



BEIJING 2017

Golang微服务

在腾讯游戏用户运营领域的探索及实践



刘家雄



促进软件开发领域知识与创新的传播



关注InfoQ官方信息
及时获取QCon软件开发者
大会演讲视频信息



扫码，获取限时优惠



全球架构师峰会 2017 [深圳站]

2017年7月7-8日 深圳·华侨城洲际酒店

咨询热线: 010-89880682



全球软件开发大会 [上海站]

2017年10月19-21日

咨询热线: 010-64738142

Web Architecture

Platform



关于我

刘家雄 <@楚吟风>



提纲

腾讯游戏用户运营介绍

服务化架构演进

DSL加速敏捷

基准数据

总结及展望

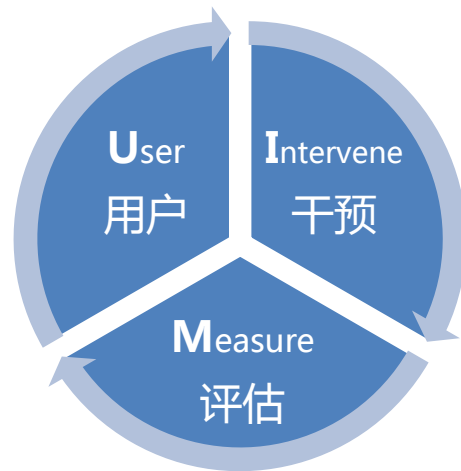
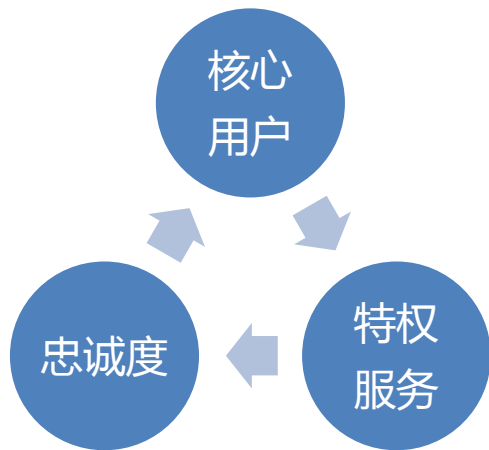
腾讯游戏用户运营

关于腾讯游戏用户运营

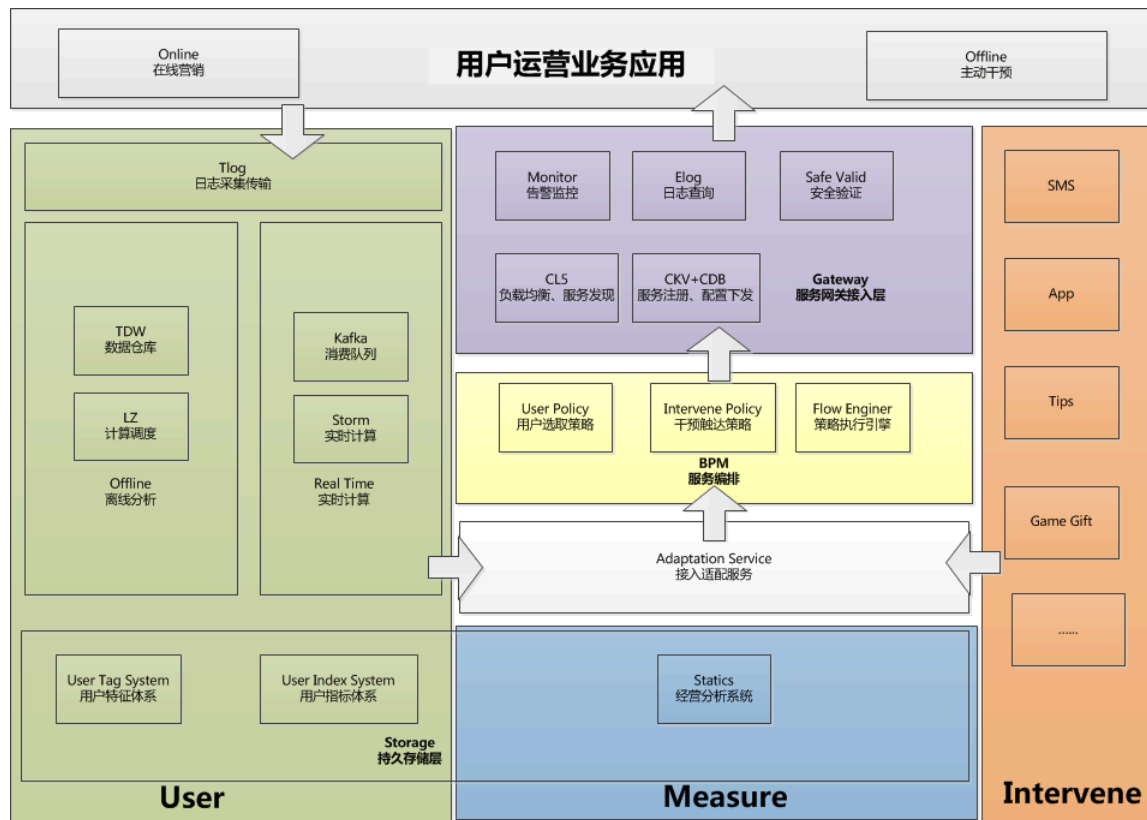


腾讯游戏 Tencent Joy Club
心悦俱乐部

心悦·诚服



技术体系



服务化架构演进

演进历程

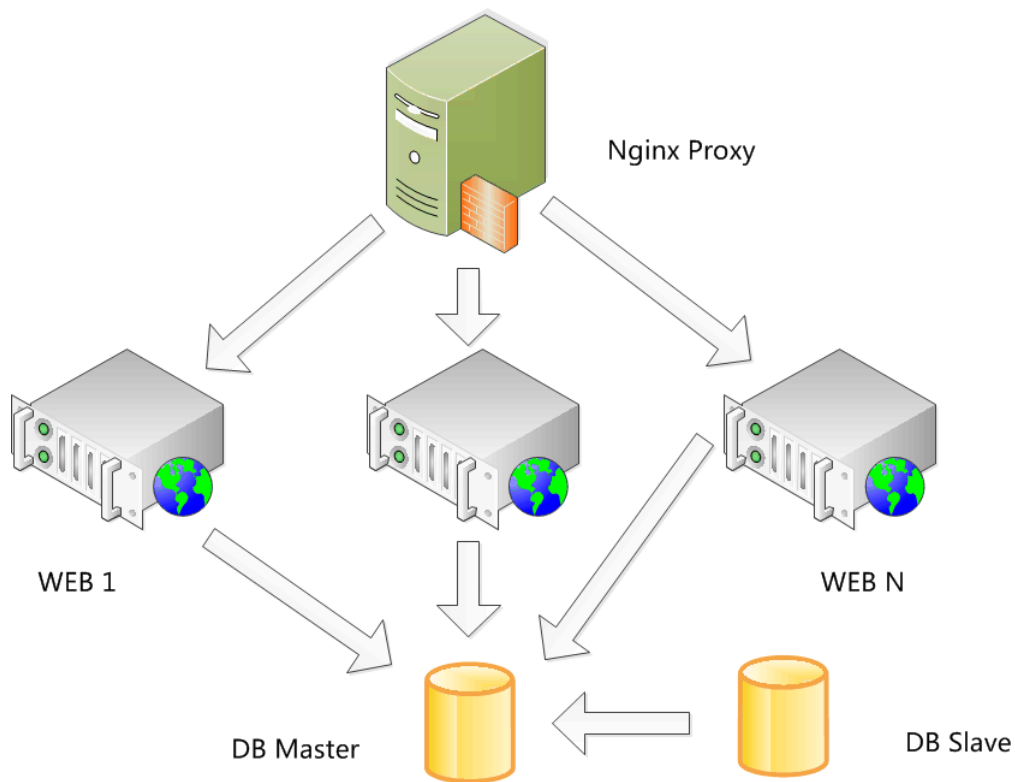


经典LNMPA架构

先抗住再优化

业务混合部署

面向需求实现



效率质量之痛

流程

敏捷

隔离

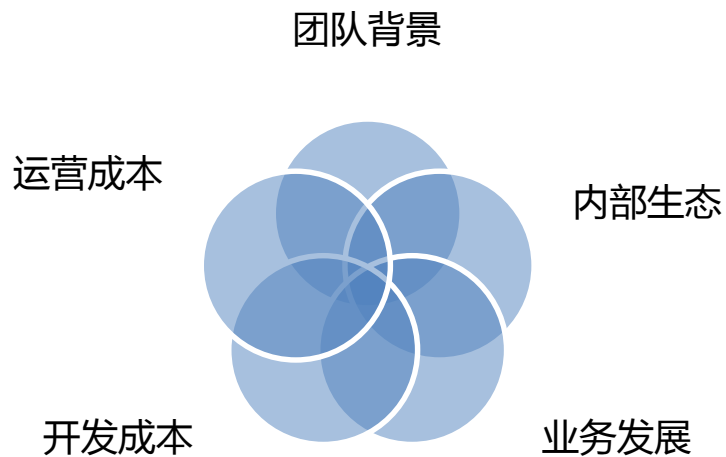
性能

安全

监控



中间件技术选型



数据中间件——指标服务

Golang

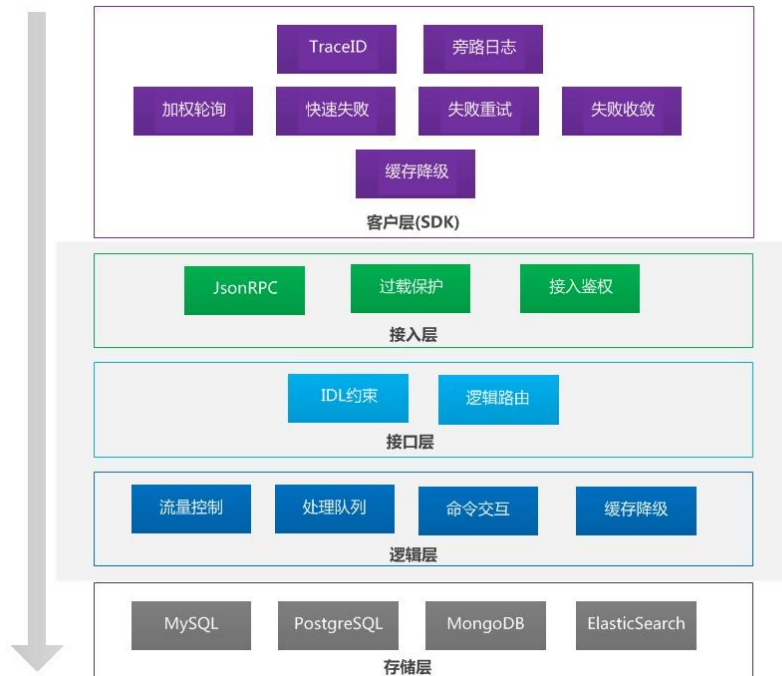
高性能

高可用

扩展性

安全性

可运营性



业务发展驱动技术演进

服务网关

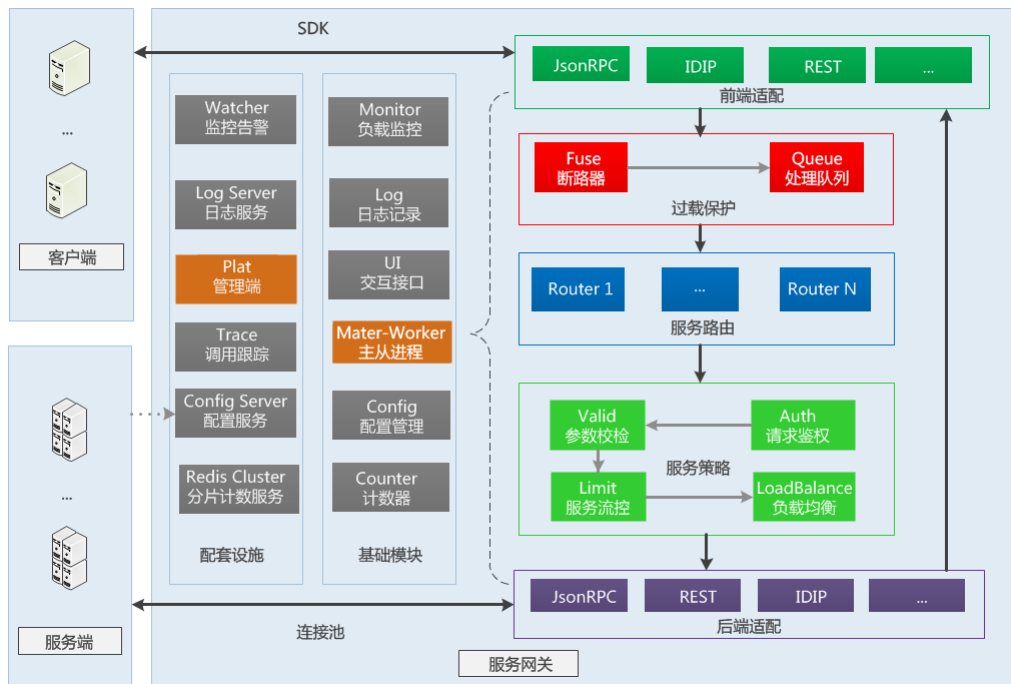
过载保护

流控降级

SET部署

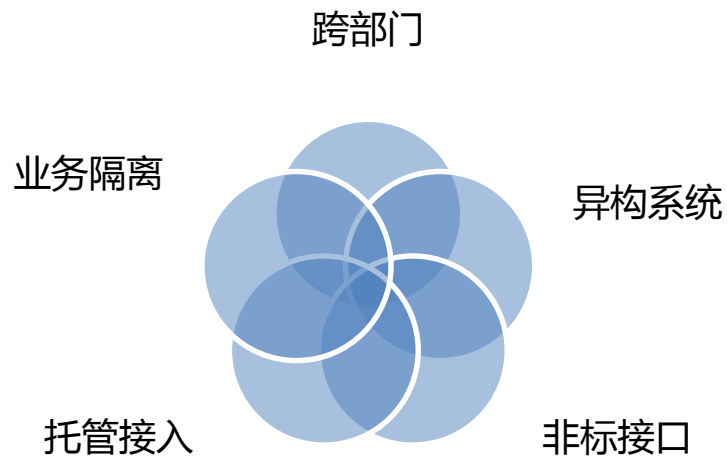
立体监控

敏捷集成

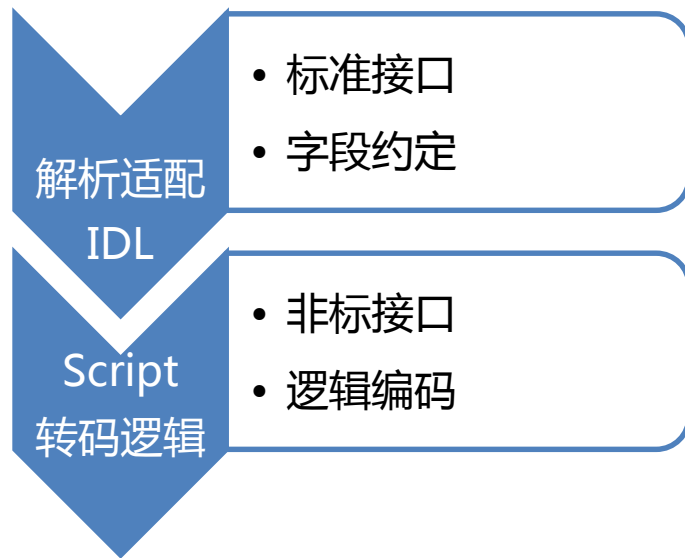


微服务探索

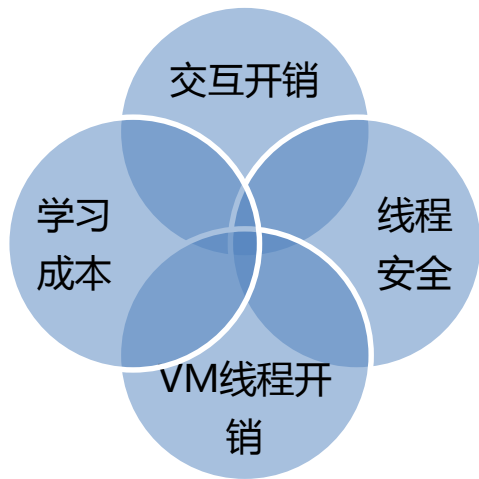
问题



手段



嵌入脚本选型



Custom	• Expression、 Learning cost
Javascript(V8)	• CGO overhead
Javascript (otto)	• Performance
Python/PHP	• CGO / Thread Safe
cLua/cLuaJIT	• CGO / Thread overhead
GopherLua	• Performance / API friendly

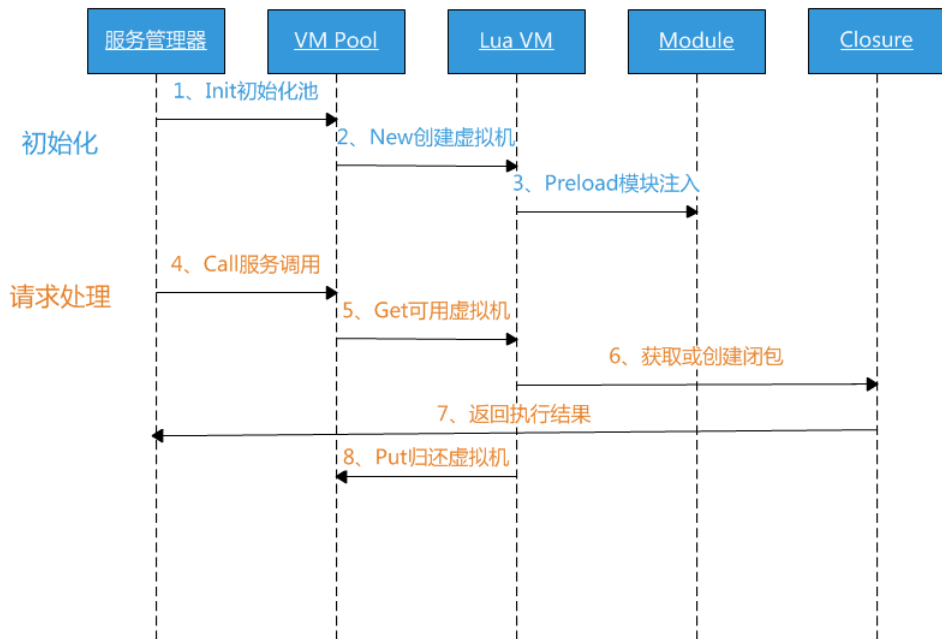
嵌入式Lua虚拟机

Gopherlua

VM Pool

Closure 隔离

即写即用



One VM One Service, Write Once Run Anywhere

服务IDL

key	<input type="text" value="sum"/>
服务唯一标识	
服务标题	<input type="text" value="求和-可用性测试"/>
服务标题	
服务描述	<input type="text" value="计算两数之和"/>
服务描述	
服务入参	<input type="text" value='{"a":{"title":"数字1","type":"int"},"b":{"title":"数字2","type":"int"}}'/> <input type="button" value="配置"/>
服务入参	
服务返回结果	<input type="text" value='{"sum":{"title":"求和结果","type":"int"}}'/> <input type="button" value="配置"/>
服务返回结果	

处理规则

处理器ID	<input type="text" value="lua"/>
处理器ID	
处理器规则	<pre>local a = params.a local b = param.b local tlib = require "tlib" local num, err = tlib.sum(a, b) if err ~= nil then return nil, {code = 500, msg = err, data = params} end return {sum = num}</pre>
处理器规则	

服务等级协议SLA

服务测试

a:
数字1

b:
数字2

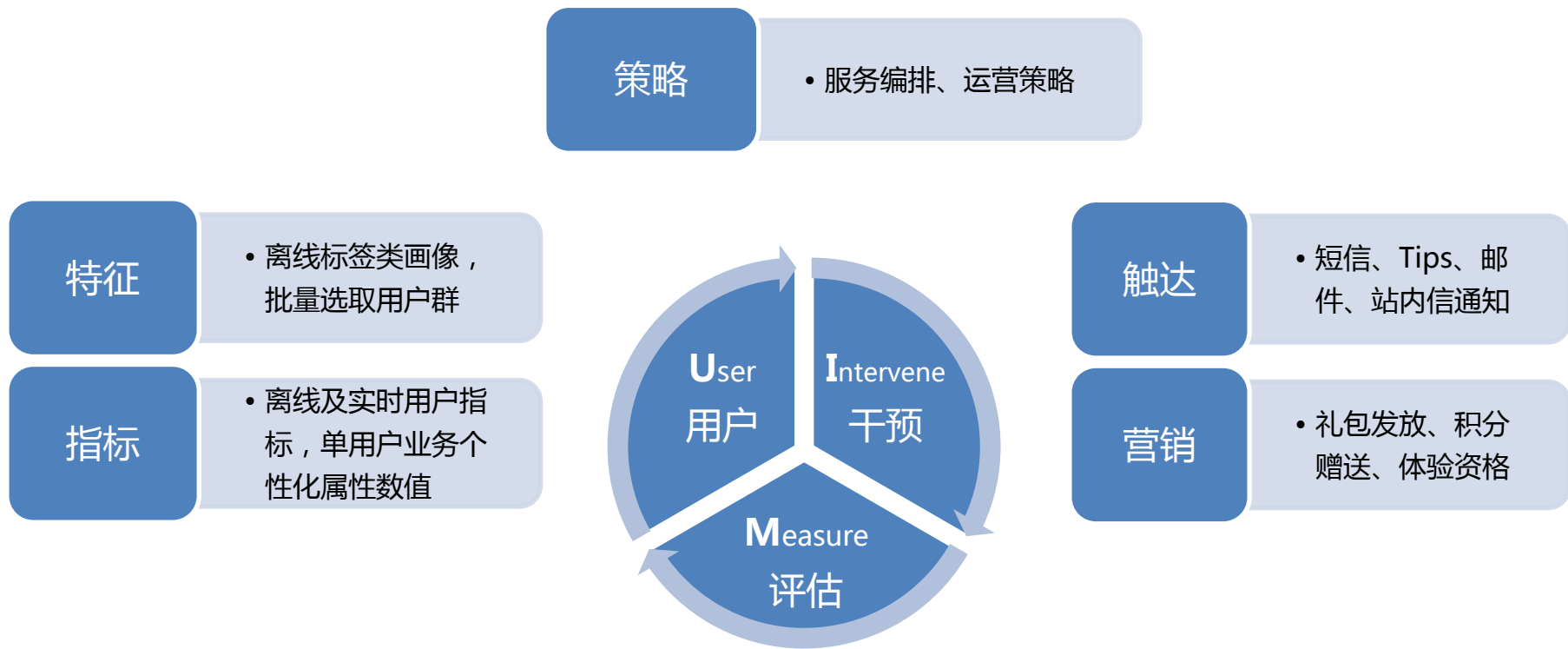
测试

输出结果

```
1 1
2 2
3 3
4 4
5 5
6 6
7 7
8 8
9 9
10 10
11 11
12 12
```

```
{
  "ext": {
    "client": "36287",
    "request_time": "2017-03-27 16:33:44.792647298",
    "response_time": "2017-03-27 16:33:44.792781966",
    "server": "10008"
  },
  "id": "20170327163344-58d8ce68b8644",
  "result": {
    "sum": 3
  }
}
```

微服务划分



微服务实践

服务注册发现

- CDB + CKV / ETCD

负载均衡

- CL5 / LVS CAE自动伸缩容

鉴权

- ID / Token / IP

流控

- Atomic + Inmem + Redis、令牌桶

降级

- 轻重分离、单元化部署、容错

日志监控告警

- 实时上报、缓存汇聚/本地文件、ELK

运行时类库

- Bind Golang to Lua

并发模型

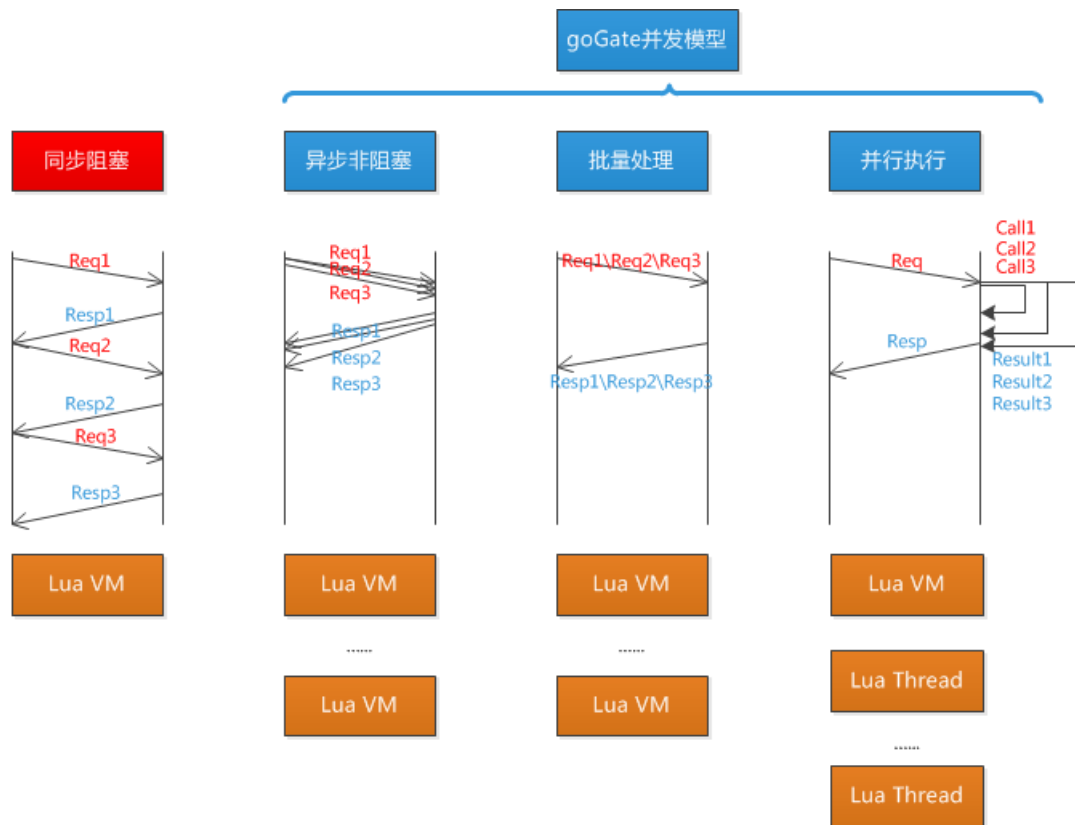
异步Async

批量Batch

多核并行Parallel

Lua协程绑定Go程

IO阻塞自动切换



高可用

负载均衡

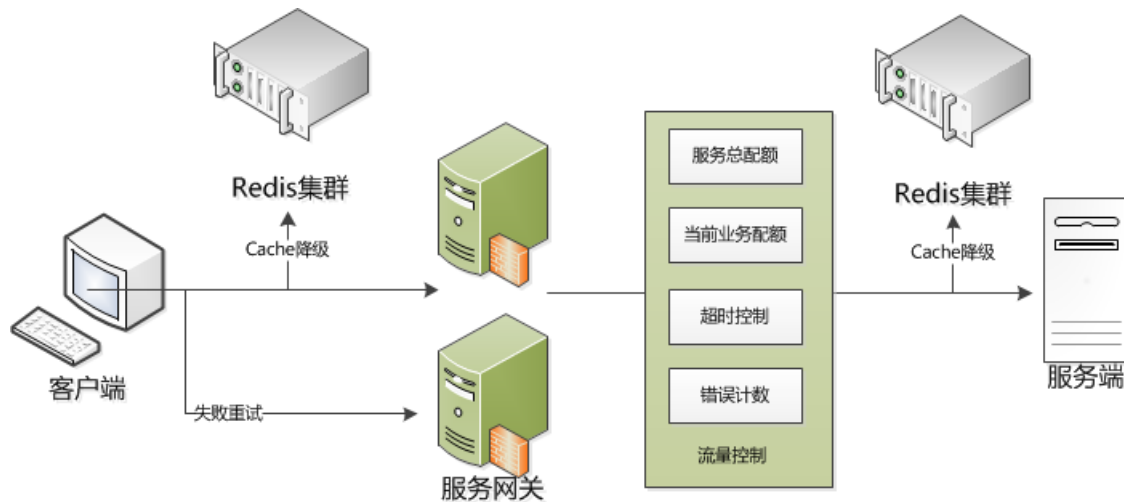
寻址

限流

缓存

降级

SLA保证



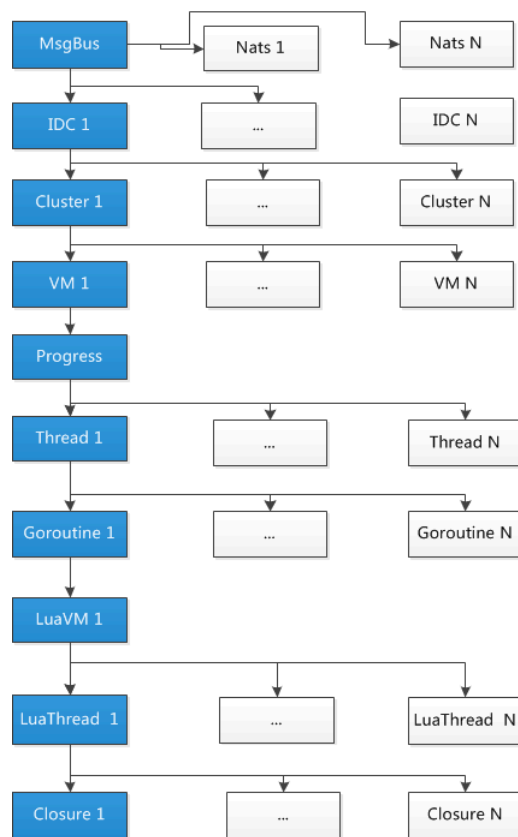
并行执行单元

消息总线

屏蔽本地网络差异

微执行单元

水平伸缩



运营监控

旁路实时上报

自定义告警策略

收敛算法

海量日志查询



DSL加速敏捷

微服务编排



```
1 local flow = function(user_id, gift_id)
2     local ret_1 = rpc_call("get_user_level", user_id)
3     if ret_1.level > 1 then
4         local ret_2 = rpc_call("send_gift", user_id, gift_id)
5         return ret_2.status
6     end
7 end
```

DSL设计

```
1 flow:set_params({                                     --设置流程入口参数
2     user_id = {},
3     gift_id = {},
4 }):set_returns({                                     --设置流程返回值
5     status = "{{status}}"
6 }):set_node("node_1", {                             --设置节点node_1
7     args = {user_id = "{{user_id}}"},               --节点入参
8     params = {user_id = {}},                        --节点形参
9     returns = {level = {}},                         --节点返回值
10    bind = {level = "level"},                       --绑定到流程上下文的值
11    caller = "get_user_level",                      --服务名
12 }):set_node("node_2", {
13     args = {user_id = "{{user_id}}", gift_id = "{{user_id}}"},
14     params = {user_id = {}, gift_id = {}},
15     returns = {status = {}},
16     bind = {status = "status"},
17     caller = "send_gift",
18 }):set_entry(                                       --设置流程的入口节点
19     "node_1"
20 ):set_graph(                                       --设置节点之间的关系图
21     "graph_1", {                                   --从节点node_1指向node_2, 如果上下文的level字段大于1
22         "node_1", "node2", {"gt", "{{level}}", 1},
23     }
24 )
```

应用落地

开发外包

- 正职开发核心服务，流程编排外包

运营自助

- 运营填写表单，活动一键上线

流程复用

- 策略嵌套、一键复制

自动测试

- 自动Mock、分支覆盖

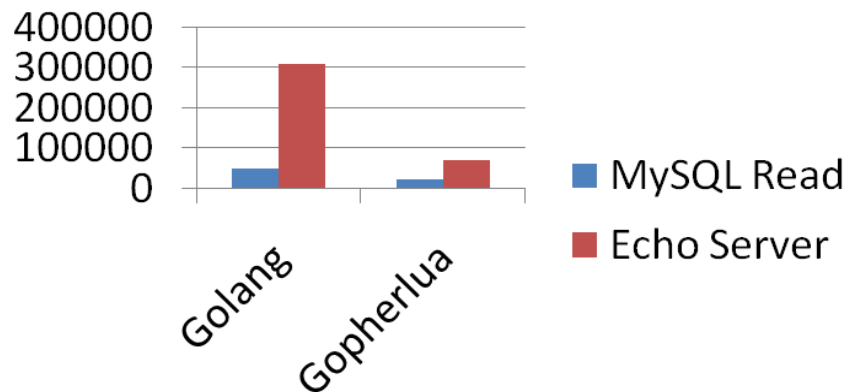
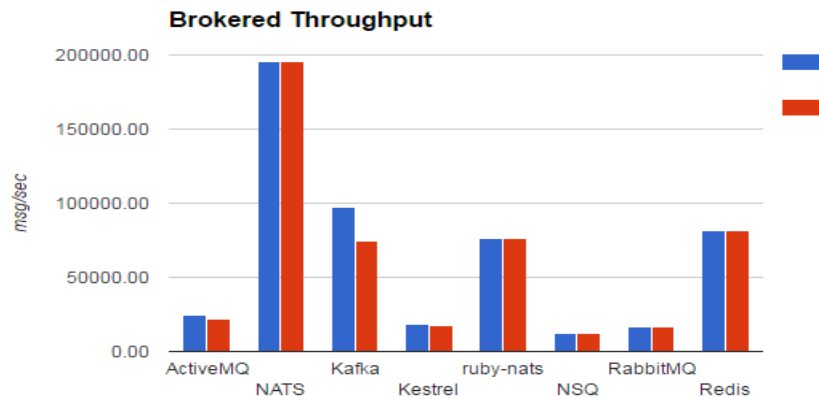
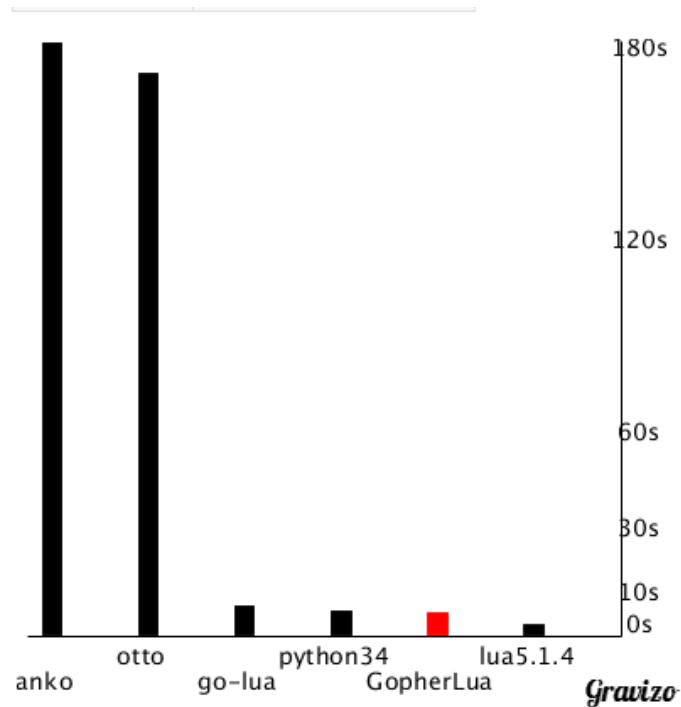
模型沉淀

- 流程固化，沉淀运营模型



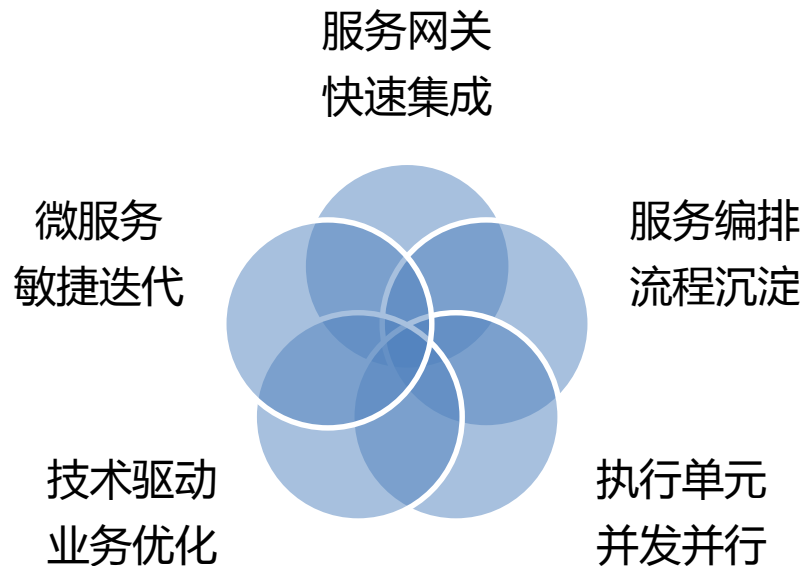
基准数据

基准数据



总结及展望

总结及展望





关注QCon微信公众号,
获得更多干货!

Thanks!



INTERNATIONAL SOFTWARE DEVELOPMENT CONFERENCE

主办方 **Geekbang** > **InfoQ**
极客邦科技