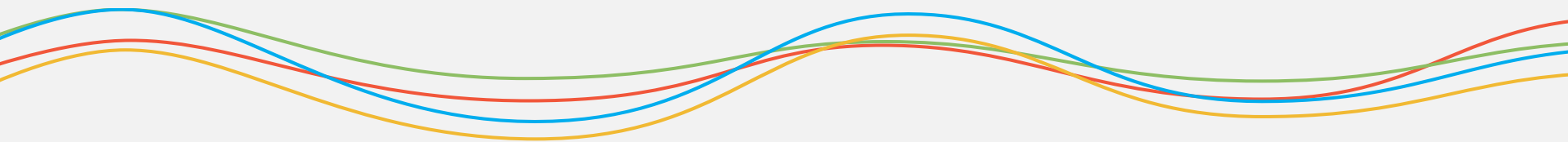


# 58到家 服务治理实践

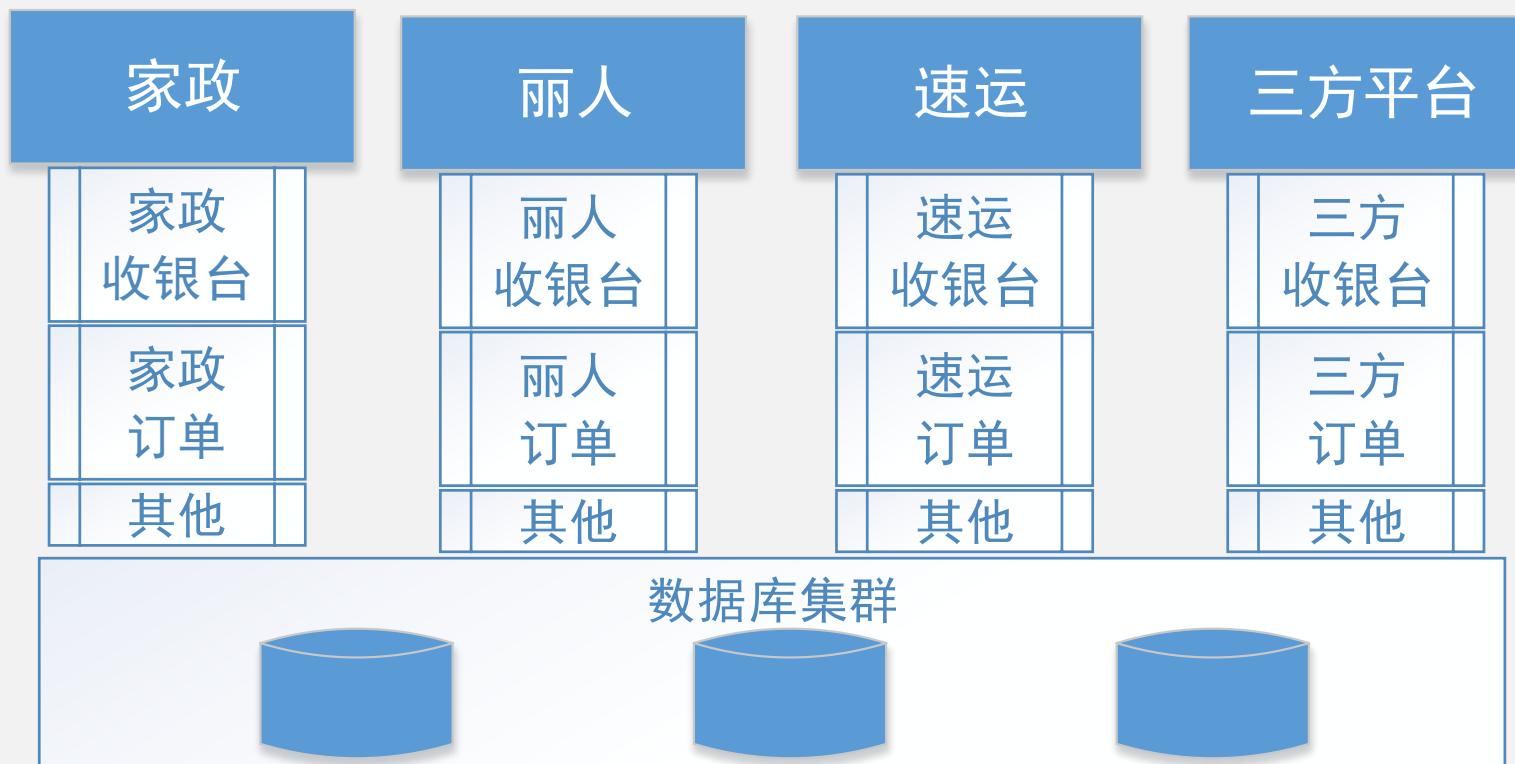


一、为什么需要服务治理

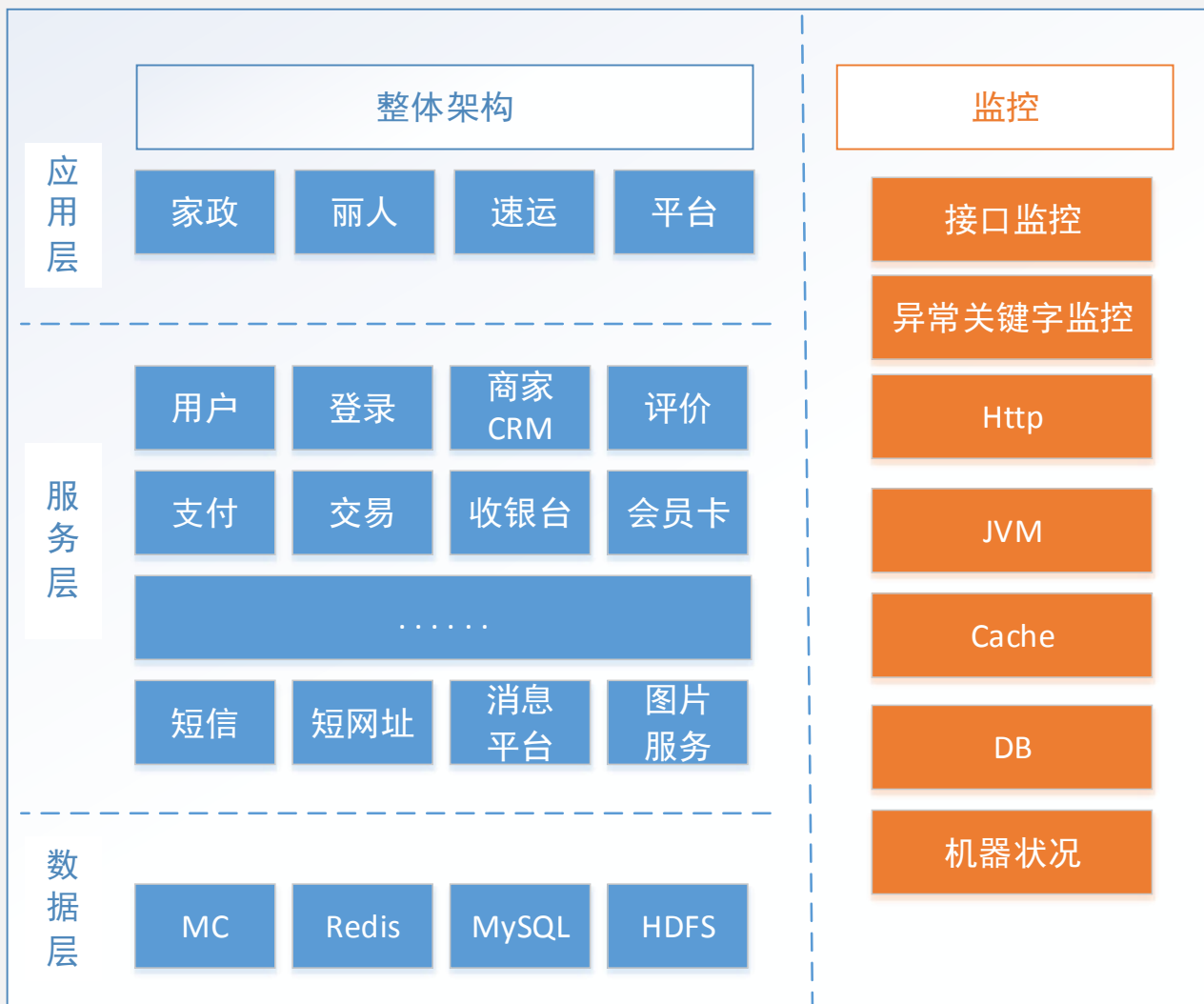
二、58到家服务治理实践

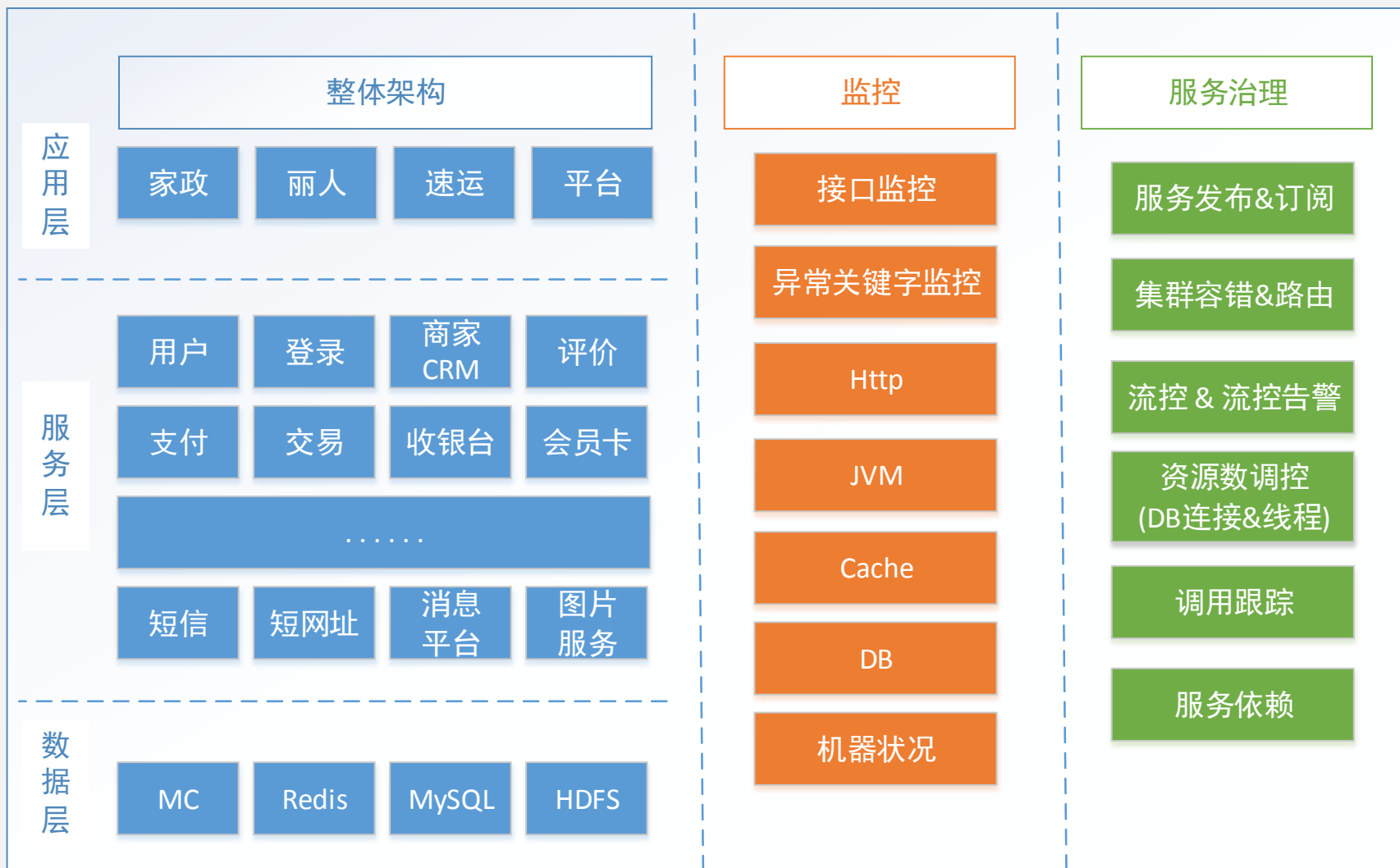
三、小结

# 为什么需要服务治理

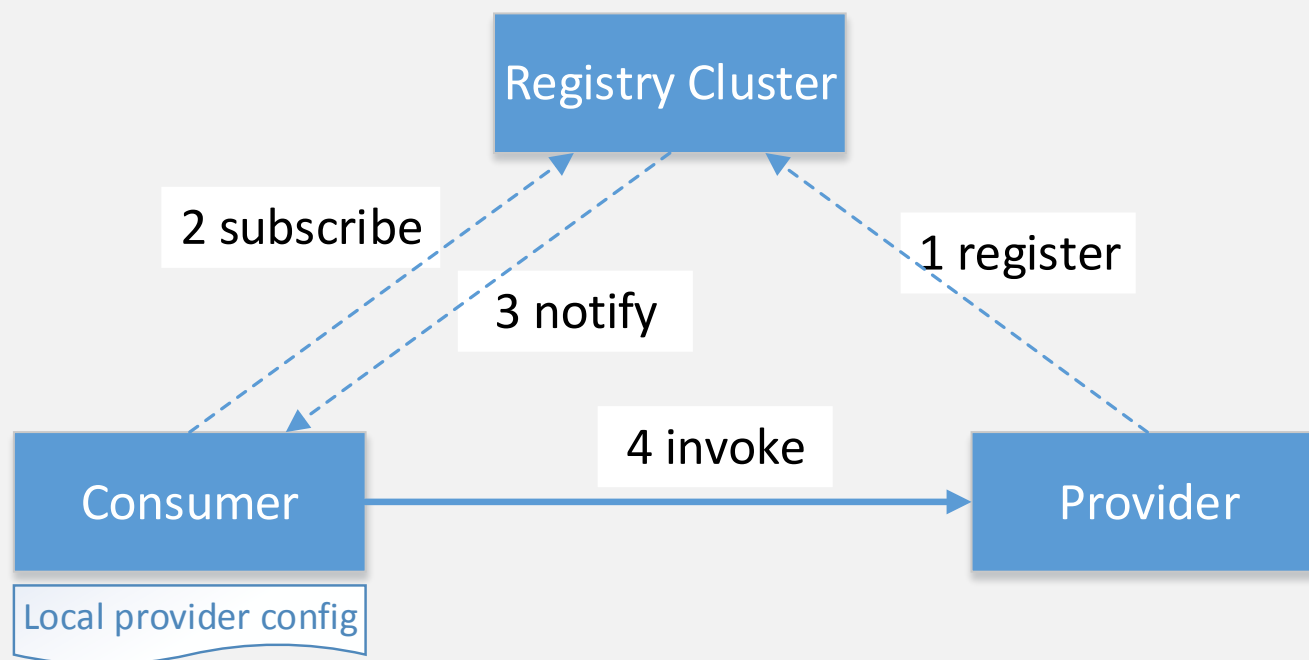


代码重复率高，可维护性差，开发、测试、部署成本高  
系统可伸缩性差、可靠性差  
需求变更困难，无法敏捷交付





# 到家 服务治理实践



- 发布 & 订阅 机制  
服务增、删节点，对服务消费方透明；  
方便做分组管理和安全策略；
- 服务健康状况监测  
TCP长连接、心跳监测；
- 高可用  
注册中心HA；  
Consumer本地配置



- 路由
  - 常见路由方式：随机、轮询、基于权重、基于负载；
  - 路由策略配置：IP地址段、方法名匹配、服务分组；
  - 路由策略扩展：提供路由策略接口，满足个性化路由；
- 容错
  - 故障转移(幂等服务)
  - 失败缓存
  - 失败通知
  - 快速失败

**服务无状态、重试、超时时延**

## ➤ 流量控制

保命

流控阈值80%使用率提前告警；

自动扩容或手动快速扩容；

流控阈值在线调整，实时生效；

## ➤ 流量告警

阈值告警；

波动告警；

服务：[REDACTED]

服务详情

服务节点列表

服务分组列表

调用方列表

报警列表

函数列表

全部服务节点

全部服务函数

[REDACTED]



查询

服务访问量(每分钟)

总量：[REDACTED] 最大值：[REDACTED] 最小值：[REDACTED]

Reset zoom

IP分布 :03:26

:98.5%

:0.1%

:0.1%

:0.1%

:0.1%

其他：1.1%

03:00

0

4:30

05:00

05:30

06:00

06:30

07:00

07:30

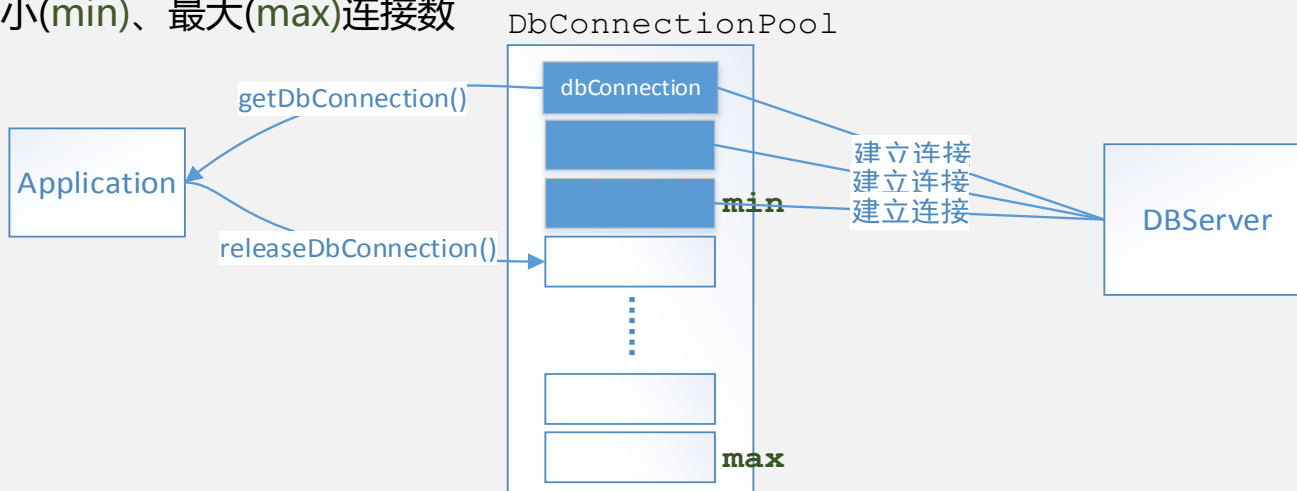
08:00

08:30

09:00

服务调用量：[REDACTED]

## ➤ 最小(min)、最大(max)连接数



## ➤ 最大并发连接数(Max\_ConcurrentConnection\_Count)

```
public DbConnection getDbConnection(){
    if(concurrentConnectionCount.get() > Max_ConcurrentConnection_Count){
        throw new SQLException("并发同时创建的链接数据超过"+ Max_ConcurrentConnection_Count);
    }
    concurrentConnectionCount.incrementAndGet();
    ConnectionWrapper connection = null;
    try {
        //.....
        connection = createConnection();
        //.....
    }catch (SQLException e) {
        throw e;
    } finally {
        concurrentConnectionCount.decrementAndGet();
    }
    return connection ;
}
```

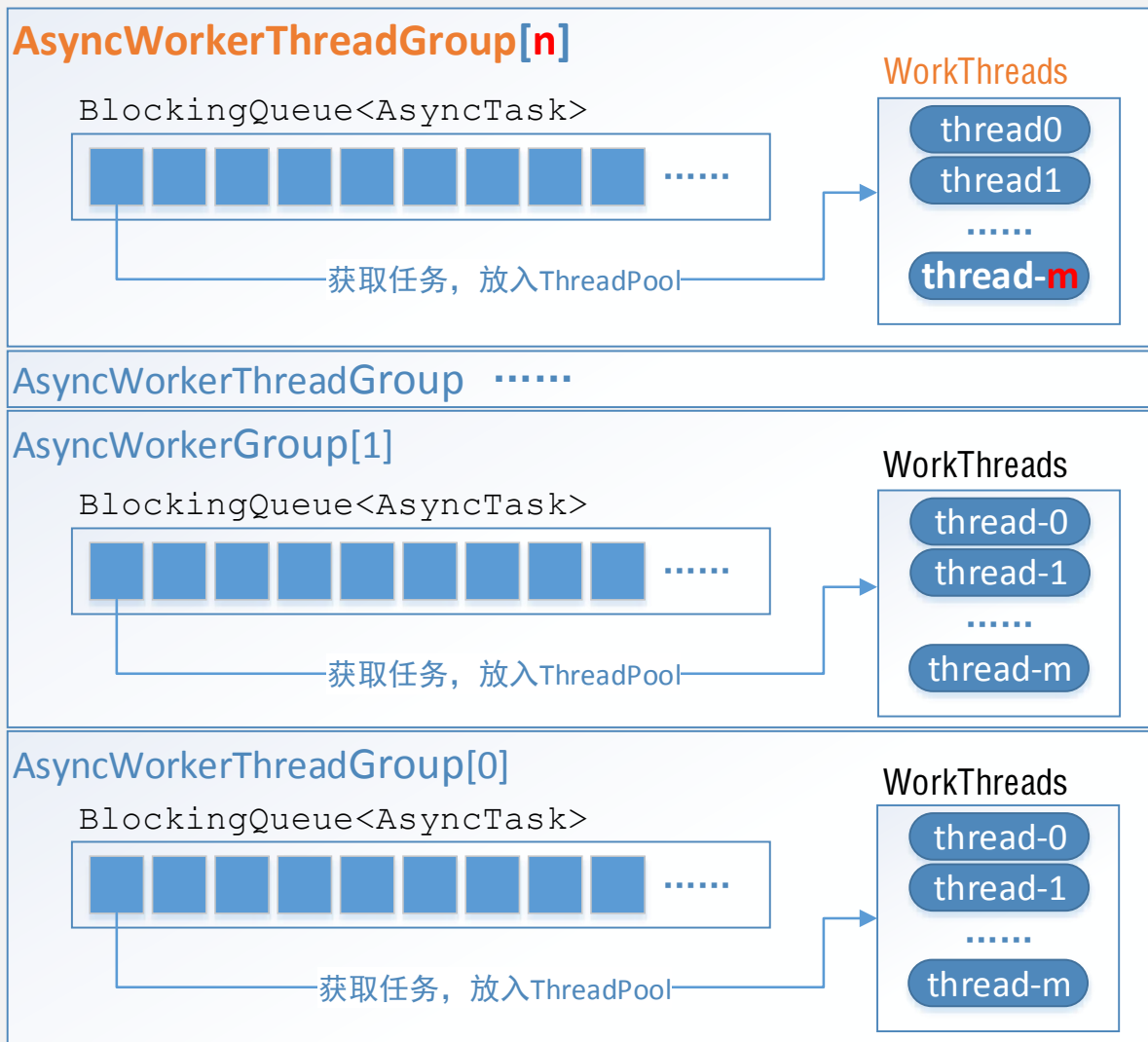
## ➤ 线程模型

工作线程组 n ;

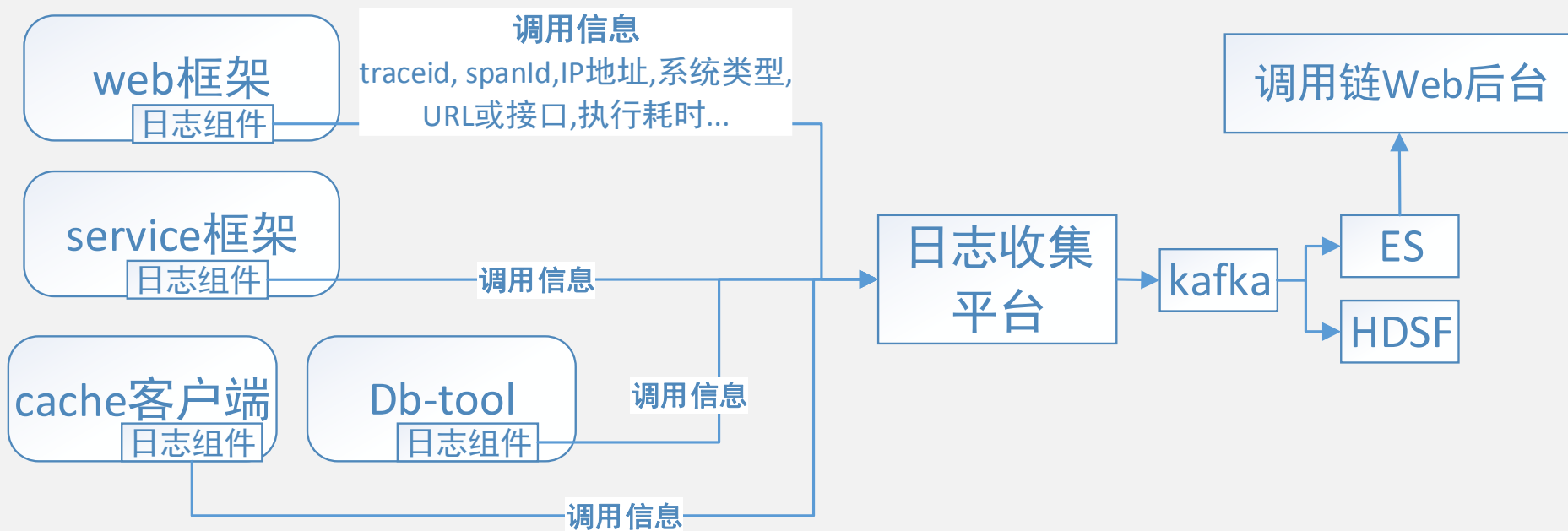
每个线程组独享一个工作队列 ;

线程组内工作线程数 m ;

**细化锁粒度，提高并发；**



## 基于日志的分布式调用跟踪系统

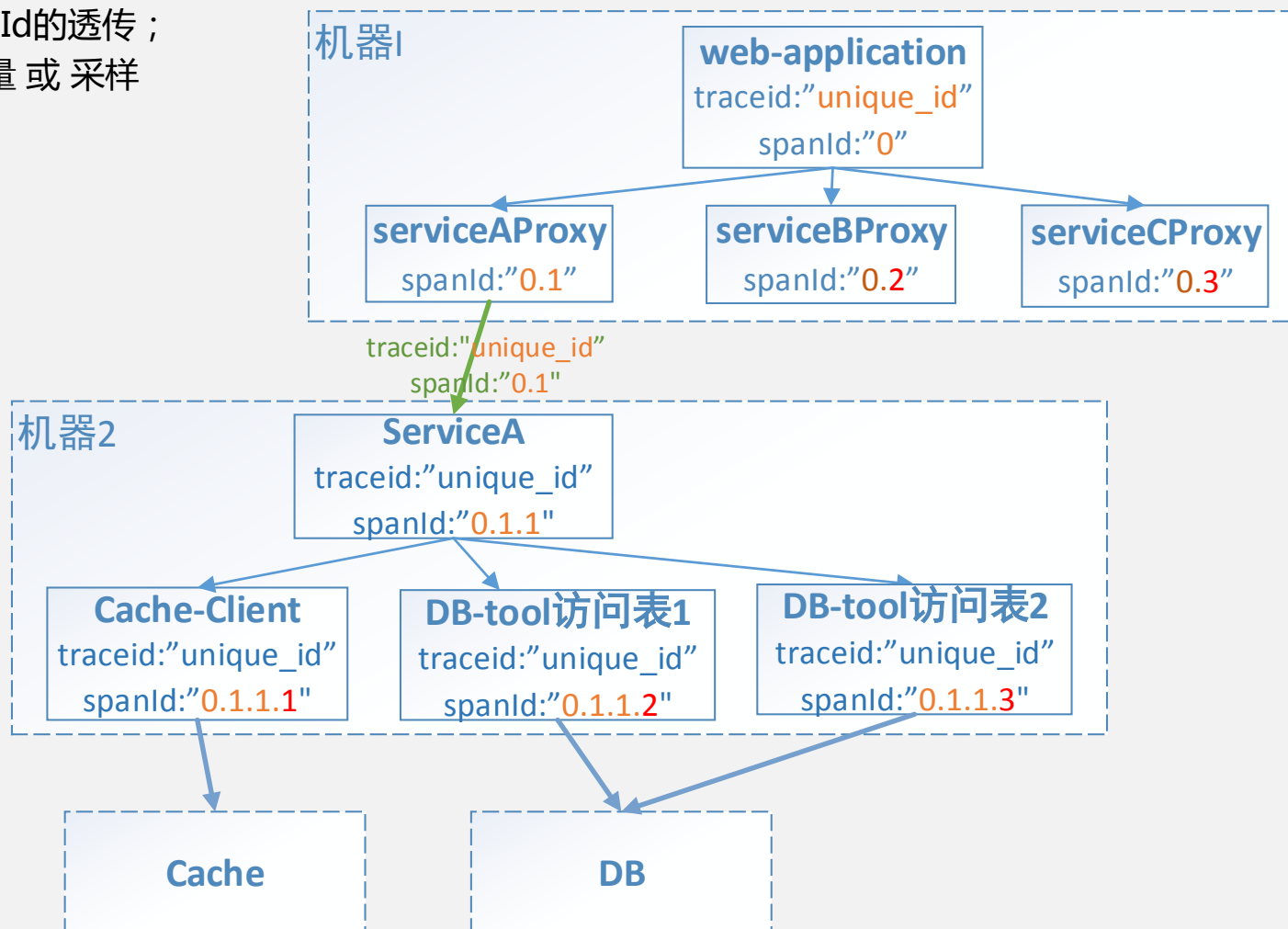


效果图：



## 相关技术点

- 1、全局唯一的traceid将单次请求调用链串联；
- 2、spanId描述调用层级关系；
- 3、traceid、spanId的透传；
- 4、数据采集：全量 或 采样







## ➤ WHY

服务化+服务治理

## ➤ HOW

到家实践：

### 1、服务发布&订阅

透明化服务地址，注册中心集中管理(分组、安全策略、健康检查)

### 2、集群容错&路由

可用性，负载

### 3、流控&告警

过载保护

### 4、数据库连接数 & 线程数

数据库过载保护 & 提高服务并发处理能力

### 5、调用跟踪

快速定位问题

### 6、服务依赖

反向推动架构优化

# THANK YOU

架构师之路

有评论必回哟。

