视频流系统设计（Youtube / Netflix）

主讲人 南帝老师

自我介绍

过去

外企、国企、互联网

10+年后端研发架构经验

面试官100+

现在

阿里巴巴

研发、架构师

面试官30+

未来

To be continued

课程介绍

九章算法
www.jiuzhang.com

	• 【互动】走进系统设计 & 新鲜事系统设计 • 【直播】秒杀系统与订票系统设计
Week1	• 【互动】从用户系统中学习数据库与缓存 • 【互动】网站系统、API设计与短网址 • 【直播】容器技术（K8S，Docker）
Week2	• 【互动】数据库拆分 & 限流器和实时数据系统设计 • 【互动】分布式数据库系统设计 • 【直播】在线文档协同编辑系统设计
Week3	• 【互动】聊天系统设计 • 【互动】分布式文件系统设计 • 【直播】视频流系统设计（Youtube / Netflix）
Week4	• 【互动】基于地理位置的信息系统 • 【互动】分布式计算系统设计 • 【互动】爬虫系统与搜索建议系统 • 【直播】Google Search / Search Ads Ranking 系统设计

Keywords

Scenario场景

设计一个YouTube

1.了解YouTube包含哪些功能

2.这些功能背后的挑战

YouTube是一个全球最大的视频分享网站之一

YouTube的功能点

Functional Requirements

用户可以上传视频
用户可以观看和分享视频
用户可以根据关键字搜索视频
用户可以点赞、评论

Non-Functional Requirements

系统的高可用
观看视频的流畅度
视频的实时推荐

YouTube关键数据

每月有效的用户总数：19亿
每日活跃的用户总数：3000多万
分享的视频总数：5亿多
平均观看时间：40分钟
每天观看视频数量：50亿
每日移动观看次数：5亿
每分钟上传视频数量：300小时

---截止18年6月

容量评估

•根据数据，我们有30M的日常活动用户，如果用户平均每天观看10个视频，则每秒视频的观看数为：
30M * 10 / 86400 秒 = 3472 视频/秒

•假设每上传一个视频，我们就会观看100部视频，则每秒视频的上传数为：
3472 / 100 = 34 视频/秒

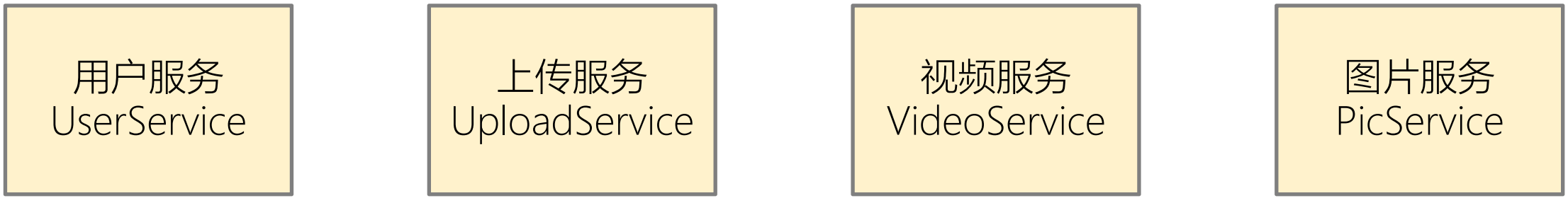
•假设平均每个视频时长为10分钟，则每秒上传的视频时长为：
34 * 10 = 340 分钟/秒

•假设在平均的情况下，一分钟视频需要50MB的存储空间（视频需要以多种格式存储），则一秒钟内上传的视频所需的总存储量为：
340 * 50 = 17000 MB/s = 17 GB/s

Service服务

YouTube相关服务

1.了解YouTube包含哪些服务



查询用户基本信息

视频上传

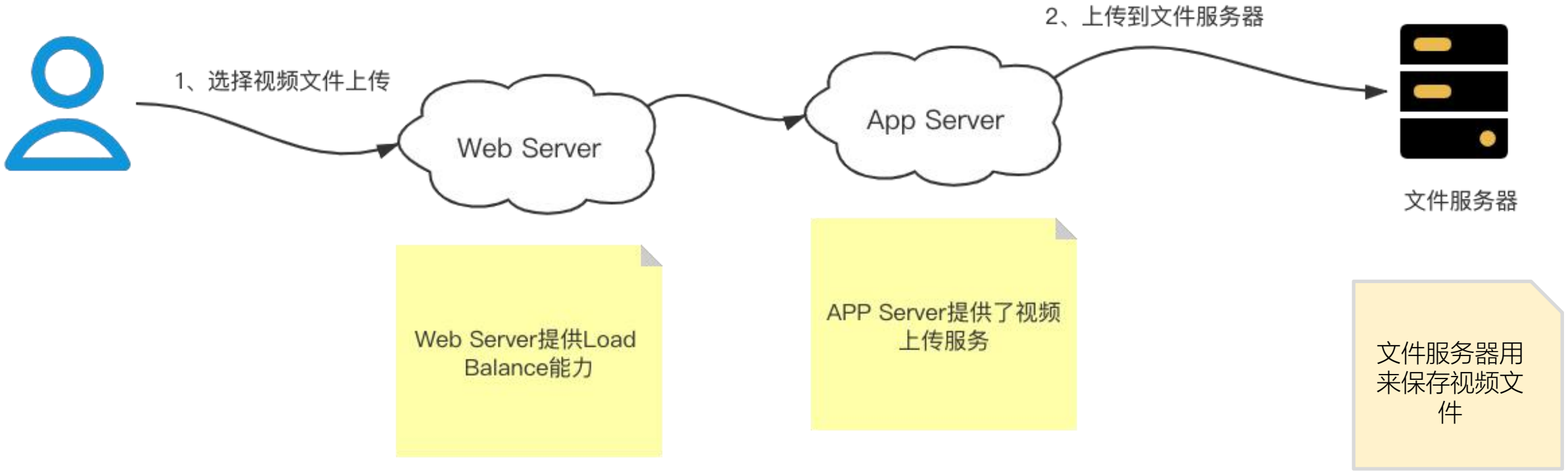
视频切分
视频编码

缩略图生成

Storage存储

模型的设计

1.YouTube模型定义



这个流程有什么问题?

文件太大，中途HTTP连接中断

断点续传如何实现？

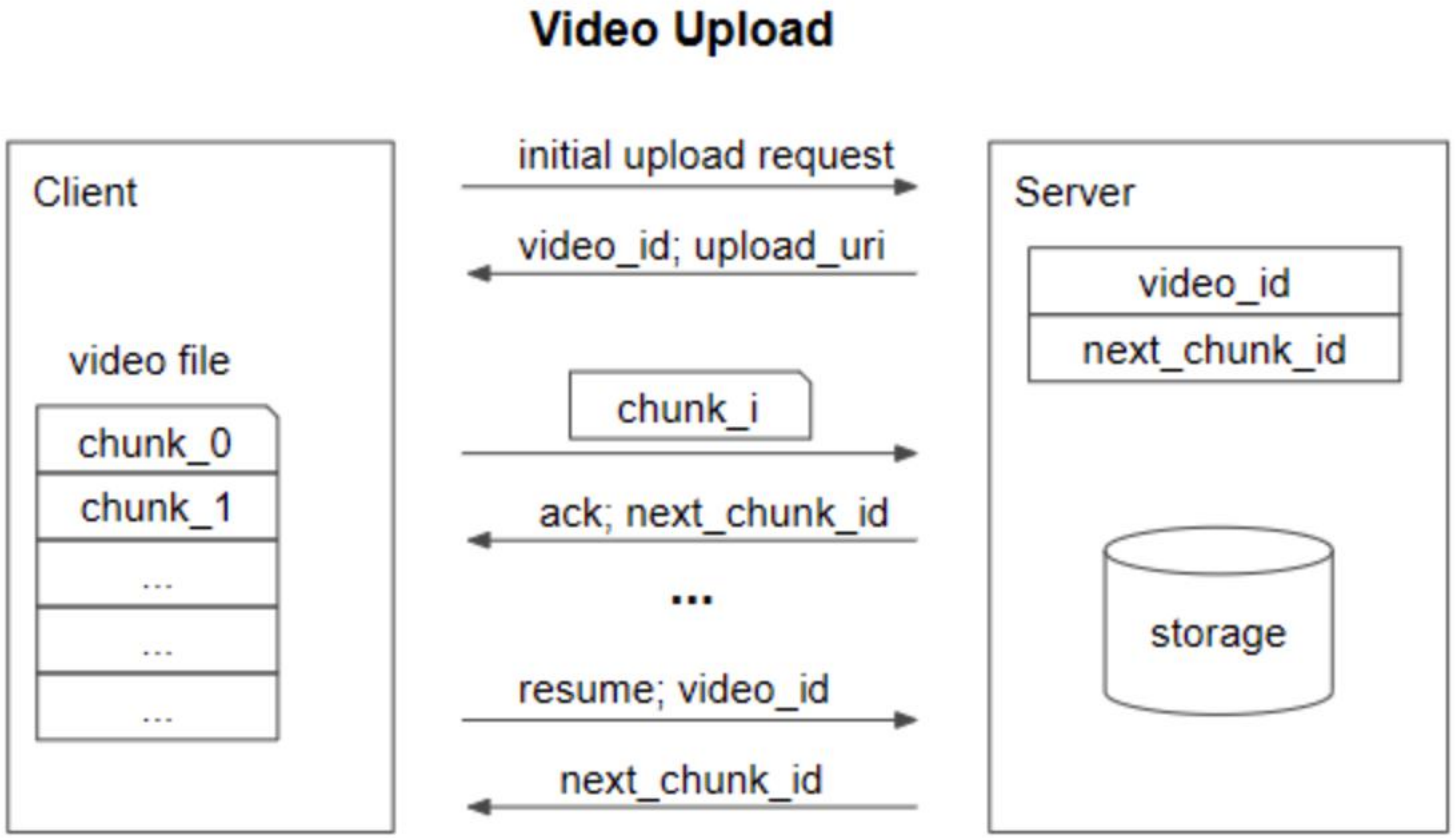
上传视频流程

upload request

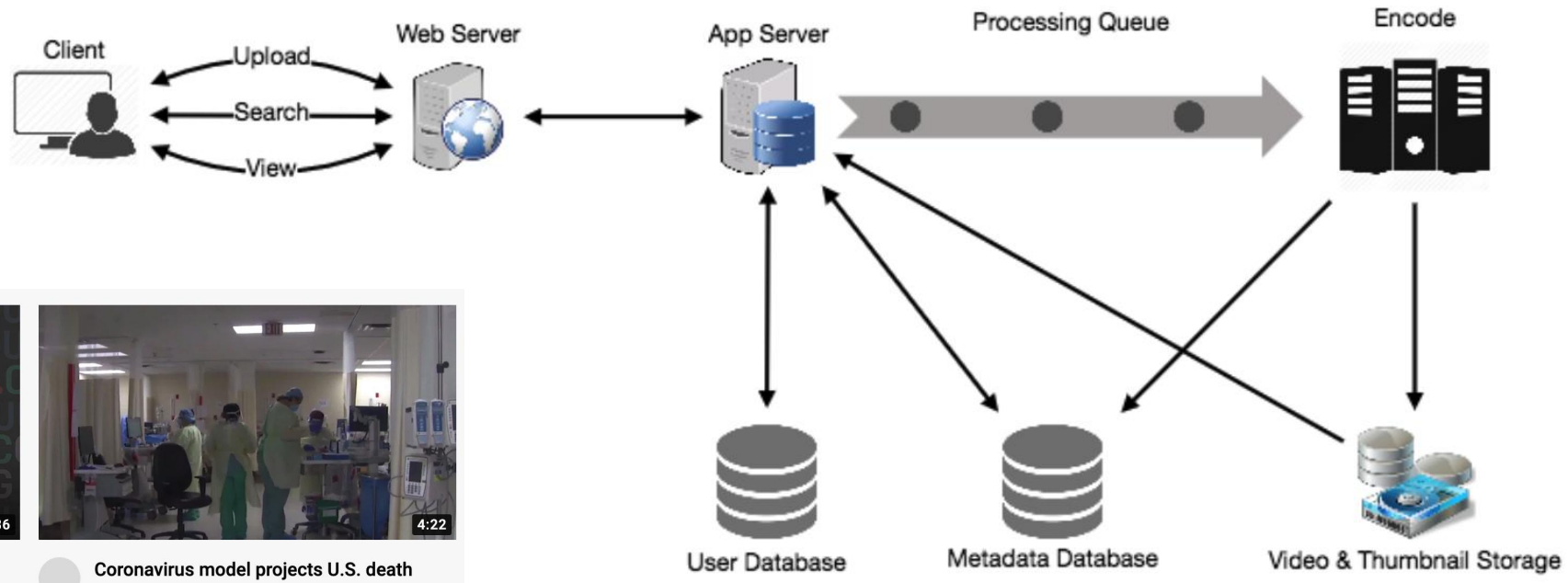
- •请求服务端生成 video_id & 保存目录
- •服务端返回 video_id & 目录
- •客户端根据 video_id 顺序生成 chunk_id 以及 chunk文件，chunk_id 可以采用 video_id + 'xxx' 的形式
- •客户端才真正开始上传 chunk 视频

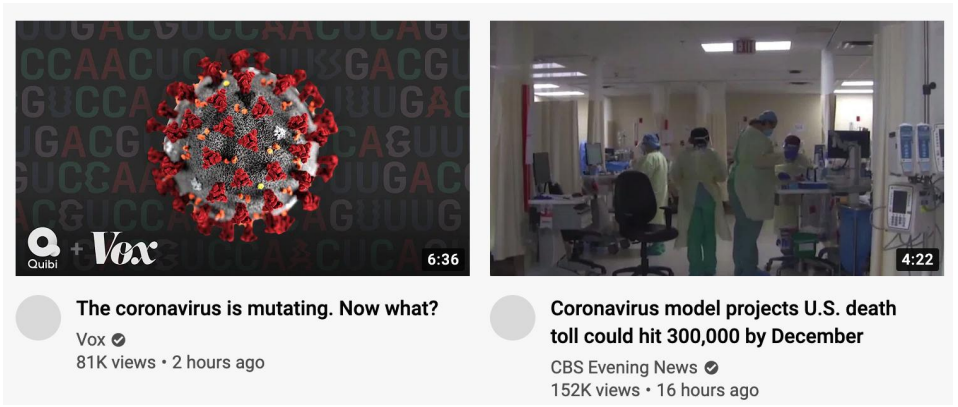
PS：

- •当上传完一个 chunk 之后，服务端告诉客户端需要传下一个 chunk



Processing Queue：上传的视频存放队列
Encode：转码服务、缩略图生成服务
Metadata Database：存储的是视频的基本信息，比如Title、缩略图路径等等





Storage

用户表	用户ID	用户名	性别
	U0001	南帝	男

Storage

视频表

视频ID	视频名	存储目录	Hash值	metadata	格式	缩略图目录	大小	总时长

V0001	YouTube入门.mp4	/xxxx/	zzzzzz	{
				title:xxx
				language:xxx
				tags:xxx
				category:xxx
				}
					1080p	/mmm	1G	120

Chunk视频表

VideoID	chunkID	start_time	end_time	chunk storage	resolution
V0001	C0001	10:00	11:00	/0001/	1080p
V0001	C0002	11:01	12:00	/0001/	1080p
V0001	C0001	10:00	11:00	/0001/	720p

缩略图表

VideoID	缩略图路径	大小	主图
V0001	/mmmm/01	5k	1
V0001	/mmmm/02	5k	0

用户视频关系表

UserID	VideoID	Upload TS
U0001	V0001	2020/08/05 09:00:00

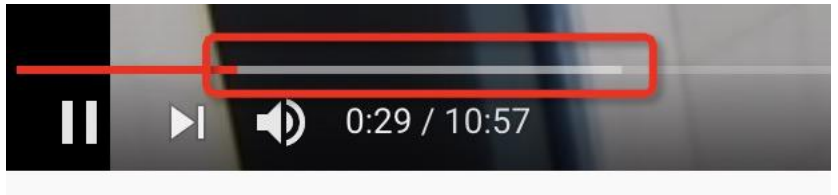
视频表为什么需要hash列?

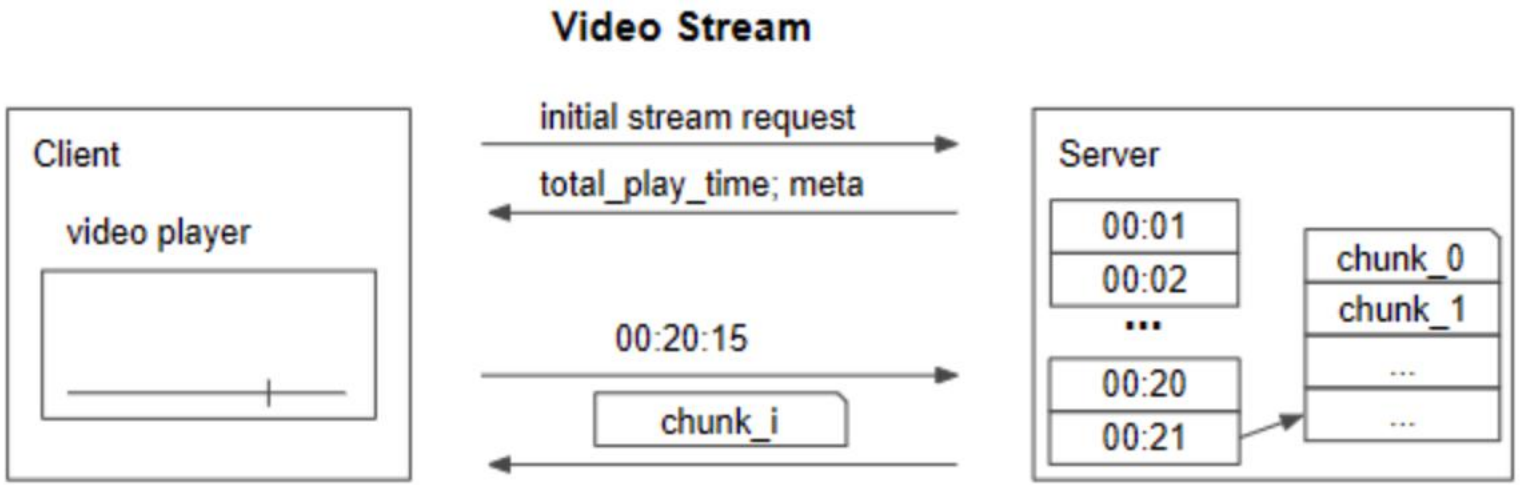
为了视频去重，避免上传重复的视频文件

那么当另一个用户上传了一个相同的视频之后，会发生什么？

会在用户视频关系表插入一条Record

如何做到边看边加载?





那这些chunks和缩略图保存在哪里呢?

- 分布式文件存储

哪些分布式存储适合保存Chunks和缩略图?

- seaweedFS
- FastFS

为什么适合保存这些?

- Chunks和图片Size一般偏小
- 读取频率较高，点击YouTube先看到的都是缩略图

分布式文件存储比较

指标	适合类型	文件分布	系统性能	复杂度	FUSE	POSIX	备份机制	通讯协议接口	社区支持	去重	开发语言
FastDFS	4KB~500MB	小文件合并存储不分片处理	很高	简单	不支持	不支持	组内冗余备份	API/http	国内用户群		C语言
TFS	所有文件	小文件合并，以block组织分片		复杂	不支持	不支持	Block存储多份,主辅灾备	API/http	少		C++
MFS	大于64K	分片存储	Master占内存多		支持	支持	多点备份动态冗余	使用fuse挂在	较多		Perl
HDFS	大文件	大文件分片分块存储		简单	支持	支持	多副本	原生api	较多		java
Ceph	对象文件块	OSD一主多从		复杂	支持	支持	多副本	原生api	较少		C++
MogileFS	海量小图片		高	复杂	可以支持	不支持	动态冗余	原生api	文档少		Perl
ClusterFS	大文件			简单	支持	支持	镜像		多		C

Scale扩展

如何做到精准实时推荐

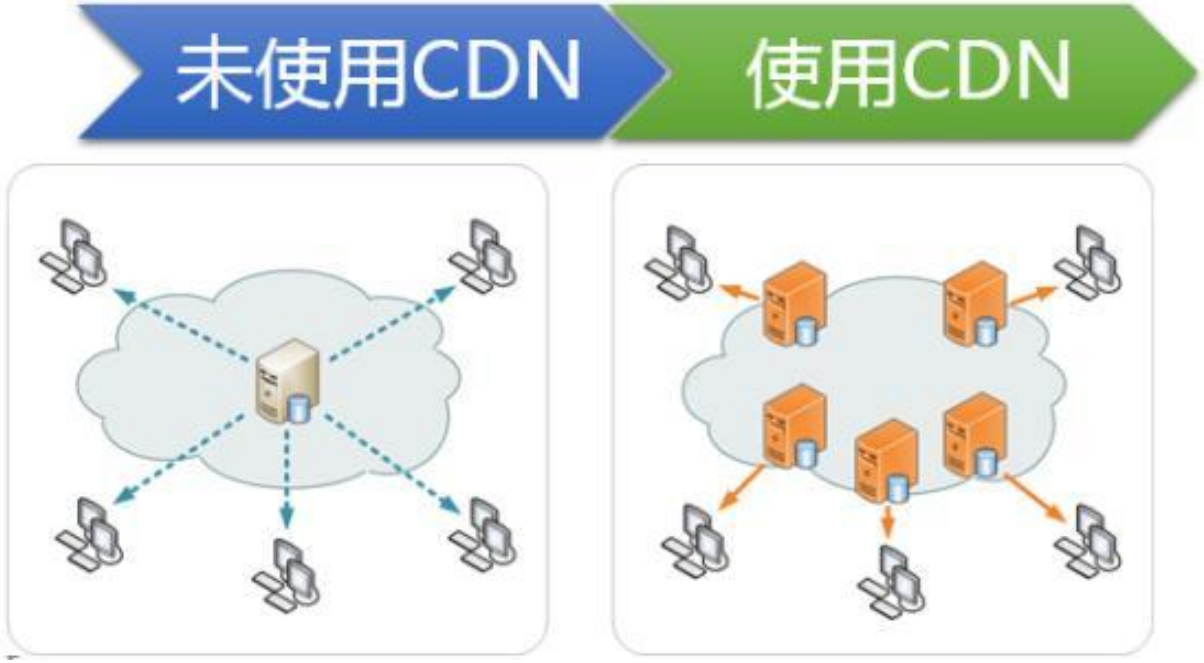
1.常见的推荐算法介绍

2.流处理介绍

不同地区的客户端如何更快的观看视频?

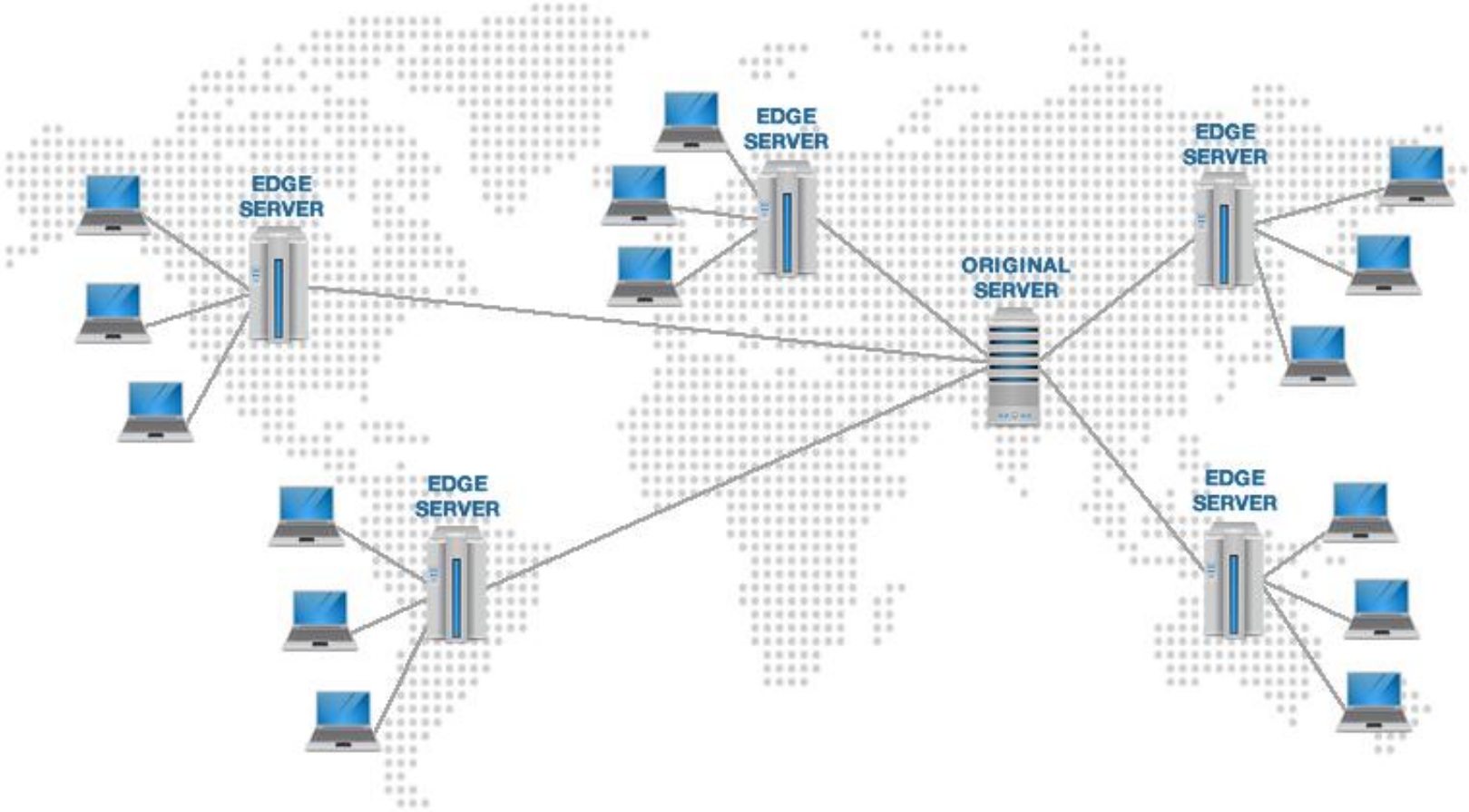
什么是CDN?

CDN是接近用户地理位置的边缘服务器，可以作为缓存处理



CDN

- 80%的流量都是20%的视频贡献的
- YouTube会选择把热点视频定时同步到CDN
- 用户可以在就近的服务器访问到热点视频，无须请求中央服务器，加快加载速度



如何给数据库做Sharding?

- 用户视频关系表以用户ID做Sharding
- 视频表、Chunk表以视频ID做Sharding

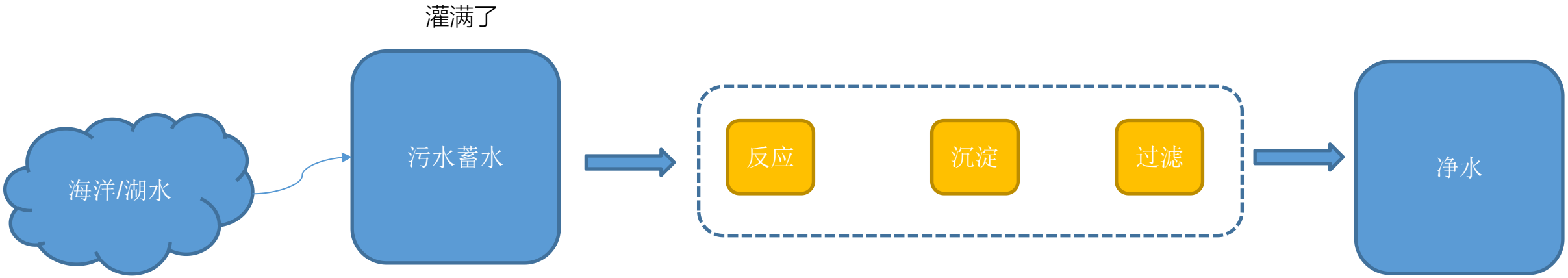
如何准实时推荐视频?

哪些场景需要准实时?

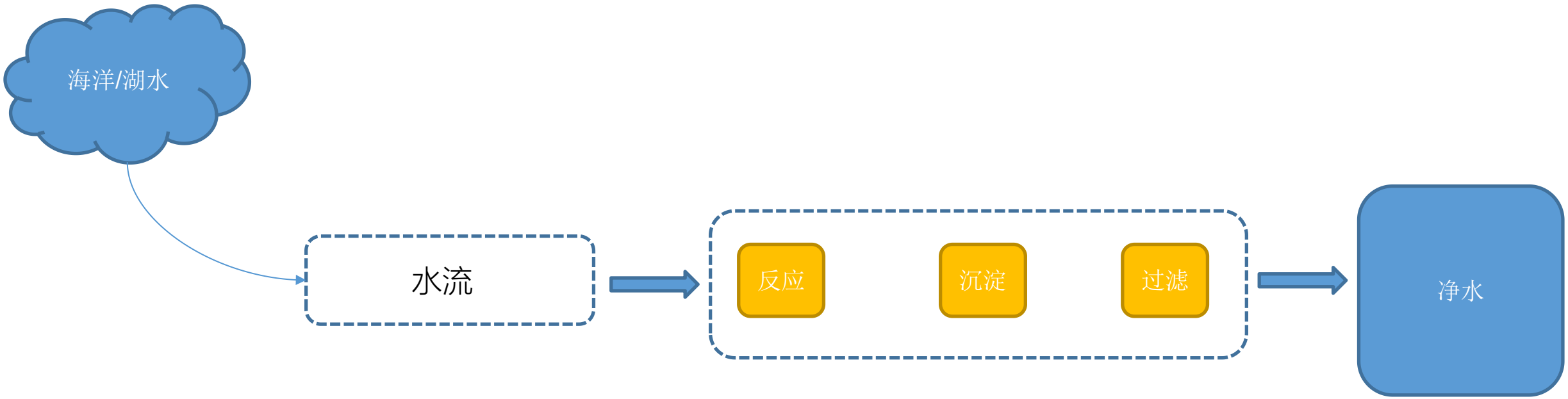
比如抖音，需要及时感知用户的偏好

流式Streaming处理

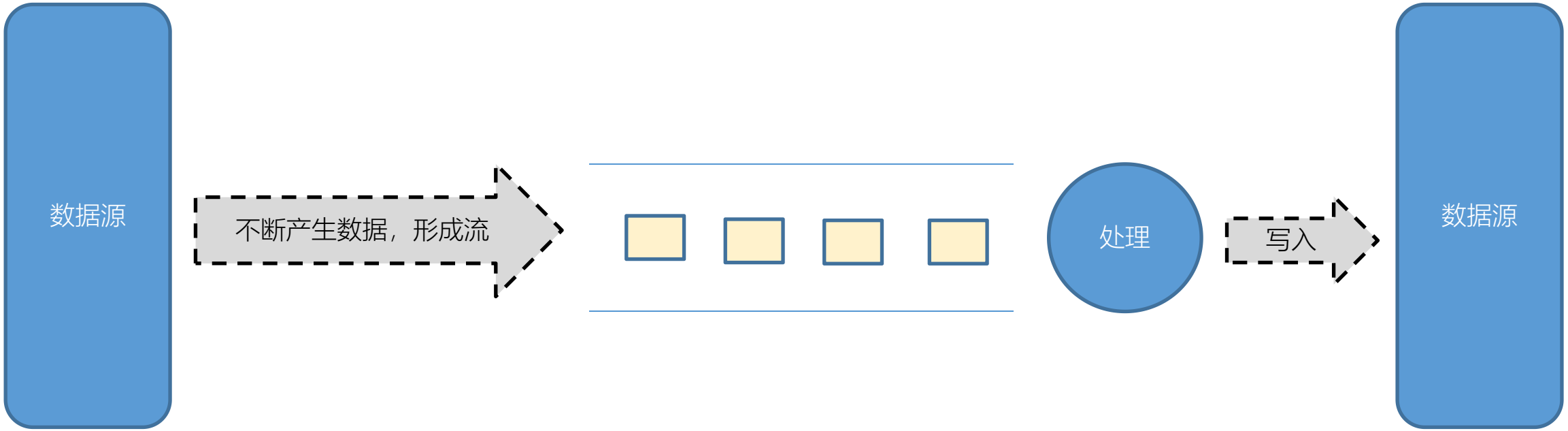
污水处理过程



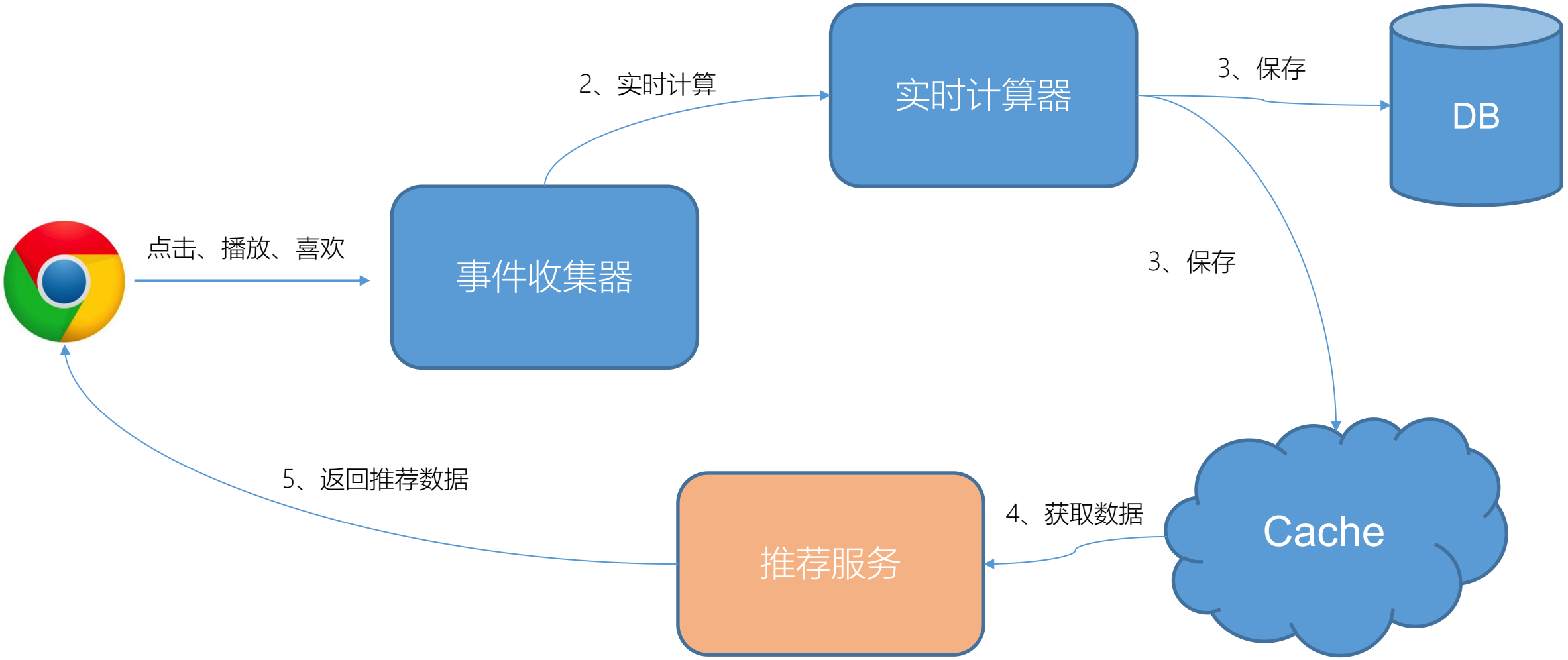
污水处理过程



Streaming



用户行为的实时推荐流程图



如何精准推荐视频?

精准

准确判断用户喜好

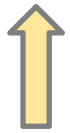
如何做到精准?

方式一：基于内容推荐

用户喜欢《钢铁侠》，系统认为他也喜欢《复仇者联盟》

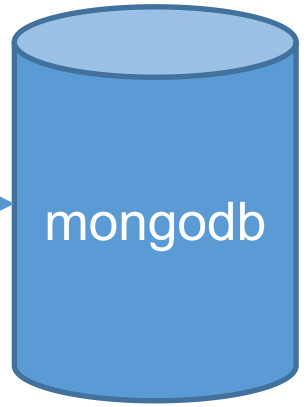
钢铁侠 { 动作片、漫威、动漫、科幻、美国、超级英雄 }

复仇者联盟 { 动作片、漫威、漫画改编、科幻、美国、超级英雄、黑寡妇 }



标签Tag

存储



计算物品（电影）间基于Tag的相似度

方式二：协同过滤推荐

计算用户A和用户B的喜好程度，互相推荐彼此喜好的物品

如何对新用户进行推荐?

- YouTube首次推荐不同主题的热门视频
- 再根据用户的点击行为进行再次推荐

课程介绍

九章算法
www.jiuzhang.com

	• 【互动】走进系统设计 & 新鲜事系统设计 • 【直播】秒杀系统与订票系统设计
Week1	• 【互动】从用户系统中学习数据库与缓存 • 【互动】网站系统、API设计与短网址 • 【直播】容器技术（K8S，Docker）
Week2	• 【互动】数据库拆分 & 限流器和实时数据系统设计 • 【互动】分布式数据库系统设计 • 【直播】在线文档协同编辑系统设计
Week3	• 【互动】聊天系统设计 • 【互动】分布式文件系统设计 • 【直播】视频流系统设计（Youtube / Netflix）
Week4	• 【互动】基于地理位置的信息系统 • 【互动】分布式计算系统设计 • 【互动】爬虫系统与搜索建议系统 • 【直播】Google Search / Search Ads Ranking 系统设计

Thanks