



(12)发明专利申请

(10)申请公布号 CN 107451203 A

(43)申请公布日 2017. 12. 08

(21)申请号 201710552947.7

(22)申请日 2017.07.07

(71)申请人 阿里巴巴集团控股有限公司

地址 英属开曼群岛大开曼资本大厦一座四
层847号邮箱

(72)发明人 陈萌萌 施宇迪 肖意

(74)专利代理机构 北京博思佳知识产权代理有
限公司 11415

代理人 林祥 李威

(51)Int.Cl.

G06F 17/30(2006.01)

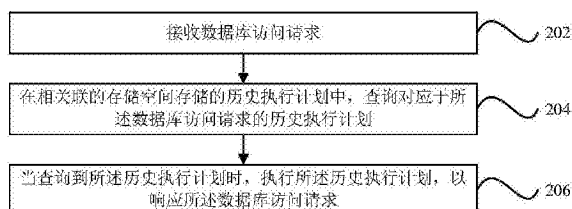
权利要求书3页 说明书13页 附图4页

(54)发明名称

数据库访问方法及装置

(57)摘要

本说明书一个或多个实施例提供一种数据库访问方法及装置,该方法可以包括:接收数据库访问请求;在相关联的存储空间存储的历史执行计划中,查询对应于所述数据库访问请求的历史执行计划;当查询到所述历史执行计划时,执行所述历史执行计划,以响应所述数据库访问请求。



1. 一种数据库访问方法,包括:

接收数据库访问请求;

在相关联的存储空间存储的历史执行计划中,查询对应于所述数据库访问请求的历史执行计划;

当查询到所述历史执行计划时,执行所述历史执行计划,以响应所述数据库访问请求。

2. 根据权利要求1所述的方法,所述相关联的存储空间包括:相关联的缓存空间。

3. 根据权利要求1所述的方法,所述相关联的存储空间包括以下至少之一:本地存储空间、相关联的分布式节点处的存储空间。

4. 根据权利要求1所述的方法,还包括:

当未查询到所述历史执行计划时,生成所述数据库访问请求对应的解释型执行计划;

执行所述解释型执行计划,以响应所述数据库访问请求。

5. 根据权利要求4所述的方法,所述历史执行计划或所述解释型执行计划被按照下述方式执行:

当所述历史执行计划或所述解释型执行计划为本地计划时,在本地执行所述历史执行计划或所述解释型执行计划;

当所述历史执行计划或所述解释型执行计划为分布式计划时,在本地执行所述历史执行计划或所述解释型执行计划中对应于本地节点的本地子计划;其中,所述历史执行计划或所述解释型执行计划中还包含对应于相关联的分布式节点的关联子计划,所述关联子计划由所述相关联的分布式节点执行。

6. 根据权利要求4所述的方法,还包括:

将所述解释型执行计划存储至所述相关联的存储空间,以作为新的历史执行计划。

7. 根据权利要求1所述的方法,还包括:

获取所述相关联的存储空间中的解释型历史执行计划;

根据所述解释型历史执行计划对应的数据库访问请求,生成相应的编译型历史执行计划,以将所述解释型历史执行计划替换为编译型历史执行计划。

8. 根据权利要求7所述的方法,所述获取所述相关联的存储空间中的解释型历史执行计划,包括:

从所述相关联的存储空间中对应于本地节点的本地存储空间,获取所述解释型历史执行计划。

9. 根据权利要求7所述的方法,所述获取所述相关联的存储空间中的解释型历史执行计划,包括:

确定所述相关联的存储空间中的解释型历史执行计划的执行次数和/或执行频率;

优先获取所述相关联的存储空间中执行次数和/或执行频率更高的解释型历史执行计划。

10. 根据权利要求7所述的方法,所述生成相应的编译型历史执行计划,包括:

获取所述解释型历史执行计划对应的数据库访问请求包含的待编译函数;

当所述相关联的存储空间中存储有所述待编译函数对应的历史编译内容时,引用所述历史编译内容,以生成所述编译型历史执行计划。

11. 根据权利要求10所述的方法,所述生成相应的编译型历史执行计划,包括:

当所述相关联的存储空间中未存储有所述待编译函数对应的历史编译内容时,生成所述待编译函数对应的编译内容,以生成所述编译型历史执行计划。

12. 根据权利要求11所述的方法,还包括:

将所述待编译函数对应的编译内容存储至所述相关联的存储空间,以作为新的历史编译内容。

13. 根据权利要求10所述的方法,还包括:

确定生成所述编译型历史执行计划的过程对所述相关联的存储空间中的历史编译内容的依赖程度;

从所述相关联的存储空间中删除依赖程度低于预设依赖程度的历史编译内容。

14. 根据权利要求13所述的方法,所述依赖程度正相关于所述历史编译内容被引用的次数。

15. 根据权利要求13所述的方法,所述相关联的存储空间中存储有与所述历史编译内容相关的历史执行计划,其中所述依赖程度负相关于被从所述相关联的存储空间中淘汰且与所述历史编译内容相关的历史执行计划的数量。

16. 根据权利要求15所述的方法,执行次数和/或执行频率更低的历史执行计划被优先从所述相关联的存储空间中淘汰。

17. 一种数据库访问装置,包括:

接收单元,接收数据库访问请求;

查询单元,在相关联的存储空间存储的历史执行计划中,查询对应于所述数据库访问请求的历史执行计划;

执行单元,当查询到所述历史执行计划时,执行所述历史执行计划,以响应所述数据库访问请求。

18. 根据权利要求17所述的装置,所述相关联的存储空间包括:相关联的缓存空间。

19. 根据权利要求17所述的装置,所述相关联的存储空间包括以下至少之一:本地存储空间、相关联的分布式节点处的存储空间。

20. 根据权利要求17所述的装置,还包括:

解释单元,当未查询到所述历史执行计划时,生成所述数据库访问请求对应的解释型执行计划;

所述执行单元具体用于执行所述解释型执行计划,以响应所述数据库访问请求。

21. 根据权利要求20所述的装置,所述历史执行计划或所述解释型执行计划被所述执行单元按照下述方式执行:

当所述历史执行计划或所述解释型执行计划为本地计划时,在本地执行所述历史执行计划或所述解释型执行计划;

当所述历史执行计划或所述解释型执行计划为分布式计划时,在本地执行所述历史执行计划或所述解释型执行计划中对应于本地节点的本地子计划;其中,所述历史执行计划或所述解释型执行计划中还包含对应于相关联的分布式节点的关联子计划,所述关联子计划由所述相关联的分布式节点执行。

22. 根据权利要求20所述的装置,还包括:

第一存储单元,将所述解释型执行计划存储至所述相关联的存储空间,以作为新的历

史执行计划。

23. 根据权利要求17所述的装置,还包括:

获取单元,获取所述相关联的存储空间中的解释型历史执行计划;

编译单元,根据所述解释型历史执行计划对应的数据库访问请求,生成相应的编译型历史执行计划,以将所述解释型历史执行计划替换为编译型历史执行计划。

24. 根据权利要求23所述的装置,所述获取单元具体用于:

从所述相关联的存储空间中对应于本地节点的本地存储空间,获取所述解释型历史执行计划。

25. 根据权利要求23所述的装置,所述获取单元具体用于:

确定所述相关联的存储空间中的解释型历史执行计划的执行次数和/或执行频率;

优先获取所述相关联的存储空间中执行次数和/或执行频率更高的解释型历史执行计划。

26. 根据权利要求23所述的装置,所述编译单元具体用于:

获取所述解释型历史执行计划对应的数据库访问请求包含的待编译函数;

当所述相关联的存储空间中存储有所述待编译函数对应的历史编译内容时,引用所述历史编译内容,以生成所述编译型历史执行计划。

27. 根据权利要求26所述的装置,所述编译单元具体用于:

当所述相关联的存储空间中未存储有所述待编译函数对应的历史编译内容时,生成所述待编译函数对应的编译内容,以生成所述编译型历史执行计划。

28. 根据权利要求27所述的装置,还包括:

第二存储单元,将所述待编译函数对应的编译内容存储至所述相关联的存储空间,以作为新的历史编译内容。

29. 根据权利要求26所述的装置,还包括:

确定单元,确定生成所述编译型历史执行计划的过程对所述相关联的存储空间中的历史编译内容的依赖程度;

删除单元,从所述相关联的存储空间中删除依赖程度低于预设依赖程度的历史编译内容。

30. 根据权利要求29所述的装置,所述依赖程度正相关于所述历史编译内容被引用的次数。

31. 根据权利要求29所述的装置,所述相关联的存储空间中存储有与所述历史编译内容相关的历史执行计划,其中所述依赖程度负相关于被从所述相关联的存储空间中淘汰且与所述历史编译内容相关的历史执行计划的数量。

32. 根据权利要求31所述的装置,执行次数和/或执行频率更低的历史执行计划被优先从所述相关联的存储空间中淘汰。

33. 一种数据库访问装置,包括:

处理器;

用于存储处理器可执行指令的存储器;

其中,所述处理器被配置为实现如权利要求1-16中任一项所述的方法。

数据库访问方法及装置

技术领域

[0001] 本说明书一个或多个实施例涉及数据库技术领域,尤其涉及一种数据库访问方法及装置。

背景技术

[0002] 在相关技术中,用户可以通过诸如SQL(Structured Query Language,结构化查询语言)等语言生成数据库访问请求,以发起对数据库的访问。SQL语言是一种描述型语言(非过程型语言),用户在基于SQL语言发起数据库访问请求时,需要通过SQL语言描述其对数据库的数据访问需求,比如查询哪张表、什么样的数据、对数据做哪些操作等。而对于数据库内核而言,在接收到用户发出的数据库访问请求时,需要将数据库访问请求生成为可执行的代码,这部分代码通常称为“执行计划”(或者“查询计划”,也可简称为“计划”)。

[0003] 执行计划从执行和优化方式上大体可分为“解释型”(interpreted)和“编译型”(compiled)两种。对于解释型的执行计划,通过将某些常用操作“函数化”,以及生成一些执行的“模板”,并对基础函数或模板的拼装、组合,即可形成解释型的执行计划。由于解释型执行计划可以通过各种模板或函数的排列组合可以组成任意复杂的执行计划,因而具有使用灵活、易于实现等优势。

[0004] 与解释型执行计划的生成过程不同,编译型执行计划的生成思路是将数据库访问请求直接翻译成代码,然后利用“即时编译(JIT, just-in-time compilation)”技术将代码动态编译为可以执行的机器代码并动态加载到运行环境中,因而编译型执行计划往往执行效率更高,相比解释型执行计划可能有多达几个数量级的性能提升。

[0005] 然而,虽然编译型执行计划的执行效率更高,但是编译型执行计划的生成过程需要调用编译器的即时编译模块,使得编译型执行计划的生成过程相比于解释型执行计划而言更加耗时,表现为数据库的反馈延迟较大。

发明内容

[0006] 有鉴于此,本说明书一个或多个实施例提供一种数据库访问方法及装置,可以减少数据库访问所需占用的处理资源,提升对执行计划的生成效率,并有助于加快数据库访问。

[0007] 为实现上述目的,提供技术方案如下:

[0008] 根据第一方面,提出了一种数据库访问方法,包括:

[0009] 接收数据库访问请求;

[0010] 在相关联的存储空间存储的历史执行计划中,查询对应于所述数据库访问请求的历史执行计划;

[0011] 当查询到所述历史执行计划时,执行所述历史执行计划,以响应所述数据库访问请求。

[0012] 根据第二方面,提出了一种数据库访问装置,包括:

- [0013] 接收单元,接收数据库访问请求;
- [0014] 查询单元,在相关联的存储空间存储的历史执行计划中,查询对应于所述数据库访问请求的历史执行计划;
- [0015] 执行单元,当查询到所述历史执行计划时,执行所述历史执行计划,以响应所述数据库访问请求。
- [0016] 根据第三方面,提出了一种数据库访问装置,包括:
- [0017] 处理器;
- [0018] 用于存储处理器可执行指令的存储器;
- [0019] 其中,所述处理器被配置为实现如第一方面所述的方法。
- [0020] 由以上技术方案可见,本说明书一个或多个实施例通过存储历史执行计划,并将历史执行计划应用于对数据库访问请求的响应,可以实现对历史执行计划的复用,而无需重复生成相同的执行计划,从而能够减少数据库访问所需占用的处理资源,提升对执行计划的生成效率,并有助于加快数据库访问。

附图说明

- [0021] 图1是一示例性实施例提供的一种数据库访问系统的架构示意图。
- [0022] 图2是一示例性实施例提供的一种数据库访问方法的流程图。
- [0023] 图3是一示例性实施例的一种对数据库访问请求进行响应的原理示意图。
- [0024] 图4是一示例性实施例的另一种对数据库访问请求进行响应的原理示意图。
- [0025] 图5是一示例性实施例提供的一种电子设备的结构示意图。
- [0026] 图6是一示例性实施例提供的一种数据库访问装置的框图。

具体实施方式

- [0027] 本说明书的一个或多个实施例通过对历史执行计划的合理复用,可以避免重复生成相同的执行计划,从而能够减少数据库访问所需占用的处理资源,提升对执行计划的生成效率,并有助于加快数据库访问。为做进一步说明,提供下列实施例:
- [0028] 图1是一示例性实施例提供的一种数据库访问系统的架构示意图。如图1所示,该系统可以包括服务器11、网络12、若干电子设备,比如手机13、PC14等。
- [0029] 服务器11可以为包含一独立主机的物理服务器,或者该服务器11可以为主机集群承载的虚拟服务器,或者该服务器11可以为云服务器。在运行过程中,服务器11可以承载数据库。
- [0030] 手机13、PC14只是用户可以使用的部分类型的电子设备。实际上,用户显然还可以使用诸如下述类型的电子设备:平板设备、笔记本电脑、掌上电脑(PDAs, Personal Digital Assistants)、可穿戴设备(如智能眼镜、智能手表等)等,本说明书一个或多个实施例并不对此进行限制。在运行过程中,该电子设备可以运行某一应用的客户端侧的程序,以实现该应用的相关业务功能,比如向服务器11发起数据库访问请求,以实现数据库访问操作。
- [0031] 而对于手机13、PC14与服务器11之间进行交互的网络12,可以包括多种类型的有线或无线网络。在一实施例中,该网络12可以包括公共交换电话网络(Public Switched Telephone Network, PSTN)和因特网。

[0032] 下面结合实施例描述服务器11在本说明书的一个或多个实施例中,如何实现高效的数据库访问方案。

[0033] 图2是一示例性实施例提供的一种数据库访问方法的流程图。如图2所示,该方法可以应用于诸如上述图1所示的服务器11;该方法可以包括以下步骤:

[0034] 步骤202,接收数据库访问请求。

[0035] 在一实施例中,服务器11可以采用分布式架构,该数据库访问请求可由该分布式架构中的任一分布式节点接收。

[0036] 在一实施例中,服务器11可以采用非分布式架构,由该服务器11接收该数据库访问请求。

[0037] 步骤204,在相关联的存储空间存储的历史执行计划中,查询对应于所述数据库访问请求的历史执行计划。

[0038] 在一实施例中,所述相关联的存储空间可以包括:相关联的缓存空间。

[0039] 在一实施例中,所述相关联的存储空间可以包括以下至少之一:本地存储空间、相关联的分布式节点处的存储空间。例如,当服务器11采用分布式架构时,对于接收到数据库访问请求的分布式节点而言,相关联的存储空间可以包括该分布式节点的本地存储空间,也可以包括其他分布式节点处的存储空间。

[0040] 步骤206,当查询到所述历史执行计划时,执行所述历史执行计划,以响应所述数据库访问请求。

[0041] 在一实施例中,通过查询并执行历史执行计划,可以省去将数据库访问请求生成执行计划的过程,有助于提升对数据库访问请求的响应效率。

[0042] 在一实施例中,当未查询到所述历史执行计划时,可以生成所述数据库访问请求对应的解释型执行计划,并执行所述解释型执行计划,以响应所述数据库访问请求。由于解释型执行计划相比于编译型执行计划而言耗时较短、延迟较小,因而有助于提升对数据库访问请求的响应效率。在其他实施例中,当未查询到所述历史执行计划时,可以生成所述数据库访问请求对应的编译型执行计划,并执行所述编译型执行计划,以响应所述数据库访问请求。

[0043] 在一实施例中,当所述历史执行计划或所述解释型执行计划为本地计划时,可以在本地执行所述历史执行计划或所述解释型执行计划。当服务器11采用分布式架构时,此处的“本地”可以理解为接收到数据库访问请求的分布式节点;当服务器11采用非分布式架构时,此处的“本地”可以理解为该服务器11本身。

[0044] 在一实施例中,当所述历史执行计划或所述解释型执行计划为分布式计划时,可以在本地(即接收到数据库访问请求的分布式节点)执行所述历史执行计划或所述解释型执行计划中对应于本地节点的本地子计划;其中,所述历史执行计划或所述解释型执行计划中还包含对应于相关联的分布式节点的关联子计划,所述关联子计划由所述相关联的分布式节点执行。

[0045] 在一实施例中,可以将所述解释型执行计划存储至所述相关联的存储空间,以作为新的历史执行计划,从而实现对“相关联的存储空间”的不断更新,使得后续接收到数据库访问请求时,有助于提升查找到相应的历史执行计划的概率。

[0046] 在一实施例中,可以获取所述相关联的存储空间中的解释型历史执行计划;然后,

根据所述解释型历史执行计划对应的数据库访问请求,生成相应的编译型历史执行计划,以将所述解释型历史执行计划替换为编译型历史执行计划。由于编译型执行计划的执行效率高于解释型执行计划,因而通过生成解释型历史执行计划对应的编译型执行计划,使得后续查找数据库访问请求对应的历史执行计划时,通过执行查找到的编译型执行计划,有助于提升对数据库访问请求的响应效率。在接收到数据库访问请求时,通过首先生成相应的解释型执行计划,可以对该数据库访问请求实现快速响应;而在完成对数据库访问请求的响应后,将数据库访问请求生成编译型执行计划的过程不再需要实现上述的快速响应,使得该生成过程具有更大的灵活性。

[0047] 在一实施例中,可以将解释型历史执行计划原子性地替换为相应的编译型历史执行计划。

[0048] 在一实施例中,将解释型历史执行计划替换为相应的编译型历史执行计划的过程,对于用户而言可以是透明的,以降低对用户的干扰和影响。

[0049] 在一实施例中,可以从所述相关联的存储空间中对应于本地节点的本地存储空间,获取所述解释型历史执行计划。换言之,每个分布式节点可以仅针对本地的解释型执行计划,生成相应的编译型执行计划,从而在各个分布式节点之间实现对解释型执行计划的异步优化(由于编译型执行计划的执行效率高于解释型执行计划,因而可以将生成编译型执行计划、以替代解释型执行计划的过程,称为优化过程)、无需同步执行,可以简化编译型执行计划的生成过程。

[0050] 在一实施例中,可以确定所述相关联的存储空间中的解释型历史执行计划的执行次数和/或执行频率,并优先获取所述相关联的存储空间中执行次数和/或执行频率更高的解释型历史执行计划,那么对于多个分布式节点之间执行异步优化的情况下,可以确保执行次数和/或执行频率更高的解释型历史执行计划在各个分布式节点上,都能够更早被优化为编译型历史执行计划,使得对应于同一数据库访问请求的执行计划在各个分布式节点上的执行计划类型(解释型执行计划或编译型执行计划)尽可能地保持一致。对于分布式架构而言,对于接收到的数据库访问请求,需要各个分布式节点都执行完本地的执行计划后,才认为完成对该数据库访问请求的响应,因而需要根据最后一个执行完本地的执行计划的分布式节点来确定整体的总体计划响应时长;可见,如果各个分布式节点上的优化节奏不一致,比如对于同一数据库访问请求而言,一部分分布式节点上采用优化后的编译型历史执行计划、而另一部分分布式节点上采用尚未优化的解释型历史执行计划,那么即便所有的编译型历史执行计划都被相应的分布式节点执行完毕,但是由于解释型历史执行计划的执行效率较低,导致总体计划响应时长并未能够得以缩短。而如果对于同一数据库访问请求而言,所有分布式节点上均被优化为编译型执行计划,那么在后续接收到该数据库访问请求时,可以确保所有分布式节点对本地的编译型执行计划的执行时长均会缩短,从而使得总体计划响应时长得得以缩短、提升了对数据库访问请求的响应效率。

[0051] 在一实施例中,可以获取所述解释型历史执行计划对应的数据库访问请求包含的待编译函数;当所述相关联的存储空间中存储有所述待编译函数对应的历史编译内容时,引用所述历史编译内容,以生成所述编译型历史执行计划。通过对历史编译内容的引用,可以避免对函数的重复编译,有助于优化生成编译型历史执行计划时对内存的占用问题。

[0052] 在一实施例中,当所述相关联的存储空间中未存储有所述待编译函数对应的历史

编译内容时,可以生成所述待编译函数对应的编译内容,以生成所述编译型历史执行计划。

[0053] 在一实施例中,可以将所述待编译函数对应的编译内容存储至所述相关联的存储空间,以作为新的历史编译内容,从而实现对“历史编译内容”的不断更新。

[0054] 在一实施例中,可以确定生成所述编译型历史执行计划的过程对所述相关联的存储空间中的历史编译内容的依赖程度,并从所述相关联的存储空间中删除依赖程度低于预设依赖程度的历史编译内容,从而通过对历史编译内容的不断筛选,在尽可能降低对编译型历史执行计划的生成过程的影响的情况下,有助于优化历史编译内容的内存占用。

[0055] 在一实施例中,所述依赖程度正相关于所述历史编译内容被引用的次数;或者,还可以通过其他方式确定该依赖程度,本说明书一个或多个实施例并不对此进行限制。

[0056] 在一实施例中,所述相关联的存储空间中存储有与所述历史编译内容相关的历史执行计划,其中所述依赖程度负相关于被从所述相关联的存储空间中淘汰且与所述历史编译内容相关的历史执行计划的数量。

[0057] 在一实施例中,执行次数和/或执行频率更低的历史执行计划被优先从所述相关联的存储空间中淘汰。

[0058] 由以上技术方案可见,本说明书中的一个或多个实施例通过存储历史执行计划,并将历史执行计划应用于对数据库访问请求的响应,可以实现对历史执行计划的复用,而无需重复生成相同的执行计划,从而能够减少数据库访问所需占用的处理资源,提升对执行计划的生成效率,并有助于加快数据库访问。

[0059] 在数据库访问的场景中,假定通过终端(例如图1所示的手机13或PC14等电子设备),向服务器(例如图1所示的服务器11)上运行的分布式数据库发起数据库访问请求,并由该服务器对该数据库访问请求进行响应。下面结合图3和图4,对上述的数据库访问过程进行描述。

[0060] 图3是一示例性实施例的一种对数据库访问请求进行响应的原理示意图。如图3所示,服务器可以维护有计划缓存,该计划缓存用于提供存储空间。在计划缓存中可以包括本地计划缓存,该本地计划缓存用于存储服务器针对历史数据库访问请求(即历史上接收到的数据库访问请求)生成的执行计划,即历史执行计划。

[0061] 服务器在接收到数据库访问请求后,通过查询本地计划缓存,确定是否存在与该数据库访问请求相匹配的历史执行计划。如果存在相匹配的历史执行计划,服务器可以执行该历史执行计划,以实现对上述数据库访问请求的响应,从而通过对历史执行计划的复用,无需重新生成数据库访问请求对应的执行计划,不仅有助于提升对数据库访问请求的响应效率,还可以减少对服务器的运算资源的占用。

[0062] 服务器可以维护有解释优化器,该解释优化器用于将数据库访问请求生成为相应的解释型执行计划。在一实施例中,如果服务器并未在本地计划缓存中查找到匹配于接收到的数据库访问请求的历史执行计划,服务器可以通过调用解释优化器,生成该数据库访问请求对应的解释型执行计划,并通过执行该解释型执行计划,响应于该数据库访问请求。

[0063] 服务器可以维护有编译优化器,该编译优化器用于将数据库访问请求生成为相应的编译型执行计划。在一实施例中,如果服务器并未在本地计划缓存中查找到匹配于接收到的数据库访问请求的历史执行计划,服务器可以通过调用编译优化器,生成该数据库访问请求对应的编译型执行计划,并通过执行该编译型执行计划,响应于该数据库访问请求。

[0064] 由于编译型执行计划相比于解释型执行计划,可能占用相对更大的处理时长,因而通过采用解释优化器生成解释型执行计划,可以更快响应于服务器接收到的数据库访问请求。下面以通过解释优化器生成解释型执行计划为例:

[0065] 对于解释优化器生成的解释型执行计划,可以存储至服务器维护的本地计划缓存中,以作为新的历史执行计划。可以理解为,在本地计划缓存中维护历史执行计划集合,该历史执行计划集合包含所有历史执行计划,而通过将生成的解释型执行计划添加至该历史执行计划集合中,以作为新的历史执行计划,可以不断实现对该历史执行计划集合的更新,以供后续接收到数据库访问请求时,可以对相应的历史执行计划予以复用。类似地,对于编译优化器生成的编译型执行计划,同样可以存储至本地计划缓存中,以作为新的历史执行计划。

[0066] 如上文所述,为了更快地响应于服务器接收到的数据库访问请求,可以选用解释优化器将数据库访问请求生成为解释型执行计划。但是由于该解释型执行计划可以被添加至本地计划缓存中,以供后续过程中的复用,因而为了提升后续复用过程中的执行效率,可以针对该本地计划缓存中的解释型执行计划,重新生成相应的编译型执行计划,并添加为本地计划缓存中的历史执行计划,以替换原本的解釋型执行计划。

[0067] 例如,编译优化器可以定期扫描本地计划缓存中的历史执行计划,以查找出其中的解释型执行计划。然后,编译优化器可以根据查找到的解释型执行计划对应的数据库访问请求,生成相应的编译型执行计划,以替换原本查找到的解释型执行计划;该替换过程可以是原子性的。

[0068] 在一实施例中,编译优化器对于本地计划缓存的扫描操作、对于编译型执行计划的生成操作、对于解释型执行计划的替换操作等,均可以对用户透明,以降低对用户可能造成的干扰和影响。

[0069] 在一实施例中,编译优化器可以利用服务器的空闲运算资源,实施对于本地计划缓存的扫描操作、对编译型执行计划的生成操作、对解释型执行计划的替换操作等,以避免对服务器的正常处理功能的影响。

[0070] 服务器可以维护动态编译函数库。在生成编译型执行计划的过程中,相比于解释型执行计划可能需要占用相对更多的内存资源,可以通过动态编译函数库在一定程度上解决该内存资源的占用问题。动态编译数据库用于存储编译优化器在历史上生成编译型执行计划的过程中生成的历史编译函数对应的历史编译内容,使得编译优化器在生成数据库访问请求对应的编译型执行计划的过程中,可以在动态编译数据库中查找该数据库访问请求中的待编译函数对应的历史编译内容,以实现对该历史编译内容的复用,而避免对该待编译函数的编译过程中造成对内存资源的额外占用。

[0071] 在一实施例中,动态编译数据库可以采用“key-value (键-值)”数据结构,key表示历史编译函数对应的函数签名(signature),value表示该历史编译函数对应的历史编译内容。例如,假定数据库访问请求1为:

[0072] `select sum(c1) from t1;`

[0073] 其中c1的类型为整形(integer),编译优化器可以生成对应sum(c1)的历史编译函数,该历史编译函数的函数签名可以为:`int sum(int arg1,int result) {}`,且该历史编译函数经过编译优化器处理得到相应的历史编译内容,并采用上述的key-value结构存储至

动态编译数据库中。

[0074] 那么,假定服务器接收到另一数据库访问请求2为:

[0075] `select sum(c1),avg(c2) from t1`

[0076] 其中,该数据库访问请求2包含待编译函数1为`sum(c1)`、待编译函数2为`avg(c2)`。

[0077] 对于作为待编译函数1的`sum(c1)`,编译优化器生成相应的函数签名为`int sum(int arg1,int result) {}`,并据此在动态编译数据库中进行搜索,以查找key为`int sum(int arg1,int result) {}`的key-value结构。而根据上文描述可知,编译优化器可以查找到相匹配的key-value结构,并将该key-value结构中的value表达的历史编译内容作为待编译函数1对应的编译内容,而无需对该待编译函数1进行重新编译,实现了对历史编译函数的复用,也提升了对数据库访问请求2的编译效率。

[0078] 而对于作为待编译函数2的`avg(c2)`,该c2的类型为float,编译优化器可以生成相应的函数签名为`int sum(float arg1,int result) {}`,编译优化器可能无法从动态编译数据库中查找到相匹配的key-value结构。那么,编译优化器无法复用历史编译函数,而需要实施编译操作,以得到相应的编译内容;类似地,对于`avg(c2)`及其对应的编译内容,可以被生成为相应的key-value结构,并存储至动态编译数据库中,以便在后续过程中被复用。

[0079] 在一实施例中,为了降低动态编译函数库占用的存储空间,可以对动态编译函数库中的key-value结构进行筛除,以确保对内存空间的有效利用。针对动态编译函数库中的每个key-value结构,可以维护对应的计数器(该计数器可以为原子性的),比如可以称之为引用计数,以用于表示相应的key-value结构在编译优化器生成编译型执行计划的过程中的被引用情况:在编译优化器将某一数据库访问请求生成为相应的编译型执行计划的过程中,如果在动态编译数据库中查找到相匹配的key-value结构并实现了复用,可以认为该key-value结构被该数据库访问请求“引用”,并将该key-value结构对应的引用计数进行增加(例如增加幅度可以为1或其他任意数值)。可见,引用计数可以表明编译型执行计划对相应的key-value结构的依赖程度,且引用计数与依赖程度呈正相关。

[0080] 在一些情况下,比如由于受到物理存储空间的限制,导致本地计划缓存的占用空间过大时,可能需要对该本地计划缓存中存储的历史执行计划进行淘汰,使其原本占用的存储空间得以释放。例如,可以按照每一历史执行计划被执行的次数和/或频率,淘汰掉执行次数和/或频率相对更低的历史执行计划,当然此处采用的淘汰方式并不受到限制。当某一历史执行计划被淘汰时,如果该历史执行计划存在引用过的key-value结构,可以对该key-value结构对应的引用计数进行减小(例如减小幅度可以与增加幅度相同;或者,减小幅度可以“惩罚性”地大于增加幅度;或者,减小幅度可以小于增加幅度)。

[0081] 当key-value结构对应的引用计数降为0时,表明不存在对该key-value结构产生依赖的历史执行计划,可以从动态编译函数库中筛除该key-value结构。例如,服务器可以通过一定时任务,定时扫描动态编译数据库中的key-value结构,并清除引用计数为0的key-value结构。

[0082] 图4是一示例性实施例的另一种对数据库访问请求进行响应的原理示意图。如图4所示,服务器可以采用分布式架构,该分布式架构可以包括多个节点,比如节点1表示接收到数据库访问请求的分布式节点,而节点2表示未接收到数据库访问请求的分布式节点。

[0083] 在节点1上维护有计划缓存1、节点2上维护有计划缓存2,所有分布式节点分别维

护的计划缓存共同构成了该分布式架构的服务器对应的分布式计划缓存。节点1维护的计划缓存1用于提供存储空间,该计划缓存1可以包括本地计划缓存1,该本地计划缓存1用于存储节点1针对历史数据库访问请求(即历史上接收到的数据库访问请求)生成的历史执行计划。类似地,节点2维护的计划缓存2可以包括本地计划缓存2,用于存储节点2针对历史数据库访问请求生成的历史执行计划。

[0084] 节点1在接收到数据库访问请求后,通过查询分布式计划缓存,确定是否存在与该数据库访问请求相匹配的历史执行计划。节点1在查询分布式计划缓存时,相当于分别确定节点1维护的本地计划缓存1、节点2维护的本地计划缓存2以及其他分布式节点维护的本地计划缓存中,是否存在该数据库访问请求对应的历史执行计划;其中,由于本地计划缓存1位于节点1本地,使得节点1可以对该本地计划缓存1进行搜索,而节点1还可以记录有节点2等其他分布式节点上的本地计划缓存中存储的历史执行计划的ID信息,以使得节点1可以据此确定其他分布式节点的本地计划缓存中是否存在该数据库访问请求对应的历史执行计划。

[0085] 如果查找到匹配于接收到的数据库访问请求的历史执行计划,可以通过执行该历史执行计划,以实现对上述数据库访问请求的响应,从而通过对历史执行计划的复用,无需重新生成数据库访问请求对应的执行计划。对于分布式架构而言,查找到的历史执行计划往往包括分别位于多个分布式节点上的多个部分,那么可以将每个部分确认为相应的历史执行子计划,而这些历史执行子计划共同构成上述的历史执行计划。例如,假定节点1确定本地计划缓存1中存在历史执行子计划1、本地计划缓存2中存在历史执行子计划2,那么节点1可以执行该历史执行子计划1以及使得节点2执行该历史执行子计划2(比如向该节点2发送执行指令,且执行指令中包含该历史执行子计划2的ID信息)。

[0086] 与图3中服务器维护的解释优化器相类似的,节点1可以维护有同步解释优化器1,用于将数据库访问请求生成为相应的解释型执行计划。在一实施例中,如果节点1并未在分布式计划缓存中查找到匹配于接收到的数据库访问请求的历史执行计划,节点1可以通过调用同步解释优化器1,生成该数据库访问请求对应的解释型执行计划。当解释型执行计划为分布式执行计划时,该解释型执行计划包括分别对应于节点1、节点2等各个分布式节点的部分,可以将对应于每个分布式节点的部分称为解释型执行子计划,比如对应于节点1的解释型执行子计划1、对应于节点2的解释型执行子计划2等。那么,节点1可以执行解释型执行子计划1,而将解释型执行子计划2发送至节点2、供节点2执行,以及将其他解释型执行子计划分别发送至相应的各个分布式节点、供各个分布式节点予以执行。通过由节点1上的同步解释优化器1生成各个解释型执行子计划,并由节点1将各个解释型执行子计划分别发送至相应的分布式节点,使得对应于各个分布式节点的解释型执行子计划可以理解为被“同步”生成和执行,因而节点1维护的解释优化器被称为“同步解释优化器”。类似地,每个分布式节点上均可以维护有相应的同步解释优化器,比如节点2维护有同步解释优化器2,以实现与上述同步解释优化器1相同或相类似的处理过程,此处不再赘述。

[0087] 与图3中服务器维护的编译优化器相类似的,节点1可以维护有异步编译优化器1,用于将数据库访问请求生成为相应的编译型执行计划。在一实施例中,如果节点1并未在分布式计划缓存中查找到匹配于接收到的数据库访问请求的历史执行计划,节点1可以通过调用异步编译优化器1,生成该数据库访问请求对应的编译型执行计划。当编译型执行计划

为分布式执行计划时,该编译型执行计划包括分别对应于节点1、节点2等各个分布式节点的部分,可以将对应于每个分布式节点的部分称为编译型执行子计划,比如对应于节点1的编译型执行子计划1、对应于节点2的编译型执行子计划2等。那么,节点1可以执行编译型执行子计划1,而将编译型执行子计划2发送至节点2、供节点2执行,以及将其他编译型执行子计划分别发送至相应的各个分布式节点、供各个分布式节点予以执行。类似地,每个分布式节点上均可以维护有相应的异步编译优化器,比如节点2维护有异步编译优化器2,以实现与上述异步编译优化器1相同或相类似的处理过程,此处不再赘述。

[0088] 下面以将节点1接收到的数据库访问请求生成为相应的解释型执行计划为例:

[0089] 在节点1根据接收到的数据库访问请求生成相应的解释型执行计划后,对于节点1执行的解释型执行子计划1,可以存储至节点1维护的本地计划缓存1中,以作为新的历史执行计划。类似地,对于节点1发送至其他分布式节点的解释型执行子计划等,也可以存储至相应的分布式节点维护的本地计划缓存中,并将相应的解释型执行子计划的ID信息告知节点1,比如解释型执行子计划2被节点2存储至维护的本地计划缓存2中,并将解释型执行子计划2的ID信息告知节点1、供节点1进行记录,以用于后续的查找过程中。

[0090] 与图3所示实施例相类似的,每个分布式节点针对自身的本地计划缓存中存储的解释型执行子计划,可以重新生成相应的编译型执行子计划,并添加为本地计划缓存中的历史执行计划,以替换原本的解释型执行子计划。如图4所示,节点1可以维护异步编译优化器1、节点2可以维护异步编译优化器2,使得节点1可以通过异步编译优化器1扫描本地计划缓存1中的解释型执行子计划、生成相应的编译型执行子计划,而节点2可以通过异步编译优化器2扫描本地计划缓存2中的解释型执行子计划、生成相应的编译型执行子计划;类似地,其他分布式节点也可以维护有相应的异步编译优化器,以用于扫描相应的本地计划缓存中的解释型执行子计划、生成相应的编译型执行子计划。在一实施例中,各个分布式节点可以分别采用自身维护的异步编译优化器执行“扫描解释型执行子计划”、“生成相应的编译型执行子计划”等操作,使得这些分布式节点之间的操作是异步实现的,不需要在各个分布式节点之间执行同步操作,可以简化对编译型执行计划(包括各个编译型执行子计划)的生成过程、提升其生成效率。

[0091] 在一实施例中,分布式计划缓存中的每个解释型执行计划可以被维护有对应的执行次数(在其他实施例中,可以采用执行频率,或者同时采用执行次数和执行频率等),而各个分布式节点上的异步编译优化器可以优先获取执行次数更高的解释型执行计划对应的解释型执行子计划,以生成相应的编译型执行子计划;由于该执行次数在各个分布式节点上是一致的,使得各个分布式节点虽然异步地生成编译型执行子计划,但是能够确保各个分布式节点对于编译型执行子计划的生成顺序一致。那么,对于某一数据库访问请求对应的执行计划而言,可以尽可能地使得该执行计划包含的执行子计划均为编译型执行子计划,而不会出现部分解释型执行子计划、部分编译型执行子计划,有助于缩短对该数据库访问请求的响应延迟(如果存在部分解释型执行子计划,那么由于解释型执行子计划的执行效率低于编译型执行子计划,使得即便所有的编译型执行子计划均已执行完毕,但只要存在一个未执行完毕的解释型执行子计划,就意味着无法响应数据库访问请求,降低了生成编译型执行子计划的意义)。

[0092] 例如,假定存在解释型执行计划1,包括位于节点1上的解释型执行子计划1、位于

节点2上的解释型执行子计划2,该解释型执行计划1对应的执行次数可以表现为节点1维护的执行次数S1、节点2维护的执行次数S2,当该解释型执行计划1被执行时,执行次数S1和执行次数S2均会产生相应的数值增加,确保该执行次数S1与执行次数S2之间的一致性。

[0093] 对于每个分布式节点而言,比如节点1通过异步编译器1扫描到解释型执行子计划1、生成相应的编译型执行子计划1,该过程与图3所示实施例中通过编译器生成编译型执行计划的过程相似,可以参考图3所示的实施例。其中,节点1可以通过计划缓存1中的动态编译函数库1,提升对编译型执行子计划1的生成效率,该过程与图3所示实施例中通过动态编译函数库生成编译型执行计划的过程相似,可以参考图3所示的实施例。

[0094] 与图3所示实施例不同的是,当图4所示实施例中的每个分布式节点在生成编译型执行子计划时,以节点1为例:异步编译器1除了可以利用计划缓存1维护的动态编译函数库1之外,还可以利用分布式计划缓存中的、维护于其他分布式节点上的其他动态编译函数库(比如节点2维护的动态编译函数库2等),使得每个分布式节点维护的key-value结构均可以在所有分布式节点上实现复用。

[0095] 在图4所示的实施例中,每个分布式节点均可以对分布式计划缓存中的所有缓存数据进行查询、调用等,从而在每个分布式节点各自的缓存空间有限的情况下,能够通过缓存数据的共享,进一步提升对数据库访问请求的响应效率。

[0096] 图5是一示例性实施例的一种电子设备的示意结构图。请参考图5,在硬件层面,该电子设备包括处理器502、内部总线504、网络接口506、内存508以及非易失性存储器510,当然还可能包括其他业务所需要的硬件。处理器502从非易失性存储器510中读取对应的计算机程序到内存508中然后运行,在逻辑层面上形成数据库访问装置。当然,除了软件实现方式之外,本说明书一个或多个实施例并不排除其他实现方式,比如逻辑器件抑或软硬件结合的方式等等,也就是说以下处理流程的执行主体并不限定于各个逻辑单元,也可以是硬件或逻辑器件。

[0097] 请参考图6,在软件实施方式中,该数据库访问装置可以包括:

[0098] 接收单元601,接收数据库访问请求;

[0099] 查询单元602,在相关联的存储空间存储的历史执行计划中,查询对应于所述数据库访问请求的历史执行计划;

[0100] 执行单元603,当查询到所述历史执行计划时,执行所述历史执行计划,以响应所述数据库访问请求。

[0101] 可选的,所述相关联的存储空间包括:相关联的缓存空间。

[0102] 可选的,所述相关联的存储空间包括以下至少之一:本地存储空间、相关联的分布式节点处的存储空间。

[0103] 可选的,还包括:

[0104] 解释单元604,当未查询到所述历史执行计划时,生成所述数据库访问请求对应的解释型执行计划;

[0105] 所述执行单元603具体用于执行所述解释型执行计划,以响应所述数据库访问请求。

[0106] 可选的,所述历史执行计划或所述解释型执行计划被所述执行单元603按照下述方式执行:

[0107] 当所述历史执行计划或所述解释型执行计划为本地计划时,在本地执行所述历史执行计划或所述解释型执行计划;

[0108] 当所述历史执行计划或所述解释型执行计划为分布式计划时,在本地执行所述历史执行计划或所述解释型执行计划中对应于本地节点的本地子计划;其中,所述历史执行计划或所述解释型执行计划中还包含对应于相关联的分布式节点的关联子计划,所述关联子计划由所述相关联的分布式节点执行。

[0109] 可选的,还包括:

[0110] 第一存储单元605,将所述解释型执行计划存储至所述相关联的存储空间,以作为新的历史执行计划。

[0111] 可选的,还包括:

[0112] 获取单元606,获取所述相关联的存储空间中的解释型历史执行计划;

[0113] 编译单元607,根据所述解释型历史执行计划对应的数据库访问请求,生成相应的编译型历史执行计划,以将所述解释型历史执行计划替换为编译型历史执行计划。

[0114] 可选的,所述获取单元606具体用于:

[0115] 从所述相关联的存储空间中对应于本地节点的本地存储空间,获取所述解释型历史执行计划。

[0116] 可选的,所述获取单元606具体用于:

[0117] 确定所述相关联的存储空间中的解释型历史执行计划的执行次数和/或执行频率;

[0118] 优先获取所述相关联的存储空间中执行次数和/或执行频率更高的解释型历史执行计划。

[0119] 可选的,所述编译单元607具体用于:

[0120] 获取所述解释型历史执行计划对应的数据库访问请求包含的待编译函数;

[0121] 当所述相关联的存储空间中存储有所述待编译函数对应的历史编译内容时,引用所述历史编译内容,以生成所述编译型历史执行计划。

[0122] 可选的,所述编译单元607具体用于:

[0123] 当所述相关联的存储空间中未存储有所述待编译函数对应的历史编译内容时,生成所述待编译函数对应的编译内容,以生成所述编译型历史执行计划。

[0124] 可选的,还包括:

[0125] 第二存储单元608,将所述待编译函数对应的编译内容存储至所述相关联的存储空间,以作为新的历史编译内容。

[0126] 可选的,还包括:

[0127] 确定单元609,确定生成所述编译型历史执行计划的过程对所述相关联的存储空间中的历史编译内容的依赖程度;

[0128] 删除单元610,从所述相关联的存储空间中删除依赖程度低于预设依赖程度的历史编译内容。

[0129] 可选的,所述依赖程度正相关于所述历史编译内容被引用的次数。

[0130] 可选的,所述相关联的存储空间中存储有与所述历史编译内容相关的历史执行计划,其中所述依赖程度负相关于被从所述相关联的存储空间中淘汰且与所述历史编译内容

相关的历史执行计划的数量。

[0131] 可选的,执行次数和/或执行频率更低的历史执行计划被优先从所述相关联的存储空间中淘汰。

[0132] 上述实施例阐明的系统、装置、模块或单元,具体可以由计算机芯片或实体实现,或者由具有某种功能的产品来实现。一种典型的实现设备为计算机,计算机的具体形式可以是个人计算机、膝上型计算机、蜂窝电话、相机电话、智能电话、个人数字助理、媒体播放器、导航设备、电子邮件收发设备、游戏控制台、平板计算机、可穿戴设备或者这些设备中的任意几种设备的组合。

[0133] 在一个典型的配置中,计算机包括一个或多个处理器(CPU)、输入/输出接口、网络接口和内存。

[0134] 内存可能包括计算机可读介质中的非永久性存储器,随机存取存储器(RAM)和/或非易失性内存等形式,如只读存储器(ROM)或闪存(flash RAM)。内存是计算机可读介质的示例。

[0135] 计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。计算机的存储介质的例子包括,但不限于相变内存(PRAM)、静态随机存取存储器(SRAM)、动态随机存取存储器(DRAM)、其他类型的随机存取存储器(RAM)、只读存储器(ROM)、电可擦除可编程只读存储器(EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器(CD-ROM)、数字多功能光盘(DVD)或其他光学存储、磁盒式磁带,磁带磁磁盘存储或其他磁性存储设备或任何其他非传输介质,可用于存储可以被计算设备访问的信息。按照本文中的界定,计算机可读介质不包括暂存电脑可读媒体(transitory media),如调制的数据信号和载波。

[0136] 还需要说明的是,术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、方法、商品或者设备不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、商品或者设备所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括所述要素的过程、方法、商品或者设备中还存在另外的相同要素。

[0137] 这里将详细地对示例性实施例进行说明,其示例表示在附图中。下面的描述涉及附图时,除非另有表示,不同附图中的相同数字表示相同或相似的要素。以下示例性实施例中所描述的实施方式并不代表与本说明书一个或多个实施例相一致的所有实施方式。相反,它们仅是与如所附权利要求书中所详述的、本说明书一个或多个实施例的一些方面相一致的装置和方法的例子。

[0138] 在本说明书一个或多个实施例使用的术语是仅仅出于描述特定实施例的目的,而非旨在限制本说明书一个或多个实施例。在本说明书一个或多个实施例和所附权利要求书中所使用的单数形式的“一种”、“所述”和“该”也旨在包括多数形式,除非上下文清楚地表示其他含义。还应当理解,本文中使用的术语“和/或”是指并包含一个或多个相关联的列出项目的任何或所有可能组合。

[0139] 应当理解,尽管在本说明书一个或多个实施例可能采用术语第一、第二、第三等来描述各种信息,但这些信息不应限于这些术语。这些术语仅用来将同一类型的信息彼此区分开。例如,在不脱离本说明书一个或多个实施例范围的情况下,第一信息也可以被称为第

二信息,类似地,第二信息也可以被称为第一信息。取决于语境,如在此所使用的词语“如果”可以被解释成为“在……时”或“当……时”或“响应于确定”。

[0140] 以上所述仅为本说明书一个或多个实施例的较佳实施例而已,并不用以限制本说明书一个或多个实施例,凡在本说明书一个或多个实施例的精神和原则之内,所做的任何修改、等同替换、改进等,均应包含在本说明书一个或多个实施例保护的范围之内。

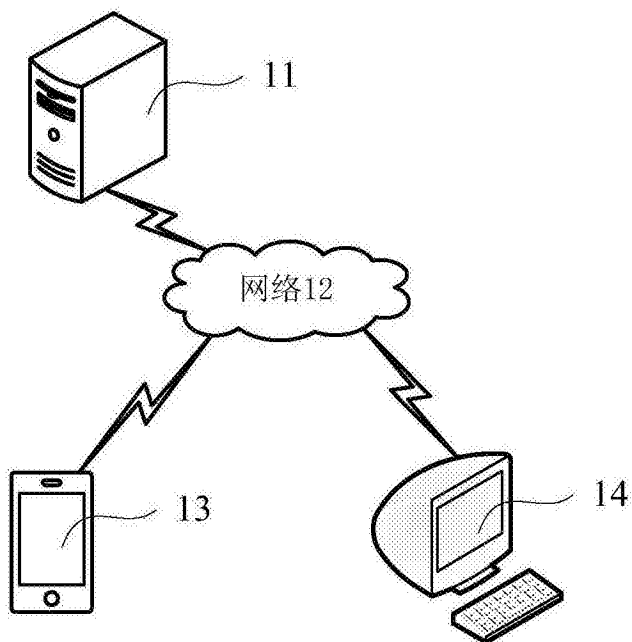


图1

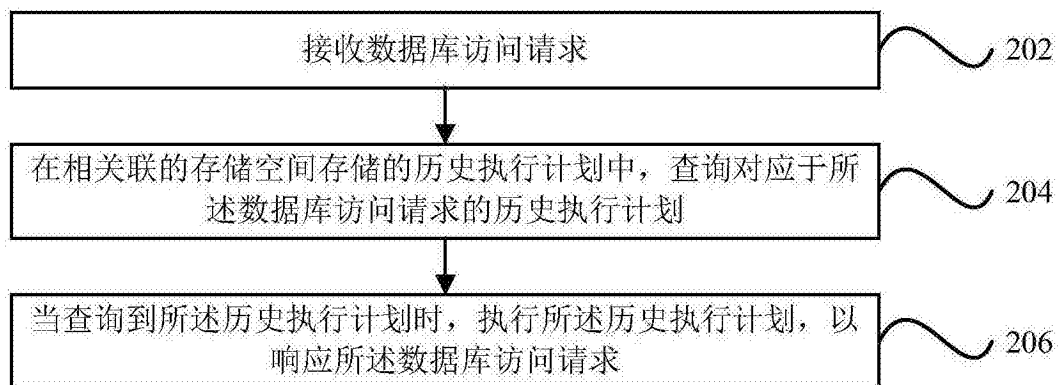


图2

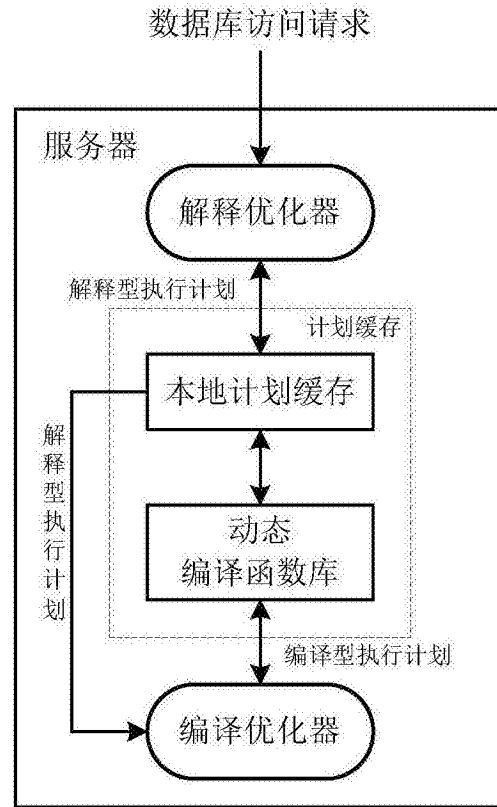


图3

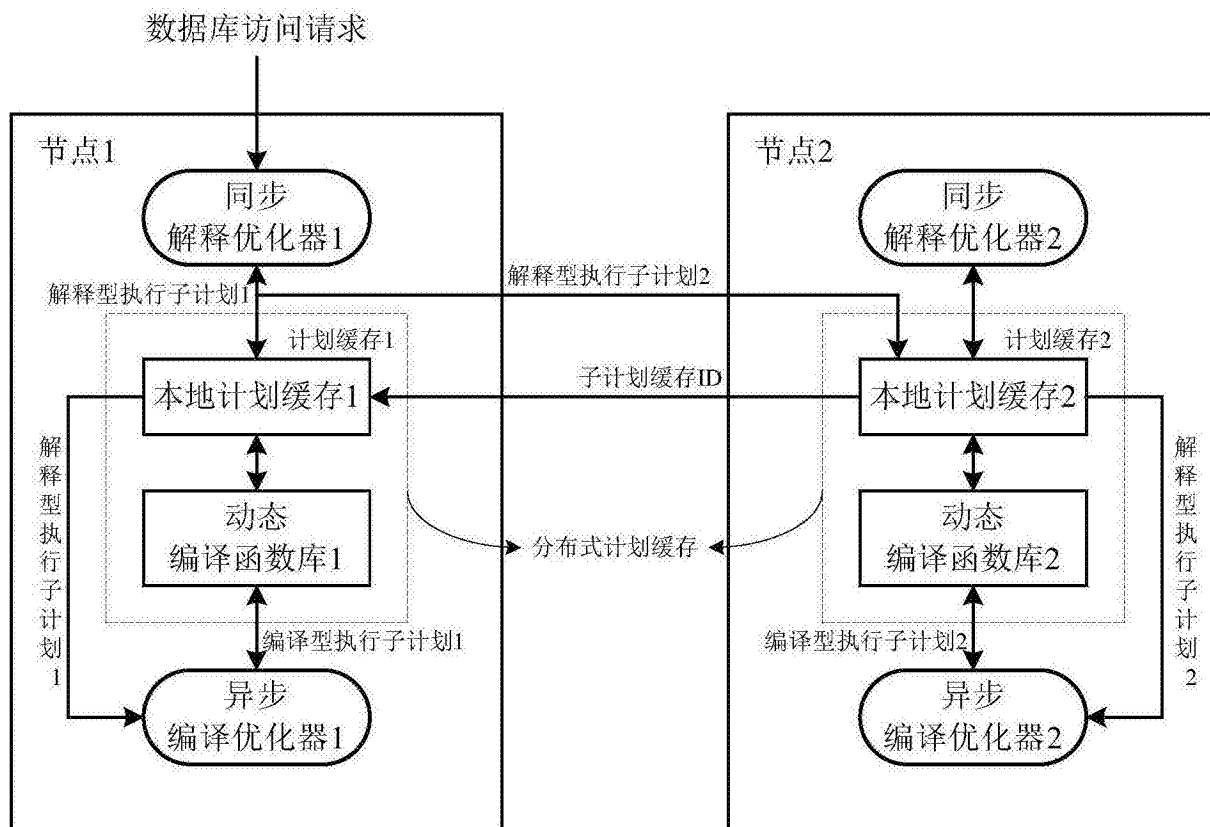


图4

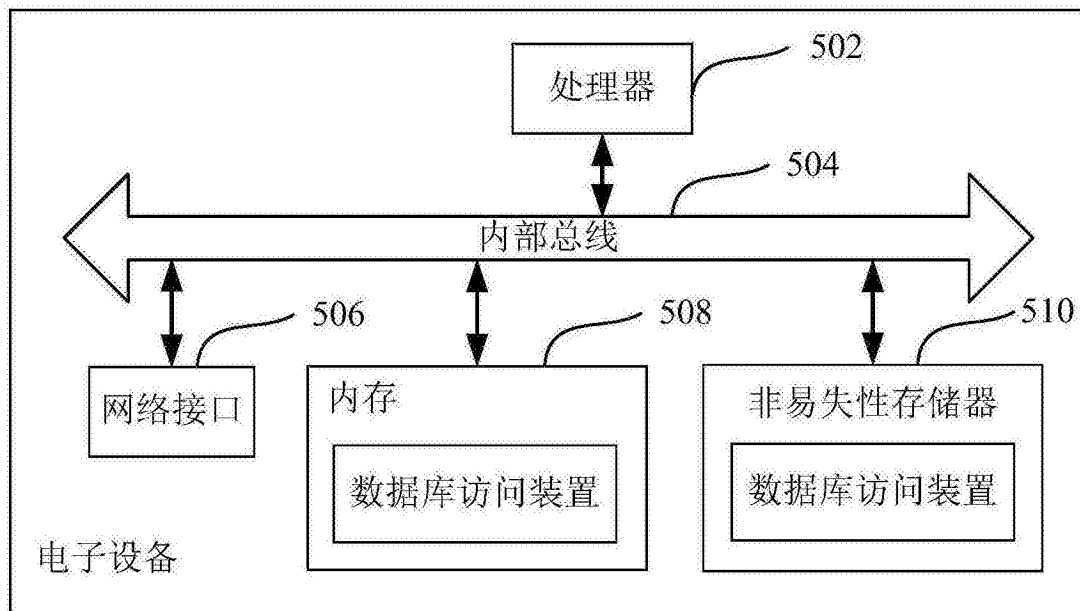


图5

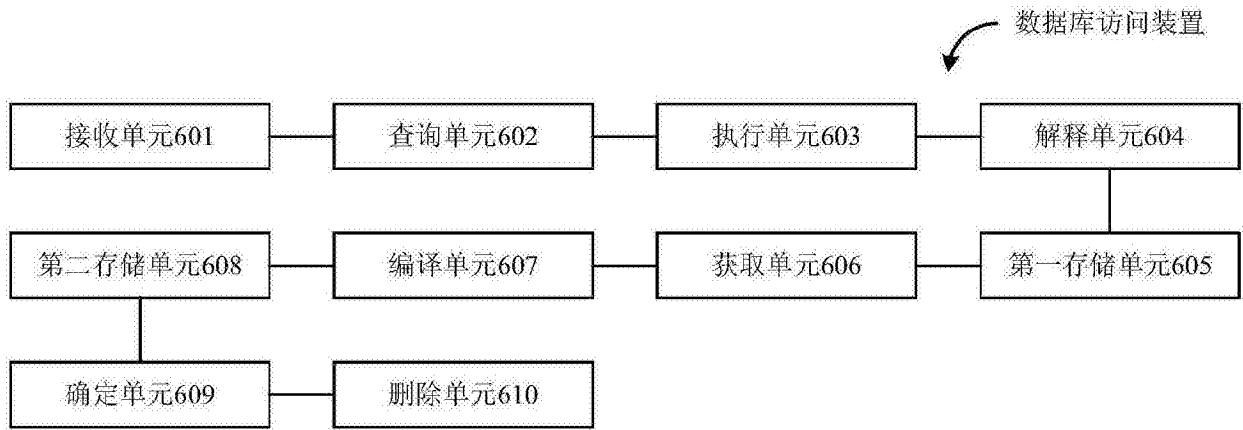


图6