



(12) 发明专利

(10) 授权公告号 CN 110825733 B

(45) 授权公告日 2022. 08. 09

(21) 申请号 201910951134.4

(22) 申请日 2019.10.08

(65) 同一申请的已公布的文献号  
申请公布号 CN 110825733 A

(43) 申请公布日 2020.02.21

(73) 专利权人 华中科技大学  
地址 430074 湖北省武汉市洪山区珞喻路  
1037号

(72) 发明人 曹强 杨阳 杨李

(74) 专利代理机构 华中科技大学专利中心  
42201

专利代理师 曹葆青 李智

(51) Int. Cl.

G06F 16/22 (2019.01)

G06F 16/2458 (2019.01)

(56) 对比文件

CN 106021560 A, 2016.10.12

CN 104899297 A, 2015.09.09

US 2010235335 A1, 2010.09.16

YangYang等.EdgeDB: An Efficient Time-Series Database for Edge Computing.《IEEE Access》.2019,第7卷

审查员 尤鑫

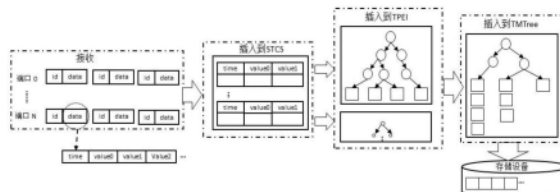
权利要求书2页 说明书8页 附图5页

(54) 发明名称

一种面向多采样流的时间序列数据管理方法及系统

(57) 摘要

本发明公开了一种面向多采样流的时间序列数据管理方法及系统,属于数据存储领域,包括:将传感器采集到的数据点组织为数据元组后,按时间顺序插入到对应的相关流单表中;当相关流单表中的数据达到第一阈值后,按照时间范围将其中的数据划分为不同的子表,并将子表的索引信息插入到对应的时间分区弹性索引中;将子表的地址插入到时间合并树中,以使得其叶子节点索引所有传感器在相同时间范围内产生的子表;定期检查时间合并树最左端的叶子节点,若已索引到所有传感器的子表,或者长时间持续未接收到新数据,则将该叶子节点索引的所有子表通过一次写操作写入存储介质中,并回收该叶子节点。本发明能够提高多采样流时间序列数据的管理效率。



1. 一种面向多采样流的时间序列数据管理方法,其特征在于,包括:

将传感器在同一时间采集到的多个数据点组织为一个数据元组后,按时间顺序插入到该传感器对应的相关流单表中;

当相关流单表中的数据达到第一阈值后,按照时间范围将其中的数据划分为不同的子表,并将子表的索引信息插入到对应的时间分区弹性索引中;

将所有传感器的子表的地址插入到全局的时间合并树中,以使得所述时间合并树的叶子节点索引所有传感器在相同时间范围内产生的子表;

定期检查所述时间合并树最左端的叶子节点,若该叶子节点已索引到所有传感器的子表,或者该叶子节点持续未接收到新数据的时间超过了第二阈值,则将该叶子节点索引的所有子表通过一次写操作写入存储介质中,并回收该叶子节点;

其中,数据点为一个传感器在一个时间点采集到的一个数据值,数据元组为一个传感器在一个时间点采集到的多个数据值,相关流单表为用于存储单个传感器采集的数据元组的表结构,时间分区弹性索引为用于索引子表的结构,时间合并树是一种按照时间范围进行分区的树结构;

所述时间合并树的内部节点记录的信息包括:所覆盖的子表的时间范围和子节点的地址;所述时间合并树的叶子节点记录的信息包括:被索引的子表的时间范围,被索引的子表的内存地址,用于标识各传感器在该时间范围内的子表是否已被索引的位图信息,以及用于记录该叶子节点持续未接收到新数据的时间的计时器;

其中,叶子节点中每次索引到新的子表,其计时器会被重置为0;

将时间合并树最左端的叶子节点索引的所有子表通过一次写操作写入存储介质中,包括:

分配一块空闲的内存区域,根据叶子节点leaf记录的内存地址信息获取到叶子节点leaf索引的所有子表,进行压缩并顺序地写入所分配的内存区域;通过一次写操作,将该内存区域中的数据写入存储介质;

其中,叶子节点leaf为时间合并树最左端的叶子节点;

所述时间分区弹性索引包括一个或多个时间分区树,其中一个时间分区树位于内存中,用于索引对应的传感器新产生的子表,其余时间分区树位于存储介质中,用于索引对应的传感器旧的子表;

其中,时间分区树是一种按照时间范围进行分区的k叉树索引结构,k为正整数;时间分区树的内部节点记录的信息包括:所覆盖的子表的时间范围和子节点的地址;时间分区树的叶子节点用于索引不同时间范围的子表;

时间分区树的内部节点记录的信息还包括其子节点索引的数据的聚合值,时间分区树的叶子节点记录的信息还包括被索引的子表的聚合值;

所述面向多采样流的时间序列数据管理方法,还包括:

用户向服务器发送查询请求,所述查询请求包括被查询的传感器ID、被查询数据的时间范围srchR以及结果类型;服务器接收到所述查询请求后,按照如下步骤处理所述查询请求:

(S1) 根据传感器ID确定被查询的时间分区弹性索引TPEI,并根据属于时间分区弹性索引TPEI的各时间分区树的时间范围和时间范围srchR,确定需要被查询的时间分区树

srchTPtree;

(S2) 根据时间范围srchR确定时间分区树srchTPtree中需要被查询的内部节点srchNode,并读取内部节点srchNode;

(S3) 若结果类型为原始数据,则根据时间范围srchR和内部节点srchNode获得被查询的子表的地址后,访问相应的子表以获取被查询的原始数据并返回给用户,查询请求处理结束;若结果类型为聚合值,则转入步骤(S4);

(S4) 若时间范围srchR与任意一个子表的时间范围重合,则直接从内部节点srchNode中获取该子表的聚合值并返回给用户,查询请求处理结束;否则,根据子表的时间范围将时间范围srchR划分为不同的时间段,对于其中覆盖到完整子表的时间段,直接从内部节点srchNode中获取被覆盖的子表的聚合值,对于其余的时间段,从内部节点srchNode中获取子表的地址后,访问相应的子表以获取该时间段内的原始数据,结合所获取到的原始数据和聚合值计算被查询的数据的聚合值,并返回给用户,所述查询请求处理结束。

2. 如权利要求1所述的面向多采样流的时间序列数据管理方法,其特征在于,还包括:定期获取内存中的时间分区树,记为时间分区树TPtree;

按照分层遍历的顺序将时间分区树TPtree的所有内部节点数据序列化后,写入存储介质中,并在内存中记录时间分区树TPtree的必要信息;

回收时间分区树TPtree所占的内存空间后,为时间分区树TPtree所属的时间分区弹性索引建立一个新的时间分区树,用于索引对应传感器新产生的子表;

其中,时间分区树的必要信息包括索引数据的时间范围、树的层数、每一层索引数据的开始时间以及存储介质中用于保存该时间分区树的文件名。

3. 如权利要求1所述的面向多采样流的时间序列数据管理方法,其特征在于,位于内存中的时间分区树,每次将新的子表索引插入到其叶子节点后,从该叶子节点开始,逐层向上更新上层节点的聚合值,直至根节点的聚合值更新结束。

4. 如权利要求1所述的面向多采样流的时间序列数据管理方法,其特征在于,所述步骤(S2)中,读取内部节点srchNode,包括:

若时间分区树srchTPtree在内存中,则直接从内存中读取内部节点srchNode;

若时间分区树srchTPtree在存储介质中,则从内存中读取时间分区树srchTPtree的必要信息,以获得存储介质中用于保存时间分区树srchTPtree的文件F;按照分层遍历的顺序确定内部节点srchNode在时间分区树srchTPtree的序号,以确定内部节点srchNode在文件F中的偏移量off,根据偏移量off访问存储介质中的文件F,以读取内部节点srchNode。

5. 一种面向多采样流的时间序列数据管理系统,包括处理器和计算机可读存储介质,其特征在于,所述计算机可读存储介质用于存储可执行程序;

所述处理器用于读取所述计算机可读存储介质中存储的可执行程序,执行权利要求1-4任一项所述的面向多采样流的时间序列数据管理方法。

## 一种面向多采样流的时间序列数据管理方法及系统

### 技术领域

[0001] 本发明属于数据存储领域,更具体地,涉及一种面向多采样流的时间序列数据管理方法及系统。

### 背景技术

[0002] 随着物联网的发展,大量高采样频率的、多采样流的传感器被广泛地部署在各个领域中,这些传感器会产生大量的时间序列数据(time-series data)。时序数据就是在时间上分布的一系列数值,生活中常见的时序数据包括股票价格、广告数据、气温变化、网站的PV/UV、个人健康数据、工业传感器数据、服务器系统监控数据(比如CPU和内存占用率)、车联网等。采集时间序列数据,能够有效地帮助人们监控、分析和预测周边环境或者设备等的状态。

[0003] 新兴的物联网应用不仅需要高效地收集和存储这些大量的时间序列数据流,还需要可以实时地查询和分析这些数据。目前的时间序列数据库基本上是基于单个采样流的处理方式,它们单独地为每个采样流建一个独立的表结构,并为每个流建立相应的索引结构,最后定期地将每个采样流的数据写入存储设备。如图1(a)所示,对于同时采集的三个数据流 $A_{CPU}$ 、 $A_{MEM}$ 和 $A_{DISK}$ ,会分别建立三个独立的表结构来存储每一个数据流的数据。在这种处理方式下,对于采集到的不同数据会分别进行组织和存储,可以有效地处理具有小规模、单采样流的传感器的场景。

[0004] 然而,随着大数据时代的到来,越来越多的高采样频率的、多采样流的传感器被使用,它们会产生大量的时间序列数据流,这些数据需要被用户通过各种方法来及时地处理。目前,对于多采样流的传感器所采集到的时间序列数据,基本上沿用了现有的基于单个采样流的处理方式,这种处理方式既不能满足插入这些数据流时的吞吐量要求,也不能对这些数据流进行及时地查询和分析。

### 发明内容

[0005] 针对现有技术的缺陷和改进需求,本发明提供了一种面向多采样流的时间序列数据管理方法及系统,其目的在于,提高多采样流时间序列数据的管理效率。

[0006] 为实现上述目的,按照本发明的第一方面,提供了一种面向多采样流的时间序列数据管理方法,包括:

[0007] 将传感器在同一时间采集到的多个数据点组织为一个数据元组后,按时间顺序插入到该传感器对应的相关流单表中;

[0008] 当相关流单表中的数据达到第一阈值后,按照时间范围将其中的数据划分为不同的子表,并将子表的索引信息插入到对应的时间分区弹性索引中;

[0009] 将所有传感器的子表的地址插入到全局的时间合并树中,以使得时间合并树的叶子节点索引所有传感器在相同时间范围内产生的子表;

[0010] 定期检查时间合并树最左端的叶子节点,若该叶子节点已索引到所有传感器的子

表,或者该叶子节点持续未接收到新数据的时间超过了第二阈值,则将该叶子节点索引的所有子表通过一次写操作写入存储介质中,并回收该叶子节点;

[0011] 其中,数据点为单个传感器在一个时间点采集到的一个数据值,数据元组为单个传感器在一个时间点采集到的多个数据值,相关流单表为用于存储单个传感器采集的数据元组的表结构,时间分区弹性索引为用于索引子表的结构,时间合并树是一种按照时间范围进行分区的树结构。

[0012] 本发明利用相关流单表将传感器同一时间采集到的多个数据点组织到一起,使得服务器可以一次处理多个数据流,也能够关联查询的性能;按照时间范围将传感器采集到的数据划分为不同的子表,利用全局的时间合并树合并所有传感器在相同时间范围内产生的子表,并通过一次写操作将这些子表写入底层存储介质中,有效减少了写IO的数量,提升了写性能。总的来说,本发明能够提高多采样流时间序列数据的管理效率。

[0013] 本发明在时间合并树最左端叶子节点已索引到所有传感器的子表,或者该叶子节点持续未接收到新数据的时间过长时,均会将该叶子节点索引的子表写入底层存储介质,能够防止由于某些传感器在该时间范围内的子表丢失,而导致其他传感器的子表不能写入底层存储介质。

[0014] 进一步地,时间合并树的内部节点记录的信息包括:所覆盖的子表的时间范围和子节点的地址;时间合并树的叶子节点记录的信息包括:被索引的子表的时间范围,被索引的子表的内存地址,用于标识各传感器在该时间范围内的子表是否已被索引的位图信息,以及用于记录该叶子节点持续未接收到新数据的时间的计时器;

[0015] 其中,叶子节点中每次索引到新的子表,其计时器会被重置为0。

[0016] 进一步地,将时间合并树最左端的叶子节点索引的所有子表通过一次写操作写入存储介质中,包括:

[0017] 分配一块空闲的内存区域,根据叶子节点leaf记录的内存地址信息获取到叶子节点leaf索引的所有子表,进行压缩并顺序地写入所分配的内存区域;通过一次写操作,将该内存区域中的数据写入存储介质;

[0018] 其中,叶子节点leaf为时间合并树最左端的叶子节点。

[0019] 进一步地,时间分区弹性索引包括一个或多个时间分区树,其中一个时间分区树位于内存中,用于索引对应的传感器新产生的子表,其余时间分区树位于存储介质中,用于索引对应的传感器旧的子表;

[0020] 其中,时间分区树是一种按照时间范围进行分区的k叉树索引结构,k为正整数;时间分区树的内部节点记录的信息包括:所覆盖的子表的时间范围和子节点的地址;时间分区树的叶子节点用于索引不同时间范围的子表。

[0021] 进一步地,本发明第一方面提供的面向多采样流的时间序列数据管理方法,还包括:定期获取内存中的时间分区树,记为时间分区树TPtree;

[0022] 按照分层遍历的顺序将时间分区树TPtree的所有内部节点数据序列化后,写入存储介质中,并在内存中记录时间分区树TPtree的必要信息;

[0023] 回收时间分区树TPtree所占的内存空间后,为时间分区树TPtree所属的时间分区弹性索引建立一个新的时间分区树,用于索引对应传感器新产生的子表;

[0024] 其中,时间分区树的必要信息包括索引数据的时间范围、树的层数、每一层索引数

据的开始时间以及存储介质中用于保存该时间分区树的文件名。

[0025] 本发明所提供的面向多采样流的时间序列数据管理方法,对于一个时间分区弹性索引,仅在内存中保留一个时间分区树接收新的子表索引,其他时间分区树则被优化地存储在底层存储介质中,同时在内存中保存以写入底层存储介质的时间分区树的必要信息,在后续查询时,结合时间分区树的必要信息,通过一次IO即可从存储介质中读取所需要的节点数据,因此,本发明能够减少查询延迟,提高查询效率。

[0026] 进一步地,时间分区树的内部节点记录的信息还包括其子节点索引的数据的聚合值,时间分区树的叶子节点记录的信息还包括被索引的子表的聚合值。

[0027] 本发明通过在时间分区树的内部节点记录其子节点索引的数据的聚合值(最大值、最小值、平均值等等),使得在后续查询中能够借助内部节点记录的聚合值加速查询,提高查询效率。

[0028] 进一步地,位于内存中的时间分区树,每次将新的子表索引插入到其叶子节点后,从该叶子节点开始,逐层向上更新上层节点的聚合值,直至根节点的聚合值更新结束。

[0029] 进一步地,本发明第一方面提供的面向多采样流的时间序列数据管理方法,还包括:

[0030] 用户向服务器发送查询请求,查询请求包括被查询的传感器ID、被查询数据的时间范围srchR以及结果类型;服务器接收到查询请求后,按照如下步骤处理查询请求:

[0031] (S1) 根据传感器ID确定被查询的时间分区弹性索引TPEI,并根据属于时间分区弹性索引TPEI的各时间分区树的时间范围和时间范围srchR,确定需要被查询的时间分区树srchTPtree;

[0032] (S2) 根据时间范围srchR确定时间分区树srchTPtree中需要被查询的内部节点srchNode,并读取内部节点srchNode;

[0033] (S3) 若结果类型为原始数据,则根据时间范围srchR和内部节点srchNode获得被查询的子表的地址后,访问相应的子表以获取被查询的原始数据并返回给用户,查询请求处理结束;若结果类型为聚合值,则转入步骤(S4);

[0034] (S4) 若时间范围srchR与任意一个子表的时间范围重合,则直接从内部节点srchNode中获取该子表的聚合值并返回给用户,查询请求处理结束;否则,根据子表的时间范围将时间范围srchR划分为不同的时间段,对于其中覆盖到完整子表的时间段,直接从内部节点srchNode中获取被覆盖的子表的聚合值,对于其余的时间段,从内部节点srchNode中获取子表的地址后,访问相应的子表以获取该时间段内的原始数据,结合所获取到的原始数据和聚合值计算被查询的数据的聚合值,并返回给用户,查询请求处理结束。

[0035] 进一步地,步骤(S2)中,读取内部节点srchNode,包括:

[0036] 若时间分区树srchTPtree在内存中,则直接从内存中读取内部节点srchNode;

[0037] 若时间分区树srchTPtree在存储介质中,则从内存中读取时间分区树srchTPtree的必要信息,以获得存储介质中用于保存时间分区树srchTPtree的文件F;按照分层遍历的顺序确定内部节点srchNode在时间分区树srchTPtree的序号,以确定内部节点srchNode在文件F中的偏移量off,根据偏移量off访问存储介质中的文件F,以读取内部节点srchNode。

[0038] 按照本发明的第二方面,提供了一种面向多采样流的时间序列数据管理系统,包括处理器和计算机可读存储介质,计算机可读存储介质用于存储可执行程序;

[0039] 处理器用于读取计算机可读存储介质中存储的可执行程序,执行本发明第一方面提供的面向多采样流的时间序列数据管理方法。

[0040] 总体而言,通过本发明所构思的以上技术方案,能够取得以下有益效果:

[0041] (1) 本发明提供的面向多采样流的时间序列数据管理方法及系统,利用相关流单表将传感器同一时间采集到的多个数据点组织到一起,使得服务器可以一次处理多个数据流,也能够关联查询的性能;按照时间范围将传感器采集到的数据划分为不同的子表,利用全局的时间合并树合并所有传感器在相同时间范围内产生的子表,并通过一次写操作将这些子表写入底层存储介质中,有效减少了写I/O的数量,提升了写性能。总的来说,本发明能够提高多采样流时间序列数据的管理效率。

[0042] (2) 本发明提供的面向多采样流的时间序列数据管理方法及系统,对于一个时间分区弹性索引,仅在内存中保留一个时间分区树接收新的子表索引,其他时间分区树则被优化地存储在底层存储介质中,同时在内存中保存以写入底层存储介质的时间分区树的必要信息,在后续查询时,结合时间分区树的必要信息,通过一次I/O即可从存储介质中读取所需要的节点数据,因此,本发明能够减少查询延迟,提高查询效率。

[0043] (3) 本发明提供的面向多采样流的时间序列数据管理方法及系统,通过在时间分区树的内部节点记录其子节点索引的数据的聚合值,使得在后续查询中能够借助内部节点记录的聚合值加速查询,提高查询效率。

## 附图说明

[0044] 图1为本发明实施例提供的相关流单表结构与传统的数据组织结构对比示意图;其中,(a)为传统的数据组织结构示意图,(b)为相关流单表结构示意图;

[0045] 图2为本发明实施例提供的时间聚合树结构示意图;

[0046] 图3为本发明实施例提供的时间合并树结构示意图;

[0047] 图4为本发明实施例提供的面向多采样流的时间序列数据管理方法示意图;

[0048] 图5为本发明实施例提供的子表插入和存储过程的流程图;

[0049] 图6为本发明实施例提供的查询过程的流程图。

## 具体实施方式

[0050] 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。此外,下面所描述的本发明各个实施方式中所涉及到的技术特征只要彼此之间未构成冲突就可以相互组合。

[0051] 在详细解释本发明的技术方案之前,先结合附图对本发明中关键术语的定义作出如下解释:

[0052] 数据点(data point):表示成 $(T,V)$ ,意味着单流传感器在时间点 $T$ 采集到一个数据值 $V$ ;

[0053] 数据元组(data tuple):表示成 $(T,V_1,V_2,V_3\cdots)$ ,表示多流传感器在时间点 $T$ 采集到多个数据值 $V_1,V_2,V_3\cdots$ ;

[0054] 相关流单表(Single Table for CorrelatedStreams,STCS):用于组织一个传感

器采集到的多个时间序列数据流数据的单张表结构；

[0055] 在传统的时间序列数据管理方法中,对于同一个传感器采集到的多个时间序列数据流,会以数据点的形式对各数据流分别进行组织,如图1(a)所示;在发明中,对于同一个传感器采集到的多个时间序列数据流,会将其组织为元组后,使用一个STCS将各数据流组织到一起,如图1(b)所示;

[0056] 子表(tablet):是STCS的一个子部分,包含一个传感器在特定时间范围(例如 $2^t$ 纳秒)内采集的所有数据元组。

[0057] 时间分区弹性索引(Time-partitioned Elastic Index,TPEI):一种按照时间范围进行分区的索引结构,用于索引一个传感器产生的tablet;在一个可选的实施方式中,一个TPEI包含多个TPTree,其中一个最新的TPTree位于内存中用于索引对应的传感器新产生的数据,剩余的索引着之前数据的TPTree被优化地存储在存储介质中;

[0058] 时间分区树(Time-partitioned Tree,TPTree):一种按照时间范围进行分区的k叉树索引结构,其中k为任意整数,一般选 $2^m$ ,其中m为任意正整数,叶子节点索引不同时间范围的tablet,TPTree中所有的内部节点包含相同数量的数据;如图2所示,TPTree的内部节点还可记录其k个子节点的聚合值(包括它们的最大值,最小值,平均值等等),TPTree的叶子节点还可记录其索引的每一个子表的聚合值。

[0059] 时间合并树(Time-Merged Tree,TMTree):一种树结构,用来合并不同传感器相同时间范围的tablet,以便于将这些tablet通过一次写IO写入底层存储介质,减少IO数量,提升写性能;

[0060] 在一个可选的实施方式中,如图3所示,TMTree的内部节点记录的信息包括:所覆盖的子表的时间范围和子节点的地址;

[0061] TMTree的叶子节点记录的信息包括:(1)被索引的子表的时间范围;(2)被索引的子表的内存地址,具体可以是一个长度为n的索引数组,n为传感器总数,索引数组中,每一个元素表示一个传感器在该时间范围内产生的子表在内存中的地址;(3)用于标识各传感器在该时间范围内的子表是否已被索引的位图信息,具体可为一个长度为n的bitmap数组,bitmap数组中每一个元素表示一个传感器在改时间段内产生的子表是否已经被索引;(4)用于记录该叶子节点持续未接收到新数据的时间的计时器timer,叶子节点中每次索引到新的子表,其计时器会被重置为0。

[0062] 基于上述数据结构,本发明所提供的面向多采样流的时间序列数据管理方法,如图4~图5所示,包括:

[0063] 将传感器在同一时间采集到的多个数据点组织为一个数据元组后,按时间顺序插入到该传感器对应的相关流单表中;

[0064] 当相关流单表中的数据达到第一阈值后,按照时间范围将其中的数据划分为不同的子表,并将子表的索引信息插入到对应的时间分区弹性索引中;

[0065] 将所有传感器的子表的地址插入到全局的时间合并树中,以使得时间合并树的叶子节点索引所有传感器在相同时间范围内产生的子表;

[0066] 定期检查时间合并树最左端的叶子节点,若该叶子节点已索引到所有传感器的子表,或者该叶子节点持续未接收到新数据的时间超过了第二阈值,则将该叶子节点索引的所有子表通过一次写操作写入存储介质中,并回收该叶子节点。由于在TMTree的同一层节



点中,被索引子表的采集时间从左到右依次增大,每次写入时间合并树最左端叶子节点索引的子表,能够保证时间序列数据在底层存储介质中按时间顺序存储。

[0067] 在一个可选的实施方式中,上述面向多采样流的时间序列数据管理方法中,将时间合并树最左端的叶子节点索引的所有子表通过一次写操作写入存储介质中,包括:

[0068] 分配一块空闲的内存区域,根据叶子节点leaf记录的内存地址信息获取到叶子节点leaf索引的所有子表,进行压缩并顺序地写入所分配的内存区域;通过一次写操作,将该内存区域中的数据写入存储介质;

[0069] 其中,叶子节点leaf为时间合并树最左端的叶子节点。

[0070] 在本实施例中,所提供的面向多采样流的时间序列数据管理方法,还可包括:定期获取内存中的时间分区树,记为时间分区树TPtree;

[0071] 按照分层遍历的顺序将时间分区树TPtree的所有内部节点数据序列化后,写入存储介质中,并在内存中记录时间分区树TPtree的必要信息;

[0072] 回收时间分区树TPtree所占的内存空间后,为时间分区树TPtree所属的时间分区弹性索引建立一个新的时间分区树,用于索引对应传感器新产生的子表;

[0073] 其中,时间分区树的必要信息包括索引数据的时间范围、树的层数、每一层索引数据的开始时间以及存储介质中用于保存该时间分区树的文件名;

[0074] 对于被优化存储在底层存储介质中的TPTree,因为内存中保存了该TPTree的时间范围和层数,同时叶子节点的时间范围为 $2^t$ 纳秒,所以每层节点的时间范围都可以计算得知,例如,见图2上半部分内存TPTree,其中ID=3, ID=4和ID=5的节点的时间范围可以通过叶子节点的时间范围与它索引的叶子节点个数计算出来,即为 $2^t \times k$ 纳秒,其中k为分支数,在图2中为2,继续向上层节点可以以此类推;同时,内存中记录着该TPTree每层节点的开始时间,所以系统可以计算出每个节点的开始时间和结束时间,例如,图2中内存TPTree的第3层的开始时间为st,那么ID=3, ID=4和ID=5节点的开始时间分别是st,  $st+2^t \times k$ ,  $st+2^t \times k \times 2$ ;当查询某个时间范围时,服务器可以根据之前在内存中保存的TPTree的信息计算出该TPTree中所有内部节点的时间范围,从而确定被需要的内部节点在该TPTree的哪一层,以及在该层中具体的偏移位置;此外,TPTree中所有的内部节点包含相同数量的数据,在存储时,系统并没有对内部节点数据进行压缩,只是进行序列化,并没有改变节点的大小,而且被从上到下从左到右顺序地保存在底层存储介质的一个文件中;所以服务器可以根据节点序列化之后的数据大小,以及被需要的节点在TPTree中的偏移,得出该内部节点在文件中的偏移,然后通过内存中保存的文件的路径,以及文件的偏移读出该节点序列化后的数据,最后通过反序列化操作,获得被需要的内部节点的具体信息。因此,相比于传统的层次遍历的方法,该方法可以通过使用计算来减少对底层存储介质中的TPTree进行检索时产生的读I0数量,只需要一次读I0即可得到需要的节点,提升了查询性能。

[0075] 可选地,在上述面向多采样流的时间序列数据管理方法中,将子表(tablet)的索引信息插入到对应的时间分区弹性索引中,具体包括:

[0076] (A1) 服务器根据该tablet所属的传感器ID,找到对应的TPEI,并获得该TPEI中位于内存中的TPTree,以找到该tablet应该被插入的TPTree的根节点;

[0077] (A2) 服务器根据tablet包含的数据的时间范围和TPTree根节点的时间范围确定该tablet是应该插入该TPTree的哪棵子树;

- [0078] (A3) 在对应的子树上继续根据时间范围判断,直达到达叶子节点;
- [0079] (A4) 将tablet的地址插入到该叶子节点中;
- [0080] (A5) 计算tablet的聚合值(最大值,最小值,平均值等等),将此聚合值保存到索引该tablet的叶子节点中;
- [0081] (A6) 从这个叶子节点逐层向上更新上层节点的聚合值,直到TPTree的根节点结束。
- [0082] 可选地,在上述面向多采样流的时间序列数据管理方法中,将任意一个子表的地址插入到全局的时间合并树中,具体包括:
- [0083] (B1) 服务器根据tablet包含的数据的时间范围和TMTree根节点的时间范围确定该tablet是应该插入该TMTree的哪棵子树;
- [0084] (B2) 在对应的子树上继续根据时间范围判断,直达到达叶子节点;
- [0085] (B3) 将tablet插入到该叶子节点中;该步骤具体包括:
- [0086] (B31) 服务器通过哈希该tablet所述的传感器的标识符(即标识符ID),生成一个独一无二的值 $h$  ( $0 \leq h \leq n$ ),其中 $n$ 表示传感器的总数;
- [0087] (B32) 服务器根据哈希值 $h$ ,确定该tablet具体被插入到的位置,然后更新该叶子节点的索引数组,将索引数组在 $h$ 位置的值改成tablet在内存中的地址;
- [0088] (B33) 服务器使用哈希值 $h$ ,将叶子节点中的bitmap数组 $h$ 位置的值更新为1,表明相应的传感器在该时间范围的tablet已经被索引到TMTree中;
- [0089] (B34) 服务器重设该叶子节点中的计时器(timer),让计时器从零开始计数;
- [0090] 相应地,对TMTree最左端叶子节点进行判断时,具体可通过检查该叶子节点中的bitmap数组来判断该叶子节点是否已索引到所有传感器的子表,若bitmap数组中的所有元素均为1,则表明该叶子节点已经索引到所有传感器的子表;具体可通过判断该叶子节点的计时器timer来判断该叶子节点持续未接收到新数据的时间是否超过了第二阈值,若timer超过了第二阈值则说明该叶子节点持续未接收到新数据的时间超过了第二阈值。
- [0091] 在本实施例中,如图6所示,进行数据查询的过程包括:
- [0092] 用户向服务器发送查询请求,查询请求包括被查询的传感器ID、被查询数据的时间范围srchR以及结果类型;服务器接收到查询请求后,按照如下步骤处理查询请求:
- [0093] (S1) 根据传感器ID确定被查询的时间分区弹性索引TPEI,并根据属于时间分区弹性索引TPEI的各时间分区树的时间范围和时间范围srchR,确定需要被查询的时间分区树srchTPtree;
- [0094] (S2) 根据时间范围srchR确定时间分区树srchTPtree中需要被查询的内部节点srchNode,并读取内部节点srchNode;
- [0095] (S3) 若结果类型为原始数据,则根据时间范围srchR和内部节点srchNode获得被查询的子表的地址后,访问相应的子表以获取被查询的原始数据并返回给用户,查询请求处理结束;若结果类型为聚合值,则转入步骤(S4);
- [0096] (S4) 若时间范围srchR与任意一个子表的时间范围重合,则直接从内部节点srchNode中获取该子表的聚合值并返回给用户,查询请求处理结束;否则,根据子表的时间范围将时间范围srchR划分为不同的时间段,对于其中覆盖到完整子表的时间段,直接从内部节点srchNode中获取被覆盖的子表的聚合值,对于其余的时间段,从内部节点srchNode

中获取子表的地址后,访问相应的子表以获取该时间段内的原始数据,结合所获取到的原始数据和聚合值计算被查询的数据的聚合值,并返回给用户,查询请求处理结束;

[0097] 通常情况下,根据子表的时间范围对被查询数据的时间范围srchR进行划分,可以得到三个连续的时间段,中间的时间段等于一个或多个节点(包括内部节点以及叶子节点)的时间范围的总和,此时间段的聚合值可以通过获取这些节点父节点中记录的聚合值得到;剩余的前后两个时间段都小于一个tablet的时间范围(如果大于 $2^t$ 纳秒,可以直接使用叶子节点父节点中的聚合值),因此它们对应的聚合值必须要读取对应的tablet的原始数据,具体操作过程可参考上述步骤(S3),再根据原始的数据求出前后两个时间段的聚合值;最终服务器使用这三个时间段的聚合值求出最终整个范围的聚合值,返回给用户;

[0098] 在查询过程中,步骤(S2)中,读取内部节点srchNode,包括:

[0099] 若时间分区树srchTPtree在内存中,则直接从内存中读取内部节点srchNode;

[0100] 若时间分区树srchTPtree在存储介质中,则从内存中读取时间分区树srchTPtree的必要信息,以获得存储介质中用于保存时间分区树srchTPtree的文件F;按照分层遍历的顺序确定内部节点srchNode在时间分区树srchTPtree的序号,以确定内部节点srchNode在文件F中的偏移量off,根据偏移量off访问存储介质中的文件F,以读取内部节点srchNode。

[0101] 在实际使用时,第一阈值、第二阈值、将TPTree写入底层介质的时间间隔以及检查TMTree最左端叶子节点的时间间隔,均可根据实际的存储系统的容量、性能以及时间序列数据的产生特性相应确定。

[0102] 本发明还提供了一种面向多采样流的时间序列数据管理系统,包括处理器和计算机可读存储介质,计算机可读存储介质用于存储可执行程序;

[0103] 处理器用于读取计算机可读存储介质中存储的可执行程序,执行上述面向多采样流的时间序列数据管理方法。

[0104] 总的来说,上述面向多采样流的时间序列数据管理方法,使用更适合多时间序列数据流的表结构组织多采样流的时间序列数据,使得服务器可以一次处理多个数据流,此外多个数据流被组织在同一个表中,提升了关联查询的性能;使用弹性索引结构TPEI索引子表,内存中只需要保存少量的索引数据(一棵TPTree)来索引新的数据,而且其他的索引数据被优化地存储在底层存储介质中,降低了查询时的读IO数量,降低了查询延迟;使用TMTree合并多个传感器的tablet,一次写入底层存储介质,减少了写IO的数量,提升了写性能。

[0105] 本领域的技术人员容易理解,以上所述仅为本发明的较佳实施例而已,并不用以限制本发明,凡在本发明的精神和原则之内所作的任何修改、等同替换和改进等,均应包含在本发明的保护范围之内。

时间戳	A <sub>CPU</sub>	时间戳	A <sub>MEM</sub>	时间戳	A <sub>DISK</sub>
T1	85.3	T1	3.32	T1	0.53
T2	101.4	T2	4.47	T2	0.70
T3	97.7	T3	4.32	T3	0.64
T4	87.4	T4	3.15	T4	0.50
T5	86.2	T5	3.14	T5	0.50

(a)

时间戳	A <sub>CPU</sub>	A <sub>MEM</sub>	A <sub>DISK</sub>
T1	85.3	3.32	0.53
T2	101.4	4.47	0.70
T3	97.7	4.32	0.64
T3	87.4	3.15	0.50
T3	86.2	3.14	0.50

(b)

图1

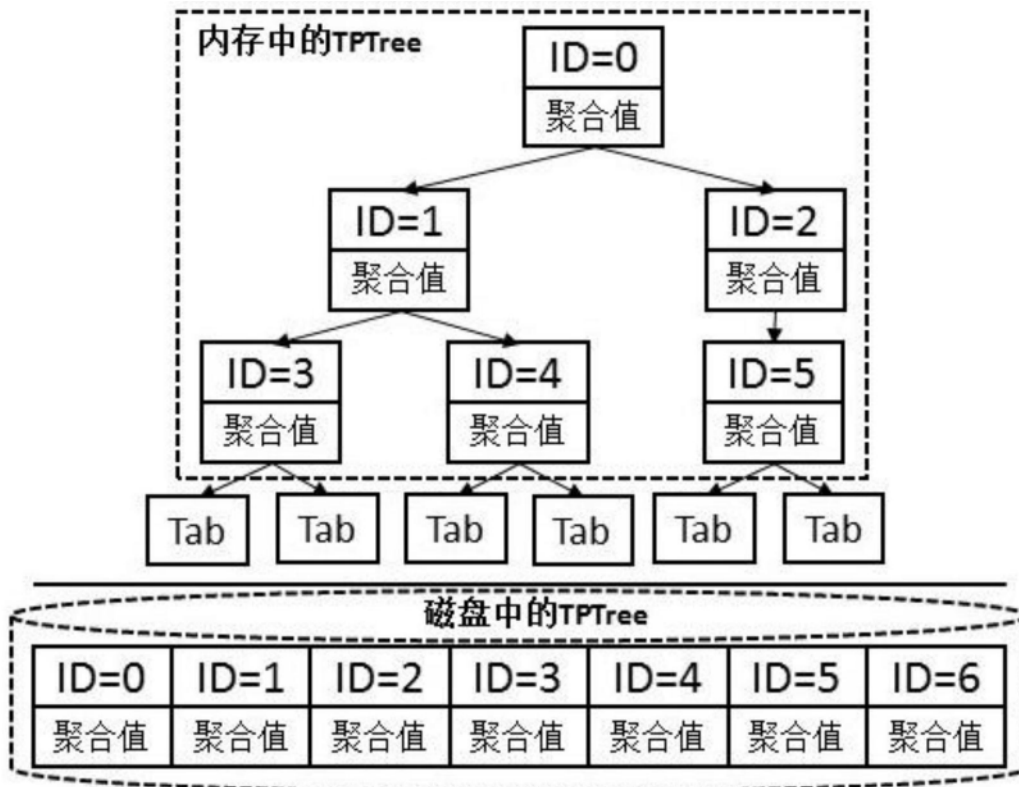


图2

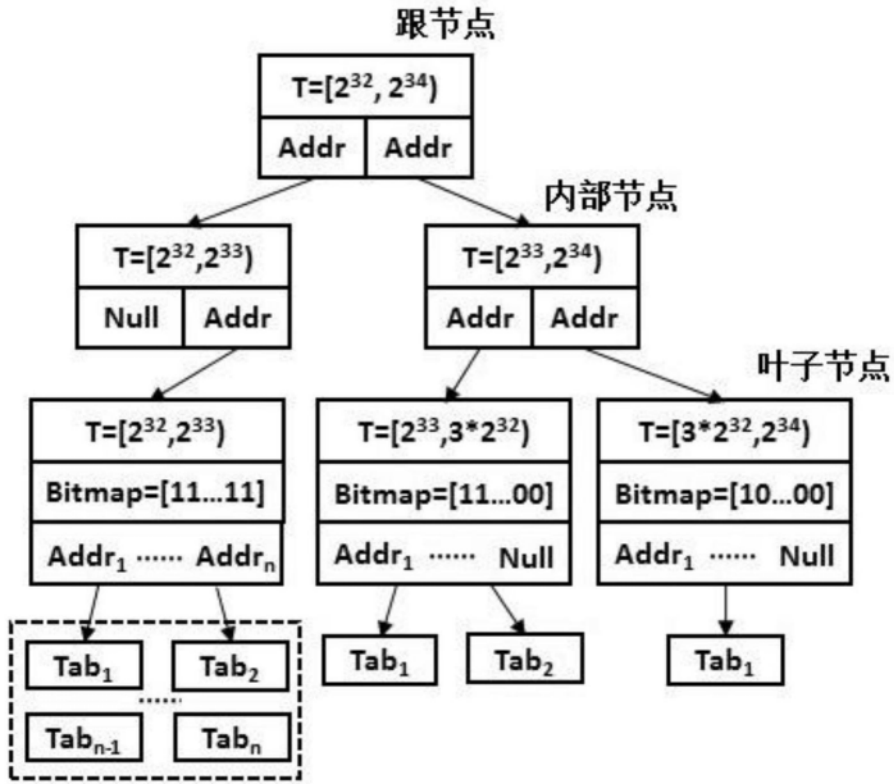


图3

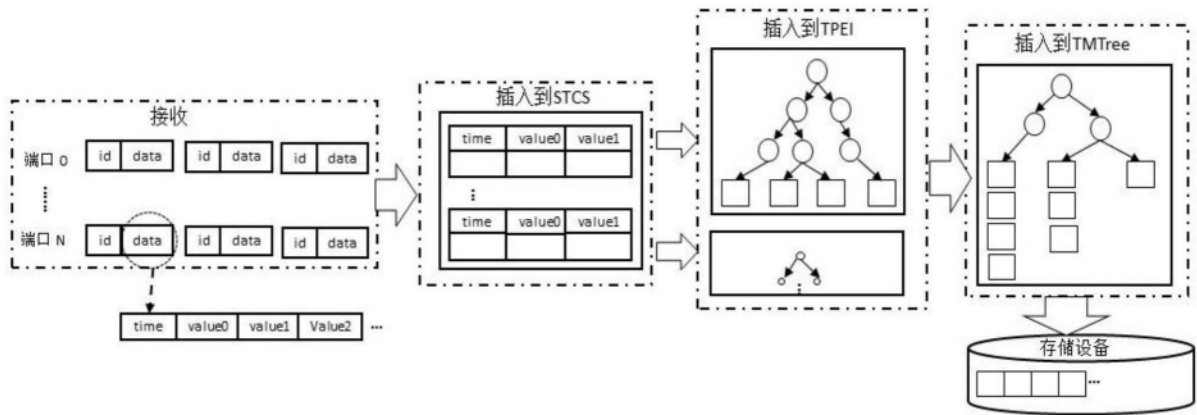


图4

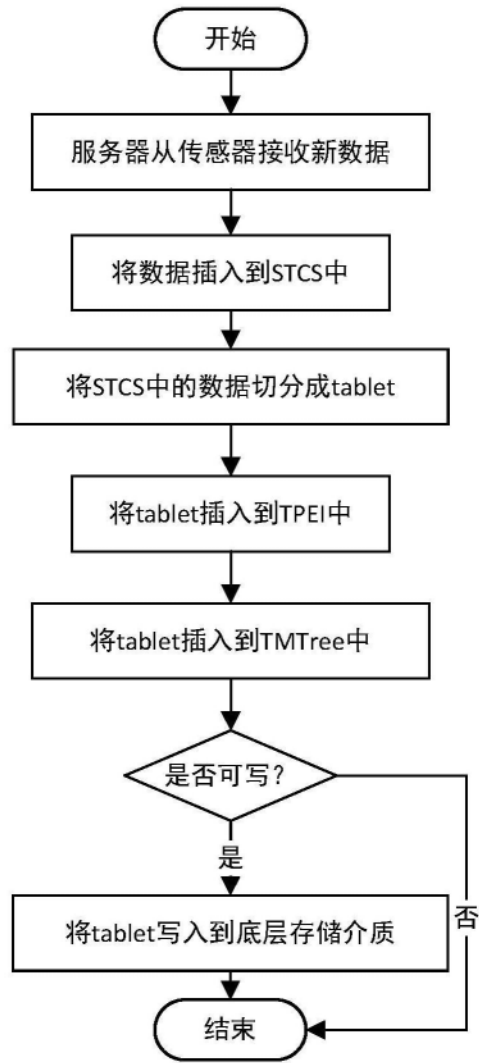


图5

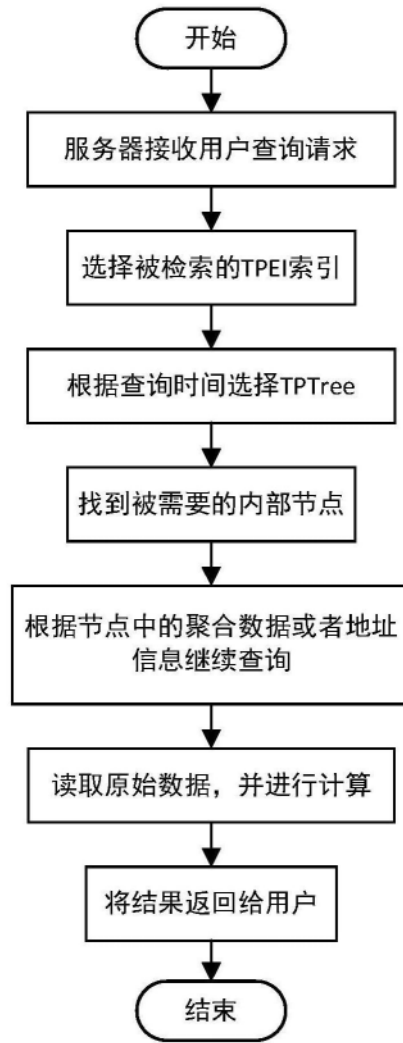


图6