慎用yii2 中的batch/each方法

Created by 杨超, last modified on 2018 Mar 16

> 先说观点，基于Mysql的数据库中, 希望大家yii2中慎用 最好不用 batch或者each方法，有性能问题 而且比较隐蔽

什么是BatchQuery?

参考文档 http://www.yiiframework.com/doc-2.0/guide-db-query-builder.html

简而言之，BatchQuery 设计的本意是为了方便进行批量的查询，使用起来比较方便，但是注意下图中的红字

Batch Query

When working with large amounts of data, methods such as yii\db\Query::all() are not suitable because they require loading the whole query result into the client's memory. To solve this issue Yii provides batch query support. The server holds the query result, and the client uses a cursor to iterate over the result set one batch at a time.

Warning: There are known limitations and workarounds for the MySQL implementation of batch queries. See below.

Batch query can be used like the following:

```
use yii\db\Query;

$query = (new Query())
    ->from('user')
    ->orderBy('id');

foreach ($query->batch() as $users) {
    // $users is an array of 100 or fewer rows from the user table
}

// or to iterate the row one by one
foreach ($query->each() as $user) {
    // data is being fetched from the server in batches of 100,
    // but $user represents one row of data from the user table
}
```

运行实例

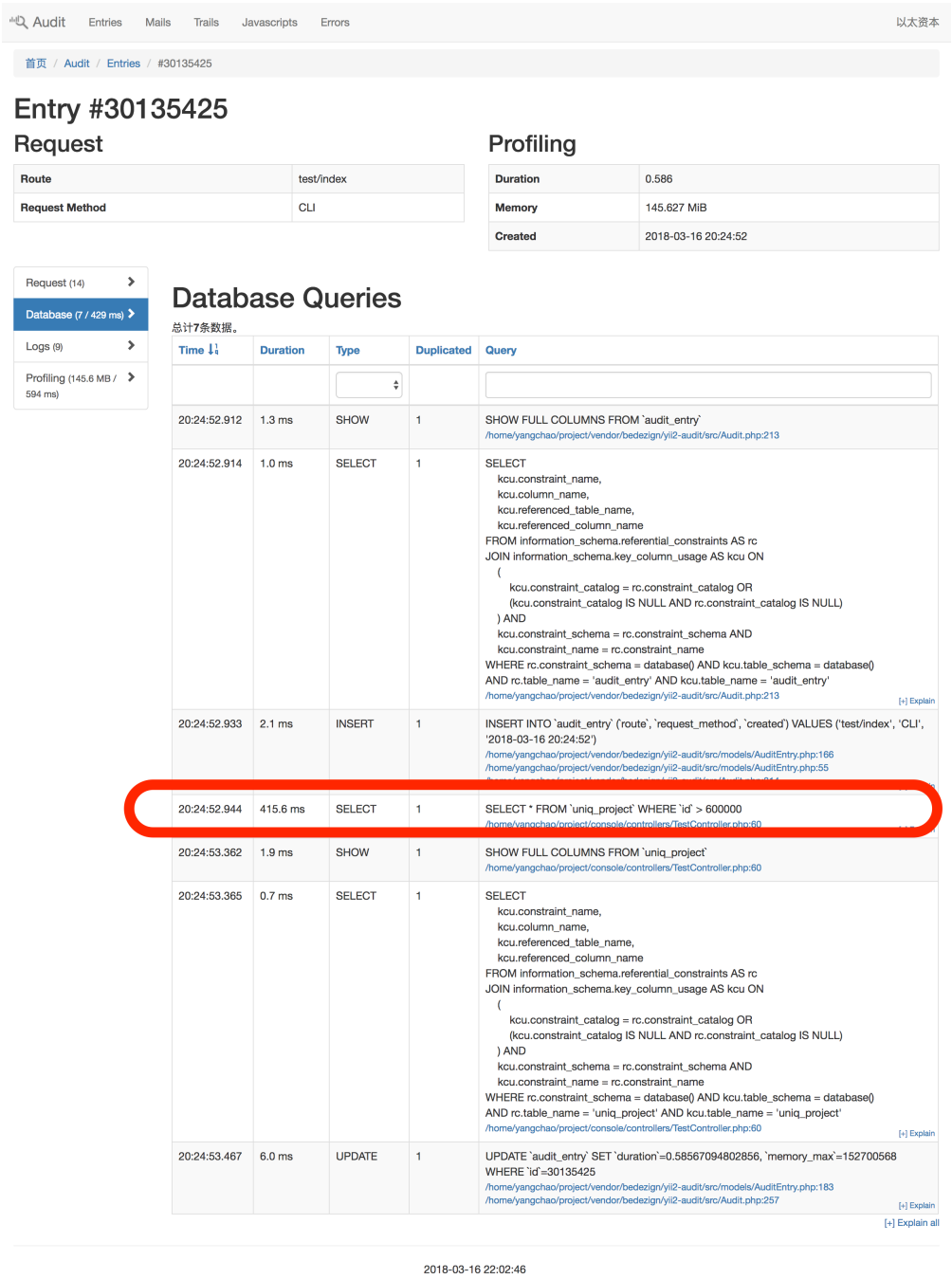
原始代码被我手抖删掉了，大家看伪代码意会就好...

```
$query = Uniqproject::find()->where(['>', 'id', 600000]);
foreach ($query->each(10) as $up) {
    // $up->method(); // do nothing here, just for test
}
```

运行结果

参考下图实例中的红色sql，发现一行让我big shock的执行结果，幸好我没直接写id>1 不然就直接db挂了可能

看了each直接执行了原生sql，把所有的结果load到了内存，然后在row by row的读结果，所以sql用了400ms+，内存用了145MB


Audit
Entries
Mails
Trails
Javascripts
Errors
以太资本
首页 / Audit / Entries / #30135425
Entry #30135425
Request
Route
test/index
Request Method
CLI
Profiling
Duration
0.586
Memory
145.627 MiB
Created
2018-03-16 20:24:52
Request (14)
Database (7 / 429 ms)
Logs (9)
Profiling (145.6 MB / 594 ms)
Database Queries
总计7条数据。
Time
Duration
Type
Duplicated
Query
20:24:52.912
1.3 ms
SHOW
1
SHOW FULL COLUMNS FROM `audit_entry`
/home/yangchao/project/vendor/bedezign/yii2-audit/src/Audit.php:213
20:24:52.914
1.0 ms
SELECT
1
SELECT
kcu.constraint_name,
kcu.column_name,
kcu.referenced_table_name,
kcu.referenced_column_name
FROM information_schema.referential_constraints AS rc
JOIN information_schema.key_column_usage AS kcu ON
(
kcu.constraint_catalog = rc.constraint_catalog OR
(kcu.constraint_catalog IS NULL AND rc.constraint_catalog IS NULL)
) AND
kcu.constraint_schema = rc.constraint_schema AND
kcu.constraint_name = rc.constraint_name
WHERE rc.constraint_schema = database() AND kcu.table_schema = database()
AND rc.table_name = 'audit_entry' AND kcu.table_name = 'audit_entry'
/home/yangchao/project/vendor/bedezign/yii2-audit/src/Audit.php:213
[+] Explain
20:24:52.933
2.1 ms
INSERT
1
INSERT INTO `audit_entry` (`route`, `request_method`, `created`) VALUES ('test/index', 'CLI', '2018-03-16 20:24:52')
/home/yangchao/project/vendor/bedezign/yii2-audit/src/models/AuditEntry.php:166
/home/yangchao/project/vendor/bedezign/yii2-audit/src/models/AuditEntry.php:55
20:24:52.944
415.6 ms
SELECT
1
SELECT * FROM `uniq_project` WHERE `id` > 600000
/home/yangchao/project/console/controllers/TestController.php:60
20:24:53.362
1.9 ms
SHOW
1
SHOW FULL COLUMNS FROM `uniq_project`
/home/yangchao/project/console/controllers/TestController.php:60
20:24:53.365
0.7 ms
SELECT
1
SELECT
kcu.constraint_name,
kcu.column_name,
kcu.referenced_table_name,
kcu.referenced_column_name
FROM information_schema.referential_constraints AS rc
JOIN information_schema.key_column_usage AS kcu ON
(
kcu.constraint_catalog = rc.constraint_catalog OR
(kcu.constraint_catalog IS NULL AND rc.constraint_catalog IS NULL)
) AND
kcu.constraint_schema = rc.constraint_schema AND
kcu.constraint_name = rc.constraint_name
WHERE rc.constraint_schema = database() AND kcu.table_schema = database()
AND rc.table_name = 'uniq_project' AND kcu.table_name = 'uniq_project'
/home/yangchao/project/console/controllers/TestController.php:60
[+] Explain
20:24:53.467
6.0 ms
UPDATE
1
UPDATE `audit_entry` SET `duration`=0.58567094802856, `memory_max`=152700568
WHERE `id`=30135425
/home/yangchao/project/vendor/bedezign/yii2-audit/src/models/AuditEntry.php:183
/home/yangchao/project/vendor/bedezign/yii2-audit/src/Audit.php:257
[+] Explain
[+] Explain all
2018-03-16 22:02:46

源码分析

本质上跟踪到BatchQueryResult里的这段，真可谓是直接暴力，直接执行query了

```php
/**
 * Fetches the next batch of data.
 * @return array the data fetched
 */
protected function fetchData()
{
    if ($this->_dataReader === null) {
        $this->_dataReader = $this->query->createCommand($this->db)->query();
    }
```

总结

Limitations of batch query in MySQL

MySQL implementation of batch queries relies on the PDO driver library. By default, MySQL queries are `buffered`. This defeats the purpose of using the cursor to get the data, because it doesn't prevent the whole result set from being loaded into the client's memory by the driver.

> **Note:** When `libmysqlclient` is used (typical of PHP5), PHP's memory limit won't count the memory used for result sets. It may seem that batch queries work correctly, but in reality the whole dataset is loaded into client's memory, and has the potential of using it up.

To disable buffering and reduce client memory requirements, PDO connection property `PDO::MYSQL_ATTR_USE_BUFFERED_QUERY` must be set to `false`. However, until the whole dataset has been retrieved, no other query can be made through the same connection. This may prevent `ActiveRecord` from making a query to get the table schema when it needs to. If this is not a problem (the table schema is cached already), it is possible to switch the original connection into unbuffered mode, and then roll back when the batch query is done.

```
Yii::$app->db->pdo->setAttribute(\PDO::MYSQL_ATTR_USE_BUFFERED_QUERY, false);

// Do batch query

Yii::$app->db->pdo->setAttribute(\PDO::MYSQL_ATTR_USE_BUFFERED_QUERY, true);
```

> **Note:** In the case of MyISAM, for the duration of the batch query, the table may become locked, delaying or denying write access for other connections. When using unbuffered queries, try to keep the cursor open for as little time as possible.

If the schema is not cached, or it is necessary to run other queries while the batch query is being processed, you can create a separate unbuffered connection to the database:

这里是原因，简单翻译过来就是说 php用的mysqlclient lib有些问题，可能会直接执行sql的数据 （不加任何limit/offset），直接把全量的数据导入到memory里 并且不受php的memory限制

当然也有所谓的解决方案，就是不开启PDO的buffer，但是这样又会造成其他的问题，总之 我觉得比较麻烦

综合上述，我个人建议大家在使用Mysql的时候弃用batch/each方法，根据实际情况按照id/time进行批量查询操作