

数据库系统概论新技术篇

流数据分析与处理

陈跃国

中国人民大学信息学院

2017年4月

概述

- ❖ 本节课讲述流数据的基本概念，介绍流数据分析与处理的典型系统
- ❖ 通过对流数据管理系统与传统数据管理系统的对比，以及流数据分析与批处理方式的数据分析的对比，加深对流数据管理与分析处理系统所针对的应用和系统实现上的理解



流数据分析与处理

1 流数据的基本概念

2 流数据分析与处理系统

- 传统的流数据管理系统
- 面向大数据的流数据分析系统
- 面向大数据的流数据处理系统

3 小结



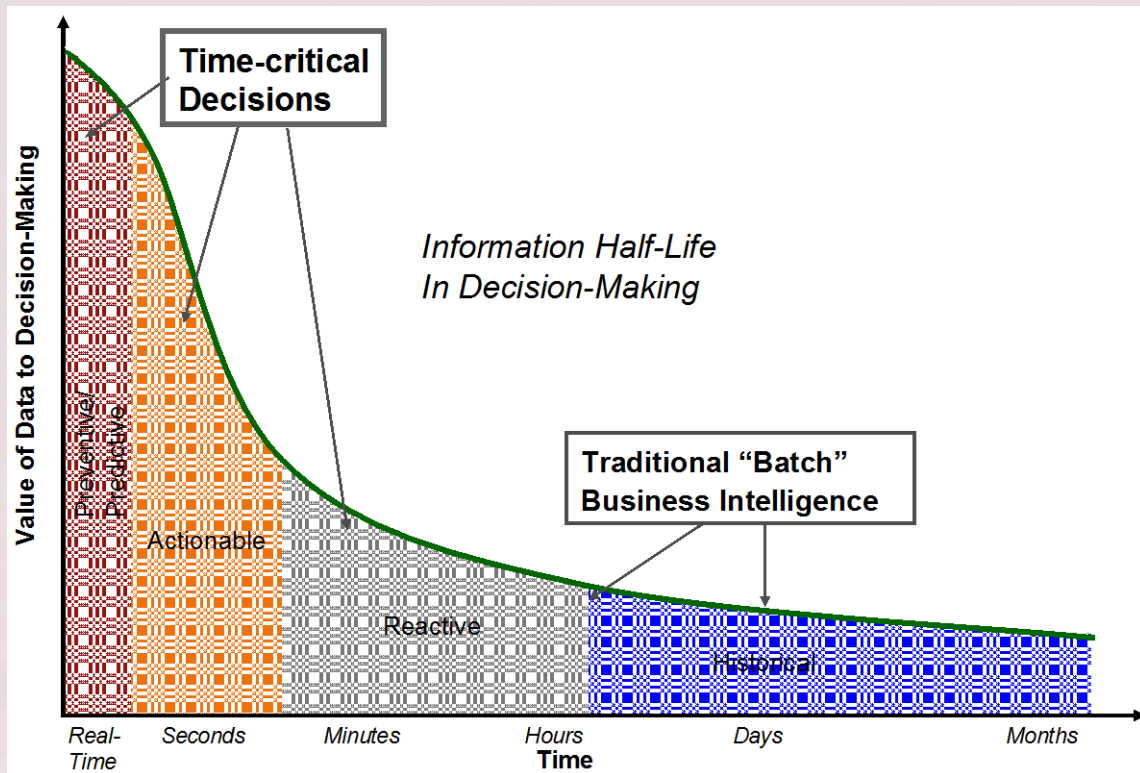
流数据

- ❖ 定义：连续的、没有边界的、快速的、随时间不断变化的系列数据项（如为结构化数据就是元组）
- ❖ 产生于很多应用领域
 - 网络监控，流量监控
 - 传感器网络, RFID
 - 电信呼叫记录
 - 金融应用
 - 网络日志，点击流
 - 制造过程
- ❖ 数据流管理系统DSMS



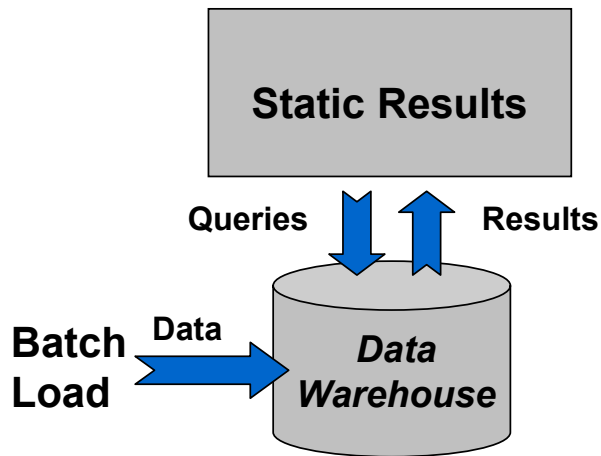
数据用于决策的价值随时间递减

- ❖ 批处理
- ❖ 近实时分析
- ❖ 流式分析与预测
 - Just-in-time decision making



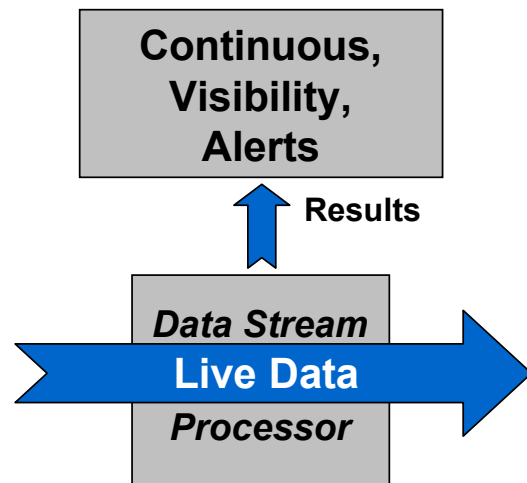
查询处理方式的不同

Traditional Query Processing



"Store First, Query Later"

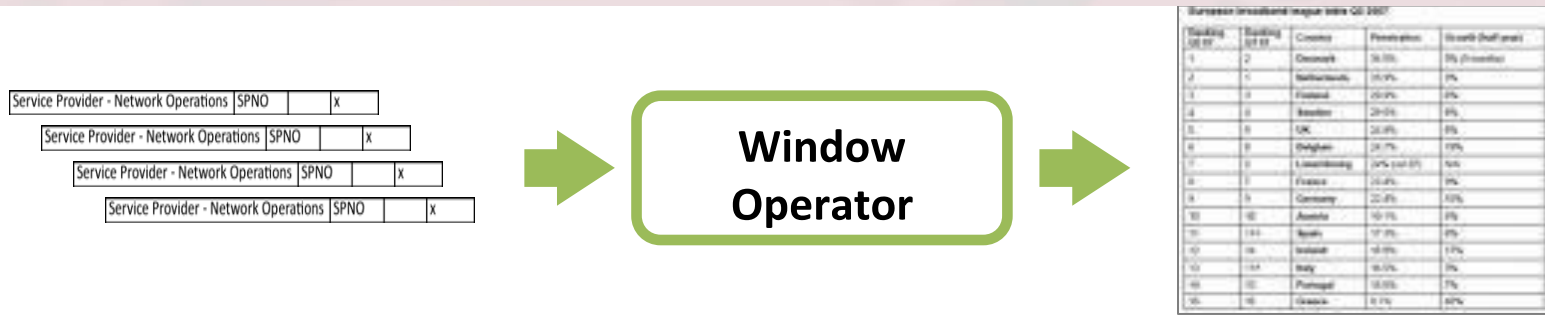
Stream Query Processing



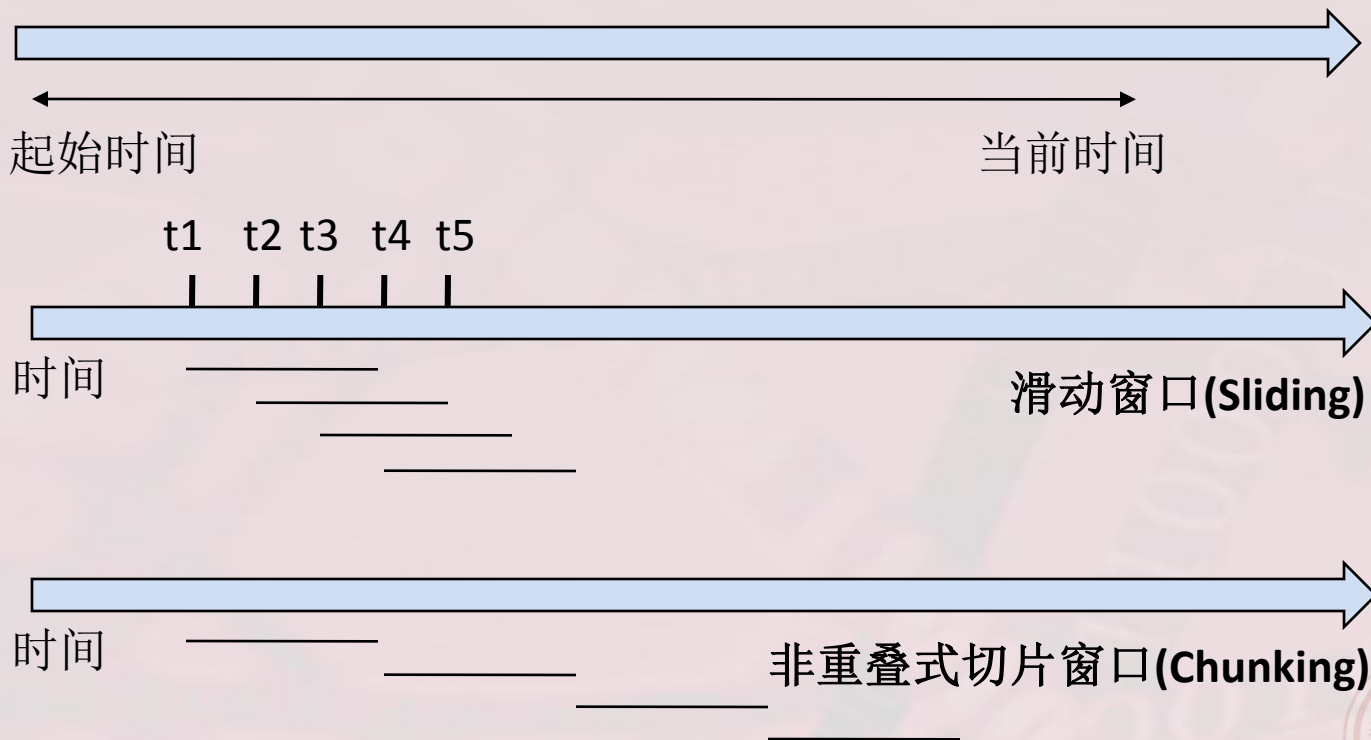
Process "On-the-Fly"

流数据处理模型

- ❖ 一个数据流是持续不断没有边界的记录集
- ❖ 一个表是给定记录的集合
- ❖ **SQL**语句是作用在表上的
- ❖ 通过时间窗口将流转换成表，每个时间窗口内的数据变成一个表
- ❖ 通过改变时间窗口而改变表，并重复应用**SQL**
- ❖ 结果不断追加到输出流中

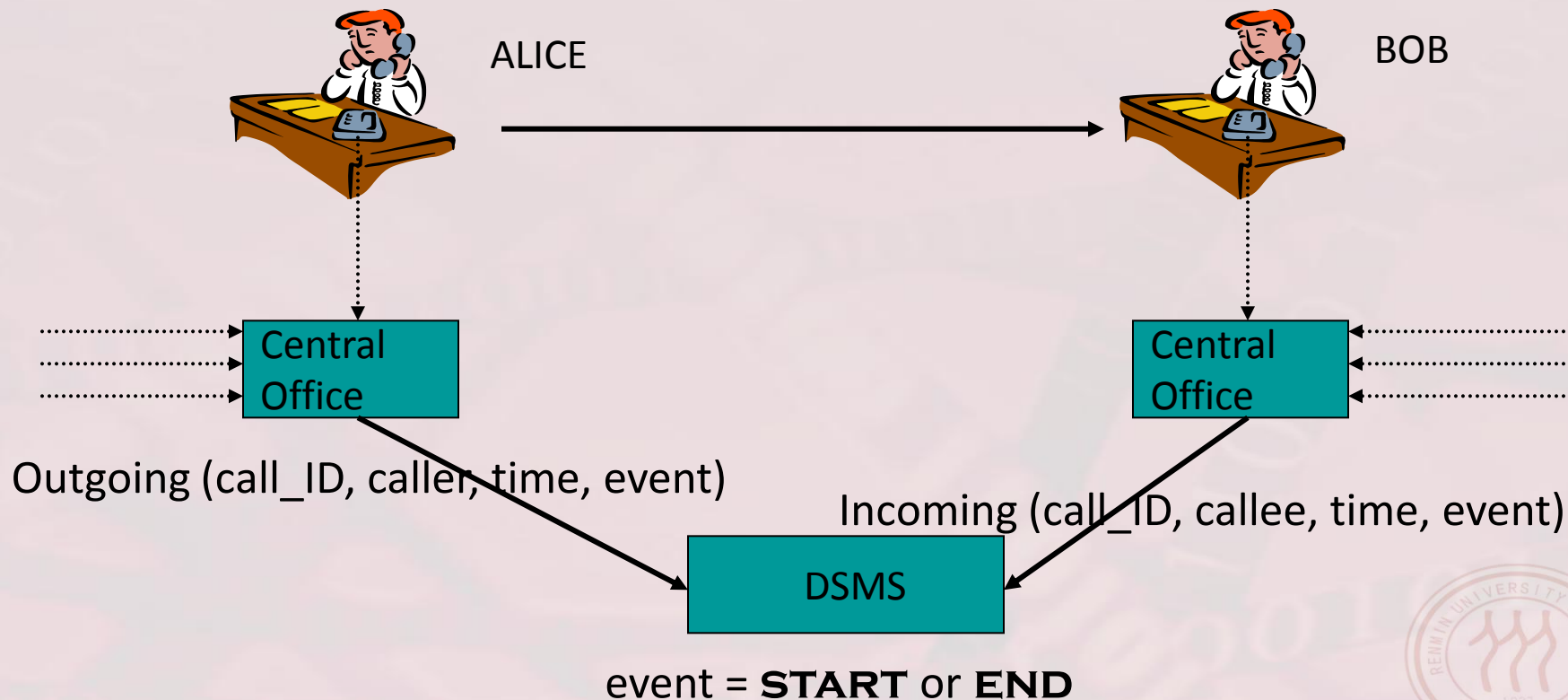


时间窗口截取表的不同方式



还有一种起始位置固定，结束位置每隔一定时间变化的窗口截取方式，不常见

实例



实例1: Self-join

- ❖ 找到所有时间超过两分钟的**outgoing calls**

```
SELECT  O1.call_ID, O1.caller
FROM    Outgoing O1, Incoming O2
WHERE   (O2.time - O1.time > 2)
        AND  O1.call_ID = O2.call_ID
        AND  O1.event = START
        AND  O2.event = END )
```

- ❖ 以流数据方式输出结果
- ❖ 超过2分钟没有发现**END**，可以输出结果



实例2：连接操作

❖ 找出所有通话对

```
SELECT  O.caller, I.callee  
FROM    Outgoing O, Incoming I  
WHERE   O.call_ID = I.call_ID
```

❖ 以流数据形式输出结果

- ❖ 由于没有时间窗口限制，需要无限大内存去处理连接操作
- ❖ 可以加上时间限制，如， $I.time - O.time < 1$



流数据分析与处理

1 流数据的基本概念

2 流数据分析与处理系统

- 传统的流数据管理系统

- 面向大数据的流数据分析系统

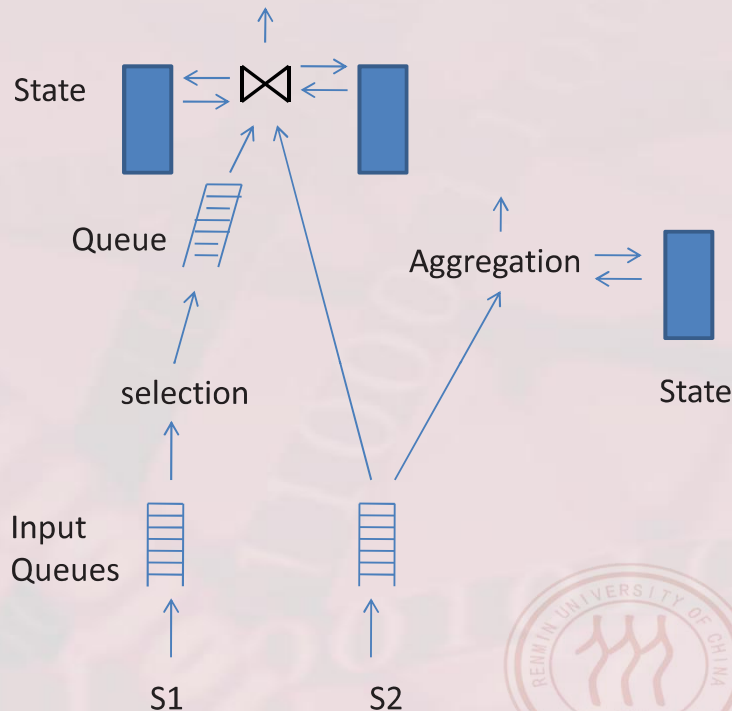
- 面向大数据的流数据处理系统

3 小结



流数据管理系统

- ❖ 及时处理数据 (**on-the-fly**)
- ❖ **SQL**语义支持
- ❖ 处理流数据的质量问题
- ❖ 预测
- ❖ 实时数据与历史数据的融合
- ❖ 保障数据可获得性
- ❖ 数据划分并获得好的可扩展性
- ❖ 实时处理能力



DBMS v.s. DSMS

指标	DBMS	DSMS
数据类型	持久化的数据	挥发性流数据
访问模式	随机读取	连续读取
查询特点	一次性查询	连续查询
存储假设	外存空间无限	有限内存
更新频率	低更新速度	高更新频率
数据质量	高质量数据	过期、不精确、错位数据
查询处理	一次性查询计划	计划需要随数据特征变化

典型实例

- ❖ Amazon/Cougar (Cornell) – sensors
- ❖ Aurora (Brown/MIT) – sensor monitoring, dataflow
- ❖ Hancock (AT&T) – telecom streams
- ❖ Niagara (OGI/Wisconsin) – Internet XML databases
- ❖ OpenCQ (Georgia) – triggers, incr. view maintenance
- ❖ Stream (Stanford) – general-purpose DSMS
- ❖ Tapestry (Xerox) – pub/sub content-based filtering
- ❖ Telegraph (Berkeley) – adaptive engine for sensors
- ❖ Tribeca (Bellcore) – network monitoring



流数据分析与处理

1 流数据的基本概念

2 流数据分析与处理系统

- 传统的流处理管理系统
- 面向大数据的流数据分析系统
- 面向大数据的流数据处理系统

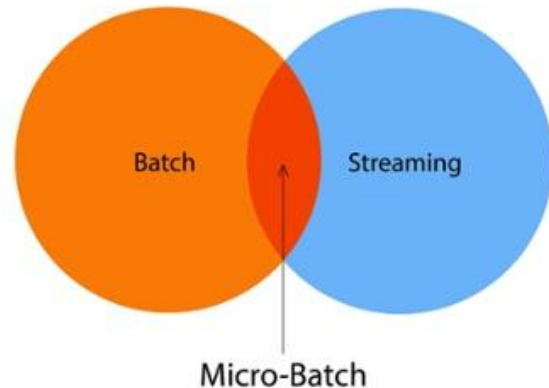
3 小结



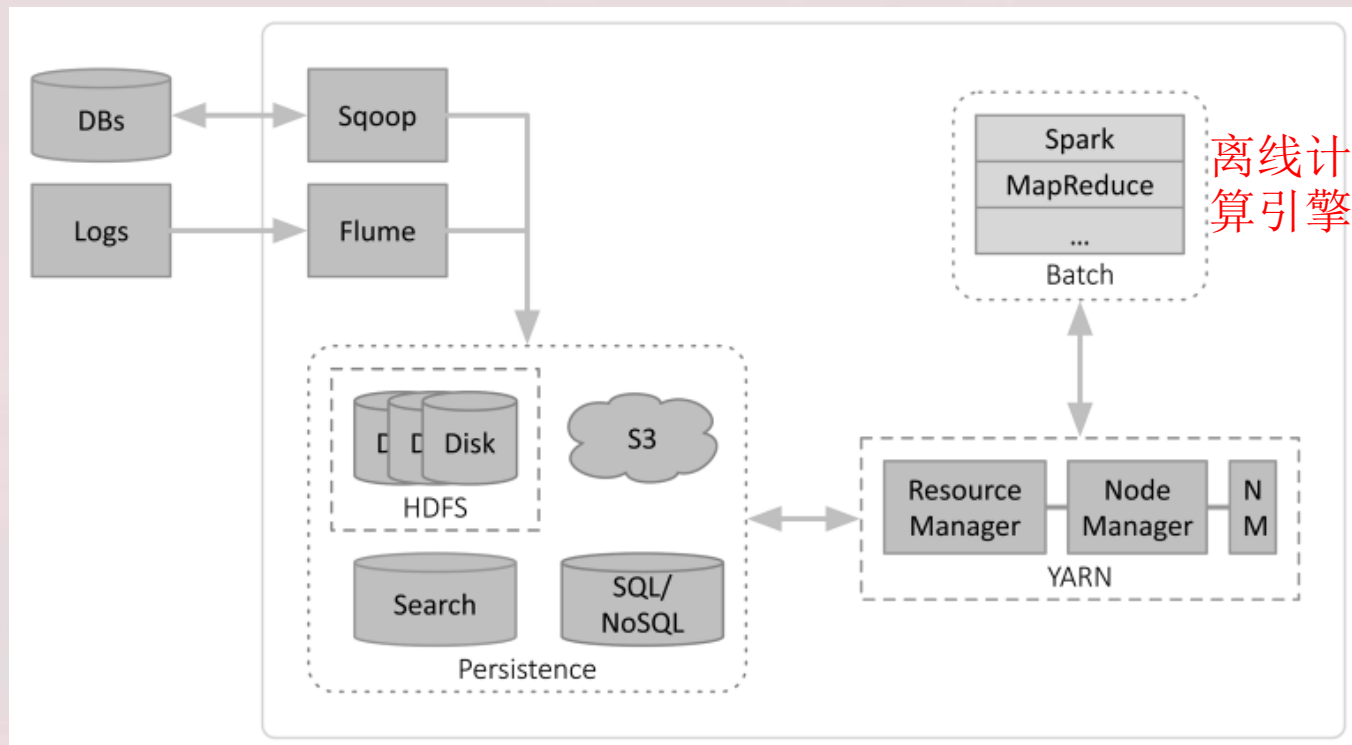
流式分析处理 v.s. 批处理系统

特征	Batch	Stream	Mini-Batch
数据	全盘	即来即处理	基于小批量的批处理
计算	计算可能较为复杂	简单独立的计算	介于两者中间
性能	慢(分钟甚至小时), 更关注数据吞吐量	亚秒级 (难于高效率的维护状态)	秒级 (更容易基于窗口和状态进行计算)
实例	MapReduce	Storm	Spark Streaming

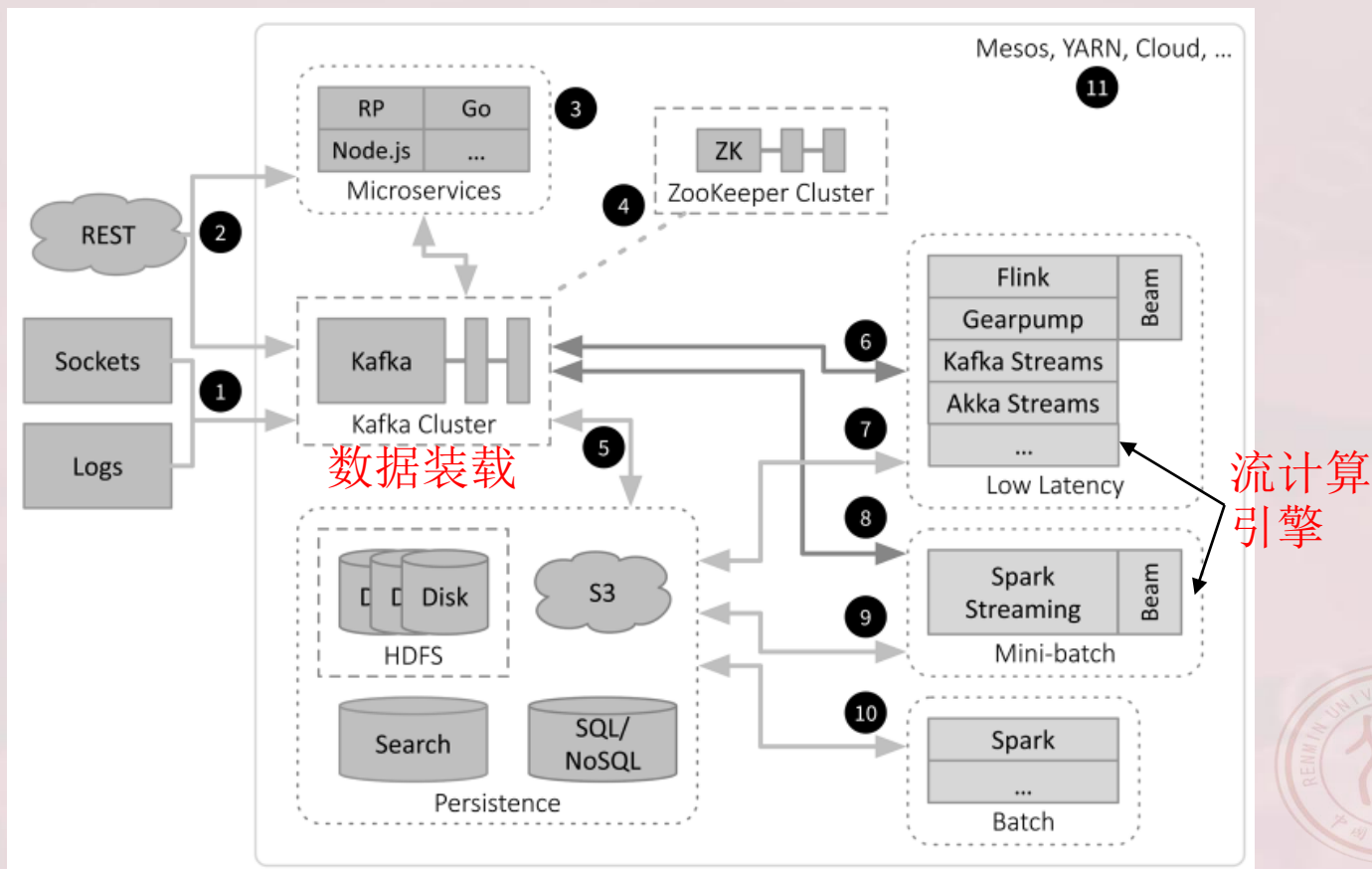
Batch vs. Streaming



典型大数据批处理系统架构



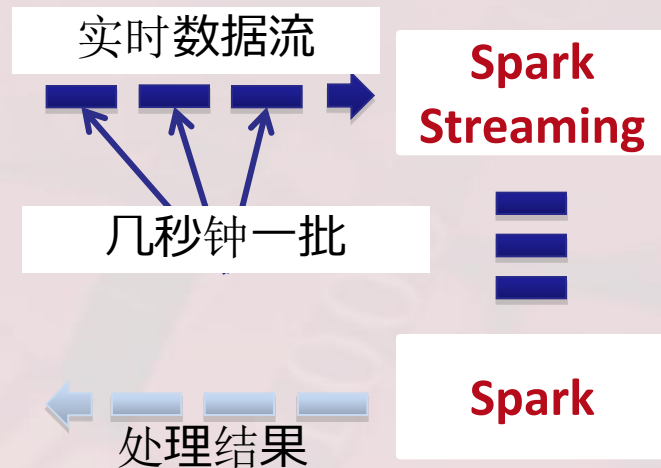
典型流式分析大数据系统架构



Spark Streaming

❖ 将数据流切成不重叠的小批的方式处理流计算

- 每批最小可以到半秒
- 延迟1秒到几秒钟左右
- **Spark RDD**的特点，使其在同一个系统中融合批处理和流处理两类计算



流数据分析与处理

1 流数据的基本概念

2 流数据分析与处理系统

- 传统的流数据管理系统
- 面向大数据的流数据分析系统
- 面向大数据的流数据处理系统

3 小结



流数据分析 v.s. 数据流处理

❖ 数据流分析

- 典型的数据流分析是以时间窗口为基础的、基于**SQL**的数据统计和分析
- 例如监控一段时间内采集数据的平均值、最大值等

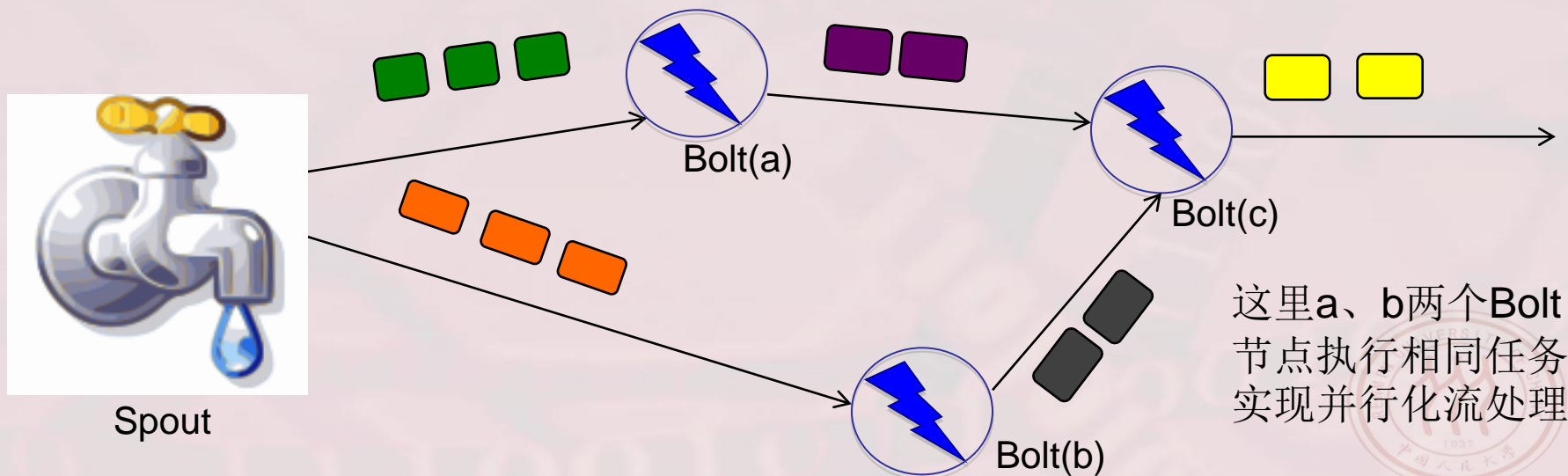
❖ 数据流处理

- 以单元组处理为主，用并行化来解决元组处理的复杂性
- 串行化处理数据，形成多步处理流水线
- 例如对博文的预处理，信息抽取，分类、聚类等



Storm: 流数据处理代表

- ❖ 由Spouts和Bolts节点构成有向图
- ❖ Spout: 流数据产生源, 通常是从数据库/爬虫等地方获得数据
- ❖ Bolt: 处理数据流, 输出数据流给其它Bolt节点



Bolt计算节点

❖ Bolt上常见的操作

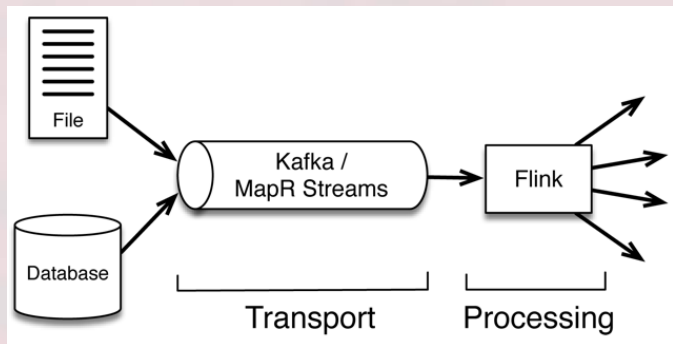
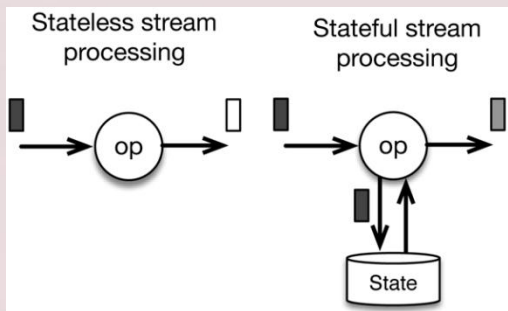
- 过滤：只继续传播满足一定过滤条件的元组
- 连接：接收两个流A和B，输出满足一定条件的 (A, B) 对
- 转换：根据函数，修改每个tuple的值
- 其它很多操作，如自然语言处理，多媒体数据处理，ETL数据处理等

❖ Bolt是计算节点，需要处理大量数据

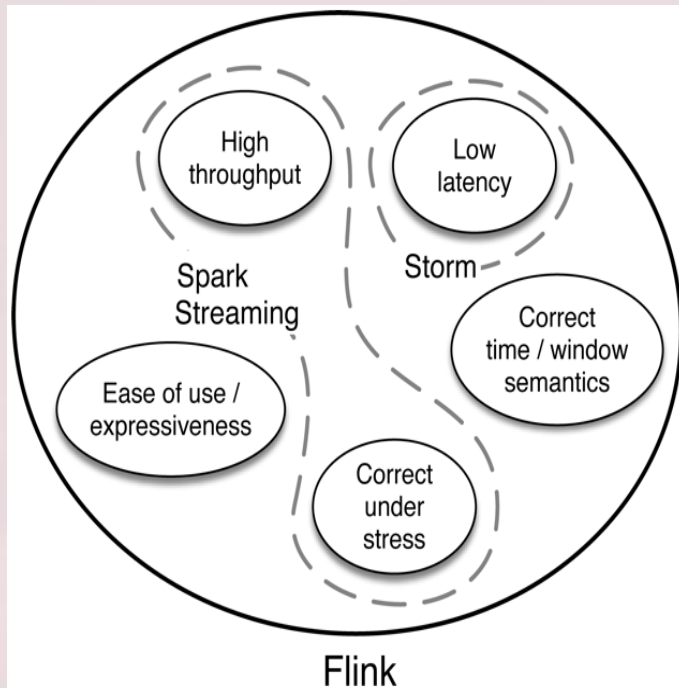
- 如何算得快是关键
- 通过并行化处理，涉及数据流划分
- 不同Bolt任务之间的协调，以及容错的支持



Flink: 有状态的流处理



有状态，容错，高可扩展，低延迟



Flink兼具流数据分析和流数据处理的功能

流数据分析与处理

1 流数据的基本概念

2 流数据分析与处理系统

- 传统的流数据管理系统
- 面向大数据的流数据分析系统
- 面向大数据的流数据处理系统

3 小结



小结

- ❖ 大数据通常是以流数据的形式不断堆积的
- ❖ 如果需要较为实时的从大数据中获得洞察，就需要流数据分析和处理技术
- ❖ 典型应用
 - 传感器网络、M2M数据中心
 - 互连网舆情监控
 - 入侵检测
 - 实时商业智能

