(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 107295106 B
(45)授权公告日 2020.08.14

(21)申请号 201710637641.1

(22)申请日 2017.07.31

(65)同一申请的已公布的文献号
申请公布号 CN 107295106 A

(43)申请公布日 2017.10.24

(73)专利权人 杭州多麦电子商务股份有限公司
地址 310000 浙江省杭州市江干区九盛路9号A13幢215室

(72)发明人 胡悦 吴文龙

(74)专利代理机构 浙江千克知识产权代理有限公司 33246
代理人 裴金华

(51)Int.Cl.
H04L 29/08(2006.01)
H04L 12/58(2006.01)

(56)对比文件
CN 106953901 A,2017.07.14
CN 106484329 A,2017.03.08
CN 106293968 A,2017.01.04
CN 101707633 A,2010.05.12
CN 106953901 A,2017.07.14
CN 105068769 A,2015.11.18
CN 106598762 A,2017.04.26
CN 103209214 A,2013.07.17
CN 104424186 A,2015.03.18
WO 2004100009 A1,2004.11.18
US 2007073821 A1,2007.03.29
US 2004240462 A1,2004.12.02
Kai Sachs et al..Performance evaluation of message-oriented middleware using the SPECjms2007 benchmark.《Performance Evaluation》.2009,

审查员 鲁卉

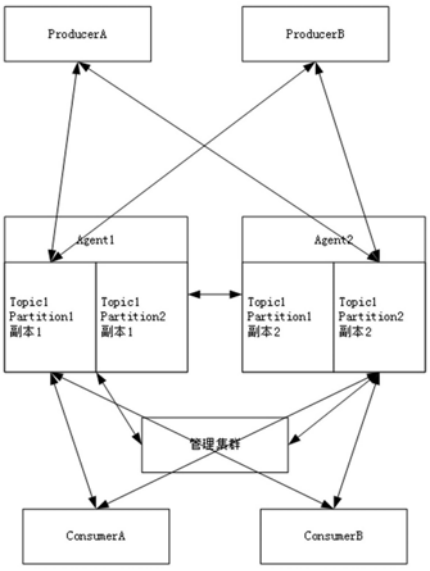
权利要求书2页 说明书7页 附图4页

(54)发明名称

消息数据服务集群

(57)摘要

本发明涉及消息处理领域,具体涉及一种消息数据服务集群,包括用于存储主题消息数据的主题分区,所述主题分区包括分布在所述服务集群的不同服务端;其特征在于:所述主题消息数据以内存模式或者文件模式存储在所述服务集群中;所述内存模式下,所述主题分区接收生产端产生的所述主题消息数据并映射到所述服务端的内存中;所述文件模式下,所述主题分区接收生产端产生的所述主题消息数据并存储到所属服务端的硬盘中。



CN 107295106 B

1.一种消息数据服务集群，包括用于存储主题消息数据的主题分区，所述主题分区包括分布在所述服务集群的不同服务端；其特征在于：所述主题消息数据以内存模式或者文件模式存储在所述服务集群中；

所述内存模式下，所述主题分区接收生产端产生的所述主题消息数据并映射到所述服务端的内存中；

所述文件模式下，所述主题分区接收生产端产生的所述主题消息数据并存储到所属服务端的硬盘中；

所述主题分区包括分布在所述服务集群的不同服务端中的主副本和从副本，所述从副本为所述主副本的冗余备份，所述主题分区通过所述主副本接收来自所述生产端的主题消息数据，所述主题分区通过所述主副本响应消费端的消费请求；

所述主题分区包括由所述从副本组成的保持同步副本集，所述保持同步副本集内的从副本相对于其他从副本用于更新的消息数据；当所述保持同步副本集中的所述从副本数量小于预设的最小同步副本数量时，所述主题分区不接收来自生产集群的主题消息数据；

在无保障等级下，所述主副本在接收到所述生产端发送的主题消息数据后，发送消息提交成功信息至所述生产端；

在快速等级下，所述主副本在接收到所述生产端发送的主题消息数据后同步给所述从副本，并且在收到第一个从副本的确定同步信息后，发送消息提交成功信息至所述生产端；

在安全等级下，所述主副本在接收到所述生产端发送的主题消息数据后同步给所述从副本，并且在收到大于所述保持同步副本集的从副本数量预设值个从副本的确定同步信息后，发送消息提交成功信息至所述生产端；

当所述主副本下线时，由保持同步副本集合中的各从副本将其最大逻辑版本号上报给管理集群，所述管理集群根据保持同步副本集合中所有所述从副本的最大逻辑版本号，选出最大逻辑版本号对应的从副本作为主副本。

2.根据权利要求1所述的一种消息数据服务集群，其特征在于：所述主题分区包括数据段落，所述数据段落包括所述主题消息数据和索引文件；所述索引文件记在所述主题消息数据的逻辑版本号与所述主题消息数据的物理偏移的一一映射关系。

3.根据权利要求2所述的一种消息数据服务集群，其特征在于：所述逻辑版本号反映所述主题消息数据到达所述服务集群的先后顺序。

4.根据权利要求3所述的一种消息数据服务集群，其特征在于：所述内存模式下，所述主题分区包括数据段落跳跃表，所述数据段落跳跃表包括多个数据层；下一数据层的节点数量大于上一数据层的节点数量，并且下一数据层包括上一数据层的所有节点，位于底层的数据层包括所述数据段落跳跃表的所有节点；所述节点包括段落数据，所述段落数据包括所述主题消息数据；所述节点包括其在下一数据层中的指针数据、所述节点所在数据层的下一节点的指针数据。

5.根据权利要求4所述的一种消息数据服务集群，其特征在于：所述文件模式下，所述主题分区记载所述数据段落与所述数据段落中的主题消息数据的物理偏移的范围。

6.根据权利要求5所述的一种消息数据服务集群，其特征在于：所述主副本记录其最新接收到的所述主题消息数据的有效逻辑版本号和缓存在内存中的所述主题消息数据的缓存逻辑版本号。

7.根据权利要求6所述的一种消息数据服务集群,其特征在于:生产端轮询各主题分区的主副本,以将其产生的主题消息数据分布式地存储在各主题分区中。

8.根据权利要求7所述的一种消息数据服务集群,其特征在于:消费端轮询各主题分区,以消费所述主题消息数据。

9.根据权利要求8所述的一种消息数据服务集群,其特征在于:所述主题分区在同一时刻最多允许同一消费集群中的一个消费端消费其主题消息数据。

消息数据服务集群

技术领域

[0001] 本发明涉及消息处理领域,具体涉及一种消息数据服务集群。

背景技术

[0002] 目前用于消息(如日志)处理的消息系统很多,比较流行的是分布式消息系统。

[0003] 分布式消息系统框架如图1所示。包括Producer(消息生产者,简称PD)、Agent(消息缓存者),Consumer(消息处理者,简称CS)以及第三方管理集群,每个角色都可以有多个。Producer发送消息到Agent,消息持久化存储在Agent上,Consumer再从Agent取得消息进行处理。第三方管理集群用来存储Producer,Consumer与Agent的一些状态信息。

[0004] 分布式消息系统基于消息的主题(Topic)进行消息管理。在消息存储设备中也是基于主题来进行存储的。并且是以文件模式持久化在服务端的硬盘中。这种类型的主题消息数据使用文件进行存储不太方便。首先,文件存储是针对生命周期相对较长的消息设计的,不能及时的淘汰过期数据;其次,对于文件存储而言,能够被消费的数据一定是已经落地(即持久化到硬盘的、具有有效逻辑版本号)的主题消息数据,而文件存储模式下主题消息数据被接收到落地之间有时间差,不能及时被消费。

发明内容

[0005] 本发明为了解决上述技术问题,提出了一种消息数据服务集群,包括用于存储主题消息数据的主题分区,所述主题分区包括分布在所述服务集群的不同服务端;其特征在于:所述主题消息数据以内存模式或者文件模式存储在所述服务集群中;所述内存模式下,所述主题分区接收生产端产生的所述主题消息数据并映射到所述服务端的内存中;所述文件模式下,所述主题分区接收生产端产生的所述主题消息数据并存储到所属服务端的硬盘中。

[0006] 作为优选,所述主题分区包括数据段落,所述数据段落包括所述主题消息数据和索引文件;所述索引文件记在所述主题消息数据的逻辑版本号与所述主体消息数据的物理偏移的一一映射关系。

[0007] 作为优选,所述逻辑版本号反映所述主题消息数据到达所述服务集群的先后顺序。

[0008] 作为优选,所述主题分区包括分布在所述服务集群的不同服务端中的主副本和从副本,所述从副本为所述主副本的冗余备份,所述主题分区通过所述主副本接收来自所述生产端的主题消息数据,所述主题分区通过所述主副本响应所述消费端的消费请求。

[0009] 作为优选,所述内存模式下,所述主题分区包括数据段落跳跃表,所述数据段落跳跃表包括多个数据层;下一数据层的节点数量大于上一数据层的节点数量,并且下一数据层包括上一数据层的所有节点,位于底层的数据层包括所述数据段落跳跃表的所有节点;所述节点包括段落数据,所述段落数据包括所述主题消息数据;所述节点包括其在下一数据层中的指针数据、所述节点所在数据层的下一节点的指针数据。

[0010] 作为优选，所述文件模式下，所述主体分区记载所述数据段落与所述数据段落中的主题消息数据的物理偏移的范围。

[0011] 作为优选，所述主副本记录其最新接收到的所述主题消息数据的有效逻辑版本号和缓存在内存中的所述主体消息数据的缓存逻辑版本号。

[0012] 作为优选，生产端轮询各主题分区的主副本，以将其产生的主题消息数据分布式地存储在各主题分区中。

[0013] 作为优选，消费端轮询各主题分区，以消费所述主题消息数据。

[0014] 作为优选，所述主题分区在同一时刻最多允许同一消费集群中的一个消费端消费其主题消息数据。

附图说明

[0015] 图1 现有技术的消息系统框架。

[0016] 图2现有技术的消息系统消费模式。

[0017] 图3本发明的分布式消息系统框架。

[0018] 图4本发明的分布式消息系统消费模式。

[0019] 图5本发明的数据段落跳跃表结构图。

具体实施方式

[0020] 以下具体实施例仅仅是对本发明的解释，其并不是对本发明的限制，本领域技术人员在阅读完本说明书后可以根据需要对本实施例做出没有创造性贡献的修改，但只要在本发明的权利要求范围内都受到专利法的保护。

[0021] 实施例一

[0022] 如图3所示为本发明分布式消息系统框架，包括用于产生主题消息数据的生产集群、存储主题消息数据的服务集群、消费主题消息数据的消费集群、以及用于管理消费集群和服务集群的管理集群。服务集群包括用于存储主题消息数据的主题分区，主题分区包括分布在服务集群的不同服务端中的主副本(Leader)和从副本(Follower)，从副本为主副本的冗余备份。服务集群由一个或多个Agent服务器(服务端)组成，每个服务端负责若干个主题分区的对外服务或者数据冗余。服务端对外提供消息的提交/消费/分区寻址功能。处于同一个主题分区的各个服务端(Agent)中有一个为前述的主副本，而其余为前述的从副本。各主题分区的Leader分散于服务集群的各个Agent之中，同样Follower也是。生产集群中的生产端(Producer)访问主题分区的主副本以存储其产生的主题消息数据，消费集群中的消费端(Consumer)访问主题分区的主副本以消费主题分区的消息数据。生产端向服务端push消息使得消息可以最快的速度被送达服务端进行存储；消费端从服务端pull消息使得消费者可以根据自身的消费能力以适当的速率消费消息，降低了网络拥塞的可能。

[0023] 管理集群通过检测各服务端和消费端的心跳信号检测其是否存活，根据主题与主题分区对服务端进行寻址。维护各个主题下的主题分区信息，包括主题分区寻址和主题分区的主副本和从副本信息。维护各个服务端的信息，包括拥有的主题分区主副本数量，从副本数量。当某个主题分区的主副本和从副本信息发生变化时，通知相应的主副本或者从副本。

[0024] 一.主体消息文件的逻辑结构

[0025] 主题消息数据以主题消息文件的形式被持久化在服务端的硬盘中。主题在逻辑上可以被认为是一个队列,每条消费都必须指定它的主题。即必须指明把这条消息放到哪个队列中。为了使得消息系统的吞吐率可以水平扩展,物理上把主题消息分为一个或多个主题分区,每个主题分区在物理上对应一个主题消息文件夹,该主题消息文件夹下存储这个主题分区的所有消息和索引文件。具体的,一个主题消息文件的结构如下:

[0026] 1.消息数据按照主题(Topic)进行区分,同一类消息属于同一种主题Topic。主题消息数据即指属于同一主题的消息数据。消息数据的主题需要在使用时,按照需要进行划分。

[0027] 2.主题被分为多个主题分区,分区数量需要在创建主题时指定,也可以在主题创建后,进行主题分区的扩展。

[0028] 3. 每个主题分区按实际情况(即该主题的消息数据的数据量大小),分为多个数据段落(Segment)。每个数据段落的大小大致相等,并且每个数据段落都与一个索引文件一一对应。数据段落的大小,需要在创建主题时确定,也可以在主题创建后进行更改。

[0029] 4.每个主题分区内的主题消息数据每条主题消息数据都被附加在该主题分区中,并且按照其到达服务端的顺序被赋予一个逻辑版本号,该逻辑版本号是连续递增的。每个逻辑版本号唯一标识其对应的主题消息数据,并且对应该主题消息数据在服务端的物理偏移。主题消息数据被顺序写入磁盘,效率非常高。

[0030] 二.主题的创建

[0031] 在创建主题时需指定主题分区数据以及目标服务实例。通过发送命令给这些服务实例进行主题创建。具体过程如下:

[0032] 1.主题创建工具向第三方管理集群(例如zookeeper)获取服务集群的Agent信息。Agent信息主要包括每个Agent下拥有的主题分区Leader和Follower数量。

[0033] 2.以创建一个拥有4个分区、3个副本的主题为例。主题创建工具根据负载情况,选择负载最轻的3个服务端(Agent)作为新的主题的第一个分区的3个副本。然后通知这几个Agent创建主题分区。

[0034] 3.这三个Agent创建分区副本成功后,进入选举主副本(Leader)的过程(选举主副本的过程参考容灾策略部分)。

[0035] 4.主题创建工具通过管理集群检测该主题的上述第一个主题分区的创建情况。如果创建成功并且主副本被选举出,那么表示第一个主题分区创建成功。随后重复步骤2-4以创建剩余的三个主题分区。

[0036] 如果在创建Topic分区过程中,有被选中的Agent下线,分为以下情况处理:如果被选中Agent中没有一个存活的(即被选中的所有Agent都下线),那么重新执行步骤2-4来重新创建新的分区。如果被选中的Agent中依然有存货的,那么通过检测管理集群的状态等待Topic创建成功。

[0037] 三.消息冗余备份

[0038] 主题分区的主副本(Leader)作为消息的接收与消费中心,直接提供对外服务。若干个从副本(Follower)从主副本(Leader)中获取消息,更新本地消息结合。主副本接收并存储来自生产端(Producer)的主题消息数据并且更新从副本,使得从副本与主副本同步。

主/从副本定时的将自己的最小逻辑版本(最小有效逻辑版本号)号和接收到的最大逻辑版本号(最大有效逻辑版本号)上报给管理集群。

[0039] 每一个主题分区的主副本需要维护一个由从副本组成的集合(保持同步副本集)。该集合内的从副本相对于其他从副本拥有更新的消息数据。在创建主题时需要指定一个最小同步副本数量,当保持同步副本集中的从副本数量小于最小同步副本数量时,该主副本不再对外提供接收消息的服务(注意,只是该主题下的该主题分区不再提供接收消息服务,而不是整个服务集群中的服务端都不提供接收服务)。

[0040] 服务端(Agent)为生产端(Producer)发送消息提供三种保障等级:

[0041] A.无保障(UNSAFE)等级,主副本在收到生产端发送的主题消息数据后立刻发送消息提交成功信息给生产端;

[0042] B.快速(FAST)等级,主副本在接收到生产端发送的主题消息数据后同步给所有的从副本,当收到第一个从副本的确定同步信息后,发送消息提交成功信息至生产端;

[0043] C.安全(SAFE)等级,主副本在接收到生产端发送的主题消息数据后同步给所有的从副本,当收到大于等于最小同步副本数量个从副本的确定同步信息后,发送消息提交成功信息至生产端。并且所有回复了确定同步信息的从副本作为保持同步副本集中的元素。

[0044] 在安全等级下,一条主题消息数据只有被保持同步副本集中的所有从副本都复制过去才会被认为已提交。避免了部分数据被写入主副本,但是还没来得及写入从副本就下线了,而造成数据丢失,使得消费端无法消费这些数据。很好的均衡了数据安全性与消息系统吞吐率。从副本可以批量的从主副本复制数据,极大的提高了复制性能,极大减少了从副本与主副本中数据同步的差异。

[0045] 四.容灾策略

[0046] 为了防止当主副本下线而导致该主题分区不可用,采用的容灾策略:

[0047] 1.服务集群中的服务端需要维护一个路由表,该路由表表示当前某主题的某个主题分区中,有哪些服务端可以用。该路由表还可以用于服务寻址,该路由表的更新由管理集群负责。

[0048] 2.当主副本下线时,新的主副本选举过程为:保持同步副本集合中的从副本将自己的最大逻辑版本号(最大有效逻辑版本号)上报给管理集群;根据保持副本同步集合中所有从副本上报的各自的最大逻辑版本号(最大有效逻辑版本号),选出最大逻辑版本号对应的从副本作为主副本并且上报给管理集群。如果具有最大逻辑版本号的从副本有多个,那么选择最先上报给管理集群的从副本作为主副本。

[0049] 3.如果一个从副本下线并且该从副本不再保持同步副本集合中,那么该从副本的下线对整个服务集群没有影响。

[0050] 4.如果一个处于保持同步副本集中的从副本下线,那么主副本需要将该从副本从保持同步副本集合中删除,并且上报给第三方管理集群。当保持同步副本集合中的从副本数量小于最小同步副本数量时,该主题分区停止服务。

[0051] 五.消息系统的处理方法

[0052] 生产端在确定所属主题后,会尝试对该主题的所有主题消分区的主副本进行连接,然后采用轮询的方式依次提交消息。

[0053] 类似日志,mysql的binlog之类的数据传输、保存。本实施例中,服务端主题消息数

据的保存都是用硬盘文件做存储载体。这类主题消息数据具有体量大、要求存储时间长的特点。

[0054] 硬盘IO一般来说是比较慢的，如果一点一点的写入，比如一次写入几个字节或者随机写入，那么硬盘带来的延迟会相当大，最终造成应用的响应速度、吞吐量减小。

[0055] 因此，本实施例中采用内存先缓存一批数据，然后批量顺序写硬盘的方式以提高吞吐量。

[0056] 用内存缓存消息，通常有两个参数需要指明。第一是缓存的大小，第二是刷新磁盘的最长时间间隔。对于服务端来说，需要在创建主题时指定这两个参数。这两个参数也可以在主题创建以后进行修改。对于服务端包括两个表示消息量的参数：消息的有效逻辑版本号，该版本号是最新的被持久化到硬盘的主题消息数据的逻辑版本号，对消费端可见；消息的缓存逻辑版本号，该版本号是主题分区主副本接收到的最新的主题消息数据的逻辑版本号，对消费端不可见。

[0057] 如图4所示为名称为A1的主题消费模式，该主题具有两个主题分区P1和P2。其中，P1、P2中的消息处理属于同一个主题以外没有任何关系，完全不同。每个主题分区有主副本和从副本用于冗余数据，保证数据安全。各个主题分区内部需要维护一个路由表，该路由表中维护该主题分区的主副本与从副本中的消息的有效逻辑版本号以及主副本的消费组信息。各主题分区在同一时刻最多允许同一消费集群中的一个消费端消费其主题消息数据。

[0058] 每个主题分区主副本和从副本的消息的有效逻辑版本号定期(例如，10秒以内)上报给管理集群以便后期监控。消费时，某消费集群中的消费端只从主题分区的主副本中消费主题消息数据，主副本记录该消费集群的消费水平(即已消费的主题消息数据的逻辑版本号)以保证同一消费集群中的消费端不会重复消费主题消息数据。同时，由于只从主题分区的主副本消费主题消息数据，消费水平在正常情况下不需要同步给其他服务器，只需要主副本在本地维护该消费水平即可。消费端轮询各主题分区以达到负载均衡的消费数据的效果。由主副本维护的消费水平定期上报给管理集群，以便监控以及重新选取主题分区的主副本以后尽可能的避免重复消费。消费端的消费模式可以有两种：

[0059] A.快速消费模式(FAST)，主副本将主题消息数据发送给消费端后，立刻更新该消费端所在消费集群的消费水平信息。

[0060] B.安全模式(SAFE)，主副本将主题消息数据发送给消费端后，收到该消费端的接收确认信息后，更新该消费端所在消费集群的消费水平。该模式下，需要设立一个响应超时时间，当超过该时间后，主副本即使没有接收到该消费端的确认信息也会更新消费集群的消费水平。

[0061] 服务端在其所属主题的主题分区发生变化时，将主题分区的变化情况(如主题分区的扩展)通知给与其相连的生产端。生产端动态的添加主题分区并使之生效并且对下线的主题分区定时检查其是否再次上线。使得生产端能够感知主题分区的扩展和主体分区主副本的下线。当一个主题分区的主副本下线时，生产端暂时将其主题消息数据提交给其他未下线的主题分区的主副本。等到下线的主题分区的主副本重新上线或者选举出新的主副本后，向该主题分区提交消息。当一个主题分区处于不可用的状态(即保持同步副本集的从副本数量小于最小同步副本数量)时，生产端将消息提交给剩余的可用状态的主题分区。

[0062] 当一个消费端加入或者下线时，该消费集群需要重新分配分区以达到该主题被正

常消费的目的。具体包括:服务端记录该主题的某消费集群中的消费端的消费水平(即已消费的主题消息数据的逻辑版本号)、负载水平(即消费的主题分区数量)。当消费端上线时,随机选择在该主题的某一个主题分区的主副本进行连接。被连接的主题分区的主副本根据当前该消费集群的负载水平进行判断:

[0063] 如果该主副本所管理的主题分区没有被任何与该消费端属于同一消费集群的其他消费端消费,那么直接接受该消费端的连接请求。并且根据该消费集群负载判断该消费端是否需要连接其他的主题分区,如果需要则告知该消费端有多余分区需要连接。

[0064] 如果该主副本已经被同一消费集群的某一个消费端消费,那么该主副本需要根据那个消费端的消费水平作出相应处理:如果那个消费端只消费了该主题分区,那么主副本给当前消费返回一个合理的主题分区(合理的主题分区是指该主题分区中该消费集群的消费的主体分区数量是否在允许范围之内)主副本的连接信息(连接地址等信息)给该消费端并告知消费端要重新连接;如果那个消费端消费了多个主题分区,那么主副本告知那个消费端断开连接,接受该消费端的连接请求。

[0065] 当一个消费端下线时,根据当前消费组的负载水平选择一个负载最轻的(即连接的主题分区数量最少的)消费端,告知该消费端有多余分区需要连接。

[0066] 当一个主题分区的主副本下线时,与该主副本相连的消费端随机选择一个主题分区的主副本进入轮询等待的过程(轮询间隔可设置为1S),等待该主题分区主副本重新上线或者有新的空余主题分区上线。如果当前剩余的所有主题分区与消费端一一对应,那么告知该消费端继续轮询。

[0067] 实施例二

[0068] 为了简明器件本实施例与实施例一相同的部分在此不再赘述,仅描述实施例二与实施例一的不同部分。

[0069] 本实施例中,主题消息数据在服务端的存储模式有两种。一种是与实施例一相同的文件模式,另一种为内存模式。文件模式适于存储过期时间较长,体量较大的消息数据,包括日志、mysql的binlog、应用的简单活动信息等。

[0070] 内存模式适用于单纯的快速消息传递,这些消息通常都具有生命周期短(可能只有几分钟),要求传输延时小需要相对快速的到达消费端的特点。这种类型的主题消息数据使用文件进行存储不太方便。首先,文件存储是针对生命周期相对较长的消息设计的,不能及时的淘汰过期数据;其次,对于文件存储而言,能够被消费的数据一定是已经落地(即持久化到硬盘的、具有有效逻辑版本号)的主题消息数据,而文件存储模式下主题消息数据被接收到落地之间有时间差,不能及时被消费。

[0071] 内存模式需要在创建主题时指定,一个主题的模式一经指定则不可更改。创建内存模式的主题时,可以指定一个默认的过期时间。如果在生产消息时,没有指定消息过期时间,那么就按默认的过期时间淘汰该消息。

[0072] 内存模式下依赖内存保存消息数据,主题分区接收到生产端产生的主题消息数据后映射到内存中进行保存。一个主题分区的内存占用需要在创建主题时指定(或者采用默认值)。在分配主题分区时,需要了解当前服务器的内存使用情况,以免溢出。

[0073] 各主题分区在内存中以数据段落跳跃表的形式存储(如图5)。数据段落跳跃表包括多个数据层,每个数据层包括如果数据节点。下一数据层的节点数量大于上一数据层的

节点数量，并且下一数据层包括上一数据层的所有节点，位于底层的数据层包括所述数据段落跳跃表的所有节点。每个节点存储该主题分区的主题消息数据的一个段落数据(Segment)、该节点包括其在下一数据层中的指针数据、以及该节点所在数据层的下一节点的指针数据。每一层的节点之间间隔的节点数量是随机的。根据主题消息数据的逻辑版本号对该主题分区的主题消息数据进行索引时，由数据段落跳跃表的顶层向下逐层查找，大大提高了查找的效率。例如图5中需要查找逻辑版本号为117的主题消息数据：

[0074] 1)与顶层(lever3)的第一个节点21比较，117大于21则往后面找；

[0075] 2)与顶层的第二个节点37比较，117大于37，而37为该层链表的最大值，则从37的下面一层(lever2)开始查找；

[0076] 3)与第二层中37后面的一个节点71比较，117大于71，而71为该层链表的最大值，则从71的下面一层(lever3)开始查找；

[0077] 4)与第三层中71后面的一个节点85比较，117大于85则往后面找；

[0078] 5)与第三层中85后面的一个节点117比较，117等于117，找到该节点。

[0079] 本文中所描述的具体实施例仅仅是对本发明精神作举例说明。本发明所属技术领域的技术人员可以对所描述的具体实施例做各种各样的修改或补充或采用类似的方式替代，但并不会偏离本发明的精神或者超越所附权利要求书所定义的范围。

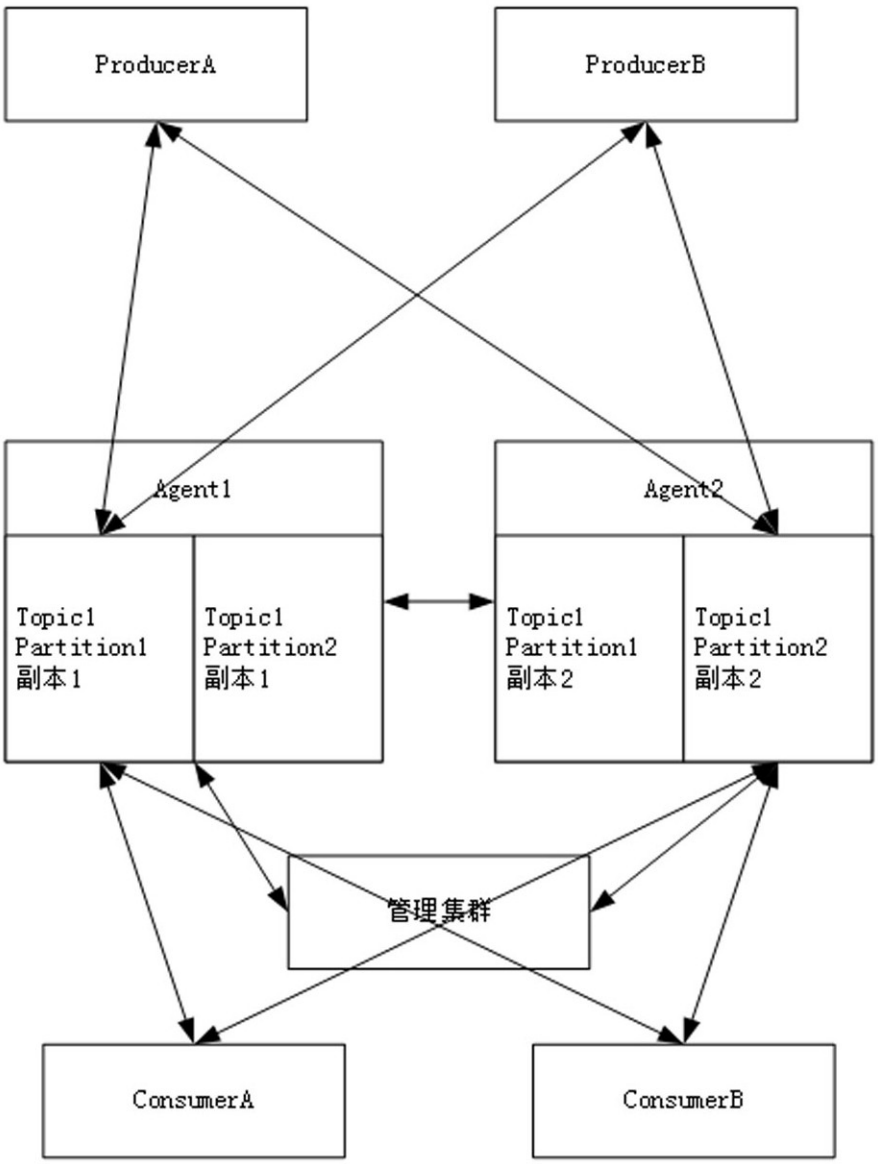

ProducerA
ProducerB
Agent1
Agent2
Topic1
Partition1
副本1
Topic1
Partition2
副本1
Topic1
Partition1
副本2
Topic1
Partition2
副本2
管理集群
ConsumerA
ConsumerB

图1

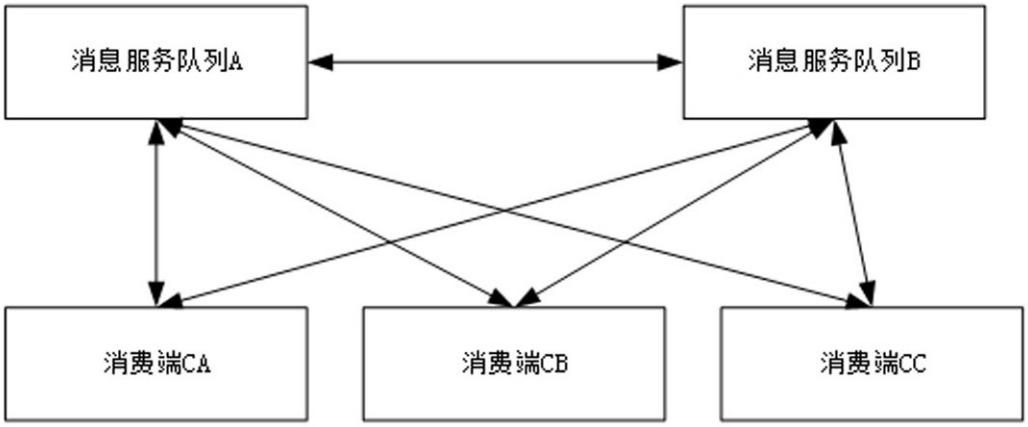


图2

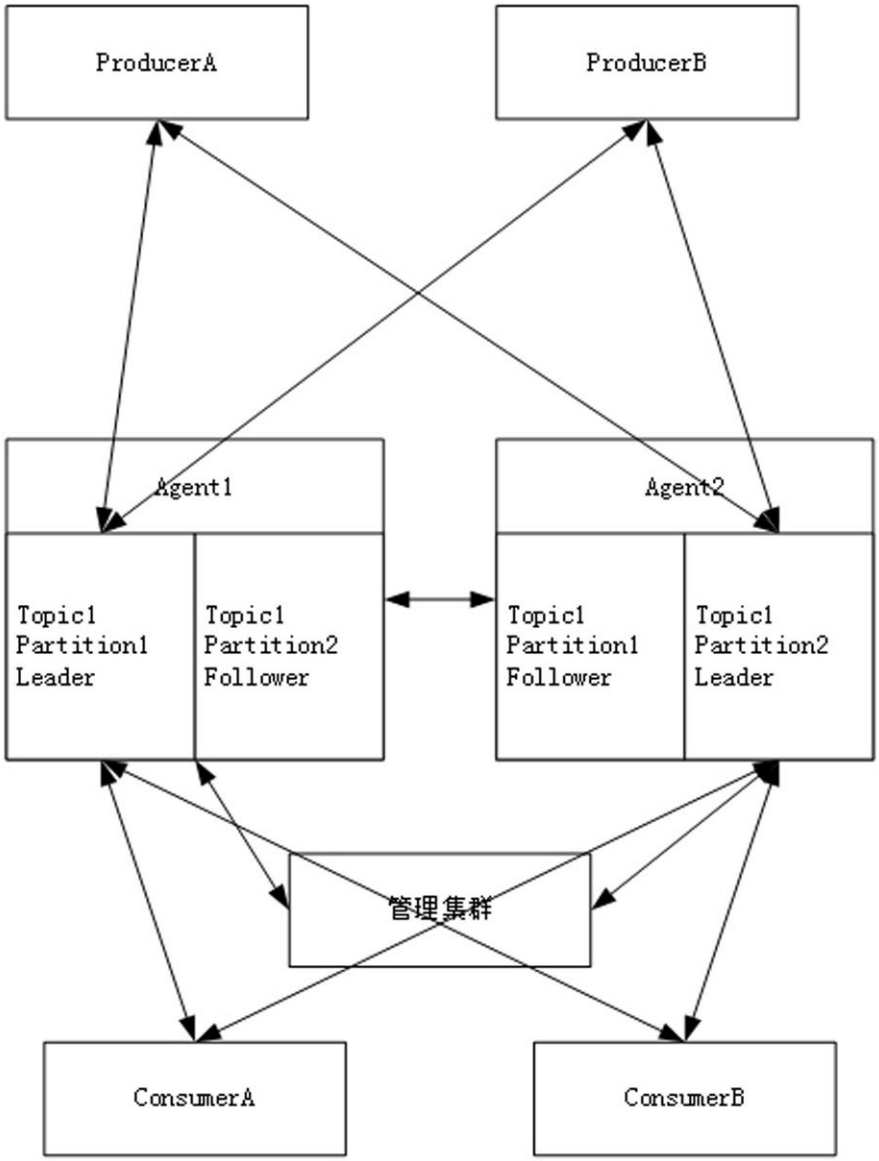

ProducerA
ProducerB
Agent1
Agent2
Topic1
Partition1
Leader
Topic1
Partition2
Follower
Topic1
Partition1
Follower
Topic1
Partition2
Leader
管理集群
ConsumerA
ConsumerB

图3

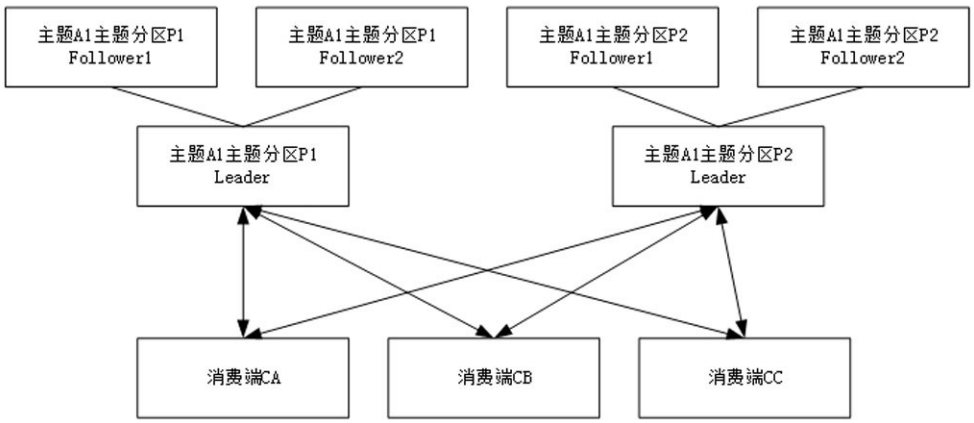

主题A1主题分区P1
Follower1
主题A1主题分区P1
Follower2
主题A1主题分区P2
Follower1
主题A1主题分区P2
Follower2
主题A1主题分区P1
Leader
主题A1主题分区P2
Leader
消费端CA
消费端CB
消费端CC

图4

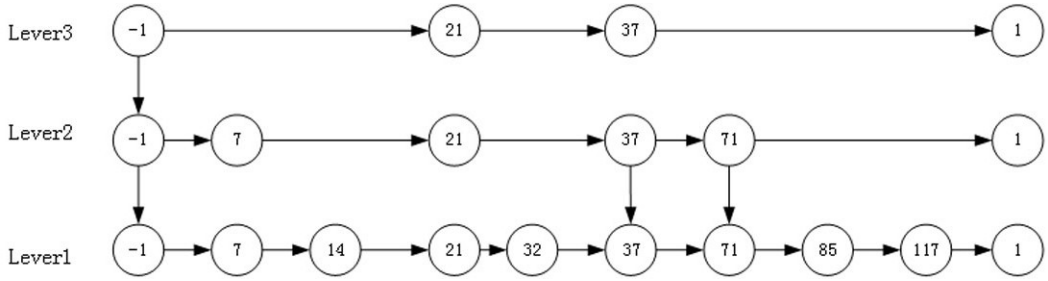

Lever3
-1
21
37
1
Lever2
-1
7
21
37
71
1
Lever1
-1
7
14
21
32
37
71
85
117
1

图5