

中间件

目标

1. 掌握缓存机制和 Redis 的用法
2. 掌握消息处理机制和消息队列软件 RabbitMQ
3. 掌握 RPC 消息协议和 gRPC

缓存

企业级应用为什么需要缓存和消息队列？

- 随着业务规模扩大,单体应用不再满足业务场景需求
- 用户数量剧增,单凭数据库无法扛住并发压力
- 业务场景复杂需要解耦
- 业务场景丰富造成访问热点

缓存

缓存的分类：

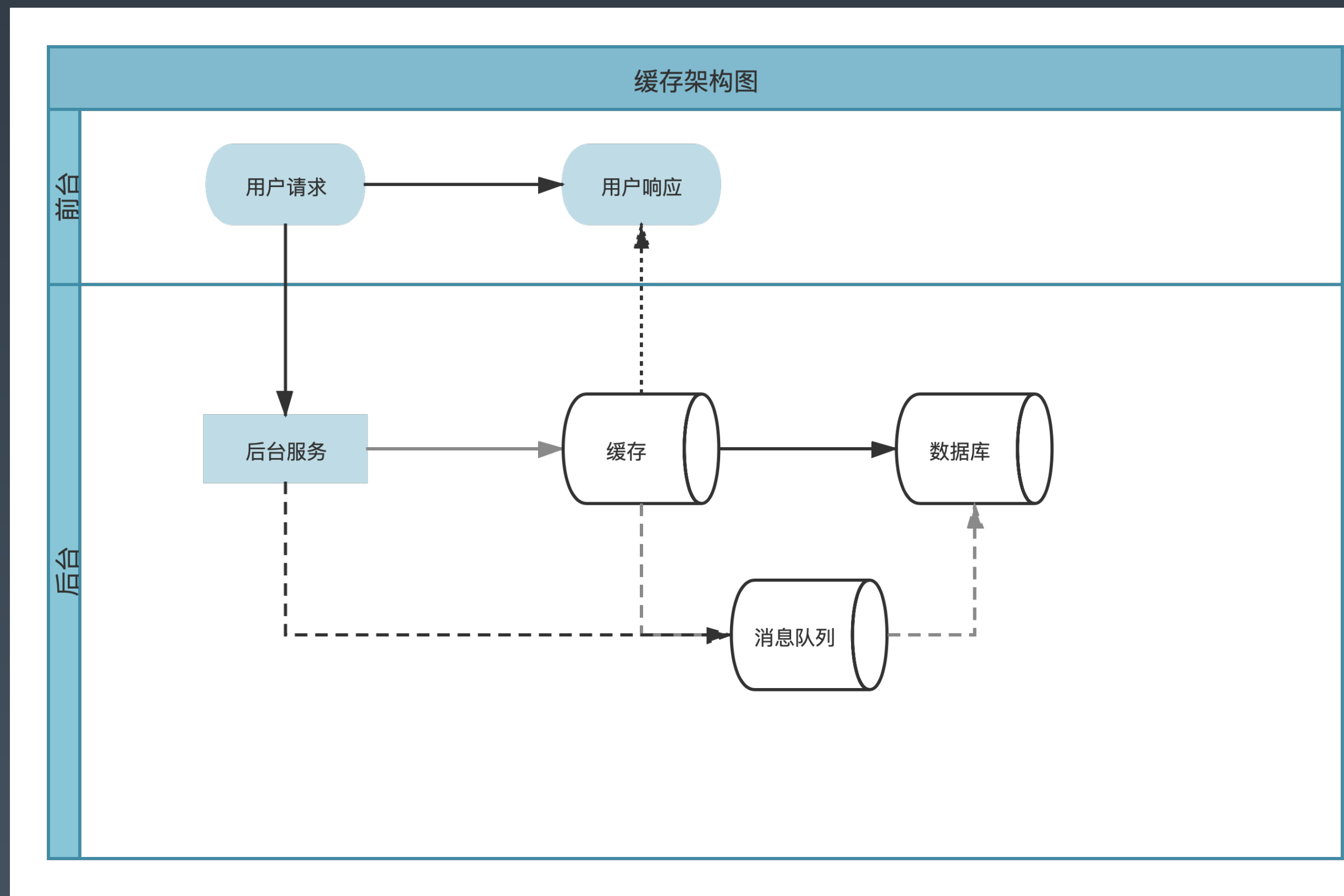
- 本地缓存
- 分布式缓存

差异：调用方式、开销

思考： 1. 各自存储数据的特点是什么？
2. 各自适合的场景是什么？

缓存

缓存与数据库之间的同步方式



缓存

缓存与数据库之间的同步方式:

Cache Aside 方式

- 读取数据流程详解
- 更新数据流程详解

整体读写方式

- 缓存和数据库是一个整体,业务逻辑不需要关注数据是来自缓存还是数据库

先写缓存方式

- 读写均在缓存,由队列异步写入数据库

思考: 不同同步方式各自有哪些应用场景

缓存

应用与缓存的同步策略

- **双写**：数据库和 Redis 各一份
- **消息队列**：写数据库和消息队列,消息队列再同步到 Redis
- **MySQL binlog**：使用 MySQL binlog 重放同步到 Redis

问题： 1. 三种同步策略优劣和应用场景是什么？
2. 各方式优劣？

缓存

架构中使用缓存有可能带来的问题？

- 缓存穿透
- 缓存并发
- 缓存雪崩

Redis

Redis 的特点:

- 使用 IO 多路复用机制
- 监听多个文件描述符实现读写事件
- 单线程（6.0以前）

注意：单线程会导致多个会话会相互干扰

Redis

Redis 的对象类型

- 字符串
- 列表
- 哈希
- 集合
- 有序集合

实战：Python 连接和操作 Redis

Redis

Redis 的重要概念

- 事务
- 管道
- 生存时间

实战： 1. Python 使用 Redis 实现计数器
2. Python 使用 Redis 实现队列

消息队列



生活中的排队问题

消息队列

计算机中的消息队列有什么用途：

- 异步处理
- 流量控制
- 服务解耦

消息队列的模型：

- 队列模型
- 发布-订阅模型

消息队列

生产级消息队列重要的三个问题：

- 如何保证事务在分布式消息队列的一致性？
- 如何保证消息不会丢？
- 如何处理重复消息？

RabbitMQ

AMQP 协议与 RabbitMQ 软件

- RabbitMQ 是 AMQP 的代理服务器
- RabbitMQ 提供了一套严格的通信方式,通信的各个方面几乎都采用了远程过程调用

实战 Python 使用 RabbitMQ 实现发布-订阅模型

gRPC

gRPC 是 Google 发布的基于 HTTP 2.0 传输层协议承载的高性能开源软件框架。

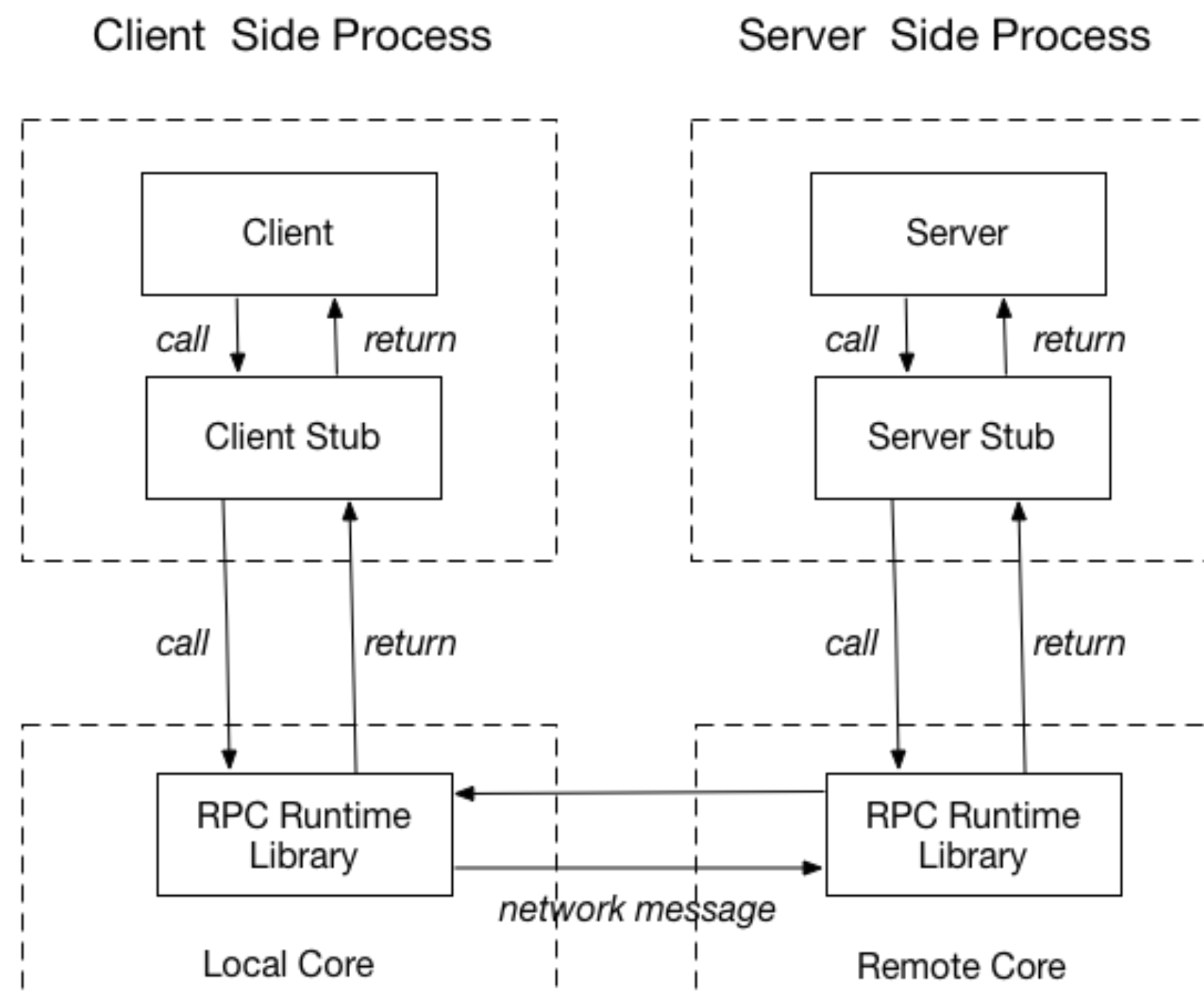
特点：

- 基于 HTTP 协议
- 开源
- 支持多语言
- 通过 Protocol Buffers 编写 proto 文件，交换机与服务器建立 gRPC 通道

gRPC

Protocol Buffers 的作用:

- 定义数据结构
- 定义服务接口
- 序列化和反序列化



Remote Procedure Call Flow

gRPC

实战 Python 和 Golang 之间的 gRPC 通信

总结

1. 掌握缓存机制和 Redis 的用法
2. 掌握消息处理机制和消息队列软件 RabbitMQ
3. 掌握 RPC 消息协议和 gRPC

THANKS! |  极客大学