# mycat压测和结果

## 1.4版本和1.3版本压测

1）测试案例一，测试说明：100个线程，1000万数据，三分片，单节点写入，三节点随机读，测试工具以mycat社区发布的stand脚本压测

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1.3版本 | 1.4版本 | 测试结果说明 | 备注 |
| insert | tps:71890 | tps:76863 |  |  |
| update | tps:14222 | tps:12577 |  |  |
| select | qps:113073 | qps:128122 |  |  |

2）测试案例二，测试说明：500个线程，1000万数据，三分片，单节点写入，三节点随机读，测试工具以mycat社区发布的stand脚本压测

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1.3版本 | 1.4版本 | 测试结果说明 | 备注 |
| insert | tps:69881 | tps:75700 |  |  |
| update | tps:15501 | tps:97943 |  | 1.4更新性能更高 |
| select | qps:154837 | qps:158524 |  |  |

结果总结：500线程下mycat1.4版本的更新并发更高，其他压测总体性能相差不大

## docker下的mysql和物理机上的mysql压测

测试案例一，测试说明：500个线程，1000万数据，单个mysql，测试工具以mycat社区发布的stand脚本压测；

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Docker版本 | 物理机版本 | 测试结果说明 | 备注 |
| insert | tps: 156006 | tps: 166389 |  |  |
| update | tps: 14335 | tps:50479 |  |  |
| select | qps:207346 | qps:217865 |  |  |

测试案例二，测试说明：单个mysql ，20个仓库，100个线程，使用tpcc的方式压测；

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Docker版本 | 物理机版本 | 测试结果说明 | 备注 |
| tpmc | 57703 | 39977 | 15核CPU，总CPU不到50%，存在一定的wait | 都存在一定的1205, HY000, Lock wait timeout exceeded |

## 自定义sql测试

根据xx用户的场景，选出一部分复杂查询SQL来做自定义SQL的测试。

目前只做了一个自定义SQL场景，user\_user\_base\_sb和user\_user\_detail\_sr联合查询。200个线程，qps 89365，从CPU使用率来说，mycat占用400%，而9个mysql服务器将近在150%左右，总共的CPU使用率是mycat的4倍左右。这个场景mycat可以很大程度上起到扩容、扩并发的作用。

## 高可用和破坏性测试

1. 压测过程中，正常停止某一个分片的mariadb集群中的一个读节点，检查这个过程中mycat是否可以很好的切换读写服务器，并检查数据库重启以后的情况；

ERROR c.c.t.TpccStatements [tpcc-thread-31] ROLLBACK FAILED

 java.sql.SQLException: Connection is closed!

1. 压测过程中，正常停止某一个分片的mariadb集群中的写节点，检查这个过程中mycat是否可以很好的切换读写服务器，并检查数据库重启以后的情况；
2. 压测过程中，kill掉某一个分片的mariadb集群中的读节点，检查这个过程中mycat是否可以很好的切换读写服务器，并检查数据库重启以后的情况；
3. 压测过程中，kill某一个分片的mariadb集群中的写节点，检查这个过程中mycat是否可以很好的切换读写服务器，并检查数据库重启以后的情况；
4. 压测过程中，正常停止掉某一个分片的mariadb集群中的两个读节点，检查这个过程中mycat是否可以很好的切换读写服务器，并检查数据库重启以后的情况；
5. 压测过程中，正常停止掉某一个分片的mariadb集群中的所有节点，检查这个过程中mycat是否可以很好的切换读写服务器，并检查数据库重启以后的情况；

对于MyCat和Mariadb集群的高可用，MyCat社区给出的意见是最好reload mycat。

这个使用应用程序做读、写切换基本上无法测试下去，因为压测过程中，压测程序不会重新连接mycat，所以在mycat与mysql连接更新以后，压测程序与mycat的连接还是不变的。在mycat发生切换后，新mycat连接是正常的。

而在我们实际的应用中，目前应用每隔一段时间会重连mysql，所以这个时间设置合理，是可以降低这种故障切换带来的影响的。

目前我们采取的写策略是只在一个点做写入，在down掉一个节点的情况下，也是可以继续跑数据，不会产生脑裂的情况。

## 长时间测试

1）1.3版本稳定性能压测

1.3版本使用tpcc-java工具，20个仓库，取3的模的分片规则，60个并发，3个分片，热身一小时，压测36小时。

 整个压测过程中，出现了mysql硬盘空间已满、mysql重启等问题，导致压测程序重启过几次，但是没重启过mycat。压测过程中，mycat的CPU（32核）在300%左右波动，虽然说压力不是太大，这也与tpcc的仓库数20个有一定的关系，导致并发上不去。并发高的时候，容易出现Lock wait timeout exceeded。

2）1.4 release版本的稳定性压测

使用tpcc-java工具，20个仓库，取3的模的分片规则，60个并发，3个分片，热身一小时，压测24小时。中间没出现任何异常和问题。

 总体来说，mycat在整个时间还是挺稳定的。

## 测试中出现数据库的一些问题

1. tpcc-java压测过程中出现java.sql.SQLException: Lock wait timeout exceeded; try restarting transaction
2. 硬盘满的时候，数据库commit等待很长时间，其他show
3. 压测过程中，停止mysql出现了ERROR!The server quit without updating PID file (/opt/local/mysql/mysql.pid)，无法启动，手工删除/opt/local/mysql/mysql.pid也无法删除，重启docker后才可以重启，所有/目录下都变成了Read-only file system
4. 压测过程中，停止mysql的时间有点长，出现如下log：InnoDB: Waiting for page\_cleaner to finish flushing of buffer pool
5. mysql在压测过程中重启，出现了比较长时间（一个小时）的SST，出现WSREP: Created page /opt/data/mysql/data/gcache.page.000008 of size 134217728 bytes
6. mysql在运行过程中偶尔出现了一些warnning日志：[Warning] WSREP: Failed to report last committed 37377419, -4 (Interrupted system call)
7. mysql在重启过程中出现如下报警日志：WSREP: Failed to prepare for incremental state transfer: Failed to open IST listener at tcp://10.6.0.224:4568', asio error 'Cannot assign requested address': 99 (Cannot assign requested address)
8. mysql出现过无法启动的情况[ERROR] Cleanup after exit with status:32 (20150907 01:42:20.546)，究其原因是复制的数据库中，mysql库下的innodb\_table\_stats表所有权为root导致复制失败

# mycat指标评分

综合上述讨论结果，以及测试结果，我们将对mycat做如下指标做出如下评分：

|  |  |  |
| --- | --- | --- |
| **指标项** | **评分** | **备注** |
| 应用场景合理性 | 90 |  |
| 性能 | 85 |  |
| 扩展性 | 95 |  |
| 稳定性 | 90 |  |
| 成熟度 | 70 | 1.3版本存在一定的问题，1.4版本比较新 |
| 与集群结合的合理性 | 70 | 高可用不太可用 |
| 开发复杂度 | 有比较多的规则需要遵循 | 需要开发遵循一定的规则，特别1.3版本中的OR查询、批量insert、跨分片的join查询等 |
| 运维复杂度 | 不太复杂 | Mycat只是一个中间件，维护的复杂度还是在数据库 |
|  |  |  |