

数据库测试说明

单元测试

数据库作为数据存储的核心引擎，其安全性和正确性的重要性是毋庸置疑的。在开发数据库的过程中，我们使用单元测试，尽量避免出现潜在的问题。

单元测试截图如下：

```
running 9 tests
test utils::crc32::test::test_crc ... ok
test storage::entry_test::tests::test_entry ... ok
test utils::time::tests::test_time ... ok
test utils::str_util::tests::test_f64_to_string ... ok
test utils::str_util::tests::test_string_to_f64 ... ok
test ds::hash_test::tests::basics ... ok
test utils::lru_cache::test::test_lru_cache ... ok
test wondkv_test::tests::test_basics ... ok
test storage::db_file_test::tests::test_db_file ... ok
```

性能测试

测试环境

我们分别在两个系统环境下进行测试。

环境一：

- Ubuntu 20.04
- 阿里云1核2G内存

环境二：

- macOS Monterey 12.2.1
- 6核16G内存

测试方式

在这里，我们使用wrk对WondKV主数据库进行性能测试，wrk是一款针对HTTP协议的基础测试工具，它能够在单机多核CPU的条件下，使用系统自带的高性能I/O机制，如epoll、kqueue等，通过多线程和事件模式，对目标机器产生大量的负载。

进入WondKV目录，运行cargo build --release编译出可执行脚本。

在8000端口或任意可行端口运行WondKV主数据库。

wrk工具发送post请求需要编写lua脚本，我们在当前目录下编写如下脚本post.lua：

```
wrk.method = "POST"
wrk.body    = '{"key": "a"}'
wrk.headers["Content-Type"] = "application/json"
~
~
```

执行下列命令运行测试：

```
wrk -t 16 -c 100 -d 30s --latency --timeout 5s -s post.lua http://localhost:8000/key/get
```

测试结果

环境1的测试结果：

```
root@iZbp1ekrg5zbc4oubfp8znZ:~/work# wrk -t 16 -c 100 -d 30s --latency --timeout 5s -s post.lua http://localhost:8000/key/get
Running 30s test @ http://localhost:8000/key/get
16 threads and 100 connections
Thread Stats      Avg      Stdev     Max    +/-  Stdev
Latency           4.63ms   647.45us  13.95ms  96.08%
Req/Sec           1.30k    64.18    1.71k    70.54%
Latency Distribution
50%      4.67ms
75%      4.82ms
90%      5.00ms
99%      5.43ms
621695 requests in 30.02s, 125.69MB read
Non-2xx or 3xx responses: 621695
Requests/sec: 20710.09
Transfer/sec: 4.19MB
root@iZbp1ekrg5zbc4oubfp8znZ:~/work#
```

环境2的测试结果：

```
# yurunjie @ RUNJIEYU-MB0 in ~/Desktop [21:52:07]
$ wrk -t 16 -c 100 -d 30s --latency --timeout 5s -s post.lua http://localhost:8000/key/get
Running 30s test @ http://localhost:8000/key/get
16 threads and 100 connections
Thread Stats      Avg      Stdev     Max    +/-  Stdev
Latency           540.09us  144.07us   3.83ms  74.54%
Req/Sec           10.64k    528.43   13.32k   64.60%
Latency Distribution
50%      536.00us
75%      607.00us
90%      708.00us
99%         0.97ms
5100440 requests in 30.10s, 1.01GB read
Non-2xx or 3xx responses: 5100440
Requests/sec: 169450.68
Transfer/sec: 34.26MB
```

结论

从以上测试结果可以发现，我们所实现的高性能KV数据库即使在性能孱弱的云服务器中也可以实现上万的并发量，在性能更加强劲的电脑中wrk测试所达到QPS更是突破了十万。从压力测试的结果可以证明我们选用的Rust技术栈和数据库算法设计所实现的WondKV满足赛题所要求的高性能数据访问技术。