

坚如磐石

-高性能、高可用、可扩展商业平台设计

北京搜狗科技发展有限公司 王磊

About me

王磊 sadsardine@163.com

搜狗搜索广告商业平台技术负责人，主持了搜狗搜索广告商业平台的设计和开发，八年互联网软件系统研发经验，专注于高性能、高可用的软件架构和设计，在分布式计算、分布式数据存储访问、高性能面向服务架构设计、可伸缩系统架构、高可靠软件系统方面有丰富项目实战经验。

在来到搜狗之前王磊曾供职与百度、美国英网软件等公司，曾参与设计并开发过百度竞价排名平台、美国英网软件BI平台等多个系统。



1挑战

意义?

面临挑战、问题

2解决之道

架构、设计

案例、体会

3思考展望

趋势

01

挑战

意义

追求三高架构的意义

稳定可用的服务是一切的根本

性能决定收益



0.1秒的响应延迟 → **1%**的销售降低

Google

0.5秒的延迟 → 流量减少**15%**

跟上互联网的节奏

灵活多变的市场需求需要系统架构能快速支持

面临挑战



01

大数据量、高访问量、性能要求

02

支持复杂灵活业务、安全处理敏感数据

03

系统升级，软、硬件等故障不能影响服务

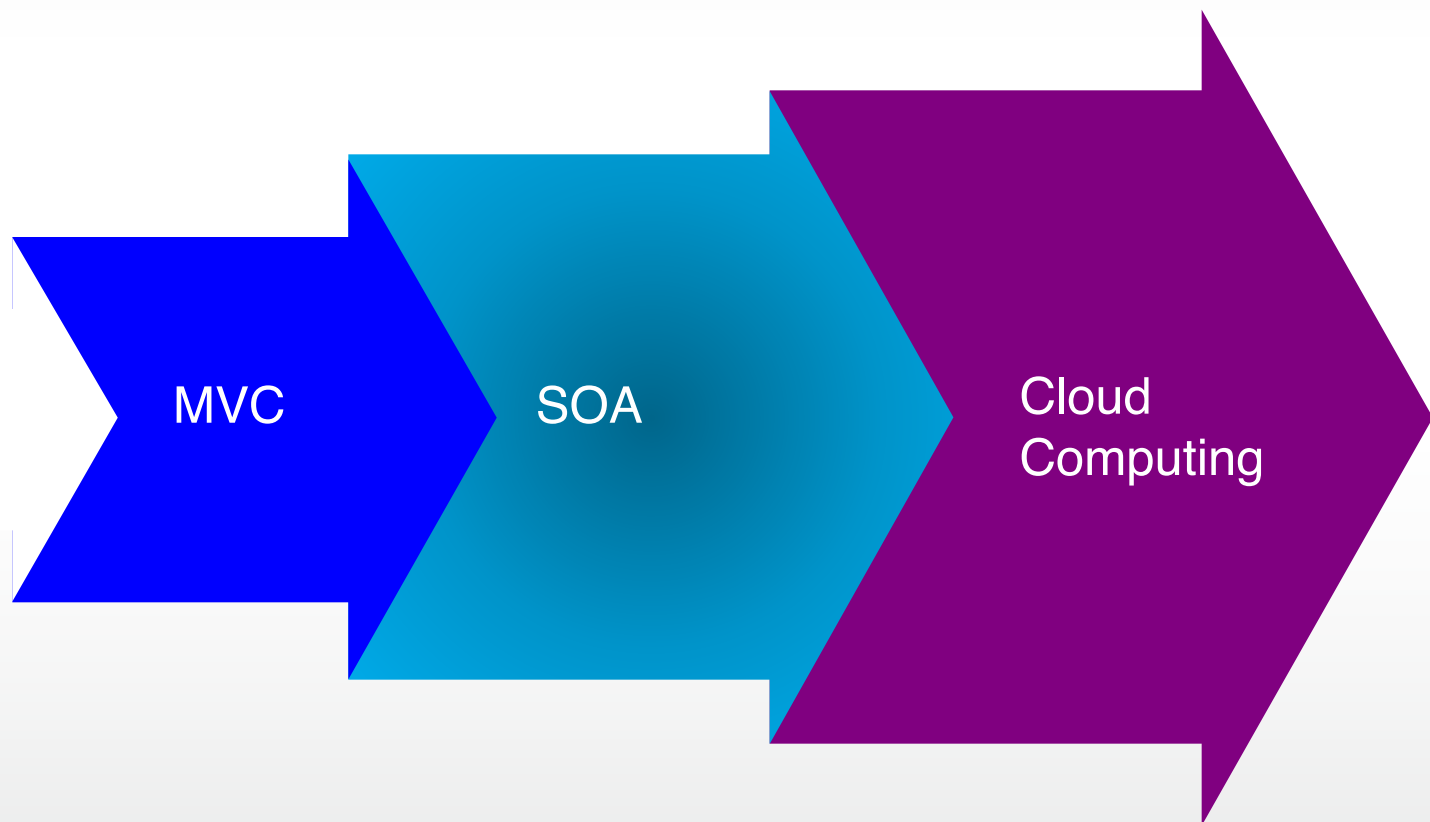
04

快速扩容支撑高速发展业务、突发事件

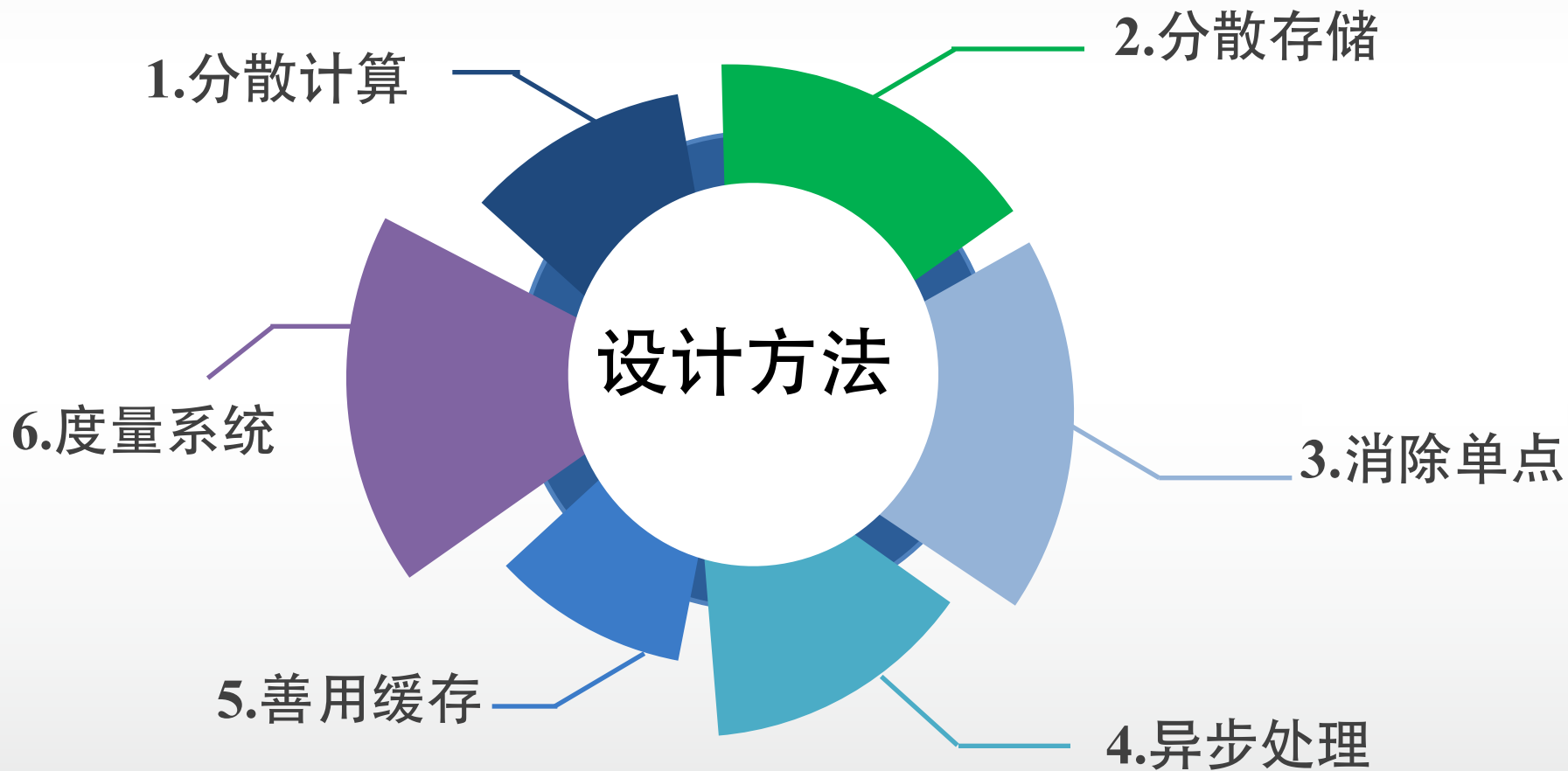
02

解决之道

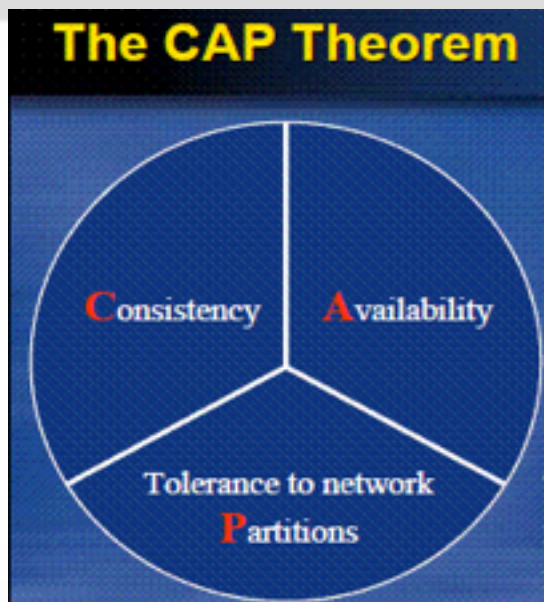
发展趋势



如何解决



CAP定理



先保证P、A，再满足C

Consistency(一致性), 数据一致更新, 所有数据变动都是同步的

Availability(可用性), 好的响应性能

Partition tolerance(分区容错性) 可靠性

三选二

1.分散计算

2.分散存储

设计方法

3.消除单点

4.异步处理

5.善用缓存

6.度量系统

分层设计

广告数据存储层

广告处理业务层

广告平台前端

集群计算



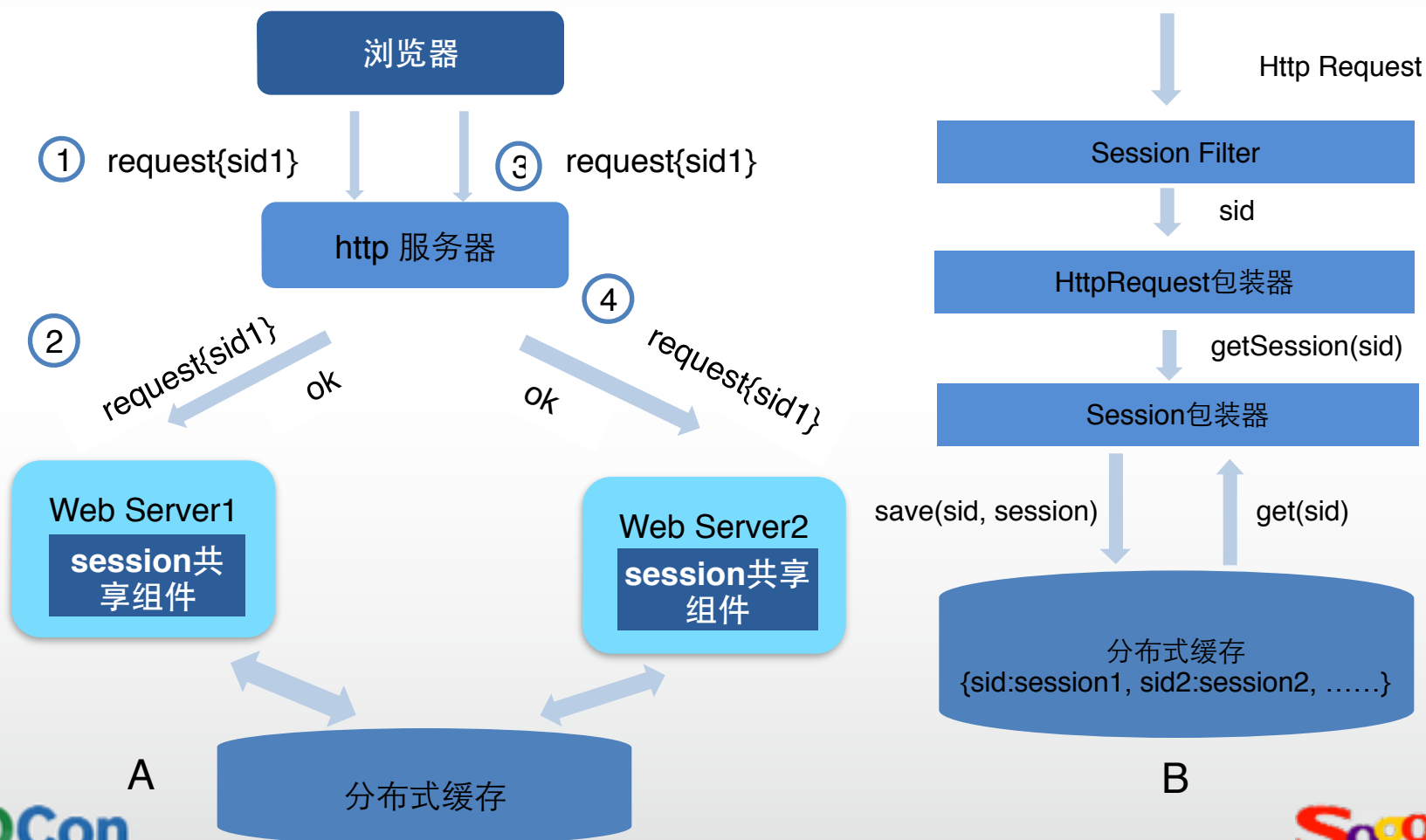
广告平台Web Server集群

共享
Session

Stateless



共享Session



业务层集群

SOA服务化



前端系统

使用服务

业务云

业务云

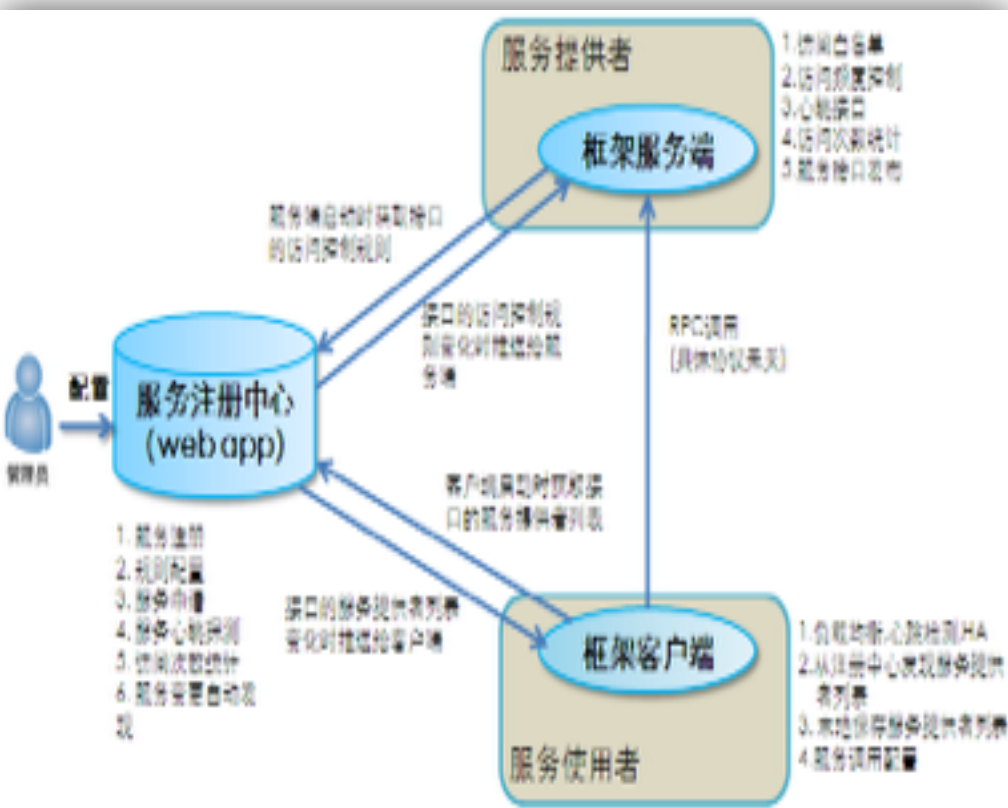
业务云

发现服务

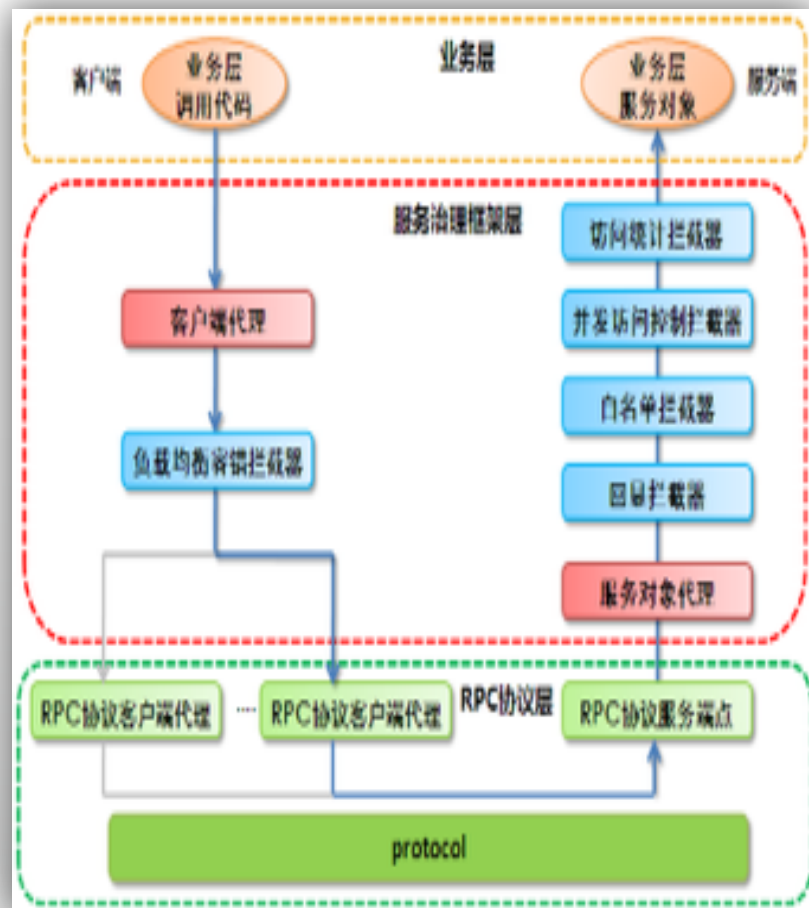
服务注册中心

注册服务

SOA服务治理框架

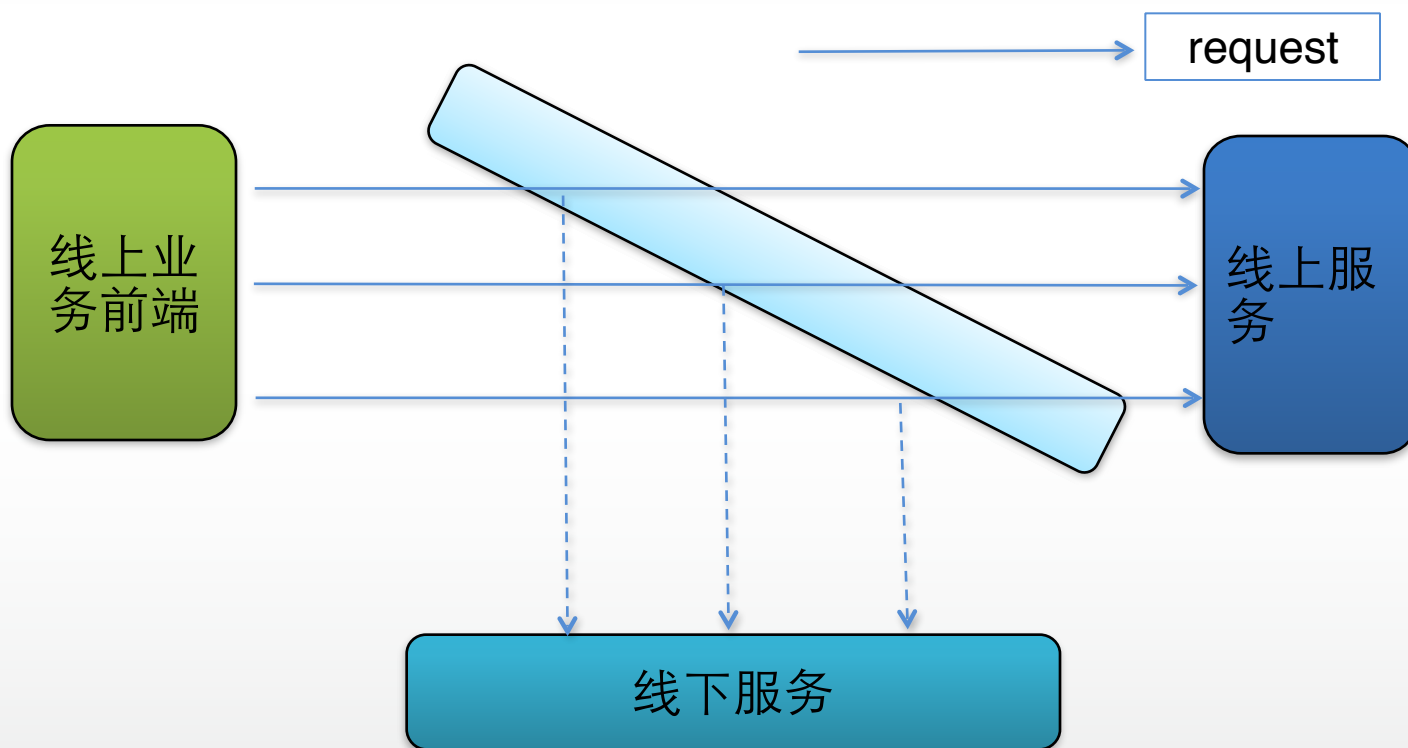


A



B

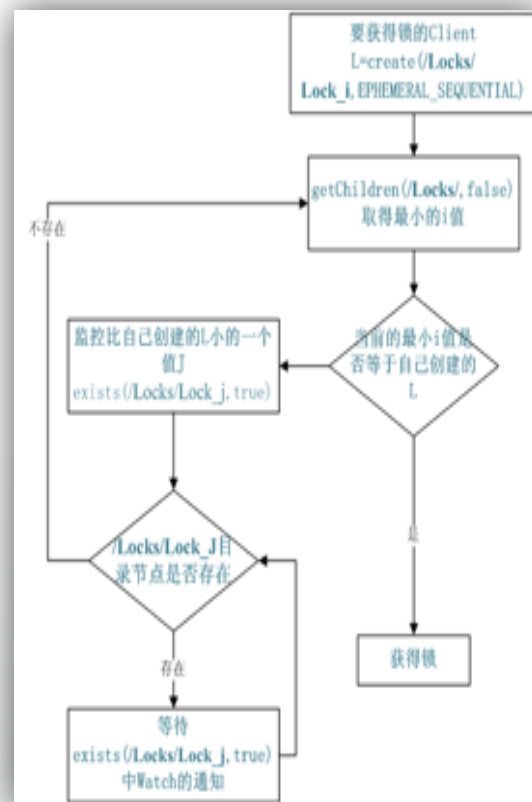
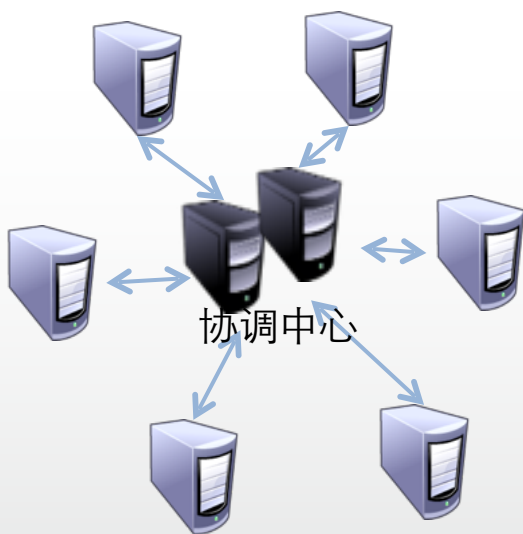
使用“棱镜”来重放线上操作



分布式并发与协同

集群行为协调

集群状态同步



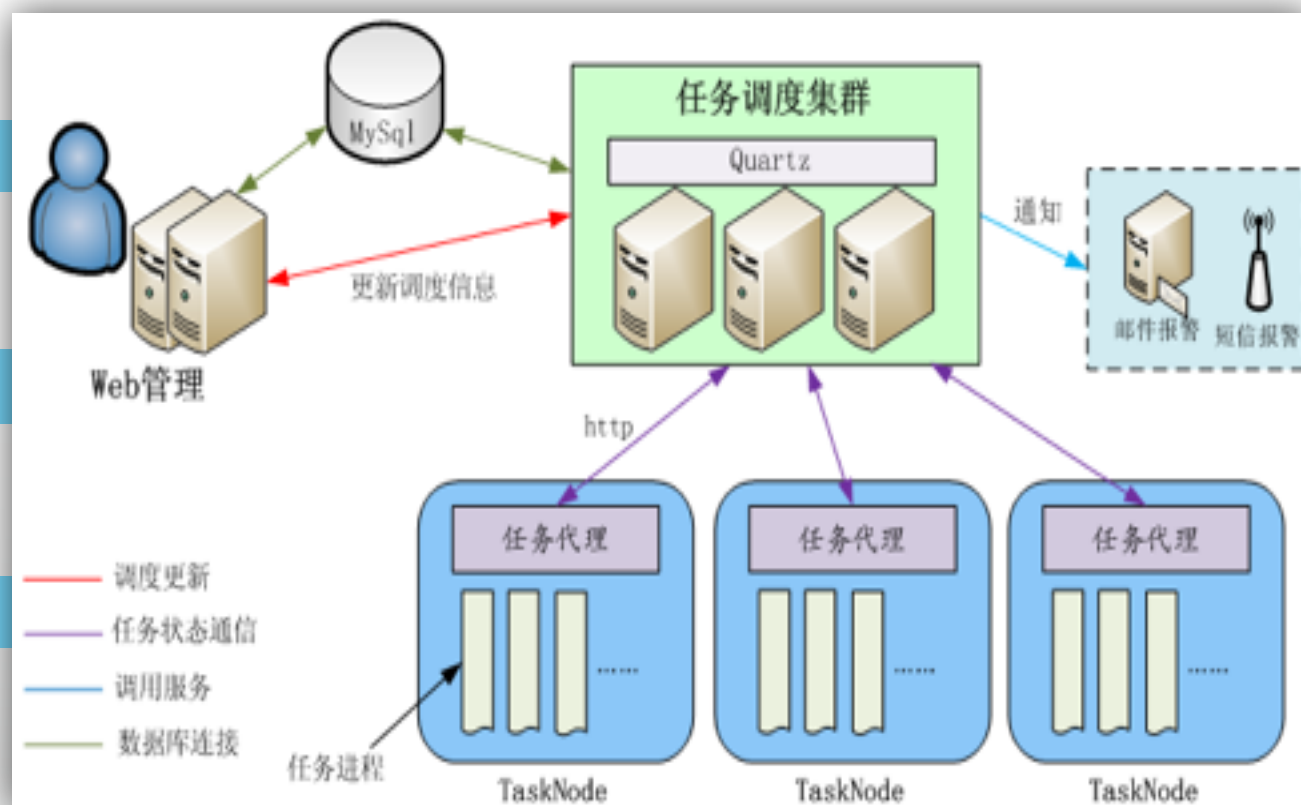
基于Zookeeper的分布式锁实现

分布式任务调度

分离任务调度和计算

多任务并行处理

任务调度执行高可靠



分散计算

① 分层设计

分散计算量

② 集群计算

使用共享Session构建Web Server集群

使用服务治理框架构建业务处理集群

构建分布式任务调度集群

解决集群分布式并发与协同

设计方法

1.分散计算

2.分散存储

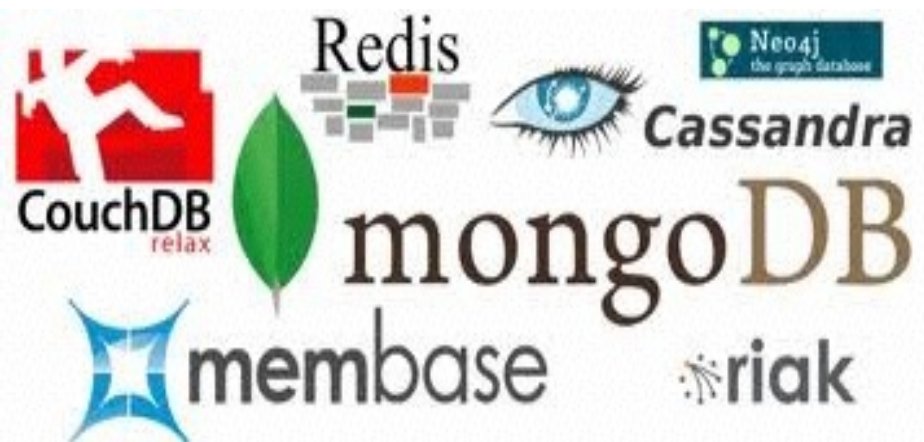
3.消除单点

4.异步处理

5.善用缓存

6.度量系统

RDBMS到RDBMS+NoSql



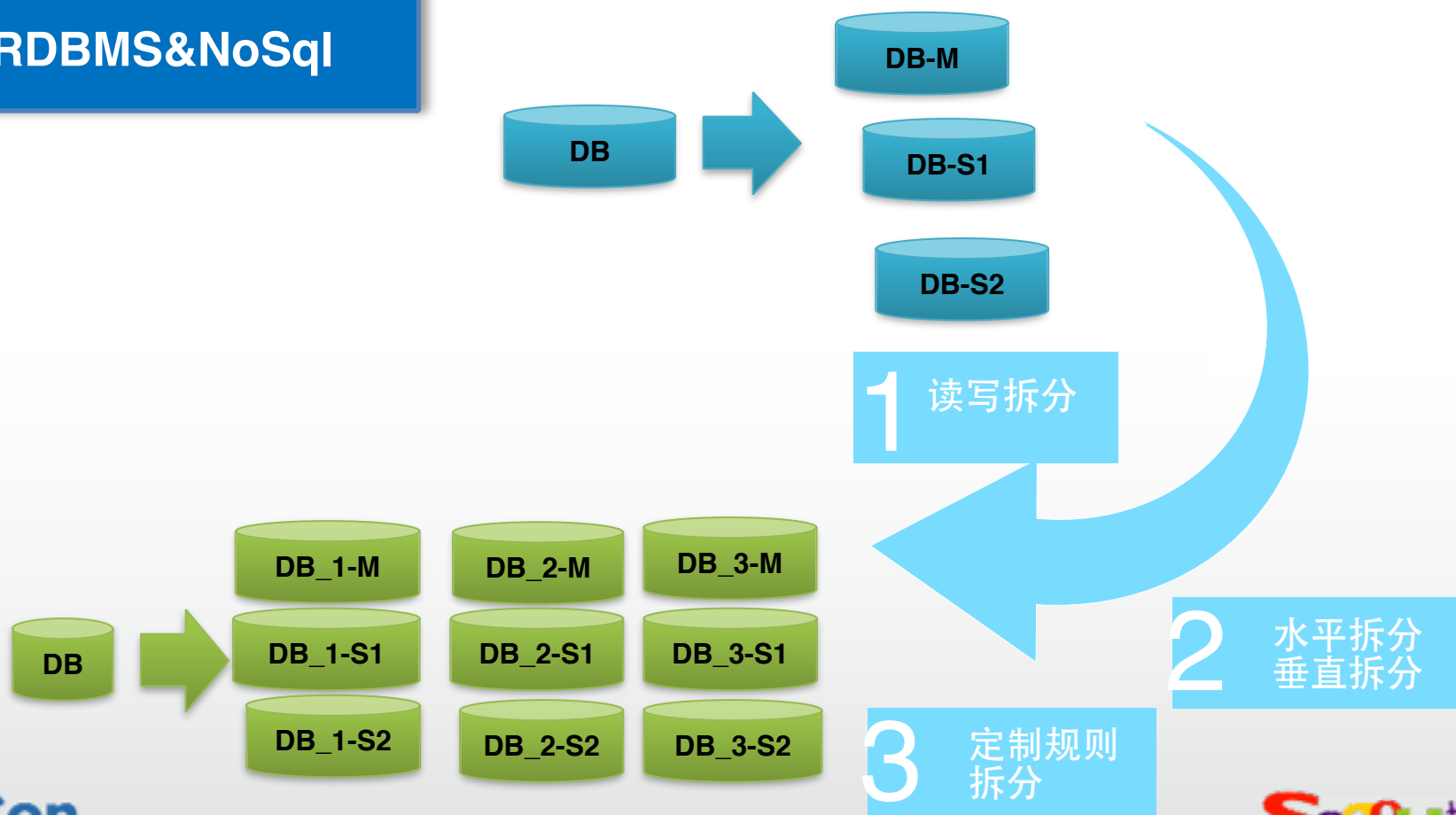
数据量大、性能要求高

非关系结构化数据

无ACID要求

集中式存储到分布式存储

RDBMS&NoSql



使用存储访问中间件



NoSql



Middleware



RDBMS

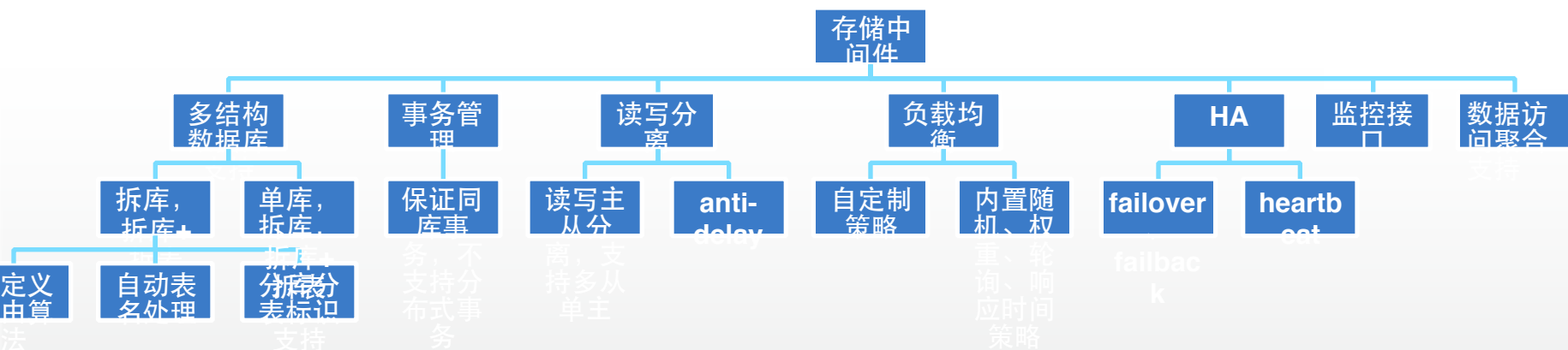


Middleware

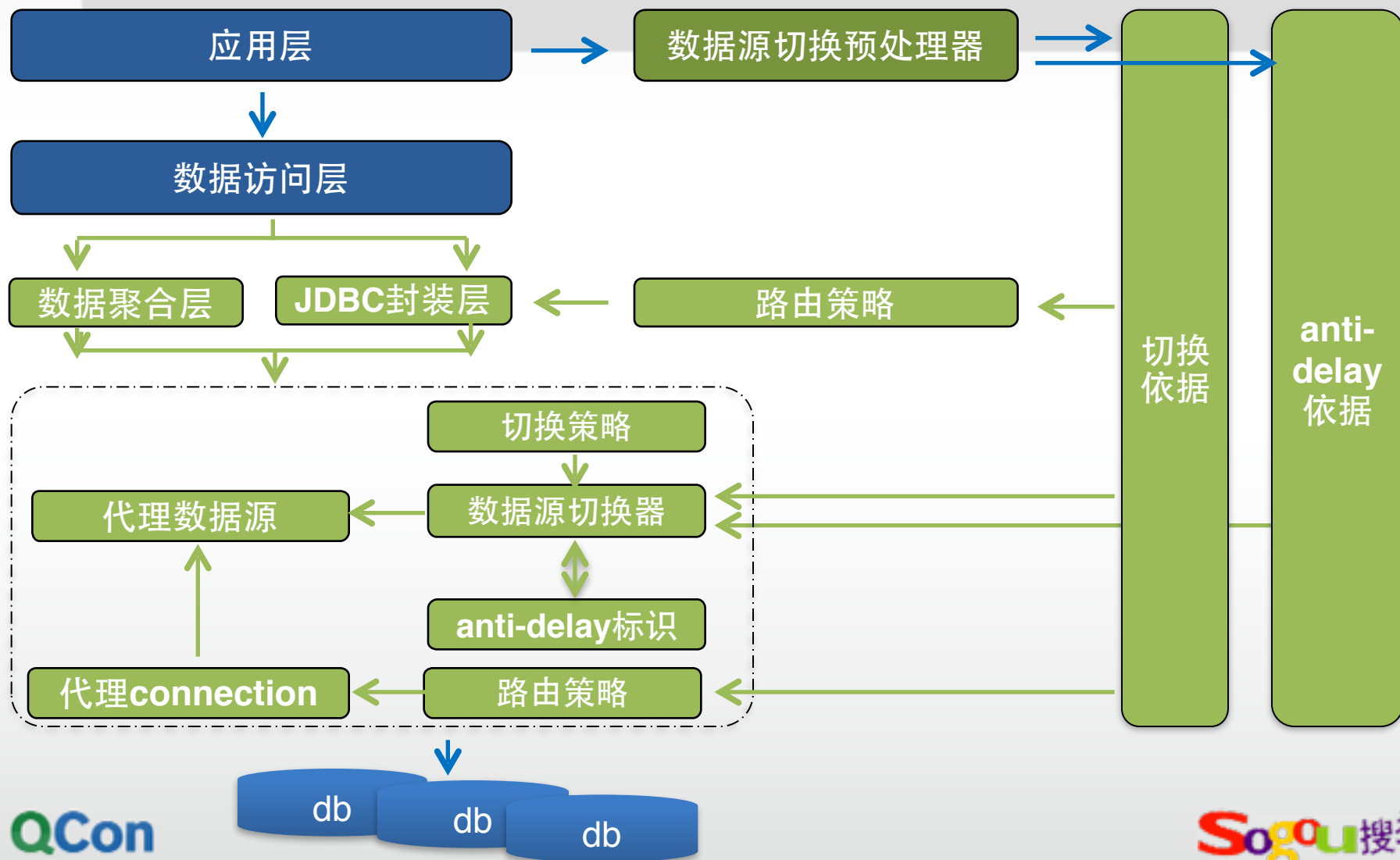


Application

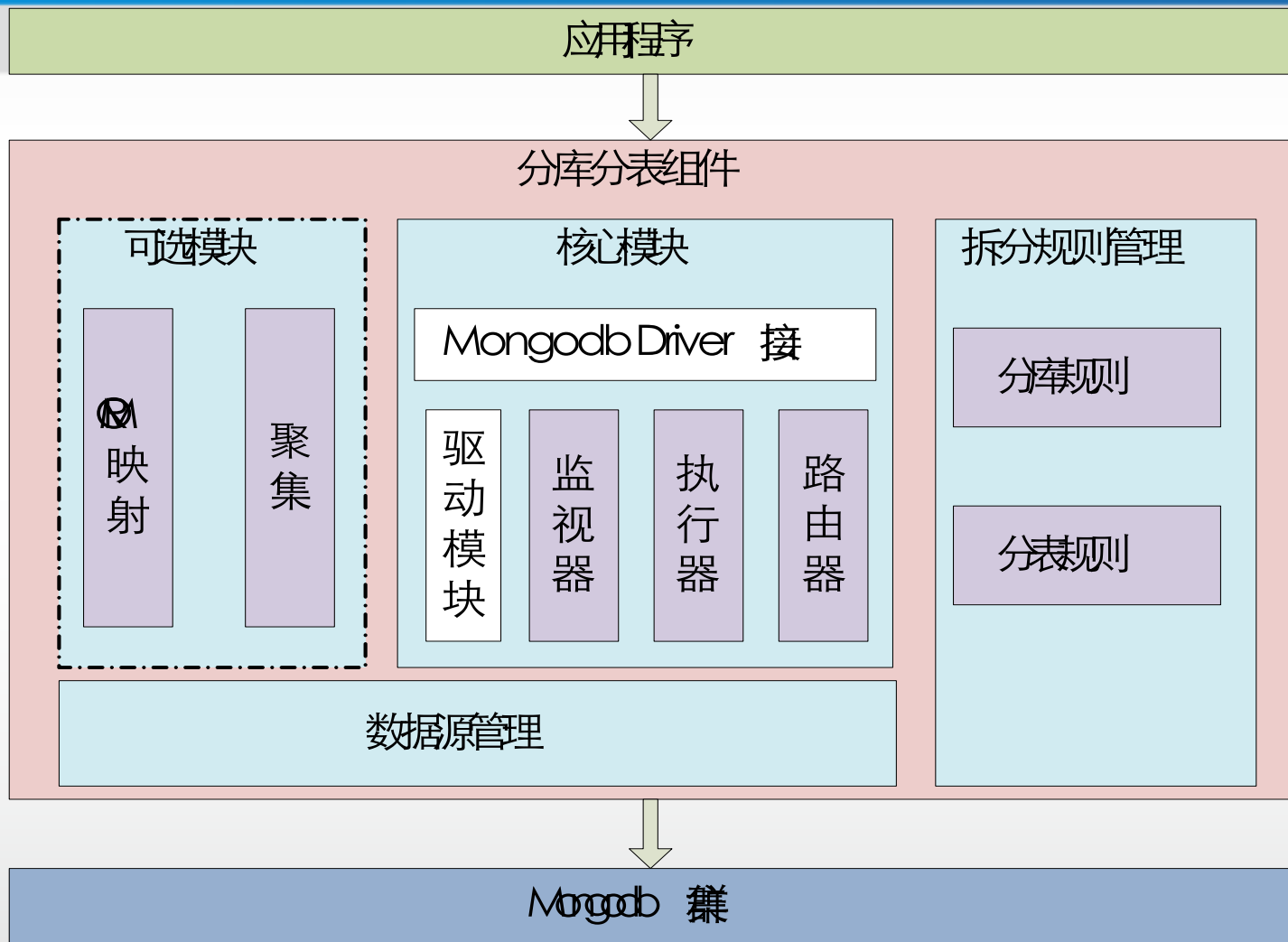
RDBMS存储中间件功能



RDBMS存储访问中间件结构



MongoDB存储中间件架构



分散存储

① 使用RBDMS +NoSql

大数据量、高性能；非结构化；无ACID；

② 集中式存储到分布式存储

使用数据存储中间件访问集群存储

- RDBMS 中间件
- MongoDB中间件

设计方法

1.分散计算

2.分散存储

3.消除单点

4.异步处理

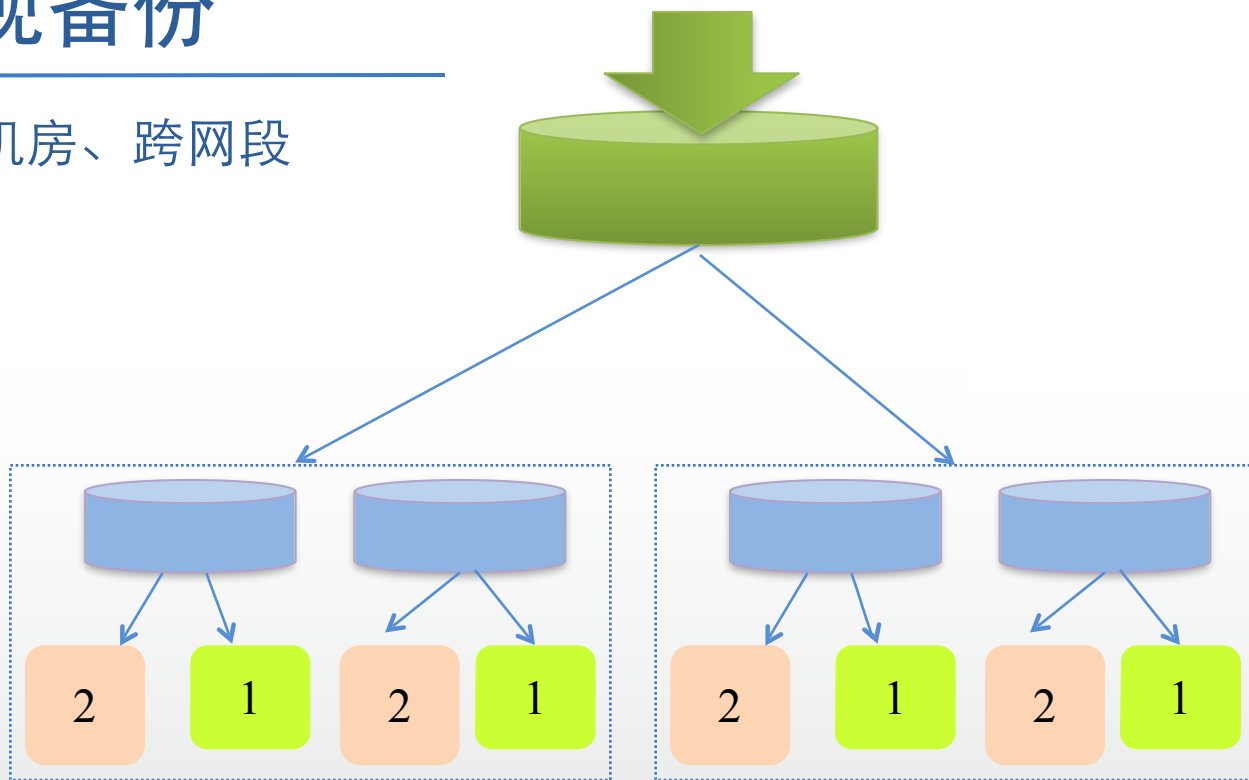
5.善用缓存

6.度量系统

构建集群

硬件实现备份

多机备份、跨机房、跨网段



构建集群

软件实现集群

集群架构

负载均衡

自动故障恢复

故障检测

冗余集群

集群处理相同的请求和数据

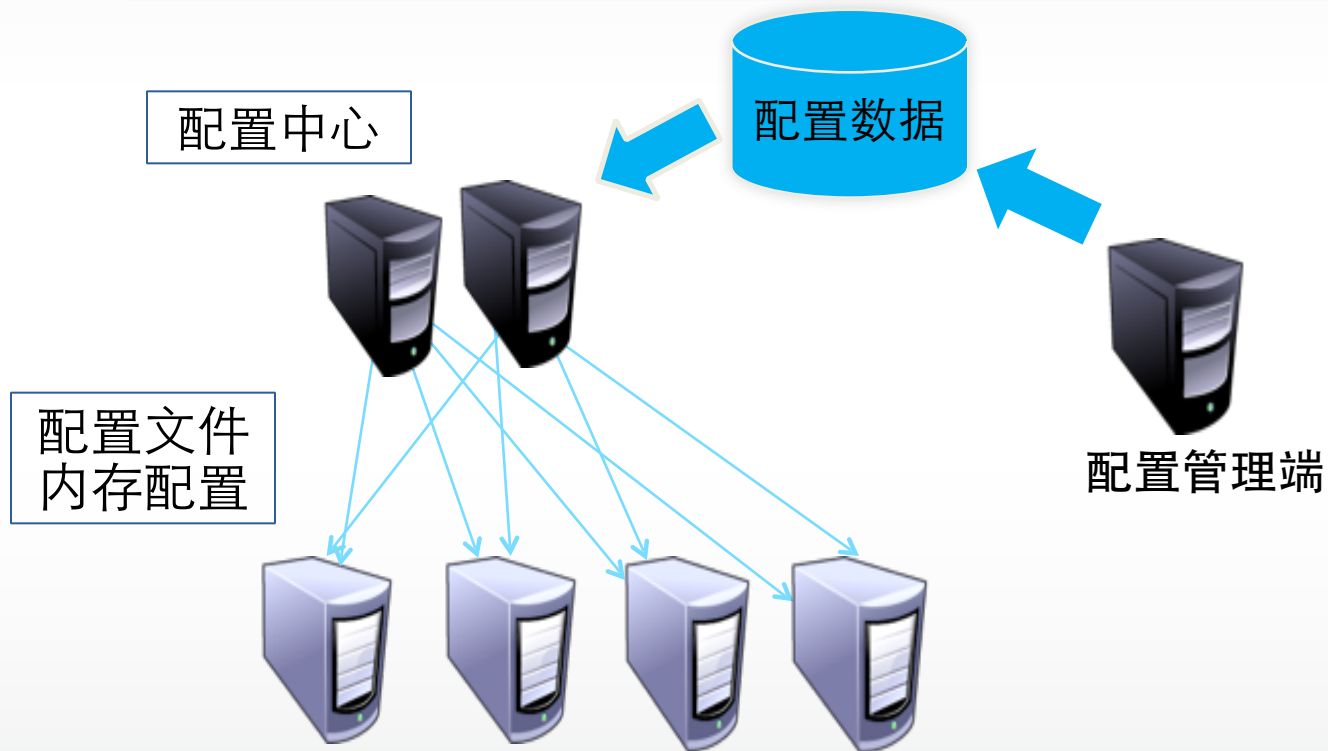


分区集群

集群处理不同的请求和数据



集群配置管理



cluster

application

config

group

item

集群配置集中管理

集群差异化配置

配置变化实时生效

快速支持集群扩容

设计方法

1.分散计算

2.分散存储

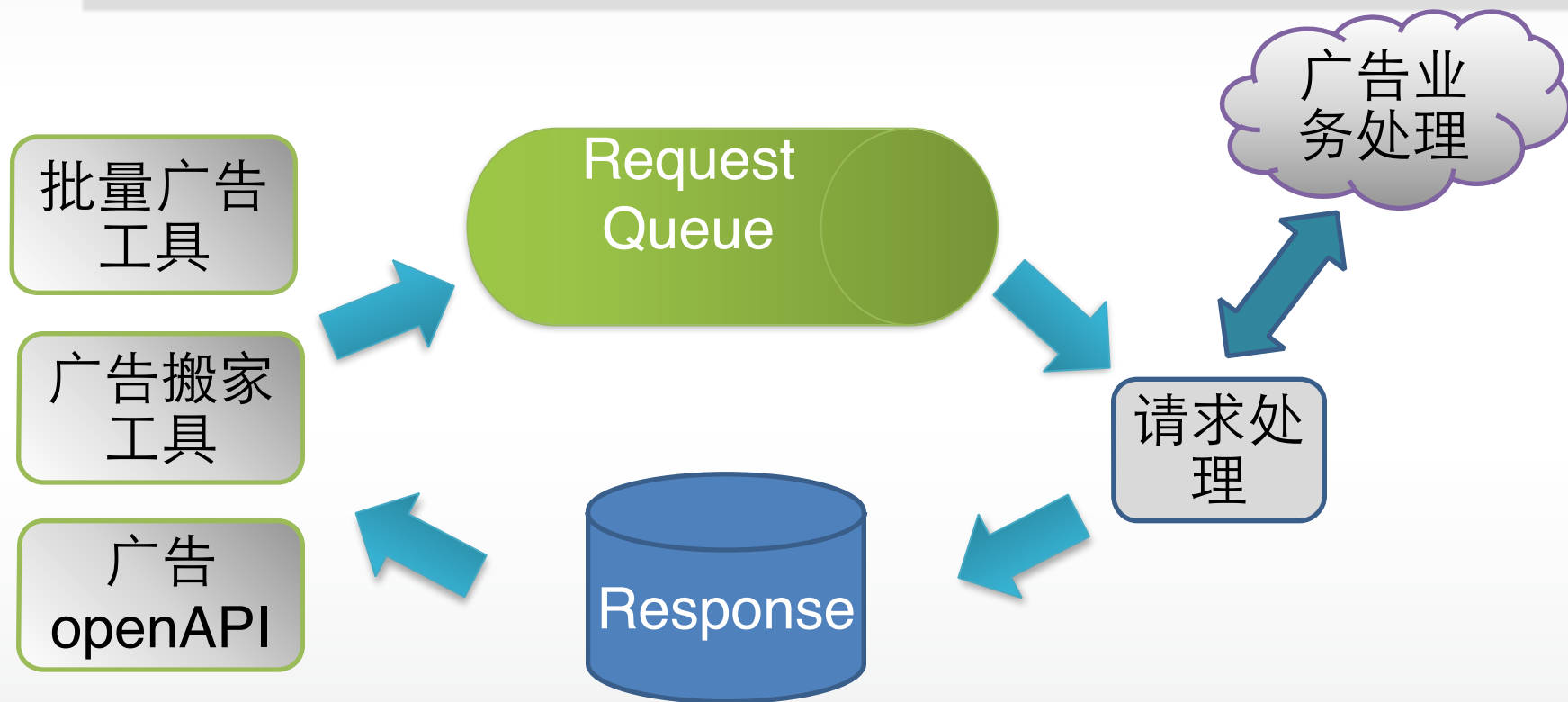
3.消除单点

4.异步处理

5.善用缓存

6.度量系统

异步处理架构



非实时、大数据、高并发请求

① 优点

- 非阻塞、高性能
- 资源利用率高
- 低耦合
- 可控制核心业务模块访问
- 易于扩展

② 设计关注点

- 请求消息重发
- 多请求处理者协调
- 请求队列时序性

设计方法



1.分散计算

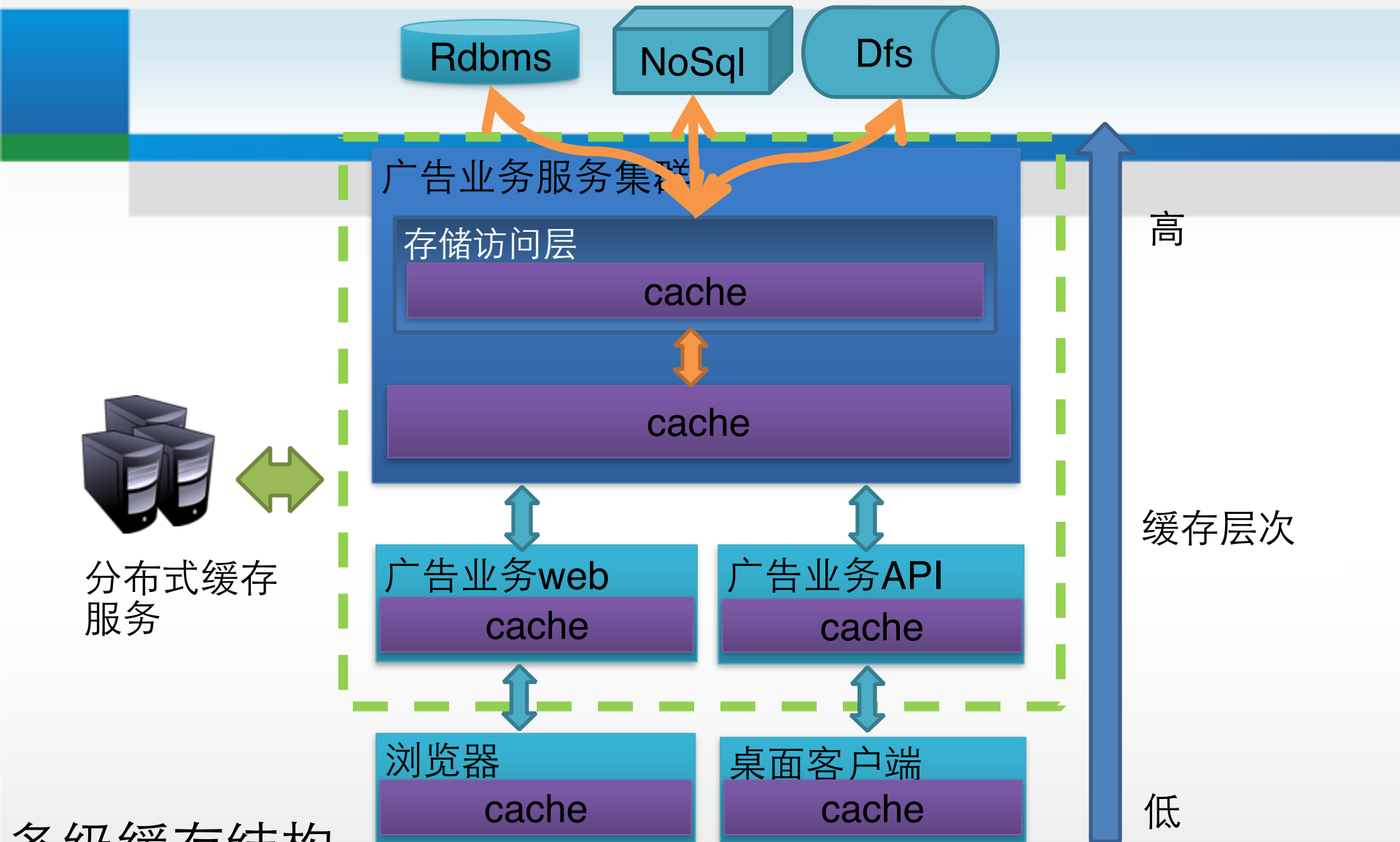
2.分散存储

3.消除单点

4.异步处理

5.善用缓存

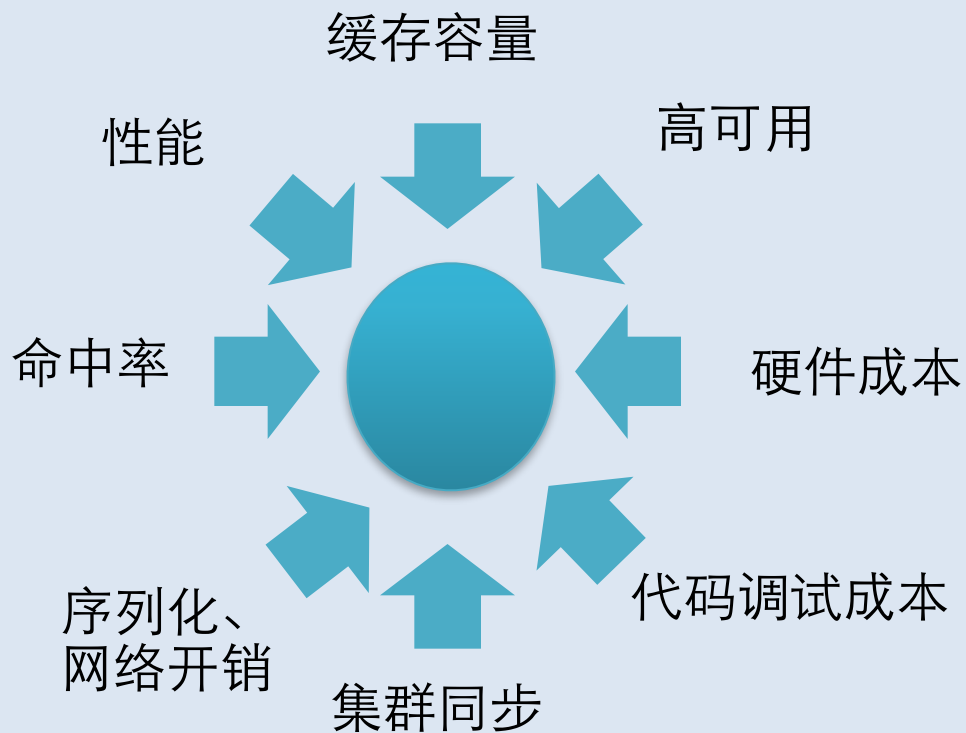
6.度量系统



多级缓存结构

按业务特点制定缓存方案

缓存设计



设计考虑

- 合适的缓存粒度
- 缓存开关
- 非复杂结构数据
- 缓存雪崩
- 缓存有限时间一致性、最终一致性
- 代码架构提升命中率

1.分散计算

2.分散存储

设计方法

3.消除单点

6.度量系统

5.系统缓存

4.异步处理

度量系统



扩容、优化决策



检测攻击



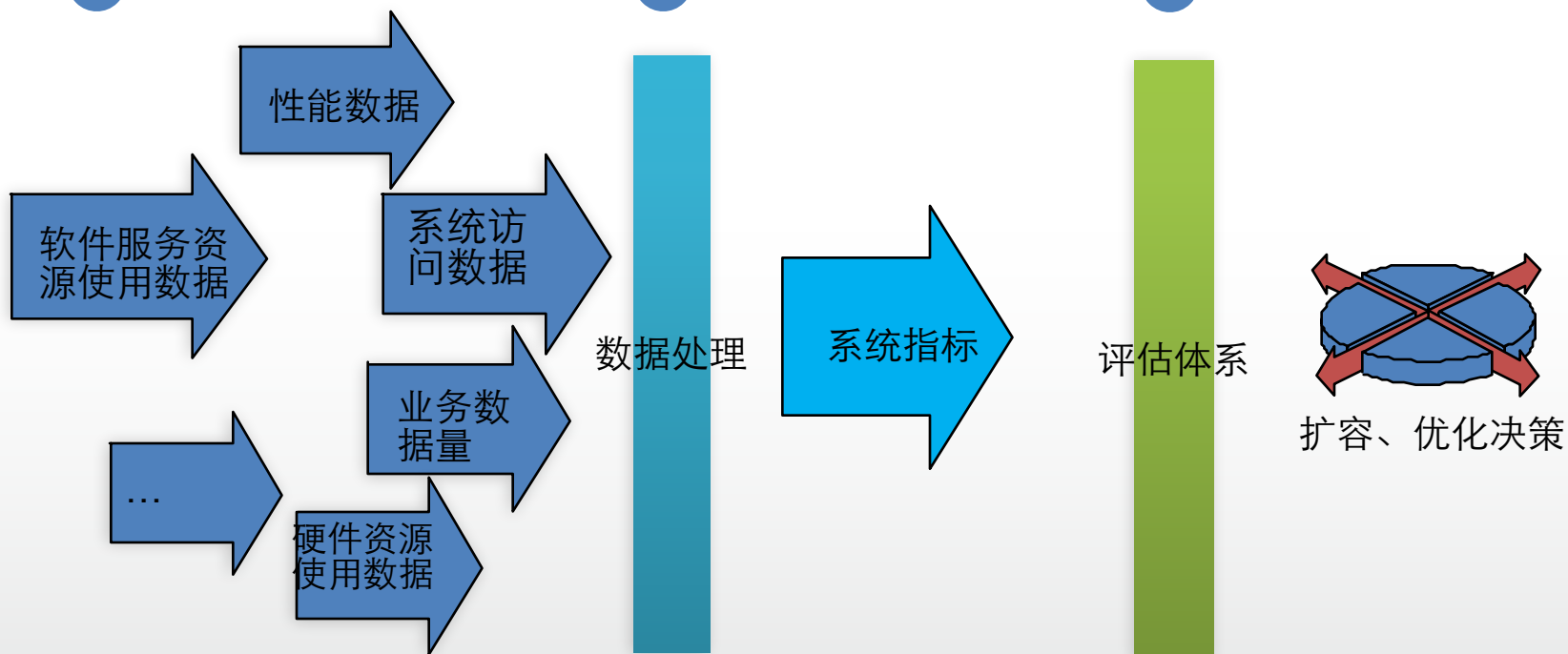
突发业务事件应对

构建测量体系

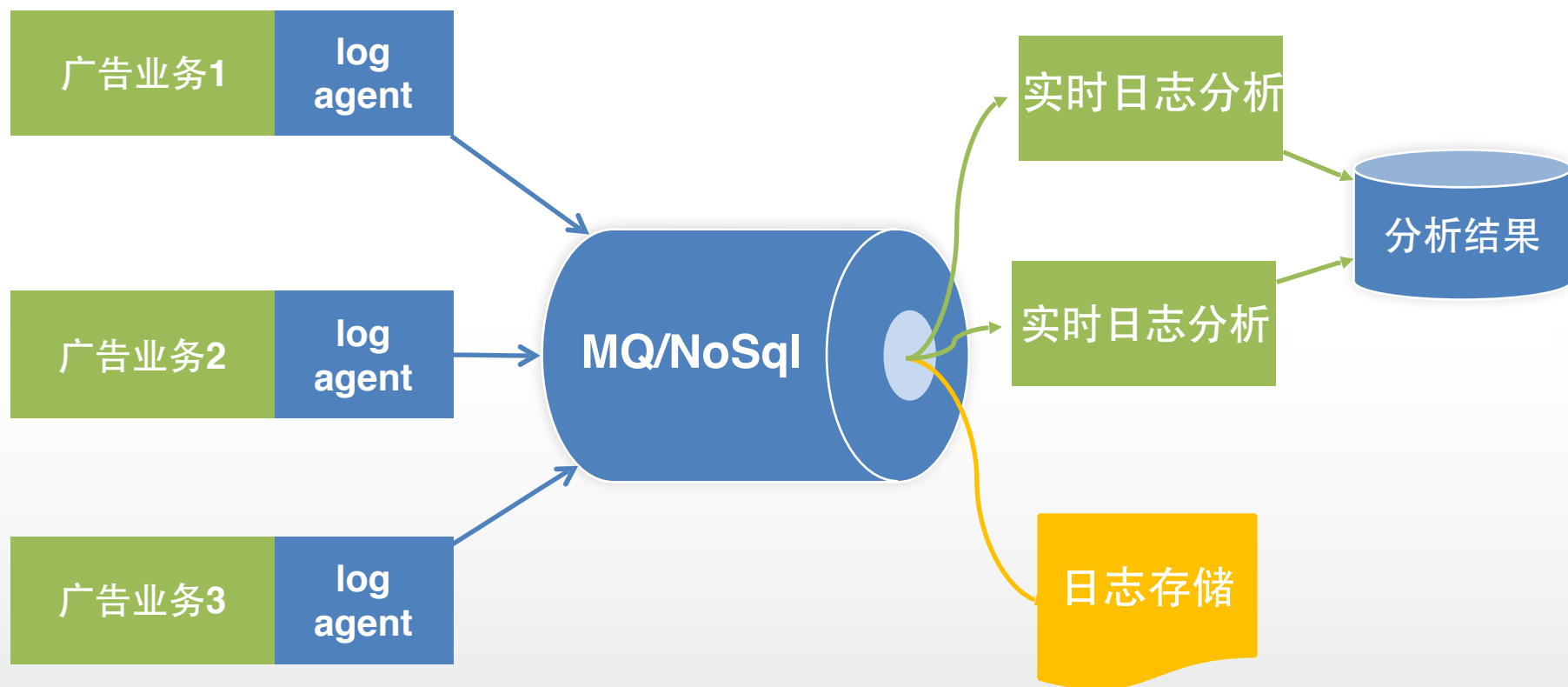
测量

评估

决策



日志分析处理



1 分散计算

进行分层设计、进行集群计算

3 消除单点

硬件实现备份、架构实现集群

5 善用缓存

使用多层缓存结构、根据业务特点设计缓存、考虑设计关注点

2 分散存储

使用NoSql、进行分布式存储、使用存储中间件

4 异步处理

使用异步处理提升性能、考虑设计关注点

6 度量系统

构建测量、评估、决策体系

03

思考展望

Future



Q & A

架构服务于业务
适合的才是最好的

THANKS !

王磊 sadsardine@163.com