(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 113378219 A
(43)申请公布日 2021.09.10

(21)申请号 202110631709.1

(22)申请日 2021.06.07

(71)申请人 北京许继电气有限公司
地址 100085 北京市海淀区上地信息产业基地信息中路3号

(72)发明人 敬俭国 李立宇 侯振

(74)专利代理机构 北京立成智业专利代理事务所(普通合伙) 11310
代理人 吕秀丽

(51)Int.Cl.
G06F 21/62(2013.01)
G06F 21/60(2013.01)
G06F 16/30(2019.01)
G06F 16/80(2019.01)
G06F 16/242(2019.01)
G06F 16/25(2019.01)
G06F 16/17(2019.01)
G06F 16/182(2019.01)
G06F 11/14(2006.01)

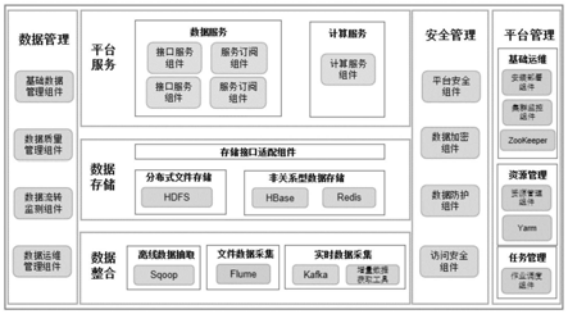
权利要求书2页 说明书7页 附图2页

(54)发明名称

一种非结构化数据的处理方法和系统

(57)摘要

本发明实施例提供了一种非结构化数据的处理方法和系统;所述系统包括:Hadoop生态系统、处理平台;其中所述Hadoop生态系统包括HDFS模块、HBase模块、MapReduce模块、HIVE模块、Sqoop模块、可视化组件;其中所述处理平台包括:数据整合层、数据存储模块、平台服务模块、数据管理模块、安全管理模块。



CN 113378219 A

1.一种非结构化数据的处理系统,其特征在于,包括:Hadoop生态系统、处理平台;

其中所述Hadoop生态系统包括HDFS模块、HBase模块、MapReduce模块、HIVE模块、Sqoop模块、可视化组件;

HDFS模块,该模块集成了kerberos,以实现数据存取访问的身份验证并提供数据存储安全;结合非结构化、半结构化数据的特点,封装优化数据块、数据切割、数据冗余等存储策略,实现高效、高吞吐的数据存储性能;封装分布式数据批量存储接口,融合分布式数据计算,支撑离线计算存取需求;封装分布式数据存储服务,满足业务应用非结构化、半结构化数据的存储需求。

HBase模块,该模块封装小文件存储接口,用户可透明化的进行大文件、小文件的存储,并自动路由存取适配介质;且提供二级索引技术,实现多维数据的高效低延时查询;封装分布式列式存储接口,融合分布式数据计算,支撑实时计算存取需求;封装分布式列表存储服务,满足业务应用非结构化、半结构化、结构化的低延时的存储需求;

MapReduce模块,该模块集成kerberos,实现数据离线计算访问提交的身份验证,提供集群的安全性;封装优化数据IO操作、数据压缩等策略,优化离线计算性能;封装任务调度策略,满足不同场景的离线计算任务调度需求;封装离线任务接口,满足高效离线计算需求;封装分布式离线任务服务,满足业务应用批量离线计算的需求;

HIVE模块,该模块封装了HIVE组件事务操作接口以提供事务能力;封装JDBC操作接口以满足业务应用数据操作;

Sqoop模块,该模块封装离线数据抽取接口以满足数据整合需求;

可视化组件,该组件统一可视化组件参数标准以适配可视化设计器;封装完善可视化组件库以提供丰富的业务指标数据展现;

其中所述处理平台包括:数据整合层、数据存储模块、平台服务模块、数据管理模块、安全管理模块;

数据整合层,其采用实时消息队列、离线数据抽取工具、文件数据采集工具、增量数据库捕获工具,导入非结构化数据,对各类数据按照统一数据规范进行标准化、格式转换及关联处理后,采用分布式文件、非关系型数据库等存储技术进行存储;

数据存储模块,基于x86服务器集群,采用关系数据库PostgreSQL、分布式文件系统HDFS、分布式列式数据库HBase、内存数据库等存储技术,构建关系型数据存储、非关系型数据存储、分布式文件存储等数据存储体系,存储结构化数据、准实时数据、非结构化数据、半结构化数据,提高数据存储的横向扩展能力与高并发条件下的快速数据响应能力,满足数据准实时存储需求;

平台服务模块,通过API、JDBC技术对存储、计算、分析展现进行统一的接口封装以提供统一的服务;

数据管理模块,通过基础数据管理组件、数据质量管理组件、数据流转监测组件和数据运维管理组件,实现统一的数据管理;

平台管理模块,用于进行集群监控和网络监控;采用开源组件ZooKeeper提供分布式应用程序协调服务;采用开源组件Yarn提供统一的资源管理和调度;采用开源组件Quartz作为作业调度引擎;并在此基础上封装形成安装部署组件、资源管理组件和作业调度组件。

安全管理模块,用于构建平台安全和数据安全组件。

2.根据权利要求1所述的非结构化数据的处理系统，其特征在于，其中：

其中所述处理平台用于提供以下数据接入接口：用于传输非结构化文件和实时消息队列的API接口、CIS接口服务、JDBC抽取\写入接口、数据库实时复制接口、日志采集接口、离线数据抽取接口；且所述处理平台还用于构建各个存储组件间流转提供常用的数据操作功能，以通过简单易用的数据流转组件提供数据接入方式、数据流转处理的操作节点和流程化配置功能，以按需构建不同的数据处理链路，通过选择不同的数据接入方式及数据流转功能来构建数据流水线以满足不同场景的数据处理需求；

其中所述处理平台还用于提供数据服务，包括：数据操作语法规范：新增类SQL的语义解析运行引擎、数据缓存及数据路由功能，支撑各类数据统一的存储与访问；数据接口服务：用于根据各类接口的应用需求制定统一的服务格式和访问协议，对外提供统一数据服务；数据服务引擎：用于实现对访问接口输入的数据服务报文进行统一的报文解析、加密、解密处理，为数据服务应用提供技术支撑；以及数据服务监测：用于实现心跳监测，准实时监测各项接口稳定性，并对异常接口进行告警；

其中所述处理平台还用于提供数据安全服务：所述处理平台通过数据加密、访问安全、数据审计、数据防护等功能模块的开发，构建大数据平台从数据接入、数据存储、数据访问、审计全过程的的数据安全防护体系。

3.一种利用如权利要求1-2任一项所述的系统进行非结构化数据处理的方法，其特征在于，包括：

流计算处理步骤，用于在时间窗口内对系统产生的流动数据到达后不进行存储而直接导入内存进行实时计算；其中所述数据计算在内存中执行，并且流动数据不通过Queue进行持久化；在服务器故障时通过预先定义的备份机器将接管失败的执行；且所述流计算处理步骤还包括：建立调度模型，其中所述调度模型中流计算组件拥有自己的主节点，其中调度模型采用对称结构且没有中心节点，且容错和负载均衡需要依赖分布式协议；其中主节点无状态，具有多个Standby节点，其中每一个节点都向Zookeeper注册；Zookeeper检测任务的存活，进而通知主节点；或本地守护进程感知故障并向主节点汇报；且所述流计算处理步骤还包括负载均衡，以在流量增加时将负载均匀地分流到集群的处理节点；

数据备份和故障恢复步骤，用于在故障发生后，系统根据预先定义的策略进行数据的重放和恢复；其中预先定义的策略包括：被动等待策略、主动等待策略、上游备份策略；

其中被动等待策略包括：主节点进行数据计算，且副本节点处于待命状态；系统会定期地将主节点上的最新的状态备份到副本节点上；当出现故障时，系统从备份数据中进行状态恢复；

其中主动等待策略包括：系统在为主节点传输数据的同时，也为副本节点传输一份数据副本；其中主节点为主进行数据计算；当主节点出现故障时，副本节点完全接管主节点的工作；其中主节点和副节点需要分配同样的系统资源；

其中上游备份策略包括：每个主节点均记录其自身的状态和输出数据到日志文件；当主节点出现故障后，上游主节点会重放日志文件中的数据到相应副节点进行数据的重新计算。

一种非结构化数据的处理方法和系统

技术领域

[0001] 本发明属于涉及信息技术领域,尤其涉及一种非结构化数据的处理方法和系统。

背景技术

[0002] 数据,在企业中扮演的角色越来越重要。经过多年的信息化建设,很多企业已略有数据资产规模,同时要想保持长远的发展,还需要协调组织、利用现有的数据沉淀经验,并构建一个协同的企业生态。但据IDC调查,目前企业结构化数据仅占到全部数据量的20%,其余80%都是以文件形式存在的非结构化和半结构化数据,且这些非结构化数据每年增长率达60%。非结构化数据,顾名思义,是存储在文件系统的信息,包括视频、音频、图片、图像、文档、文本等形式,具有某种特定和持续的价值,这种价值在共享、检索、分析等使用过程中得到放大。如何处理这些海量的非结构化数据,是企业构建协同生态的关键一环其存储和流转。

[0003] 现有技术中大都采用邮件、FTP以及QQ等IM工具为主。这些工具传递文件时速度不稳定,安全性得不到保障,并且无法很好地满足企业中一对多的高频数据传递场景。

发明内容

[0004] 针对现有技术中存在的非结构化数据处理存在的问题,本公开实施例提出了一种非结构化数据的处理方法和系统。

[0005] 为了解决上述问题,本公开实施例提出了一种非结构化数据的处理系统,包括:Hadoop生态系统、处理平台;

[0006] 其中所述Hadoop生态系统包括HDFS模块、HBase模块、MapReduce模块、HIVE模块、Sqoop模块、可视化组件;

[0007] HDFS模块,该模块集成了kerberos,以实现数据存取访问的身份验证并提供数据存储安全;结合非结构化、半结构化数据的特点,封装优化数据块、数据切割、数据冗余等存储策略,实现高效、高吞吐的数据存储性能;封装分布式数据批量存储接口,融合分布式数据计算,支撑离线计算存取需求;封装分布式数据存储服务,满足业务应用非结构化、半结构化数据的存储需求。

[0008] HBase模块,该模块封装小文件存储接口,用户可透明化的进行大文件、小文件的存储,并自动路由存取适配介质;且提供二级索引技术,实现多维数据的高效低延时查询;封装分布式列式存储接口,融合分布式数据计算,支撑实时计算存取需求;封装分布式列表存储服务,满足业务应用非结构化、半结构化、结构化的低延时的存储需求;

[0009] MapReduce模块,该模块集成kerberos,实现数据离线计算访问提交的身份验证,提供集群的安全性;封装优化数据IO操作、数据压缩等策略,优化离线计算性能;封装任务调度策略,满足不同场景的离线计算任务调度需求;封装离线任务接口,满足高效离线计算需求;封装分布式离线任务服务,满足业务应用批量离线计算的需求;

[0010] HIVE模块,该模块封装了HIVE组件事务操作接口以提供事务能力;封装JDBC操作

接口以满足业务应用数据操作；

[0011] Sqoop模块，该模块封装离线数据抽取接口以满足数据整合需求；

[0012] 可视化组件，该组件统一可视化组件参数标准以适配可视化设计器；封装完善可视化组件库以提供丰富的业务指标数据展现；

[0013] 其中所述处理平台包括：数据整合层、数据存储模块、平台服务模块、数据管理模块、安全管理模块；

[0014] 数据整合层，其采用实时消息队列、离线数据抽取工具、文件数据采集工具、增量数据库捕获工具，导入非结构化数据，对各类数据按照统一数据规范进行标准化、格式转换及关联处理后，采用分布式文件、非关系型数据库等存储技术进行存储；

[0015] 数据存储模块，基于x86服务器集群，采用关系数据库PostgreSQL、分布式文件系统HDFS、分布式列式数据库HBase、内存数据库等存储技术，构建关系型数据存储、非关系型数据存储、分布式文件存储等数据存储体系，存储结构化数据、准实时数据、非结构化数据、半结构化数据，提高数据存储的横向扩展能力与高并发条件下的快速数据响应能力，满足数据准实时存储需求；；

[0016] 平台服务模块，通过API、JDBC技术对存储、计算、分析展现进行统一的接口封装以提供统一的服务；

[0017] 数据管理模块，通过基础数据管理组件、数据质量管理组件、数据流转监测组件和数据运维管理组件，实现统一的数据管理；

[0018] 平台管理模块，用于进行集群监控和网络监控；采用开源组件ZooKeeper提供分布式应用程序协调服务；采用开源组件Yarn提供统一的资源管理和调度；采用开源组件Quartz作为作业调度引擎；并在此基础上封装形成安装部署组件、资源管理组件和作业调度组件。

[0019] 安全管理模块，用于构建平台安全和数据安全组件。

[0020] 其中所述处理平台用于提供以下数据接入接口：用于传输非结构化文件和实时消息队列的API接口、CIS接口服务、JDBC抽取\写入接口、数据库实时复制接口、日志采集接口、离线数据抽取接口；且所述处理平台还用于构建各个存储组件间流转提供常用的数据操作功能，以通过简单易用的数据流转组件提供数据接入方式、数据流转处理的操作节点和流程化配置功能，以按需构建不同的数据处理链路，通过选择不同的数据接入方式及数据流转功能来构建数据流水线以满足不同场景的数据处理需求；

[0021] 其中所述处理平台还用于提供数据服务，包括：数据操作语法规范：新增类SQL的语义解析运行引擎、数据缓存及数据路由功能，支撑各类数据统一的存储与访问；数据接口服务：用于根据各类接口的应用需求制定统一的服务格式和访问协议，对外提供统一数据服务；数据服务引擎：用于实现对访问接口输入的数据服务报文进行统一的报文解析、加密、解密处理，为数据服务应用提供技术支撑；以及数据服务监测：用于实现心跳监测，准实时监测各项接口稳定性，并对异常接口进行告警；

[0022] 其中所述处理平台还用于提供数据安全服务：所述处理平台通过数据加密、访问安全、数据审计、数据防护等功能模块的开发，构建大数据平台从数据接入、数据存储、数据访问、审计全过程的的数据安全防护体系。

[0023] 同时，本发明实施例还提出了一种应用如前任一项所述的系统进行非结构化数据

处理的方法,包括:

[0024] 流计算处理步骤,用于在时间窗口内对系统产生的流动数据到达后不进行存储而直接导入内存进行实时计算;其中所述数据计算在内存中执行,并且流动数据不通过Queue进行持久化;在服务器故障时通过预先定义的备份机器将接管失败的执行;且所述流计算处理步骤还包括:建立调度模型,其中所述调度模型中流计算组件拥有自己的主节点,其中调度模型采用对称结构且没有中心节点,且容错和负载均衡需要依赖分布式协议;其中主节点无状态,具有多个Standby节点,其中每一个节点都向Zookeeper注册;Zookeeper检测任务的存活,进而通知主节点;或本地守护进程感知故障并向主节点汇报;且所述流计算处理步骤还包括负载均衡,以在流量增加时将负载均匀地分流到集群的处理节点;

[0025] 数据备份和故障恢复步骤,用于在故障发生后,系统根据预先定义的策略进行数据的重放和恢复;其中预先定义的策略包括:被动等待策略、主动等待策略、上游备份策略;

[0026] 其中被动等待策略包括:主节点进行数据计算,且副本节点处于待命状态;系统会定期地将主节点上的最新的状态备份到副本节点上;当出现故障时,系统从备份数据中进行状态恢复;

[0027] 其中主动等待策略包括:系统在为主节点传输数据的同时,也为副本节点传输一份数据副本;其中主节点为主进行数据计算;当主节点出现故障时,副本节点完全接管主节点的工作;其中主节点和副节点需要分配同样的系统资源;;

[0028] 其中上游备份策略包括:每个主节点均记录其自身的状态和输出数据到日志文件;当主节点出现故障后,上游主节点会重放日志文件中的数据到相应副节点进行数据的重新计算。

[0029] 本发明的有益效果在于:本公开实施例的技术方案提出了一种非结构化数据的处理方法和系统,能够对非结构化数据进行有效的管理,将数据进行统一存储、聚合。通过开放API接口,整合归集各业务系统或其他第三方来源的文件,完成聚合工作,也就完成企业的协同环境搭建的关键一步,团队可以在系统中共享到最新的文档,并以文件为桥梁,缩短同事的协作成本。

附图说明

[0030] 图1是本公开实施例的非结构化数据的处理方法的原理图;

[0031] 图2是被动等待策略的原理图;

[0032] 图3是主动等待策略的原理图;

[0033] 图4是上游备份策略的原理图。

具体实施例

[0034] 为了使本发明的目的、技术方案及优点更加清楚明白,以下结合附图,对本发明进一步详细说明。应当理解,此处所描述的具体实施例仅仅用以解释本发明,并不用于限定本发明。

[0035] 下面结合附图及本发明的实施例对后评估的方法进一步说明。

[0036] 本发明实施例提出了一种非结构化数据的处理方法和系统,其原理如图1所示。

[0037] 为了满足非结构化数据的存储、管理需求,本公开实施例的技术方案的整体技术

架构采用Hadoop平台为核心进行构建，同时针对Hadoop生态系统中的第三方开源软件进行升级及自主的封装与完善，具体包括：

[0038] 1.HDFS：

[0039] 1.1、集成kerberos，实现数据存取访问的身份验证，提供数据存储安全；

[0040] 1.2、结合非结构化、半结构化数据的特点，封装优化数据块、数据切割、数据冗余等存储策略，实现高效、高吞吐的数据存储性能；

[0041] 1.3、封装分布式数据批量存储接口，融合分布式数据计算，支撑离线计算存取需求；

[0042] 1.4、封装分布式数据存储服务，满足业务应用非结构化、半结构化数据的存储需求。

[0043] 2.HBase:

[0044] 2.1、封装小文件存储接口，用户可透明化的进行大文件、小文件的存储，并自动路由存取适配介质。

[0045] 2.2、提供二级索引技术，实现多维数据的高效低延时查询。

[0046] 2.3、封装分布式列式存储接口，融合分布式数据计算，支撑实时计算存取需求。

[0047] 2.4、封装分布式列表存储服务，满足业务应用非结构化、半结构化、结构化的低延时的存储需求。

[0048] 3.MapReduce

[0049] 3.1、集成kerberos，实现数据离线计算访问提交的身份验证，提供集群的安全性；

[0050] 3.2、封装优化数据IO操作、数据压缩等策略，优化离线计算性能；

[0051] 3.3、封装任务调度策略，满足不同场景的离线计算任务调度需求；

[0052] 3.4、封装离线任务接口，满足高效离线计算需求；

[0053] 3.5、封装分布式离线任务服务，满足业务应用批量离线计算的需求。

[0054] 4.HIVE

[0055] 4.1、封装HIVE组件事务操作接口，提供事务能力；

[0056] 4.2、封装JDBC操作接口，满足业务应用数据操作。

[0057] 5.Sqoop

[0058] 5.1、封装离线数据抽取接口，满足数据整合需求。

[0059] 6.可视化组件

[0060] 6.1、统一可视化组件参数标准，适配可视化设计器；

[0061] 6.2、封装完善可视化组件库，提供丰富的业务指标数据展现。

[0062] 主要功能描述：

[0063] 1.数据整合层采用实时消息队列、离线数据抽取工具、文件数据采集工具、增量数据库捕获工具等多种技术手段，导入非结构化数据、，对各类数据按照统一数据规范进行标准化、格式转换及关联处理后，采用分布式文件、非关系型数据库等存储技术进行存储。

[0064] 2.数据存储，基于x86服务器集群，采用关系数据库PostgreSQL、分布式文件系统HDFS、分布式列式数据库HBase、内存数据库等存储技术，构建关系型数据存储、非关系型数据存储、分布式文件存储等数据存储体系，存储结构化数据、准实时数据、非结构化数据、半结构化数据，提高数据存储的横向扩展能力与高并发条件下的快速数据响应能力，满足数

据准实时存储需求

[0065] 3.平台服务,通过API、JDBC等技术对存储、计算、分析展现进行统一的接口封装,提供统一的服务。

[0066] 4.数据管理,通过研发基础数据管理组件、数据质量管理组件、数据流转监测组件和数据运维管理组件,实现统一的数据管理。

[0067] 5.平台管理,进行集群监控和网络监控;采用开源组件ZooKeeper提供分布式应用程序协调服务;采用开源组件Yarn提供统一的资源管理和调度;采用开源组件Quartz作为作业调度引擎;并在此基础上封装形成安装部署组件、资源管理组件和作业调度组件。

[0068] 6.安全管理,构建平台安全和数据安全组件。

[0069] 关键技术说明:

[0070] 1.流计算处理技术:

[0071] 一种高实时性的计算技术。是指当一定时间窗口内应有系统产生的流动数据到达后不进行存储,而是将流式数据直接导入内存进行实时计算,从流动的、无序的数据中获取有价值的信息输出。流计算具备分布式、低延迟、高性能、可扩展、高容错、高可靠、消息严格有序、定制开发等特点,流计算适用于对动态产生的数据进行实时计算并及时反馈结果,但往往不要求结果绝对精确的应用场景。

[0072] 数据流处理技术的应用在各行各业中都得到了广泛的认同,包括金融服务、网络监控、电信数据管理、Web应用、生产制造、传感检测等等。对于这些实时性要求很高的应用,通常有两种解决方案:一种是把持续到达的数据简单地放到传统数据库管理系统(DBMS)中,并在其中进行操作。然而该种方案的弊端在于传统的DBMS并不是为快速连续的存放单独的数据单元而设计的,而且也并不支持“持续处理”。与此同时,人们都认识到,“近似性”和“自适应性”是对数据流进行快速查询和其他处理(如数据分析和数据采集)的关键要素,而传统DBMS的主要目标恰恰与之相反:通过稳定的查询设计,得到精确的答案。

[0073] 1.1处理模型

[0074] 在传统的实时数据流处理系统中,业内早期经常使用Queue+Worker的处理方式。系统维护者静态配置Worker和Queue的对应关系,即哪些Worker从哪些Queue中读数据,往哪些Queue中写,如果流量或业务增加,需要扩容Queue或worker时,可能需要重新规划两者的对应关系。为了保证可靠性,Queue通常具有高可用的特性,worker发来的消息都会被Queue持久化保存,而每个queue对每条消息都持久化的开销就会有些高,并且会造成消息处理的延迟增大。阿里、百度和腾讯都有基于这种框架的业务处理系统,甚至Facebook早期也是这么处理的,只不过包括Facebook Puma2、支付宝的海狗等系统使用HBase作为Queue,屏蔽了高可用和扩容方面的问题,但低延迟的问题依然会呈现。

[0075] 为了使得数据流处理系统的延迟变低,所有数据计算必须在内存中执行,并且流动数据不能每条都通过Queue进行持久化,这样子数据处理的高可用又变成了亟需解决的问题。大部分数据流处理的高可用技术基于故障恢复,如果服务器故障,一组预先定义的备份机器将接管失败的执行。通常基于故障恢复的高可用方法有Passive Standby、Active Standby、Upstream Backup三类。

[0076] 1.2调度模型

[0077] 在调度模型方面,流计算组件拥有自己的主节点,而Puma和S4号称对称架构,没有

中心节点。对称结构没有单点,扩展性理论上无限,容错和负载均衡需要依赖分布式协议,例如借助Zookeeper;而主从结构在实现故障恢复和负载均衡上更加容易,缺点是存在单点,可能会造成性能或稳定性方面的问题。仔细分析下,其实数据流系统的主节点不存在单点问题:

[0078] 主节点无状态,可以有多个Standby节点,都向Zookeeper注册,当主节点故障后,自动切换到Standby节点,这点和BigTable类的系统是类似的;

[0079] 主节点并无性能瓶颈,批处理系统由于任务是有生命周期的,因此主节点需要频繁调度任务,规模增大时容易造成主节点的调度压力增大;而数据流系统的任务常驻内存,一旦执行就不退出,即只有启动或故障时才会调度,因此调度并无压力。

[0080] 流计算组件将每个任务都注册到Zookeeper,由Zookeeper去检测任务的存活,进而通知主节点;也有一些系统通过任务的本地守护进程感知故障并向主节点汇报,两者并无本质差异。数据流系统有主节点之后,任务的调度和故障的处理就变得方便了许多,将哪些任务调度到哪些机器去执行,这依赖于系统状态和机器的资源使用等因素。

[0081] 1.3负载分流

[0082] 当流量增加时,如果将负载均匀地分流到集群的处理节点。一般有几种做法,根据流量可以动态地对任务进行分裂,分裂后的多个任务再进行调度;也可以先静态地配置任务的粒度,让每个相关任务处理的流量较少,当发现流量增加时进行任务迁移。后者实现简单,但对资源的使用存在一些浪费;前者实现复杂,对无状态的任务拆分可行,但如果用户定义的任务有自身状态,那么拆分用户状态是非常困难的事情。

[0083] 2.流计算高可用技术

[0084] 批量计算将数据事先存储到持久设备上,节点失效后容易实现数据重放;而流式计算对数据不进行持久化存储。因此,批量计算中的高可用技术不完全适用于流式计算环境,需要根据流式计算新特征及其新的高可用要求,有针对性地研究更加轻量、高效的高可用技术和方法。

[0085] 流计算高可用是通过状态备份和故障恢复策略实现的。当故障发生后,系统根据预先定义的策略进行数据的重放和恢复。按照实现策略,可以细分为被动等待(passive standby)、主动等待(active standby)和上游备份(upstream backup)这3种策略:

[0086] 2.1被动等待策略

[0087] 如图2所示,主节点B进行数据计算,副本节点B℃处于待命状态,系统会定期地将主节点B上的最新的状态备份到副本节点B℃上。出现故障时,系统从备份数据中进行状态恢复。被动等待策略支持数据负载较高、吞吐量较大的场景,但故障恢复时间较长,可以通过对备份数据的分布式存储缩短恢复时间。该方式更适合于精确式数据恢复,可以很好地支持不确定性计算应用,在当前流式数据计算中应用最为广泛。

[0088] 2.2主动等待策略

[0089] 如图3所示,系统在为主节点B传输数据的同时,也为副本节点B℃传输一份数据副本。以主节点B为主进行数据计算,当主节点B出现故障时,副本节点B℃完全接管主节点B的工作,主副节点需要分配同样的系统资源。该种方式故障恢复时间最短,但数据吞吐量较小,也浪费了较多的系统资源。在广域网环境中,系统负载往往不是过大时,主动等待策略是一个比较好的选择,可以在较短的时间内实现系统恢复。

[0090] 2.3上游备份策略：

[0091] 如图4所示，每个主节点均记录其自身的状态和输出数据到日志文件，当某个主节点B出现故障后，上游主节点会重放日志文件中的数据到相应副本节点Bspanlang＝"EN-US"style＝'font-family:Symbol'xml:lang＝"EN-US"＞⊄中，进行数据的重新计算。上游备份策略所占用的系统资源最小，在无故障期间，由于副本节点B⊄保持空闲状态，数据的执行效率很高。但由于其需要较长的时间进行恢复状态的重构，故障的恢复时间往往较长。如当需要恢复时间窗口为30分钟的聚类计算，就需要重放该30分钟内的所有元组。可见，对于系统资源比较稀缺、算子状态较少的情况，上游备份策略是一个比较好的选择方案。

[0092] 由上述描述可以看出，本公开实施例的技术方案着重从数据整合、数据服务、数据安全等方面实现海量非结构化数据的存储管理服务，具体如下：

[0093] 1.数据整合

[0094] 提供丰富数据接入方式，API接口(非结构化文件、实时消息队列)、CIS接口服务、JDBC抽取\写入、数据库实时复制、日志采集、离线数据抽取等方式；构建各个存储组件间流转提供常用的数据操作功能；构建简单易用的数据流转组件，提供数据接入方式、数据流转处理的操作节点和流程化配置功能，按需构建不同的数据处理链路。数据开发人员按需选择不同的数据接入方式及数据流转功能来构建数据流水线，以满足不同场景的数据处理需求。

[0095] 2.数据服务

[0096] 设计数据操作语法规范，新增“类SQL”的语义解析运行引擎、数据缓存及数据路由功能，支撑各类数据统一的存储与访问。

[0097] 数据接口服务：根据各类接口的应用需求制定统一的服务格式和访问协议，对外提供统一数据服务。

[0098] 数据服务引擎：实现对访问接口输入的数据服务报文进行统一的报文解析、加密、解密处理，为数据服务应用提供技术支撑。

[0099] 数据服务监测：实现心跳监测，准实时监测各项接口稳定性，并对异常接口进行告警。

[0100] 3.数据安全

[0101] 通过数据加密、访问安全、数据审计、数据防护等功能模块的开发，构建大数据平台从数据接入、数据存储、数据访问、审计全过程的的数据安全防护体系。

[0102] 以上所述是本发明的优选实施方式，应当指出，对于本技术领域的普通技术人员来说，在不脱离本发明所述原理的前提下，还可以作出若干改进和润饰，这些改进和润饰也应视为本发明的保护范围。

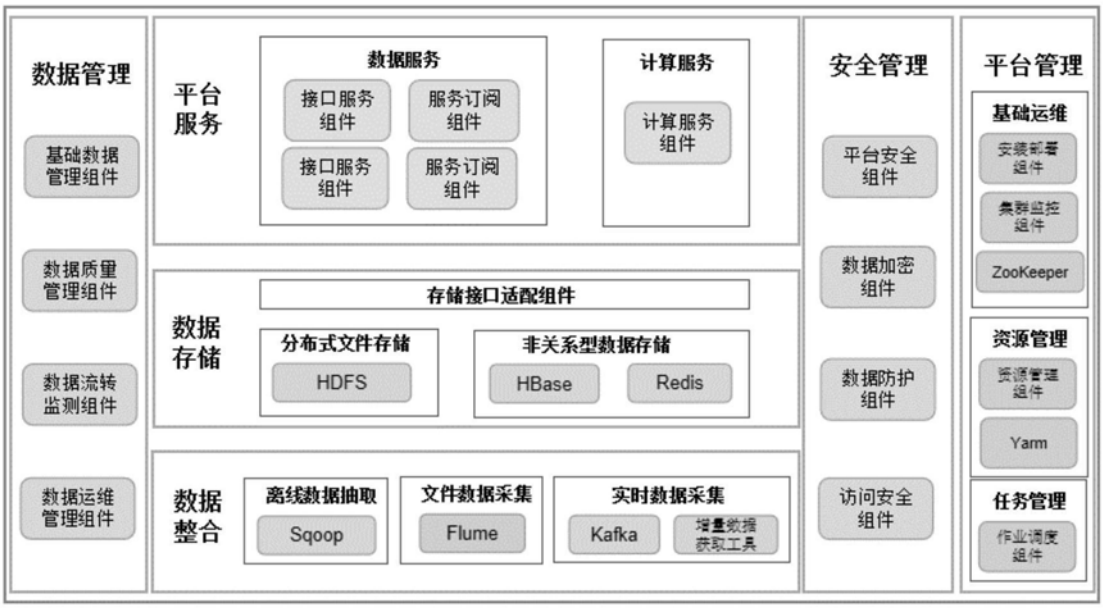

数据管理
基础数据管理组件
数据质量管理组件
数据流转监测组件
数据运维管理组件
平台服务
数据服务
接口服务组件
服务订阅组件
接口服务组件
服务订阅组件
计算服务
计算服务组件
数据存储
存储接口适配组件
分布式文件存储
HDFS
非关系型数据存储
HBase
Redis
数据整合
离线数据抽取
Sqoop
文件数据采集
Flume
实时数据采集
Kafka
增量数据获取工具
安全管理
平台安全组件
数据加密组件
数据防护组件
访问安全组件
平台管理
基础运维
安装部署组件
集群监控组件
ZooKeeper
资源管理
资源管理组件
Yarn
任务管理
作业调度组件

图1

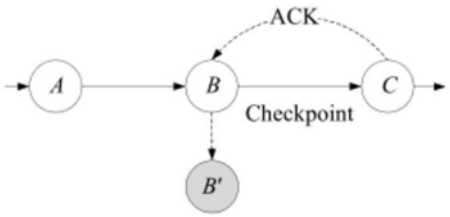

ACK
A
B
C
Checkpoint
B'

图2

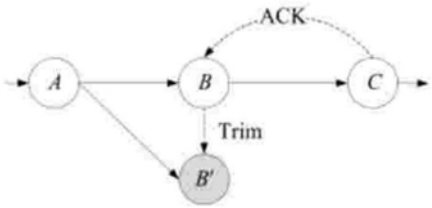

ACK
A
B
C
Trim
B'

图3

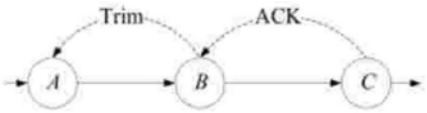

Trim
ACK
A
B
C

图4a

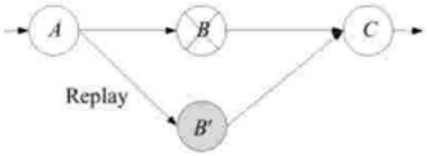


图4b