(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 111949687 A
(43)申请公布日 2020.11.17

(21)申请号 202010772287.5

(22)申请日 2020.08.04

(71)申请人 贵州易鲸捷信息技术有限公司
地址 550000 贵州省贵阳市贵阳综合保税区都拉营综保路349号海关大楼8楼801

(72)发明人 王效忠 冀贤亮 何振兴 李英帅

(74)专利代理机构 成都中炬新汇知识产权代理有限公司 51279
代理人 罗韬

(51)Int.Cl.
G06F 16/2453(2019.01)
G06F 16/2455(2019.01)
G06F 16/2458(2019.01)
G06F 9/54(2006.01)

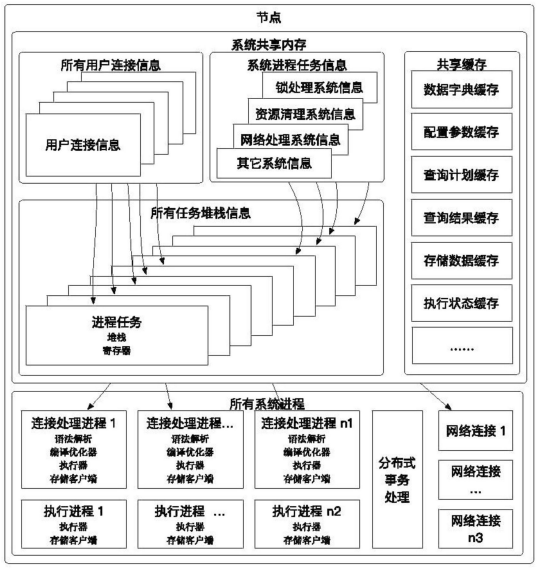
权利要求书2页 说明书6页 附图2页

(54)发明名称

基于共享内存和多进程的分布式数据库架构及其实现方法

(57)摘要

本发明公开了一种基于共享内存和多进程的分布式数据库架构及实现方法,属一种分布式数据库架构,其包括分布式数据库节点,分布式数据库内置系统共享内存单元与系统进程单元;系统共享内存单元包括任务堆栈信息模块与共享缓存模块;任务堆栈信息模块内置多个进程任务;进程任务为系统进程任务信息中的多种用途的系统信息,每个系统信息均对应一个进程任务;通过在分布式数据库节点使用系统共享内存单元,使得在该分布式数据库架构中用户的连接数不与进程或者线程存在对应关系,整个节点的进程或者线程数都不会因为用户连接数的增加而增加,从而有效避免因瞬时用户连接数过多而导致系统响应速度变慢,从而使系统性能不会因此而受到影响。



CN 111949687 A

1.一种基于共享内存和多进程的分布式数据库架构,其特征在于:其包括分布式数据库节点,所述分布式数据库内置系统共享内存单元与系统进程单元;

所述系统共享内存单元包括任务堆栈信息模块与共享缓存模块;

所述任务堆栈信息模块内置多个进程任务;

所述进程任务为系统进程任务信息中的多种用途的系统信息,每个系统信息均对应一个进程任务;以及所有用户连接信息所关联的进程任务;

所述进程任务中包括对应的堆栈信息与寄存器信息;

所述共享缓存模块用于保存分布式数据库节点的多个连接,或者各个进程之间需要共享的信息;

所述系统进程单元包括连接处理进程、执行进程、分布式事务处理进程与网络连接进程;用于由系统进程单元中的各个进程加载执行所述任务堆栈信息模块中的进程任务,或者在多个进程任务之间切换,其中:

所述连接处理进程用于执行客户端发送的查询、插入、更新和删除请求;

所述执行进程用于执行连接处理进程派生出的并发执行进程任务;

所述分布式事务处理进程与网络连接进程分别用于执行分布式事务处理任务与网络连接任务。

2.根据权利要求1所述的基于共享内存和多进程的分布式数据库架构,其特征在于:所述分布式数据库节点为多个,每个所述分布式数据库节点的内部架构均相同,且每个所述分布式数据库节点的系统共享内存单元均由系统统一分配使用。

3.根据权利要求1所述的基于共享内存和多进程的分布式数据库架构,其特征在于:所述系统进程任务信息中,根据不同的用途划分为不同的系统信息,其至少包括锁处理系统信息、资源清理系统信息、网络处理系统信息与其它系统信息。

4.根据权利要求1所述的基于共享内存和多进程的分布式数据库架构,其特征在于:所述共享缓存模块用于保存各个进程之间需要共享的信息包括数据字典、配置参数、查询计划、查询结果、执行状态和存储数据。

5.根据权利要求1或4所述的基于共享内存和多进程的分布式数据库架构,其特征在于:所述共享缓存模块中保存的各个进程之间需要共享的信息,在系统启动时加载,并在系统执行过程中与存储层的信息保持一致。

6.一种权利要求1至5任意一项所述分布式数据库架构的实现方法,其特征在于所述的方法包括如下步骤:

步骤A、系统启动,分配分布式数据库节点中的系统共享内存单元,并进行初始化,使系统共享内存单元中的所有用户连接信息、系统进程任务信息、进程任务中的堆栈信息与寄存器信息、以及共享缓存模块中的信息初始化;此时如内存已经分配,则报告系统以及该启动的错误;

步骤B、系统进程单元中的各个进程从任务堆栈信息模块中获取需要执行的进程任务,如没有更多的任务,就执行空任务;

当有客户端发起连接请求时,系统进程单元中的连接处理进程接收到并响应的用户请求,连接处理进程为当前用户请求生成一个相应的进程任务,并放入共享内存中的所有用户连接信息中;

步骤C、当系统进程单元中的任意一个进程执行任务堆栈信息模块中的任意一个进程任务时，出现阻塞等待，当前进程即放弃执行当前进程任务，并将该进程任务交由后台进程异步等待。

7.根据权利要求6所述的方法，其特征在于：所述步骤B中，如用户发起的是客户端连接请求，连接处理进程会创建一个用户连接信息，并将用户连接信息与一个进程任务相关联，然生成的进程任务放入所有用户连接信息中；如用户发起的是客户端查询请求，连接处理进程会将查询请求信息加入到对应进程任务关联的用户连接信息中。

8.根据权利要求6所述的方法，其特征在于：所述的方法还包括步骤D、所述系统进程单元中的各进程循环的从任务堆栈信息模块中获取进程任务并执行；当执行到有用户信息的进程任务时，进一步查看是否为用户连接请求，如是则由连接处理进程执行创建连接的相关流程并创建用户连接信息；如是用户查询信息，则执行进程会从用户信息上获取用户的查询语句，并执行相关查询。

9.根据权利要求6或8所述的方法，其特征在于：所述的方法还包括步骤E、系统关闭时，先关闭所有的处理进程，然后再由一个进程释放所有分布式数据库节点的系统共享内存单元的共享内存资源。

基于共享内存和多进程的分布式数据库架构及其实现方法

技术领域

[0001] 本发明涉及一种分布式数据库架构，更具体的说，本发明主要涉及一种基于共享内存和多进程的分布式数据库架构及实现方法。

背景技术

[0002] 目前要求高并发的应用程序，基本都是通过连接中间件实现，连接中间件要做很多分析合并处理，将用户的查询请求经过分析以后分发到相应的单机数据库处理，将查询得到的结果通过合并处理以后发给前面的应用程序，因此中间件存在性能瓶颈，为了解决这个问题，必须将中间件也开发成分布式系统，这就导致每一层都是分布式架构系统，对于管理和运维带来了很多问题，另外对资源也是一种极大的浪费。随着技术的发展，数据库也发展出了分布式架构，分布式数据库能提高应用开发的复杂度，同时分布式数据库也可以大大的提高可支持的连接数。分布式中间件加分布式数据库虽然可以解决高并发的问题，但中间件和数据库都是分布式系统，这会大大的增加应用系统后面的请求和数据的流转时间，导致响应时间会大大的增加。一种可行的解决办法是提高分布式数据库的并发支持能力，去掉分布式中间件以减少流转流程，降低响应时间。但无论基于那种架构的应用程序，都需要有分布式数据库的支撑。目前要支持高并发都摆脱不了分布式数据库，目前的分布式数据库架构都基于独立的多进程，或者多线程实现，这两种实现方式或多或少都存在一些问题。多个进程或者线程会以操作系统的时间片为单位切换，进程或者线程多了以后会导致切换的代价越来越大，从而引起系统的性能衰减严重。在目前主流的Linux系统上进行测试发现，在一个忙碌的系统上，当创建的进程或者线程接近一千的时候，创建线程或者进程的速度明显的越来越慢。因此，上述的分布式数据库节点，基于多线程的架构仍然存在如下几个方面的问题：一是一个连接出现的异常会影响到本节点其它所有连接；二是一个节点的连接数只能支持到上千，很难支持到上万连接；三是多线程架构无法控制用户使用的资源，尤其是CPU资源，因此有必要针对此类分布式数据库系统的架构做进一步的研究和改进。

发明内容

[0003] 本发明的目的之一在于针对上述不足，提供一种基于共享内存和多进程的分布式数据库架构及实现方法，以期望解决现有技术中同类数据库一个连接出现的异常会影响到本节点其它所有连接，节点支持的最大连接数最大有限等技术问题。

[0004] 为解决上述的技术问题，本发明采用以下技术方案：

[0005] 本发明一方面提供了一种基于共享内存和多进程的分布式数据库架构，其包括分布式数据库节点，所述分布式数据库内置系统共享内存单元与系统进程单元；所述系统共享内存单元包括任务堆栈信息模块与共享缓存模块；所述任务堆栈信息模块内置多个进程任务；所述进程任务为系统进程任务信息中的多种用途的系统信息，每个系统信息均对应一个进程任务；以及所有用户连接信息所关联的进程任务；所述进程任务中包括对应的堆

栈信息与寄存器信息;所述共享缓存模块用于保存分布式数据库节点的多个连接,或者各个进程之间需要共享的信息;所述系统进程单元包括连接处理进程、执行进程、分布式事务处理进程与网络连接进程;用于由系统进程单元中的各个进程加载执行所述任务堆栈信息模块中的进程任务,或者在多个进程任务之间切换,其中:所述连接处理进程用于执行客户端发送的查询、插入、更新和删除请求;所述执行进程用于执行连接处理进程派生出的并发执行进程任务;所述分布式事务处理进程与网络连接进程分别用于执行分布式事务处理任务与网络连接任务。

[0006] 作为优选,进一步的技术方案是:所述分布式数据库节点为多个,每个所述分布式数据库节点的内部架构均相同,且每个所述分布式数据库节点的系统共享内存单元均由系统统一分配使用。

[0007] 更进一步的技术方案是:所述系统进程任务信息中,根据不同的用途划分为不同的系统信息,其至少包括锁处理系统信息、资源清理系统信息、网络处理系统信息与其它系统信息。

[0008] 更进一步的技术方案是:共享缓存模块用于保存各个进程之间需要共享的信息包括数据字典、配置参数、查询计划、查询结果、执行状态和存储数据。

[0009] 更进一步的技术方案是:所述共享缓存模块中保存的各个进程之间需要共享的信息,在系统启动时加载,并在系统执行过程中与存储层的信息保持一致。

[0010] 本发明另一方面还提供了一种上述分布式数据库架构的实现方法,所述的方法包括如下步骤:

[0011] 步骤A、系统启动,分配分布式数据库节点中的系统共享内存单元,并进行初始化,使系统共享内存单元中的所有用户连接信息、系统进程任务信息、进程任务中的堆栈信息与寄存器信息、以及共享缓存模块中的信息初始化;此时如内存已经分配,则报告系统以及该启动的错误;

[0012] 步骤B、系统进程单元中的各个进程从任务堆栈信息模块中获取需要执行的进程任务,如没有更多的任务,就执行空任务;当有客户端发起连接请求时,系统进程单元中的连接处理进程接收到并响应的用户请求,连接处理进程为当前用户请求生成一个相应的进程任务,并放入共享内存中的所有用户连接信息中;

[0013] 步骤C、当系统进程单元中的任意一个进程执行任务堆栈信息模块中的任意一个进程任务时,出现阻塞等待,当前进程即放弃执行当前进程任务,并将该进程任务交由后台进程异步等待。

[0014] 作为优选,进一步的技术方案是:所述步骤B中,如用户发起的是客户端连接请求,连接处理进程会创建一个用户连接信息,并将用户连接信息与一个进程任务相关联,然生成的进程任务放入所有用户连接信息中;如用户发起的是客户端查询请求,连接处理进程会将查询请求信息加入到对应进程任务关联的用户连接信息中。

[0015] 更进一步的技术方案是:所述的方法还包括步骤D、所述系统进程单元中的各进程循环的从任务堆栈信息模块中获取进程任务并执行;当执行到有用户信息的进程任务时,进一步查看是否为用户连接请求,如是则由连接处理进程执行创建连接的相关流程并创建用户连接信息;如是用户查询信息,则执行进程会从用户信息上获取用户的查询语句,并执行相关查询。

[0016] 更进一步的技术方案是:所述的方法还包括步骤E、系统关闭时,先关闭所有的处理进程,然后再由一个进程释放所有分布式数据库节点的系统共享内存单元的共享内存资源。

[0017] 与现有技术相比,本发明的有益效果之一是:通过在分布式数据库节点使用系统共享内存单元,使得在该分布式数据库架构中用户的连接数不与进程或者线程存在对应关系,整个节点的进程或者线程数都不会因为用户连接数的增加而增加,从而有效避免因瞬时用户连接数过多而导致系统响应速度变慢,从而使系统性能不会因此而受到影响,同时本发明所提供的一种基于共享内存和多进程的分布式数据库架构简单易行,尤其适于作为电子商务等瞬时连接数较大的分布式数据库使用。

附图说明

[0018] 图1为用于说明本发明一个实施例的分布式数据库节点架构示意框图;

[0019] 图2为用于说明本发明另一个实施例的系统架构框图。

具体实施方式

[0020] 下面结合附图对本发明作进一步阐述。

[0021] 为解决上述提到的同类分布式数据库的不足,发明人经过研究与设计,提出了一种分布式数据库节点基于共享内存和多进程的架构,用来支持高并发,该架构主要利用共享内存技术来解决缓存无法共享的问题,利用多进程技术来解决一个连接的异常导致很多连接无法使用的问题。分布式数据库中所有节点都采用相同的架构,不需要独立设计。该技术方案基于共享内存和多进程的分布式数据库节点的总体架构如下图表1所示:

[0022] 参考图1所示,本发明的一个实施例是一种基于共享内存和多进程的分布式数据库架构,其包括分布式数据库节点,多个分布式数据库内置系统共享内存单元与系统进程单元;其中:

[0023] 上述系统共享内存单元包括任务堆栈信息模块与共享缓存模块;其中,任务堆栈信息模块内置多个进程任务;前述的进程任务为系统进程任务信息中的多种用途的系统信息,每个系统信息均对应一个进程任务;以及所有用户连接信息所关联的进程任务;并且,进程任务中包括对应的堆栈信息与寄存器信息;前述共享缓存模块用于保存分布式数据库节点的多个连接,或者各个进程之间需要共享的信息;前述的共享信息主要包括数据字典、配置参数、查询计划、查询结果、执行状态和存储数据;确切的说:

[0024] 上述用户连接信息为给用户的连接请求创建一块共享内存,用来保存用户连接信息,节点上所有的用户连接信息统一管理,每个连接信息会对应的创建一个进程任务,进程任务中记录每个进程当前执行的堆栈和寄存器等相关信息。系统进程任务信息为每个系统进程必须要有对应的一些信息,为了和用户连接信息区分,将这些信息根据不同的用途划分成不同的系统信息。每个系统信息也要对应一个进程任务,只有有进程任务的信息才会被进程执行。

[0025] 任务堆栈信息模块包含了需要执行的任务,每个需要执行的任务必须要有一个进程任务才能被执行。这些进程任务中主要保存了对应的堆栈信息,寄存器等进程执行状态信息。进程任务可以在任何进程上执行,进程可以加载进程任务,也可以在多个进程任务之

前切换。这些进程任务中有一个非常特殊的空任务，该任务任何操作都不执行，就直接进入睡眠状态，该任务主要用于系统处于空闲状态时使用。

[0026] 优选的是，上述共享缓存模块中保存的各个进程之间需要共享的信息，在系统启动时加载，并在系统执行过程中与存储层的信息保持一致，以便各进程执行的时候取到最新的一致的结果；

[0027] 在上述的系统进程任务信息中，可根据不同的用途划分为不同的系统信息，即系统进程任务信息至少包括锁处理系统信息、资源清理系统信息、网络处理系统信息与其它系统信息；

[0028] 上述系统进程单元包括连接处理进程、执行进程、分布式事务处理进程与网络连接进程；用于由系统进程单元中的各个进程加载执行所述任务堆栈信息模块中的进程任务，或者在多个进程任务之间切换，其中：连接处理进程用于执行客户端发送的查询、插入、更新和删除请求；执行进程用于执行连接处理进程派生出的并发执行进程任务；分布式事务处理进程与网络连接进程分别用于执行分布式事务处理任务与网络连接任务。

[0029] 确切的说，上述连接处理进程主要执行用户发送的查询、插入、更新和删除等用户请求。连接处理进程会从进程任务中挑选用户连接相关的进程任务执行，如果执行过程中发现进程必须进入到等待状态，进程会自动放弃该任务，并从进程任务中挑选其它可执行的任务继续执行。

[0030] 执行进程专门用于执行连接处理进程派生出的并发执行进程任务，这些任务的优先级往往会比较高，所以将这些任务用单独的进程来执行，以防因用户连接过多导致的用户查询请求响应过慢的问题。

[0031] 除此之外，上述的分布式事务和网络连接等处理进程统称为其它系统进程。系统任务应该得到及时的处理，因此将所有的系统任务用独立的进程执行，以免因系统任务响应不及时导致的性能问题。

[0032] 在本实施例中，数据库节点架构中用户的连接数不会和进程或者线程有对应关系，整个节点的进程或者线程数都不会因为用户连接数的增加而增加。极端情况下，该架构只需要启动一个连接处理进程，即可支持上千连接，只不过每个连接的响应时间会受到一些影响，仅此而已。但传统的多进程或者多线程架构都无法做到这一点，传统架构启动的进程或者线程数都会随着用户连接数的增加而增加。且据发明人试验，架构的一个连接只是占用一块10MB大小内存，以现在常见的机器配置估计，单节点超过10GB内存时就可以支持上万连接。尤其是大部分连接都只是创建了连接，而并没有实际数据库查询或操作时，该架构的优势更加明显。

[0033] 结合图2所示，在本发明的一个典型应用实例中，将上述架构的分布式数据库节点设计为多个，且每个所述分布式数据库节点的内部架构均相同，且每个所述分布式数据库节点的系统共享内存单元均由系统统一分配使用。

[0034] 本发明的另一个实施例是上述实施例中的基于共享内存和多进程的分布式数据库架构的实现方法，该方法包括如下步骤：

[0035] 步骤S1、系统启动。系统启动后，首先初始化所有的全局变量，然后分配分布式数据库节点中的系统共享内存单元，即分配系统启动所需要的共享内存，并进行初始化，主要初始化系统共享内存单元中的所有用户连接信息、系统进程任务信息、进程任务中的堆栈

信息与寄存器信息、以及共享缓存模块中的信息;在启动系统所需的共享内存前,首先检查共享内存状态,如内存已经分配,则报告系统以及该启动的错误;

[0036] 系统初始化完成以后,启动用户配置参数指定的相关进程;

[0037] 步骤S2、系统进程单元中的各个进程从任务堆栈信息模块中获取需要执行的进程任务,如没有更多的任务,就执行空任务;

[0038] 当有客户端发起连接请求时,系统进程单元中的连接处理进程接收到并响应的用户请求,连接处理进程为当前用户请求生成一个相应的进程任务,并放入共享内存中的所有用户连接信息中;

[0039] 在本步骤中,如用户发起的是客户端连接请求,连接处理进程会创建一个用户连接信息,并将用户连接信息与一个进程任务相关联,然生成的进程任务放入所有用户连接信息中;如用户发起的是客户端查询请求,连接处理进程会将查询请求信息加入到对应进程任务关联的用户连接信息中;

[0040] 步骤S3、当系统进程单元中的任意一个进程执行任务堆栈信息模块中的任意一个进程任务时,出现阻塞等待(例如需要等待接收信息,或者需要等待I/O 完成),当前进程即放弃执行当前进程任务,并将该进程任务交由后台进程异步等待;

[0041] 在本实施例所述的方法中,亦可继续执行如下步骤:

[0042] 步骤S4、所述系统进程单元中的各进程循环的从任务堆栈信息模块中获取进程任务并执行;当执行到有用户信息的进程任务时,进一步查看是否为用户连接请求,如是则由连接处理进程执行创建连接的相关流程并创建用户连接信息;如是用户查询信息,则执行进程会从用户信息上获取用户的查询语句,并执行相关查询。如果该查询代价较高,并可以并发执行,则会创建有相互依赖关系的进程任务,从而触发执行进程的执行。

[0043] 步骤S5、系统关闭时,先关闭所有的处理进程,然后再由一个进程释放所有分布式数据库节点的系统共享内存单元的共享内存资源。

[0044] 基于本发明上述的实施例,本发明的技术方案还具有如下特点:

[0045] 1)利用共享内存技术和多进程任务处理机制。

[0046] 2)进程执行任务化。

[0047] 3)用户连接用共享内存管理。

[0048] 4)共享内存作为最主要缓存内存。

[0049] 5)用户不执行语句的时候不占用进程资源。

[0050] 6)任何进程都可以等价的处理任何用户连接。

[0051] 7)多个进程共享一份缓存数据。

[0052] 8)可以通过配置进程数有效控制占用的CPU资源。

[0053] 本发明上述优选的一个实施例在实际使用中,如图2所示,上述分布式数据库节点作为分布式数据库的节点,通过系统应用程序接收并执行来自于互联网的用户请求,并采用上述的方式执行,且各个分布式数据库节点的共享内存由系统统一调用与启动。

[0054] 除上述以外,还需要说明的是在本说明书中所谈到的“一个实施例”、“另一个实施例”、“实施例”等,指的是结合该实施例描述的具体特征、结构或者特点包括在本申请概括性描述的至少一个实施例中。在说明书中多个地方出现同种表述不是一定指的是同一个实施例。进一步来说,结合任一实施例描述一个具体特征、结构或者特点时,所要主张的是结

合其他实施例来实现这种特征、结构或者特点也落在本发明的范围内。

[0055] 尽管这里参照本发明的多个解释性实施例对本发明进行了描述，但是，应该理解，本领域技术人员可以设计出很多其他的修改和实施方式，这些修改和实施方式将落在本申请公开的原则范围和精神之内。更具体地说，在本申请公开、附图和权利要求的范围内，可以对主题组合布局的组成部件和/或布局进行多种变型和改进。除了对组成部件和/或布局进行的变型和改进外，对于本领域技术人员来说，其他的用途也将是明显的。

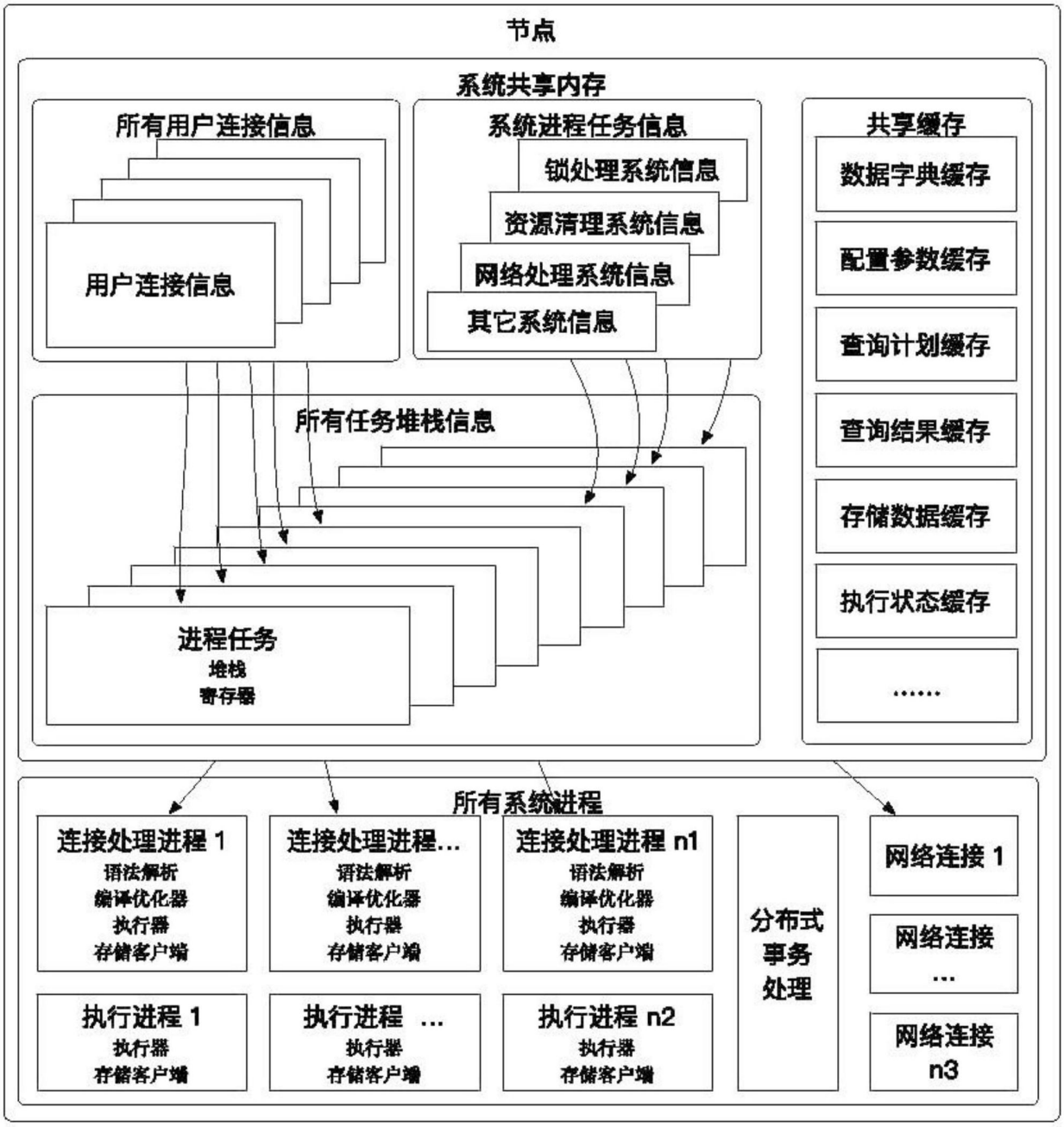

节点
系统共享内存
所有用户连接信息
用户连接信息
系统进程任务信息
锁处理系统信息
资源清理系统信息
网络处理系统信息
其它系统信息
共享缓存
数据字典缓存
配置参数缓存
查询计划缓存
查询结果缓存
存储数据缓存
执行状态缓存
.....
所有任务堆栈信息
进程任务
堆栈
寄存器
所有系统进程
连接处理进程 1
语法解析
编译优化器
执行器
存储客户端
连接处理进程...
语法解析
编译优化器
执行器
存储客户端
连接处理进程 n1
语法解析
编译优化器
执行器
存储客户端
分布式事务处理
网络连接 1
网络连接 ...
网络连接 n3
执行进程 1
执行器
存储客户端
执行进程 ...
执行器
存储客户端
执行进程 n2
执行器
存储客户端

图1

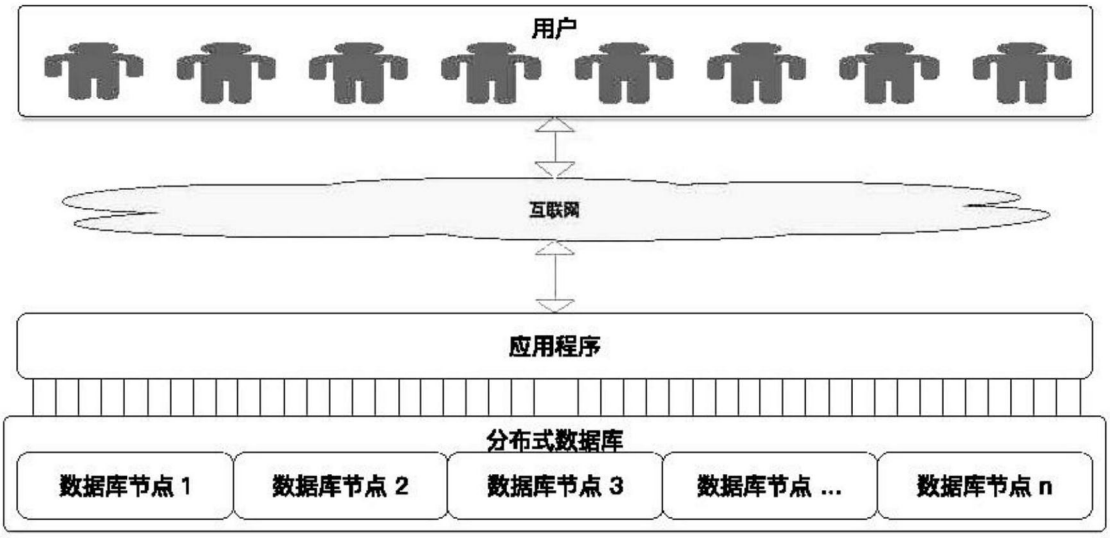


图2