



大规模社区构建- 贴吧的实践

李瀚

SACC2011

Outline

- ✓ 什么是贴吧
- ✓ 贴吧面临的技术挑战
- ✓ 贴吧的解决方案
 - ✓ 接入层
 - ✓ 业务逻辑层
 - ✓ 存储&通用服务层
 - ✓ 集群管理

什么是贴吧

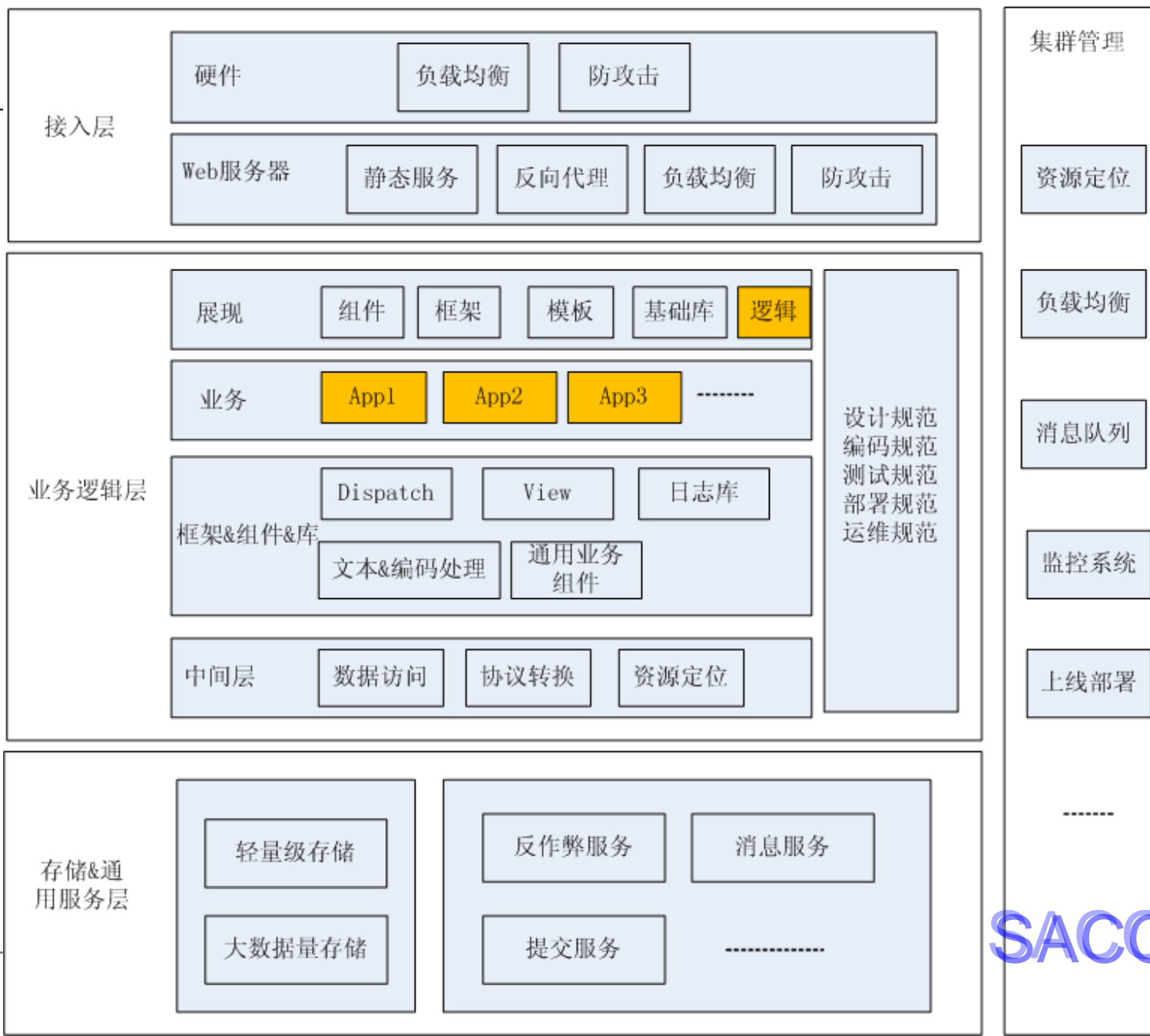
- ✓ 大型综合社区
 - 讨论区+视频区+相册区+游戏区+ itieba+无线贴吧+..
- ✓ 技术
 - 前端+lamp+nosql+数据挖掘+反作弊+无线+..

贴吧架构面对的技术挑战

- ✓ 流量&数据量
 - 百亿贴子的存储，某些热门主题可达千万回复
 - P级的视频数据存储
 - 来自浏览器每秒10w量级的浏览请求数
 - 内部每秒数十w量级的更新请求转发
- ✓ 快速开发
 - 快鱼吃慢鱼
- ✓ 可用性&数据安全性
 - 7*24小时的互联网服务，容灾，冗余
- ✓ 丰富的应用类型，迥异的访问模式
 - 数百个服务
 - 不同应用有不同要求：检索，推送

贴吧的社区技术解决方案

实践



SACC2011 百度

贴吧的存储架构解决方案

接入层

服务器端的最前线



接入层

- ✓ webserver
 - 异步
 - 集群化
 - 防攻击体系
 - 静态文件服务：跑满网卡

- ✓ 开发人员要做什么？

贴吧的存储架构解决方案

业务逻辑层

需求的集中地



业务逻辑层

- ✓ php开发框架
 - MVC
 - 统一开发模式
 - 性能
- ✓ 组件化
 - 公共库
 - 通用**

业务逻辑层

✓ 中间层

- 网络交互，负载均衡
- 协议转换，接口兼容

✓ 开发者要做什么？



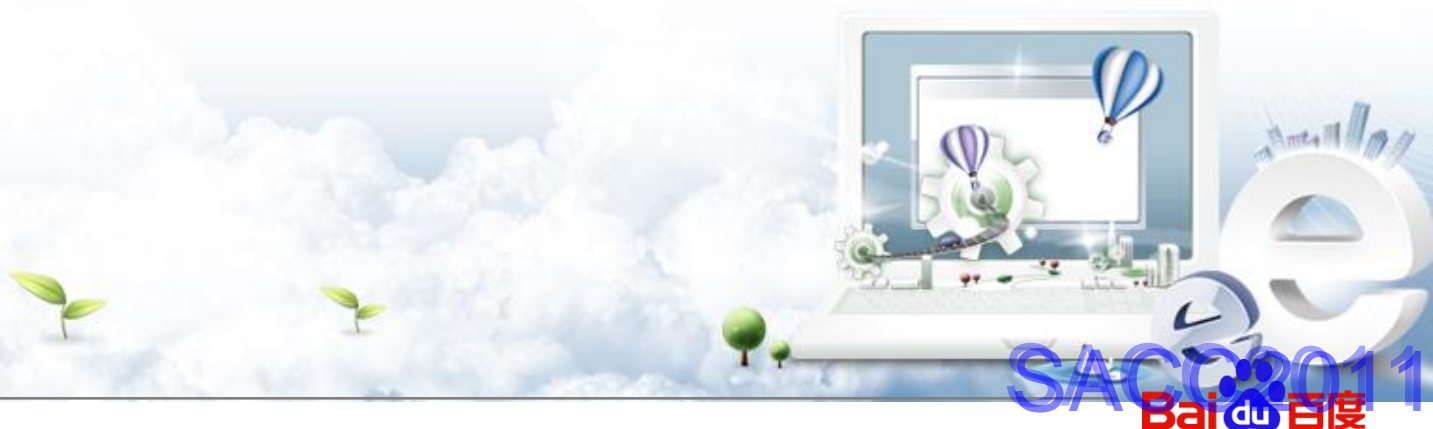
存储层

- ✓ 轻量型解决方案
- ✓ 大数据存储解决方案

贴吧的存储架构解决方案

轻量型解决方案

cover大部分日常快速开发需求



存储层-轻量型解决方案

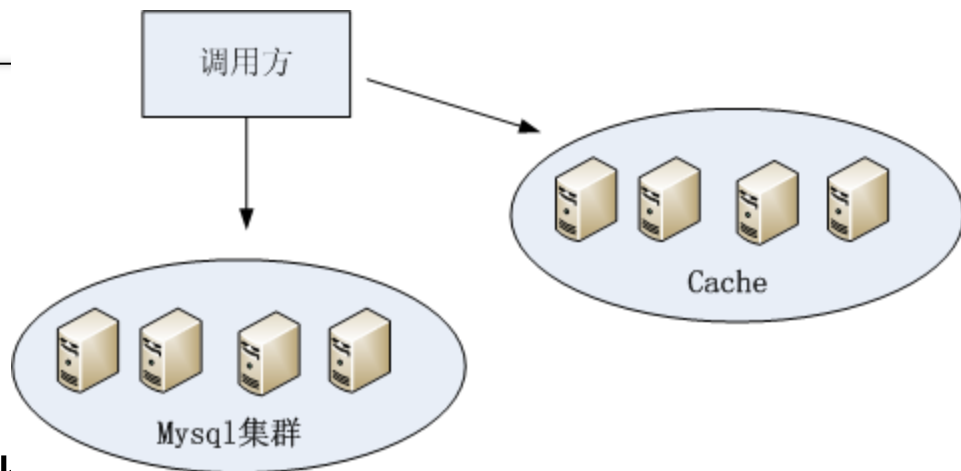
- ✓ Mysql+cache+flash
 - Mysql : 持久化
 - Cache : 加速
 - Flash : 硬件scale up
- ✓ 目标
 - 解决80%的日常产品开发需求

Mysql-单机

- ✓ 单机
 - 引擎选择，引擎优化，访问模式设计，表设计
 - 性能：一般几百qps到几千qps不等，数据量<1T
- ✓ 分布式
 - 统一接入，读写分离

Cache-加速

- ✓ Cache的性能
 - 5-10w qps , 瓶颈在网卡
 - 2-8法则
- ✓ Cache的种类
 - 页面级cache vs 单条数据的cache
 - Ex: 贴子内容页 vs 贴吧图片页
- ✓ 设计难点：cache更新
- ✓ 局限性
 - 只解决浏览瓶颈，不解决更新瓶颈



Flash卡

- ✓ Flash卡：天下有免费的午餐
 - 随机读写性能比磁盘有量级上的提升
- ✓ 缺点
 - 存储空间

参考资料

轻量型社区存储：mysql+cache+flash

- ✓ 适用场景：常规需求
 - 单机数据量几百G量级
 - 流量亿量级
- ✓ 优点
 - 开发灵活快速
 - 维护成本低
- ✓ 缺点
 - 通用存储，性能受限

大数据存储解决方案

Cover某些特定的大数据量产品需求



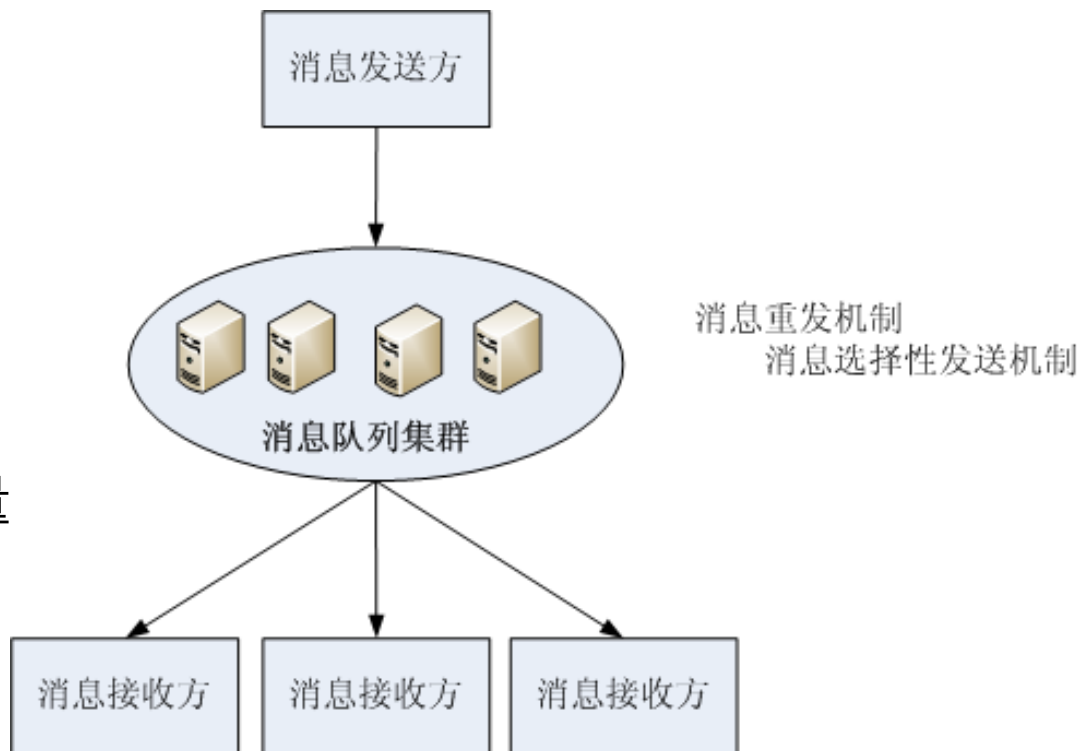
分区

- ✓ 分区概念
 - 垂直分区：按功能
 - 水平分区：按key

- ✓ 分区的目的
 - 冗余
 - 可扩展性
 - 性能：将不同的访问模式分开，利于优化

分区-消息队列(MQ)

- ✓ 分区的实现：消息队列
- ✓ 消息队列
 - Replication
 - 可靠性：
- ✓ 贴吧的消息队列集群
 - 峰值数十w/s的转发量

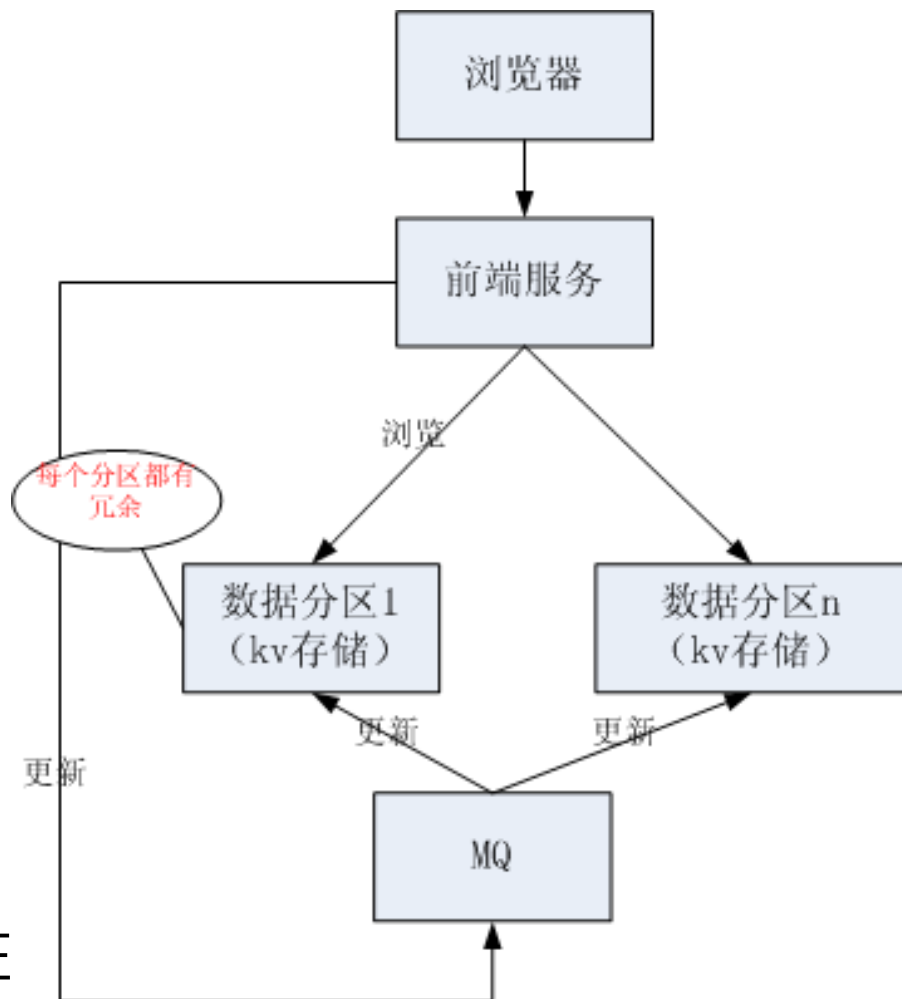


贴吧帖子存储

- ✓ 性能
 - 随机存储和连续存储
 - 内存patch
 - 多种cache
- ✓ 单机数据安全性
 - Binlog
- ✓ 整体数据安全性
 - 消息队列
- ✓ 效果
 - 单机可以跑满网卡

Key-value存储

- ✓ 视频存储
 - 查询模式：
 - 视频id->视频流
 - 数据量P量级
 - 典型的KV存储
- ✓ 单机kv设计考虑
 - 数据安全性
 - 可和外围cache配合使用
- ✓ 优点
 - 模式简单，易于分片
 - 采用追加写，更新性能有保证
- ✓ 缺点
 - 不支持关系查询



大数据量存储解决方案

- ✓ 适用场合
 - 某些数据量特别大或者对性能要求特别苛刻的应用
 - 某些需要特殊功能的需求
- ✓ 优点
 - 专用存储，性能可以极限优化
- ✓ 缺点
 - 开发维护代价较高
 - 灵活性偏弱
- ✓ 更多的例子：检索，推送，日志分析等

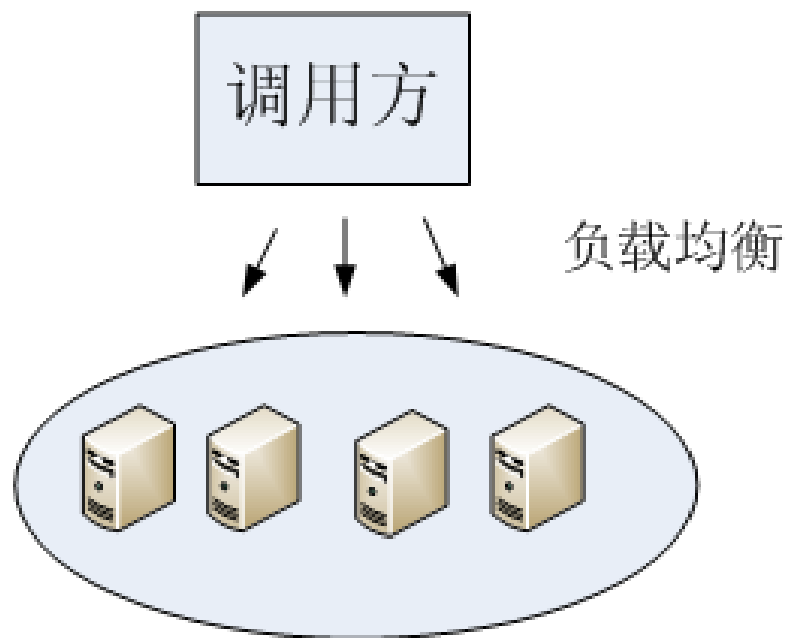
服务集群管理方案

解决机器和服务数量增多带来的管理问题



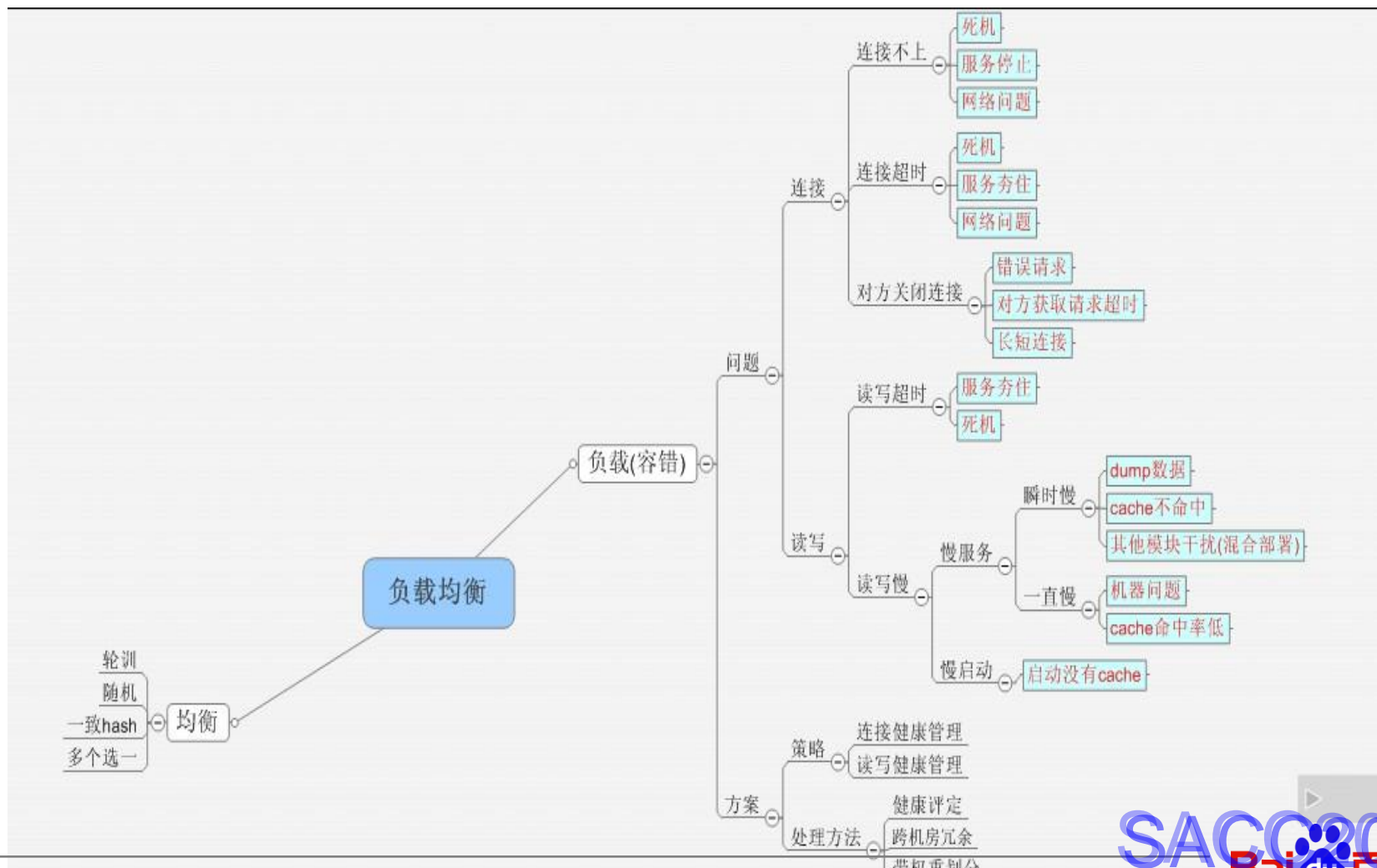
负载均衡

- ✓ 面向的问题
 - 服务故障
 - 蝴蝶效应
 - 数据迁移
 - 机器差异
 - 等等



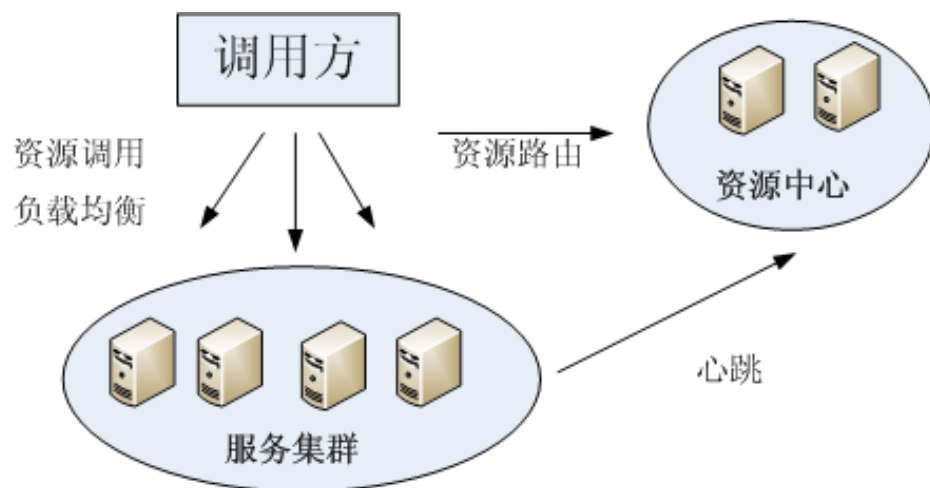
负载均衡

✓ [参考文章](#)



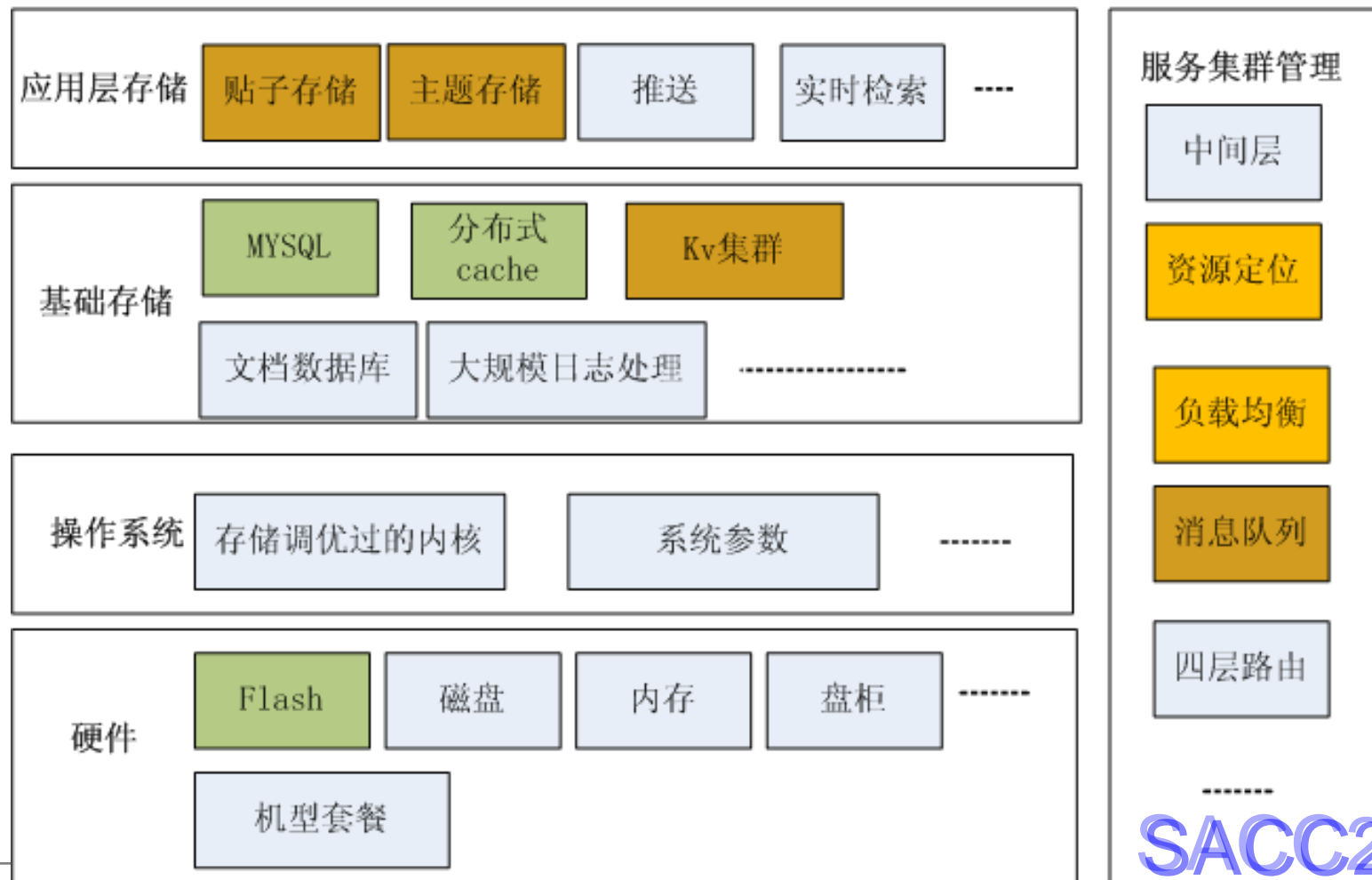
资源定位

- ✓ 服务数量扩大带来的问题
- ✓ 资源定位
 - 资源中心：服务元信息存储
 - 资源发现
 - 资源路由
- ✓ 设计思路
 - 心跳机制
 - 资源中心的单点问题和性能问题



大规模社区存储方案

社区存储&集群技术



回顾

- ✓ 流量&数据量：
 - 大数据量存储解决方案
- ✓ 快速开发
 - ✓ 业务逻辑层的快速开发框架
 - ✓ 组件化，通用化，集群化
 - ✓ 轻量级存储方案
- ✓ 可用性&数据安全性
 - 集群化技术
 - 存储技术
- ✓ 丰富的应用类型，迥异的访问模式
 - 多样化的存储解决方案

- ✓ [贴吧技术blog](#)
- ✓ 这里提供全面的社区技术实践机会 :mail : jointieba@baidu.com

FAQ

