in tables DEVICE and DATA_READING: Execution time: 0 min, 0 sec, 267 millisec

Insert real data (SIM):
Execution time: 11 min, 39 sec, 643 millisec **Data size:**
40 * 5000

Update single record:
Execution time: 0 min, 0 sec, 74 millisec

Get single record:
Execution time: 0 min, 0 sec, 52 millisec

Delete single record:
Execution time: 0 min, 0 sec, 20 millisec

Get all records:
Execution time: 0 min, 13 sec, 615 millisec

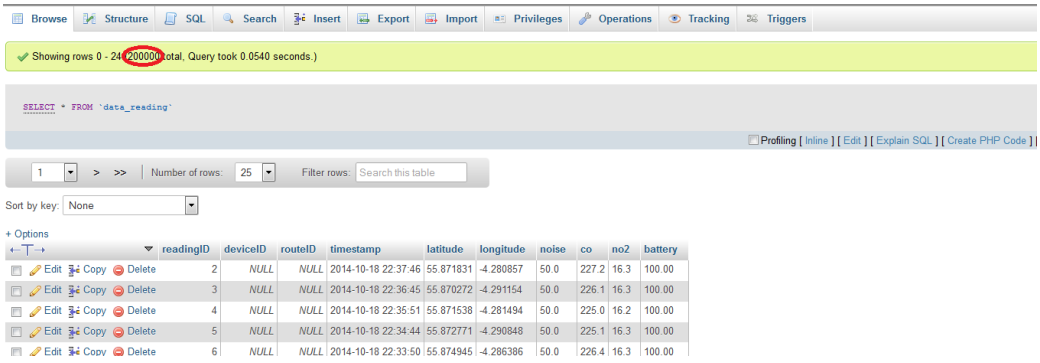
TOTAL:

Execution time: 11 min, 53 sec, 674 millisec


```
Test MySQL DB
MySQL

DATA: skipped

RESULT:

Insert single record:
```

```
Single insert in tables DEVICE and DATA_READING: Execution time: 0 min, 0 sec, 267 millisec

Insert real data (SIM):
Execution time: 11 min, 39 sec, 643 millisec Data size:
40 * 5000

Update single record:
Execution time: 0 min, 0 sec, 74 millisec

Get single record:
Execution time: 0 min, 0 sec, 52 millisec

Delete single record:
Execution time: 0 min, 0 sec, 20 millisec

Get all records:
Execution time: 0 min, 13 sec, 615 millisec

TOTAL:

Execution time: 11 min, 53 sec, 674 millisec
```

- MySQL Admin page:



AVG_TIME/ OP = 713 674 millisec/ 200 010 op ≈ 3.5681 millisec

ppyordanov referenced this issue from a commit on Nov 2, 2014

[MYSQL_PERFORMANCE] uploaded benchmarks + results 86fceb1

PY

ppyordanov commented on Nov 2, 2014

Conclusions

- Retrieving all records might be a bit slow in the case of extreme data size manipulation (200 000 records in the DATA_READING table):

```
Test MySQL DB
MySQL

DATA: skipped

RESULT:

Insert single record:

Insert real data (SIM):
Data size:
40 * 5000

Update single record:

Get single record:

Delete single record:

Get all records:
Execution time: 0 min, 12 sec, 894 millisec

TOTAL:
```

```
Execution time: 0 min, 12 sec, 894 millisec
```

Each consecutive retrieval in the same session reduces the time by half approximately (session caching).

And for 60 000 entries complete retrieval is around 3-5 seconds:

```
Test MySQL DB
MySQL

DATA: skipped

RESULT:

Insert single record:

Insert real data (SIM):
Data size:
12 * 5000

Update single record:

Get single record:

Delete single record:

Get all records:
Execution time: 0 min, 4 sec, 31 millisec

TOTAL:

Execution time: 0 min, 4 sec, 31 millisec
```

ppyordanov closed this on Nov 2, 2014

ppyordanov reopened this on Nov 2, 2014

ppyordanov closed this on Nov 10, 2014

Write Preview Markdown supported Edit in fullscreen

Leave a comment

Attach images by dragging & dropping, selecting them, or pasting from the clipboard.

Reopen issue Comment