企点微服务网关演进之路

郑榕

郑榕(roman),腾讯后台高级工程师, 腾讯企点Web后台基础技术建设负责人。

十年C/C++开发,曾就职于大唐电信, 科达科技等通信领域公司,涉及音视频实时 通信,音视频存储,流媒体点播等方面。

两年PHP开发,喜欢折腾Nginx,PHP扩展,Zend内核。





初代架构

公司接入层 接入层 业务接入层 展现层 逻辑层 监控 业务逻辑层 日志 数据访问层 数据层 Cache DB

- 手动配置路由
- 集群难以管理

• 按大业务划分

• 手动注册服务

• 自我监控

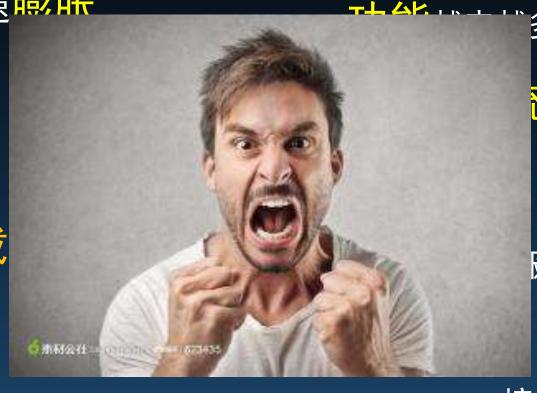
单体服务越来越大

代码量迅速膨胀

轻重不分离

服务负载

一个服务多人编写



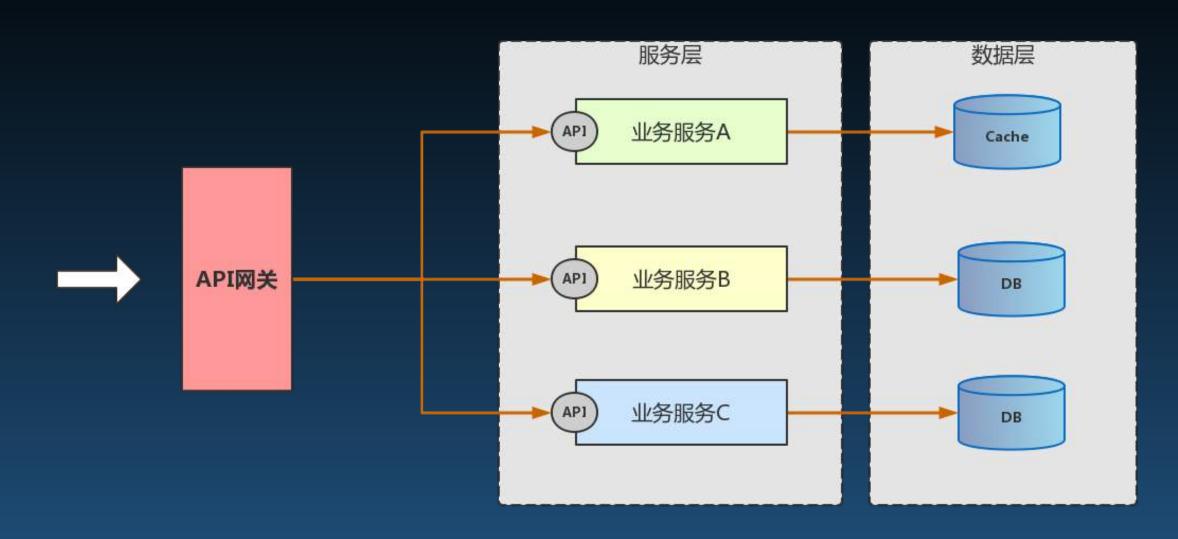
量越来越大

B务又异常了

接口性能时好时坏

接口A扩容,接口B: 关表陪婚考虑微服务了

微服务主流架构



网关技术选型

主流方案





考量因素

1 调用链增加

2 技术栈 统一

3 特殊业务场景

4 对接公司服务

网关技术选型

- 最美好的就在身边
- 高性能服务器的典范
- 资源占用少
- 稳定性极佳
- 灵活的扩展性



Why Not?

NGINX

API网关

聊聊 ProxyPass

```
Location /test{
  proxy set header Host testhost.com;
  proxy set header X-Forwarded-For $remote addr;
  proxy set header X-Real-IP $remote addr;
  proxy_pass http://testhost.com/$request_uri;
upstream testhost.com {
  ip hash;
  server XX. XX. XX. XX: XXXXX;
  server XX. XX. XX. XX: XXXXX;
```

聊聊 ProxyPass

赖以成名的反向代理

优秀的性能

良好的协议兼容

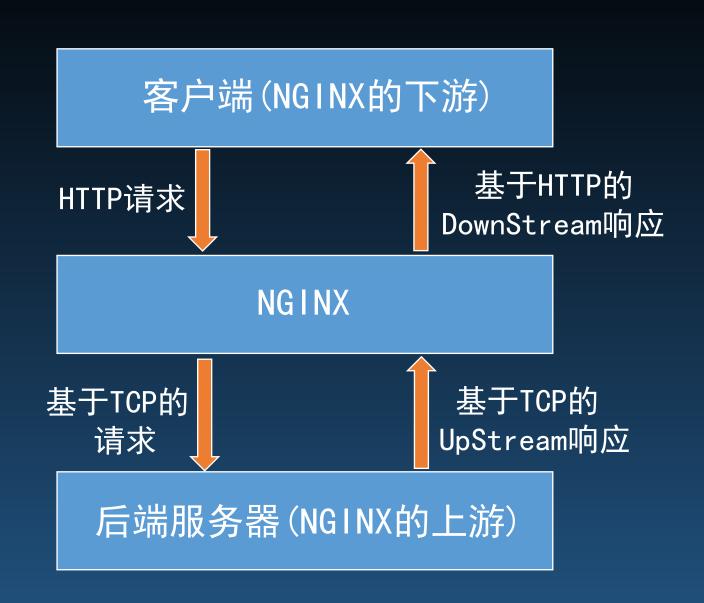
极佳的稳定性

NGINX的精华

但是,我们要的是动态路由……

- 重写反向代理吗?
- 如果要,怎么做?
- 使用upstream机制
- •什么是upstream?

其实······ 原生反向代理模块用 的也是upstream机制



原生Proxy_module代码行数 5000+ 我们只要动态路由, 原生Proxy_module提供命令数 62个 真的要放开原生Proxy_module吗······ 一个灵活可靠的反向代理不简单

- 先看个例子
- 好像有办法了……

```
pkoxaytipmasstesttp://10.1.1.1/$request_uri;
    proxy_set_header Host testhost.com;
    proxy_set_header X-Forwarded-For $remote_addr;
proxy_pass http://$\frac{1}{5}\text{qurouter}^{\text{semote}}\text{semote}^{\text{semote}}\text{con;}
proxy_pass http://10.1.1.1/$request_uri;
}
```

• HTTP十一个处理阶段

HTTP_POST_READ_PHASE HTTP_SERVER_REWRITE_PHASE HTTP_FIND_CONFIG_PHASE HTTP_REWRITE_PHASE HTTP_POST_REWRITE_PHASE

HTTP_PREACCESS_PHASE HTTP_ACCESS_PHASE HTTP_POST_ACCESS_PHASE HTTP_TRY_FILES_PHASE HTTP_CONTENT_PHASE HTTP_LOG_PHASE

REWRITE_PHASE

HTTP_ROUTER_MODULE

(自定义模块)

CONTENT_PHASE

HTTP_PROXY_MODULE

(原生模块)

```
ngx_http_add_variable
( qdrouter )
```

```
Location /{
proxy_pass
http://$qdrouter/$request_uri;
}
```

- 优雅重用
- 事半功倍

• 终于可以尽情控制路由行为了

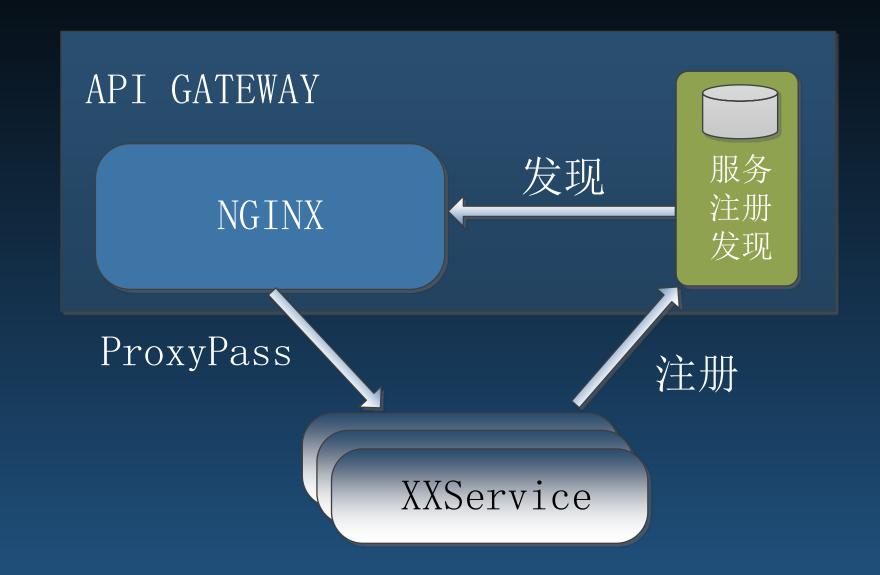


坐稳,出发了!



服务注册发现

- SNAME->IP:PORT
- SNAME->LBID
- LBID->IP:PORT

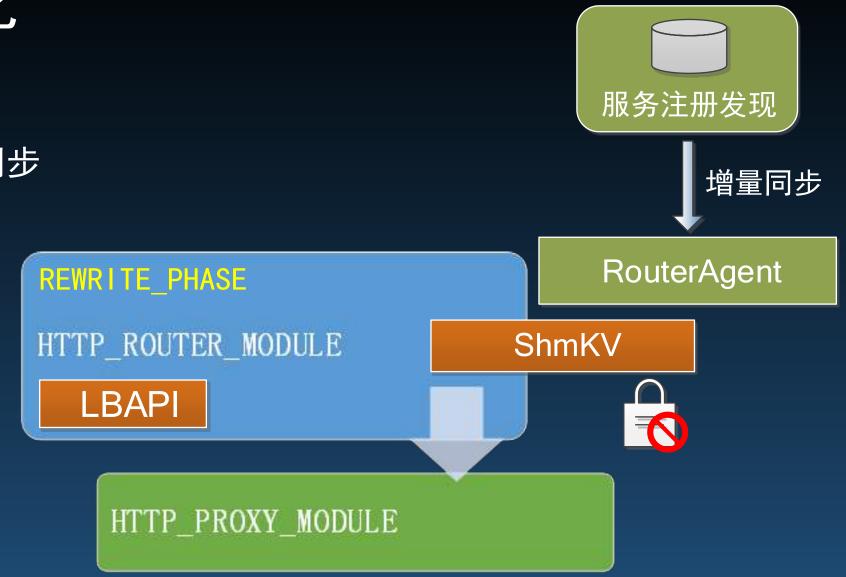


不只是寻址

• 负载均衡算法 • 门限收缩算法 对接公司级路由决策系统 • 宕机探测算法 LBRouteGet NGINX LBAGENT LBAPI ProxyPass **LBCENTER** XXService 横向伸缩热更新

性能优化

- 路由表缓存
- 本地Agent同步

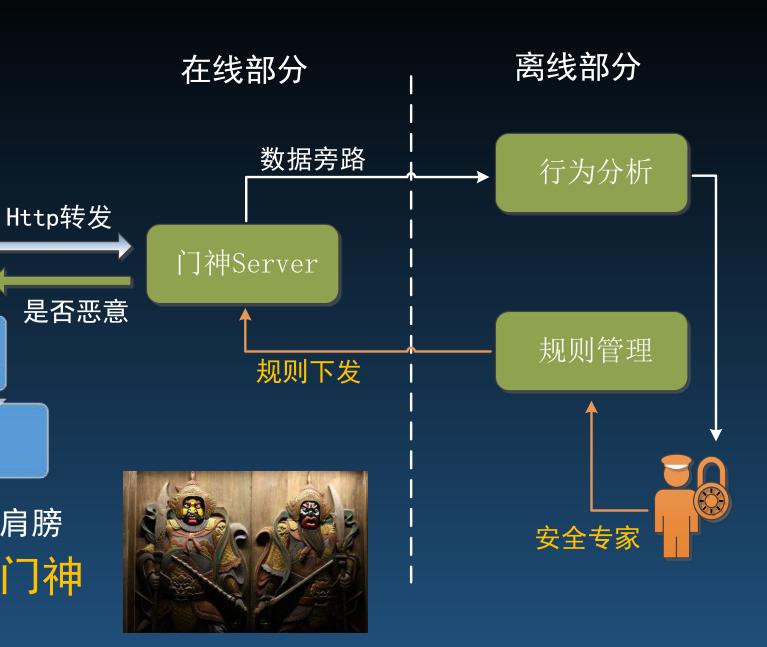


安全控制

HTTP MENSHEN MODULE HTTP QDAUTH MODULE HTTP ROUTER MODULE

- 处理阶段前置
- 降级容错

- 站在巨人的肩膀
- 公司级服务门神



安全控制

• 频控维度 频控后台服务 RID (UIN) IP 鉴权中心 HTTP MENSHEN MODULE 访问上报 拦截拉取 本地 Agent HTTP QDAUTH MODULE 本地统计 • 统一鉴权 HTTP ROUTER MODULE 调用上报队列 共享 内存 拦截记录存储

服务监控

上帝 视角



- 服务调用链
- 实时告警

• 服务流量

• 服务质量

• 大盘数据



服务监控

- 日志阶段, 天生的监控基因
- 对接公司平台

HTTP_ROUTER_MODULE

HTTP_PROXY_MODULE

LOG_PHASE

HTTP_MONITOR_MODULE

数据旁路





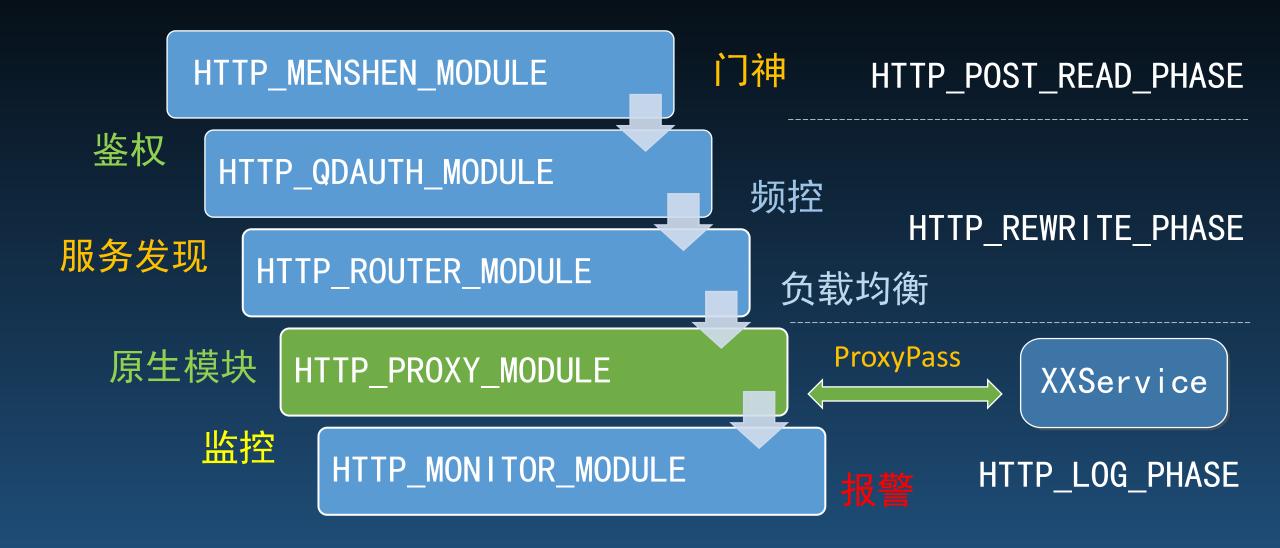








整体回顾



未来规划





腾讯Web后台开发

JOIN US!

romanzheng@tencent.com