

一种执行计划的在线演进方法、装置、设备和存储介质

技术领域

[01] 本说明书涉及数据库处理技术领域,尤其涉及一种执行计划的在线演进方法、装置、设备和存储介质。

背景技术

[02] 在执行数据库语句时,具体过程包括生成一个数据库语句对应的执行计划,以及执行生成的执行计划。同一个数据库语句可以存在不同的执行计划,且不同执行计划之间存在性能差异。为了避免性能较优的执行计划被性能较差的执行计划所替代,在相关技术中,部分数据库引入了数据库计划管理功能来解决计划回退的问题,其核心思路是总是执行过去使用的执行计划,而不去执行新的执行计划。

[03] 然而现有数据库的数据库计划管理功能在新的执行计划出现时,只会将其记录下来,如果用户不主动触发一个演进任务去验证新的执行计划的性能,那么数据库永远都不会去使用这个新的执行计划,即使它的性能比之前一直执行的执行计划更好。

发明内容

[04] 为克服相关技术中存在的问题,本说明书提供了一种执行计划的在线演进方法、装置、设备和存储介质。

[05] 根据本说明书实施例的第一方面,提供一种执行计划的在线演进方法,所述方法包括:响应于接收到的包含目标数据库语句的当前数据库请求,从所述目标数据库语句对应的计划缓存中获取演进任务,所述演进任务用于比较演进类型的执行计划和基线类型的执行计划的性能,所述演进类型的执行计划为对应于所述目标数据库语句且未经过验证的执行计划,所述基线类型的执行计划为对应于所述目标数据库语句且已经过验证的执行计划;选取所述当前数据库请求实际采用的执行计划的类型,以使得每种类型的执行计划在所述演进任务的实施过程中的累计执行次数与其性能呈正相关;在所述演进任务结束后,若确定所述演进类型的执行计划性能优于所述基线类型的执行计划,则将所述演进类型的执行计划转换为对应于所述目标数据库语句的基线类型的执行计划。

[06] 根据本说明书实施例的第二方面,提供一种执行计划的在线演进装置,包括:获取

模块，用于响应于接收到的包含目标数据库语句的当前数据库请求，从所述目标数据库语句对应的计划缓存中获取演进任务，所述演进任务用于比较演进类型的执行计划和基线类型的执行计划的性能，所述演进类型的执行计划为对应于所述目标数据库语句且未经过验证的执行计划，所述基线类型的执行计划为对应于所述目标数据库语句且已经过验证的执行计划；选择模块，用于选取所述当前数据库请求实际采用的执行计划的类型，以使得每种类型的执行计划在所述演进任务的实施过程中的累计执行次数与其性能呈正相关；判断模块，用于在所述演进任务结束后，若确定所述演进类型的执行计划性能优于所述基线类型的执行计划，则将所述演进类型的执行计划转换为对应于所述目标数据库语句的基线类型的执行计划。

[07]根据本说明书实施例的第三方面，提供一种电子设备，包括：处理器；用于存储处理器可执行指令的存储器；其中，所述处理器通过运行所述可执行指令以实现上述任一项所述的一种执行计划的在线演进方法。

[08]根据本说明书一个或多个实施例的第四方面，提出了一种计算机可读存储介质，其上存储有计算机指令，该指令被处理器执行时实现上述任一项所述的一种执行计划的在线演进方法的步骤。

[09]本说明书的实施例提供的技术方案可以包括以下有益效果：在本说明书实施例中，通过在目标数据库语句对应的计划缓存中记录对应的演进任务，使得在接收到的包含目标数据库语句的数据库请求时，可以基于该演进任务触发执行基线类型或演进类型的执行计划，以将基线类型的执行计划与演进类型的执行计划进行性能比对，从而实现对执行计划的自动演进。由于演进任务在实施过程中是针对真实的数据库请求进行处理，而并非额外的模拟处理，因而不会带来额外的性能消耗。同时，通过为接收到的包含目标数据库语句的数据库请求动态选择执行计划的类型，保证每种类型的执行计划在所述演进任务的实施过程中的累计执行次数与其性能呈正相关，使性能更优的执行计划执行次数更多，如果演进类型的执行计划性能相对更差，可以尽量弱化其带来的计划回退的影响，而如果演进类型的执行计划性能相对更优，可以提升演进任务的执行效率、加快演进，还有助于提升对于目标数据库语句的整体执行效率。

[10]应当理解的是，以上的一般描述和后文的细节描述仅是示例性和解释性的，并不能限制本说明书。

附图说明

[11] 此处的附图被并入说明书中并构成本说明书的一部分，示出了符合本说明书的实施例，并与说明书一起用于解释本说明书的原理。

[12] 图 1 是本说明书根据一示例性实施例示出的一种执行计划的在线演进方法的流程图。

[13] 图 2 是本说明书根据一示例性实施例示出的生成演进任务的流程图。

[14] 图 3 是本说明书根据一示例性实施例示出的演进任务的流程图。

[15] 图 4 是本说明书根据一示例性实施例示出的一种执行计划的在线演进装置的框图。

[16] 图 5 是本说明书根据一示例性实施例示出的一种电子设备的结构示意图。

具体实施方式

[17] 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、本说明书的一些方面相一致的装置和方法的例子。

[18] 在本说明书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。在本说明书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。

[19] 应当理解，尽管在本说明书可能采用术语第一、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”。

[20] 数据库是结构化信息或数据的有序集合，该有序集合通常由数据库管理系统（Database Management System, DBMS）来控制。在现实中，数据、数据库管理系统及关联应用一起被称为数据库系统，通常简称为数据库。当用户需要对数据库进行访问、管理、修改、更新、控制和组织数据等操作时，通常会使用数据库语句来实现，例如，在关系数据库中会采用遵循 SQL（结构化查询语言，Structured Query Language）标准的

数据库语句，也可以称为 SQL 语句。

[21] 用户所使用的数据库语句可以指明所需操作的数据库表，以及针对所述数据库表所需采用的算子，所述算子用于指示针对相应数据库表所采取的操作。但由于数据库语句只能向数据库表达“想要什么”，而无法直接向数据库表达“如何获取”这个结果，并且数据库表之间在实现连接操作时存在多种可能的连接顺序，每种算子也存在多种具体的实现算法，使得一条数据库语句可能存在很多种等价的获取结果的方法，需要由数据库中的查询优化器来决定获取结果的方法。这里每一种获取结果的方法都被称作一个“执行计划”，即执行计划可以指数据库用来访问数据的方法。

[22] 通常来说，数据库在为一条数据库请求生成一个新的执行计划时，总是会直接执行新的执行计划，导致在新的执行计划的性能比过去的执行计划的性能更差时，会产生计划的回退，影响后续数据库语句的执行效率。

[23] 基于上述问题，常用数据库提出离线演进的数据计划管理功能。大致流程包括：当数据库为同一条数据库请求生成新的执行计划后，新的执行计划被记录，后续由用户主动发起演进任务，数据库会在后台模拟接收数据库请求，分别执行新的执行计划和过去使用的执行计划，比较两者谁的性能更优。这种离线演进的处理方式，演进任务的执行过程会带来额外的资源消耗以实现对数据库请求的模拟处理；同时，由于新的执行计划不会用于执行数据库真实接收到的数据库请求，而是依旧会执行过去的执行计划，那么即便新的执行计划具有更高的执行效率，也无法立即奏效，降低了对于真实数据库请求的处理效率。

[24] 因此，本说明书提供了一种执行计划的在线演进方法，可以在避免计划回退的同时，提升演进任务的效率，实现自动执行最优计划，从而提升后续数据库语句的执行效率。

[25] 接下来对本说明书实施例进行详细说明。

[26] 如图 1 所示，图 1 是本说明书根据一示例性实施例示出的一种执行计划的在线演进方法的流程图，包括以下步骤 S101 至步骤 S103。

[27] S101，响应于接收到的包含目标数据库语句的当前数据库请求，从所述目标数据库语句对应的计划缓存中获取演进任务，所述演进任务用于比较演进类型的执行计划和基线类型的执行计划的性能，所述演进类型的执行计划为对应于所述目标数据库语句且未经过验证的执行计划，所述基线类型的执行计划为对应于所述目标数据库语句且已经过验证的执行计划。

[28]由用户侧设备发出包含数据库语句的数据库请求，并由数据库侧的设备进行数据库请求的接收和响应。在本说明书实施例中，数据库侧的设备可以为用于部署数据库的设备，当所述设备收到数据库请求后，直接在本地执行数据库语句对应的执行计划。或者，数据库侧的设备也可以为专用于对接用户侧设备与部署数据库的设备，该数据库侧的设备接收到来自用户侧设备的数据库请求后，将该请求发送至部署数据库的设备，最终由部署数据库的设备执行数据库语句对应的执行计划。

[29]所述数据库侧的设备存储有数据库语句对应的计划缓存，其中可以包括本说明书实施例所述的目标数据库语句对应的计划缓存。不同的数据库语句可以分别对应于不同的计划缓存。计划缓存用于记录相应的数据库语句所对应的执行计划，该执行计划具体可以体现为多种形式，比如已生成的基线类型的执行计划，再比如本说明书实施例中的演进任务，其中该演进任务进一步涉及到基线类型和演进类型的执行计划。实际上，计划缓存可以认为是针对执行计划的缓存，这使得数据库侧的设备在处理数据库请求中的数据库语句时，可以直接读取计划缓存中的执行计划予以执行，而不用临时生成执行计划。

[30]计划缓存中所存储的内容是动态变化的。当数据库侧的设备首次接收到包含目标数据库语句的数据库请求时，该目标数据库语句对应的计划缓存中显然不会存在任何执行计划。数据库侧的设备需要临时为该目标数据库语句生成执行计划。由于是首次处理，必然尚未存在该目标数据库语句对应的基线类型的执行计划，因而直接将临时生成的执行计划作为该目标数据库语句对应的基线类型的执行计划，并将该执行计划存储至计划缓存中。

[31]计划缓存中存储的基线类型的执行计划同样遵循“缓存”的一些处理机制，比如老化机制。具体地，在计划缓存中存储的基线类型的执行计划在预设时长内未被执行的情况下，该基线类型的执行计划将被从计划缓存中删除，从而节省数据库侧的设备的缓存空间，提升缓存利用率。除了上述的老化机制外，一些数据库方面的变化等也可能导致计划缓存中的基线类型的执行计划被淘汰，比如该变化可以包括新建索引等，本说明书并不对此进行限制。

[32]当然，即便计划缓存中的基线类型的执行计划被删除，也应当认为该目标数据库语句存在对应的基线类型的执行计划，只不过在具体执行的过程中需要临时生成基线类型的执行计划而无法从计划缓存中读取而已。因此，当数据库侧的设备后续接收到包含目标数据库语句的数据库请求时，如果从该目标数据库语句对应的计划缓存中读取到基线类型的执行计划，则直接执行读取到的执行计划，如果该目标数据库语句对应的计划缓

存中不存在基线类型的执行计划，则可能需要临时生成执行计划，并且需要确定该临时生成的执行计划是否与该目标数据库语句存在对应的基线类型的执行计划一致，若一致则表明其属于基线类型的执行计划，否则表明其属于演进类型的执行计划。在一实施例中，当数据库发生如前所述的新建索引等变化时，会导致针对目标数据库语句生成演进类型的执行计划，而非原本的基线类型的执行计划。对于生成的演进类型的执行计划，数据库侧的设备会进一步生成演进任务，从而针对该目标数据库语句的执行计划进行演进。

[33]在本说明书实施例中，演进任务用于对演进类型的执行计划和基线类型的执行计划的执行性能进行比较。其中，当数据库为一条数据库请求生成一个新的执行计划时，生成的执行计划为演进类型的执行计划，所述演进类型的执行计划为对应于所述目标数据库语句且未经过验证的执行计划。而在数据库生成新的执行计划之前使用的执行计划为基线类型的执行计划，所述基线类型的执行计划为对应于所述目标数据库语句且已经过验证的执行计划。

[34]在本说明书实施例中，可通过两种方式对生成的目标数据库语句对应的新的执行计划进行验证，第一种为根据演进任务得到的性能比对结果，若演进类型的执行计划性能优于基线类型的执行计划，则所述演进类型的执行计划的性能已经得到验证，从而可以转变为基线类型的执行计划。第二种为当目标数据库语句不存在对应的基线类型的执行计划时，由于不存在可对比的基线类型的执行计划，可直接将演进类型的执行计划认定为已经过验证，从而将其转变为基线类型的执行计划。如前文所述，数据库侧的设备在首次接收到包含目标数据库语句的数据库请求后，就基于第二种情形将新的执行计划认定为基线类型的执行计划。

[35]此外，目标数据库语句对应的基线类型的执行计划也可以存在淘汰机制。比如，目标数据库语句对应的任一基线类型的执行计划的未使用时长达到预设时长阈值，则表明该任一基线类型的执行计划的热度不高（可能由于性能不足等造成），为了避免占用数据库侧的设备的存储空间（用于记录该目标数据库语句对应的基线类型的执行计划），可以删除该任一基线类型的执行计划。如果目标数据库语句对应的基线类型的执行计划都被删除，就使得该目标数据库语句重新恢复为不存在对应的基线类型的执行计划的状态。可见，目标数据库语句可以同时存在一个或多个基线类型的执行计划，比如原本存在基线类型的执行计划 1，且某一演进类型的执行计划通过演进任务的验证后成为基线类型的执行计划 2；类似地，还可以存在更多数量的基线类型的执行计划。以基线类型

的执行计划 1 为例,如果该基线类型的执行计划 1 长时间不被使用,就可能基于上述的淘汰机制被删除,从而使得目标数据库语句只剩下常用的基线类型的执行计划,即基线类型的执行计划 2。这种方式也就实现了对基线类型的执行计划的汰换,并且为基线类型的执行计划 1 提供了预设时长阈值的过渡期,使得临时切换回使用基线类型的执行计划 1 时,无需再次经历演进任务的实施过程,降低资源消耗。

[36]数据库侧的设备在生成某一数据库语句对应的演进任务后,会将该演进任务存储至该数据库语句对应的计划缓存中。在上述实施例,由于在收到当前数据库请求时,目标数据库语句对应的计划缓存中已经存在相应的演进任务,表明数据库侧的设备已经在先前生成了针对该目标数据库语句的演进任务。该数据库侧的设备可以基于先前接收到的报案目标数据库语句的历史数据库请求,生成该目标数据库语句对应的演进任务,并存储至相应的计划缓存中。例如,响应于接收到的包含所述目标数据库语句的历史数据库请求,查询所述目标数据库语句对应的计划缓存,在所述计划缓存中不存在对应的演进任务和已生成的执行计划的情况下,生成所述目标数据库语句对应的数据库语句执行计划。在计划缓存中,要么存在已生成的执行计划,要么存在演进任务,两者不会同时存在。若生成的数据库语句执行计划为所述演进类型的执行计划,则生成所述演进任务并将所述演进任务缓存至所述目标数据库语句对应的计划缓存中。在针对历史数据库请求的处理过程中,如果目标数据库语句对应的计划缓存中存在演进任务,将直接执行该演进任务而不会触发生成新的演进任务,如果目标数据库语句对应的计划缓存中存在已生成的执行计划,将直接执行该执行计划而不会触发生成新的执行计划或演进任务,这与前文所述的演进任务的生成过程是一致的:只有目标数据库语句对应的计划缓存既不存在演进任务也不存在已生成的执行计划时,才会针对该目标数据库语句生成新的执行计划,并且在该新的执行计划属于演进类型的执行计划的情况下,才会进而触发生成演进任务。

[37]本领域技术人员应当理解的是:在本说明书的实施例,当前数据库请求与历史数据库请求只是相对概念,以便于区分不同数据库请求之间的接收顺序。具体地,数据库侧的设备首先接收到历史数据库请求,并据此生成演进任务,然后接收到当前数据库请求。

[38]在一示例性实施例中,当前数据库请求也可以不存在对应的历史数据库请求,或者该历史数据库请求可以并未生成演进任务。例如,响应于接收到的包含目标数据库语句的当前数据库请求,查询所述目标数据库语句对应的计划缓存,在所述计划缓存中不存

在对应的演进任务和已生成的执行计划的情况下，生成所述目标数据库语句对应的数据库语句执行计划，相当于上文中新的执行计划或其他类似概念。

[39]一种情况下，存在所述目标数据库语句对应的基线类型的执行计划，即表明当前数据库请求并非数据库侧的设备首次接收到的包含目标数据库语句的数据库请求，计划缓存中存储的基线类型的执行计划可能由于老化机制或其他原因被删除，故而需要临时生成上述的数据库语句执行计划。那么，可以将生成的数据库语句执行计划与所述基线类型的执行计划进行比较，当比较结果为两者一致时，表明生成的数据库语句执行计划就是基线类型的执行计划，则执行所述数据库语句执行计划，并将所述数据库语句执行计划缓存至所述目标数据库语句对应的计划缓存中。其中，可以比较生成的数据库语句执行计划和基线类型的执行计划的哈希值，若哈希值一致，则表明比较结果为两者一致，否则不一致。当然，除了基于哈希值进行比较之外，还可以通过其他任意方式对两个执行计划进行比较，本说明书并不对此进行限制。

[40]另一种情况下，不存在所述目标数据库语句对应的基线类型的执行计划，即表明当前数据库请求是数据侧的设备首次接收到的包含目标数据库语句的数据库请求，或者虽然并非首次接收到的包含目标数据库语句的数据库请求，但该目标数据库语句对应的基线类型的执行计划由于淘汰机制或其他原因被删除，故而需要临时生成上述的数据库语句执行计划。那么，可以将生成的数据库语句执行计划加入基线类型的执行计划中，执行所述数据库语句执行计划，并将所述数据库语句执行计划缓存至所述目标数据库语句对应的计划缓存中。

[41]S102，选取所述当前数据库请求实际采用的执行计划的类型，以使得每种类型的执行计划在所述演进任务的实施过程中的累计执行次数与其性能呈正相关。

[42]如前所述，执行计划的类型包括基线类型和演进类型。因此，为当前数据库请求选取的执行计划要么是演进类型的执行计划，要么是基线类型的执行计划。在目标数据库语句对应的演进任务的实施过程中，每收到一条包含该目标数据库语句的数据库请求，就会触发该演进任务执行一轮，每一轮都会选取执行一次演进类型的执行计划或基线类型的执行计划。

[43]在本说明书实施例中，可以通过任意方式来确定出每一轮所选取的执行计划的类型，只要确保整体的统计数据能够满足每种类型执行计划的累计执行次数与其性能呈正相关即可，本说明书对于执行计划的选取方式并不限制。

[44]通过设置每个执行计划的类型的累计执行次数与其性能呈正相关,使得执行性能越差的执行计划类型累计执行的次数越少,执行性能越好的执行计划类型累计执行的次数越多,从而最大程度上减小执行性能较差的执行计划带来的负面影响,快速验证出更优的执行计划,提升演进任务的执行效率。其中,如果演进类型的执行计划的执行效率相对更高,则证明演进类型的执行计划性能相对更优。由于本说明书实施例是基于在线演进的技术方案,即对于接收到的每一个包含目标数据库语句的数据库请求都进行了执行计划的类型的动态选择,并切实地执行了,因此当演进类型的执行计划性能相对更优时,选择演进类型的执行计划进行执行的包含目标数据库语句的那部分数据库请求的执行效率能够得到提升,再加上演进任务的实施过程中会更倾向于选择演进类型的执行计划,能够大幅提升数据库请求的整体执行效率。如果演进类型的执行计划的执行效率相对更低,则证明基线类型的执行计划性能相对更优,通过执行计划的类型的累计执行次数与其性能呈正相关的限制,减少了演进类型的执行计划在整个演进任务的实施过程中的累计执行次数,从而最大程度上减小性能较差的执行计划带来的负面影响,做到对计划回退基本无感知,在只存在少量可控的计划回退的前提下,快速验证出更优的执行计划,提升演进任务的执行效率。

[45]在一示例性实施例中,演进任务的实施过程可以包括多个阶段,其中:在首个阶段中,演进类型的执行计划和基线类型的执行计划的执行次数相同,这样可以直观地对演进类型的执行计划和基线类型的执行计划进行比较,以确定两者在首个阶段的执行效率。其中,每种类型的执行计划的执行次数可以为任意数值,例如分别执行 10 次或更少或更多次,本说明书并不对此进行限制。

[46]在非首个阶段的其他任一阶段中,各类型的执行计划的执行次数分别与自身在所述其他任一阶段之前的至少一个阶段的平均执行时长占比呈负相关。假定总共存在 N 个阶段, $N > 1$ 且为正整数,那么非首个阶段包括第 2 个阶段至第 N 个阶段;当然,如果 $N=2$,则非首个阶段仅为第 2 个阶段。本说明书对于 N 个取值并不限制。

[47]对于其他任一阶段而言,以第 i 个阶段为例 ($1 < i \leq N$, 且 i 为正整数):在该第 i 个阶段之前的所有阶段包括第 1 个阶段(即首个阶段)至第 $i-1$ 个阶段;那么,在该第 i 个阶段之前的至少一个阶段即为第 1 个阶段至第 $i-1$ 个阶段中的至少一个阶段。比如当 $i=2$ 时,该第 2 个阶段之前的至少一个阶段即为第 1 个阶段。再比如当 $i=3$ 时,该第 3 个阶段之前的至少一个阶段可以为第 1 个阶段,也可以为第 2 个阶段,还可以同时包括第 1 个阶段和第 2 个阶段。以此类推,此处不再一一列举。当然,不论该至少一个阶段

如何选取，应当确保为各个类型的执行计划选取相同的阶段，以使得相应计算的平均执行时长具有可比性。

[48]每一类型的执行计划在其他任一阶段之前的至少一个阶段的平均执行时长，可以为该类型的执行计划在该至少一个阶段的总执行时长与执行次数之比，也即该类型的执行计划在该至少一个阶段中执行一次的平均时长。本领域技术人员应当理解的是：当某一类型的执行计划在所述其他任一阶段之前的至少一个阶段的平均执行时长越长，则表明其执行效率越低，而平均执行时长越短，则表明其执行效率越高。因此，通过令各类型的执行计划的执行次数分别与自身在所述其他任一阶段之前的至少一个阶段的平均执行时长占比呈负相关，使得平均执行时长相对更长、执行效率相对更低的执行计划，在该其他任一阶段的执行次数相对更少，而使得平均执行时长相对更短、执行效率相对更高的执行计划，在该其他任一阶段的执行次数相对更多，确保性能较优的执行计划被选取执行的概率不断增大、性能较差的执行计划被选取执行的概率不断减小，有助于降低性能较差的执行计划带来的计划回退的影响，并提升演进任务的执行效率。

[49]当然，演进任务也可以不划分上述的多个阶段，即相当于仅包含一个阶段。那么，可以在演进任务的实施过程中，动态累计各个类型的执行计划的执行效率，比如累计各个类型的执行计划的平均执行时长，并据此对各个类型的执行计划进行选择执行，从而确保在整个演进任务的实施过程中，各个类型的执行计划的累计执行次数与其性能呈正相关，即执行性能越差，累计执行的次数越少，执行性能越好，累计执行的次数越多。

[50]在一示例性实施例中，在完成所述当前数据库请求选取的执行计划后，确定所述演进任务是否达到预定义的演进结束条件，其中，所述演进结束条件可以包括演进次数到达阈值和/或演进时长到达阈值，当然也可以采用其他的演进结束条件，本说明书并不对此进行限制。

[51]S103，在所述演进任务结束后，若确定所述演进类型的执行计划性能优于所述基线类型的执行计划，则将所述演进类型的执行计划转换为对应于所述目标数据库语句的基线类型的执行计划，

[52]在一示例性实施例中，根据所述演进任务的已执行情况，分别确定所述演进类型的执行计划和所述基线类型的执行计划的平均执行时长；将所述演进类型的执行计划和所述基线类型的执行计划中对应的平均执行时长相对较小的一方确定为性能相对较优的执行计划。例如，假定演进类型的执行计划在演进任务的整个实施过程中，每次的平均执行时长为 $T1$ ，而基线类型的演进计划在演进任务的整个实施过程中，每次的平均执

时长为 T_2 。那么，当 $T_1 < T_2$ 时，表明演进类型的执行计划的执行效率比基线类型的执行计划的执行效率更高，此时可以确定演进类型的执行计划的性能相对更优，而基线类型的执行计划的性能相对更差；而当 $T_1 > T_2$ 时，表明基线类型的执行计划的执行效率比演进类型的执行计划的执行效率更高，此时可以确定基线类型的执行计划的性能相对更优，而演进类型的执行计划的性能相对更差；当 $T_1 = T_2$ 时，可以根据预定义的处理策略，确定是否认可演进类型的执行计划通过了验证。

[53]通过演进任务得到性能比对结果，若演进类型的执行计划性能优于基线类型的执行计划，则所述演进类型的执行计划的性能已经得到验证，将其转换为对应于所述目标数据库语句的基线类型的执行计划，并缓存至对应于所述目标数据库语句的计划缓存中；若基线类型的执行计划性能优于演进类型的执行计划，则丢弃性能较差的演进类型的执行计划，将基线类型的执行计划缓存至对应于所述目标数据库语句的计划缓存中。当数据库侧的设备再接收到包含目标数据库语句的数据库请求时，即可直接执行对应于所述目标数据库语句的计划缓存中缓存的执行计划。通过演进任务，快速判断演进类型的执行计划和基线类型的执行计划的性能优劣，将性能相对更优的执行计划作为目标数据库语句对应的后续的执行计划，从而提升对于目标数据库语句的整体执行效率。

[54]在本说明书实施例中，通过在目标数据库语句对应的计划缓存中记录对应的演进任务，使得在接收到的包含目标数据库语句的数据库请求时，可以基于该演进任务触发执行基线类型或演进类型的执行计划，以将基线类型的执行计划与演进类型的执行计划进行性能比对，从而实现对执行计划的自动演进。由于演进任务在实施过程中是针对真实的数据库请求进行处理，而并非额外的模拟处理，因而不会带来额外的性能消耗。同时，通过为接收到的包含目标数据库语句的数据库请求动态选择执行计划的类型，保证每种类型的执行计划在所述演进任务的实施过程中的累计执行次数与其性能呈正相关，使性能更优的执行计划执行次数更多，如果演进类型的执行计划性能相对更差，可以尽量弱化其带来的计划回退的影响，而如果演进类型的执行计划性能相对更优，可以提升演进任务的执行效率、加快演进，还有助于提升对于目标数据库语句的整体执行效率。

[55]不仅在基线类型的执行计划优于演进类型的执行计划时，能做到及时淘汰演进类型的执行计划，弱化性能较差的执行计划带来的计划回退的影响，也能在演进类型的执行计划优于基线类型的执行计划时，及时使用更好的执行计划，同时提升选择演进类型的执行计划进行执行的部分数据库请求的执行效率。最终以只存在少量可控的计划回退的前提下，快速验证出更优的执行计划，提升数据库语句的执行效率。

[56]如图 2 所示，图 2 是本说明书根据一示例性实施例示出的生成演进任务的流程图。

具体步骤如下 S201 至 S210。

[57]S201，接收数据库请求。数据库接收用户发送的包含目标数据库语句的数据库请求。

[58]S202，计划缓存中是否存在已生成的执行计划或演进任务。查询所述目标数据库语句对应的计划缓存，判断在计划缓存中是否存在对应于目标数据库请求的已生成的执行计划或对应的演进任务，若没有，进入步骤 S203，若有，则进入步骤 S204。

[59]S203，生成新的执行计划。若所述目标数据库语句对应的计划缓存中不存在对应于目标数据库请求的可用的执行计划和演进任务，此时数据库为所述包含目标数据库语句的数据库请求生成新的数据库语句执行计划。

[60]S204，执行已生成的执行计划或演进任务。若所述目标数据库语句对应的计划缓存中存在对应于目标数据库请求的已生成的执行计划或演进任务，则直接执行该对应的已生成的执行计划或演进任务。

[61]S205，是否存在基线类型的执行计划。判断目标数据库语句是否存在基线类型的执行计划，若存在，进入步骤 S206，若不存在，则进入步骤 S207。

[62]S206，是否为基线类型的执行计划。当目标数据库语句存在基线类型的执行计划，判断对应于目标数据库语句的新的执行计划是否为基线类型的执行计划，若是，进入步骤 S209，若不是，则进入步骤 S208。

[63]S207，加入基线类型的执行计划。当目标数据库语句不存在基线类型的执行计划，将对应于目标数据库语句的新的执行计划加入基线类型的执行计划，用于后续再次接收到包含目标数据库语句的数据库请求，并生成新的执行计划时，判断所述新的执行计划是否为基线类型的执行计划。

[64]S208，生成演进任务。若对应于目标数据库语句的新的执行计划不是基线类型的执行计划，则确定其为演进类型的执行计划，将演进类型的执行计划和性能最优的基线类型的执行计划的数据保存，并执行其中任一类型的执行计划。

[65]S209，执行当前执行计划。当确定对应于目标数据库语句的新的执行计划为基线类型的执行计划或加入基线类型的执行计划时，执行当前的执行计划，进入步骤 210。

[66]S210，加入计划缓存。将步骤 S208 中生成的对应于目标数据库请求的演进任务或步骤 S209 中生成的对应于目标数据库请求的执行计划加入对应于目标数据库请求的计

划缓存中。

[67]上述步骤的实现过程具体详见图 1 实施例中对应步骤的实现过程，在此不再赘述。

[68]图 3 是本说明书根据一示例性实施例示出的演进任务的流程图。具体步骤如下 S301 至 S311。

[69]S301，接收数据库请求。当接收到包含目标数据库语句的当前数据库请求并且目标数据库语句对应的计划缓存中存在演进任务时，获取所述演进任务。

[70]S302，选择执行计划的类型。动态选择执行演进类型的执行计划或基线类型的执行计划。

[71]S303，执行选择的类型的执行计划。执行选择的类型的执行计划，并记录此次执行时长。

[72]S304，是否达到演进结束条件。若到达演进结束条件，则进入步骤 S305；若未到达演进结束条件，则进入步骤 S306。

[73]S305，结束演进任务。结束演进任务，将所述演进任务从目标数据库语句对应的计划缓存中删除。

[74]S306，继续演进任务。包含所述目标数据库语句的当前数据库请求执行结束，但目标数据库语句对应的计划缓存中仍保留演进任务用于下一次接收包含所述目标数据库语句的数据库请求。

[75]S307，演进类型的执行计划是否优于基线类型的执行计划。若演进类型的执行计划优于基线类型的执行计划，则进入步骤 S308；若基线类型的执行计划优于演进类型的执行计划，则进入步骤 S310。

[76]S308，将演进类型的执行计划转换为基线类型的执行计划。将演进类型的执行计划转换为目标数据库语句对应的基线类型的执行计划。

[77]S309，加入计划缓存。将演进类型的执行计划转换而来的目标数据库语句对应的基线类型的执行计划作为目标数据库语句对应的执行计划加入目标数据库语句对应的计划缓存中。

[78]S310，丢弃演进类型的执行计划，使用基线类型的执行计划。

[79]S311，加入计划缓存。将基线类型的执行计划作为目标数据库语句对应的执行计划加入目标数据库语句对应的计划缓存中。

[80]上述步骤的实现过程具体详见图 1 实施例中对对应步骤的实现过程，在此不再赘述。

[81]与前述方法的实施例相对应，本说明书还提供了装置及其所应用的终端的实施例。

[82]如图 4 所示，图 4 是本说明书根据一示例性实施例示出的一种装置的框图，所述装置包括：获取模块 401，用于响应于接收到的包含目标数据库语句的当前数据库请求，从所述目标数据库语句对应的计划缓存中获取演进任务，所述演进任务用于比较演进类型的执行计划和基线类型的执行计划的性能，所述演进类型的执行计划为对应于所述目标数据库语句且未经过验证的执行计划，所述基线类型的执行计划为对应于所述目标数据库语句且已经过验证的执行计划；选择模块 402，用于选取所述数据库请求实际采用的执行计划的类型，以使得每种类型的执行计划在所述演进任务的实施过程中的累计执行次数与其性能呈正相关；判断模块 403，用于在所述演进任务结束后，若确定所述当前演进类型的执行计划性能优于所述基线类型的执行计划，则将所述演进类型的执行计划转换为对应于所述目标数据库语句的基线类型的执行计划。

[83]上述装置中各个模块的功能和作用的实现过程具体详见上述方法中对对应步骤的实现过程，在此不再赘述。

[84]对于装置实施例而言，由于其基本对应于方法实施例，所以相关之处参见方法实施例的部分说明即可。以上所描述的装置实施例仅仅是示意性的，其中所述作为分离部件说明的模块可以是或者也可以不是物理上分开的，作为模块显示的部件可以是或者也可以不是物理模块，即可以位于一个地方，或者也可以分布到多个网络模块上。可以根据实际的需要选择其中的部分或者全部模块来实现本说明书方案的目的。本领域普通技术人员在不付出创造性劳动的情况下，即可以理解并实施。

[85]如图 5 所示，图 5 是根据本说明书一示例性实施例示出的一种电子设备的结构示意图。

[86]参考图 5，在硬件层面，该电子设备包括处理器 501、内部总线 502、网络接口 503、内存 504 以及非易失性存储器 505，当然还可能包括其他业务所需要的硬件。处理器 501 从非易失性存储器 505 中读取对应的计算机程序到内存 504 中然后运行。当然，除了软件实现方式之外，本说明书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于各个逻辑单元，也可以是硬件或逻辑器件。

[87]上述实施例阐明的装置或单元，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。

[88] 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。

[89] 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。

[90] 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、磁盘存储、量子存储器、基于石墨烯的存储介质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。

[91] 上述对本说明书特定实施例进行了描述。其它实施例在所附权利要求书的范围内。在一些情况下，在权利要求书中记载的动作或步骤可以按照不同于实施例中的顺序来执行并且仍然可以实现期望的结果。另外，在附图中描绘的过程不一定要求示出的特定顺序或者连续顺序才能实现期望的结果。在某些实施方式中，多任务处理和并行处理也是可以的或者可能是有利的。

[92] 本领域技术人员在考虑说明书及实践这里申请的发明后，将容易想到本说明书的其它实施方案。本说明书旨在涵盖本说明书的任何变型、用途或者适应性变化，这些变型、用途或者适应性变化遵循本说明书的一般性原理并包括本说明书未申请的本技术领域中的公知常识或惯用技术手段。说明书和实施例仅被视为示例性的，本说明书的真正范围和精神由权利要求指出。

[93] 应当理解的是，本说明书并不局限于上面已经描述并在附图中示出的精确结构，并

且可以在不脱离其范围进行各种修改和改变。本说明书的范围仅由所附的权利要求来限制。

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

权利要求书

1、一种执行计划的在线演进方法，包括：

响应于接收到的包含目标数据库语句的当前数据库请求，从所述目标数据库语句对应的计划缓存中获取演进任务，所述演进任务用于比较演进类型的执行计划和基线类型的执行计划的性能，所述演进类型的执行计划为对应于所述目标数据库语句且未经过验证的执行计划，所述基线类型的执行计划为对应于所述目标数据库语句且已经过验证的执行计划；

选取所述当前数据库请求实际采用的执行计划的类型，以使得每种类型的执行计划在所述演进任务的实施过程中的累计执行次数与其性能呈正相关；

在所述演进任务结束后，若确定所述演进类型的执行计划性能优于所述基线类型的执行计划，则将所述演进类型的执行计划转换为对应于所述目标数据库语句的基线类型的执行计划。

2、根据权利要求1所述的方法，所述方法还包括：

响应于接收到的包含所述目标数据库语句的历史数据库请求，查询所述目标数据库语句对应的计划缓存，在所述计划缓存中不存在对应的演进任务和已生成的执行计划的情况下，生成所述目标数据库语句对应的数据库语句执行计划；

若生成的数据库语句执行计划为所述演进类型的执行计划，则生成所述演进任务并将所述演进任务缓存至所述目标数据库语句对应的计划缓存中。

3、根据权利要求1所述的方法，所述方法还包括：

响应于接收到的包含目标数据库语句的当前数据库请求，查询所述目标数据库语句对应的计划缓存，在所述计划缓存中不存在对应的演进任务和已生成的执行计划的情况下，生成所述目标数据库语句对应的数据库语句执行计划；

若存在所述目标数据库语句对应的基线类型的执行计划，则将生成的数据库语句执行计划与所述基线类型的执行计划进行比较，当比较结果为两者一致时，则执行所述数据库语句执行计划，并将所述数据库语句执行计划缓存至所述目标数据库语句对应的计划缓存中；

若不存在所述目标数据库语句对应的基线类型的执行计划，则将生成的数据库语句执行计划加入基线类型的执行计划中，执行所述数据库语句执行计划，并将所述数据库语句执行计划缓存至所述目标数据库语句对应的计划缓存中。

4、根据权利要求1所述的方法，所述演进任务的实施过程包括多个阶段，其中：

在首个阶段中，演进类型的执行计划和基线类型的执行计划的执行次数相同；

在非首个阶段的其他任一阶段中，各类型的执行计划的执行次数分别与自身在所述其他任一阶段之前的至少一个阶段的平均执行时长占比呈负相关。

5、根据权利要求1所述的方法，所述方法还包括：

在完成所述当前数据库请求选取的执行计划后，确定所述演进任务是否达到预定义的演进结束条件；其中，所述演进结束条件包括演进次数到达阈值和/或演进时长到达阈值；

若到达演进结束条件，则结束所述演进任务。

6、根据权利要求1所述的方法，所述方法还包括：

根据所述演进任务的已执行情况，分别确定所述演进类型的执行计划和所述基线类型的执行计划的平均执行时长；

将所述演进类型的执行计划和所述基线类型的执行计划中对应的平均执行时长相比较小的一方确定为性能相对较优的执行计划。

7、根据权利要求1所述的方法，所述方法还包括：

在所述目标数据库语句对应的任一基线类型的执行计划的未使用时长到达预设时长阈值的情况下，删除所述任一基线类型的执行计划。

8、一种执行计划的在线演进装置，包括：

获取模块，用于响应于接收到的包含目标数据库语句的当前数据库请求，从所述目标数据库语句对应的计划缓存中获取演进任务，所述演进任务用于比较演进类型的执行计划和基线类型的执行计划的性能，所述演进类型的执行计划为对应于所述目标数据库语句且未经过验证的执行计划，所述基线类型的执行计划为对应于所述目标数据库语句且已经过验证的执行计划；

选择模块，用于选取所述当前数据库请求实际采用的执行计划的类型，以使得每种类型的执行计划在所述演进任务的实施过程中的累计执行次数与其性能呈正相关；

判断模块，用于在所述演进任务结束后，若确定所述演进类型的执行计划性能优于所述基线类型的执行计划，则将所述演进类型的执行计划转换为对应于所述目标数据库语句的基线类型的执行计划。

9、一种电子设备，包括：

处理器；

用于存储处理器可执行指令的存储器；

其中，所述处理器通过运行所述可执行指令以实现如权利要求1-7中任一项所述的一种执行计划的在线演进方法。

10、一种计算机可读存储介质，其上存储有计算机指令，该指令被处理器执行时实现如权利要求 1-7 中任一项所述的一种执行计划的在线演进方法。