(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 110147353 A
(43)申请公布日 2019.08.20

(21)申请号 201910331821.6

(22)申请日 2019.04.24

(71)申请人 深圳先进技术研究院
地址 518055 广东省深圳市南山区西丽大学城学苑大道1068号

(72)发明人 石婧文 须成忠 叶可江 王洋

(74)专利代理机构 深圳市科进知识产权代理事务所(普通合伙) 44316
代理人 曹卫良

(51)Int.Cl.
G06F 16/17(2019.01)
G06F 16/21(2019.01)

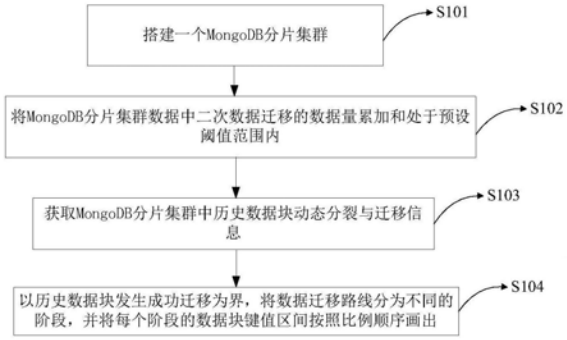
权利要求书2页 说明书7页 附图3页

(54)发明名称

基于日志分析的MongoDB数据迁移监控方法及装置

(57)摘要

本发明涉及电子信息技术领域，具体涉及一种基于日志分析的MongoDB数据迁移监控方法及装置，首先搭建一个MongoDB分片集群；将MongoDB分片集群数据中二次数据迁移的数据量累加和处于预设阈值范围内；获取MongoDB分片集群中历史数据块动态分裂与迁移信息；再以历史数据块发生成功迁移为界，将数据迁移路线分为不同的阶段，并将每个阶段的数据块键值区间按照比例顺序画出。该方法及装置利用了MongoDB配置服务器中的日志数据，观测数据块在不同服务器之间现有分布与过去分布迁移情况，并定义写放大估算公式评估分裂与迁移策略好坏，帮助MongoDB数据库更好地进行预划分和资源分配。与传统的观测方法相比，不受其他因素干扰，使用历史日记数据，结果更加准确。



CN 110147353 A

1.一种基于日志分析的MongoDB数据迁移监控方法,其特征在于,包括以下步骤:

搭建一个MongoDB分片集群,所述MongoDB分片集群包含Shard、Mongos和Config server3种组件;

将所述MongoDB分片集群数据中二次数据迁移的数据量累加和处于预设阈值范围内;

获取所述MongoDB分片集群中历史数据块动态分裂与迁移信息;

以历史数据块发生成功迁移为界,将数据迁移路线分为不同的阶段,并将每个阶段的数据块键值区间按照比例顺序画出。

2.根据权利要求1所述的MongoDB数据迁移监控方法,其特征在于,所述MongoDB数据迁移监控方法还包括:

将每个阶段的数据块键值区间用不同颜色代表不同的服务器填充数据块。

3.根据权利要求1所述的MongoDB数据迁移监控方法,其特征在于,所述MongoDB分片集群数据中二次数据迁移的数据量累加和为transfer size,其计算公式为:

transfer size＝∑clonedBytes;

Mongos获取Config server上的changelog集合数据,transfer size通过遍历changelog集合数据获取,changelog集合数据以字典形式保存;clonedBytes代表数据量累加字节。

4.根据权利要求1所述的MongoDB数据迁移监控方法,其特征在于,所述MongoDB分片集群数据中二次数据迁移的数据量累加计算中采用两种操作类型:

moveChunks.commit:该日志记录从数据块迁出服务器获取,包含有数据块键值信息、迁出服务器、迁入服务器、从属集合名称、拷贝数据量信息;

moveChunks.from:该日志记录从数据块迁移接收服务器获取,包含有数据块键值信息、迁出服务器、迁入服务器、从属集合名称、是否成功信息。

5.根据权利要求1所述的MongoDB数据迁移监控方法,其特征在于,利用Config server上的chunks集合,描绘当前数据块集群中的分布情况,从所述MongoDB分片集群的Changelog集合数据中获取历史数据块动态分裂与迁移信息。

6.根据权利要求5所述的MongoDB数据迁移监控方法,其特征在于,从所述MongoDB分片集群的Changelog集合数据中获取历史数据块动态分裂与迁移信息过程中采用三种操作类型:

moveChunks.from:该日志记录从数据块迁移接收服务器获取,包含有数据块键值信息、迁出服务器、迁入服务器、从属集合名称、是否成功信息;

shardCollection.start:该日志记录由mongos执行创建,指定了初始数据块MinKey、MaxKey所在shard服务器;

multi-split:该日志记录从执行分裂的shard服务器获取,包含分片前数据块信息、分片后数据块信息、集合名称、数据块所在shard服务器信息。

7.根据权利要求6所述的MongoDB数据迁移监控方法,其特征在于,初始数据块的键值区间和所在shard服务器信息从shardCollection.start中获取,之后所有数据块都由已存在数据块分裂而来,从multi-split中获取,数据块迁移信息从moveChunks.from中获取。

8.一种基于日志分析的MongoDB数据迁移监控装置,其特征在于,包括:

集群搭建单元,用于搭建一个MongoDB分片集群,所述MongoDB分片集群包含Shard、

Mongos和Config server3种组件；

阈值单元，用于将所述MongoDB分片集群数据中二次数据迁移的数据量累加和处于预设阈值范围内；

信息获取单元，用于获取所述MongoDB分片集群中历史数据块动态分裂与迁移信息；

键值区间划分单元，用于以历史数据块发生成功迁移为界，将数据迁移路线分为不同的阶段，并将每个阶段的数据块键值区间按照比例顺序画出。

9.一种存储介质，其特征在于，所述存储介质存储有能够实现权利要求1至7中任意一项所述基于日志分析的MongoDB数据迁移监控方法的程序文件。

10.一种处理器，其特征在于，所述处理器用于运行程序，其中，所述程序运行时执行权利要求1至7中任意一项所述的基于日志分析的MongoDB数据迁移监控方法。

基于日志分析的MongoDB数据迁移监控方法及装置

技术领域

[0001] 本发明涉及电子信息技术领域，具体而言，涉及一种基于日志分析的MongoDB数据迁移监控方法及装置。

背景技术

[0002] 随着海量非结构化数据(传感器采集的空间数据、路网数据)源源不断地产生，分布式Nosql数据库，如MongoDB、Hbase等地位日益提高。MongoDB支持数据在集群中分片(shard)存储与副本集(Replica Set)存储两种存储方式。副本集存储的主要目的是利用主从模式进行自动故障恢复功能，而分片存储是为了将键值区间无重叠地划分给不同服务器存储，提高读写吐量。另外，当服务器存储的数据块不均匀时，Mongodb会启动数据迁移模块进行数据块迁移，保证各台服务器存储数据量大致相同。但由于数据可能存在严重不可预测的数据倾斜，分片和迁移过程可能带来很多冗余开销。

发明内容

[0003] 本发明实施例提供了一种基于日志分析的MongoDB数据迁移监控方法及装置，以至少解决现有MongoDB数据分片和迁移过程中存在冗余开销的技术问题。

[0004] 根据本发明的一实施例，提供了一种基于日志分析的MongoDB数据迁移监控方法，包括以下步骤：

[0005] 搭建一个MongoDB分片集群，MongoDB分片集群包含Shard、Mongos和Config server3种组件；

[0006] 将MongoDB分片集群数据中二次数据迁移的数据量累加和处于预设阈值范围内；

[0007] 获取MongoDB分片集群中历史数据块动态分裂与迁移信息；

[0008] 以历史数据块发生成功迁移为界，将数据迁移路线分为不同的阶段，并将每个阶段的数据块键值区间按照比例顺序画出。

[0009] 进一步地，MongoDB数据迁移监控方法还包括：

[0010] 将每个阶段的数据块键值区间用不同颜色代表不同的服务器填充数据块。

[0011] 进一步地，MongoDB分片集群数据中二次数据迁移的数据量累加和为transfer size,其计算公式为：

[0012] transfer size＝∑clonedBytes；

[0013] Mongos可获取Config server上的changelog集合数据，transfer size可通过遍历changelog集合数据获取，changelog集合数据以字典形式保存；clonedBytes代表数据量累加字节。

[0014] 进一步地，MongoDB分片集群数据中二次数据迁移的数据量累加计算中采用两种操作类型：

[0015] moveChunks.commit:该日志记录从数据块迁出服务器获取，包含有数据块键值信息、迁出服务器、迁入服务器、从属集合名称、拷贝数据量信息；

[0016] moveChunks.from:该日志记录从数据块迁移接收服务器获取,包含有数据块键值信息、迁出服务器、迁入服务器、从属集合名称、是否成功信息。

[0017] 进一步地,利用Config server上的chunks集合,描绘当前数据块集群中的分布情况,从MongoDB分片集群的Changelog集合数据中获取历史数据块动态分裂与迁移信息。

[0018] 进一步地,从MongoDB分片集群的Changelog集合数据中获取历史数据块动态分裂与迁移信息过程中采用三种操作类型:

[0019] moveChunks.from:该日志记录从数据块迁移接收服务器获取,包含有数据块键值信息、迁出服务器、迁入服务器、从属集合名称、是否成功信息;

[0020] shardCollection.start:该日志记录由mongos执行创建,指定了初始数据块MinKey、MaxKey所在shard服务器;

[0021] multi-split:该日志记录从执行分裂的shard服务器获取,包含分片前数据块信息、分片后数据块信息、集合名称、数据块所在shard服务器信息。

[0022] 进一步地,初始数据块的键值区间和所在shard服务器信息从shardCollection.start中获取,之后所有数据块都由已存在数据块分裂而来,从multi-split中获取,数据块迁移信息从moveChunks.from中获取。

[0023] 根据本发明的另一实施例,提供了一种基于日志分析的MongoDB数据迁移监控装置,包括:

[0024] 集群搭建单元,用于搭建一个MongoDB分片集群,MongoDB分片集群包含Shard、Mongos和Config server3种组件;

[0025] 阈值单元,用于将MongoDB分片集群数据中二次数据迁移的数据量累加和处于预设阈值范围内;

[0026] 信息获取单元,用于获取MongoDB分片集群中历史数据块动态分裂与迁移信息;

[0027] 键值区间划分单元,用于以历史数据块发生成功迁移为界,将数据迁移路线分为不同的阶段,并将每个阶段的数据块键值区间按照比例顺序画出。

[0028] 一种存储介质,存储介质存储有能够实现上述任意一项基于日志分析的MongoDB数据迁移监控方法的程序文件。

[0029] 一种处理器,处理器用于运行程序,其中,程序运行时执行上述任意一项的基于日志分析的MongoDB数据迁移监控方法。

[0030] 本发明实施例中的基于日志分析的MongoDB数据迁移监控方法及装置,利用了MongoDB配置服务器中的日志数据,观测数据块在不同服务器之间现有分布与过去分布迁移情况,并定义写放大估算公式评估分裂与迁移策略好坏,帮助MongoDB数据库更好地进行预划分和资源分配。与传统的观测方法相比,不受其他因素干扰,使用历史日记数据,结果更加准确。结果直观,通过公式指标或可视化评估呈现分片数据库性能,并能直观体现衡量观察数据迁移策略、分裂机制、键值设计是否合理。

附图说明

[0031] 此处所说明的附图用来提供对本发明的进一步理解,构成本申请的一部分,本发明的示意性实施例及其说明用于解释本发明,并不构成对本发明的不当限定。在附图中:

[0032] 图1为本发明基于日志分析的MongoDB数据迁移监控方法的流程图;

[0033] 图2为本发明基于日志分析的MongoDB数据迁移监控方法的优选流程图；

[0034] 图3为本发明基于日志分析的MongoDB数据迁移监控方法中数据块分裂与迁移过程示意图；

[0035] 图4为本发明基于日志分析的MongoDB数据迁移监控装置的模块图；

[0036] 图5为本发明基于日志分析的MongoDB数据迁移监控装置的优选模块图。

具体实施方式

[0037] 为了使本技术领域的人员更好地理解本发明方案，下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本发明一部分的实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都应当属于本发明保护的范围。

[0038] 需要说明的是，本发明的说明书和权利要求书及上述附图中的术语“第一”、“第二”等是用于区别类似的对象，而不必用于描述特定的顺序或先后次序。应该理解这样使用的数据在适当情况下可以互换，以便这里描述的本发明的实施例能够以除了在这里图示或描述的那些以外的顺序实施。此外，术语“包括”和“具有”以及他们的任何变形，意图在于覆盖不排他的包含，例如，包含了一系列步骤或单元的过程、方法、系统、产品或设备不必限于清楚地列出的那些步骤或单元，而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它步骤或单元。

[0039] 现有工具或方法虽然实时性好，能够通过测量网络或者磁盘I/O，绘制出长期跟踪曲线，但是难以体现资源消耗与上层机制(如迁移策略)的关系。测量结果易受到各种干扰，如I/O的观察往往混合了数据库其他I/O影响或者其他应用的I/O干扰，并且难以从一个混合指标中分解出实际需要的资源消耗。这不利于找到性能问题存在、评估上层策略、改进数据库机制等。本发明提出了一种从日志文件准确提取数据块迁移信息的方案，可用来衡量数据迁移策略、分裂机制、键值设计是否合理。

[0040] 分布式数据库比起单机数据库引入了很多的新问题，如数据在服务器之间的分发与迁移问题，这些新过程带来的开销和影响往往被人们忽视，可视化与量化公式可以帮助数据库管理员更好地判断预划分效果。但数据块的分裂与迁移是一个持续进行的长期过程，中间伴有数据块动态分裂，迁移过程中部分数据可能发生多次冗余网络传输，以上各种因素增加了观测复杂性，现有技术中还没有具体方法将迁移分裂过程中的写放大和冗余网络传输直观观测与量化。为此，我们针对分布式MongoDB数据库集群提出了新的监控分析方法。

[0041] 其中MongoDB分片集群由Shard、Mongos和Config server3种组件构成：

[0042] (1)Mongos负责提供集群访问接口，保证集群一致性，并将用户请求正确路由到对应的Shard。同时，Mongos提供了用户命令行工具mongos shell，通过mongos shell我们可以获取数据库与数据集合少量统计信息。数据库中的部分数据来源于shell命令。

[0043] (2)Shard负责存储数据，数据以chunk形式在Shard集群中进行存储和迁移。

[0044] (3)Config server保存Shard集群所有元数据，Mongos连接Config server获取元数据信息。其中元数据信息包含日志集changelog和chunks集合，changelog集合存储了数

据库变动情况，chunks集合存储了当前所有数据块信息。

[0045] 以往的数据库监控方案与工具大多直接测量资源利用情况，如：MongoDB自带的监控工具mongostat可以显示执行操作花费时间、cache命中情况；MongoDB官网提供的网页监控工具MMS(MongoDB Monitoring Service)可以检测硬件事件。针对MongoDB等nosql数据库性能改进的现有技术大多以插入、查询时间代价和存储代价为指标，没有进一步的数据迁移分析。

[0046] 本发明的技术方案可以衡量当前数据库迁移和配置是否合理，并可在分片集群中可视化观测历史键值区间分布、数据块分裂、数据块在不同服务器之间的迁移。

[0047] 实施例1

[0048] 根据本发明一实施例，提供了一种基于日志分析的MongoDB数据迁移监控方法，参见图1，包括以下步骤：

[0049] S101：搭建一个MongoDB分片集群，MongoDB分片集群包含Shard、Mongos和Config server3种组件；

[0050] S102：将MongoDB分片集群数据中二次数据迁移的数据量累加和处于预设阈值范围内，即数据量累加和越小越好；

[0051] S103：获取MongoDB分片集群中历史数据块动态分裂与迁移信息；

[0052] S104：以历史数据块发生成功迁移为界，将数据迁移路线分为不同的阶段，并将每个阶段的数据块键值区间按照比例顺序画出。

[0053] 本发明的基于日志分析的MongoDB数据迁移监控方法，利用了MongoDB配置服务器中的日志数据，观测数据块在不同服务器之间现有分布与过去分布迁移情况，并定义写放大估算公式评估分裂与迁移策略好坏，帮助MongoDB数据库更好地进行预划分和资源分配。与传统的观测方法相比，不受其他因素干扰，使用历史日记数据，结果更加准确。结果直观，通过公式指标或可视化评估呈现分片数据库性能，并能直观体现衡量观察数据迁移策略、分裂机制、键值设计是否合理。

[0054] 作为优选的技术方案中，参见图2，该MongoDB数据迁移监控方法方法还包括：

[0055] S105：将每个阶段的数据块键值区间用不同颜色代表不同的服务器填充数据块，可视化了整个数据集合数据块的分裂、迁移过程。

[0056] 下面以具体实施例，对本方法进行详细说明，本发明一种基于日志分析的MongoDB数据迁移监控方法中：

[0057] 首先，搭建一个MongoDB分片集群，包含Shard、Mongos和Config server3种组件，创建分片集合，并向分片集合中进行数据处理。

[0058] 平衡开销计算方法：用transfer size代表数据在平衡组件指导下进行二次数据迁移的数据量累加和。在使数据块在分片集群中达到尽可能均匀分布同时，数据迁移的网络传输资源开销越小越好。定义如下公式：

[0059] transfer size＝∑clonedBytes；

[0060] transfer size可通过遍历changelog集合获取，Mongos可以获取Config server上的changelog数据，clonedeBytes代表数据量累加字节。该数据以字典形式保存：

[0061] {"_id":"silverdew-2018-10-06T20:42:02.820+0800-5bb8ad9a11fa6074beda8f4b","server":"silverdew","clientAddr":"127.0.0.1:

33058","time":ISODate("2018-10-06T12:42:02.820Z"),"what":"moveChunk.commit","ns":"two_zero_one_seven.Jan_sh_hil_fourlogic","details":{"min":{"key":{"$minKey":1}},"max":{"key":"03100002001021021033022023100231"},"from":"shard0000","to":"shard0001","counts":{"cloned":NumberLong(1),"clonedBytes":NumberLong(310),"catchup":NumberLong(0),"steady":NumberLong(0)}}}。

[0062] “what”属性代表操作类型,写放大比例计算过程中主要用到两种操作类型:

[0063] “moveChunks.commit”:该日志记录从数据块迁出服务器获取,包含有数据块键值信息、迁出服务器、迁入服务器、从属集合名称、拷贝数据量等信息。

[0064] “moveChunks.from”:该日志记录从数据块迁移接收服务器获取,包含有数据块键值信息、迁出服务器、迁入服务器、从属集合名称、是否成功等信息。在该种操作类型中,transfer size为历史记录中经过moveChunks.from确认迁移成功的拷贝数据量累加和。

[0065] 历史数据块分裂迁移可视化方法:利用Config server上的chunks集合,描绘当前数据块集群中的分布情况,从Changelog获取历史数据块动态分裂与迁移情况。

[0066] 可视化过程中主要的“what”操作类型有:

[0067] “moveChunks.from”:该日志记录从数据块迁移接收服务器获取,包含有数据块键值信息、迁出服务器、迁入服务器、从属集合名称、是否成功等信息。在该种操作类型中,transfer size为历史记录中经过moveChunks.from确认迁移成功的拷贝数据量累加和。

[0068] "shardCollection.start":该日志记录由mongos执行创建,指定了初始数据块【MinKey,MaxKey】所在shard服务器。

[0069] "multi-split":该日志记录从执行分裂的shard服务器获取,包含分片前数据块信息、分片后数据块信息、集合名称、数据块所在shard服务器等信息。

[0070] 以数据块发生成功迁移为界,将数据迁移路线分为不同的阶段,并将每个阶段的数据块键值区间按照比例顺序画出,用不同颜色代表不同的shard服务器填充数据块,可视化了整个数据集合数据块的分裂、迁移过程。其中初始数据块的键值区间和所在shard服务器信息从"shardCollection.start"中获取,之后所有数据块都由已存在数据块分裂而来,因此都从"multi-split"中获取,数据块迁移信息从“moveChunks.from”中获取。

[0071] 参见图3,图3中不同数据块之间有间隔,数据块长度与负责存储的键值区间成正比,绿色、紫色、蓝色分别代表数据块所在不同的服务器(shard000为蓝色、shard001为绿色、shard002为紫色)。除了stage0到stage1是由数据块第一次分裂造成,之后新的stage都是由数据迁移导致。

[0072] 实施例2

[0073] 根据本发明另一实施例,提供了一种基于日志分析的MongoDB数据迁移监控装置,参见图4,包括:

[0074] 集群搭建单元201,用于搭建一个MongoDB分片集群,MongoDB分片集群包含Shard、Mongos和Config server3种组件;

[0075] 阈值单元202,用于将MongoDB分片集群数据中二次数据迁移的数据量累加和处于预设阈值范围内;

[0076] 信息获取单元203,用于获取MongoDB分片集群中历史数据块动态分裂与迁移信息;

[0077] 键值区间划分单元204，用于以历史数据块发生成功迁移为界，将数据迁移路线分为不同的阶段，并将每个阶段的数据块键值区间按照比例顺序画出。

[0078] 本发明实施例中的基于日志分析的MongoDB数据迁移监控装置，利用了MongoDB配置服务器中的日志数据，观测数据块在不同服务器之间现有分布与过去分布迁移情况，并定义写放大估算公式评估分裂与迁移策略好坏，帮助MongoDB数据库更好地进行预划分和资源分配。与传统的观测方法相比，不受其他因素干扰，使用历史日记数据，结果更加准确。结果直观，通过公式指标或可视化评估呈现分片数据库性能，并能直观体现衡量观察数据迁移策略、分裂机制、键值设计是否合理。

[0079] 作为优选的技术方案中，参见图5，该装置还包括：

[0080] 颜色填充单元205，用于将每个阶段的数据块键值区间用不同颜色代表不同的服务器填充数据块，可视化了整个数据集合数据块的分裂、迁移过程。

[0081] 实施例3

[0082] 一种存储介质，存储介质存储有能够实现上述任意一项基于日志分析的MongoDB数据迁移监控方法的程序文件。

[0083] 实施例4

[0084] 一种处理器，处理器用于运行程序，其中，程序运行时执行上述任意一项的基于日志分析的MongoDB数据迁移监控方法。

[0085] 上述本发明实施例序号仅仅为了描述，不代表实施例的优劣。

[0086] 在本发明的上述实施例中，对各个实施例的描述都各有侧重，某个实施例中没有详述的部分，可以参见其他实施例的相关描述。

[0087] 在本申请所提供的几个实施例中，应该理解到，所揭露的技术内容，可通过其它的方式实现。其中，以上所描述的系统实施例仅仅是示意性的，例如单元的划分，可以为一种逻辑功能划分，实际实现时可以有另外的划分方式，例如多个单元或组件可以结合或者可以集成到另一个系统，或一些特征可以忽略，或不执行。另一点，所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口，单元或模块的间接耦合或通信连接，可以是电性或其它的形式。

[0088] 作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0089] 另外，在本发明各个实施例中的各功能单元可以集成在一个处理单元中，也可以是各个单元单独物理存在，也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现，也可以采用软件功能单元的形式实现。

[0090] 集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用时，可以存储在一个计算机可读取存储介质中。基于这样的理解，本发明的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质中，包括若干指令用以使得一台计算机设备(可为个人计算机、服务器或者网络设备等)执行本发明各个实施例方法的全部或部分步骤。而前述的存储介质包括：U盘、只读存储器(ROM，Read-Only Memory)、随机存取存储器(RAM，Random Access Memory)、移动硬盘、磁碟或者光盘等各种可以存储程序代码的介质。

[0091] 以上所述仅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，在不脱离本发明原理的前提下，还可以做出若干改进和润饰，这些改进和润饰也应视为本发明的保护范围。

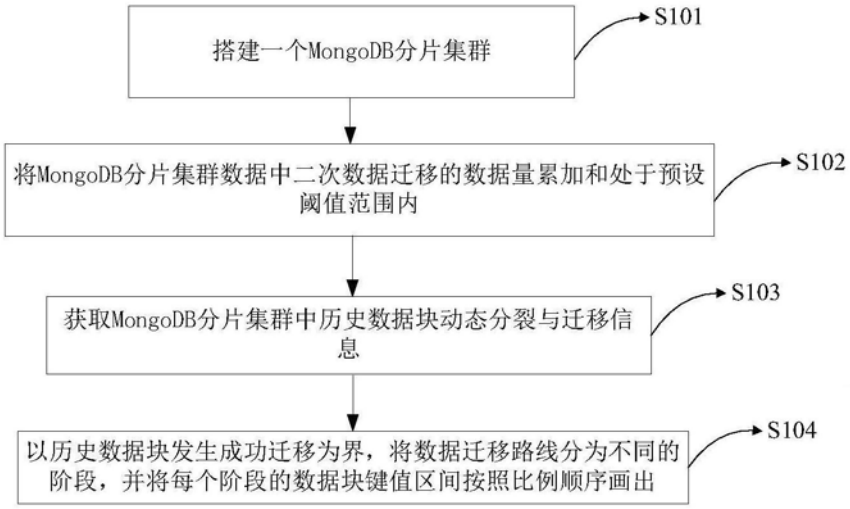

搭建一个MongoDB分片集群
S101
将MongoDB分片集群数据中二次数据迁移的数据量累加和处于预设阈值范围内
S102
获取MongoDB分片集群中历史数据块动态分裂与迁移信息
S103
以历史数据块发生成功迁移为界，将数据迁移路线分为不同的阶段，并将每个阶段的数据块键值区间按照比例顺序画出
S104

图1

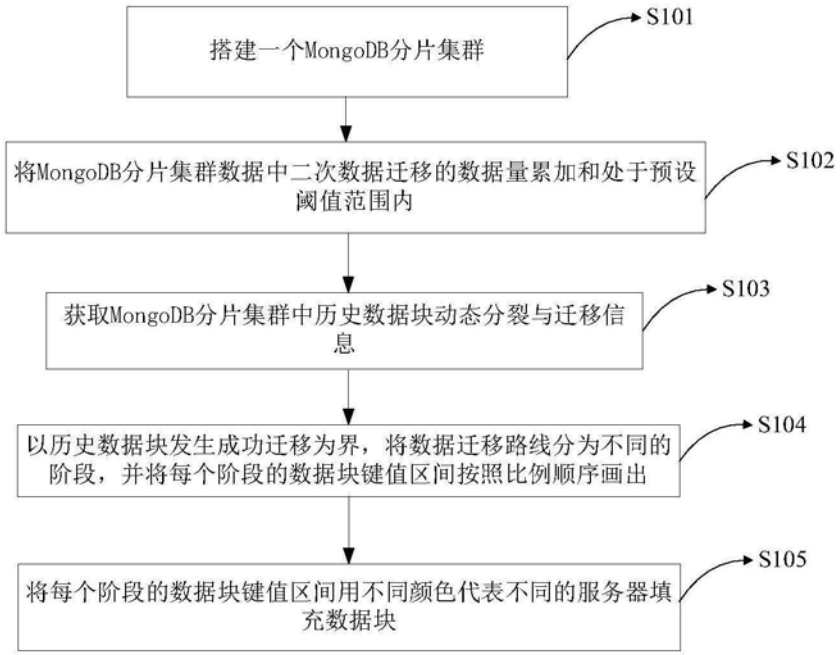

搭建一个MongoDB分片集群
S101
将MongoDB分片集群数据中二次数据迁移的数据量累加和处于预设阈值范围内
S102
获取MongoDB分片集群中历史数据块动态分裂与迁移信息
S103
以历史数据块发生成功迁移为界，将数据迁移路线分为不同的阶段，并将每个阶段的数据块键值区间按照比例顺序画出
S104
将每个阶段的数据块键值区间用不同颜色代表不同的服务器填充数据块
S105

图2

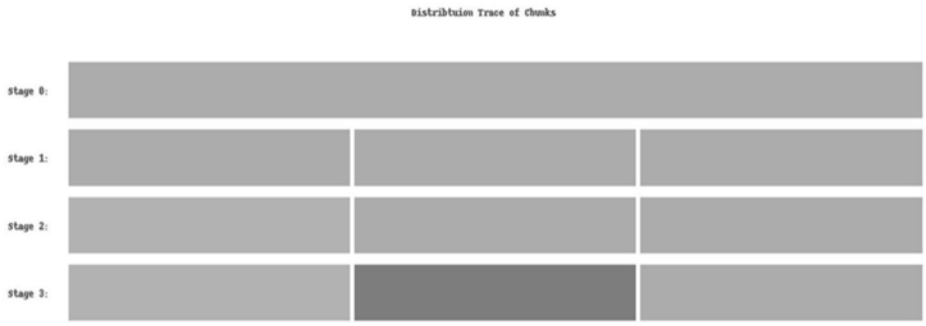

Distribtuion Trace of Chunks
Stage 0:
Stage 1:
Stage 2:
Stage 3:

图3

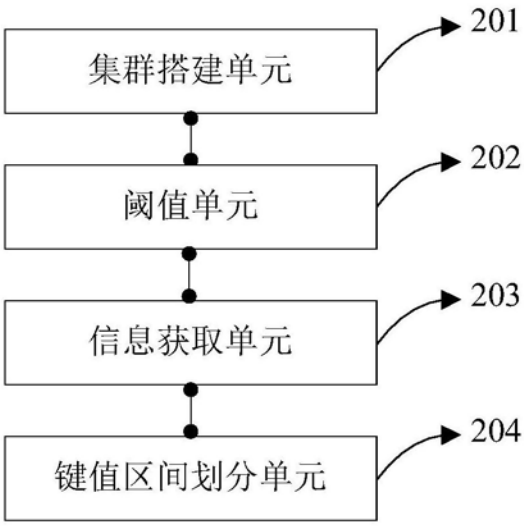

集群搭建单元
201
阈值单元
202
信息获取单元
203
键值区间划分单元
204

图4

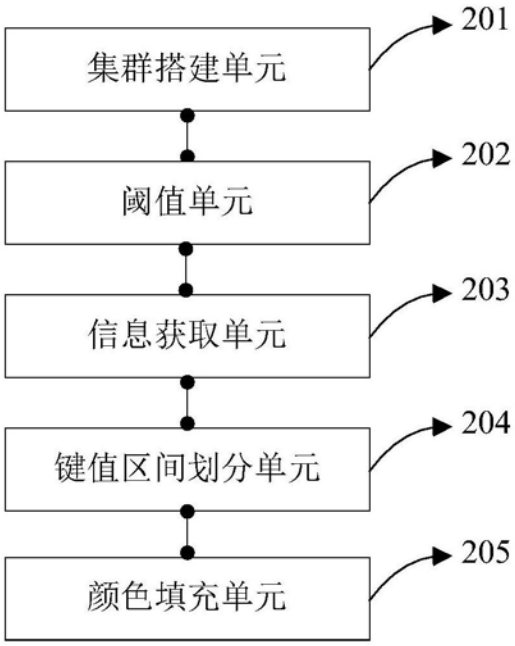


图5