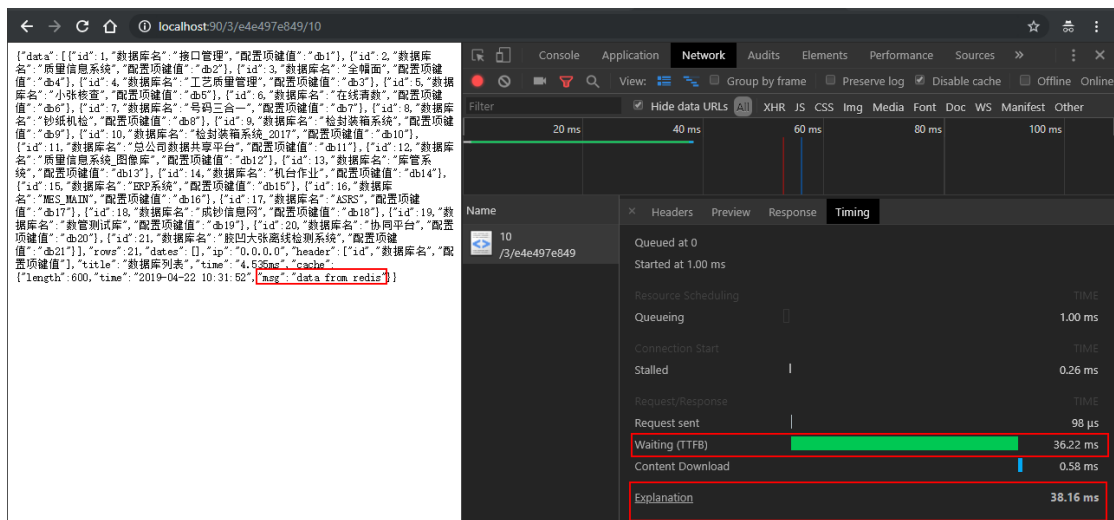


服务器并发性能测试

一、背景

近期在 PHP (7.1.12) 数据接口增加 redis 数据缓存后发现性能表现不及预期, 在数据直接从 redis 读取的情况下, 单个接口本地 chrome 隐私模式测试耗时 38ms。



二、PHP 性能测试

1. 环境及工具:

- (1) 平台: I7-7700HQ(4 核 8 线程)+16G 内存+Windows 10 64bit
- (2) 环境: Apache 2.4.9 + PHP 7.1.12 + ThinkPHP 5.0 + Redis 3.2.1
- (3) 工具: autocannon

(<https://www.npmjs.com/package/autocannon>)

ThinkPHP 关闭 session、cookies 等功能, 测试工具支持测两次自动求平均响应值。

- (4) 测试对象: 上图的数据接口

2. 测试结果

RPS: request per second, 每秒请求数, 146.

```
C:\Users\realeve>autocannon -c 100 -d 40 -o 10 http://localhost:90/3/e4e497e849/10 *2
Running 40s test @ http://localhost:90/3/e4e497e849/10
100 connections

running [=====] 97%



| Stat    | 2.5%  | 50%    | 97.5%   | 99%     | Avg       | Stdev     | Max        |
|---------|-------|--------|---------|---------|-----------|-----------|------------|
| Latency | 39 ms | 479 ms | 1307 ms | 6945 ms | 622.62 ms | 909.78 ms | 8639.26 ms |



| Stat      | 1%  | 2.5% | 50%    | 97.5%  | Avg    | Stdev  | Min     |
|-----------|-----|------|--------|--------|--------|--------|---------|
| Req/Sec   | 0   | 0    | 171    | 220    | 146.13 | 65.7   | 39      |
| Bytes/Sec | 0 B | 0 B  | 380 kB | 489 kB | 325 kB | 146 kB | 86.7 kB |



Req/Bytes counts sampled once per second.

6k requests in 40.78s, 12.7 MB read
141 errors (41 timeouts)
```

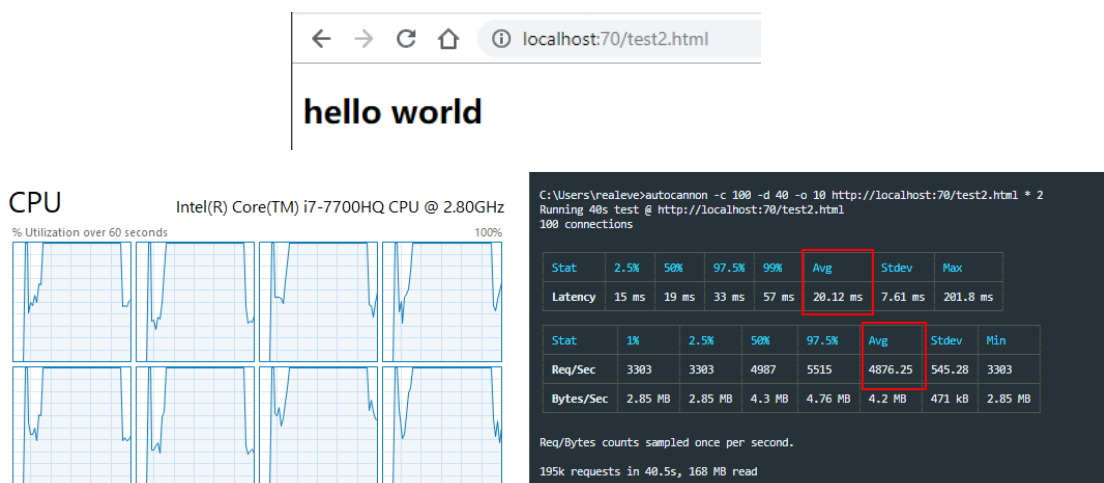
3. 分析

在该硬件配置下，146rps 性能表现不佳，结合 chrome 浏览器中测试结果，主要耗时在 TTFB（Time to First Byte）阶段，主要耗时在 PHP 框架的应用初始化、资源加载、依赖注入等。

三、PHP 基准性能测试

（1）原生性能测试

为消除 PHP 框架的影响，同时对 PHP 7.1 做了原生性能测试：



（2）结论

PHP 7.1 在该硬件环境下，rqs 在 4870 左右，该值为 PHP 应用框架下的性能上限。在测试中还能发现，高并发请求中 CPU 占用过高，硬件性能瓶颈主要表现在 CPU。

在 PHP 平台，ThinkPHP 5.0 较轻量级，但性能表现不佳，而另一流行框架 Laravel 较为重量级，性能不及 ThinkPHP，而目前 PHP 性能表现最好的 swoole 开

发体验不佳，为此我们选择了更轻量级的 NodeJS 平台。

四、NodeJS 性能测试

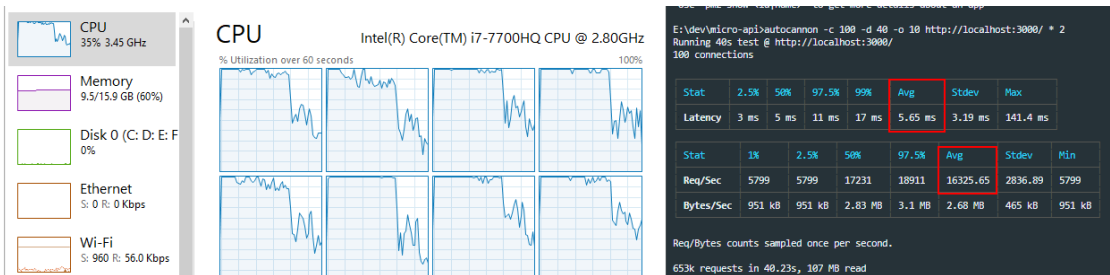
(1) 测试环境

NodeJS 10.8.0 + PM2 + Fastify 应用框架，PM2 开启 8 线程。

(2) 测试页面：

输出 hello world

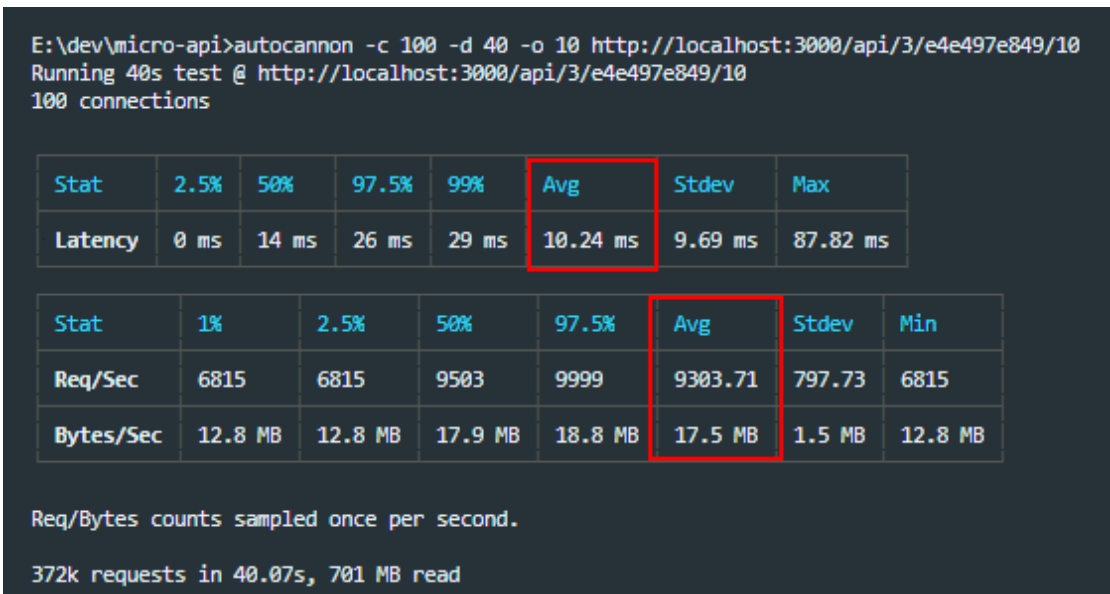
(2) 测试结果



在同样硬件配置下，NodeJS 的 RPS 在 16325，并发性能达到 PHP7 四倍。

(3) 连接数据库+redis 性能测试

针对 fastify 框架，我们同 ThinkPHP 一样编写了基于微服务 (Micro Services) 架构的 API 管理接口，同时加入 redis 做缓存管理，在相同硬件配置下，性能测试如下：



结论：在 NodeJS 的 fastify 框架中，开启 redis 后，数据不直接从数据库中读取而从缓存中读取，在能满足实际应用的要求（接口鉴权、缓存、数据库读写、路由管理等）下，性能损耗较小，同时较 PHP 的 146 并发比较，NodeJS 的并

发性能提高了 60 倍。

五、NodeJS 及 PHP 在云平台性能测试

为了探索不同平台在云服务器上的实际性能表现，我们在阿里云中针对 NodeJS 及 PHP 做了两次并发性能对比测试：

（一）第一次测试

硬件：单台 2 核 16G 内存 ECS+Redis + RDS 数据库 1 核 1G 内存

操作系统：CentOS

1. 硬件详细配置如下

注：ECS 服务器节点在 **XX**

配置信息		
规格族：通用型	数据库类型：MySQL 5.7	CPU：1 核
数据库内存：1024MB	最大IOPS：600	最大连接数：300
可维护时间段：05:00-06:00 设置	实例规格族：rds.mysql.t1.small	

配置信息	升降配	更多
CPU：2核		
内存：16 GiB		
实例类型：I/O优化		
操作系统：CentOS 7.4 64位		
弹性网卡：eni-m5e67ph7mm7wnn7fycjq		
公网IP：118.190.201.162		
弹性公网IP：-		
私有IP：172.31.85.39		
带宽计费方式：按固定带宽		
当前使用带宽：2Mbps		
专有网络：vpc-m5egh06j2rd2gpcnzfrdk		
虚拟交换机：vsw-m5ed5rw7u4bemtash5zg		

2. 测试内容

我们针对不同应用场景在服务器本地和成都远程连接外网分别做了以下并发性能测试(两次测试中取性能最优值)：

注：本地测试中，外网环境测试在两台配置相同网络不同的客户端进行,NodeJS 为 20M 宽带，PHP 为 100M 宽带，网络延迟可能有不同。

(1) 输出 hello world

测试结果	本地		外网	
	NodeJS	PHP	NodeJS	PHP

平均延迟 (ms)	4.91	12.63	96.29	50
Req/Sec	18486	7649	1032	1948

(2) 直连数据库

测试结果	本地		外网	
	NodeJS	PHP	NodeJS	PHP
平均延迟 (ms)	27.96	1249.73	133.94	1207.41
Req/Sec	3514	81.02	743	78.62

(3) 读取 Redis 数据库缓存，不直连数据库

测试结果	本地		外网	
	NodeJS	PHP	NodeJS	PHP
平均延迟 (ms)	10.39	1195.1	126.45	1181.28
Req/Sec	9226	84.41	784	82.27

(4) 写数据库

测试结果	本地		外网	
	NodeJS	PHP	NodeJS	PHP
平均延迟 (ms)	58.58	1246	114.4	1246.06
Req/Sec	1690	82.47	869	82.47

原因分析及说明：(略)

1. 内外网的差异
2. 读写的性能差异
3. 数据库配置低对测试的影响

(二) 第二次测试

硬件：单台 8 核 16G 内存 ECS+Redis + Polar RDB 数据库 4 核 16G 内存

操作系统：CentOS

1. 硬件详细配置如下

华东 1 (杭州)

随机分配

可用区 H

可用区 G

可用区 F

可用区 B

可用区 E

可用区 I

不同地域的实例之间内网互不相通，选择靠近您客户的地域，可降低网络时延，提高您客户的访问速度。

I/O 优化实例

vCPU: 8 vCPU

内存: 16 GiB

实例规格: 知: ecs.sn1ne.large

IPv6: 请选择

仅显示最新一代 所有代 8 vCPU 16 GiB

规格族	实例规格	vCPU	内存	处理器型号	处理器主频
<input type="radio"/> 计算型 c5	ecs.c5.2xlarge	8 vCPU	16 GiB	Intel Xeon(Skylake) Platinum 8163	2.5 GHz
<input checked="" type="radio"/> 计算网络增强型 sn1ne	ecs.sn1ne.2xlarge	8 vCPU	16 GiB	Intel Xeon E5-2682v4 / Intel Xeon(Skylake) Platinum 8163	2.5 GHz

节点信息

增加节点

减少节点

节点名称	可用区	状态	当前角色	规格	最大连接数	最大IOPS
pi-bp1j2aj123pfhw73	杭州 可用区G	运行中	主节点	4核 16GB	5000	32000
pi-bp13q75a49fg32qbn	杭州 可用区G	运行中	只读节点	4核 16GB	5000	32000

(1) 输出 hello world

测试结果	本地		外网	
	NodeJS	PHP	NodeJS	PHP
平均延迟 (ms)	4.91	12.63	96.29	50
Req/Sec	18486	7649	1032	1948
平均延迟 (ms)	1.8		43.75	
Req/Sec	45215		2261	

(2) 直连数据库

测试结果	本地		外网	
	NodeJS	PHP	NodeJS	PHP
平均延迟 (ms)	27.96	1249.73	133.94	1207.41
Req/Sec	3514	81.02	743	78.62
平均延迟 (ms)	5.28		164.7	
Req/Sec	17318		602	

(3) 读取 Redis 数据库缓存，不直连数据库

测试结果	本地		外网	
	NodeJS	PHP	NodeJS	PHP

平均延迟（ms）	10.39	1195.1	126.45	1181.28
Req/Sec	9226	84.41	784	82.27
平均延迟（ms）	2.39		191.96	
Req/Sec	34475		516.72	

（4）写数据库

测试结果	本地		外网	
	NodeJS	PHP	NodeJS	PHP
平均延迟（ms）	58.58	1246	114.4	1246.06
Req/Sec	1690	82.47	869	82.47
平均延迟（ms）	5.14		176.68	
Req/Sec	17755		561.34	
1 分钟写入量	1065K		34K	

原因分析及说明：（略）