



(12) 发明专利申请

(10) 申请公布号 CN 113422842 A

(43) 申请公布日 2021.09.21

(21) 申请号 202110960124.4

(22) 申请日 2021.08.20

(71) 申请人 国网江西省电力有限公司供电服务
管理中心

地址 330100 江西省南昌市南昌经济技术
开发区芙蓉路999号

(72) 发明人 卢婕 刘向向 严勤 范志夫
胡志强 俞林刚 冯颖 田静
刘波 曾子厦 伍栋文 晏依

(74) 专利代理机构 北京众合诚成知识产权代理
有限公司 11246

代理人 王焕巧

(51) Int. Cl.

H04L 29/08 (2006.01)

G16Y 10/35 (2020.01)

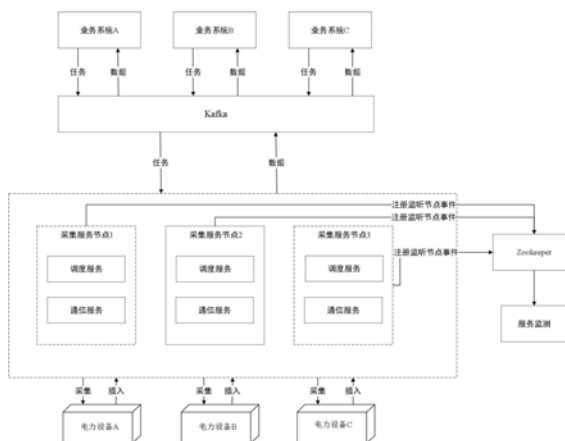
权利要求书3页 说明书9页 附图3页

(54) 发明名称

一种考虑网络负载的分布式电力用电信息
数据采集系统

(57) 摘要

本发明提供了一种考虑网络负载的分布式
电力用电信息数据采集系统,包括:S1:基于异步
事件驱动网络模型构建通信服务;S2:基于分布
式协调组件zookeeper构建分布式采集集群;S3:
设计考虑供需平衡的分布式负载管理模型;S4:
引入分布式消息中间件kafka处理剧增的业务数
据。本发明利用EPOLL设计基于通信频次的读写
缓冲区的异步数据处理策略,可以更加有效的利
用服务器资源;弹性扩招的采集服务集群避免了
采集服务的单点故障,可以随着设备的接入量在
不停机的情况下弹性伸缩,极大的提高了采集服
务的可靠性;基于图论构建供需平衡的分布式负
载模型,考虑各个节点的资源限制,极大的提高
了数据在分布式节点中的传输效率。



1. 一种考虑网络负载的分布式电力用电信息数据采集系统,其特征在于:包括以下步骤:

- S1:基于异步事件驱动网络模型构建通信服务;
- S2:基于分布式协调组件zookeeper构建分布式采集集群;
- S3:设计考虑供需平衡的分布式负载管理模型;
- S4:引入分布式消息中间件kafka处理剧增的业务数据。

2. 根据权利要求1所述的一种考虑网络负载的分布式电力用电信息数据采集系统,其特征在于:S1具体包括:

S1.1:借助EPOLL的I/O多路复用模型基于通讯频次构建高并发的网络通讯模型;

借助于EPOLL的事件通知机制,实时检测到网络通道上的读写就绪事件,根据相应的读写就绪事件,基于每个设备与通讯服务之间建立的网络连接上的通讯频次,对每一个设备维护一个数据写缓冲区和数据读缓冲区,其中读写缓冲区均具有一个回调函数,回调函数用来真正处理缓冲区中的数据,回调函数的应用,使主线程和上下行数据的处理过程剥离,使主线程专注于对读写缓冲的轮询判断,每个设备的数据由异步的回调函数处理,将需要下发的数据打包发送到网络通道或者将上行的数据发送到数据解析模块处理,业务处理线程按照通讯频次从大到小依次轮询处理每个设备对应缓冲区的数据;某个设备有数据需要下发时,将设备下行数据消息和当前时间写入设备对应的数据写缓冲区,当收到设备上行的数据后,收到EPOLL通知的读就绪事件后,从网络中读取数据,将设备上行数据消息和当前时间写入设备对应的数据读缓冲区;

业务处理线程以轮询的方式处理设备数据读写缓冲区,每次处理数据前对读写缓冲区按照通讯频次进行排序,每次轮询处理缓冲区中前1万个设备的数据,触发数据读写缓冲区对应的回调函数,对上行或下行数据进行真正的处理,并对读写缓冲区中通讯频次较低的设备的未处理时长是否达到一定的阈值,当达到一定的阈值时,触发数据读写缓冲区对应的回调函数,对上行或下行数据进行真正的处理;在处理数据读写缓冲区时,当某个设备有多条消息需要处理时,将多条消息打包成一个请求;

S1.2:通信服务使用主从Reactor多线程模型,一组线程池接收请求,一组线程池处理I/O;

通信服务端用于接收设备连接的不再是一个单独的I/O线程,而是一个独立的I/O线程池;监听线程组接收到设备网络连接的请求并处理完成后,将新创建的网络连接并注册到I/O线程池的某个I/O线程上,由它负责跟设备的网络连接的读写和数据的处理工作;监听线程组仅仅用于设备的登录、握手和安全认证,一旦链路建立成功,就将链路注册到I/O线程池的I/O线程上,由I/O线程负责后续的I/O操作;如果有数据需要下发到设备,由业务处理线程处理将需要下发的数据发送到I/O线程池对应的I/O通道上进行下发,如果设备有上行数据需要处理,则由I/O通道将数据发送给业务处理线程进行处理;

S1.3:对网络I/O操作进行异步处理;

通讯服务使用Future模式达到异步调用,在调用者提交任务后被调用者直接返回一个Future凭据,调用者在未来某个时间点根据Future凭据检查对应的调用是否返回结果,让原来需要同步等待的这段时间用来做其他的事情。

3. 根据权利要求1所述的一种考虑网络负载的分布式电力用电信息数据采集系统,其

特征在于:S2具体包括:基于ZooKeeper中事件监听器、异步通知和文件目录结构,构建弹性扩展的分布式采集集群;其中步骤如下:

S2.1:在Zookeeper里面创建名为cluster的永久节点,表示是整个采集服务集群的根节点;

S2.2:每一个采集服务节点启动时,在cluster的节点下创建自己的临时节点,并将自己的节点编号写入节点消息体中,表示此服务节点处于活跃状态;临时节点有个重要的特性,当创建此类节点的客户端与Zookeeper服务器的连接关闭时,此节点自动删除,利用此特性来感知采集服务节点的上下线;

S2.3:每个采集服务向ZooKeeper中的cluster节点注册监听器,监听cluster节点下节点的新增和删除事件;

S2.4:当采集服务监听到cluster节点下有新增或者删除事件时,获取新增或者删除的节点对应的编号,并更改集群变更标识;

S2.5:每个采集服务后一个后台守护线程每隔30s根据集群变更标识判断集群是否发生变更,若集群发生变更,等待60s后再次判断Zookeeper中的节点信息,是否和采集服务中保存的上一次节点信息一致,若一致,则不进行处理,认为是网络变更,向运维人员发送网络波动告警,若不一致,则触发设备档案重新分配的流程,每个采集服务重新负载到本服务节点的设备信息。

4.根据权利要求3所述的一种考虑网络负载的分布式电力用电信息数据采集系统,其特征在于:S3具体包括:在分布式系统中,各个服务节点之间都需要相互通信来传输一些数据,包括通知当前节点上设备的在线/离线总数、当前节点的硬件负载信息;为了快速地将数据传输到各个分布式服务节点,借助图考虑供需平衡的分布式负载管理和调度;其中,具体步骤如下:

S3.1:构建分布式网络;

边的初始传输容量定义,设一条边的容量为:

$$c_{ij} = \frac{k_i k_j}{\sum_{i,j} k_i k_j} b_{ij} N(N-1)$$

其中 k_i 为节点 i 的度,即连接节点 i 的边的数量, k_j 为节点 j 的度,即连接节点 j 的边的数量; b_{ij} 为边介数,其值等于通过这条边的最短路径数量除以网络中总的最短路径数量; i 和 j 为1至 N 的整数, N 为初始时网络中的节点数量;

S3.2:网络负载统计;

设每个节点以概率 $p \in [0, 1]$ 生成 $p(N-1)$ 个流量,分别随机选取分布式网络中的节点为目的地,同一个节点中的流量的目的地不能相同,即每个目的地只能对应一次流量生成;当网络中的所有节点都以概率 p 生成流量后,网络中的流量总需求为 $pN(N-1)$;设每条流量对于边容量的需求为1,且流量的传输采用最短路径,记一条流量最短传输路径为:

$f_{ij} = [e_{ix}, e_{xy}, \dots, e_{zj}]$,其中 e_{ix} 为从 i 到 x 路径的边, e_{xy} 为从 x 到 y 路径的边, \dots , e_{zj} 为从 z 到 j 路径的边;

S3.3:容量分配和边的负载计算;

边 e_{xy} 的容量为所有经过该边的流量提供传输支持,其所能给某条流量分配的传输容量为:

$$c_{xy} = \frac{I_{ij}}{\sum_{F_{xy}, i, j \in N} I_{ij}}$$

其中 c_{xy} 为从x到y的传输容量, F_{xy} 为边 e_{xy} 上所有流量的集合; I_{ij} 为这条流量的优先级,随机选择 $I_{ij} \in [1, 2, 3, 4, 5]$,5等级最高,以对应不同优先级的流量;

对于流量最短路径,边 e_{xy} 实际提供传输容量为:

$$b_{f_{ij}, e_{xy}} = \min \left\{ c_{xy} \frac{I_{ij}}{\sum_{F_{xy}, i, j \in N} I_{ij}}, 1 \right\}$$

即如果传输容量大于1则取1,否则采用通过计算得出的边 e_{xy} 所能提供的传输容量;考虑到路由上存在链路瓶颈效应,流量最短路径 f_{ij} 在网络中的传输效率为:

$b_{f_{ij}} = \min \{ b_{f_{ix}, e_{ix}}, b_{f_{xy}, e_{xy}}, \dots, b_{f_{ij}, e_{ij}} \}$,那么对于边 e_{xy} 来说,其总效率 l_{xy} 就是所有通过它的流量的传输效率总和,如下式所示:

$$l_{xy} = \sum_{F_{xy}} b_{f_{ij}}$$

S3.4:在分布式节点之间需要进行数据的传输时,数据发送的节点根据步骤S3.1-S3.3计算当前节点到目标节点所涉及所有边的传输效率,根据传输效率最高原则选择对应的边将数据通过分布式节点发送到目标节点。

5.根据权利要求1所述的一种考虑网络负载的分布式电力用电信息数据采集系统,其特征在于:S4具体包括:基于kafka构建高吞吐的消息处理机制,其中步骤如下:

S4.1:对于每个其他电力业务服务,定义一个用来发送消息和接收消息的主题,称topic;

S4.2:定义每个topic对应业务的消息格式;

S4.3:业务系统和采集服务分别订阅接收topic和发送topic;

S4.4:业务系统向消息发送topic发送消息后直接返回,处理自己的其他数据,不需要同步等待;

S4.5:采集服务采集到数据后向对应的接收消息的主题写入采集结果;

S4.6:业务系统收到采集服务返回的消息后进行对应的处理。

一种考虑网络负载的分布式电力用电信息数据采集系统

技术领域

[0001] 本发明涉及电力计量自动化和用电信息采集领域,具体为一种考虑网络负载的分布式电力用电信息数据采集系统。

背景技术

[0002] 电力信息采集系统的全面上线,电力数据的采集彻底告别了手工抄表的时代,电信息采集系统的远程自动抄表方式极大的提升了抄表效率,近年来,随着远程抄表方式的优化和能源互联网建设的持续推进,电力采集系统的新兴业务不断涌现,例如电力设备负荷识别、电力现货交易等,采集系统接入了更多类型和数量的设备,采集频度大幅提高,采集数据量越来越庞大,呈爆发式增长。现有的采集系统在面对海量设备的接入,海量数据的采集的需求时,存在着明显的性能瓶颈,只能增加服务器部署服务来满足海量设备的接入,不能充分运用现有服务器资源,严重增加了系统部署成本。

发明内容

[0003] 本发明的目的在于提供一种考虑网络负载的分布式电力用电信息数据采集系统,主要解决的是用电信息采集系统中,面对海量设备接入和高频采集需求,现有前置机存在技术瓶颈,无法满足新兴业务的开展的问题。

[0004] 为实现上述目的,本发明提供如下技术方案:一种考虑网络负载的分布式电力用电信息数据采集系统,包括以下步骤:

- S1:基于异步事件驱动网络模型构建通信服务;
- S2:基于分布式协调组件zookeeper构建分布式采集集群;
- S3:设计考虑供需平衡的分布式负载管理模型;
- S4:引入分布式消息中间件kafka处理剧增的业务数据。

[0005] 进一步的,S1具体包括:

- S1.1:借助EPOLL的I/O多路复用模型基于通讯频次构建高并发的网络通讯模型;

借助于EPOLL的事件通知机制,实时检测到网络通道上的读写就绪事件,根据相应的读写就绪事件,基于每个设备与通讯服务之间建立的网络连接上的通讯频次,对每一个设备维护一个数据写缓冲区和数据读缓冲区,其中读写缓冲区均具有一个回调函数,回调函数用来真正处理缓冲区中的数据,回调函数的应用,使主线程和上下行数据的处理过程剥离,使主线程专注于对读写缓冲的轮询判断,每个设备的数据由异步的回调函数处理,将需要下发的数据打包发送到网络通道或者将上行的数据发送到数据解析模块处理,业务处理线程按照通讯频次从大到小依次轮询处理每个设备对应缓冲区的数据;某个设备有数据需要下发时,将设备下行数据消息和当前时间写入设备对应的数据写缓冲区,当收到设备上行的数据后,收到EPOLL通知的读就绪事件后,从网络中读取数据,将设备上行数据消息和当前时间写入设备对应的数据读缓冲区;

业务处理线程以轮询的方式处理设备数据读写缓冲区,每次处理数据前对读写缓

缓冲区按照通讯频次进行排序,每次轮询处理缓冲区中前1万个设备的数据,触发数据读写缓冲区对应的回调函数,对上行或下行数据进行真正的处理,并对读写缓冲区中通讯频次较低的设备的未处理时长是否达到一定的阈值,当达到一定的阈值时,触发数据读写缓冲区对应的回调函数,对上行或下行数据进行真正的处理;在处理数据读写缓冲区时,当某个设备有多条消息需要处理时,将多条消息打包成一个请求;

S1.2:通信服务使用主从Reactor多线程模型,一组线程池接收请求,一组线程池处理 I/O;

通信服务端用于接收设备连接的不再是一个单独的I/O线程,而是一个独立的I/O线程池;监听线程组接收到设备网络连接的请求并处理完成后,将新创建的网络连接并注册到I/O线程池的某个I/O线程上,由它负责跟设备的网络连接的读写和数据的处理工作;监听线程组仅仅用于设备的登录、握手和安全认证,一旦链路建立成功,就将链路注册到I/O线程池的I/O线程上,由I/O线程负责后续的I/O操作;如果有数据需要下发到设备,由业务处理线程处理将需要下发的数据发送到I/O线程池对应的I/O通道上进行下发,如果设备有上行数据需要处理,则由I/O通道将数据发送给业务处理线程进行处理;

S1.3:对网络I/O操作进行异步处理;

通讯服务使用Future模式达到异步调用,在调用者提交任务后被调用者直接返回一个Future凭据,调用者在未来某个时间点根据Future凭据检查对应的调用是否返回结果,让原来需要同步等待的这段时间用来做其他的事情。

[0006] 进一步的,S2具体包括:基于ZooKeeper中事件监听器、异步通知和文件目录结构,构建弹性扩展的分布式采集集群;其中步骤如下:

S2.1:在ZooKeeper里面创建名为cluster的永久节点,表示是整个采集服务集群的根节点;

S2.2:每一个采集服务节点启动时,在cluster的节点下创建自己的临时节点,并将自己的节点编号写入节点消息体中,表示此服务节点处于活跃状态;临时节点有个重要的特性,当创建此类节点的客户端与ZooKeeper服务器的连接关闭时,此节点自动删除,利用此特性来感知采集服务节点的上下线;

S2.3:每个采集服务向ZooKeeper中的cluster节点注册监听器,监听cluster节点下节点的新增和删除事件;

S2.4:当采集服务监听到cluster节点下有新增或者删除事件时,获取新增或者删除的节点对应的编号,并更改集群变更标识;

S2.5:每个采集服务后一个后台守护线程每隔30s根据集群变更标识判断集群是否发生变更,若集群发生变更,等待60s后再次判断ZooKeeper中的节点信息,是否和采集服务中保存的上一次节点信息一致,若一致,则不进行处理,认为是网络变更,向运维人员发送网络波动告警,若不一致,则触发设备档案重新分配的流程,每个采集服务重新负载到本服务节点的设备信息。

[0007] 进一步的,S3具体包括:在分布式系统中,各个服务节点之间都需要相互通信来传输一些数据,包括通知当前节点上设备的在线/离线总数、当前节点的硬件负载信息;为了快速地将数据传输到各个分布式服务节点,借助图考虑供需平衡的分布式负载管理和调度;其中,具体步骤如下:

S3.1: 构建分布式网络;

边的初始传输容量定义, 设一条边的容量为:

$$c_{ij} = \frac{k_i k_j}{\sum_{i,j} k_i k_j} b_{ij} N(N-1)$$

其中 k_i 为节点 i 的度, 即连接节点 i 的边的数量, k_j 为节点 j 的度, 即连接节点 j 的边的数量; b_{ij} 为边介数, 其值等于通过这条边的最短路径数量除以网络中总的最短路径数量; i 和 j 为 1 至 N 的整数, N 为初始时网络中的节点数量;

S3.2: 网络负载统计;

设每个节点以概率 $p \in [0, 1]$ 生成 $p(N-1)$ 个流量, 分别随机选取分布式网络中的节点为目的地, 同一个节点中的流量的目的地不能相同, 即每个目的地只能对应一次流量生成; 当网络中的所有节点都以概率 p 生成流量后, 网络中的流量总需求为 $pN(N-1)$; 设每条流量对于边容量的需求为 1, 且流量的传输采用最短路径, 记一条流量最短传输路径为:

$f_{ij} = [e_{ix}, e_{xy}, \dots, e_{zj}]$, 其中 e_{ix} 为从 i 到 x 路径的边, e_{xy} 为从 x 到 y 路径的边, \dots , e_{zj} 为从 z 到 j 路径的边;

S3.3: 容量分配和边的负载计算;

边 e_{xy} 的容量为所有经过该边的流量提供传输支持, 其所能给某条流量分配的传输容量为:

$$c_{xy} \frac{I_{ij}}{\sum_{F_{xy}, i, j \in N} I_{ij}}$$

其中 c_{xy} 为从 x 到 y 的传输容量, F_{xy} 为边 e_{xy} 上所有流量的集合; I_{ij} 为这条流量的优先级, 随机选择 $I_{ij} \in [1, 2, 3, 4, 5]$, 5 等级最高, 以对应不同优先级的流量;

对于流量最短路径 f_{ij} , 边 e_{xy} 实际提供传输容量为:

$$b_{f_{ij}, e_{xy}} = \min \left\{ c_{xy} \frac{I_{ij}}{\sum_{F_{xy}, i, j \in N} I_{ij}}, 1 \right\}$$

即如果传输容量大于 1 则取 1, 否则采用通过计算得出的边 e_{xy} 所能提供的传输容量; 考虑到路由上存在链路瓶颈效应, 流量最短路径 f_{ij} 在网络中的传输效率为:

$b_{f_{ij}} = \min \{ b_{f_{ix}, e_{ix}}, b_{f_{xy}, e_{xy}}, \dots, b_{f_{zj}, e_{zj}} \}$, 那么对于边 e_{xy} 来说, 其总效率就是所有通过它的流量的传输效率总和, 如下式所示:

$$l_{xy} = \sum_{F_{xy}} b_{fij}$$

S3.4:在分布式节点之间需要进行数据的传输时,数据发送的节点根据步骤S3.1-S3.3计算当前节点到目标节点所涉及所有边的传输效率,根据传输效率最高原则选择对应的边将数据通过分布式节点发送到目标节点。

[0008] 进一步的,S4具体包括:基于kafka构建高吞吐的消息处理机制,其中步骤如下:

S4.1:对于每个其他电力业务服务,定义一个用来发送消息和接收消息的主题,称topic;

S4.2:定义每个topic对应业务的消息格式;

S4.3:业务系统和采集服务分别订阅接收topic和发送topic;

S4.4:业务系统向消息发送topic发送消息后直接返回,处理自己的其他数据,不需要同步等待;

S4.5:采集服务采集到数据后向对应的接收消息的主题写入采集结果;

S4.6:业务系统收到采集服务返回的消息后进行对应的处理。

[0009] 与现有技术相比,本发明的有益效果是:

采用本发明进行海量设备的数据采集,极大的增加了单台服务器设备的接入量,利用EPOLL设计基于通信频次的读写缓冲区的异步数据处理策略,打包多条数据处理,可以极大的提高网络吞吐量,可以更加有效的利用服务器资源,节省硬件成本;弹性扩招的采集服务集群避免了采集服务的单点故障,可以随着设备的接入量在不停机的情况下弹性伸缩,极大的提高了采集服务的可靠性;基于图论构建供需平衡的分布式负载模型,考虑各个节点的资源限制,极大的提高了数据在分布式节点中的传输效率;其他业务服务与采集服务的耦合度大幅降低,增加了采集服务的峰值处理能力和吞吐量;本发明极大的提高了采集服务的接入能力和并发能力,满足了高效高频数据采集的功能。

附图说明

[0010] 图1为本发明的采集服务流程图;

图2为基于EPOLL的高并发网络通讯模型图;

图3为通讯服务中的主从Reactor多线程模型图;

图4为Zookeeper集群管理框图;

图5为Kafka消息中间件处理业务消息框图。

具体实施方式

[0011] 为了使本发明的目的、技术方案及优点更加清楚明白,以下结合附图及实施例,对本发明进行进一步的详细说明。此处所描述的具体实施例仅用于解释本发明技术方案,并不限于本发明。

[0012] 本发明针对海量设备的接入,优化现有通信服务,构建高性能的通信服务,保证设备稳定接入,保证与设备通信的高并发和高可靠。海量设备的接入必然导致与其他业务系统的交互更加频繁,业务量的会急剧增加,引入分布式消息中间件和分布式协调组件构建

分布式弹性伸缩的采集架构,保证采集服务的高并发和高可用,分布式系统中,考虑到各个节点的网络I/O负载能力不同,各个节点的负载能力在每日不同的时间点也可能不同,可以通过相应的算法实时计算不同节点的网络承受能力,合理的分配数据在分布式节点中的传输路径。高性能的通信服务、弹性伸缩的采集架构,可极大提升系统的负载能力,满足采集业务中高频采集的需求。采集服务流程如图1所示,具体包括:

1、基于异步事件驱动网络模型构建的通信服务

海量设备的接入,需要高可靠、高并发、高吞吐的网络通信服务,保持与海量设备的连接,基于已有的网络通讯服务,使用以下技术对现有网络通讯服务进行改造,构建高性能、高并发、高吞吐的网络通讯服务。

[0013] 1) 借助EPOLL的I/O多路复用模型基于通讯频次构建高并发的网络通讯模型

目前市场上的I/O模型大多采用基于EPOLL的I/O多路复用模型,基于EPOLL的I/O多路复用模型是一种改进后的I/O模型,实现一个线程可以监视多个网络连接;就没有事件就绪时会阻塞应用程序,交出CPU;一旦某个网络连接上有事件就绪,反过来通知应用程序的业务处理线程调用事件对应的处理逻辑处理事件,解决了传统I/O模型轮询方式效率低下,CPU资源耗费大的问题。

[0014] 如图2所示,借助于EPOLL的事件通知机制,可实时检测到网络通道上的读写就绪事件,根据相应的读写就绪事件,基于每个设备与通讯服务之间建立的网络连接上的通讯频次,对每一个设备维护一个数据写缓冲区和数据读缓冲区,其中读写缓冲区均具有一个回调函数,回调函数中用来真正处理缓冲区中的数据,回调函数的应用,可以使主线程和上下行数据的处理过程剥离,使主线程专注于对读写缓冲的轮询判断,每个设备的数据由异步的回调函数处理,大大增加了系统的数据处理速度,将需要下发的数据打包发送到网络通道或者将上行的数据发送到数据解析模块处理,业务处理线程按照通讯频次从大到小依次轮询处理每个设备对应缓冲区的数据,提高高频通讯网络通道对应设备的响应速度。某个设备有数据需要下发时,将设备下行数据消息和当前时间写入设备对应的数据写缓冲区,当收到设备上行的数据后,收到EPOLL通知的读就绪事件后,从网络中读取数据,将设备上行数据消息和当前时间写入设备对应的数据读缓冲区。

[0015] 业务线程以轮询的方式处理设备数据读写缓冲区,每次处理数据前对读写缓冲区按照通讯频次进行排序,每次轮询处理缓冲区中前1万个设备的数据,触发数据读写缓冲区对应的回调函数,对上行或下行数据进行真正的处理,并对读写缓冲区中通讯频次较低的设备的未处理时长是否达到一定的阈值(ms级别),当达到一定的阈值时,也需要触发数据读写缓冲区对应的回调函数,对上行或下行数据进行真正的处理。在处理数据读写缓冲区时,当某个设备有多条消息需要处理时,将多条消息打包成一个请求,打包多条消息为一个请求的策略会极大地提高网络的吞吐量和数据处理效率。

[0016] 2) 主从Reactor多线程模型的应用

Reactor模型,是指通过一个或多个输入同时传递给服务处理器的服务请求的事件驱动处理模式。服务端程序处理传入多路请求,并将它们同步分派给请求对应的处理线程,Reactor模式也叫Dispatcher模式,即I/O多路复用统一监听事件,收到事件后分发(Dispatch给某进程),是编写高性能网络服务器的必备技术之一。

[0017] Reactor模型中有2个关键组成:

①Reactor在一个单独的线程中运行,负责监听和分发事件,分发给适当的处理程序来对I/O事件做出反应。它就像公司的电话接线员,它接听来自客户的电话并将线路转移到适当的联系人。

[0018] ②Handlers处理程序执行I/O事件要完成的实际事件,类似于客户想要与之交谈的公司中的实际官员。Reactor通过调度适当的处理程序来响应I/O事件,处理程序执行非阻塞操作。

[0019] 如图3所示,通信服务使用了Reactor线程模型中的主从 Reactor 多线程模型,这种线程模型适用于高并发场景,以极低的延迟处理网络请求,一组线程池接收请求,一组线程池处理 I/O。通信服务端用于接收设备连接的不再是一个单独的I/O线程,而是一个独立的I/O线程池。监听线程组接收到设备网络连接的请求并处理完成后,将新创建的网络连接并注册到I/O 线程池的某个I/O线程上,由它负责跟设备的网络连接的读写和数据的处理工作。监听线程组仅仅用于设备的登录、握手和安全认证,一旦链路建立成功,就将链路注册到I/O线程池的 I/O 线程上,由 I/O 线程负责后续的 I/O 操作。如果有数据需要下发到设备,由业务处理线程处理将需要下发的数据发送到I/O线程池对应的I/O通道上进行下发,如果设备有上行数据需要处理,则由I/O通道将数据发送给业务处理线程进行处理。

[0020] 3)对网络I/O操作进行异步处理

从操作系统层面来说,网络I/O往往是同步的,同步意味着应用程序发出一个网络I/O相关的调用,在没有得到结果之前,该调用就不返回。但是一旦调用返回,就得到返回值了。换句话说,就是由调用者主动等待这个调用的结果。异步的概念和同步相对。当一个异步过程调用发出后,调用者不能立刻得到结果。实际处理这个调用的部件在完成后,通过状态、通知和回调来通知调用者。通讯服务使用Future模式达到异步调用,Future模式解决的是任务需要执行比较长的时间,通常需要等待任务执行结束或者出错才能返回结果,在此期间调用者只能陷入阻塞,对此Future设计模式提供了一种凭据式的解决方案。具体到程序中,在调用者提交任务后被调用者直接返回一个Future凭据,调用者可以在未来某个时间点根据Future凭据检查对应的调用是否返回结果,Future模式的核心思想是能够让原来需要同步等待的这段时间用来做其他的事情。异步操作可以在稍后获得执行结果,不用一直同步等待去获得执行结果,在发送网络请求后,网络操作并不会阻塞,会简单的返回一个FutureTask,调用者并不能立刻获得结果,在稍后的一段时间里,用户可以方便的主动获取或者通过通知机制获得I/O操作结果;异步处理的好处是不会造成线程阻塞,线程在I/O操作期间可以执行别的程序,在高并发情形下会更稳定,具备更高的吞吐量。

[0021] 2、分布式调度

海量设备稳定接入采集系统后,对系统的考验便在剧增的电力业务的及时性和可靠性上,业务的顺利进行,取决于采集系统的稳定性,能够保证采集系统的负载情况较为稳定,不会在业务高峰期由于高查询率(QPS)导致服务器负载持续增加,影响电费结算、费控、电力现货交易等电力业务的进行。

[0022] 基于上述需求,基于分布式协调组件zookeeper的构建高可用的分布式采集集群,保证采集服务的高可用和弹性伸缩;设计考虑供需平衡的分布式负载管理模型;引入分布式消息中间件kafka,应对剧增的电力业务,提高系统的峰值处理能力,采集服务平稳运行。

[0023] 1)基于分布式协调组件zookeeper构建分布式采集集群

要保证采集服务的高频采集和稳定运行,采集服务需要保证高可用,避免单点故障,将采集设备平均负载于采集集群中的各个采集节点上,在接入设备量进一步提升时需要在不停机状态行动态扩展采集服务,借助于ZooKeeper构建高可用的弹性伸缩的分布式采集集群。

[0024] Zookeeper 是一个高可用的开源的分布式协调服务,广泛使用于各种分布式系统当中,是一个基于观察者模式设计的分布式服务管理框架,它负责存储和管理大家都关心的数据,然后接受客户端的注册,一旦这些数据的状态发生变化,Zookeeper 就将负责通知已经在 Zookeeper 上注册的那些客户端做出相应的反应,ZooKeeper 允许用户在指定节点上注册一些 Watcher(事件监听器),并且在一些特定事件触发的时,ZooKeeper服务端将事件通知到感兴趣的客户端上去,该机制是 ZooKeeper 实现分布式协调服务的重要特性。

[0025] 如图4所示,基于ZooKeeper中watcher(事件监听器)、异步通知和文件目录结构,构建弹性扩展的分布式采集集群。其中步骤如下:

①在Zookeeper里面创建名为cluster的永久节点,表示是整个采集服务集群的根节点。

[0026] ②每一个采集服务节点启动时,在cluster的节点下创建自己的临时节点,并将自己的节点编号写入节点消息体中,表示此服务节点处于活跃状态。临时节点有个重要的特性,当创建此类节点的客户端与Zookeeper服务器的连接关闭时,此节点自动删除,可利用此特性来感知采集服务节点的上下线。

[0027] ③每个采集服务向ZooKeeper中的cluster节点注册监听器,监听cluster节点下节点的新增和删除事件。

[0028] ④当采集服务监听到cluster节点下有新增或者删除事件时,获取新增或者删除的节点对应的编号,并更改集群变更标识。

[0029] ⑤每个采集服务后一个后台守护线程每隔30s根据集群变更标识判断集群是否发生变更,若集群发生变更,等待60s后再次判断Zookeeper中的节点信息,是否和采集服务中保存的上一次节点信息一致,此操作是为了防止短时间内的网络波动导致的采集服务短暂上下线造成集群节点频繁变更,若一致,则不进行处理,认为是网络变更,向运维人员发送网络波动告警,若不一致,则触发设备档案重新分配的流程,每个采集服务重新负载到本服务节点的设备信息。

[0030] 2)考虑供需平衡的分布式负载模型

在分布式系统中,各个服务节点之间都需要相互通信来传输一些数据,如通知当前节点上设备的在线/离线总数、当前节点的硬件负载等信息,分布式节点中每个节点的网络带宽,负载情况都有不同,因此,为了快速地将数据传输到各个分布式服务节点,借助图考虑供需平衡的分布式负载管理和调度。其中,具体步骤如下:

①构建分布式网络

边的初始传输容量定义,设一条边的容量为:

$$c_{ij} = \frac{k_i k_j}{\sum_{i,j} k_i k_j} b_{ij} N(N-1)$$

其中 k_i 为节点 i 的度,即连接节点 i 的边的数量, k_j 为节点 j 的度,即连接节点 j 的边的数量; b_{ij} 为边介数,其值等于通过这条边的最短路径数量除以网络中总的最短路径数量; i 和 j 为1至 N 的整数, N 为初始时网络中的节点数量。

[0031] ②网络负载统计

设每个节点以概率 $p \in [0, 1]$ 生成 $p(N-1)$ 个流量,分别随机选取分布式网络中的节点为目的地,同一个节点中的流量的目的地不能相同,即每个目的地只能对应一次流量生成;当网络中的所有节点都以概率 p 生成流量后,网络中的流量总需求为 $pN(N-1)$;设每条流量对于边容量的需求为1,且流量的传输采用最短路径,记一条流量最短传输路径为:

$f_{ij} = [e_{ix}, e_{xy}, \dots, e_{zj}]$,其中 e_{ix} 为从 i 到 x 路径的边, e_{xy} 为从 x 到 y 路径的边, \dots , e_{zj} 为从 z 到 j 路径的边。

[0032] ③容量分配和边的负载计算

边 e_{xy} 的容量为所有经过该边的流量提供传输支持,其所能给某条流量分配的传输容量为:

$$c_{xy} \frac{I_{ij}}{\sum_{F_{xy}, i, j \in N} I_{ij}}$$

其中 c_{xy} 为从 x 到 y 的传输容量, F_{xy} 为边 e_{xy} 上所有流量的集合; I_{ij} 为这条流量的优先级,可以随机选择 $I_{ij} \in [1, 2, 3, 4, 5]$,5等级最高,以对应不同优先级的流量;

对于流量 f_{ij} ,边 e_{xy} 实际提供传输容量为:

$$b_{f_{ij}, e_{xy}} = \min \left\{ c_{xy} \frac{I_{ij}}{\sum_{F_{xy}, i, j \in N} I_{ij}}, 1 \right\}$$

即如果传输容量大于1则取1,否则采用通过计算得出的边 e_{xy} 所能提供的传输容量;考虑到路由上存在链路瓶颈效应,流量 f_{ij} 在网络中的传输效率为:

$b_{f_{ij}} = \min \{b_{f_{ij}, e_{ix}}, b_{f_{ij}, e_{xy}}, \dots, b_{f_{ij}, e_{zj}}\}$,那么对于边 e_{xy} 来说,其总效率 l_{xy} 就是所有通过它的流量的传输效率总和,如下式所示:

$$l_{xy} = \sum_{F_{xy}} b_{f_{ij}}$$

④在分布式节点之间需要进行数据的传输时,数据发送的节点根据上述算法计算当前节点到目标节点所涉及所有边的传输效率,根据传输效率最高原则选择对应的边将数据通过分布式节点发送到目标节点。

[0033] 3) 引入分布式消息中间件kafka处理剧增的业务数据

采集服务首先要保证的是高效的与设备通讯,实时响应数据的召测,业务数据的剧增,意味着采集服务与其他服务是之间的交互也越来越频繁,采集服务面对着来自页面召测、后台定时任务、第三方接口等一系列服务的交互,之前往往采用REST或者直接建立SOCKET连接进行通讯,使用REST方式或者socket连接时采集服务与其他服务之间的耦合度较高,其他业务服务需要同步等待采集服务返回执行结果,在业务高峰期各服务的负载压力会大幅增加,性能受到严重影响,数据采集效率无法得到保障。故我们引入Apache Kafka消息中间件解决不同业务系统与采集服务之间的高耦合、业务高峰期吞吐量服务负载大、服务吞吐量不高的问题;Kafka是一个分布式消息系统,天然具有高可用的特性,在高可用的前提下,具有高吞吐量、低延迟、高并发的特性,每秒可处理十万级别的消息。利用kafka,可实现不同业务服务之间的解耦,其他业务服务将消息写入kafka后直接返回,不需要长时间阻塞等待;可实现高峰期削峰功能,不至于由于业务量太大导致服务性能受到影响,使业务平稳运行。

[0034] 如图5所示,基于kafka优秀的性能,构建高吞吐的消息处理机制,其中步骤如下:

①对于每个其他电力业务服务,例如费控、对时、数据召测等,定义一个用来发送消息和接收消息的主题,称topic。

[0035] ②定义每个topic对应业务的消息格式。

[0036] ③业务系统和采集服务分别订阅接收topic和发送topic。

[0037] ④业务系统向消息发送topic发送消息后直接返回,处理自己的其他数据,不需要同步等待。

[0038] ⑤采集服务采集到数据后向对应的接收消息的主题写入采集结果。

[0039] ⑥业务系统收到采集服务返回的消息后进行对应的处理。

[0040] 本发明提出了一种考虑网络负载的分布式电力用电信息数据采集系统,优化海量设备的网络接入通讯模型,采用分布式协调组件和高性能的消息中间件,综合分析采集系统中各个节点的网络负载情况,优化数据在分布式节点之间的传输算法,充分利用现有服务器资源,构建高性能的弹性伸缩的分布式采集服务集群,提高集群中每台服务器的资源利用率,支持在采集服务不停机的情况下根据设备的接入和集群的负载情况弹性伸缩,保证采集服务的高并发和高可用,保证数据的高频采集。

[0041] 以上所述仅表达了本发明的优选实施方式,其描述较为具体和详细,但并不能因此而理解为对本发明专利范围的限制。应当指出的是,对于本领域的普通技术人员来说,在不脱离本发明构思的前提下,还可以做出若干变形、改进及替代,这些都属于本发明的保护范围。因此,本发明专利的保护范围应以所附权利要求为准。

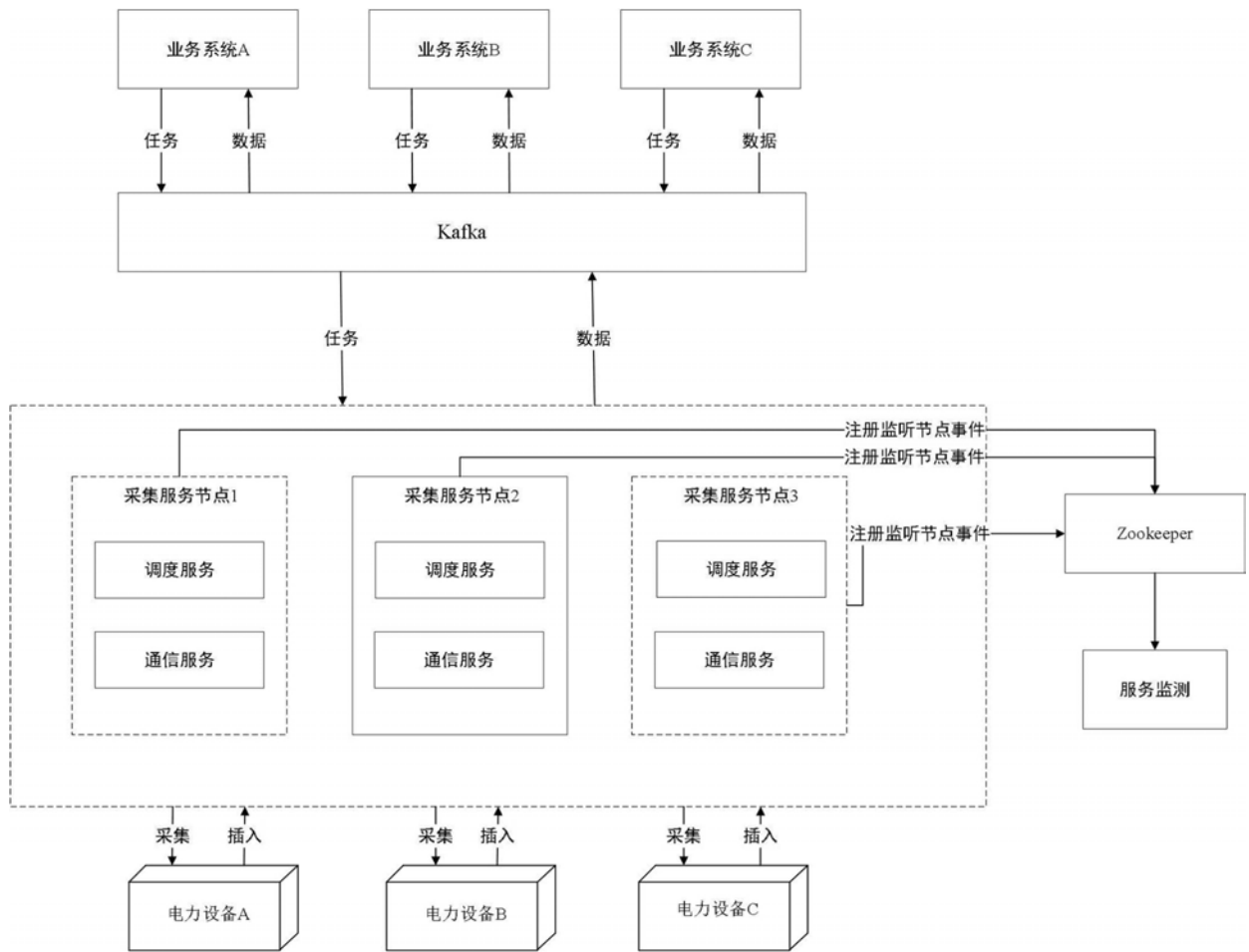


图1

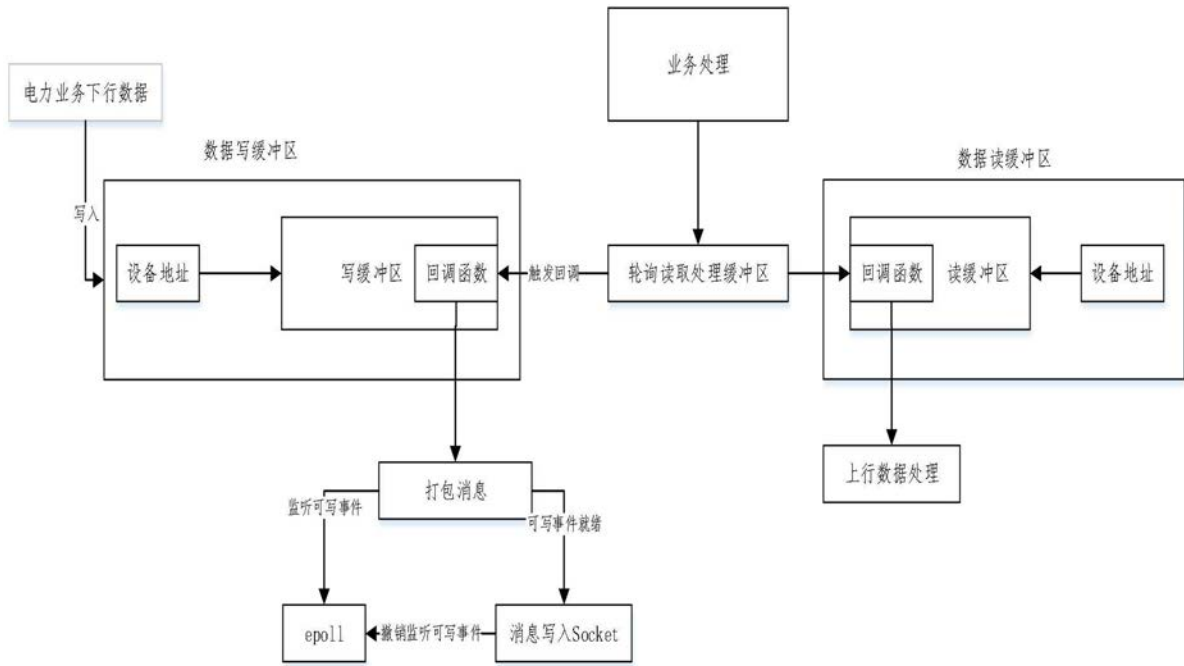


图2

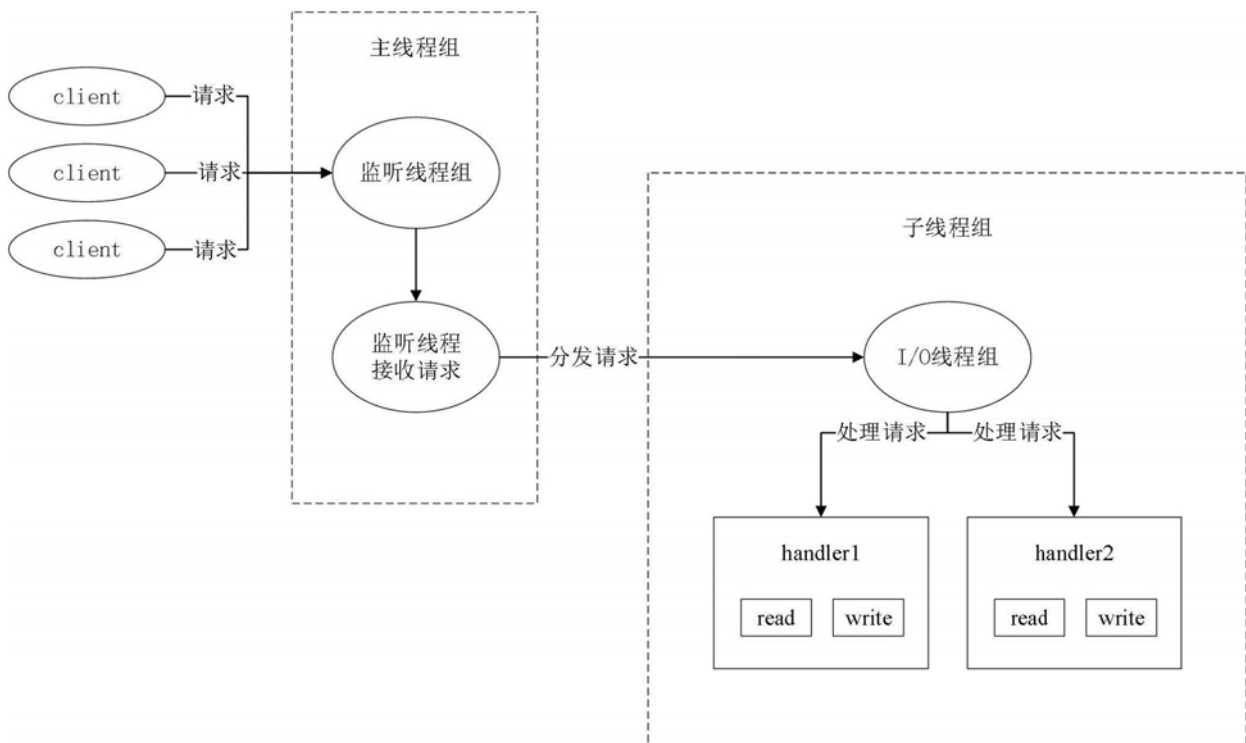


图3

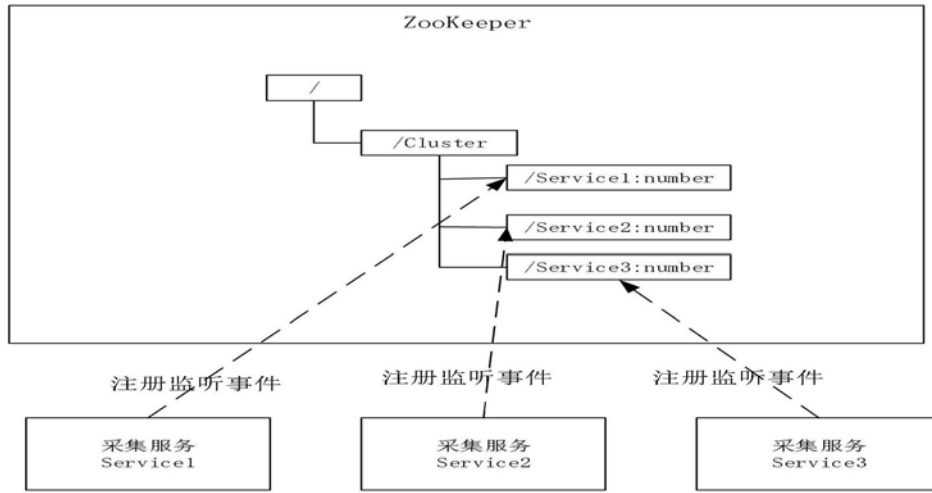


图4

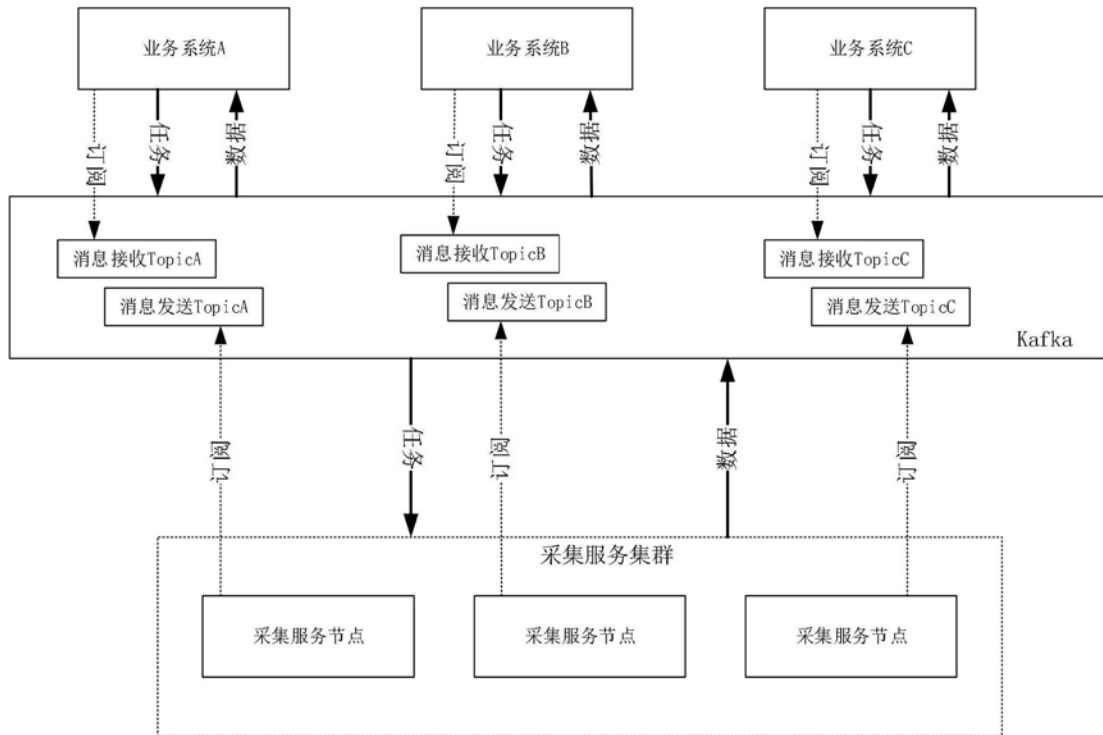


图5