课题编号：2016YFB1000103

JOINT CLOUD

国家重点研发计划项目

课题三：软件定义的云际存储
软件定义策略

沃天宇

2019.11.23 杭州

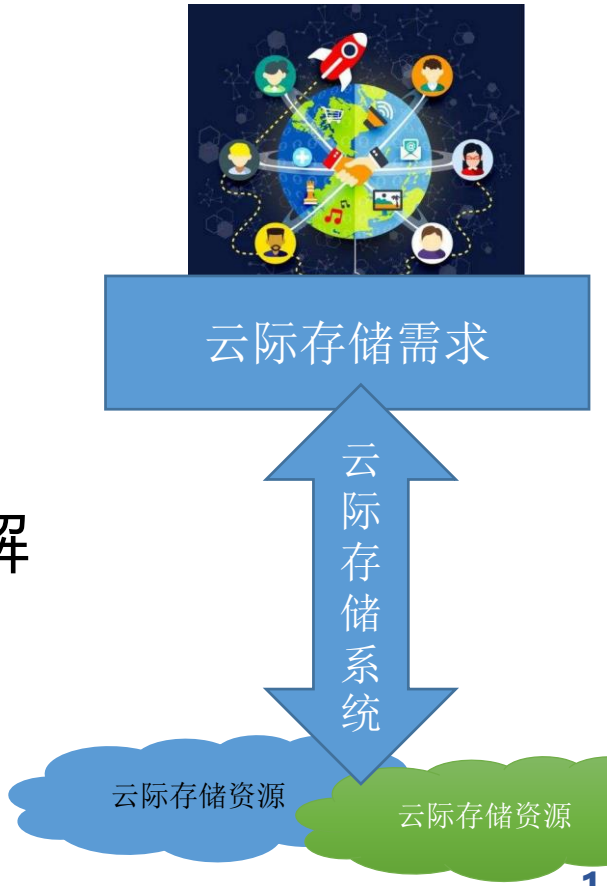
北京航空航天大学 BEIHANG UNIVERSITY 1952
ISCAS 中国科学院软件研究所 Institute of Software Chinese Academy of Sciences
金山云 WWW.KSYUN.COM

内容提要

- 云际存储需求回顾
- 云际存储策略与机制
- 总结

科学问题

- 云际存储资源抽象模型与聚合
  - 从资源角度解读云际环境下，存储资源的抽象管理模型与聚合方法
  - VFS --> CloudVFS
- 多尺度的大数据分布与存储管理
  - 从系统的角度考虑云际存储系统的构造方法
  - 定义软件 --> 软件定义
- 能效优化的云际存储资源协同
  - 云际存储需求高度动态、复杂，不断加深对需求的理解
  - 人工建模 --> 数据驱动



课题背景

研究目标

协作模型

- 云际软件定义模型和构造演化机理
- 云际网络互联模型与协同服务机制

协作机制

- 云际存储资源的聚合模型与协同机制
- 云际计算资源的多尺度聚合与效能评价
- 云际计算中虚拟专用云的动态构建

考核指标

模型机制

- 论文：33篇
- 专利：11个
- 标准：存储方面

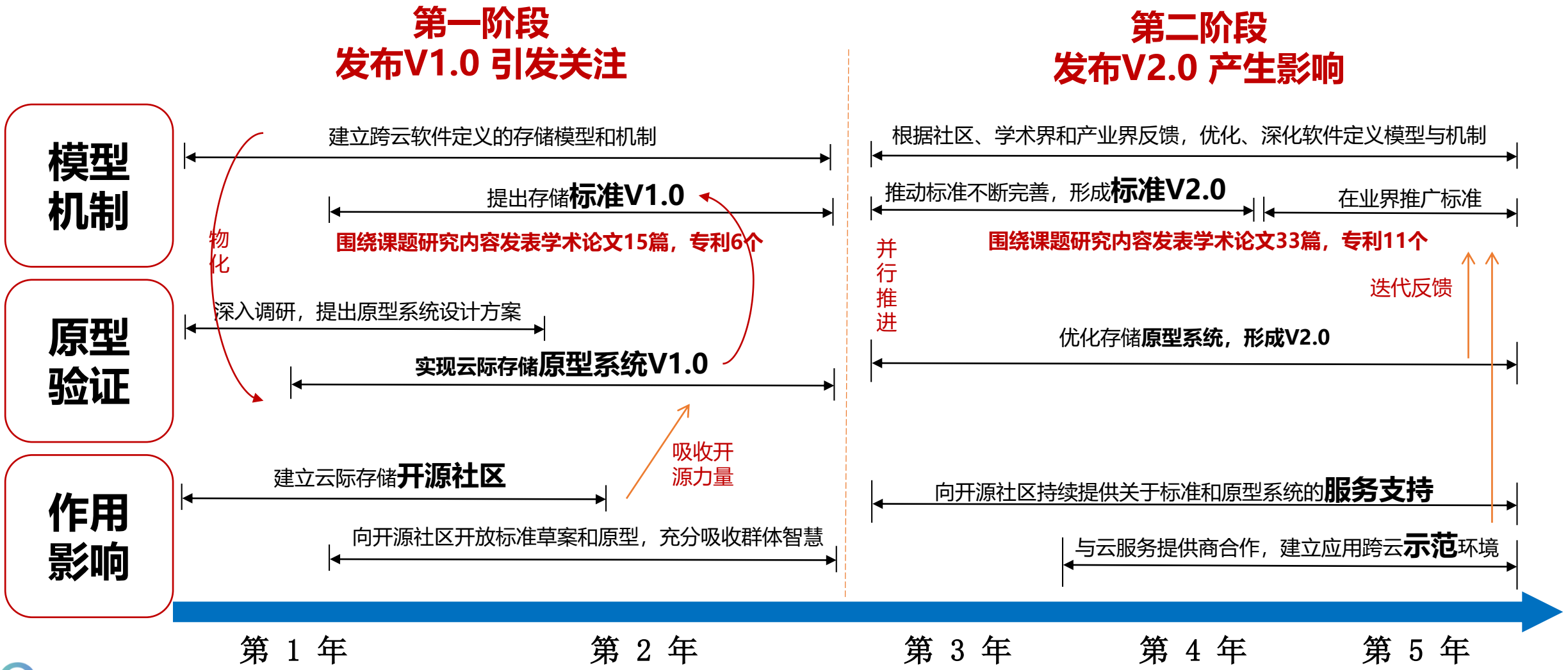
原型验证

- 云际互联原型
- 云际存储原型
- 跨云计算原型
- 虚拟专用云原型

作用影响

- 开源标准文本
- 开源原型代码
- 提供在线服务
- 行业示范应用

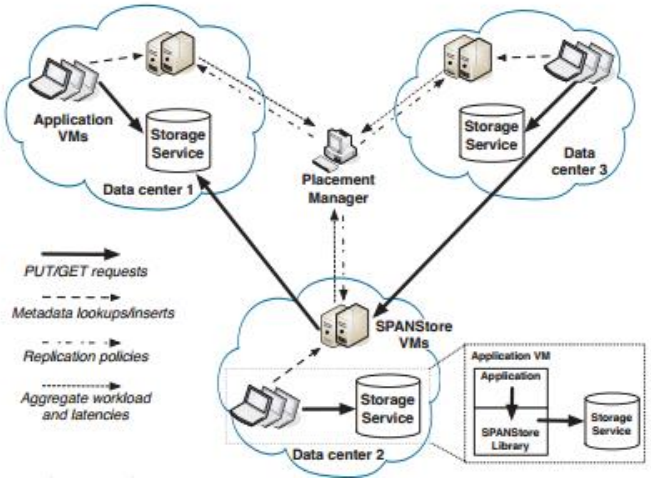
目标和任务


第一阶段
发布V1.0 引发关注
第二阶段
发布V2.0 产生影响
模型机制
原型验证
作用影响
建立跨云软件定义的存储模型和机制
提出存储标准V1.0
围绕课题研究内容发表学术论文15篇，专利6个
物化
深入调研，提出原型系统设计方案
实现云际存储原型系统V1.0
建立云际存储开源社区
吸收开源力量
向开源社区开放标准草案和原型，充分吸收群体智慧
根据社区、学术界和产业界反馈，优化、深化软件定义模型与机制
推动标准不断完善，形成标准V2.0
在业界推广标准
围绕课题研究内容发表学术论文33篇，专利11个
并行推进
迭代反馈
优化存储原型系统，形成V2.0
向开源社区持续提供关于标准和原型系统的服务支持
与云服务提供商合作，建立应用跨云示范环境
第 1 年
第 2 年
第 3 年
第 4 年
第 5 年

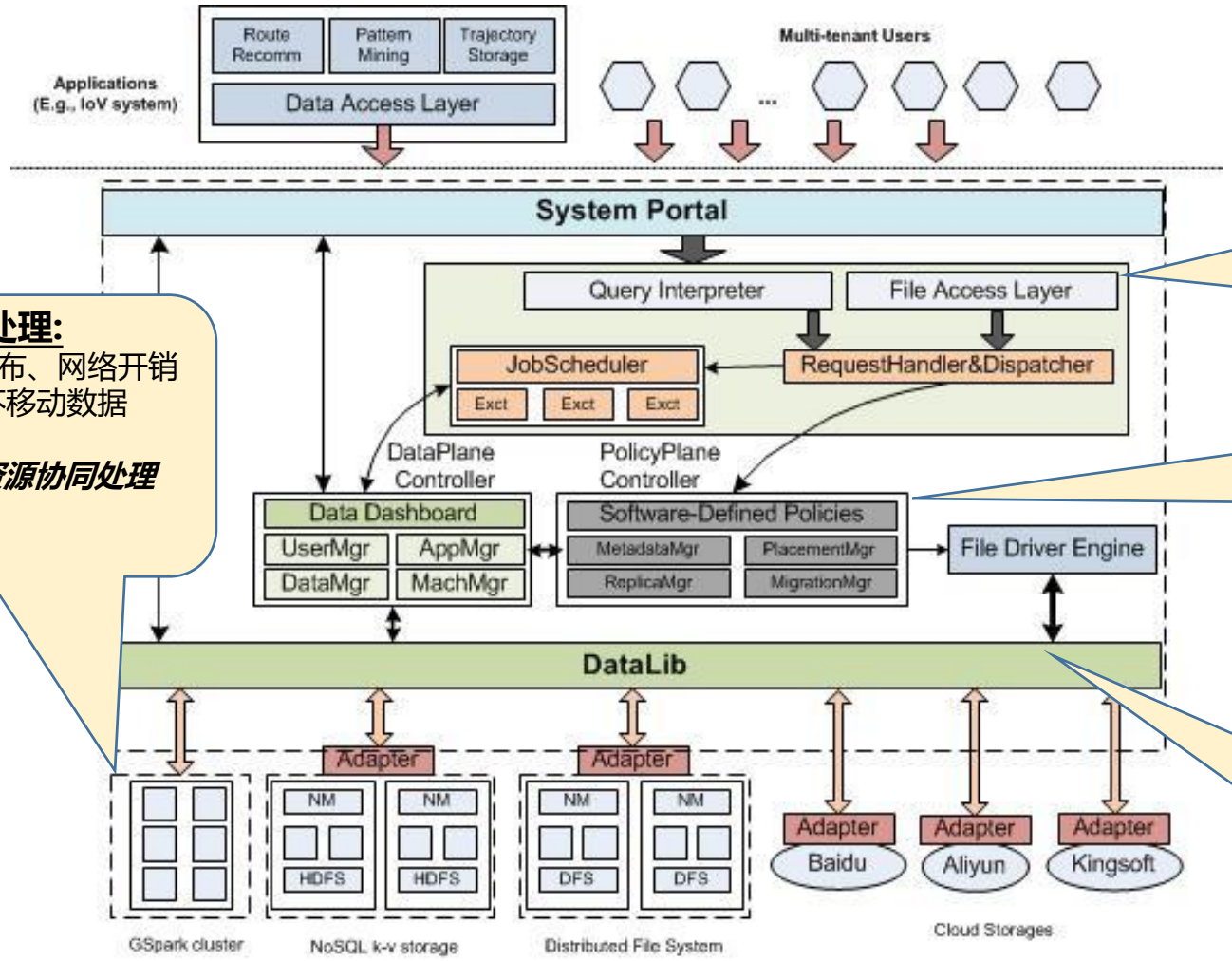
软件定义云际存储

- 软件定义存储已经发展了多年
  - 经济性：非硬件绑定
  - 灵活性：自动调优
  - 扩展性：无限扩容

- 软件定义云际存储增加什么？
  - 策略上
    - ✓跨域分布、可信容错、优化成本、对等协商
  - 机制上
    - ✓标准化接口、穿透云管理边界



总体架构



多源数据的分布式处理:
✓考虑数据全局地理分布、网络开销
✓计算任务的分发 + 不移动数据

能效优化的云际存储资源协同处理方法

请求处理和分发: 查询解释（从语义翻译成DAG任务）、文件访问

能效优化的云际存储资源协同处理方法

策略平面管理控制: 元数据管理、数据副本管理、数据放置管理、迁移的管理策略

软件定义的云际存储与分布管理技术

数据统一访问类库DataLib
✓适配不同的数据存储API，并对数据操作进行封装
✓为上层提供数据抽象和封装的统一形式

软件定义的云际存储资源抽象模型与聚合方法

内容提要

- 云际存储需求回顾
- 云际存储策略与机制
  - 多云数据放置与动态调整策略
  - 云际环境下分布式流处理系统容错机制
- 总结

研究背景与意义

- **单云存储的弊端：**
- **1）云供应商锁定问题**：用户将数据存储在一个云，更换云的成本较大
- **2）云故障问题**：云存储服务故障会对用户造成损失
  - 例如，阿里云2018年6月27日下午故障1小时，服务不可用，相关依赖受到影响；
  - 腾讯云2018年7月20号故障导致数据丢失
- **采用多云存储：**
- 1）多云冗余存储可避免供应商锁定，改善云存储的可用性。
- 2）不同**云供应商的特性：**
- **a.定价**，不同云的存储费用和下载流量费用不同
- **b.地域**，云存储的数据中心部署在不同地域
- 3）**用户数据的访问特性：**
- **a.访问频率**，文件数据不同的下载频率
- **b.访问地域**，文件数据不同的下载地域

云存储供应商	标准存储 元/GB/月	流量费用 元/GB
阿里云	0.148 元/GB/月	0.5 元/GB
百度云	0.128 元/GB/月	0.6 元/GB

阿里云和百度云 对象存储服务价格对比

云存储供应商在不同的地域部署数据中心

研究背景与意义

问题提出：

如何综合考虑**多云存储特性**和**用户数据访问特性**，

制定一种**数据放置策略**，使**成本和延迟最优**，

同时满足**可用性和容错性**要求？

云存储供应商	标准存储 元/GB/月	流量费用 元/GB
阿里云	0.148 元/GB/月	0.5 元/GB
百度云	0.128 元/GB/月	0.6 元/GB

阿里云和百度云 对象存储服务价格对比

- **a.定价**，不同云的存储费用和下载流量费用不同
- **b.地域**，云存储的数据中心部署在不同地域
- 3）**用户数据的访问特性：**
- **a.访问频率**，文件数据不同的下载频率
- **b.访问地域**，文件数据不同的下载地域

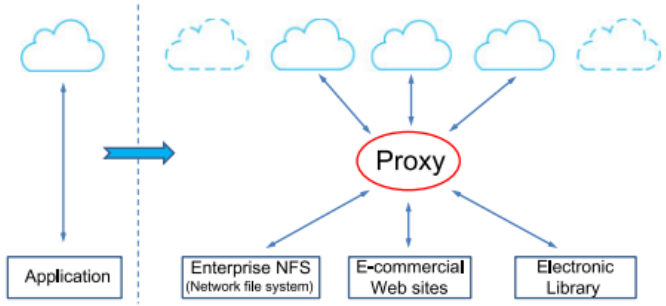
云存储供应商在不同的地域部署数据中心

国内外研究现状

在用户和云存储供应商之间**构建多云存储服务代理**，
通过**代理**对**多个云存储服务**进行管理，同时为**用户提供存储服务**。

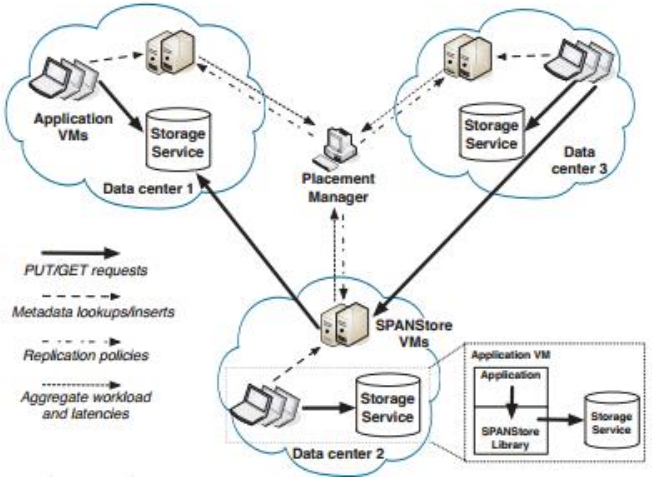
1）集中式代理架构
优点：
方便文件元数据信息的管理
多云冗余采用多副本和纠删码编码方案
不足：
没有考虑代理**多地域**部署
对**延迟**和**动态调整**优化

相关论文：
[1]Triones(TOC,2016) [2]CHARM(2015)
[3]Scalia(SC, 2012)



2）地理分布式代理架构
优点：
多地域分布，方便用户访问最近的副本数据，减少延迟
不足：
多副本冗余的数据存储量和传输量大，
没有考虑**进一步减少成本**

相关论文：
[7]DAR(TON,2017) [6]SPANStore(SOSP,2013)



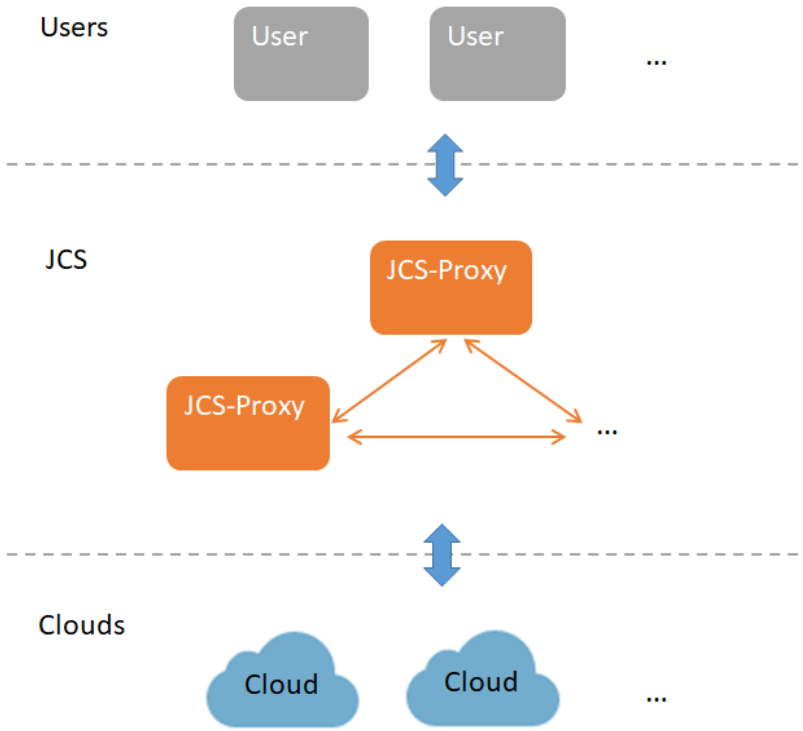
研究目标与研究内容

研究目标：

- 在云际环境下，针对单云故障问题，
- 考虑多云存储特性和用户访问特性，
- **研究一种全局成本和延迟优化的数据放置策略，**
- 并满足可用性和容错性要求。

研究内容：

- 1）面向云际存储的**数据放置优化方法**
- 2）针对数据访问模式变化的**动态调整方法**
- 3）针对云存储服务故障的**数据恢复方法**



总体架构设计图

1.面向云际存储的数据放置优化方法

输入：（默认/自定义）
1.多云冗余配置信息：
供应商锁定级别，容错级别，可用性
2.文件访问特性：
文件大小，存储时间，
下载频率，下载地域
3.多云存储特性：
云存储定价，多个代理和云之间的延迟
4.成本和延迟权重

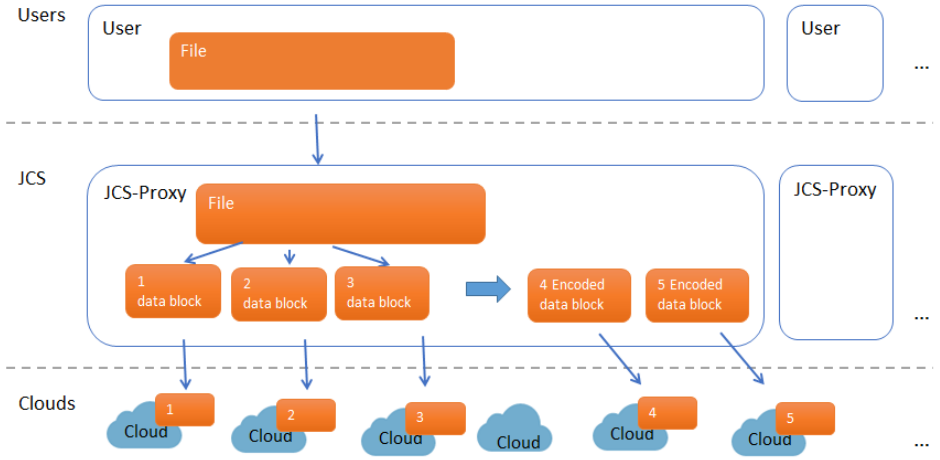
↓

数据放置优化问题求解
优化目标：全局成本和延迟最优
建模：混合整数规划问题
求解：启发式求解方法

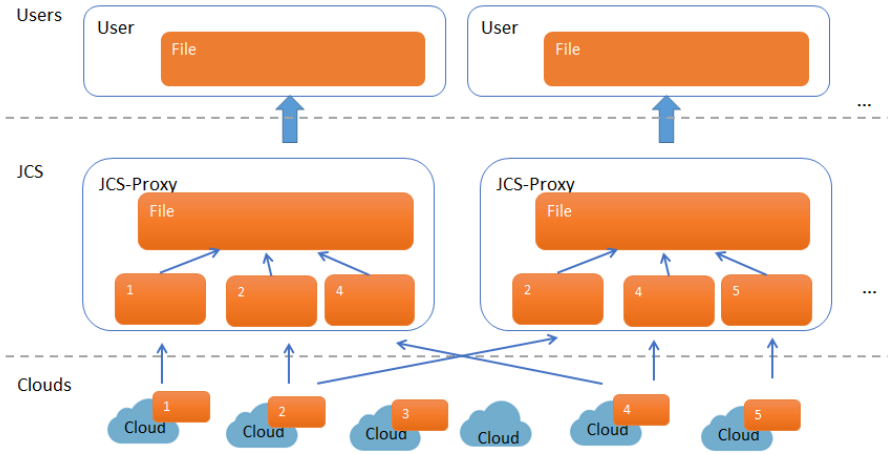
↓

输出：
数据放置优化策略结果
1.数据块存储方案
2.多个云际存储代理的数据下载方案

求解流程



数据块存储（示意图）



数据块下载（示意图）

1.面向云际存储的数据放置优化策略，建模

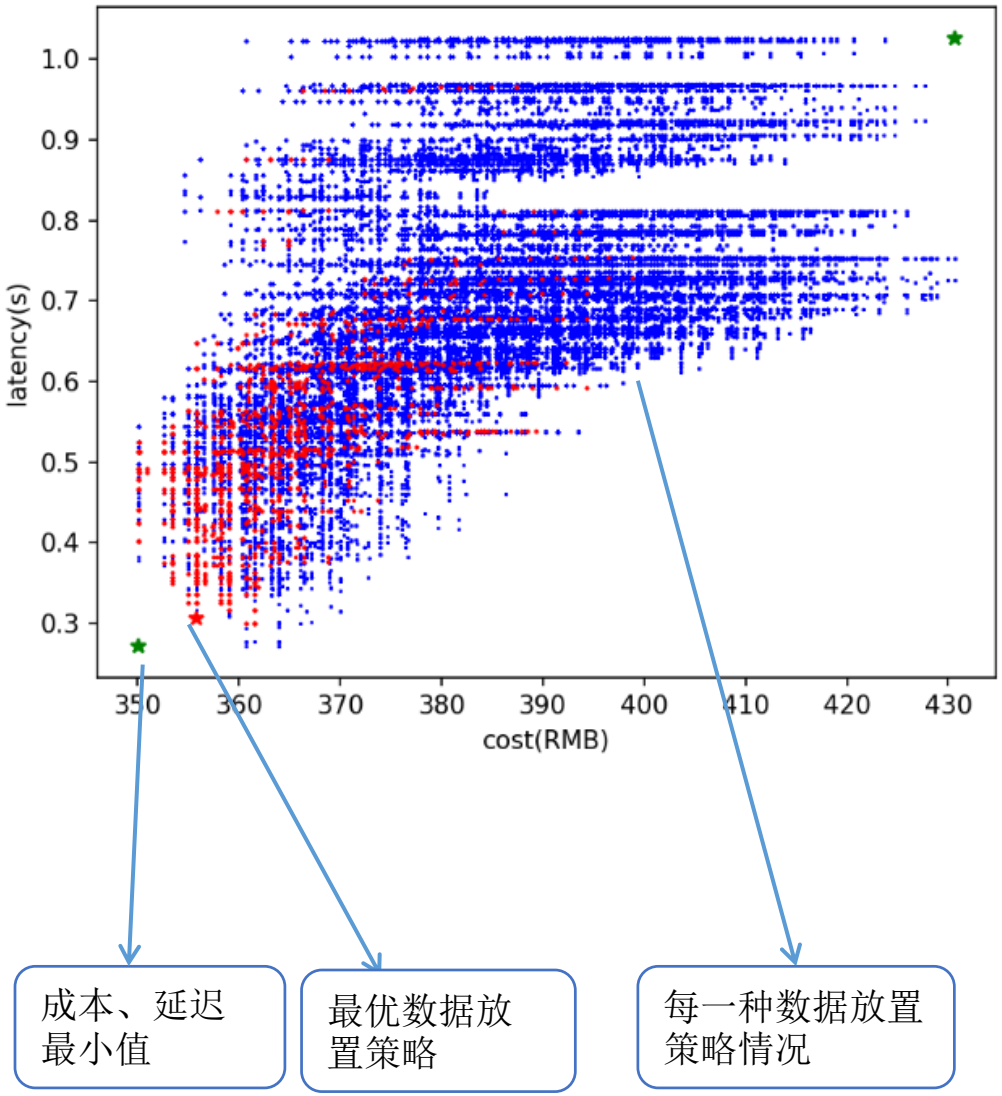
用**混合整数规划**来描述数据放置优化问题

约束条件：
1. 供应商锁定级别v，限制云个数范围
2. 容错级别m，表示最多允许故障的云个数
3. 可用性A，定义为不超过m个云地域同时崩溃的概率

总成本和总延迟定义：
4. 总成本c：存储成本，下载成本，请求次数成本
5. 总延迟l：每个代理下载延迟，与下载频率权重的乘积求和

目标函数f：最小化成本和延迟
（数据放置策略到成本和延迟最小值的欧氏距离）

$$f_{c,l} = \min\left(\sqrt{w_c * \left(\frac{c - c_{min}}{c_{max} - c_{min}}\right)^2 + w_l * \left(\frac{l - l_{min}}{l_{max} - l_{min}}\right)^2}\right)$$



1.面向云际存储的数据放置优化策略，求解

1.遍历求解

a)满足约束条件的纠删码参数

b)数据存储方案所有情况

c) 多个云际存储代理的数据下载方案

所有情况个数：

$$\sum_{n=1}^{N}\sum_{k=1}^{n} C_N^n (C_n^k)^p$$

2.启发式求解

a)满足约束条件的纠删码参数

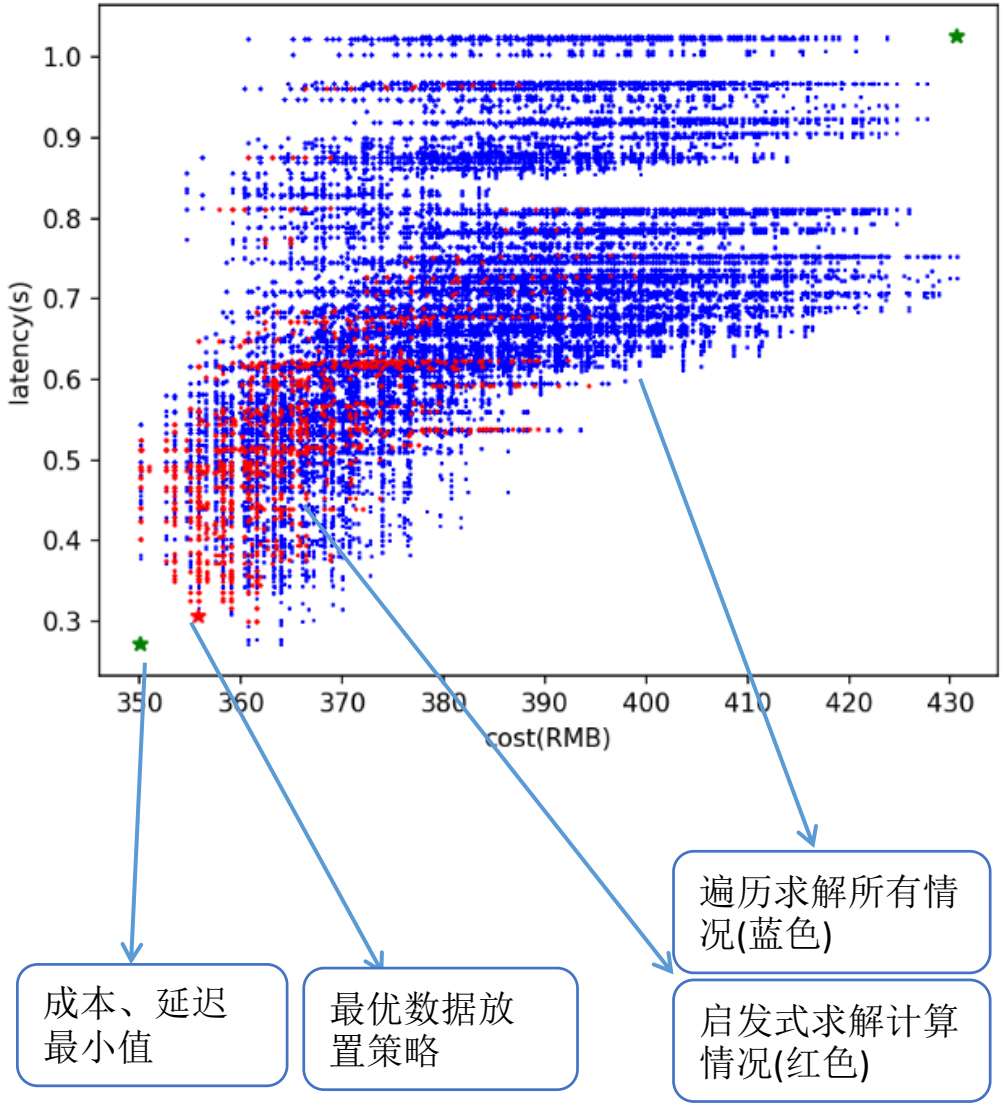
b)迭代选择数据存储方案

c)迭代选择多个云际存储代理的数据下载方案

每次选择 目标函数值 梯度下降最大 的方案进行迭代

计算情况个数：

$$\sum_{n=1}^{N}\sum_{k=1}^{n} (N-n)^2 n * p(n-k)^2 k$$



2.针对数据访问特性变化的动态调整方法

为什么需要动态调整：

用户访问模式发生变化，

按照原来的数据放置策略导致成本和延迟增加

动态调整条件：

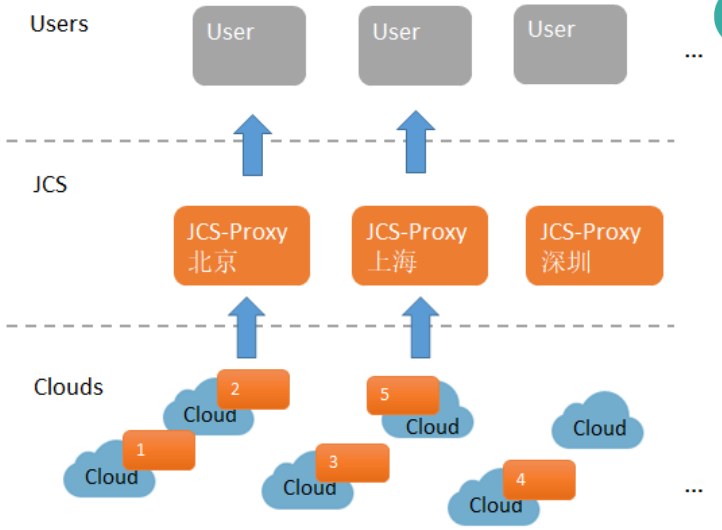
（一段时间内）旧策略的目标函数值 >

（一段时间内） 新策略的目标函数值

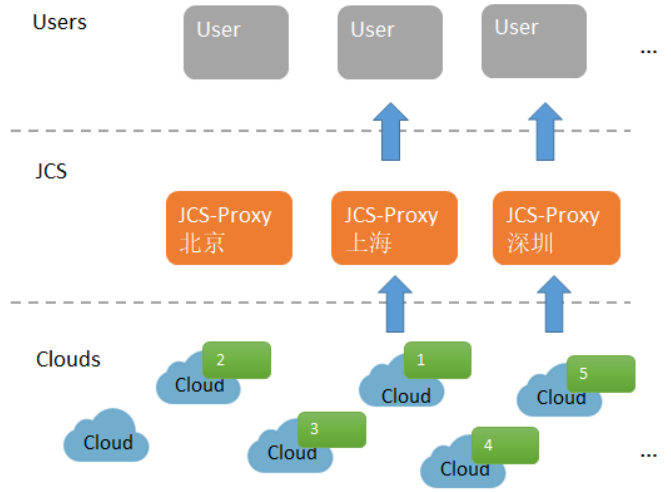
其中新策略的目标函数值考虑了**迁移成本的影响**

迁移过程需要根据新策略重新上传

迁移成本是文件数据下载一次的成本



北京、上海下载（示意图）



上海、深圳下载（示意图）

3.针对云存储服务故障的数据恢复方法

为什么要进行数据恢复：

当某个云发生故障时，需要保证文件数据可继续下载，并且文件数据满足容错性条件

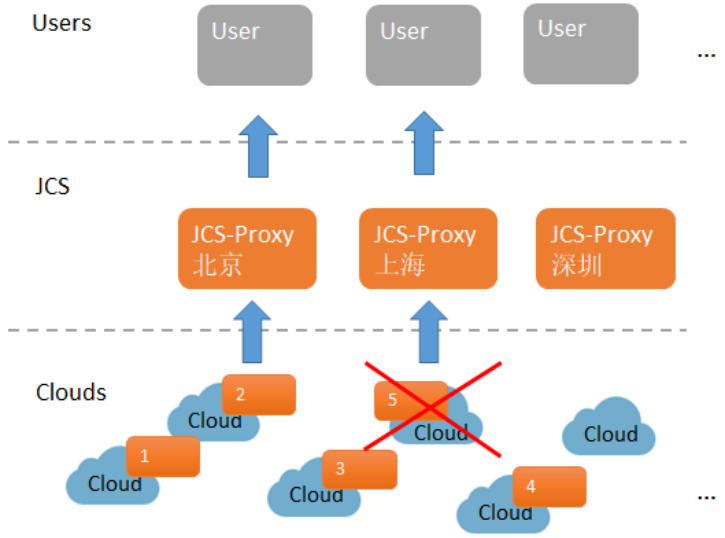
云临时故障：

下载文件需要从可用云中重新计算临时数据下载方案

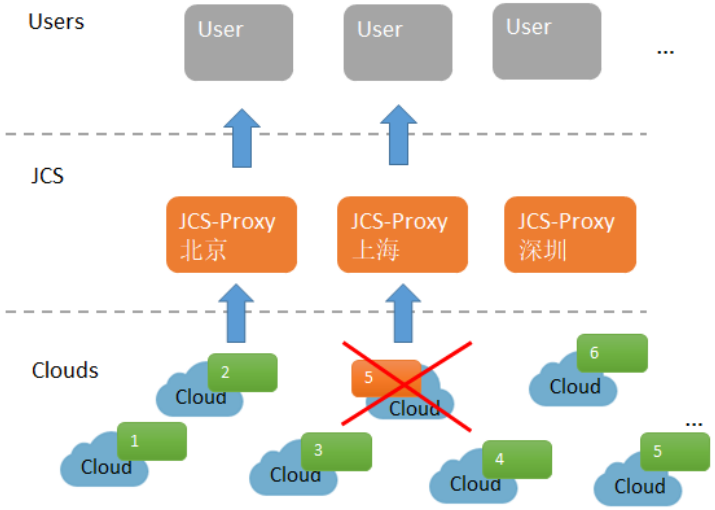
删除文件需要 记录故障云中待删除的数据块

云超时故障和待删除：

进行数据恢复，重新计算数据放置策略并上传（因为新的数据放置策略纠删码参数可能会发生变化）



云临时故障（示意图）



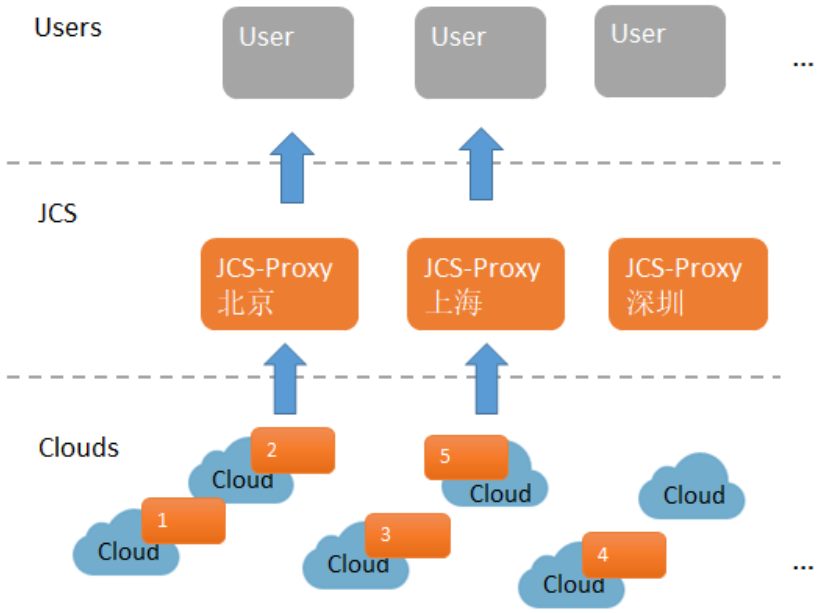
云超时故障，数据恢复（示意图）

实验分析

实验环境：
云际存储管理系统代理部署：**北京、上海、深圳**三个地域
使用的云存储服务：对象存储服务
使用云存储供应商：**阿里云，百度云，金山云**
由于不同云地域、存储类型之间的隔离性，
选择**22个不同云地域数据**中心视为22个云进行存储

实验数据：
统一约束条件下，文件大小4~512MB
多地域下载，每月平均访问频率1000次
（根据论文Triones中数据集特征）

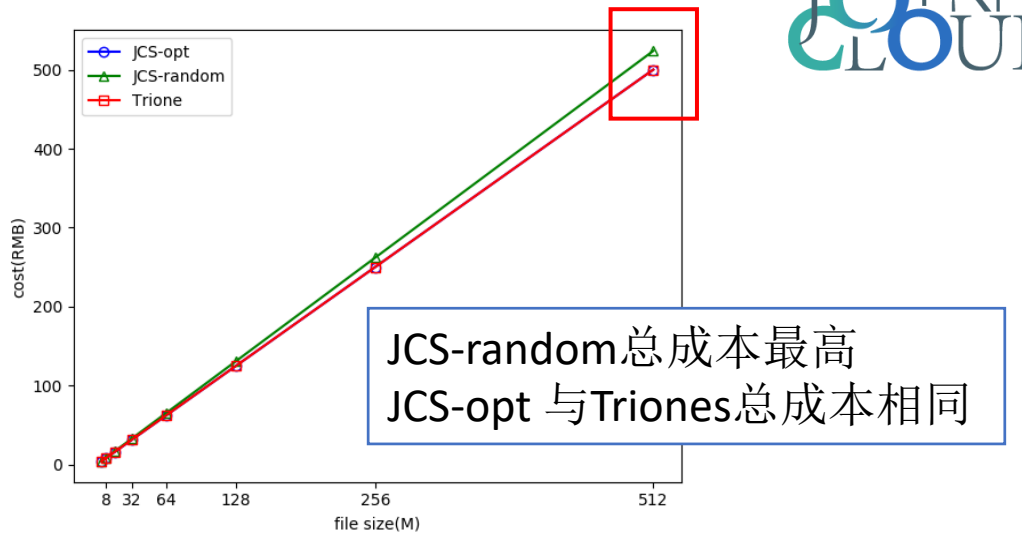
实验设计：
1.数据放置优化策略对比实验
2.数据放置优化算法的性能与准确性
3.针对用户访问模式变化的动态调整的有效性
4.云故障情况下云际存储管理系统的可用性



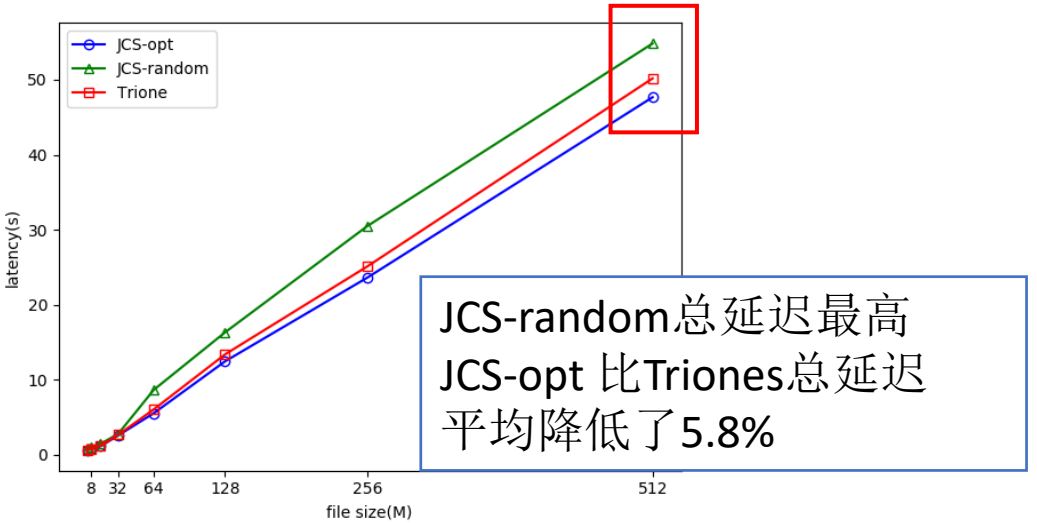
实验分析，1.数据放置优化策略有效性对比

实验目的：
数据放置优化策略的全局成本和延迟的优化效果

实验对比：
北京、深圳两地域下载，
每月下载频率1000次，
每个文件100次下载，测试下载延迟
对比方案：
数据放置优化策略**JCS-opt**
随机放置策略**JCS-random**
论文**Triones**中集中式代理放置策略



总成本对比（成本越低越好）



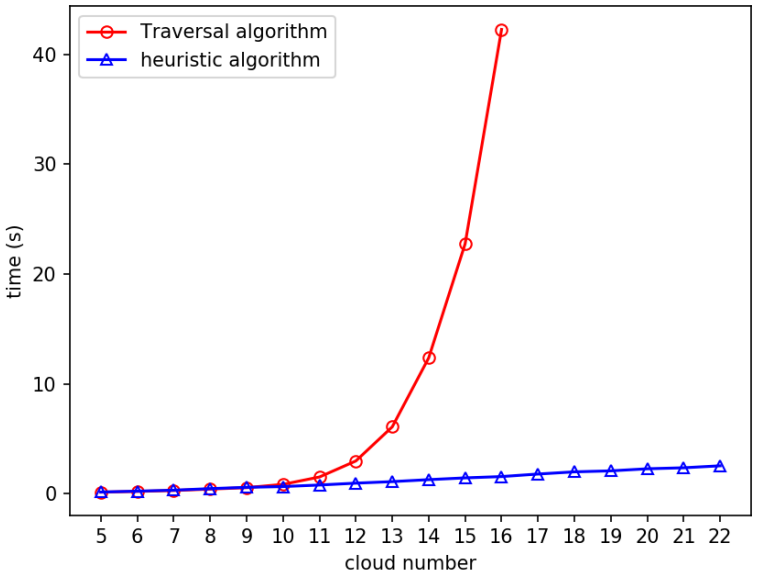
总延迟对比（延迟越低越好）

实验分析，2.数据放置优化算法性能与准确性

实验目的：
对比 **遍历算法**和**启发式算法**的**性能与准确性**

算法时间开销：
随着云个数N的增多，
启发式算法比遍历算法具有明显的时间优势
当N=15表示共有15个可用云时，
遍历算法时间开销是启发式算法的27倍

算法结果准确性：
1000次实验，对比遍历算法和启发式算法结果
启发式算法得到全局最优解的概率为99.5%



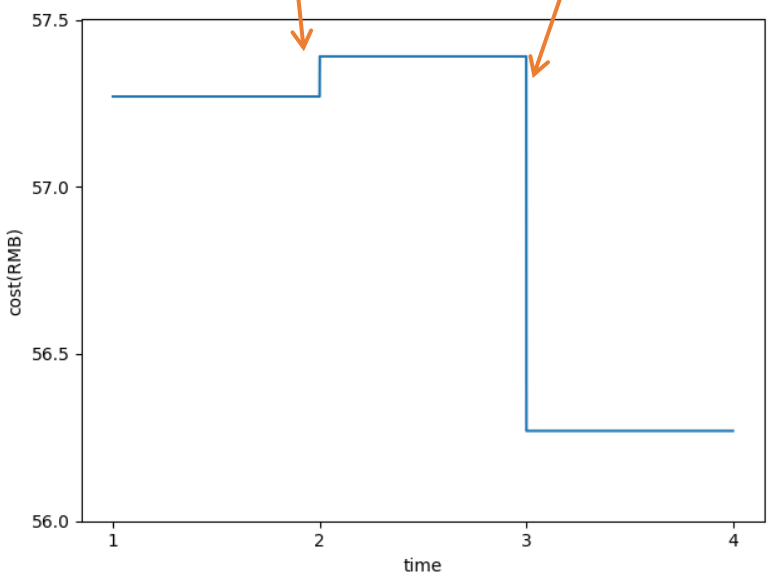
遍历算法和启发式算法时间开销对比

实验分析，3.针对用户访问模式变化的动态调整有效性

实验目的：
验证动态调整能够进一步优化成本和延迟

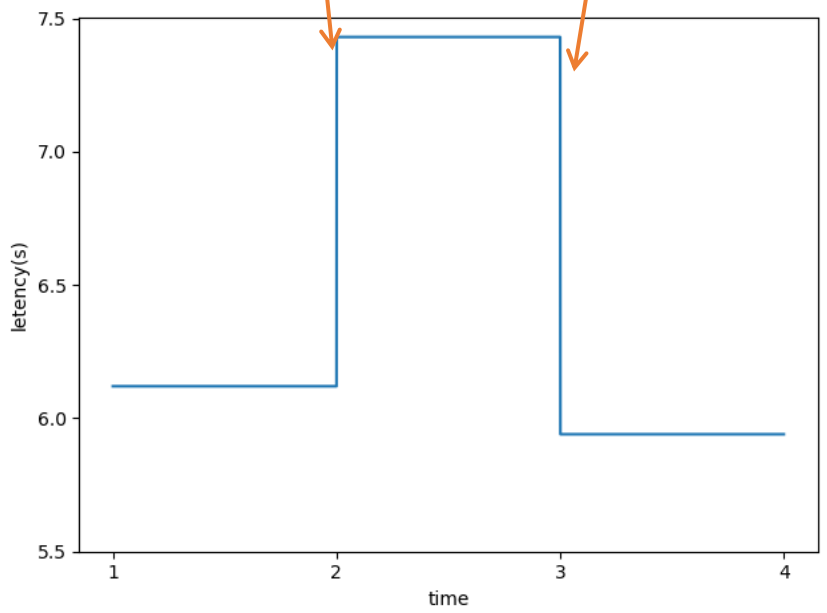
访问模式发生变化，
旧策略存取成本增加

进行动态调整，
新策略相比旧策略成本减少1.9%



访问模式发生变化，
旧策略下载延迟增加

进行动态调整，
新策略相比旧策略延迟减少20%



云际存储管理系统-界面展示

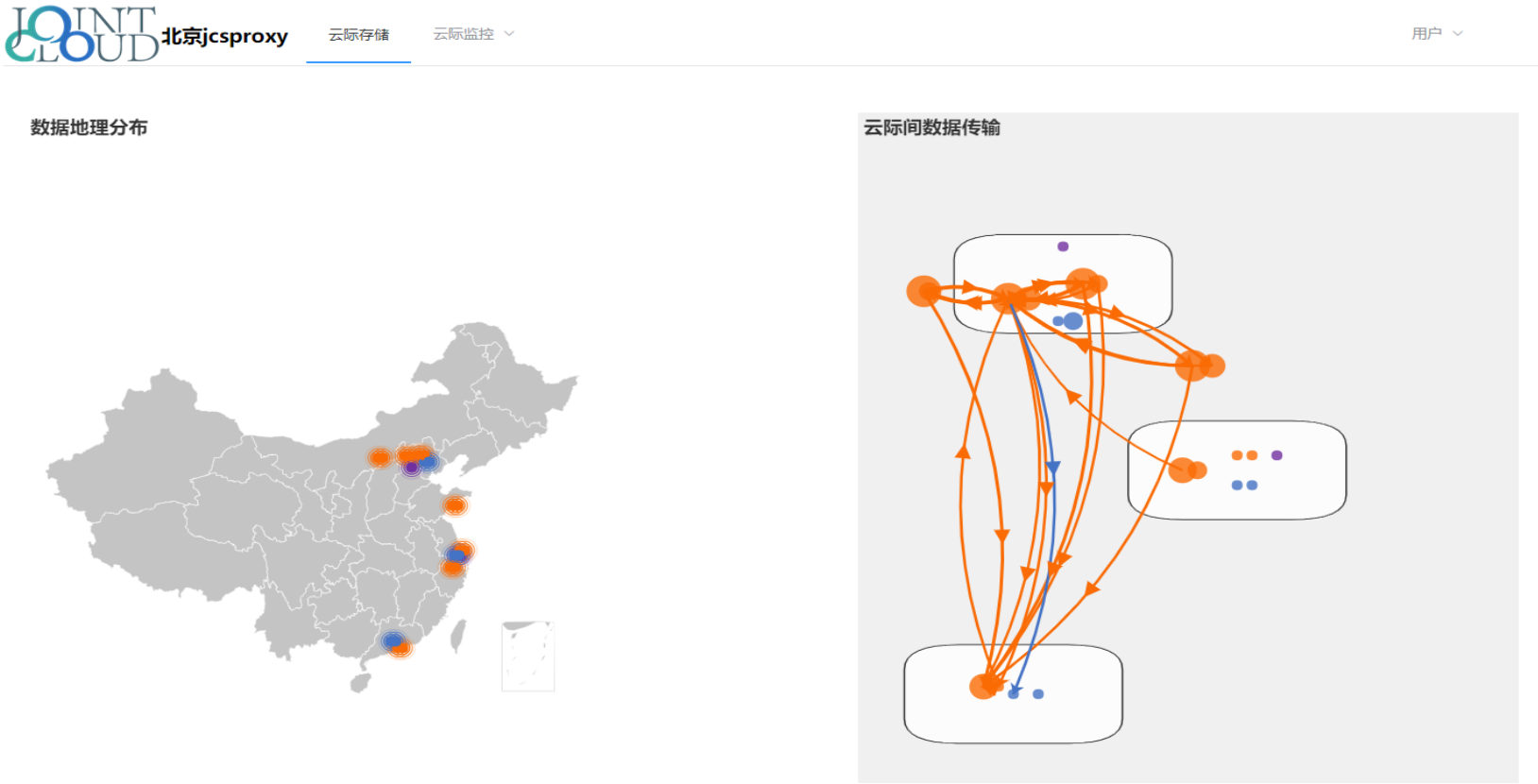
1.云际存储

上传文件、下载文件、删除文件、查询文件、文件列表、新建文件夹、删除文件夹

云际存储管理系统-界面展示

2.云际监控

每个云中的存储量，云际存储代理和多云之间数据流量



内容提要

- 云际存储需求回顾
- 云际存储策略与机制
  - 云际数据放置与动态调整策略
  - 云际环境下分布式流处理系统容错机制
- 总结

背景与意义

- 多云部署带来的数据分散
  - 原因：更多的云、降低成本、防止单云故障，...
  - IDC CloudView survey(2017)：85% 多云, 其中58% 4个云，15% 10个云
  - 不同云上的应用数据，日志等
- 数据源分散
  - 科学计算
  - 物联网（IoT）数据
- 处理分散数据的方法
  - 迁移数据：简单易用，但处理成本高，适合小规模、低频率、低实时性
  - 迁移计算：处理成本低，适合做大规模，复杂逻辑

背景与意义

- 应用分布式流处理系统
  - 可以处理数据源分散
  - 方便处理需要在不同阶段连接多个数据源
  - 实时性

Google MillWheel

  - 现有流处理系统更多的被设计为适用于单云环境

- 环境对比
  - 单云：延迟低(~1ms)，带宽高，全连通，故障率低
  - 云际：单云内部与上述相同，单云之间延迟高(~100ms)，带宽较低，非全连通，网络故障率高

问题描述

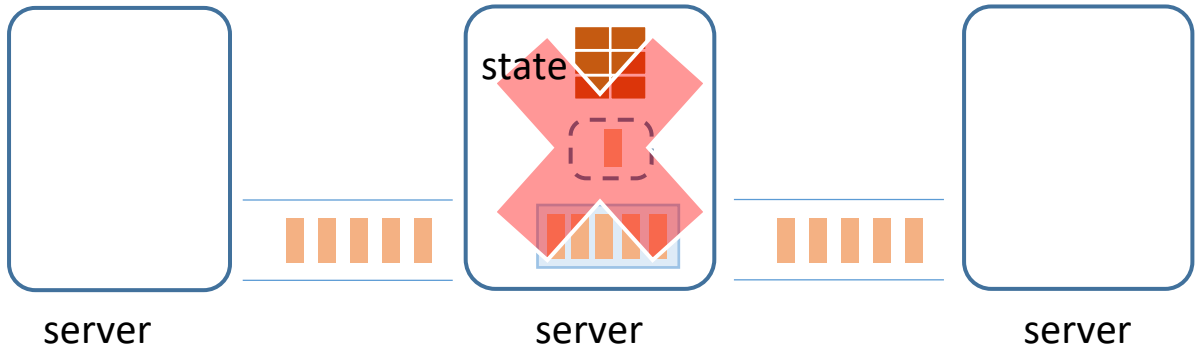
- 故障模型

  fail-stop model

  ✓处理中的数据

  ✓缓存中的数据

  ✓状态数据



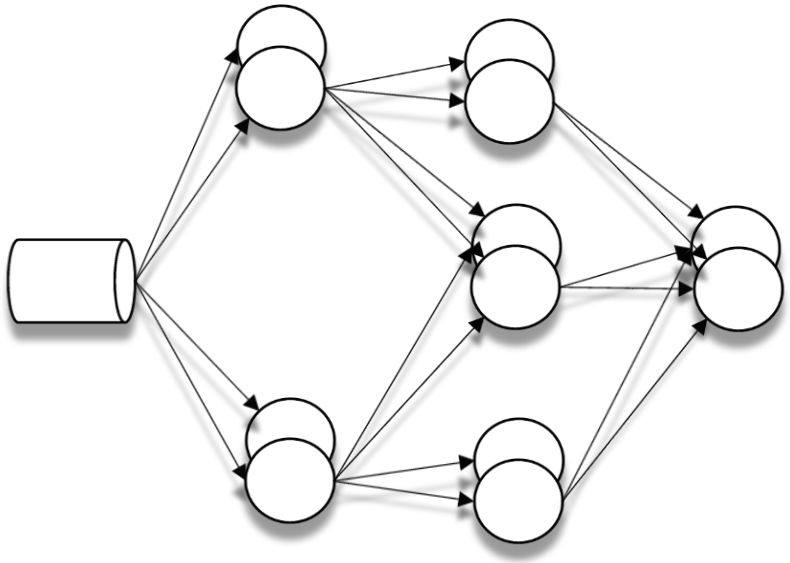
- 故障恢复目标

  向上层应用提供exactly-once的传递保证

研究现状

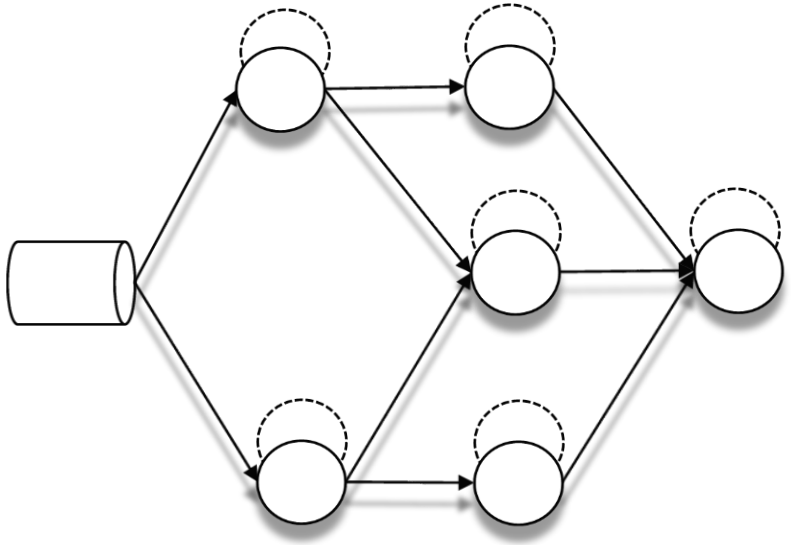
- 常用容错机制

方案1：主动备份



SIGMOD 2005

方案2：被动备份

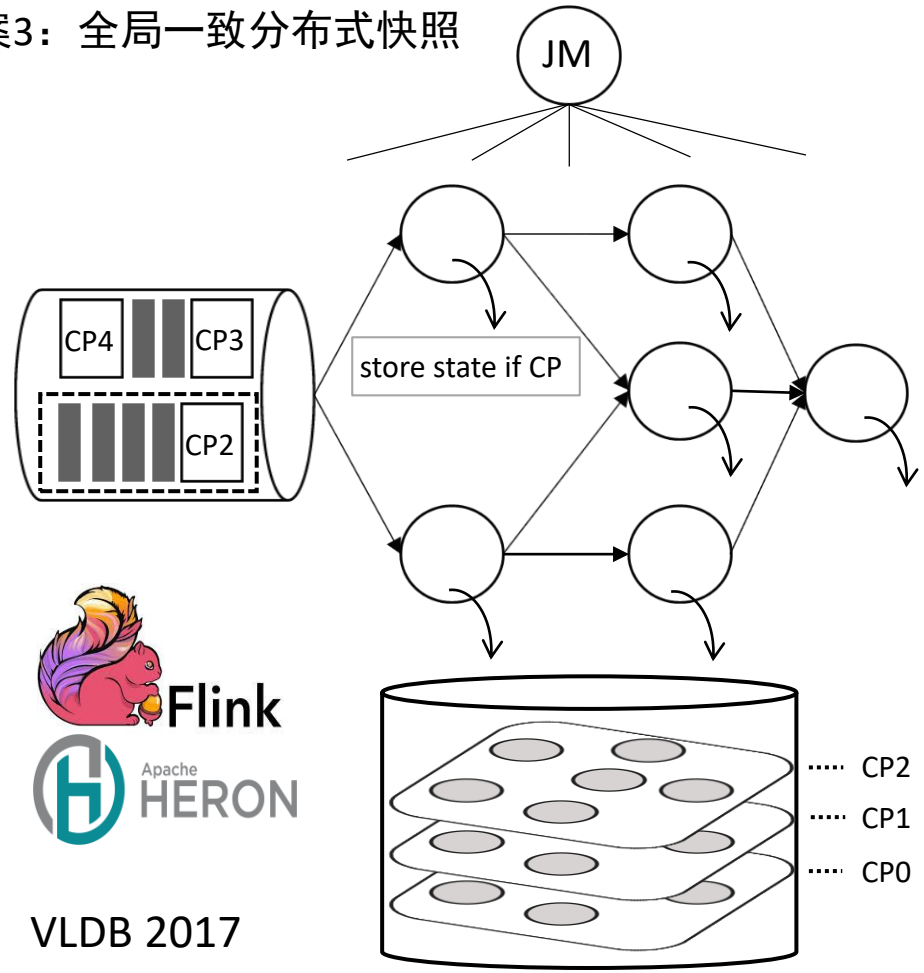


SIGMOD 2013

研究现状

■ 常用容错机制

方案3：全局一致分布式快照

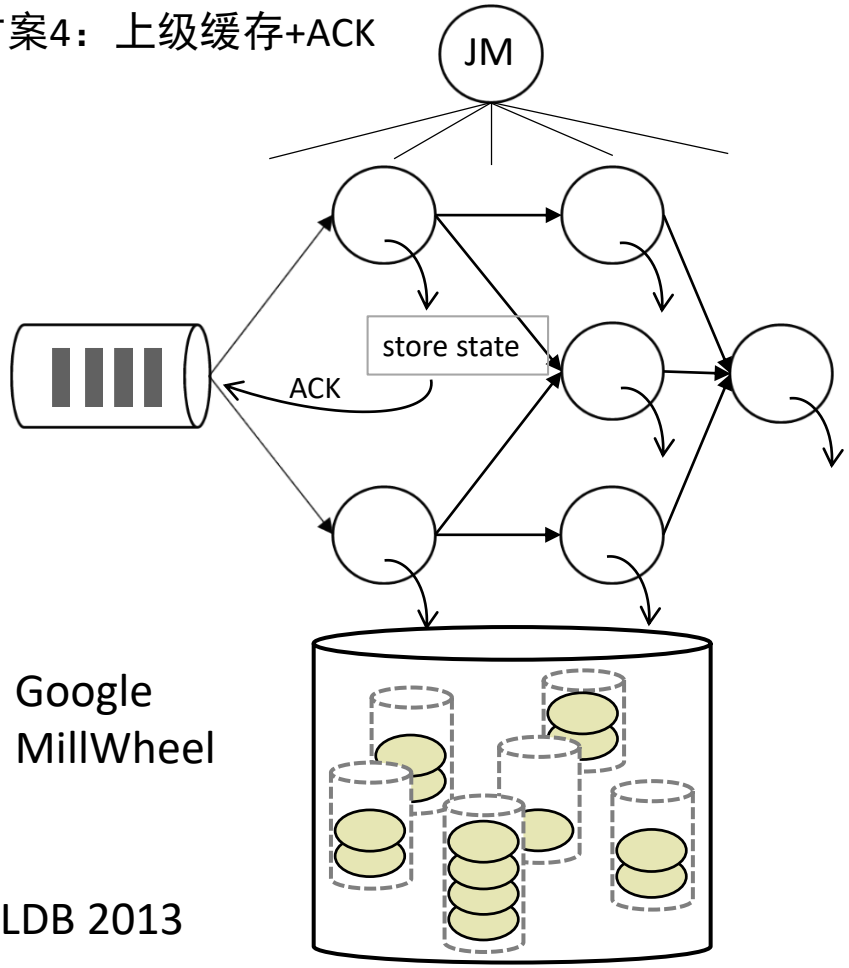


Flink

Apache HERON

VLDB 2017

方案4：上级缓存+ACK

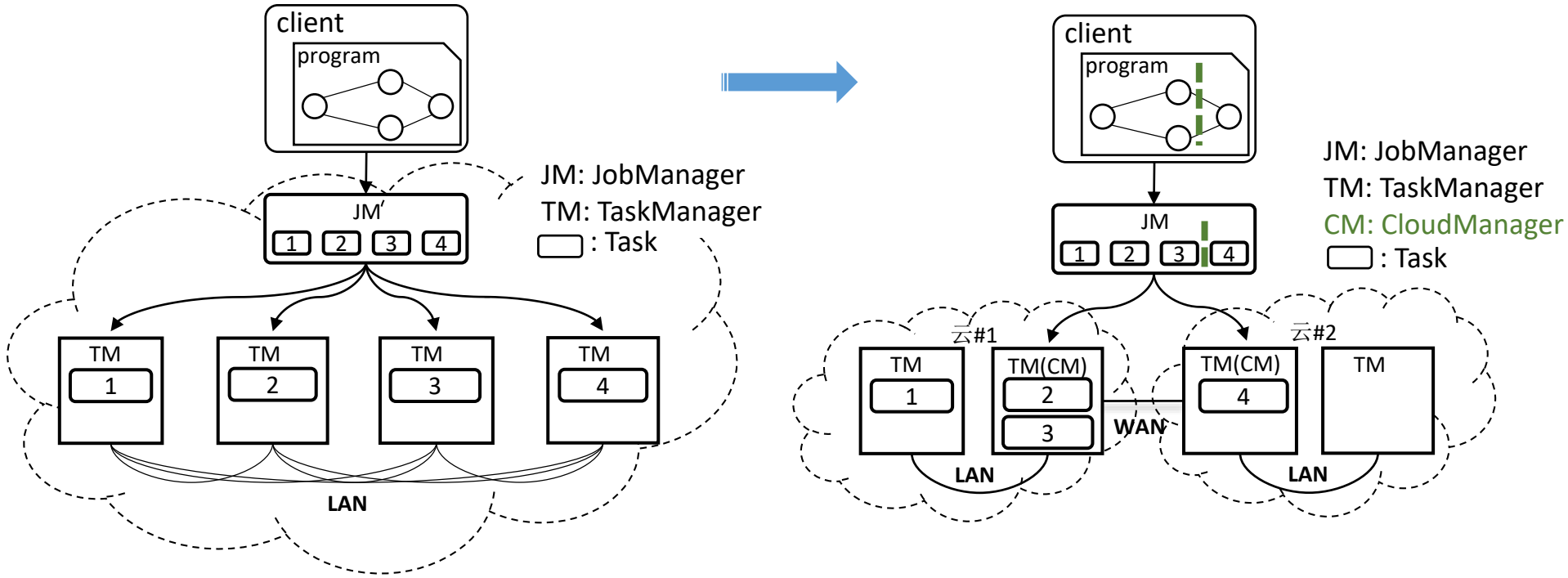


Google MillWheel

VLDB 2013

技术思路

- 1. 基于Apache Flink的流处理任务切分（1）

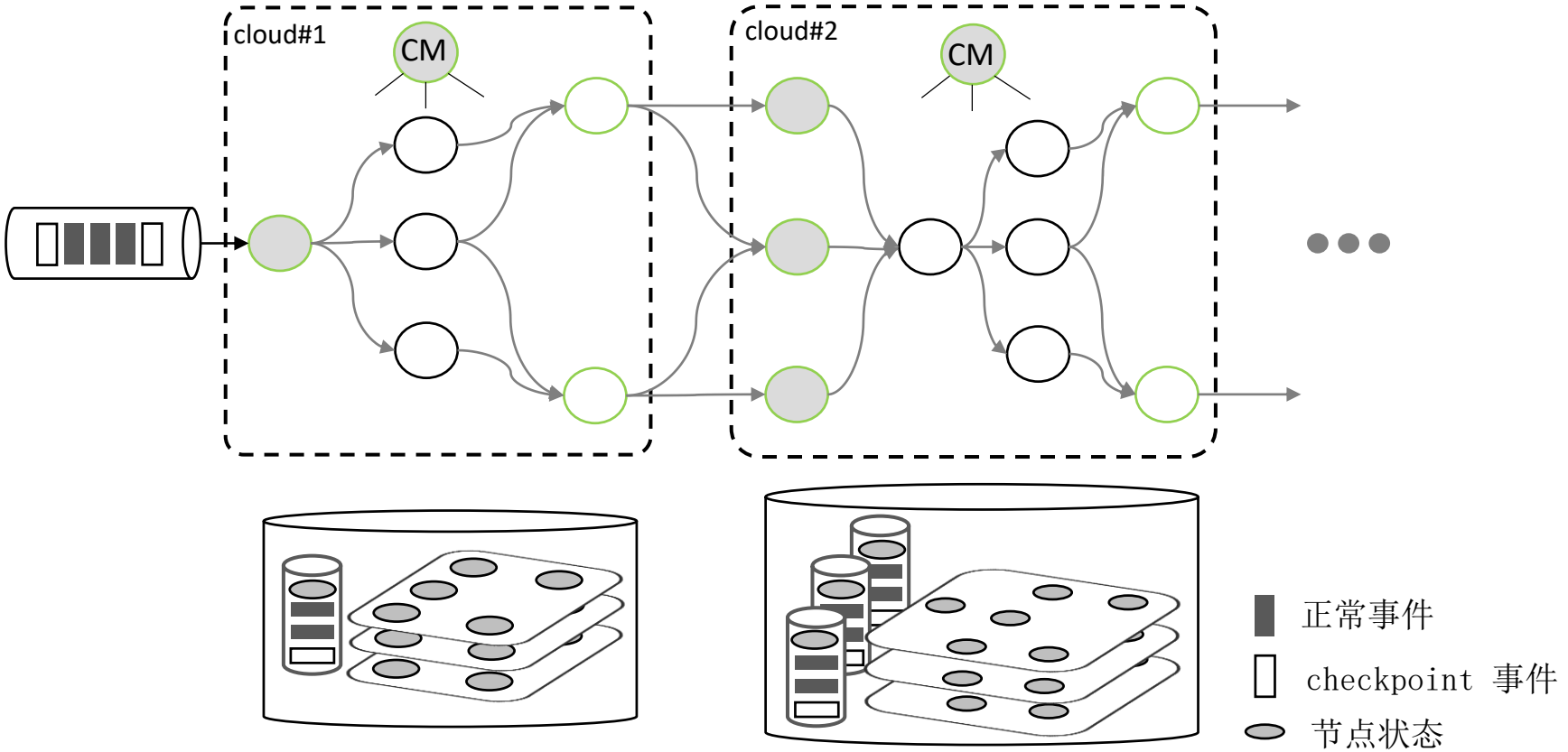


单云任务下发

多云任务下发

技术思路

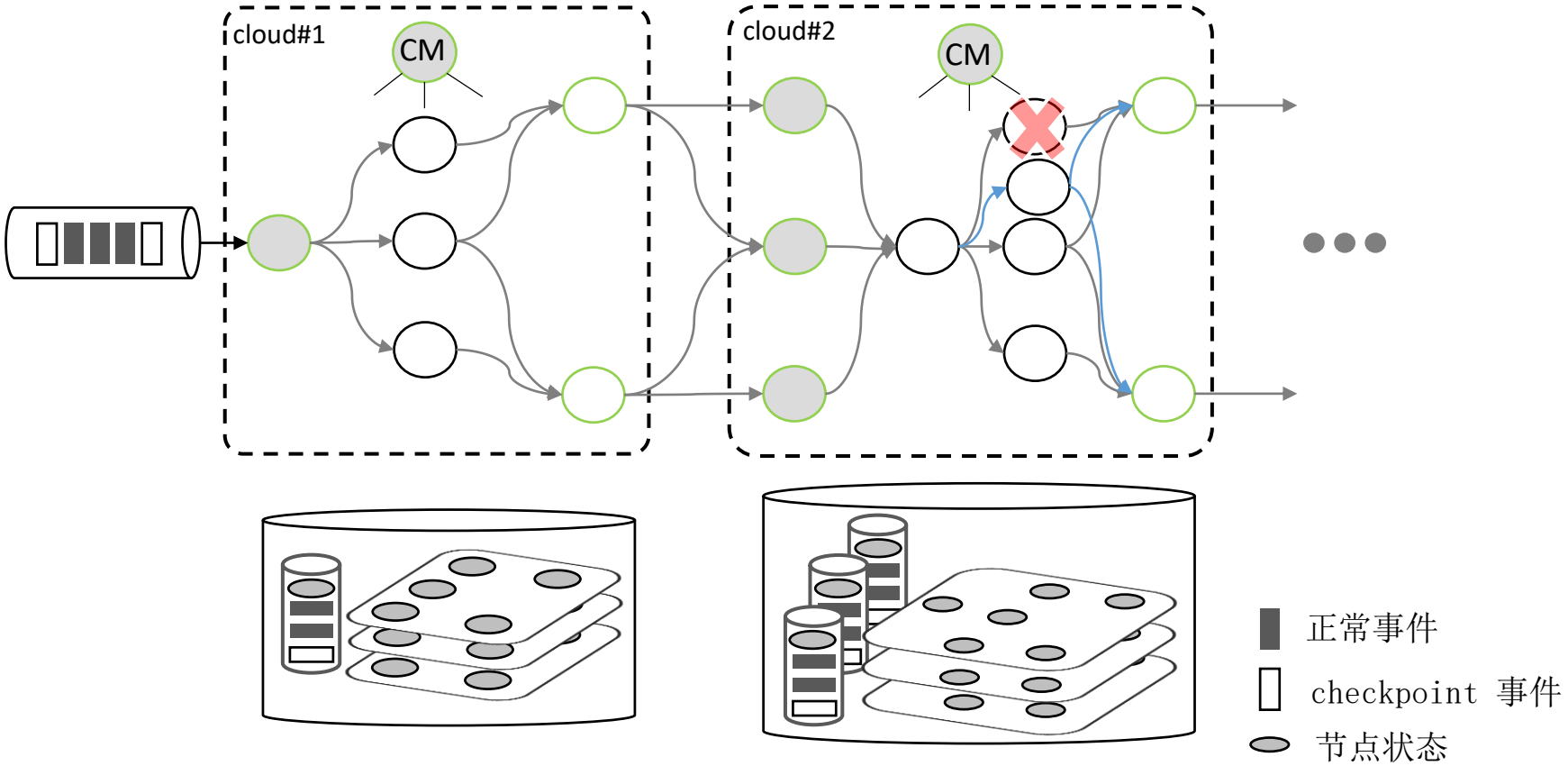
- 2. 基于Apache Flink的分阶段分布式快照（1）



拟采取的容错机制：分阶段分布式快照

技术思路

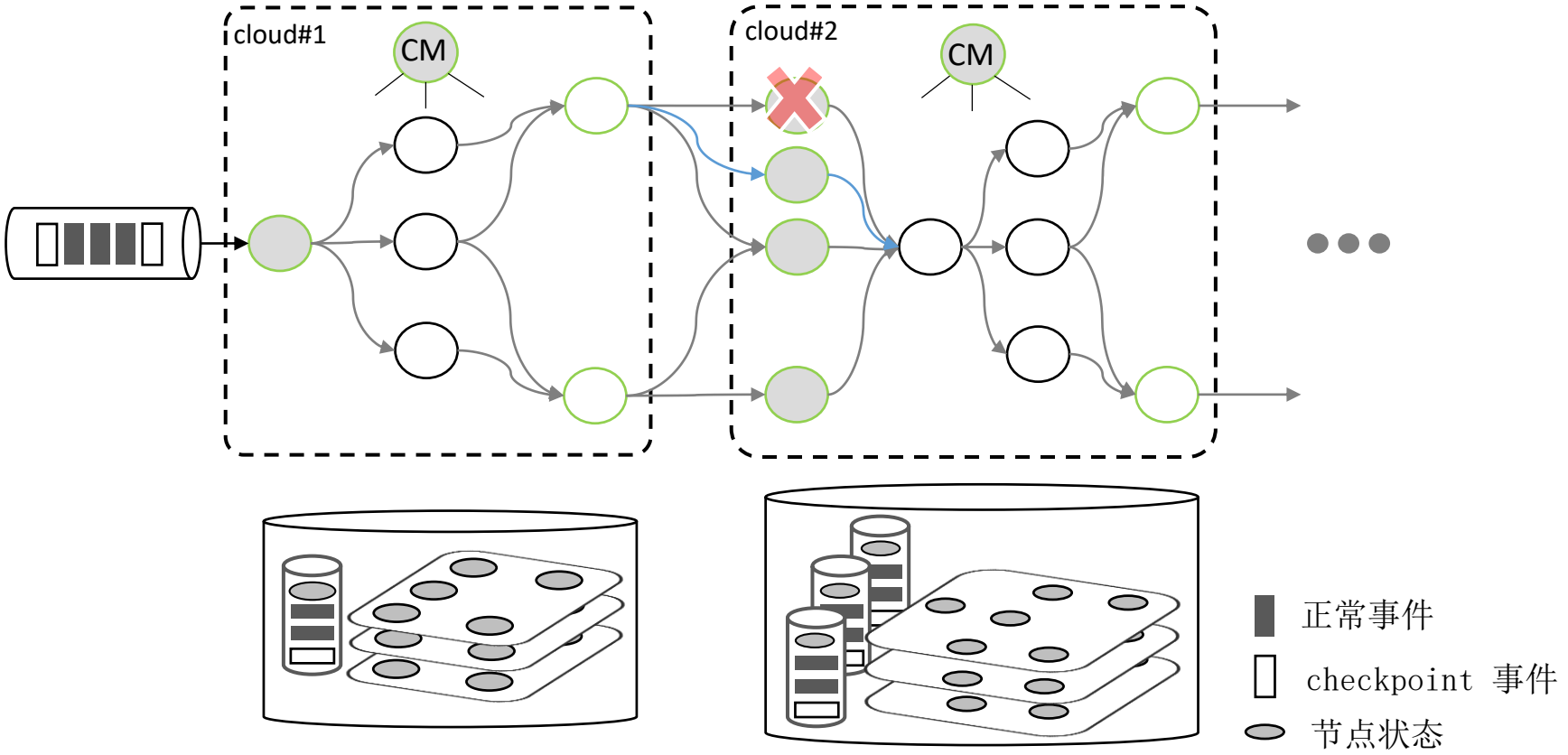
- 2. 基于Apache Flink的分阶段错误恢复（2）



拟采取的容错机制：分阶段分布式快照

技术思路

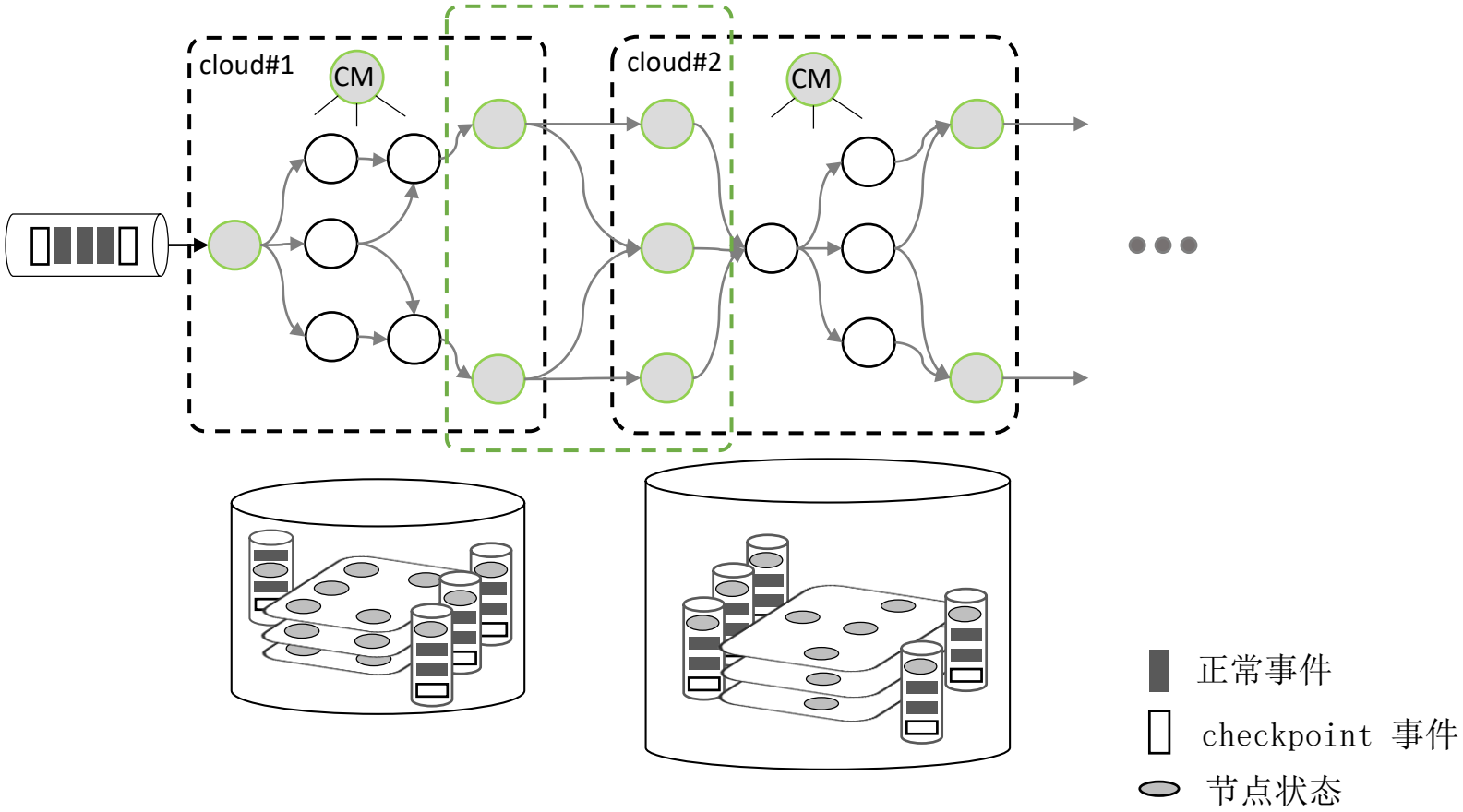
- 2. 基于Apache Flink的分阶段错误恢复（3）



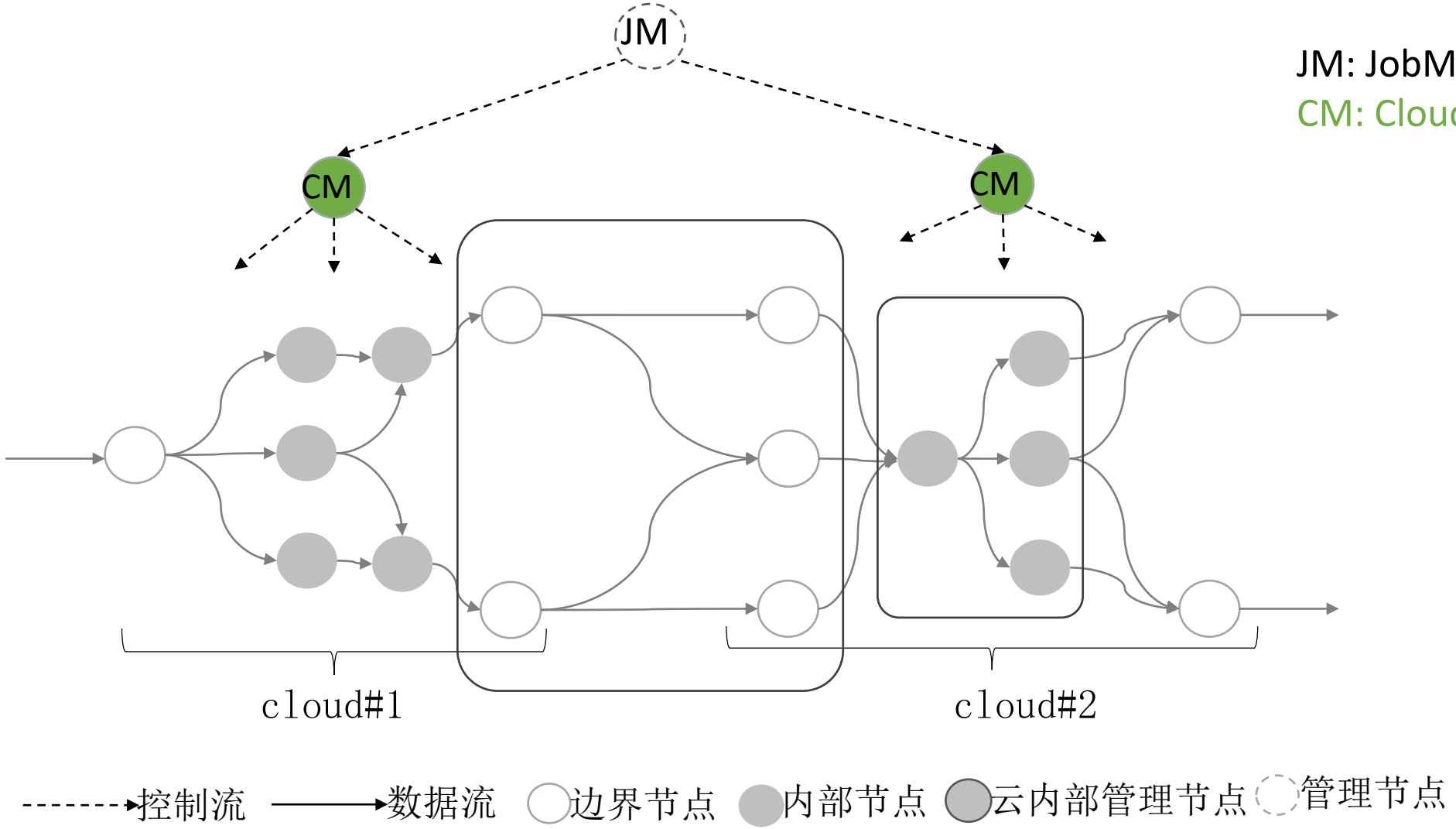
拟采取的容错机制：分阶段分布式快照

技术思路

- 3. 阶段间快恢复



系统整体结构



JM: JobManager

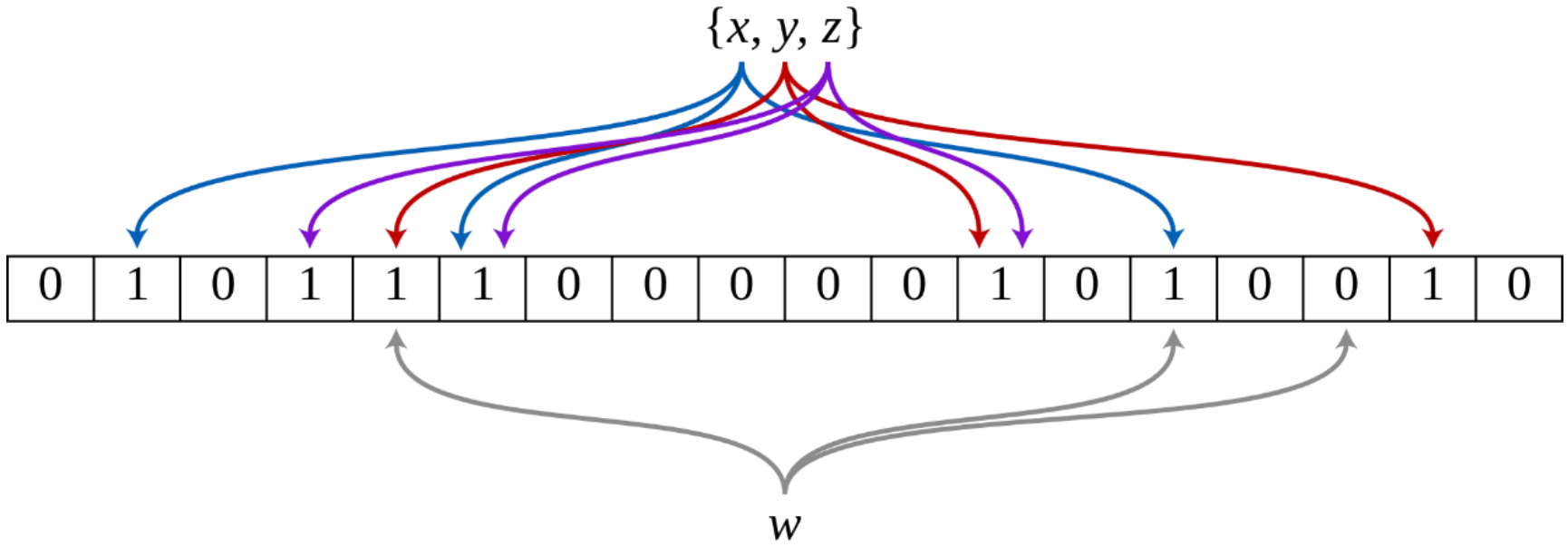
CM: CloudManager

阶段间快速恢复

- 在Flink框架中为每个进入系统的事件增加一个唯一标识ID

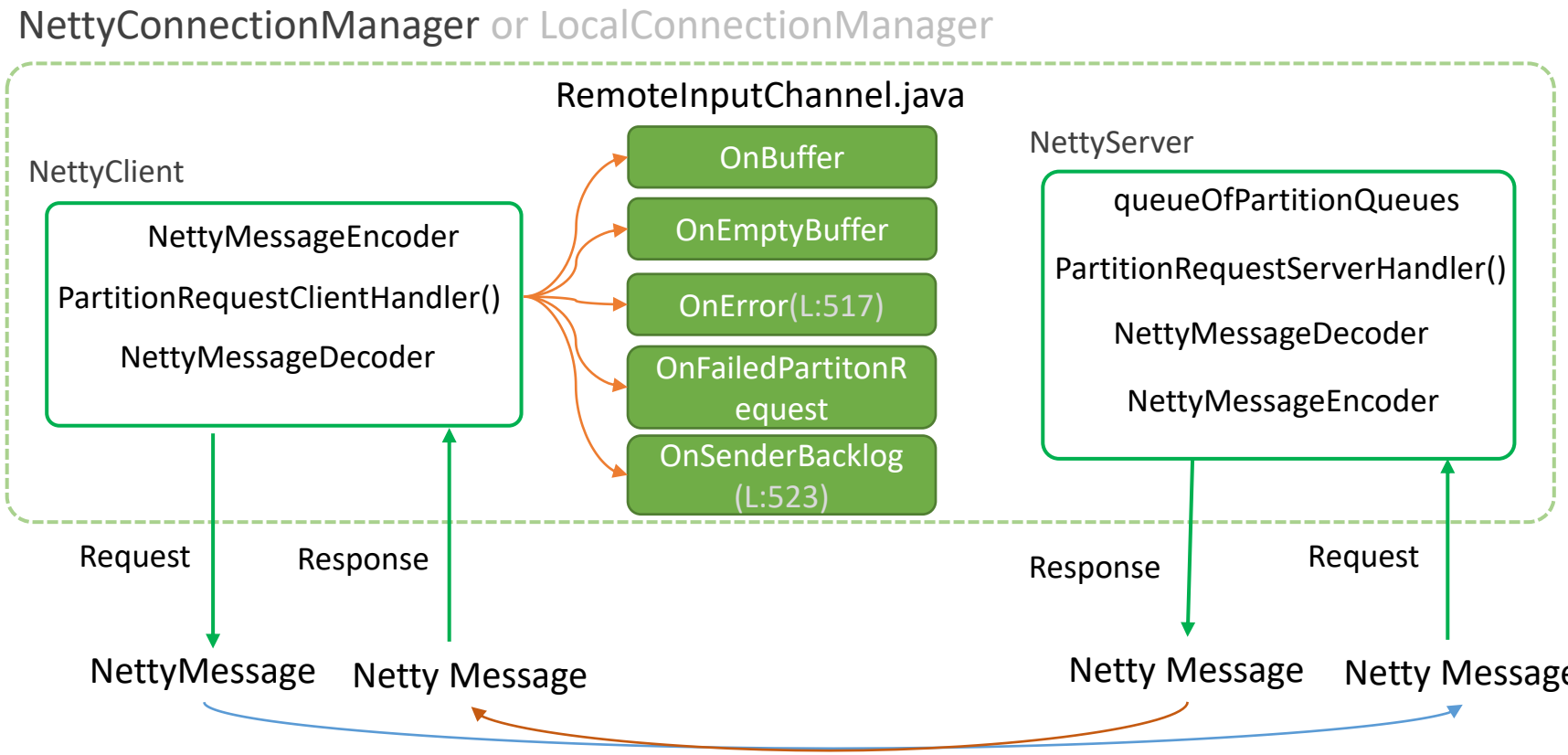
  IP地址、当前进程ID、系统时间戳及一个4byte随机数

- 边界工作节点的处理模块中增加一个Bloom Filter，过滤重复事件



阶段间快速恢复

- Apache Flink现有的消息系统中增加一个确认消息及相应的处理逻辑



abstract class RuntimeEvent

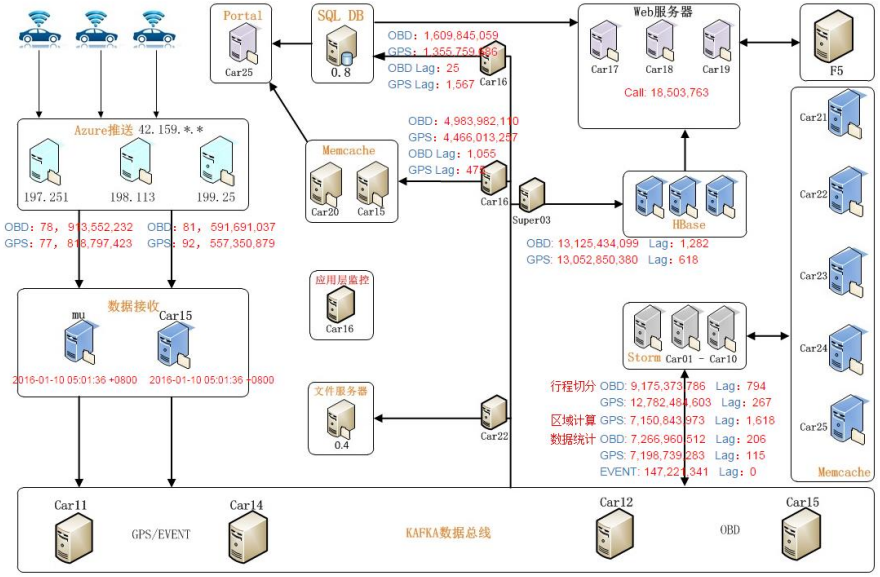
1. CancelCheckpointMarker
2. CheckpointBarrier
3. EndOfPartitionEvent
4. EndOfSuperstepEvent
5. ACKEvent

NettyMessage:org.apache.flink.runtime.io.network.netty.NettyMessage:L236

1. BufferResponse
2. PartitionRequest
3. TaskEventRequest()
4. ErrorResponse
5. CancelPartitionRequest
6. CloseRequest
7. AddCredit

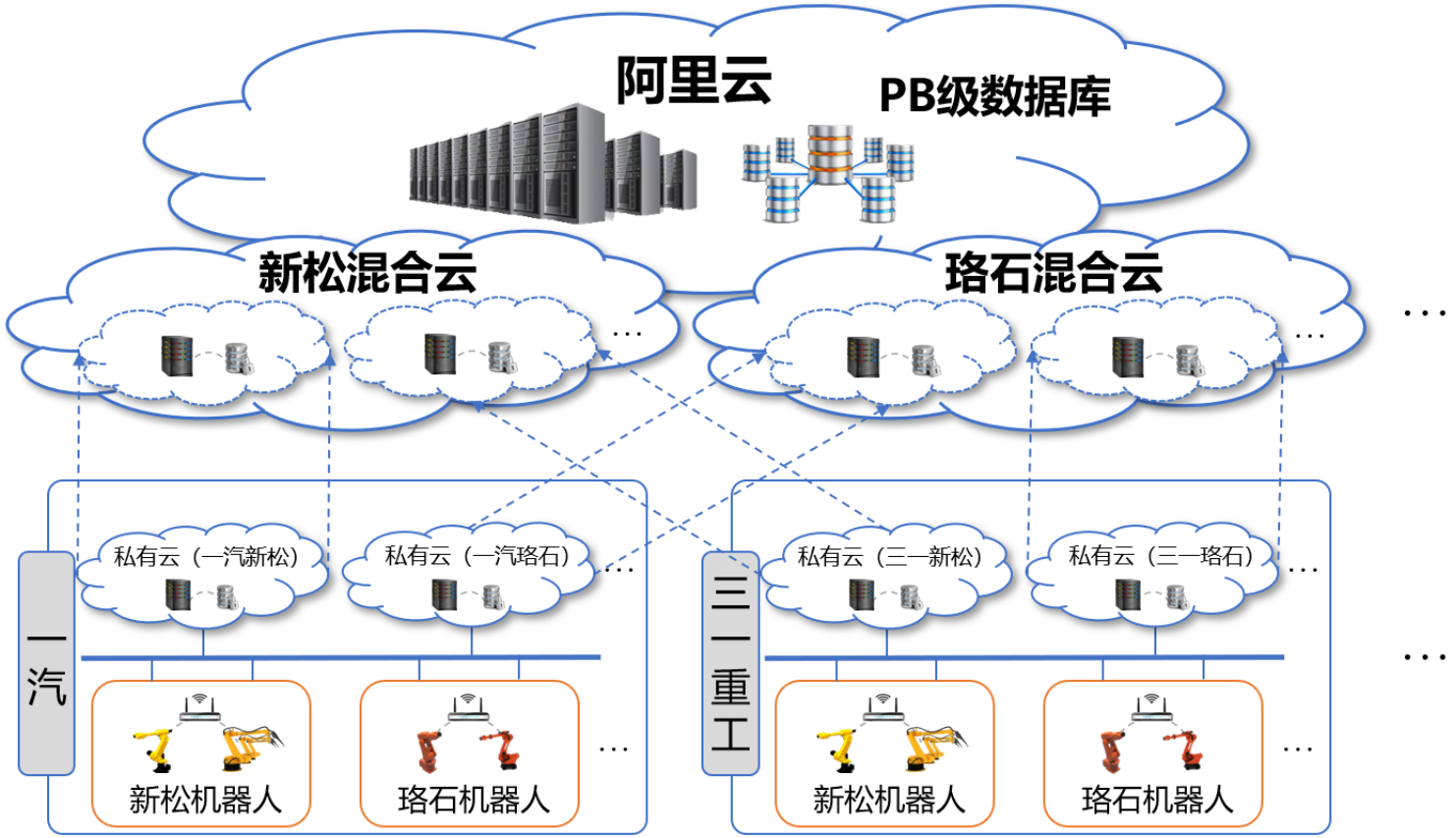
应用场景

- 车联网大数据流式处理
  - 数据属地管理
  - 数据应用跨域集成服务



应用场景——工业大数据上云

- PB级实时流式数据
- 云边端协同
- 预测性维护+工艺优化



内容提要

- 云际存储需求回顾
- 云际存储策略与机制
  - 云际数据放置与动态调整策略
  - 云际环境下分布式流处理系统容错机制
- 总结

总结

- 软件定义云际存储
  - 策略 + 机制
- 策略：建模、表示与生成
  - 表达什么、怎么表示、如何生成
  - 数据访问体验（延迟）、可靠性、价格成本
- 机制：如何执行策略
  - 纠删码、动态调整、冗余副本、流式故障恢复
- 下一步：
  - 策略：自动策略生成、（跨云边界）策略协商
  - 机制：对接云内软件定义机制

课题编号：2016YFB1000103

JOINT CLOUD

国家重点研发计划项目

敬请批评指正