

初步测试

1 测试对象

- 静态
 - 主页
- 动态
 - 成员信息查询结果页面
 - 工作日志详细信息查询结果页面

2 测试工具

http_load

3 测试类型

压力测试

4 测试参数

- 并发数
- 响应速度
- 字节传输
- 正确率
- CPU
- 内存

5 测试结果

✓ 并发数50，持续300s

✓ 并发数100，持续300s

✓ cup、内存

✓ 微信端

6 结论

1 node.js作为web server的效果应该还是不错的；

2 从代码上，数据库和web server程序应该还有很大的优化的空间；

3 主要的瓶颈应该是服务器带宽；（可以在局域网内进行测试对比）

4 小规模的应用，应该可以满足用户需求

5 微信端的压力测试，无现成工具，需手动编程实现；
且微信端的性能，主要来自于微信服务器，可能无法改进

1701 fetches, 50 max parallel, 3.43282e+07 bytes, in 300 seconds
20181.2 mean bytes/connection
5.67 fetches/sec, 114427 bytes/sec
msecs/connect: 542.447 mean, 7410.73 max, 32.623 min
msecs/first-response: 2838.93 mean, 47756.2 max, 66.201 min
22 timeouts
394 bad byte counts
HTTP response codes:
code 200 -- 1701

1729 fetches, 100 max parallel, 3.33419e+07 bytes, in 300 seconds
19283.9 mean bytes/connection
5.76333 fetches/sec, 111140 bytes/sec
msecs/connect: 830.582 mean, 15461.9 max, 63.792 min
msecs/first-response: 2414.29 mean, 42959.3 max, 249.152 min
91 timeouts
91 bad byte counts
HTTP response codes:
code 200 -- 1726

相差不大：
CPU峰值 32%-100%
内存 80-119M

在测试期间，只测试了本人的单个微信客户端，功能正常