IT大咖说
知识分享平台

3rd NJSD
Global Software Development Conference . Nanjing
全球软件大会
2017

“千万级”用户架构

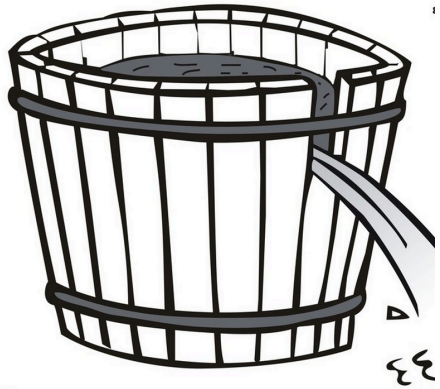
王毅 亚马逊AWS 解决方案架构师， 区域主管，

2017年4月

亚马逊 aws

云计算不是有弹性吗？

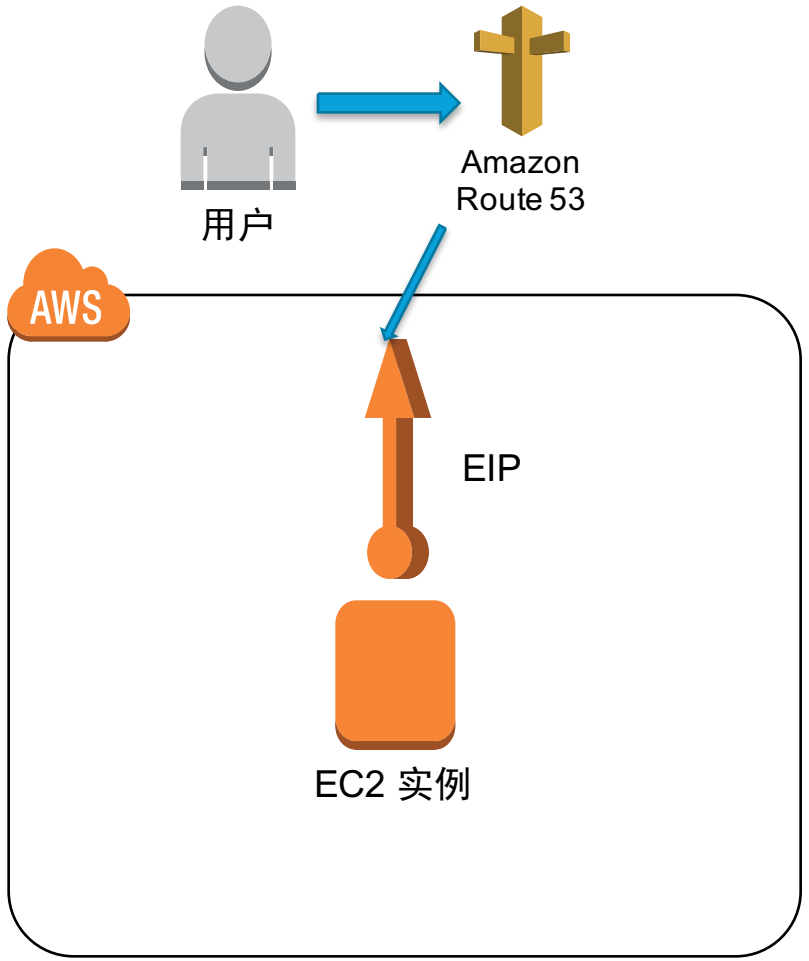
- 弹性是云计算的一个重要特性
- AWS云平台也提供Auto Scaling功能来帮助用户实现弹性伸缩
- 但是，应用服务的弹性伸缩需要良好的设计
  - 应用的架构
  - 使用的服务类型



我们以典型的Web应用为例…

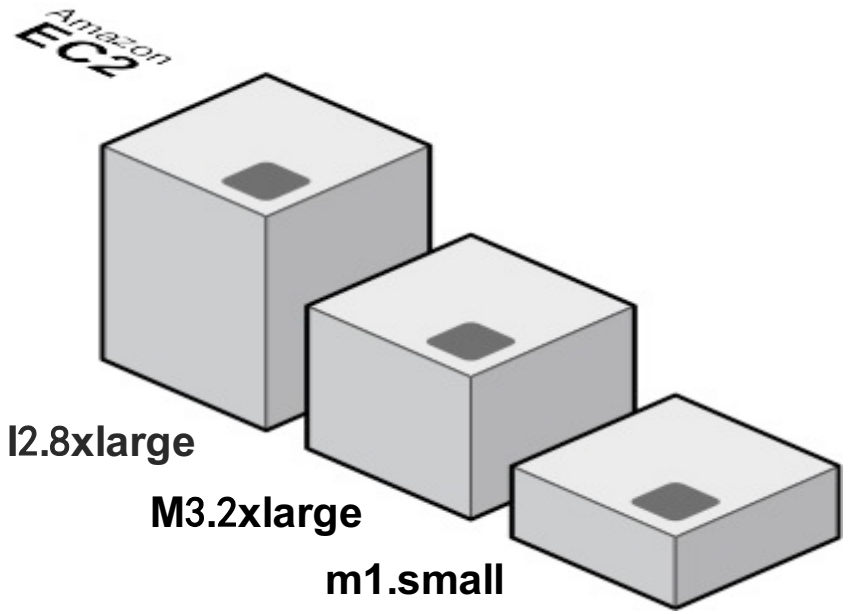
第一阶段：从一个用户开始…

- 一个EC2实例
  - 安装了所有相关软件堆栈
    - Web应用
    - 数据库
    - 管理等
- 一个EIP
- DNS服务：Route53



最简单的扩展：换个更大的机器

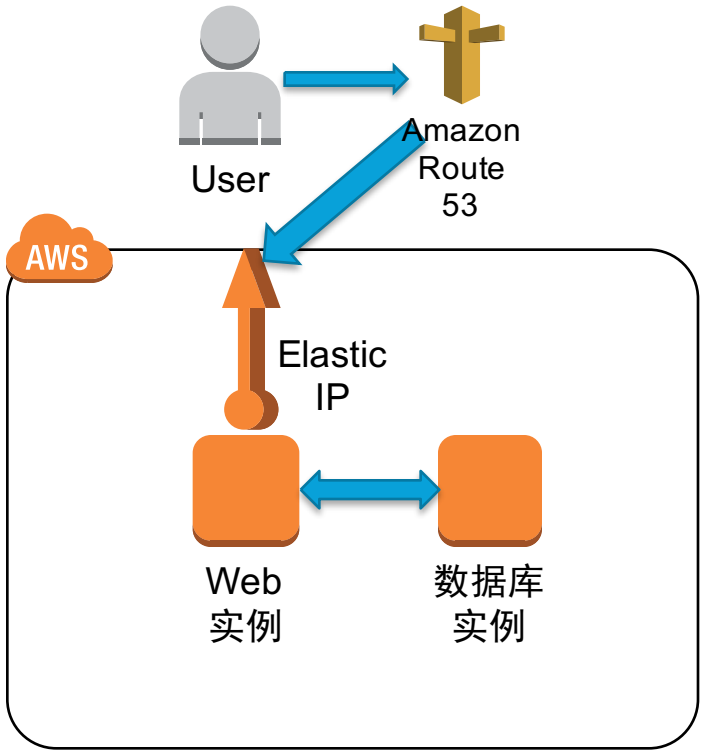
- 可以使用PIOPS
- 高I/O实例
- 高内存实例
- 高CPU实例
- 高存储实例



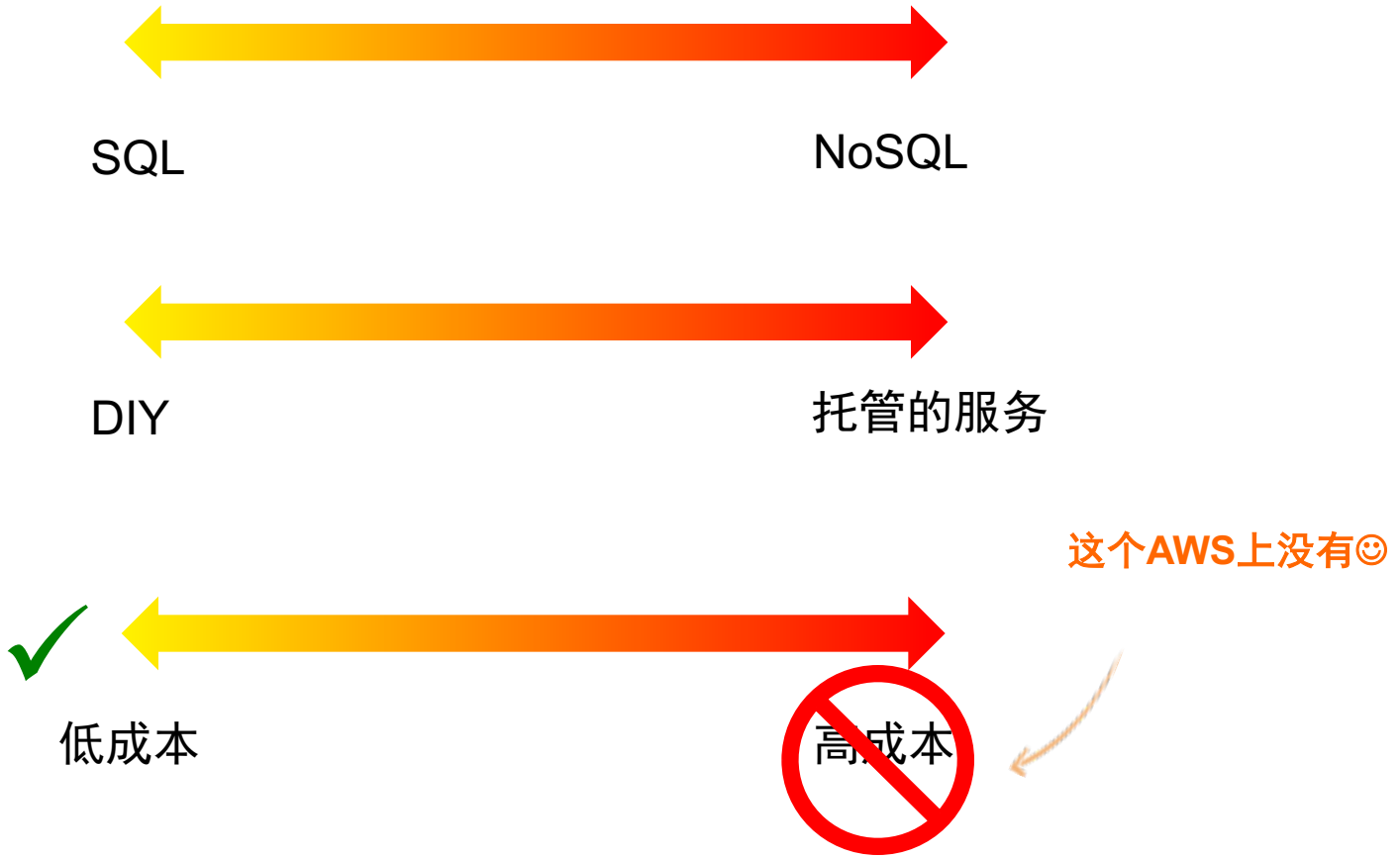
最终都会遇到极限！

第二阶段：多个用户

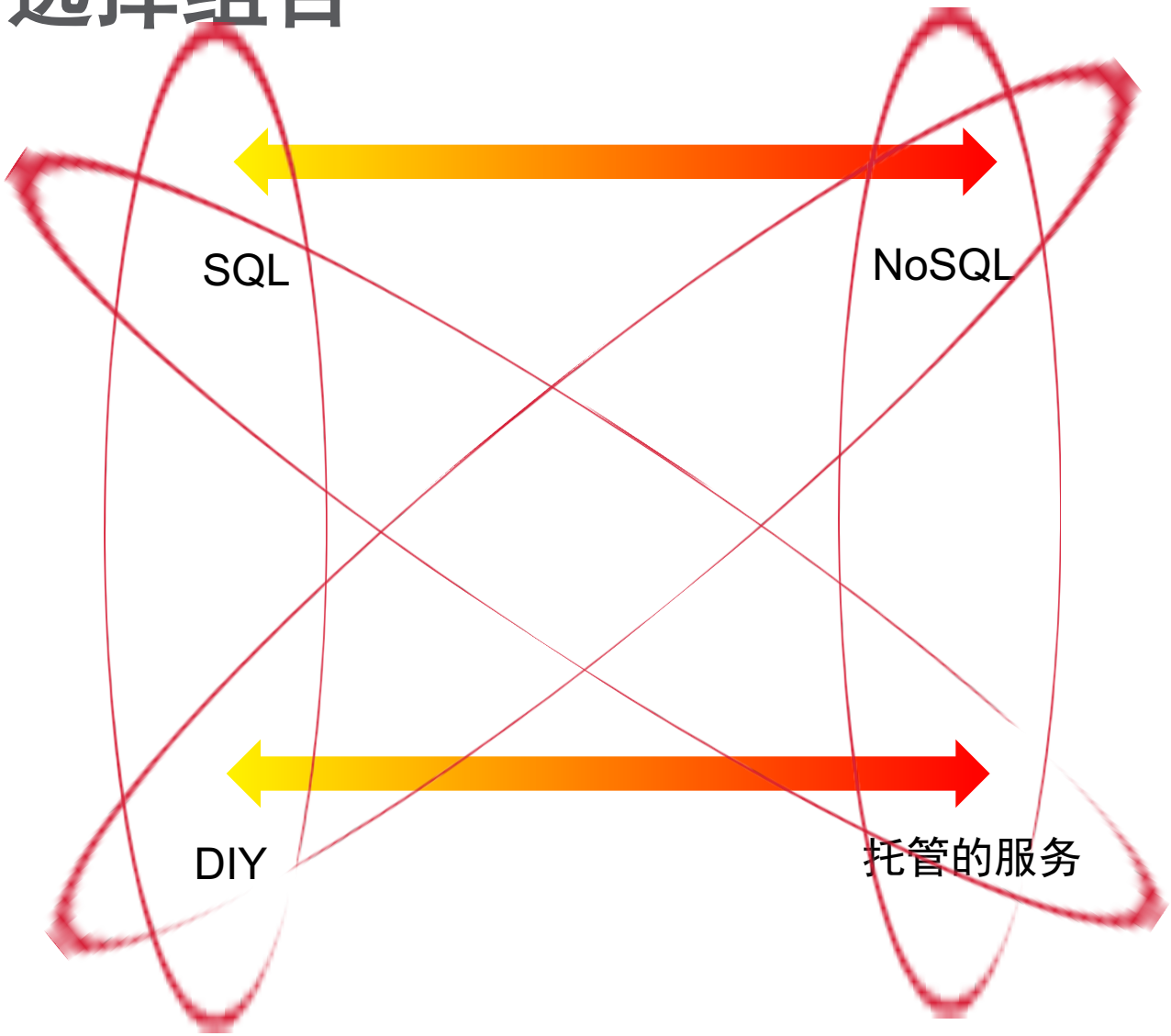
- 首先，把单服务器变成多个
  - Web层
  - 数据库层
- 其次，数据库层的实现方式
  - DIY vs. 管理服务?
  - SQL vs. NoSQL?



几种不同的选择组合



几种不同的选择组合

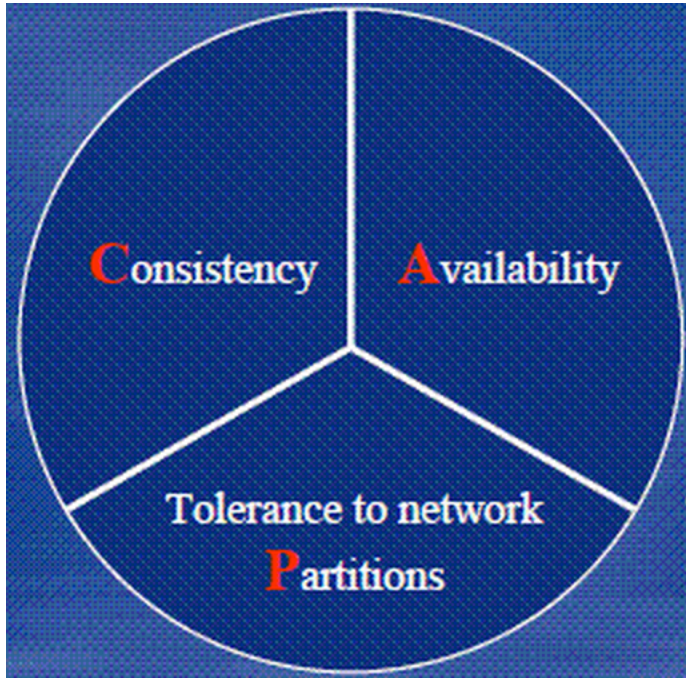


大部分应用建议从SQL开始

- SQL相对成熟，优缺点都比较明确
- 大量工具、已有代码、图书资料、技能等
- 一开始SQL能够满足需求，尤其是百万级用户量内的
- 有一些常用的扩展SQL的方法

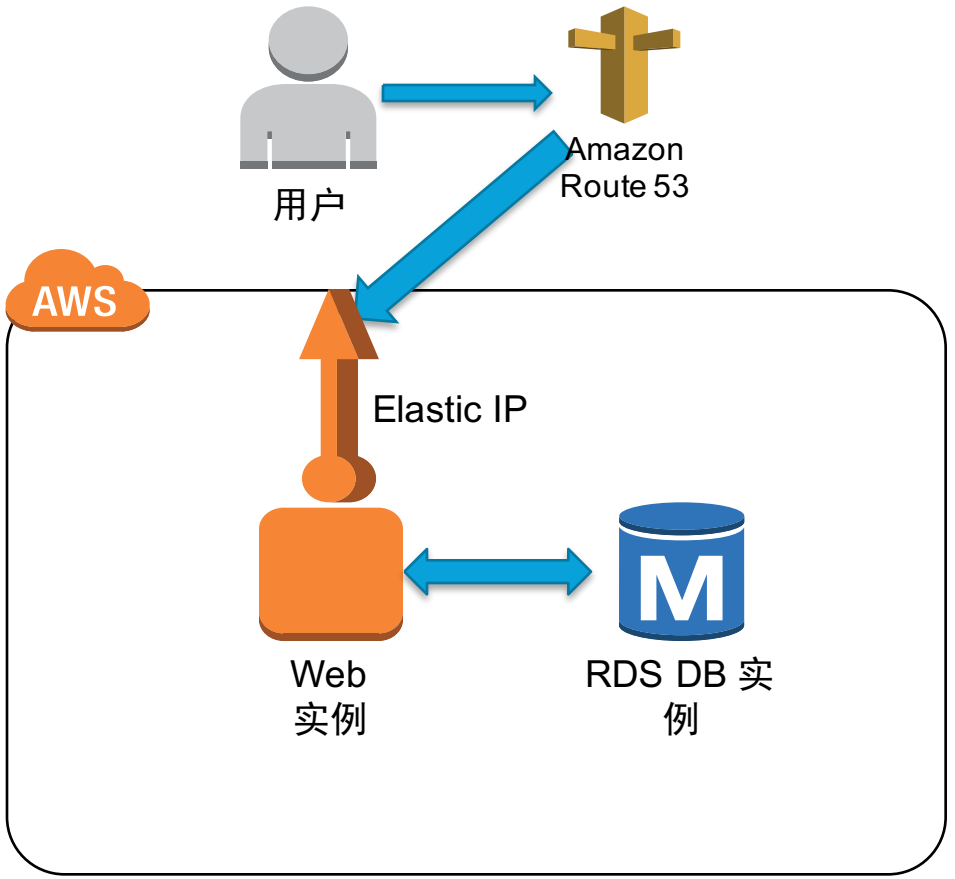
什么时候要考虑使用NoSQL？

- 短期内增长大量数据
  - 比如一年5TB以上
- 应用要求非常低的延时
- 非关系型数据
  - 非结构化数据
  - 半结构化数据
- 快速的数据写入
  - 每秒几千个记录



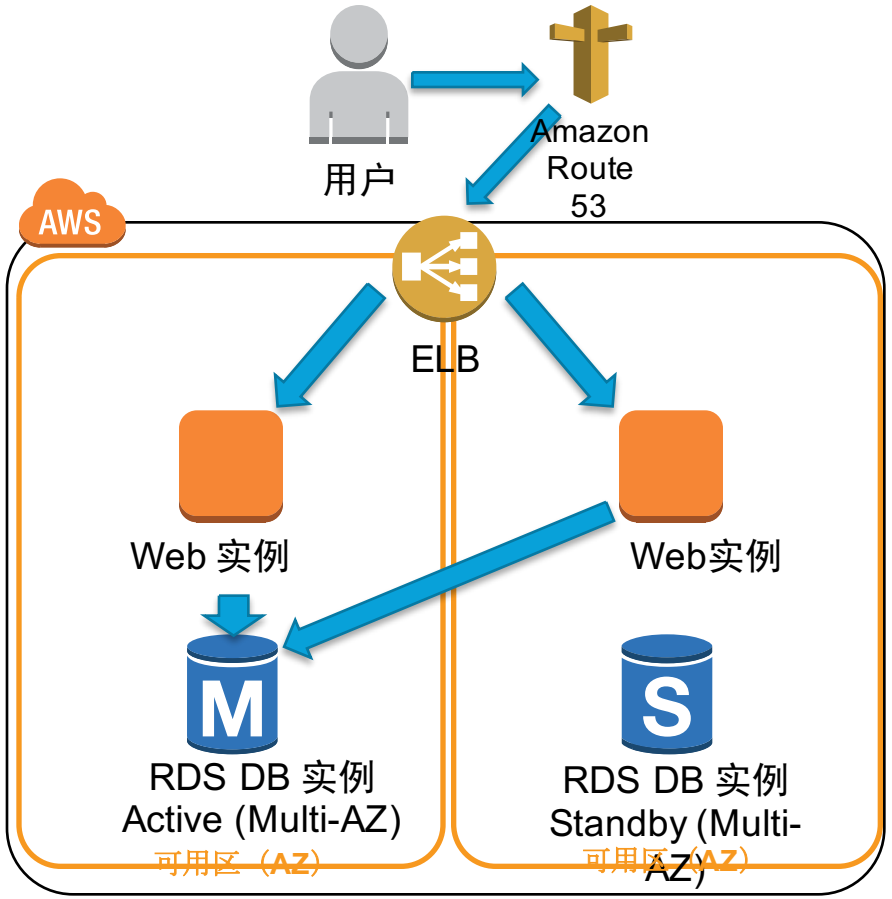
用户数>100

- 数据库层采用RDS服务



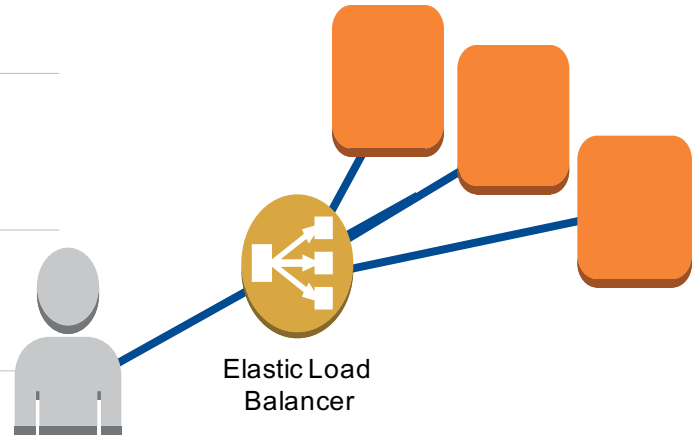
用户数>1000

- 接下来我们处理故障转移和冗余
  - 添加一个Web实例
  - ELB
  - 在另一个可用区（AZ）
- 启用RDS的多可用区部署

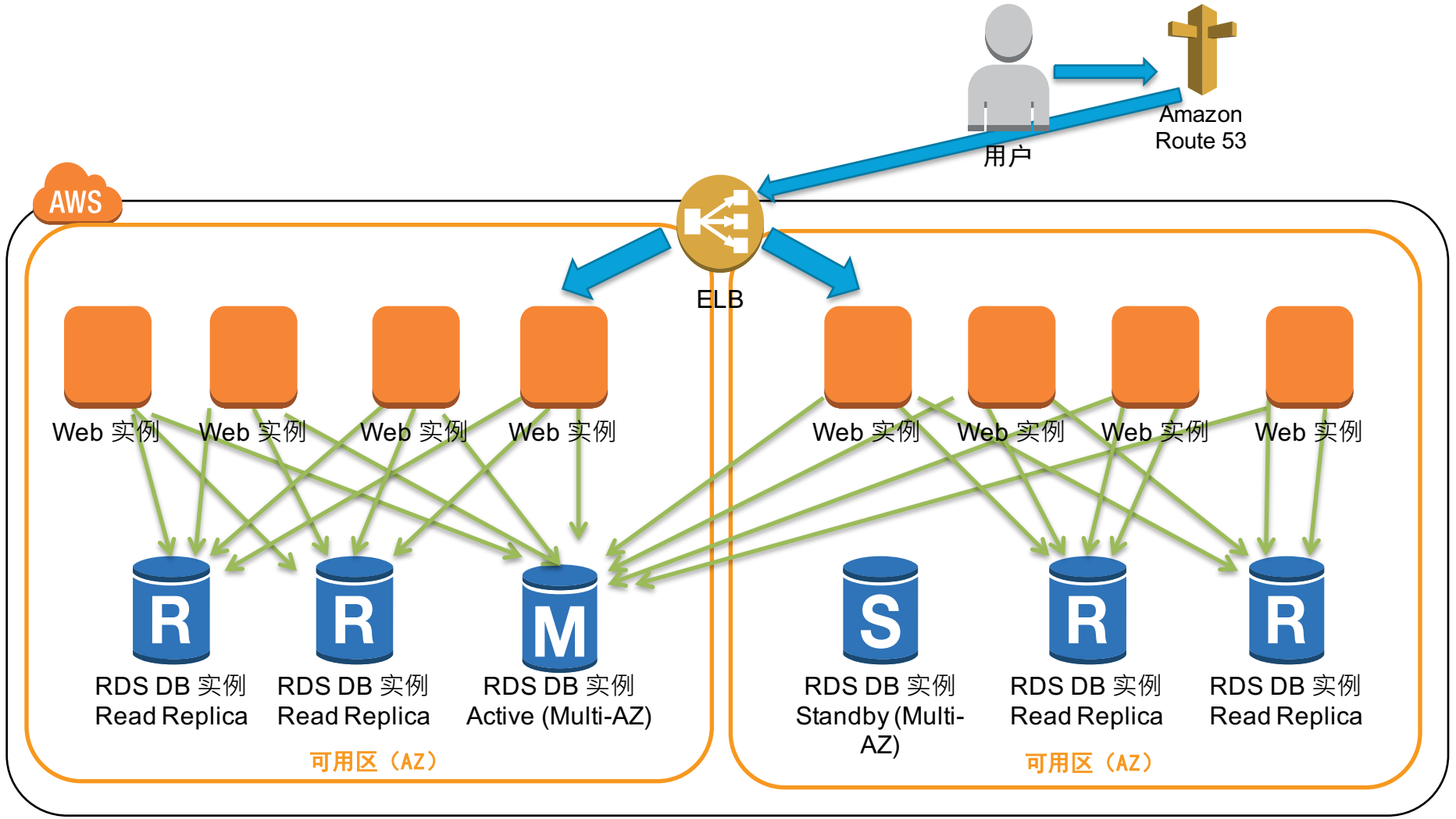


负载均衡服务 —— ELB

特性	细节
可用性	可以为跨可用区（AZ）的实例提供负载均衡服务
健康检查	自动检查实例的健康情况，并根据它来决定分发路由
会话的亲和性/粘性	把请求发送给相同的实例
SSL	支持把SSL负载卸载到ELB上
监控	CloudWatch提供监控指标



一百万级用户规模


用户
Amazon
Route 53
AWS
ELB
Web 实例
Web 实例
Web 实例
Web 实例
Web 实例
Web 实例
Web 实例
Web 实例
R
R
M
S
R
R
RDS DB 实例
Read Replica
RDS DB 实例
Read Replica
RDS DB 实例
Active (Multi-AZ)
RDS DB 实例
Standby (Multi-
AZ)
RDS DB 实例
Read Replica
RDS DB 实例
Read Replica
可用区（AZ）
可用区（AZ）

架构设计小结

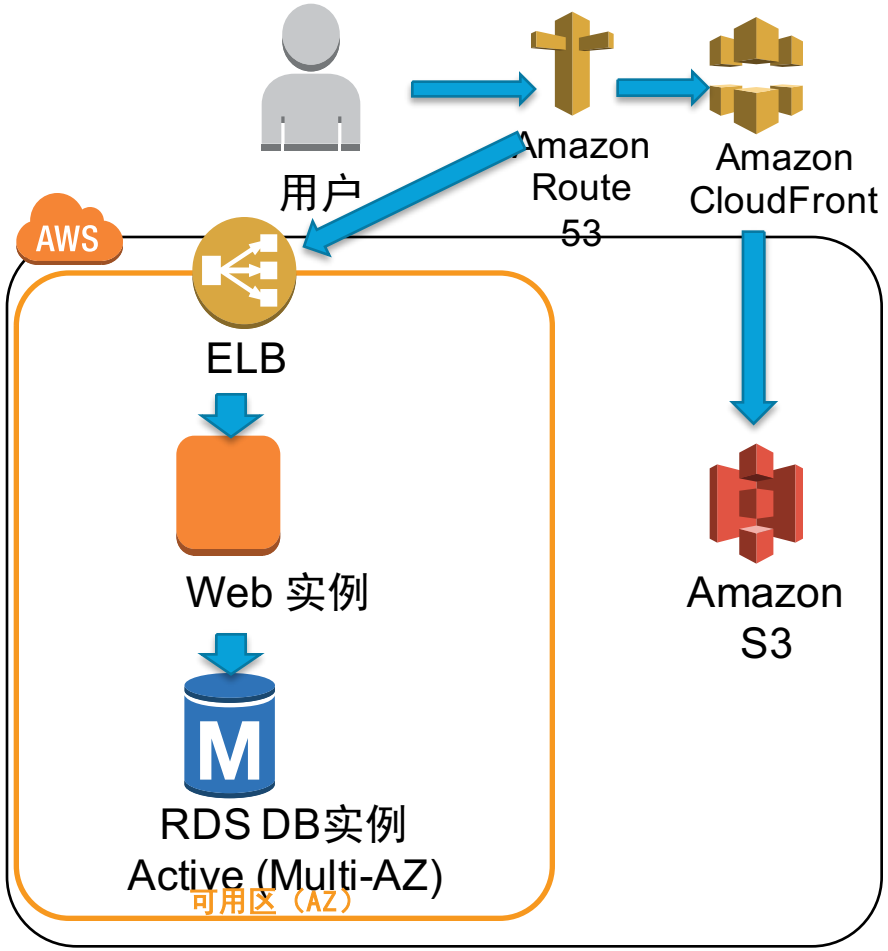
到目前为止，基本是用传统的思路在做应用架构

一般可以应付几十万到一百万用户级别的访问量

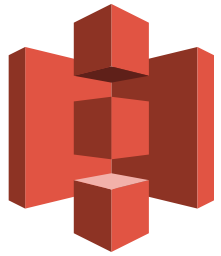
没有充分使用AWS云服务

改变部分工作负载设计方式（一）

- 减少一点Web和数据库实例的压力
- 把静态内容从Web实例中迁移到S3
- 把部分数据通过CloudFront访问



使用S3服务

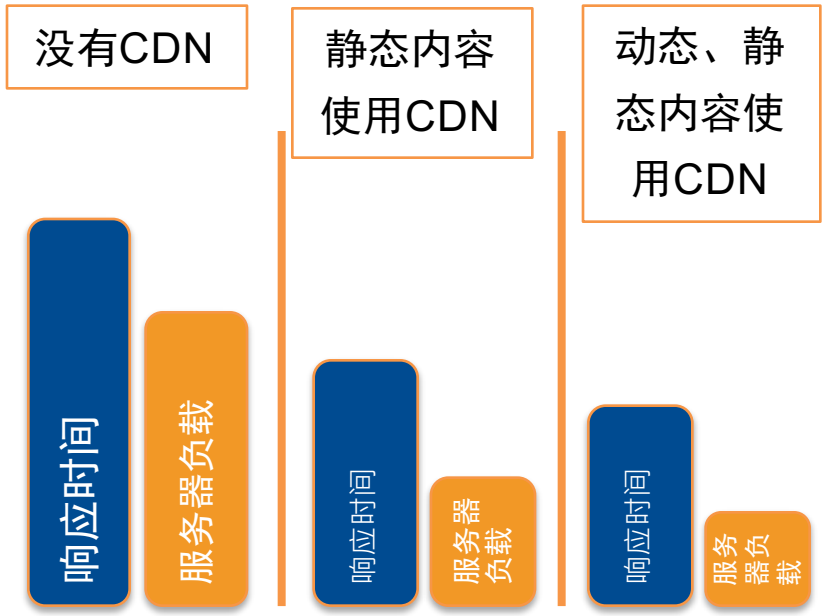


- 针对Web的对象数据存储服务
- 11个9的持久性
- 海量存储
- 适合用来
  - 静态数据（CSS，JS，图片，视频等）
  - 日志
  - 备份
  - 存放处理的文件

- 支持多种访问控制方式
- 支持数据加密
- 与多种服务紧密集成
  - CloudFront
  - EMR
  - Glacier

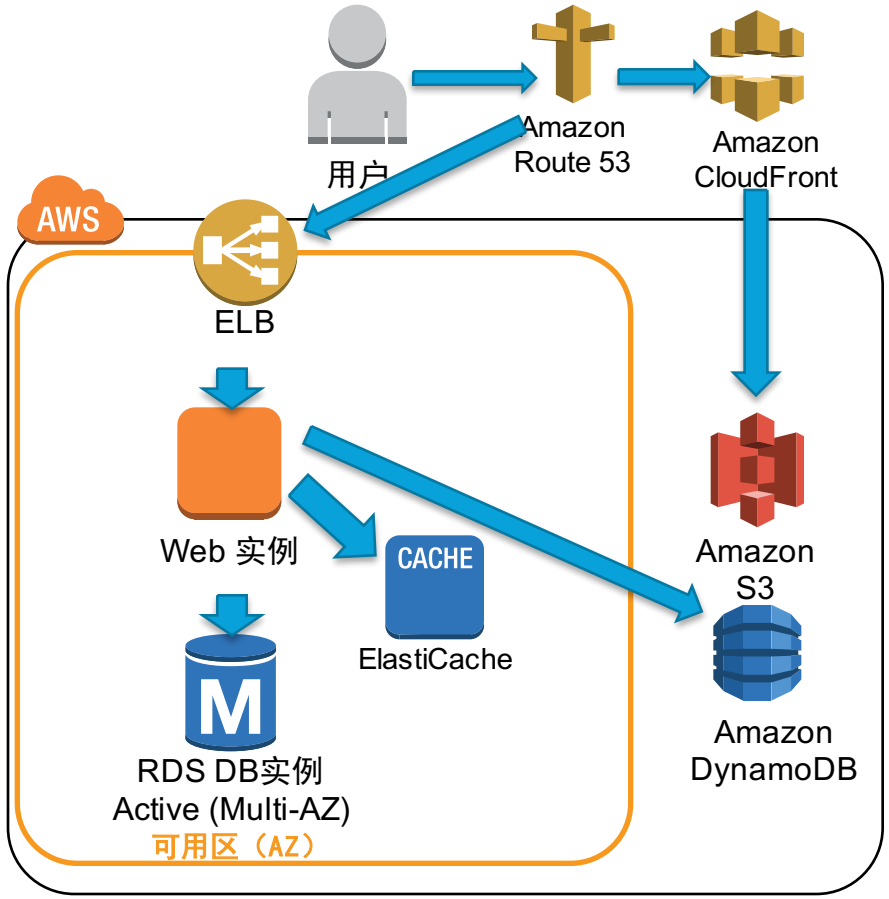
使用CloudFront服务

- 提供Web Service接口的CDN服务
- 支持动态和静态内容
- 支持流式视频
- 支持根域（Zone Apex）
- 支持客户化SSL证书
- 低TTL（最低为0）



改变部分工作负载设计方式（二）

- 减少一点Web和数据库实例的压力
- 使用分布式内存缓存：ElastiCache
- 使用DynamoDB



使用ElastiCache服务

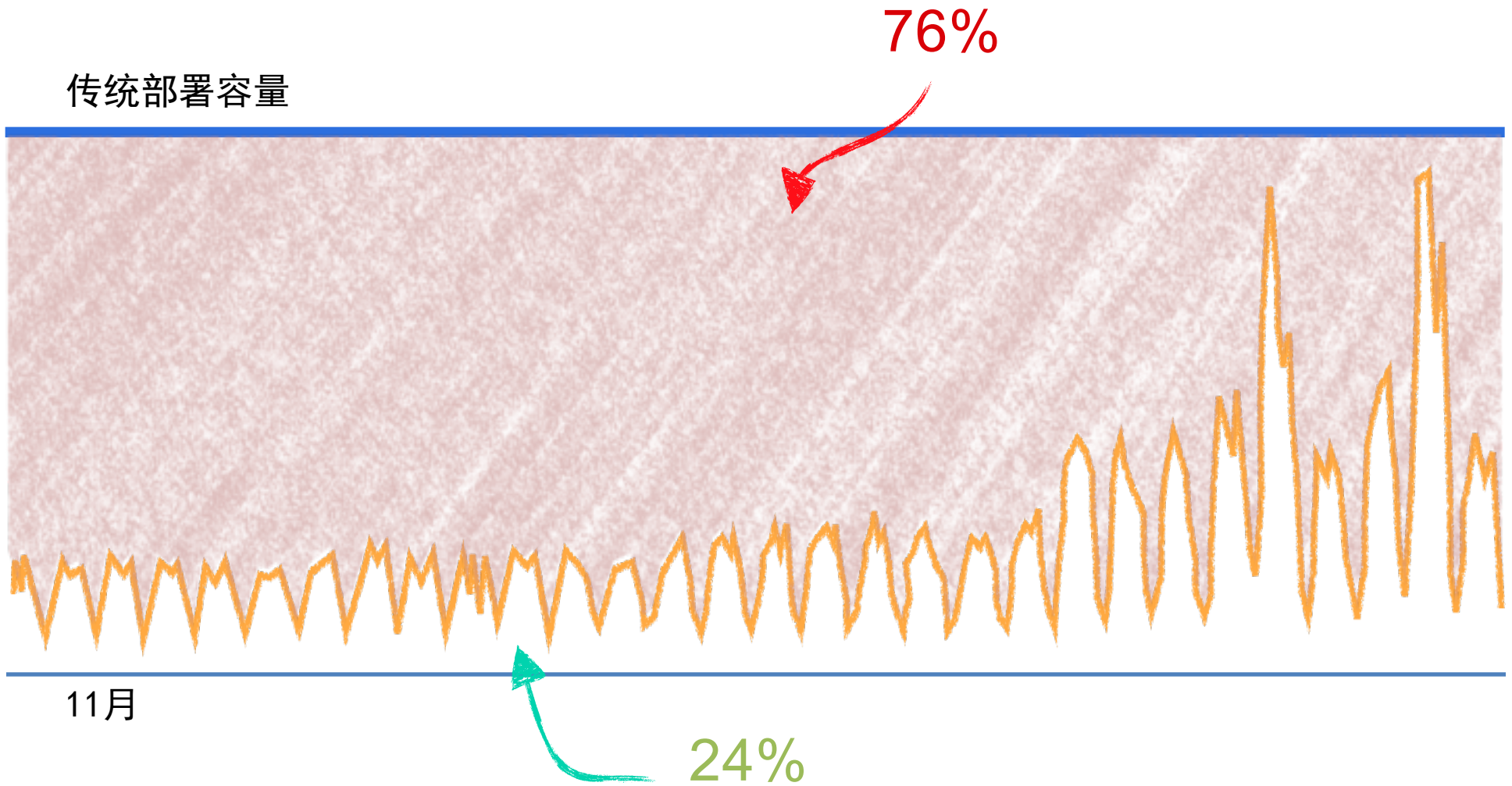
- 托管的Memcached和Redis服务
  - 使用一样的API
- 从一个节点扩展到多个节点
- 自我修复（替换失效节点）
- 非常快（通常是几个毫秒）
- Memcache使用一个AZ，Redis可以跨AZ复制
- 使用AWS的Auto Discovery客户端可以使群集的伸缩与应用透明

使用DynamoDB服务

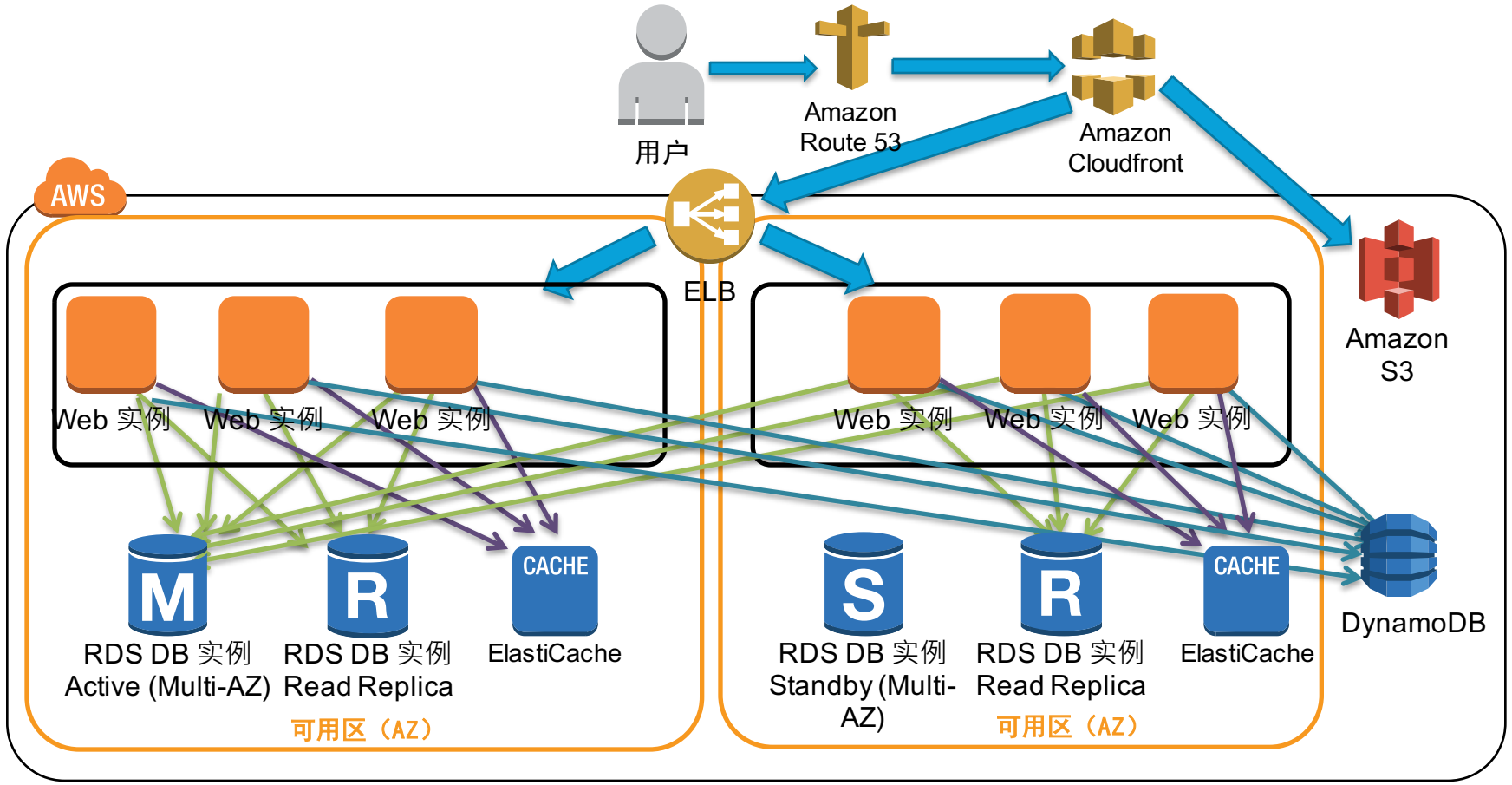
特性	细节
可配置的吞吐量	配置读操作性能，写操作性能
可预测的性能	后台存储基于SSD构建，平均延时为几个毫秒
强一致性	确保你读取的数据是最新的
故障冗余	跨可用区的数据复制
监控	与CloudWatch集成
安全	与IAM（Identity and Access Management）集成
Elastic MapReduce	与EMR集成做大数据集的复杂数据分析

给我们的应用加点弹性…

Amazon.com 11月份的流量



百万级用户规模


用户
Amazon
Route 53
Amazon
Cloudfront
AWS
ELB
Amazon
S3
Web 实例
Web 实例
Web 实例
Web 实例
Web 实例
Web 实例
M
R
CACHE
S
R
CACHE
DynamoDB
RDS DB 实例
Active (Multi-AZ)
RDS DB 实例
Read Replica
ElastiCache
RDS DB 实例
Standby (Multi-
AZ)
RDS DB 实例
Read Replica
ElastiCache
可用区（AZ）
可用区（AZ）

工欲善其事，必先利其器

- 管理指标和告警设置
- 自动化构建
- 自动化部署
- 集中化日志管理

部署和管理服务

AWS Elastic Beanstalk

自动化资源管理 – 简化Web应用部署

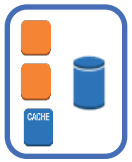
AWS OpsWorks

应用程序生命周期管理和自动化部署的框架

AWS CloudFormation

基于模板的资源部署和管理

Web App

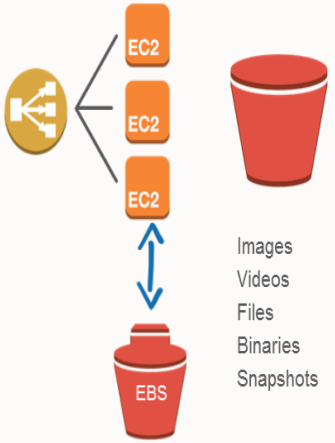


Enterprise App

Databas e

DIY

自己客户化管理和部署: EC2, S3, 定制AMI等

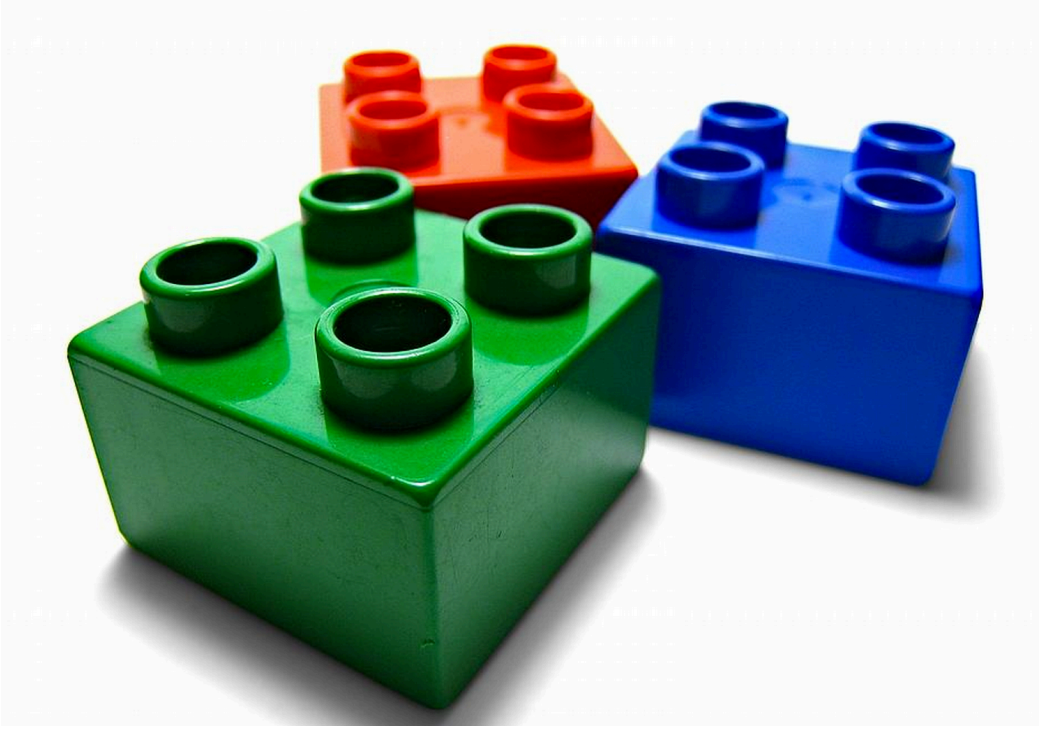


方便 ←→ *控制*

前面架构设计的小结

- 用了多种AWS基础服务
- 充分利用了多个层次的缓存服务
- 利用AWS的管理服务
- 但是架构上没有做好千万级用户量的准备

面向服务的架构设计（SOA）

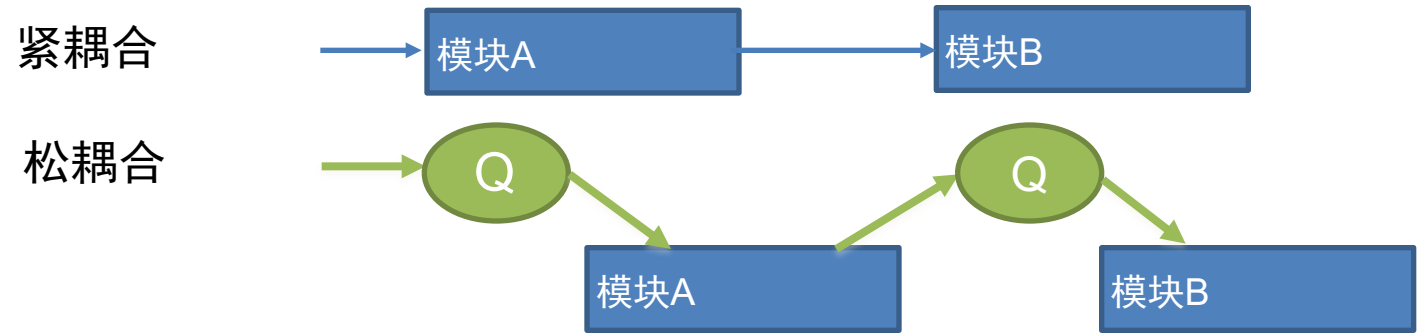


- 把服务构建成为不同的层次或模块，相互比较独立并且可以各自扩展。

松耦合带来灵活性

- 越松的耦合，越高的扩展性
  - 使用相对独立的模块服务
  - 把模块设计成为黑盒子
  - 把交互设计成为松耦合方式
  - 尽量使用自带冗余和扩展性的服务，而不是自建

使用SQS作为一个缓冲



不再重新发明轮子

Amazon SNS

Amazon CloudSearch

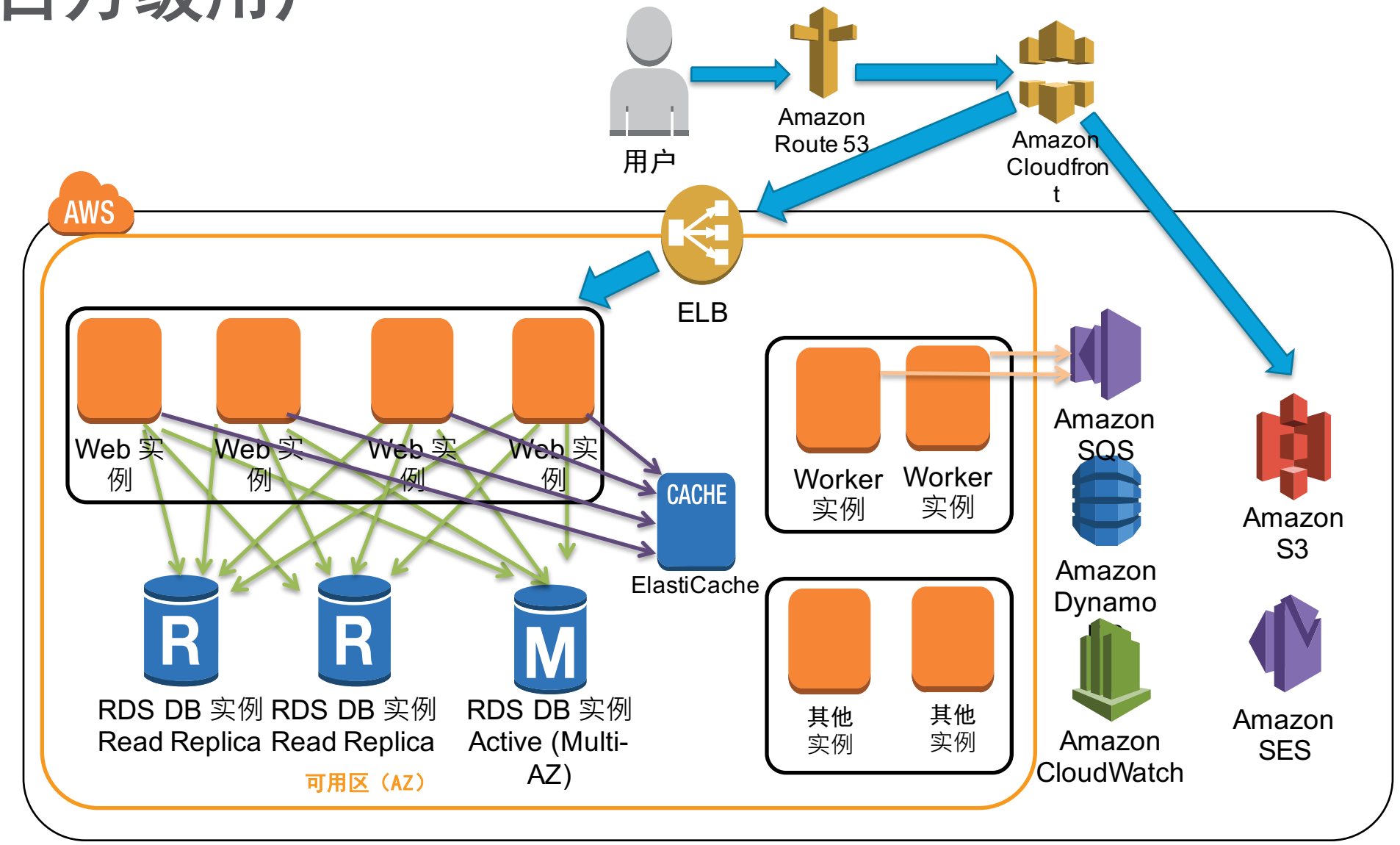
Amazon SQS

Amazon SES

Amazon SWF

Amazon Elastic Transcoder

超过百万级用户


用户
Amazon Route 53
Amazon Cloudfront
AWS
ELB
Web 实例
Web 实例
Web 实例
Web 实例
CACHE
ElastiCache
Worker 实例
Worker 实例
其他 实例
其他 实例
R
R
M
RDS DB 实例 Read Replica
RDS DB 实例 Read Replica
RDS DB 实例 Active (Multi-AZ)
可用区（AZ）
Amazon SQS
Amazon Dynamo
Amazon CloudWatch
Amazon S3
Amazon SES

从5百万到千万用户级别的挑战

- 数据库的性能挑战
  - 联邦（根据功能把数据库分成多个库）
  - 分表/分片（把数据库表分成多个，或多个服务器上）
  - 把部分数据迁移至NoSQL（如常用的/“热”表，大量快速写数据等）
- 应用的性能挑战
  - 确保有监控、报警等服务
  - 及时获取客户的反馈
  - 逐步调优各个模块服务

总结

- 良好的云架构设计
- 充分使用AWS上能够自扩展、自冗余的服务（ELB，S3，SNS，SQS，SES等）
- 采用不同的缓存技术
- 综合使用SQL和NoSQL数据库
- 自动化管理基础设施
- 不要重新发明轮子
- SOA，SOA，SOA…

参考资料

- AWS官网
  - http://aws.amazon.com
- AWS参考架构
  - http://aws.amazon.com/architecture/
- AWS白皮书
  - http://aws.amazon.com/whitepapers/
- AWS英文博客
  - http://aws.typepad.com
- 在线动手实验
  - http://run.qwiklab.com/

谢谢！

中文站点
www.amazonaws.cn

微博
weibo.com/amazonaws

中文博客
blog.csdn.net/awschina

微信
AWS 中国