



(12)发明专利申请

(10)申请公布号 CN 111639082 A

(43)申请公布日 2020.09.08

(21)申请号 202010514803.4

(22)申请日 2020.06.08

(71)申请人 成都信息工程大学

地址 610225 四川省成都市西南航空港经济开发区学府路1段24号

(72)发明人 曹亮 刘魁 李超

(74)专利代理机构 成都金英专利代理事务所 (普通合伙) 51218

代理人 袁英

(51)Int.Cl.

G06F 16/22(2019.01)

G06F 16/27(2019.01)

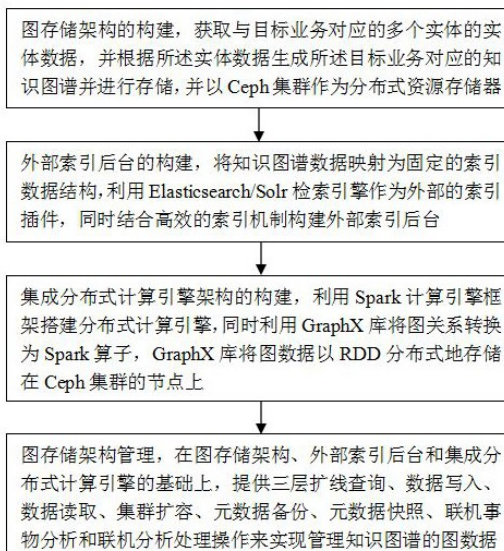
权利要求书2页 说明书6页 附图5页

(54)发明名称

基于Ceph的十亿级节点规模知识图谱的对象存储管理方法及系统

(57)摘要

本发明公开了基于Ceph的十亿级节点规模知识图谱的对象存储管理方法及系统,方法包括:图存储架构的构建设计,获取与目标业务对应的多个实体的实体数据,并根据实体数据生成所述目标业务对应的知识图谱并进行存储,并以Ceph作为分布式资源存储器,同时新增外部索引后台机制,利用分布式计算引擎将大任务分解成多个子任务,分发给不同机器去执行,完成后汇总,以提供大规模数据处理能力来支撑OLAP需求,供用户基于知识图谱进行数据分析。本发明还提供了基于Ceph的十亿级节点规模知识图谱的对象存储管理系统。本方案引用了分布式资源管理器,具有可扩展和高可用特征,同时能存储和表达海量知识,支持数十亿节点数据量,有着可靠、易用、高效的特性。



1. 基于Ceph的十亿级节点规模知识图谱的对象存储管理方法, 其特征在于, 包括以下步骤:

S1, 图存储架构的构建, 获取与目标业务对应的多个实体的实体数据, 并根据所述实体数据生成所述目标业务对应的知识图谱并进行存储, 并以Ceph作为分布式资源存储器, 采用Client/Server架构, 以多个Monitor组成的小集群构建一个Ceph集群, 同时单个Monitor小集群下采用多个OSD存储图数据;

S2, 外部索引后台的构建, 将知识图谱数据映射为固定的索引数据结构, 利用Elasticsearch/Solr检索引擎作为外部的索引插件, 实现非等值查询, 同时结合高效的索引机制构建外部索引后台;

S3, 集成分布式计算引擎架构的构建, 利用Spark计算引擎框架搭建分布式计算引擎, 同时利用GraphX库将图关系转换为Spark算子, GraphX库将图数据以RDD分布式地存储在Ceph集群的节点上, 使用顶点RDD和边RDD分别对应存储顶点集合和边集合;

S4, 图存储架构管理, 在图存储架构、外部索引后台和集成分布式计算引擎构建的基础上, 提供三层扩线查询、数据写入、数据读取、集群扩容、元数据备份、元数据快照、联机事务分析和联机分析处理操作来实现管理知识图谱的图数据。

2. 根据权利要求1所述的基于Ceph的十亿级节点规模知识图谱的对象存储管理方法, 其特征在于, 所述步骤S2中高效的索引机制包括图形索引和顶点中心索引, 图形索引为整个知识图谱的全局索引结构; 顶点中心索引是为每个顶点建立的本地索引结构。

3. 根据权利要求1所述的基于Ceph的十亿级节点规模知识图谱的对象存储管理方法, 其特征在于, 所述步骤S3中还包括分区操作, 具体包括以下子步骤:

S101, 顶点RDD通过按顶点的ID进行哈希分区, 将顶点数据以多分区形式分布在集群上;

S102, 边RDD按指定的分区策略进行分区, 将边数据以多分区形式分布在集群上;

S103, 将记录RDD分区内顶点与所有边RDD分区关系的路由表存储在顶点RDD的分区中, 在边RDD需要顶点数据时, 顶点RDD根据路由表把顶点数据发送至边RDD分区。

4. 根据权利要求1所述的基于Ceph的十亿级节点规模知识图谱的对象存储管理方法, 其特征在于, 所述步骤S4中的数据写入步骤包括以下子步骤:

S201, 客户端连接上Monitor, 获取集群Map信息, 请求对应的主OSD数据节点;

S202, 主OSD数据节点同时写入另外两个副本节点数据, 等待主节点以及另外两个副本节点完成数据写入状态, 主节点及副本节点写入状态都成功后, 返回完成信号给Client, 数据写入完成。

5. 根据权利要求1所述的基于Ceph的十亿级节点规模知识图谱的对象存储管理方法, 其特征在于, 所述步骤S4中的集群扩容步骤包括以下子步骤:

S301, Client连接Monitor获取集群Map信息, 新主节点OSD1向Monitor上传请求, 让OSD2节点接替OSD1节点成为临时主节点;

S302, 临时主节点OSD2将数据全量同步给新主节点OSD1, ClientIO读写直接连接临时主节点OSD2进行数据读写;

S303, 临时主节点OSD2收到读写IO, 同时写入另外两个副本节点中的数据, 临时主节点OSD2以及另外两副本节点中的三份数据都写入成功后, 返回信号给Client, ClientIO读写

完毕；

S304,若节点OSD1数据同步完毕,临时主节点OSD2向Monitor上传请求,交出主节点角色,OSD1节点重新成为主节点,OSD2节点变成副本节点；

S305,同时在图数据层面,实现节点扩容后,按照图数据切割方式将图数据切割,分别存储到多台机器上。

6.根据权利要求4所述的基于Ceph的十亿级节点规模知识图谱的对象存储管理方法,其特征在于,所述图数据切割方式包括按点切割和按边切割两种切割方式;按点切割方式以图的顶点进行数据切割,切割线通过图的顶点,每一条边只保存一次,并且每一条边只出现在一台机器上,邻居顶点多的顶点会被分发到多台不同的机器上存储;按边切割方式以图的边进行数据切割,切割线只穿过连接顶点的边,每一个顶点只保存一次,切断的边分发到多台不同到的机器上存储。

7.根据权利要求1所述的基于Ceph的十亿级节点规模知识图谱的对象存储管理方法,其特征在于,所述步骤S4中的元数据快照步骤包括:根据元数据信息有效的恢复到以前的数据状态,也可恢复程序到系统运行历史状态;保存特定时间点的系统数据,生成系统相应时间点报告;导出快照数据进行离线工作。

8.根据权利要求1所述的基于Ceph的十亿级节点规模知识图谱的对象存储管理方法,其特征在于,所述步骤S4中的三层扩线查询步骤包括以下子步骤:

S401,将用户给定顶点集合Vset设置为第一层扩线查询的基础数据,设置第一层的查询过滤条件为顶点Label/顶点属性的过滤条件ConditionA,进行第一层的顶点扩线查询;

S402,将满足第一层查询过滤条件边的顶点集合作为第二层扩线查询的基础数据,同时设置第二层的查询过滤条件为边Label/边属性的过滤条件ConditionB,并进行第二层的边扩线查询;

S403,将满足第二层查询过滤条件的边集合作为第三层扩线查询的基础数据,设置属性查询条件,进行第三层的属性扩线查询,输出经过三层扩线查询的查询结果。

9.基于Ceph的十亿级节点规模知识图谱的对象存储管理系统,其特征在于,包括

图数据存储模块,用于分布式存储大规模知识图谱的对象数据,提供对象存储、块设备存储和文件系统服务;

分布式计算模块,用于利用SparkRDD内存计算,将大任务分解成多个子任务,分别部署到不同的机器上执行,完成后汇总,以提供高效的大规模数据处理能力,来支撑OLAP需求,供用户基于知识图谱进行数据分析;

索引模块,用于将知识数据映射为固定的索引数据结构,为用户提供图形索引、顶点中心索引和外部索引功能;

元数据管理模块,用于元数据的备份、元数据快照、程序恢复、生成时间点报告和系统离线工作。

基于Ceph的十亿级节点规模知识图谱的对象存储管理方法及系统

技术领域

[0001] 本发明涉及信息处理技术领域,尤其涉及基于Ceph的十亿级节点规模知识图谱的对象存储管理方法及系统。

背景技术

[0002] 知识图谱,是一种用可视化技术描述知识资源及其载体,挖掘、分析、构建、绘制和显示知识及它们之间的相互联系。知识图谱可以将大规模数据中的隐藏的知识抽取出来,构建一个基于图的数据模型。近些年的数据挖掘、大数据、人工智能、机器学习等等和信息处理关联的热门技术,可以用知识图谱来辅助,这些技术的最终目的,实质是把数据收集整理成结构化的、可复用、可推理的存储,进而用于更多的使用场景,而知识图谱这种存储格式,可以近乎完美的匹配这些需求。知识图谱旨在描述真实世界中存在的各种实体或概念,以及他们之间的关联关系,它的每一个实体用全局唯一确定的ID来标识,就如每个人都有身份证号码;第二个就是用属性-值对来刻画实体的内在特性,用关系来连接两个实体,刻画他们之间的关联。

[0003] 目前图存储系统最大的缺陷在于并非真正的分布式,大数据时代可获取的数据越来越多,单机的容量有限,当数据量超过单机的承载能力以后很难处理,底层存储远远没有块存储与对象存储方式效率高,且图查询及图分析效率低下,系统较差的容灾性和实时性,面临数亿节点量级有着难以动态扩容,节点关联查询效率低下等问题。

发明内容

[0004] 本发明的目的在于克服现有技术的不足,提供基于Ceph的十亿级节点规模知识图谱的对象存储管理方法及系统,能处理十亿节点规模的知识图谱数据,支持大规模图数据存储和持弹性和线性扩展,可用性和容错率高,具备OLTP与CRUD特性,同时还支持OLAP数据分析和外部索引。

[0005] 本发明的目的是通过以下技术方案来实现的:

[0006] 基于Ceph的十亿级节点规模知识图谱的对象存储管理方法,方法包括以下步骤:

[0007] S1:图存储架构的构建,获取与目标业务对应的多个实体的实体数据,并根据所述实体数据生成所述目标业务对应的知识图谱并进行存储,并以Ceph作为分布式资源存储器,采用Client/Server架构,以多个Monitor组成的小集群构建一个Ceph集群,同时单个Monitor小集群下采用多个OSD存储图数据;

[0008] S2,外部索引后台的构建,将知识图谱数据映射为固定的索引数据结构,利用Elasticsearch/Solr检索引擎作为外部的索引插件,实现非等值查询,同时结合高效的索引机制构建外部索引后台;

[0009] S3,集成分布式计算引擎架构的构建,利用Spark计算引擎框架搭建分布式计算引擎,同时利用GraphX库将图关系转换为Spark算子,GraphX库将图数据以RDD分布式地存储

在Ceph集群的节点上,使用顶点RDD和边RDD分别对应存储顶点集合和边集合;

[0010] S4,图存储架构管理,在图存储架构、外部索引后台和集成分布式计算引擎构建的基础上,提供三层扩线查询、数据写入、数据读取、集群扩容、元数据备份、元数据快照、联机事物分析和联机分析处理操作来实现管理知识图谱的图数据。

[0011] 具体的,所述步骤S2中高效的索引机制包括图形索引和顶点中心索引,图形索引为整个知识图谱的全局索引结构;顶点中心索引是为每个顶点建立的本地索引结构。

[0012] 具体的,所述步骤S3中还包括分区操作,具体包括以下子步骤:

[0013] S101,顶点RDD通过按顶点的ID进行哈希分区,将顶点数据以多分区形式分布在集群上;

[0014] S102,边RDD按指定的分区策略进行分区,将边数据以多分区形式分布在集群上;

[0015] S103,将记录RDD分区内顶点与所有边RDD分区关系的路由表存储在顶点RDD的分区中,在边RDD需要顶点数据时,顶点RDD根据路由表把顶点数据发送至边RDD分区。

[0016] 具体的,所述步骤S4中的数据写入步骤包括以下子步骤:

[0017] S201,客户端连接上Monitor,获取集群Map信息,请求对应的主OSD数据节点;

[0018] S202,主OSD数据节点同时写入另外两个副本节点数据,等待主节点以及另外两个副本节点完成数据写入状态,主节点及副本节点写入状态都成功后,返回完成信号给Client,数据写入完成。

[0019] 具体的,所述步骤S4中的集群扩容步骤包括以下子步骤:

[0020] S301,Client连接Monitor获取集群Map信息,新主节点OSD1向Monitor上传请求,让OSD2节点接替OSD1节点成为临时主节点;

[0021] S302,临时主节点OSD2将数据全量同步给新主节点OSD1,ClientIO读写直接连接临时主节点OSD2进行数据读写;

[0022] S303,临时主节点OSD2收到读写IO,同时写入另外两个副本节点中的数据,临时主节点OSD2以及另外两副本节点中的三份数据都写入成功后,返回信号给Client,ClientIO读写完毕;

[0023] S304,若节点OSD1数据同步完毕,临时主节点OSD2向Monitor上传请求,交出主节点角色,OSD1节点重新成为主节点,OSD2节点变成副本节点;

[0024] S305,同时在图数据层面,实现节点扩容后,按照图数据切割方式将图数据切割,分别存储到多台机器上。

[0025] 具体的,所述图数据切割方式包括按点切割和按边切割两种切割方式;按点切割方式以图的顶点进行数据切割,切割线通过图的顶点,每一条边只保存一次,并且每一条边只出现在一台机器上,邻居顶点多的顶点会被分发到多台不同的机器上存储;按边切割方式以图的边进行数据切割,切割线只穿过连接顶点的边,每一个顶点只保存一次,切断的边分发到多台不同到的机器上存储。

[0026] 具体的,所述步骤S4中的元数据快照步骤包括:根据元数据信息有效的恢复到以前的数据状态,也可恢复程序到系统运行历史状态;保存特定时间点的系统数据,生成系统相应时间点报告;导出快照数据进行离线工作。

[0027] 具体的,所述步骤S4中的三层扩线查询步骤包括以下子步骤:

[0028] S401,将用户给定顶点集合Vset设置为第一层扩线查询的基础数据,设置第一层

的查询过滤条件为顶点Label/顶点属性的过滤条件ConditionA,进行第一层的顶点扩线查询;

[0029] S402,将满足第一层查询过滤条件边的顶点集合作为第二层扩线查询的基础数据,同时设置第二层的查询过滤条件为边Label/边属性的过滤条件ConditionB,并进行第二层的边扩线查询;

[0030] S403,将满足第二层查询过滤条件的边集合作为第三层扩线查询的基础数据,设置属性查询条件,进行第三层的属性扩线查询,输出经过三层扩线查询的查询结果。

[0031] 基于Ceph的十亿级节点规模知识图谱的对象存储管理系统,系统包括图数据存储模块、分布式计算模块、索引模块和元数据管理模块。其中,图数据存储模块用于分布式存储大规模知识图谱的对象数据,提供对象存储、块设备存储和文件系统服务;

[0032] 分布式计算模块用于利用SparkRDD内存计算,将大任务分解成多个子任务,分别部署到不同的机器上执行,完成后汇总,以提供高效的大规模数据处理能力,来支撑OLAP需求,供用户基于知识图谱进行数据分析;

[0033] 索引模块用于将知识数据映射为固定的索引数据结构,为用户提供图形索引、顶点中心索引和外部索引功能;

[0034] 元数据管理模块用于元数据的备份、元数据快照、程序恢复、生成时间点报告和系统离线工作。

[0035] 本发明的有益效果:本方案新增大数据分布式架构,引用分布式资源管理器,具有可扩展,高可用等主要性能特征,主要体现在分布式集群,外部索引,数据可靠性,分布式资源管理器方面。同时获得了能存储和表达海量知识,支持数十亿节点数据量,有着可靠、易用、高效的特性。

附图说明

[0036] 图1是本发明的方法流程图。

[0037] 图2是本发明的整体分布架构图。

[0038] 图3是本发明的分布式资源管理架构图。

[0039] 图4是本发明的集成式分布式计算引擎架构图。

[0040] 图5是本发明的外部索引插件架构图。

[0041] 图6是本发明的数据写入流程图。

[0042] 图7是本发明的集群扩容流程图。

[0043] 图8是本发明的系统功能模块图。

具体实施方式

[0044] 为了对本发明的技术特征、目的和效果有更加清楚的理解,现对照附图说明本发明的具体实施方式。

[0045] 本实施例中,如图1-2所示,基于Ceph的十亿级节点规模知识图谱的对象存储管理方法,方法包括以下步骤:

[0046] 步骤1:图存储架构的构建,获取与目标业务对应的多个实体的实体数据,并根据所述实体数据生成所述目标业务对应的知识图谱并进行存储。如图3所示,对于整体分布式

架构,并以Ceph作为分布式资源存储器,采用Client/Server架构,以多个Monitor组成的小集群构建一个Ceph集群,同时单个Monitor小集群下采用多个OSD存储图数据。

[0047] 步骤2,首先将知识图谱数据映射为固定的索引数据结构,为了具备能处理十亿级节点知识数据能力,如图5所示,新增了外部索引后台机制,利用Elasticsearch/Solr检索引擎作为外部的索引插件,实现在进行非等值查询时也能利用到索引,同时结合高效的索引机制构建外部索引后台。外部索引后台与索引引擎通过API方式交换数据。

[0048] 步骤3,集成分布式计算引擎架构的构建,利用Spark计算引擎框架搭建分布式计算引擎,同时利用GraphX库将图关系转换为Spark算子,GraphX库将图数据以RDD分布式地存储在Ceph集群的节点上,使用顶点RDD和边RDD分别对应存储顶点集合和边集合。

[0049] 步骤4,图存储架构管理,在图存储架构、外部索引后台和集成分布式计算引擎构建的基础上,提供三层扩线查询、数据写入、数据读取、集群扩容、元数据备份、元数据快照、联机事物分析和联机分析处理操作来实现管理知识图谱的图数据。

[0050] 本实施例中,高效的索引机制包括图形索引和顶点中心索引,图形索引为整个知识图谱的全局索引结构,通过对实体或者边的属性进行索引来获得更好的选择性,从而加快图遍历的速度,通过一个或者一组属性组成的固定属性组合进行等值检索。顶点中心索引是为每个顶点建立的本地索引结构,但当在大型的图中,每个顶点存在数千条或者更多的边,对这些顶点遍历会存在对应边的过滤,遍历效率较低,所以顶点中心索引只支持最左匹配。

[0051] 其中,对于基于索引的三层扩线查询,首先将用户给定顶点集合Vset设置为第一层扩线查询的基础数据,设置第一层的查询过滤条件为顶点Label/顶点属性的过滤条件ConditionA,进行第一层的顶点扩线查询。然后将满足第一层查询过滤条件边的顶点集合作为第二层扩线查询的基础数据,同时设置第二层的查询过滤条件为边Label/边属性的过滤条件ConditionB,并进行第二层的边扩线查询。最后一次扩线查询只查询除了满足ConditionB的边,但与这些边相关的顶点仅有顶点ID的信息,尚不包含任何属性信息,更不确定是否满足ConditionA,因此,需要再做一次属性查询。先将满足第二层查询过滤条件的边集合作为第三层扩线查询的基础数据,设置属性查询条件,进行第三层的属性扩线查询,输出经过三层扩线查询的查询结果,通过如上实施高效的索引得以发挥。

[0052] 本实施例中,如图4所示,为了支持OLA需求P,还扩展了一套高性能计算框架API,支持Spark,利用GraphX库将图关系转换为Spark算子,GraphX将图数据以RDD分布式地存储在集群的节点上,使用顶点RDD(VertexRDD)、边RDD(EdgeRDD)存储顶点集合和边集合。顶点RDD通过按顶点的ID进行哈希分区,将顶点数据以多分区形式分布在集群上。边RDD按指定的分区策略(PartitionStrategy)进行分区,将边数据以多分区形式分布在集群上。此外,顶点RDD中还拥有顶点到边RDD分区的路由信息—路由表。路由表存在顶点RDD的分区中,它记录分区内顶点跟所有边RDD分区的关系。在边RDD需要顶点数据时,顶点RDD会根据路由表把顶点数据发送至边RDD分区。至此,将图数据存储为Spark的RDD。

[0053] 在Spark底层,算子执行启动SparkContext,SparkContext向资源管理器注册并申请运行Executor资源,资源管理器分配Executor资源并启动StandaloneExecutorBackend(任务调度),Executor运行情况将随着心跳发送到资源管理器上,SparkContext构建成DAG图,将DAG图分解成Stage,并把Taskset发送给TaskScheduler。Executor向SparkContext申

请Task,TaskScheduler将Task发放给Executor运行同时SparkContext将应用程序代码发放给Executor,Task在Executor上运行,运行完毕释放所有资源。从而达到高效的mapEdges,mapVertices,aggregateMessages等操作,快速响应数据分析需求。

[0054] 本实施例中,如图6所示,对于数据写入,客户端(Client)连接上Monitor,获取集群Map信息,请求对应的主OSD数据节点,主OSD数据节点同时写入另外两个副本节点数据,等待主节点以及另外两个副本节点写完数据状态,主节点及副本节点写入状态都成功后,返回给Client,数据写入完成。数据读取方式与数据写入同理。

[0055] 本实例中,对于集群扩容,Client连接Monitor获取集群Map信息。同时新主节点OSD1由于没有PG(PlacementGroups)数据,会主动上报Monitor,告知让OSD2节点临时接替为主节点,临时主节点OSD2会把数据全量同步给新主节点OSD1,ClientIO读写直接连接临时主节点OSD2进行读写,OSD2节点收到读写IO,同时写入另外两副本节点,等待OSD2节点以及另外两副本节点写入成功,OSD2节点的三份数据都写入成功后,返回信号给Client。此时ClientIO读写完毕,如果OSD1节点数据同步完毕,临时主节点OSD2向Monitor上传请求,临时主节点OSD2会交出主角色,OSD1成为主节点,OSD2变成副本节点。同时在图数据层面,实现扩容后,将图切割,即需要将数据切分存储到多台机器上,第一类切割为按点切割,切割线通过图的顶点(Vertex),而不是边(Edge)。每一条边只保存一次,并且每一条边只出现在一台机器上,邻居多的顶点会被分发到不同的机器上;第二类按边切割,切割线只穿过连接顶点的边(Edge),每一个顶点只保存一次,切断的边会被保存到多台机器上,至此集群扩容完成。

[0056] 本实施例中,还提供了基于Ceph的十亿级节点规模知识图谱的对象存储管理系统,系统包括图数据存储模块、分布式计算模块、索引模块和元数据管理模块。

[0057] 其中,图数据存储模块用于分布式存储大规模知识图谱的对象数据,提供对象存储、块设备存储和文件系统服务。

[0058] 分布式计算模块用于利用SparkRDD内存计算,将大任务分解成多个子任务,分别部署到不同的机器上执行,完成后汇总,以提供高效的大规模数据处理能力,来支撑OLAP需求,供用户基于知识图谱进行数据分析。

[0059] 索引模块用于将知识数据映射为固定的索引数据结构,为用户提供图形索引、顶点中心索引和外部索引功能。

[0060] 元数据管理模块用于元数据的备份、元数据快照、程序恢复、生成时间点报告和系统离线工作。

[0061] 本实施例中,综上整个具体实施方案,可应用于反欺诈检测场景案列。即将用户信息、设备信息及社交关系构建了一个异构网络,并将该异构网络图应用在用户关联分析及反欺诈检测场景。导入数据后,节点数达到11亿的数量级,关系数据达到500亿左右的数量级,构成包含11类节点与13类边的复杂异构网络。通过特定规则筛选可疑用户,查看与可疑用户有特定关联的用户;查看与可疑用户有特定关联的所有用户组成子网的网络特征及用户特征;分析特定用户可以通过什么样的关联关系关联在一起;最多可分析6层关联关系的数据等完成一系列数据分析任务,在拥有11亿量级节点数的本专利的图谱中,相比现有图谱存储系统,本方案的图遍历和查询响应时间快4倍至100倍。本技术方案与现有图谱存储解决方案对比情况如下:

[0062] 表1数据加载

[0063]	本技术方案	NE04J-OFFLINE	NE04J-CYPHER
	45375秒	24小时内未完成	24小时内未完成

[0064] 表2数据存储大小

[0065]	本技术方案	NE04J-OFFLINE	NE04J-CYPHER
	609375MB	275950MB	1276175MB

[0066] 表3查询性能

[0067]	本技术方案	NE04J-OFFLINE	NE04J-CYPHER
	7.5毫秒	55.0毫秒	34.1毫秒

[0068] 以上显示和描述了本发明的基本原理和主要特征和本发明的优点。本行业的技术人员应该了解,本发明不受上述实施例的限制,上述实施例和说明书中描述的只是说明本发明的原理,在不脱离本发明精神和范围的前提下,本发明还会有各种变化和改进,这些变化和改进都落入要求保护的本发明范围内。本发明要求保护的范围由所附的权利要求书及其等效物界定。

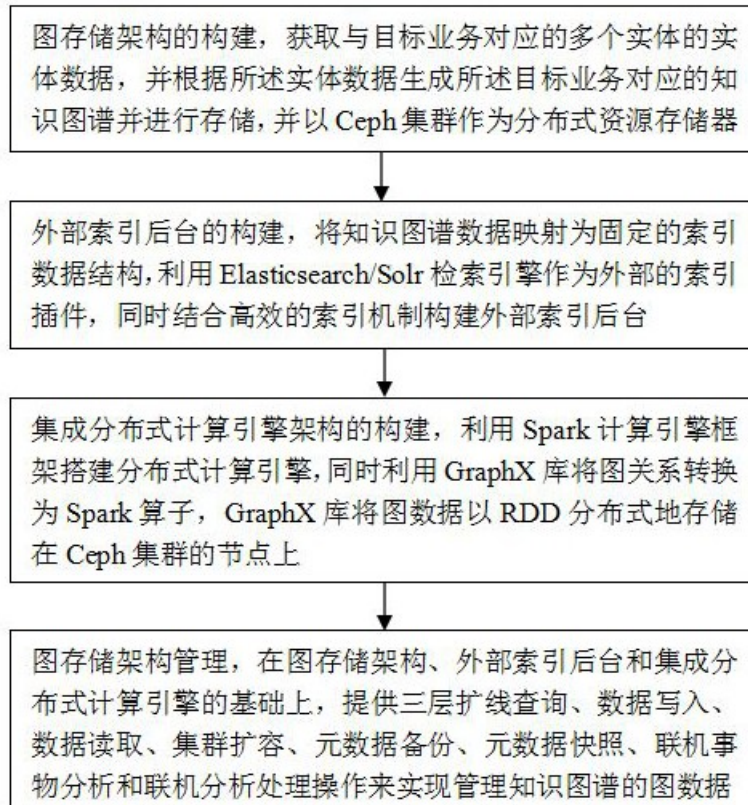


图1

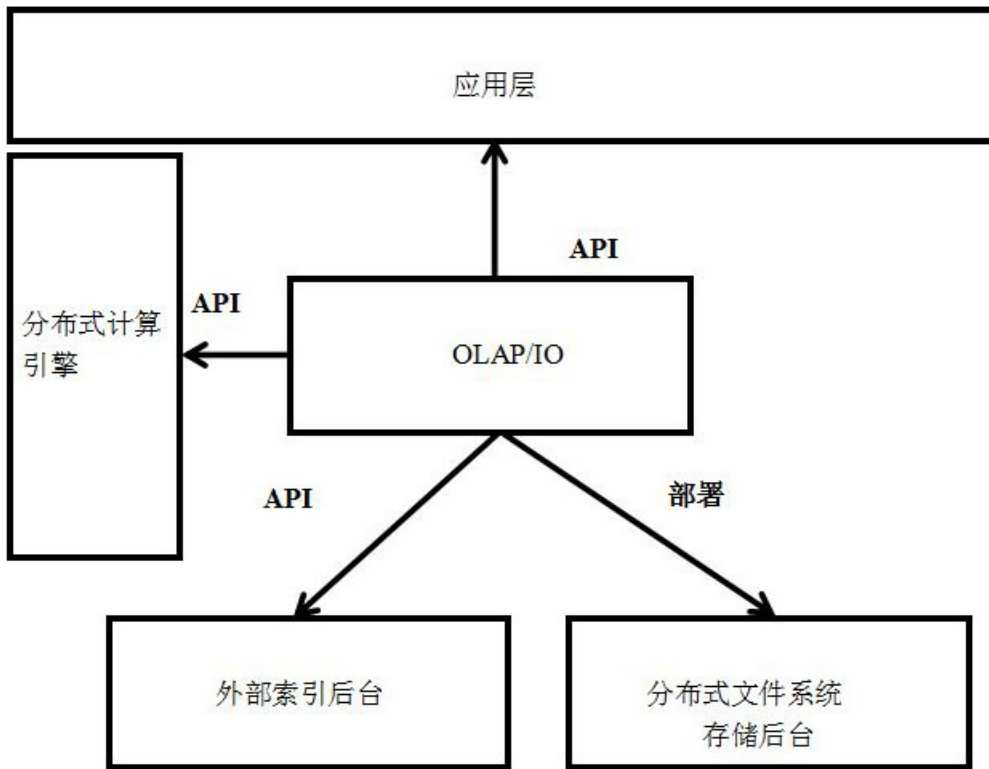


图2

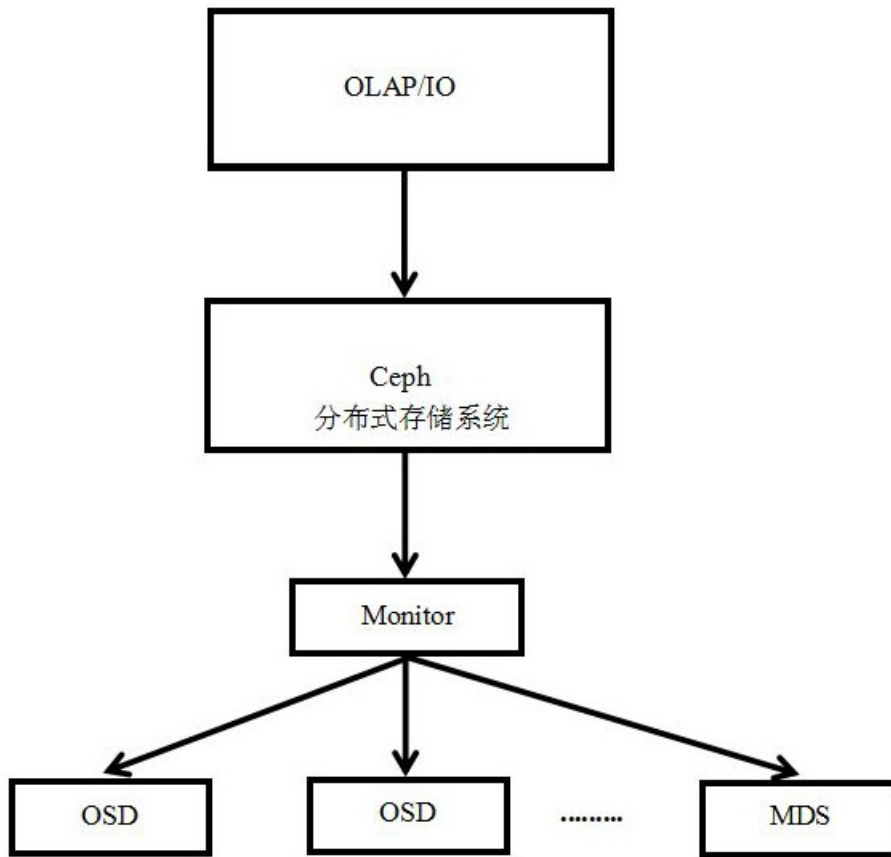


图3

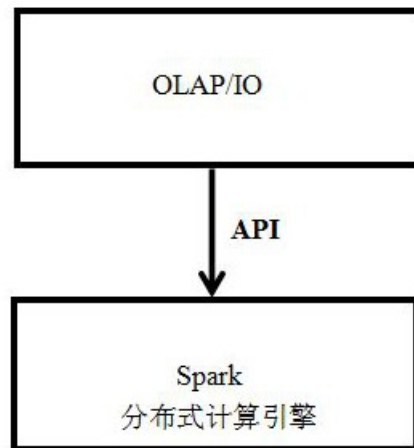


图4

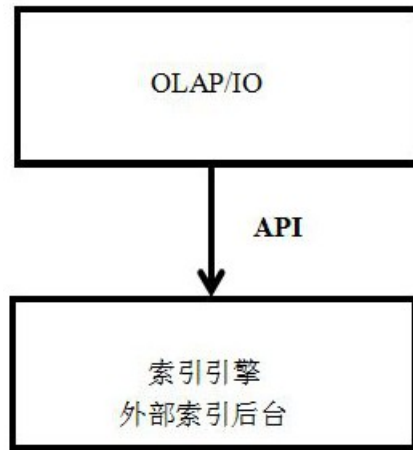


图5

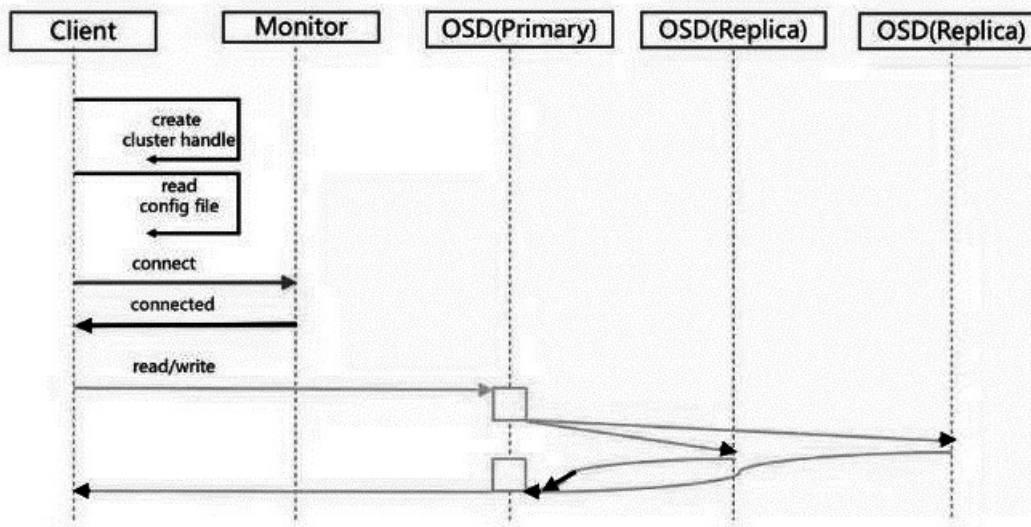


图6

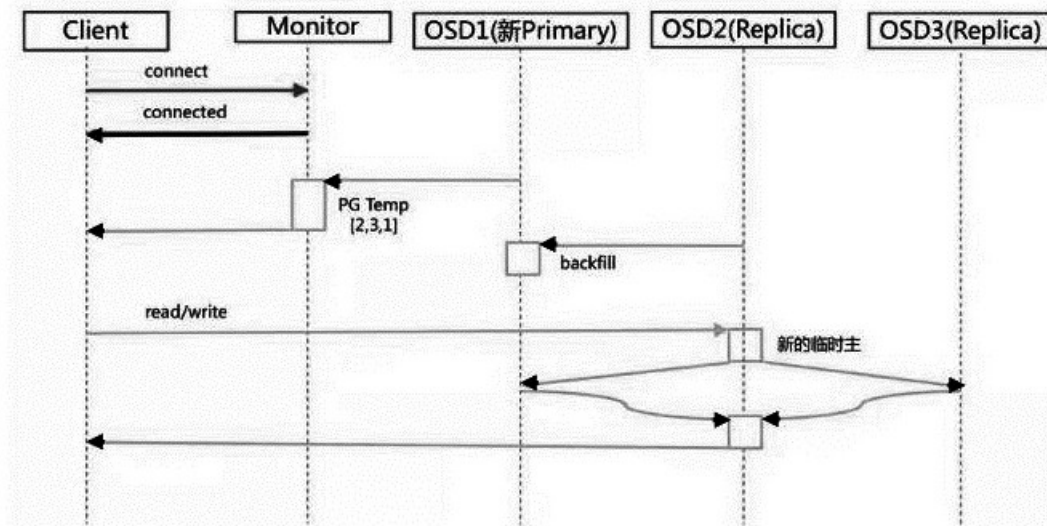


图7

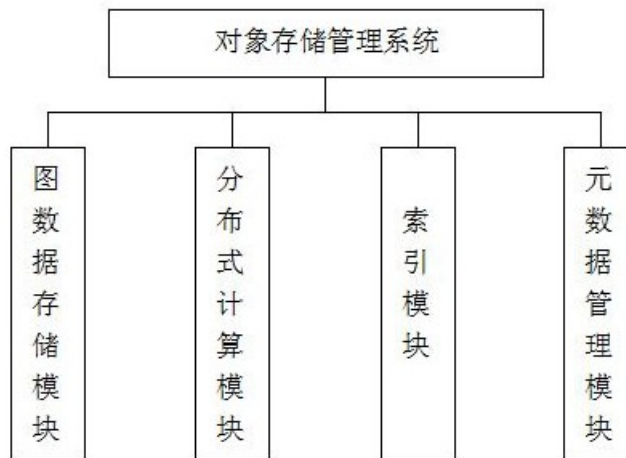


图8