

一种业务处理方法及装置

本申请要求 2016 年 04 月 11 日递交的申请号为 201610219408.7、发明名称为“一种业务处理方法及装置”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

5 技术领域

本申请涉及计算机技术领域，尤其涉及一种业务处理方法及装置。

背景技术

在线系统（如：网站）的后台通常包含多种业务系统，不同的业务系统具有不同的业务功能，从而，在线系统在这些业务系统的支持下，可以为用户提供丰富的业务服务。

目前，在实际应用场景下，由用户通过诸如浏览器、应用（Application，APP）等客户端所发起的业务请求，可能需要多种业务系统共同完成。

现有技术中，对上述的业务请求的处理过程具体为：按照业务流程的顺序，先对该业务请求进行处理的业务系统会生成相应的处理结果（也称为中间结果），返回给客户端，客户端根据中间结果，重定向至业务流程中的下一业务系统，并发起请求（也称为中间请求），使得下一业务系统进行后续处理，以此类推，直到完成整个业务流程为止。

例如：假设客户端所发出的某业务请求需要业务系统 A 和 B 共同完成。按照业务流程，该业务请求首先由业务系统 A 进行处理，生成中间结果 a，此时，业务系统 A 会将中间结果 a 返回给客户端，该客户端基于中间结果 a 重定向至业务系统 B，并进一步向业务系统 B 发出请求，从而，业务系统 B 会对该请求处理，生成业务结果 b，并返回给客户端。

但是，在现有技术的方式中，在线系统内各业务系统与客户端之间通过互联网进行交互，而互联网的网络环境的稳定性较差，可能由于网络环境的影响，产生网络延时，导致客户端根据中间业务结果向下一业务系统发送中间请求的过程需要耗费较长的时间；此外，上述的方式中，业务系统只会在接收到了客户端的请求后，才进行相应的处理，而在线系统通常会面对大量的用户访问，造成各业务系统的工作负荷较高，导致请求在业务系统的处理队列中产生等待延时。显然，在请求的传输阶段和处理阶段都可能存在一定的延时，这必然会造成客户端的等待时间较长，使得业务处理过程的时效性较差。

发明内容

本申请实施例提供一种业务处理方法及装置，用以解决现有的业务处理过程的时效性较差的问题。

5 本申请实施例提供一种业务处理方法，包括：

具有第一业务功能的第一服务器，接收客户端发送的第一业务请求；

根据所述第一业务请求，生成第一业务结果；

10 将所述第一业务结果发送给所述客户端、以及具有第二业务功能的第二服务器，以使得所述第二服务器根据所述第一业务结果直接生成第二业务结果，并当所述第二服务器接收到客户端根据所述第一业务结果生成的第二业务请求后，将已生成的所述第二业务结果发送给客户端。

本申请实施例还提供的一种业务处理方法，包括：

15 具有第二业务功能的第二服务器接收具有第一业务功能的第一服务器发送的第一业务结果，其中，所述第一业务结果是所述第一服务器针对客户端发送的第一业务请求进行处理后生成的；

根据所述第一业务结果生成第二业务结果。

本申请实施例另提供一种业务处理方法，包括：

20 具有第二业务功能的第二服务器接收具有第一业务功能的第一服务器发送的第一业务结果，其中，所述第一业务结果是所述第一服务器针对客户端发送的第一业务请求进行处理后生成的；

根据所述第一业务结果生成第二业务结果；

根据所述第二业务结果渲染并构建业务界面；

当所述第二服务器接收到所述客户端发送的第二业务请求后，将已渲染并构建的所述业务界面发送给所述客户端进行显示。

25 本申请实施例提供一种业务处理装置，包括：

接收模块，用于接收客户端发送的第一业务请求；

处理模块，用于根据所述第一业务请求，生成第一业务结果；

30 发送模块，用于将所述第一业务结果发送给所述客户端、以及具有第二业务功能的第二服务器，以使得所述第二服务器根据所述第一业务结果直接生成第二业务结果，并当所述第二服务器接收到客户端根据所述第一业务结果生成的第二业务请求后，将已生成

的所述第二业务结果发送给客户端。

本申请实施例另提供一种业务处理装置，包括：

接收模块，用于接收具有第一业务功能的第一服务器发送的第一业务结果，其中，所述第一业务结果是所述第一服务器针对客户端发送的第一业务请求进行处理后生成的；

处理模块，用于根据所述第一业务结果生成第二业务结果。

本申请实施例另提供一种业务处理装置，包括：

接收模块，用于接收具有第一业务功能的第一服务器发送的第一业务结果，其中，所述第一业务结果是所述第一服务器针对客户端发送的第一业务请求进行处理后生成的；

生成模块，用于根据所述第一业务结果生成第二业务结果；

渲染处理模块，用于根据所述第二业务结果渲染并构建业务界面；

反馈模块，用于当所述接收模块接收到所述客户端发送的第二业务请求后，将已渲染并构建的所述业务界面发送给所述客户端进行显示。

本申请实施例提供一种业务处理方法及装置，通过该方法，第一服务器接收到了客户端所发出的第一业务请求后，将对该第一业务请求进行处理，并生成相应的第一业务结果，此时，第一服务器除了将第一业务结果返回给客户端之外，还会按照业务流程，将该第一业务结果发送给第二服务器，这样一来，第二服务器便可以及时对该第一业务结果进行处理，进一步生成第二业务结果。正是采用了这样的方式，使得第二服务器在接收到来自客户端所发送的第二业务请求之前，便可以生成第二业务结果，当第二服务器接收到了客户端实际发送的第二请求后，便可以将已经生成的第二业务结果直接返回给客户端。显然，相较于现有技术中的方式而言，第二服务器在接收到客户端的第二业务请求之前，就可以根据第一业务结果预先生成第二业务结果，从而有效节省了客户端的等待时长，也提升了处理业务请求的时效性。

附图说明

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

图 1 为本申请实施例提供的基于第一服务器侧的业务处理过程；

图 2 为本申请实施例提供的应用实例中的业务处理过程的示意图；

图 3a 为本申请实施例提供的基于第二服务器侧的业务处理过程；

图 3b 为本申请实施例提供的另一种基于第二服务器侧的业务处理过程；

图 4 为本申请实施例提供的基于第一服务器侧的业务处理装置结构示意图；

图 5 为本申请实施例提供的基于第二服务器侧的业务处理装置结构示意图；

5 图 6 为本申请实施例提供的另一种基于第二服务器侧的业务处理装置结构示意图。

具体实施方式

为使本申请的目的、技术方案和优点更加清楚，下面将结合本申请具体实施例及相应的附图对本申请技术方案进行清楚、完整地描述。显然，所描述的实施例仅是本申请
10 一部分实施例，而不是全部的实施例。基于本申请中的实施例，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，都属于本申请保护的范围。

正如前述，用户通过客户端从在线系统中获取需要由多个业务系统共同完成的业务服务时，客户端端通常需要接收不同业务系统所返回的中间业务结果，并基于中间业务结果进行重定向后，向下一业务系统发出请求，直到完成完整的业务流程为止，但在该
15 过程中，客户端在向业务系统发送请求时，容易受到网络环境的影响，产生传输延时，且业务系统只能够在接收到客户端所发出的业务请求后，才会进行处理，在业务系统工作负荷较高的情况下，业务系统处理该业务请求时也会产生一定的处理延时，这样一来，对于客户端而言，将受到传输延时和处理延时的双重影响，使得客户端的等待耗时较长，显然，这会对业务请求的时效性造成严重影响。

20 基于此，就需要一种能够减少客户端等待时间的业务处理方式，也即，在本申请实施例中，提供一种业务处理方法，如图 1 所示。

需要说明的是，在本申请实施例中，由于用户通过客户端，从在线系统内所获取的业务服务往往需要多个业务系统共同完成，故客户端会按照业务流程，针对不同的业务系统会发起不同的业务请求，但可以认为，客户端所发起的不同的业务请求，是为了完
25 成同一业务。

此外，在本申请后续内容中的所提及的第一服务器和第二服务器，具体可以是在业务流程中相邻的两个业务服务器，例如：假设现有 3 个业务服务器 A、B、C，某种业务服务需要按照业务服务器 A 至 B 再至 C 的顺序完成，那么，针对业务服务器 A、B 而言，第一服务器就可以是业务服务器 A，第二服务器就可以是业务服务器 B；针对业务服务
30 器 B、C 而言，第一服务器就可以是业务服务器 B，第二服务器就可以是业务服务器 C。

换言之，本申请中的第一服务器和第二服务器并非只限于 2 个服务器的场景。

下面将对本申请中的业务处理方法进行说明，具体而言，如图 1 所示的业务处理过程包括如下步骤：

步骤 S101，具有第一业务功能的第一服务器，接收客户端发送的第一业务请求。

- 5 在实际应用场景下，当用户使用客户端获取业务服务时，通常需要向在线系统内相应的业务服务器发出业务请求。例如：用户使用客户端，针对某商品进行下单操作，此时，客户端就会向下单服务器发送下单请求。在该示例中，下单服务器就是第一服务器，下单请求就是第一业务请求。

- 10 其中，第一服务器，可以是在线系统后台的任一业务服务器。第一业务请求，可以由第一服务器所接收到的业务请求，该第一业务请求既可以由客户端根据用户的操作指令所发出的业务请求，也可以是客户端接收到了业务流程中的前一业务服务器所反馈的中间业务结果后所发出的业务请求。当然，这里并不构成对本申请的限定。

所述的客户端，可以是运行在终端中的浏览器、APP 应用等，这里不作具体限定。

步骤 S102，根据所述第一业务请求，生成第一业务结果。

- 15 当第一服务器接收到了第一业务请求后，便会针对该第一业务请求进行处理，从而生成相应的业务结果，即第一业务结果，这里的第一业务结果通常是一种中间业务结果。

当然，在本申请实施例中的一种方式下，第一业务请求中往往会携带客户端的标识信息，那么，第一服务器所生成的第一业务结果中也会包含客户端的标识信息。

- 20 步骤 S103，将所述第一业务结果发送给所述客户端、以及具有第二业务功能的第二服务器，以使得所述第二服务器根据所述第一业务结果直接生成第二业务结果，并当该第二服务器接收到客户端根据所述第一业务结果生成的第二业务请求后，将已生成的所述第二业务结果发送给客户端。

- 25 所述的第二服务器，可以是在业务流程中处理顺序位于第一服务器之后的业务服务器，那么，第一服务器将第一业务结果发送给客户端之后，客户端便会根据该第一业务结果重定向至第二服务器，以便向第二服务器发出中间业务请求（也即，第二业务请求）。

- 30 考虑到实际应用中，客户端向第二服务器发送第二业务请求的过程可能会受到网络环境的影响而产生延时，第二服务器在接收到第二业务请求并进行处理时，也可能产生处理延时，显然，这两种情况均会造成客户端的等待时间较长，因此，为了减少客户端的等待时长，在本申请实施例中，第一服务器除了将第一业务结果发送给客户端之外，还会直接向第二服务器发送该第一业务结果，使得第二服务器可根据第一业务结果，生

成第二业务结果。

作为本申请实施例中的一种方式，第一服务器在向第二服务器发送第一业务结果时，可以生成相应的请求，并将该第一业务结果携带在该请求中，以“模拟”客户端，使得第二服务器根据请求中所携带的第一业务结果生成相应的第二业务结果。当然，这并不构成对本申请的限定。

需要说明的是，第一服务器和第二服务器均属于在线系统后台的业务服务器，其进行传输时将使用在线系统内部网络，这就保证了第一服务器和第二服务器之间进行传输的过程并不会受到网络环境的影响，也就不会产生网络延时。换言之，第一服务器向第二服务器发送了第一业务结果后，第二服务器将立即接收到该第一业务结果，从而可对该第一业务结果进行处理。即使在第二业务服务器的处理负荷较大的情况下，采用上述的方式，也可以节省客户端向第二服务器发送第二请求所耗费的时间。

当然，本申请实施例中的上述内容并不限于应用在两个服务器的场景下，在实际应用中，共同完成某一业务服务的多个业务服务器也将按照上述的方式，在生成了相应的中间业务结果后，除了将该中间业务结果发送给客户端之外，还会按照业务流程中的处理顺序，将中间业务结果发送给下一业务系统，使得下一业务系统提前对中间业务结果进行处理，生成后续的中间业务结果，以此类推，直到完成完整的业务流程为止。

通过上述步骤，第一服务器接收到了客户端所发出的第一业务请求后，将对该第一业务请求进行处理，并生成相应的第一业务结果，此时，第一服务器除了将第一业务结果返回给客户端之外，还会按照业务流程，将该第一业务结果发送给第二服务器，这样一来，第二服务器便可以及时对该第一业务结果进行处理，进一步生成第二业务结果。正是采用了这样的方式，使得第二服务器在接收到来自客户端所发送的第二业务请求之前，便可以生成第二业务结果，当第二服务器接收到了客户端实际发送的第二请求后，便可以将已经生成的第二业务结果直接返回给客户端。显然，相较于现有技术中的方式而言，第二服务器在接收到客户端的第二业务请求之前，就可以根据第一业务结果预先生成第二业务结果，从而有效节省了客户端的等待时长，也提升了处理业务请求的时效性。

为了清楚的阐述本申请的上述业务处理方法，现以实际的支付业务场景进行详细说明：

假设某用户使用某商品网站的客户端，在该商品网站中购买某商品。可以认为，用户通过客户端针对某商品的购买操作涉及下单和支付两种业务（即，需要该网站后台的

下单服务器和支付服务器共同完成)。那么,在该场景中,第一业务请求为下单请求,第一服务器包括下单服务器,第一业务结果包括订单信息,第二服务器包括支付服务器。

在此场景下,上述步骤 S101 中,第一服务器接收客户端发送的第一业务请求,具体包括:下单服务器接收客户端发送的下单请求。

- 5 上述步骤 S102 中,根据所述第一业务请求,生成第一业务结果,具体包括:根据所述下单请求生成订单信息。在本申请实施例中的一种方式下,订单信息中可包括:账户信息、商品信息、金额信息等等,这里并不构成对本申请的限定。

上述步骤 S103 中,将所述第一业务结果发送给所述客户端、以及与所述业务请求相关联的第二服务器,具体包括:

- 10 将所述下单信息发送给所述客户端、以及与所述下单请求相关联的支付服务器。这里与下单请求相关联的支付服务器,可以根据预先定义的业务流程确定出来,这里并不构成对本申请的限定。

实际应用中,如图 2 所示,示出了上述场景下的业务流程:

步骤 S201: 用户向客户端发出下单操作指令。

- 15 步骤 S202: 客户端根据下单操作指令,向下单服务器发送下单请求。

步骤 S203: 下单服务器针对接收到的下单请求进行处理,生成订单信息。

步骤 S204: 下单服务器将订单信息发送给支付服务器。以使得支付服务器执行步骤 S206。

步骤 S205: 下单服务器将订单信息发送给客户端。以使得客户端执行步骤 S207。

- 20 步骤 S206: 支付服务器根据订单信息,生成支付页面。

步骤 S207: 客户端根据订单信息,重定向至支付服务器,并向支付服务器发出支付页面请求。

步骤 S208: 支付服务器向客户端返回支付页面。

- 25 从上述内容及示例中可见,在实际支付场景中,一旦订单服务器生成了订单信息后,将立即发送给支付服务器进行处理,以生成相应的支付页面,那么,当客户端向支付服务器发出支付页面请求后,支付服务器可以立即向客户端返回已经生成的支付页面。显然,这样的方式将有效减少客户端等待生成支付页面的时间,进一步也使得用户可更快地浏览到支付页面,以便在支付页面中完成支付业务。

- 30 以上内容是基于第一服务器侧的业务处理方法,对于第二服务器而言,本申请还提

供一种业务处理方法，如图 3a 所示。

图 3a 所示的业务处理方法具体包括以下步骤：

步骤 S301：具有第二业务功能的第二服务器接收具有第一业务功能的第一服务器发送的第一业务结果。

5 其中，第一业务结果是第一服务器针对客户端发送的第一业务请求进行处理后生成的。

与前述方法相类似，第一服务器和第二服务器是在线系统后台中具有不同业务功能的业务服务器。

步骤 S302：根据所述第一业务结果生成第二业务结果。

10 第一业务结果作为一种中间业务结果，由第一服务器分别发送给客户端和第二服务器，第二服务器便会针对该第一业务结果进行处理，生成该第二服务器的处理结果（即，第二业务结果）。具体过程与前述内容相类似，这里不再过多赘述。

与现有技术不同的是，本申请实施例中的第二服务器无需等待客户端发送的业务请求（即，前述的第二业务请求），而是接收由第一服务器发送的第一业务结果，并基于
15 第一业务结果生成第二业务结果（例如：前述示例中，支付服务器基于订单信息，生成包含订单信息的支付界面），从而可以在实际接收到客户端发送的第二业务请求时，直接将已经生成的第二业务结果立即反馈给客户端，减少客户端的等待时间。

基于此，如图 3a 所示的方法，还可以包括：所述第二服务器接收所述客户端发送的第二业务请求，在接收到所述第二业务请求后，将已生成的第二业务结果反馈给所述客
20 户端。

在本申请实施例中的一种方式下，第二业务请求中通常会包含客户端的标识信息，那么，第二服务器也就可以根据标识信息，确定出与该第二业务请求相匹配的第二业务结果。

相类似地，在实际支付场景中，第二服务器可以为支付服务器，第一服务器可以为
25 下单服务器，第一业务结果可以为订单信息，第二业务结果可以为包含订单信息的支付页面，第二业务请求可以为支付请求。

那么，在上述步骤中，第二服务器接收第一服务器发送的第一业务结果，具体包括：支付服务器接收下单服务器发送的订单信息。

根据所述第一业务结果生成第二业务结果，具体包括：根据所述订单信息生成包含
30 有该订单信息的支付页面。

在接收到所述第二业务请求后，将已生成的第二业务结果反馈给所述客户端，具体包括：在接收到所述支付请求后，将已生成的支付页面反馈给所述客户端。

具体的应用场景如图 2 所示，这里不再过多赘述。

5 在上述如图 3a 所示的业务处理方法的基础上，第二服务器根据第一业务结果所生成的第二业务结果通常是以一种业务界面的方式呈现，若客户端向第二服务器发出了后续的业务请求后，该业务界面将显示于客户端中，以便于用户通过该业务界面进行业务确认的相关操作。正如在前述示例中，支付服务器（即，第二服务器）接收到下单服务器（即，第一服务器）所发送的订单信息（即，第一业务结果）后，将会根据该订单信息
10 生成相应的支付信息（即，第二业务结果），并进一步依据支付信息渲染构建相应的支付页面，如果客户端向该支付服务器发出支付请求，那么，支付服务器就会将该支付页面发送给客户端，以便用户可以通过该支付页面直观地获知所购买的商品、所需金额等信息，进行支付的确认操作。

 基于此，在本申请实施例中，还提供一种业务处理方法，如图 3b 所示，该方法具体
15 包括以下步骤：

 步骤 S311：具有第二业务功能的第二服务器接收具有第一业务功能的第一服务器发送的第一业务结果。

 其中，所述第一业务结果是所述第一服务器针对客户端发送的第一业务请求进行处理后生成的。

20 步骤 S312：根据所述第一业务结果生成第二业务结果。

 步骤 S313：根据所述第二业务结果渲染并构建业务界面。

 在本申请实施例中，第二服务器可以根据预先定义的界面模板，将第二业务结果进行渲染，从而构建出业务界面，这里并不构成对本申请的限定，故不作具体赘述。

 步骤 S314：当所述第二服务器接收到所述客户端发送的第二业务请求后，将已渲染
25 并构建的所述业务界面发送给所述客户端进行显示。

 如图 3b 所示的方法中具体的执行过程、应用场景与前述内容相类似，故这里不再过多赘述。

 以上为本申请提供的业务处理方法的几种实施例，基于同样的思路，本申请还提供了业务处理装置的实施例，如图 4 所示。图 4 中的业务处理装置，所述装置可以设置于
30

第一服务器中，该装置包括：

接收模块 401，用于接收客户端发送的第一业务请求；

处理模块 402，用于根据所述第一业务请求，生成第一业务结果；

5 发送模块 403，用于将所述第一业务结果发送给所述客户端、以及具有第二业务功能的第二服务器，以使得所述第二服务器根据所述第一业务结果直接生成第二业务结果，并当所述第二服务器接收到客户端根据所述第一业务结果生成的第二业务请求后，将已生成的所述第二业务结果发送给客户端。

在本申请实施例中的一种场景下，第一服务器可以是下单服务器，第一业务请求可包括下单请求，第一业务结果可包括订单信息，所述第二服务器可包括支付服务器。

10 基于此，接收模块 401，具体用于接收客户端端发送的下单请求；

处理模块 402，具体用于根据所述下单请求生成订单信息；

发送模块 403，具体用于将下单信息发送给所述客户端、以及与所述下单请求相关联的支付服务器。

15 在第二服务器侧，本申请还提供一种业务处理装置，如图 5 所示，该装置可以设置在第二服务器中，包括：

接收模块 501，用于接收具有第一业务功能的第一服务器发送的第一业务结果，其中，所述第一业务结果是所述第一服务器针对客户端发送的第一业务请求进行处理后生成的；

20 处理模块 502，用于根据所述第一业务结果生成第二业务结果。

在本申请实施例中的一种方式下，接收模块 501 还用于接收所述客户端发送的第二业务请求。基于此，该装置还可以包括：反馈模块 503，用于在所述接收模块接收到所述第二业务请求后，将已生成的第二业务结果反馈给所述客户端。

25 在本申请实施例中的一种场景下，第一服务器可包括下单服务器，第二服务器可包括支付服务器，第一业务结果可包括订单信息，第二业务结果可以包括包含有订单信息的支付页面，第二业务请求可包括支付请求。

基于此，接收模块 501，具体用于接收下单服务器发送的订单信息。

处理模块 502，具体用于根据所述订单信息生成包含有该订单信息的支付页面。

30 反馈模块 503，用于在所述接收模块接收到所述支付请求后，将已生成的支付页面反馈给所述客户端。

本申请还提供一种业务处理装置，如图 6 所示，该装置可以设置在第二服务器中，包括：

接收模块 601，用于接收具有第一业务功能的第一服务器发送的第一业务结果，其中，所述第一业务结果是所述第一服务器针对客户端发送的第一业务请求进行处理后生成的。

生成模块 602，用于根据所述第一业务结果生成第二业务结果。

渲染处理模块 603，用于根据所述第二业务结果渲染并构建业务界面。

反馈模块 604，用于当所述接收模块接收到所述客户端发送的第二业务请求后，将已渲染并构建的所述业务界面发送给所述客户端进行显示。

本领域内的技术人员应明白，本发明的实施例可提供为方法、系统、或计算机程序产品。因此，本发明可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且，本发明可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质（包括但不限于磁盘存储器、CD-ROM、光学存储器等）上实施的计算机程序产品的形式。

本发明是参照根据本发明实施例的方法、设备（系统）、和计算机程序产品的流程图和 / 或方框图来描述的。应理解可由计算机程序指令实现流程图和 / 或方框图中的每一流程和 / 或方框、以及流程图和 / 或方框图中的流程和 / 或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器，使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和 / 或方框图一个方框或多个方框中指定的功能的装置。

这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中，使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品，该指令装置实现在流程图一个流程或多个流程和 / 或方框图一个方框或多个方框中指定的功能。

这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上，使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理，从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和 / 或方框图一个方框或多个方框中指定的功能的步骤。

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

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。

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。

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。

本领域技术人员应明白，本申请的实施例可提供为方法、系统或计算机程序产品。因此，本申请可采用完全硬件实施例、完全软件实施例或结合软件和硬件方面的实施例的形式。而且，本申请可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质（包括但不限于磁盘存储器、CD-ROM、光学存储器等）上实施的计算机程序产品的形式。

以上所述仅为本申请的实施例而已，并不用于限制本申请。对于本领域技术人员来说，本申请可以有各种更改和变化。凡在本申请的精神和原理之内所作的任何修改、等同替换、改进等，均应包含在本申请的权利要求范围之内。