**API性能测试结果**

1. **测试环境**
2. 硬件环境

双核CPU 2.6GHz 2G内存

1. Jdk 1.6.0\_45 32位
2. **测试说明**
3. 测试用例分为 threadNum和loopNum，其中threadNum是线程的个数，loopNum是每个线程中循环的次数，循环的次数越多，平均时间越接近实际的真实系统响应时间
4. 因为cpu个数比较少，所以在测试是threadNum 小于等于cpu个数时是比较准确的，当大于cpu个数时，将会有线程等待执行上下文切换等开销，反而得到的平均时间会大一些
5. 测试条件是查询一张表的前10条记录，如果查询全表记录，那么肯定会比较慢，因为数据太多，避免不了磁盘IO的读写，所以测试时采用返回小批量的数据

测试url: 通过改变threadNum和loopNum来改变测试参数

[http://127.0.0.1:8080/service/query?serviceType=2&threadNum=1&loopNum=10&requestInfo= {"tableName":"user\_feature\_category","columnNames":[],"rowKeys":[],"commandId":8,"maxSize":10}](http://127.0.0.1:8080/service/query?serviceType=2&threadNum=1&loopNum=10&requestInfo=%20%7b%22tableName%22:%22user_feature_category%22,%22columnNames%22:%5b%5d,%22rowKeys%22:%5b%5d,%22commandId%22:8,%22maxSize%22:10%7d)

1. **测试结果**

threadNum = 1

|  |  |  |
| --- | --- | --- |
| loopNum | 总时间 | 平均每个请求时间 |
| 10 | 187ms | 18.7ms |
| 100 | 1375ms | 13.75ms |
| 1000 | 9359ms | 9.4ms |
| 10000 | 94266ms | 9.4ms |

threadNum = 2

|  |  |  |
| --- | --- | --- |
| loopNum | 总时间 | 平均每个请求时间 |
| 10 | 156ms | 15.6ms |
| 100 | 1063ms | 10.6ms |
| 1000 | 9875ms | 9.9ms |
| 10000 | 93875ms | 9.4ms |

threadNum = 4

|  |  |  |
| --- | --- | --- |
| loopNum | 总时间 | 平均每个请求时间 |
| 10 | 156ms | 15.6ms |
| 100 | 1000ms | 10ms |
| 1000 | 9765ms | 9.76ms |
| 10000 | 108313ms | 10.8ms |

threadNum = 8

|  |  |  |
| --- | --- | --- |
| loopNum | 总时间 | 平均每个请求时间 |
| 10 | 156ms | 15.6ms |
| 100 | 1172ms | 11.7ms |
| 1000 | 14250ms | 14.2ms |
| 10000 | 160469 | 16ms |

**四． 测试结论**

1.平均一次正常的访问服务端需要消耗9-10ms,通过日志可以看出主要的时间99.9全部耗费在和hbase相关操作上，平均的api业务逻辑代码耗时在50-60um之间(通过纳秒/1000得到微秒)

2.由于测试机器的cpu关系，可以知道当线程数超过cpu核数+1时访问反而比较慢，这是由于每个线程都需要执行一段时间，并不是短期任务，所以线程间的上下文切换消耗了不小的时间，导致最终的总时间拉长.