



(12) 发明专利申请

(10) 申请公布号 CN 105702257 A

(43) 申请公布日 2016. 06. 22

(21) 申请号 201510497684. 5

(22) 申请日 2015. 08. 12

(71) 申请人 乐视致新电子科技(天津)有限公司

地址 300467 天津市滨海新区天津生态城
动漫中路 126 号动漫大厦 B1 区二层
201-427

(72) 发明人 王育军

(74) 专利代理机构 北京润泽恒知识产权代理有
限公司 11319

代理人 苏培华

(51) Int. Cl.

G10L 15/28(2013. 01)

G10L 15/34(2013. 01)

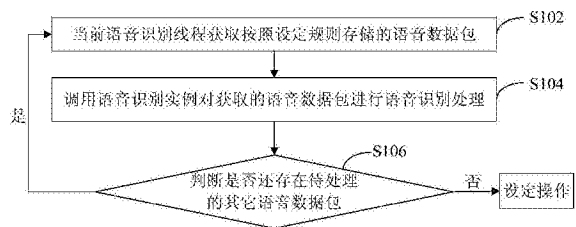
权利要求书3页 说明书12页 附图5页

(54) 发明名称

语音处理方法与装置

(57) 摘要

本发明实施例提供一种语音处理方法和装置,其中,所述方法包括:当前语音识别线程获取按照设定规则存储的语音数据包;调用语音识别实例对获取的所述语音数据包进行语音识别处理;在对获取的所述语音数据包处理完成后、判断是否还存在待处理的其它语音数据包;若存在,则返回所述当前语音识别线程获取按照设定规则存储的语音数据包的步骤,继续对所述其它语音数据包进行语音识别处理,其中,获取的所述语音数据包与所述其它语音数据包属于不同的语音连接。通过本发明实施例,能够使语音识别服务器资源被充分利用于处理语音请求。



1. 一种语音处理方法,其特征在于,包括:

当前语音识别线程获取按照设定规则存储的语音数据包;

调用语音识别实例对获取的所述语音数据包进行语音识别处理;

在对获取的所述语音数据包处理完成后、判断是否还存在待处理的其它语音数据包;

若存在,则返回所述当前语音识别线程获取按照设定规则存储的语音数据包的步骤,继续对所述其它语音数据包进行语音识别处理,其中,获取的所述语音数据包与所述其它语音数据包属于不同的语音连接。

2. 根据权利要求1所述的方法,其特征在于,所述调用语音识别实例对获取的所述语音数据包进行语音识别处理的步骤包括:

调用与所述语音数据包对应的语音识别实例;

采用调用的所述语音识别实例对所述语音数据包进行语音识别处理,并将处理完成的所述语音数据包压入处理完毕的语音数据包存储空间中,以供所述语音数据包所属的语音连接获取。

3. 根据权利要求2所述的方法,其特征在于,其中,所述调用与所述语音数据包对应的语音识别实例的步骤包括:

判断识别实例列表中是否存储有与所述语音数据包所属的语音连接对应的语音识别实例;

若存在,则直接调用与所述语音连接对应的语音识别实例;

若不存在,则从所述语音识别实例列表中调用一个当前在状态为空闲的语音识别实例,并建立调用的所述语音识别实例与所述语音数据包所属的语音连接之间的对应关系,其中,所述空闲的语音识别实例为未与语音连接建立对应关系的语音识别实例。

4. 根据权利要求1所述的方法,其特征在于,按照设定规则存储的语音数据包通过以下方式存储:

所述语音数据包所属的语音连接判断待处理的语音数据包存储空间中、当前正在处理的语音数据包存储空间中、以及处理完毕的语音数据包存储空间中是否存储有本语音连接对应的语音数据包;

若均未存储有本语音连接对应的语音数据包,则将所述语音数据包存储至所述待处理的语音数据包存储空间中。

5. 根据权利要求2所述的方法,其特征在于,在所述将处理完毕的所述语音数据包压入处理完毕的语音数据包队列中的步骤之后,所述方法还包括:

判断所述语音数据包中是否包含有结束标识,其中,所述结束标识用于指示当前语音数据包为所属的语音连接对应的最后一个语音数据包;

若是,则将所述语音数据包对应的语音识别实例的释放,并将释放后的所述语音识别实例的状态更新为空闲。

6. 根据权利要求1所述的方法,其特征在于,在所述当前语音识别线程获取按照设定规则存储的语音数据包的步骤之前,所述方法还包括:

服务器主线程依据服务器包含的物理核数量,分别为每个物理核创建一个语音识别线程。

7. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

当服务器主线程接收到对服务器的外部中断请求时,生成中断标识;

根据所述中断标识,针对所述服务器中已建立的语音连接,继续对所述语音连接对应的语音数据包进行语音识别处理,待所述语音连接对应的全部语音数据包进行语音识别处理后,中断所述语音连接;

针对未与所述服务器建立连接的语音连接,直接取消与所述服务器建立连接的请求,不再建立于所述服务器之间的语音连接。

8. 一种语音处理装置,其特征在于,包括:

获取模块,用于供当前语音识别线程获取按照设定规则存储的语音数据包;

处理模块,用于调用语音识别实例对获取的所述语音数据包进行语音识别处理;

判断模块,用于在对获取的所述语音数据包处理完成后、判断是否还存在待处理的其它语音数据包;

返回模块,用于若存在,则返回所述获取模块继续对所述其它语音数据包进行语音识别处理,其中,获取的所述语音数据包与所述其它语音数据包属于不同的语音连接。

9. 根据权利要求 8 所述的装置,其特征在于,所述处理模块包括:

语音识别实例调用模块,用于调用与所述语音数据包对应的语音识别实例;

语音识别模块,用于采用调用的所述语音识别实例对所述语音数据包进行语音识别处理,并将处理完成的所述语音数据包压入处理完毕的语音数据包存储空间中,以供所述语音数据包所属的语音连接获取。

10. 根据权利要求 9 所述的装置,其特征在于,其中,所述语音识别实例调用模块调用与所述语音数据包对应的语音识别实例时:

判断识别实例列表中是否存储有与所述语音数据包所属的语音连接对应的语音识别实例;

若存在,则直接调用与所述语音连接对应的语音识别实例;

若不存在,则从所述语音识别实例列表中调用一个当前在状态为空闲的语音识别实例,并建立调用的所述语音识别实例与所述语音数据包所属的语音连接之间的对应关系,其中,所述空闲的语音识别实例为未与语音连接建立对应关系的语音识别实例。

11. 根据权利要求 8 所述的装置,其特征在于,按照设定规则存储的语音数据包通过以下方式存储:

所述语音数据包所属的语音连接判断待处理的语音数据包存储空间中、当前正在处理的语音数据包存储空间中、以及处理完毕的语音数据包存储空间中是否存储有本语音连接对应的语音数据包;

若均未存储有本语音连接对应的语音数据包,则将所述语音数据包存储至所述待处理的语音数据包存储空间中。

12. 根据权利要求 9 所述的装置,其特征在于,所述语音处理装置还包括:

标识判断模块,用于在所述语音识别模块将处理完毕的所述语音数据包压入处理完毕的语音数据包队列中之后,判断所述语音数据包中是否包含有结束标识,其中,所述结束标识用于指示当前语音数据包为所属的语音连接对应的最后一个语音数据包;

语音识别实例释放模块,用于若所述标识判断模块的判断结果为是时,则将所述语音数据包对应的语音识别实例的释放,并将释放后的所述语音识别实例的状态更新为空闲。

13. 根据权利要求 8 所述的装置,其特征在于,所述语音处理装置还包括:

创建模块,用于在所述获取模块供当前语音识别线程获取按照设定规则存储的语音数据包之前,供服务器主线程依据服务器包含的物理核数量,分别为每个物理核创建一个语音识别线程。

14. 根据权利要求 8 所述的装置,其特征在于,所述语音处理装置还包括:

生产模块,用于当服务器主线程接收到对服务器的外部中断请求时,生成中断标识;

连接处理模块,用于根据所述中断标识,针对所述服务器中已建立的语音连接,继续对所述语音连接对应的语音数据包进行语音识别处理,待所述语音连接对应的全部语音数据包进行语音识别处理后,中断所述语音连接;

针对未与所述服务器建立连接的语音连接,直接取消与所述服务器建立连接的请求,不再建立于所述服务器之间的语音连接。

语音处理方法与装置

技术领域

[0001] 本发明涉及语音处理技术领域,特别是涉及一种语音处理方法与装置。

背景技术

[0002] 基于互联网的语音识别云服务能够帮助用户实现语音搜索,语音输入,语音对话等。如图 1 所示,在这些服务中,待处理的语音从用户终端,通过互联网被传输给语音识别服务器,语音识别服务器端的识别结果沿着原链路返回给用户终端展示给用户。基于互联网的语音识别服务的用户请求量巨大、且日益增加,使得原本就有限的识别资源变得越发紧张。可见,对于本领域技术人员而言,使语音识别服务器资源能够被充分的利用于语音请求处理,使得语音识别服务器在单位时间内可以识别更多的语音请求,是亟待解决的问题。

[0003] 现有的两种解决方案分别如下:

[0004] 现有的第一种解决方案:

[0005] 如图 2 所示,在语音识别服务中,为识别服务器的每个物理核创建一个语音识别线程。为了实时给用户反馈语音识别结果,并减小用户等待语音识别结果的时间,在用户输入语音的同时,将用户请求识别的语音以 1 秒语音时长为单位,切分成语音数据包发送到语音识别服务器(如图 2 中所示的请求 1 包 1,请求 1 包 2)。在该方案中为来自一个客户端的语音数据包建立一个语音连接,同时为这个语音连接绑定一个语音识别线程和语音识别实例。也就是说,一个语音识别线程仅为一个语音连接所请求的语音数据包服务。

[0006] 而事实上,语音识别线程在处理 1 秒语音时长的语音数据包时,实时率通常在 0.6 至 0.9 之间,即处理 1 秒钟的语音数据包需要 0.6 秒至 0.9 秒即可识别完毕。可见,语音识别线程的识别速度要比用户输入语音的速度快。这将导致,在语音识别线程服务器语音连接所请求的一个语音数据包处理完后,等待该语音连接所请求的下一个语音数据包时,语音识别线程处于闲置状态。本领域技术人员能够明了,语音线程闲置则说明语音识别服务器资源未能得到充分利用,语音识别服务器资源利用率低。

[0007] 现有的第二种解决方案:

[0008] 为语音识别服务器中的每个物理核创建多个语音识别线程,并建立与识别线程相同数量的语音识别实例,即,识别服务器的一个物理核需要为多个语音识别线程提供服务。比如一台识别服务器有 12 个逻辑处理器机物理核,可以创建 48 组语音连接、语音识别线程和语音识别实例。

[0009] 现有的第二种解决方案,当一个语音识别线程当前处理的语音数据包处理完成后,此时该语音处理线程对应的物理核闲置,此时,其他具有待处理的语音数据包的语音处理线程抢用闲置的该物理核。由于各语音线程切换时刻不受控制,往往多个语音线程争抢一个物理核,将造成语音线程间的物理核处理时间轮转,这样,虽然物理核未闲置,但是物理核的部分资源由于语音线程的轮转被消耗,依然未被利用于处理语音请求。可见,现有的第二种简单依靠增加语音识别线程组数的解决方案,依然存在语音识别服务器资源无法被充分用于处理语音请求的问题。

[0010] 可见,现有的处理方案均无法使语音识别服务器资源被充分利用于处理语音请求。

发明内容

[0011] 鉴于上述现有的处理方案,均无法使语音识别服务器资源被充分利用于处理语音请求的问题,提出了本发明以便提供一种克服上述问题或者至少部分地解决上述问题的语音处理方法与装置。

[0012] 为了解决上述问题,本发明公开了一种语音处理方法,包括:当前语音识别线程获取按照设定规则存储的语音数据包;调用语音识别实例对获取的所述语音数据包进行语音识别处理;在对获取的所述语音数据包处理完成后、判断是否还存在待处理的其它语音数据包;若存在,则返回所述当前语音识别线程获取按照设定规则存储的语音数据包的步骤,继续对所述其它语音数据包进行语音识别处理,其中,获取的所述语音数据包与所述其它语音数据包属于不同的语音连接。

[0013] 为了解决上述问题,本发明还公开了一种语音处理装置,包括:获取模块,用于供当前语音识别线程获取按照设定规则存储的语音数据包;处理模块,用于调用语音识别实例对获取的所述语音数据包进行语音识别处理;判断模块,用于在对获取的所述语音数据包处理完成后、判断是否还存在待处理的其它语音数据包;返回模块,用于若存在,则返回所述获取模块继续对所述其它语音数据包进行语音识别处理,其中,获取的所述语音数据包与所述其它语音数据包属于不同的语音连接。

[0014] 本发明实施例提供的语音处理方案,当前语音识别线程并非仅为一个语音连接提供语音处理服务,在当前语音连接对应的语音数据包处理完成后即当前语音识别线程空闲时,当前语音识别线程主动判断是否还存在待处理的其它请求的语音数据包,若存在,则直接获取一个待处理的其他请求的语音数据包进行处理。本发明实施例提供的语音处理方案,一方面,当语音识别线程空闲时主动获取其他待处理的语音数据包进行语音处理,因此,语音识别线程并不存在现有技术中所存在的语音识别线程等待语音连接对应的下一个语音数据包而造成资源浪费的问题。另一方面,本发明实施例中的语音识别线程空闲时是通过主动获取其他待处理的语音数据包,来改变其对应的物理核的空闲状态,以使物理核的资源能够被充分利用。整个过程中都不存在语音识别线程之间的切换,语音识别线程对应的物理核的全部资源都利用于处理语音数据包,因此,能够有效避免现有的处理方案中存在的语音识别线程之间的轮转而消耗物理核资源的问题。再一方面,语音线程之间的轮转是由处理器按照自身设定的时钟周期进行的,即在处理同一个语音数据包时,可能暂时中断一段时间,来处理另外的一个语音识别线程对应的语音数据包,因此,语音识别线程之间的轮转将导致单个语音请求处理速度的下降。而本发明实施例中,并不存在语音线程之间的轮转,因此,能够避免语音识别线程轮转而导致的单个语音请求处理速度下降的问题。

附图说明

[0015] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作一简单地介绍,显而易见地,下面描述中的附图是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根

据这些附图获得其他的附图。

[0016] 图 1 是现有的语音交互云示意图；

[0017] 图 2 是现有的语音识别的交互示意图；

[0018] 图 3 是根据本发明实施例一的一种语音处理方法的步骤流程图；

[0019] 图 4 是根据本发明实施例二的一种语音处理方法的步骤流程图；

[0020] 图 5 示出了采用实施例二中所示的语音处理方法与现有的第一种语音处理方法在处理语音请求时的性能对比图；

[0021] 图 6 示出了执行实施例二中所所述的语音处理方法的语音处理系统启动和中断的流程示意图；

[0022] 图 7 示出了语音处理过程中语音数据包的生命周期示意图；

[0023] 图 8 是根据本发明实施例三的一种语音处理装置的结构框图；

[0024] 图 9 是根据本发明实施例四的一种语音处理装置的结构框图。

具体实施方式

[0025] 为使本发明实施例的目的、技术方案和优点更加清楚，下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

[0026] 实施例一

[0027] 参照图 3，示出了本发明实施例一的一种语音处理方法的步骤流程图。

[0028] 本发明实施例的语音处理方法包括以下步骤：

[0029] 步骤 S102：当前语音识别线程获取按照设定规则存储的语音数据包。

[0030] 存储语音数据包的设定规则可以由本领域技术人员根据实际需求进行设定，本发明实施例对此不作具体限制。例如：设置成语音连接请求处理的前一个语音数据包未处理完成前，不对该语音连接请求的下一个语音数据包进行存储。例如：设置语音连接请求处理的前一个语音数据包未处理完成后、设定时间后，再对该语音连接请求的下一个语音数据包进行存储。

[0031] 其中，存储的各语音数据包可以以任意适当的方式进行存储。例如：在识别服务器中设置一个队列，语音数据包以存储时间的先后顺序依次存储在设置的该队列中。再例如：设置一个存储池，待存储的各语音数据包存储在该存储池中。

[0032] 步骤 S104：调用语音识别实例对获取的语音数据包进行语音识别处理。

[0033] 在对语音进行识别时，语音识别线程为每个已建立成功的语音连接分配一个语音识别实例，每个语音连接处理一个语音请求所请求的全部语音数据包。其中，建立成功的语音连接为已经被分配处理语音请求的连接。语音识别实例中保存有语音请求当前的识别状态和识别历史，通过识别历史可以将当前识别的语音数据包与之前已经处理完成的语音数据包之间的数据进行前后语义的关联，以成功对将当前语音数据包进行处理。

[0034] 步骤 S106：在对获取的语音数据包处理完成后、判断是否还存在待处理的其它语音数据包；当判断结果为存在时，返回执行步骤 S102，当判断结果为不存在时，执行设定操作。

[0035] 本领域技术人员能够明了,服务器通常情况下会同时处理多个语音连接,而每个语音连接对应多个语音数据包,因此,在具体实现过程中,识别服务器中同时会存在多个待处理的语音数据包。

[0036] 本发明实施例中,当前语音识别线程处理完当前语音连接所请求的语音数据包后,主动获取其他待处理的语音数据包进行语音处理,直至全部待处理的语音数据包处理完毕。

[0037] 需要说明的是,设定操作可以由本领域技术人员根据实际需求进行设置。例如:设置成当判断出不存在待处理的语音数据包时,停止并释放语音识别线程。

[0038] 通过本发明实施例提供的语音处理方法,当前语音识别线程并非仅为一个语音连接提供语音处理服务,在当前语音连接对应的语音数据包处理完成后即当前语音识别线程空闲时,当前语音识别线程主动判断是否还存在待处理的其它请求的语音数据包,若存在,则直接获取一个待处理的其他请求的语音数据包进行处理。本发明实施例提供的语音处理方法,一方面,当语音识别线程空闲时主动获取其他待处理的语音数据包进行语音处理,因此,语音识别线程并不存在现有技术中所存在的语音识别线程等待语音连接对应的下一个语音数据包而造成资源浪费的问题。另一方面,本发明实施例中的语音识别线程空闲时是通过主动获取其他待处理的语音数据包,来改变其对应的物理核的空闲状态,以使物理核的资源能够被充分利用。整个过程中都不存在语音识别线程之间的切换,语音识别线程对应的物理核的全部资源都利用于处理语音数据包,因此,能够有效避免现有的处理方法中存在的语音识别线程之间的轮转而消耗物理核资源的问题。再一方面,语音线程之间的轮转是由处理器按照自身设定的时钟周期进行的,即在处理同一个语音数据包时,可能暂时中断一段时间,来处理另外的一个语音识别线程对应的语音数据包,因此,语音识别线程之间的轮转将导致单个语音请求处理速度的下降。而本发明实施例中,并不存在语音线程之间的轮转,因此,能够避免语音识别线程轮转而导致的单个语音请求处理速度下降的问题。

[0039] 实施例二

[0040] 参照图 4,示出了本发明实施例二的一种语音处理方法的步骤流程图。

[0041] 本发明实施例的语音处理方法具体包括以下步骤:

[0042] 步骤 S202:服务器主线程依据服务器包含的物理核数量,分别为每个物理核创建一个语音识别线程。

[0043] 例如:服务器包含 12 个物理核,服务器主线程在语音处理系统启动后,分别为这 12 个物理核创建一个对应的语音识别线程。为了提高处理语音请求的并发度,可以建立多于 12 个的语音连接和语音识别实例。这样,服务器便可以同时处理 12 个以上的语音请求,其中,每个语音连接用于处理一个语音连接所请求处理的全部语音数据包。当一个语音识别线程识别完一个语音数据包后,自动获取另一个待处理的语音数据包,并调用所获取的语音数据包对应的语音识别实例,进行语音识别处理。

[0044] 本发明实施例中,各语音识别线程并行处理语音数据包、且各语音识别线程对语音数据包进行处理时的处理方法相同。因此,本发明实施例中仅以一个语音识别线程对语音数据包进行处理为例进行说明。对于所创建的其他语音识别线程对语音数据包的处理流程参见本发明实施例即可,在此不再赘述。

[0045] 本发明实施例中,由于语音识别线程的个数等于服务器逻辑内核即服务器包含的

物理核的个数,因此,语音识别线程间不会轮流或随意争抢处理器资源。

[0046] 本发明实施例中为每个物理核创建一个语音识别线程,每个语音识别线程可以作为一个语音连接请求建立语音连接,建立成功的语音连接处理其对应的语音连接请求所发送的各语音数据包。

[0047] 步骤 S204:当前语音识别线程获取按照设定规则存储的语音数据包。

[0048] 由于服务器需要同时处理多个语音连接所请求的语音数据包,因此,在服务器中存储有多个待处理的语音数据包。而这些语音数据包在存储至服务器中时,需要遵循设定规则。

[0049] 一种优选的按照设定规则存储的语音数据包的方式如下:

[0050] S1:语音数据包所属的语音连接判断待处理的语音数据包存储空间中、当前正在处理的语音数据包存储空间中、以及处理完毕的语音数据包存储空间中是否存储有本语音连接对应的语音数据包;

[0051] 其中,在服务器中可以预先设置三个存储空间,分别为待处理的语音数据包存储空间、正在处理的语音数据包存储空间以及处理完毕的语音数据包存储空间,这个三个存储空间可以是存储队列、也可以是存储池。

[0052] 待处理的语音数据包存储在待处理的语音数据包存储空间中,语音识别线程空闲时则可从该存储空间中获取该待处理的语音数据包,并将该语音数据包存储至当前正在处理的语音数据包存储空间中对其进行语音处理。当该语音数据包处理完成后,语音识别线程将处理完的语音数据包存储至处理完毕的语音数据包存储空间中。该语音数据包所属的语音连接扫描处理完毕的语音数据包存储空间,获取处理完毕的该语音数据包对应的识别结果。此时,如果下个语音数据包已经到达,语音连接则向待处理的语音数据包存储空间中存储下一个语音数据包。

[0053] S2:若均未存储有本语音连接对应的语音数据包,则将语音数据包存储至待处理的语音数据包存储空间中。

[0054] 语音数据包所属的语音连接在向服务器中存储待处理的语音数据包时,需要保证其请求处理的前一个语音数据包完全处理完毕后,再向服务器请求处理下一个语音数据包。因此,需要保证三个存储空间中均未存储有本语音连接对应的语音数据包,才会向服务器中存储下一个语音数据包。

[0055] 步骤 S206:当前语音识别线程调用与语音数据包对应的语音识别实例。

[0056] 服务器初始化过程中,会初始化语音识别实例,初始化后的语音识别实例可以存储在预设的识别实例列表中,语音识别过程中需要调用时,即可从预设的队列中调用语音识别实例。

[0057] 一种优选的调用与语音数据包对应的语音识别实例的方法如下:

[0058] S1:判断识别实例列表中是否存储有与语音数据包所属的语音连接对应的语音识别实例;若存在则执行 S2,若不存在,则执行 S3。

[0059] 与服务器建立的每个语音连接都被服务器赋予一个全局唯一标识,该标识可以为 UID 即唯一识别码。当语音连接首次发送语音数据包时,语音识别线程会语音连接分配一个语音识别实例,并将该语音识别实例的标识与该语音连接对应的全局唯一标识建立对应关系。

[0060] 当该语音连接发送下一个语音数据包时,在语音连接发送的语音数据包中携带有所属语音连接对应的全局唯一标识,语音处理线程通过语音数据包中携带的唯一标识即可确定该语音数据包所属的连接,再通过所建立的语音识别实例标识与语音连接的全局唯一标识之间的对应关系,即可确定语音数据包所属的语音连接对应的语音识别实例。

[0061] 当然,若当前语音数据包是所属的语音连接发送的首个数据包,语音识别线程还未为该语音数据包所属的语音连接分配语音识别实例,因此,在识别实例列表中不存在语音数据包所属的语音连接对应的语音识别实例。

[0062] 需要说明的是,由于每个语音连接仅对应一个语音请求,因此,也可以通过为每个语音请求赋予一个全局唯一标识,来标识语音数据包所属的语音连接以及语音请求。

[0063] S2:若存在,则直接调用与语音连接对应的语音识别实例。

[0064] 若识别实例列表中存储有与语音数据包所属的语音连接对应的语音识别实例,则证明当前语音数据包并非其所属的语音连接发送的首个语音数据包,此时,则直接调用已分配给该语音连接的语音识别实例对当前语音数据包进行语音识别即可。

[0065] S3:若不存在,则从语音识别实例列表中调用一个当前在状态为空闲的语音识别实例,并建立调用的语音识别实例与语音数据包所属的语音连接之间的对应关系。

[0066] 其中,空闲的语音识别实例为未与语音连接建立对应关系的语音识别实例。

[0067] 步骤 S208:当前语音识别线程采用调用的语音识别实例对语音数据包进行语音识别处理,并将处理完成的语音数据包压入处理完毕的语音数据包存储空间中。

[0068] 本步骤中,当前语音识别线程将处理完成的语音数据包压入处理完毕的语音数据包存储空间中,以供语音数据包所属的语音连接获取。

[0069] 语音数据包所属的语音连接会按照设定规则扫描处理完毕的语音数据包存储空间,以确定所请求处理的语音数据包是否已处理完毕,在存储空间中扫描到该语音数据包,则证明该语音数据包处理完毕,语音数据包所属的语音连接即可将处理完毕的语音数据包对应的识别结果取走。

[0070] 当然在具体实现过程中,可以在处理完毕的语音数据包存储空间中设置语音数据包列表,在语音数据列表中添加处理完成的语音数据包所属的语音连接对应的 UID,语音连接通过扫描、并判断设置的语音数据包列表中是否包含其对应的 UID 即可确定所请求处理的语音数据包是否已处理完毕。若其对应的 UID 在该列表中,则证明该语音数据包已处理完毕,则可将处理完毕的语音数据包对应的识别结果取走。

[0071] 步骤 S210:当前语音识别线程判断语音数据包中是否包含有结束标识;若包含,则执行步骤 S212,若未包含,则执行设定操作。

[0072] 其中,结束标识用于指示当前语音数据包为所属的语音连接对应的最后一个语音数据包。

[0073] 本步骤中当前语音识别线程判断语音数据包中是否包含有结束标识,目的是判断语音连接所要请求的语音数据包是否全部处理完毕,若完毕,则可将语音连接对应的语音识别实例释放,供其他语音连接使用。

[0074] 所述设定操作可以由本领域技术人员根据实际需求进行设定,本实施例对此不作具体限制。例如:设置成直接获取其他待处理的语音数据包。

[0075] 一个语音连接所请求处理的语音数据被分成多个语音数据包,在最后一个数据包

中携带有结束标识。例如：共将当前语音连接所请求处理的语音数据分成五个语音数据包，在前五个语音数据包中分别携带有用于指示当前语音数据包并非第一个语音数据包的标识（如将第一至第五个语音数据包中携带的标识分别设置为 1、2、3、4、5），而第六个语音数据包由于是最后一个语音数据包，因此，在该语音数据包中携带有用于指示当前语音数据包为最后一个语音数据包的结束标识（如将该结束标识设置成 -6）。

[0076] 上面仅是列举了一种通过将语音数据包的标号进行正、负标注来确定语音数据包是都为最后一个包的方式。在具体实现过程中，本领域技术人员可以采用任意适当的设定方式来设定结束标识，例如：仅在最后一个语音数据包中携带设定标识，当数据包中携带有设定标识时，即可确定该包为其所属语音连接所请求的最后一个语音数据包。再例如：还可以将非最后一个语音数据包中均携带标识 1，而在最后一个语音数据包中携带标识 0，将标识 0 定义为结束标识。

[0077] 步骤 S212：若语音数据包中包含有结束标识，语音识别线程则将语音数据包对应的语音识别实例的释放，并将释放后的语音识别实例的状态更新为空闲。

[0078] 将语音识别实例的状态更新为空闲，便可供其他的语音连接进行绑定。

[0079] 步骤 S214：语音识别线程在对获取的语音数据包处理完成后、判断是否还存在待处理的其它语音数据包；若存在，则返回执行步骤 S204，若不存在，则执行设定操作。

[0080] 本领域技术人员能够明了，服务器通常情况下会同时处理多个语音连接请求，而每个语音连接请求对应多个语音数据包，因此，在具体实现过程中，服务器中待处理的语音数据包存储空间中同时会存在多个待处理的语音数据包。本发明实施例中，当前语音识别线程处理完当前语音连接所请求的语音数据包后，主动获取其他待处理的语音数据包进行语音处理，直至全部待处理的语音数据包处理完毕。

[0081] 其中，设定操作可以由本领域技术人员根据实际需求进行设置。例如：设置成当判断出不存在待处理的语音数据包时，停止并释放语音识别线程。

[0082] 图 5 示出了采用本实施例中所示的语音处理方法与现有的第一种语音处理方法（为服务器的每个物理核创建一个语音识别线程，并且将语音识别线程和语音识别实例绑定，每个语音识别线程仅处理一个语音连接对应的语音请求）在处理语音请求时的性能对比图。

[0083] 该对比图，是通过把同样一批数据压入分别两个语音处理系统中进行压力测试后的测试数据进行统计所生成的，测试的在这两个进行测试的语音处理系统中服务器均包含 12 个物理核。

[0084] 其中，横坐标表示：最大识别并发数，即服务器同时处理的语音连接的数量；纵坐标表示语音识别实时率，即语音线程处理 1 秒钟的语音对应的语音数据包的时间，通过语音识别实时率能够体现服务器的处理语音请求的性能。

[0085] 在图 5 中，圆点串接的线指示采用本实施例中的语音处理方法处理语音请求时的性能曲线；方块串接的线指示采用现有的第一种线程绑定的语音处理方法处理语音请求时的性能曲线。从图 5 中可知，随着最大识别并发数增加，两个语音识别系统的语音识别实时率均在增加。当最大识别并发数大于 24 时，本发明的优势开始体现，此时虽然本发明的实时率也在升高，但升高趋势不明显。可见，本发明实施例提供的语音处理方法能够提升语音处理系统的吞吐量。

[0086] 在具体实现过程中,通常会面临对服务中存储的语音识别的语音模型进行更新的问题,当需要对服务器中用于进行语音识别的语音模型进行更新时,则需要对服务器进行外部中断。现有的技术方案在处理该中断时,则直接将服务器连接的各语音连接中断,影响用户的使用体验。本发明实施例中的语音处理方法,当服务器主线程接收到对服务器的外部中断请求后,会维持已经建立语音连接并继续处理这些已经建立的语音连接所请求处理的语音数据包,直至语音连接所请求处理的全部语音数据包处理完毕后,再中断该语音连接。而对于未建立的语音连接,则不再接受其连接请求。

[0087] 具体实现如下:

[0088] 当服务器主线程接收到对服务器的外部中断请求时,生成中断标识;根据中断标识,针对服务器中已建立的语音连接,继续对语音连接对应的语音数据包进行语音识别处理,待语音连接对应的全部语音数据包进行语音识别处理后,中断语音连接;针对未与服务器建立连接的语音连接,直接取消与服务器建立连接的请求,不再建立于服务器之间的语音连接。

[0089] 本发明实施例提供的这种处理外部中断请求方案,对已经建立语音连接的用户继续提供服务,直至语音连接所请求的全部语音数据包处理完毕,对于用户而言语音识别请求不会受到影响,更不会突然中断,因此,能够提升用户的使用体验。

[0090] 通过本发明实施例提供的语音处理方法,当前语音识别线程并非仅为一个语音连接提供语音处理服务,在当前语音连接对应的语音数据包处理完成后即当前语音识别线程空闲时,当前语音识别线程主动判断是否还存在待处理的其它请求的语音数据包,若存在,则直接获取一个待处理的其他请求的语音数据包进行处理。本发明实施例提供的语音处理方法,一方面,当语音识别线程空闲时主动获取其他待处理的语音数据包进行语音处理,因此,语音识别线程并不存在现有技术中所存在的语音识别线程等待语音连接对应的下一个语音数据包而造成资源浪费的问题。另一方面,本发明实施例中的语音识别线程空闲时是通过主动获取其他待处理的语音数据包,来改变其对应的物理核的空闲状态,以使物理核的资源能够被充分利用。整个过程中都不存在语音识别线程之间的切换,语音识别线程对应的物理核的全部资源都利用于处理语音数据包,因此,能够有效避免现有的处理方法中存在的语音识别线程之间的轮转而消耗物理核资源的问题。再一方面,语音线程之间的轮转是由处理器按照自身设定的时钟周期进行的,即在处理同一个语音数据包时,可能暂时中断一段时间,来处理另外的一个语音识别线程对应的语音数据包,因此,语音识别线程之间的轮转将导致单个语音请求处理速度的下降。而本发明实施例中,并不存在语音线程之间的轮转,因此,能够避免语音识别线程轮转而导致的单个语音请求处理速度下降的问题。

[0091] 下面参照图 6 以一具体实例对本发明实施例的语音处理方法进行说明。

[0092] 图 6 示出了执行实施例二中所所述的语音处理方法的语音处理系统启动和中断的流程示意图。

[0093] 如图 6 所示,语音处理系统即服务器启动时预先执行以下操作:

[0094] S1:加载识别资源。

[0095] 其中,服务器在语音识别处理时使用相应地的识别资源,因此,在服务器启动时,需要预先加载识别资源。识别资源存储在硬盘中,加载的目的是将存储在硬盘中的识别资源加载至服务器内存中,以避免频繁访问硬盘。

[0096] S2 :初始化识别实例。

[0097] 初始化识别实例即建立设定个数的语音识别实例,并将语音识别实例即识别实例存储在语音识别实例表中。在未连接语音连接前,语音识别实例表中可能仅存储有初始化的各语音识别实例,但是,当服务器连接了语音连接后,语音识别实例表中将存储有语音请求对应的全局唯一标识与语音识别实例的对应关系。

[0098] S3 :初始化任务队列。

[0099] 本具体实例中以服务器包含 M 个物理核、初始化的任务队列包含等待处理的数据包队列、正在处理的数据包列表、处理完毕的数据包列表和识别实例表,并且初始化后的各任务队列中可存储目标的个数均为 N,可并发处理的语音连接个数为 K(通常 $K = N$) 个为例,对后续的流程进行说明。

[0100] S4 :创建并启动工作线程。

[0101] 其中,工作线程即语音识别线程。

[0102] 本具体实例中由于服务器包含 M 个物理核,因此,在创建语音识别线程时,服务器主线程为每个物理核创建一个语音识别线程,即共创建 M 个语音识别线程。由于语音识别线程的个数等于服务器逻辑内核即服务器包含的物理核的个数,因此,语音识别线程间不会轮流或随意争抢处理器资源。

[0103] 在执行完上述操作后,识别服务器即可执行 S5 与客户端建立语音连接。本具体实例中,为了使识别服务器的资源能够充分利用,在 S4 中所创建的各语音识别线程均会处理语音连接对应的语音数据包,即执行 S6。

[0104] S6 :处理语音识别请求。

[0105] 下面参照图 3 对处理一个语音识别请求中的一个语音数据包的处理进行说明。

[0106] 从图 3 中可知,一个语音识别请求对应一个语音连接。在具体实现过程中,每个语音请求被赋予一个全局唯一标识即 UID,也就是说一个语音连接对应一个 UID,每个语音连接接收来自一个语音请求的数据包。一个 UID 对应的语音请求中包含的某个语音数据包被压到一个先进先出的等待处理的数据包队列中。

[0107] 向等待处理的数据包队列中压入新的语音数据包需要同时满足下述 4 个条件:

[0108] 1、队列未滿;

[0109] 2、请求压入的语音数据包所属的语音请求对应的 UID 不存在于等待处理的数据包队列;如果存在,说明该 UID 对应的语音请求的上个语音数据包还没有被处理。

[0110] 3、请求压入的语音数据包所属的语音请求对应的 UID 不存在于正在处理的数据包列表中;如果存在,说明该 UID 对应的语音请求的上个语音包还没有处理完。

[0111] 4、请求压入的语音数据包所属的语音请求对应的 UID 不存在于处理完毕的数据包列表中;如果存在,说明该 UID 对应的语音请求的上个语音包已经处理完,但还没有被其所属的语音连接取走。

[0112] 以上 4 个条件,但凡一个条件不满足,语音连接将进入等待状态,直到条件满足后,再压入所服务的语音请求的下一个语音数据包。语音识别请求中的一个语音数据包的处理流程如下:

[0113] 步骤 1 :语音识别线程在处理语音数据包时,会到等待处理的数据包队列中获取一个语音数据包,进行处理。

[0114] 步骤 2:根据语音数据包中携带的所属语音请求的 UID 去识别实例表中找到相应的语音识别实例。

[0115] 如果该语音数据包中携带的 UID 未与任何语音识别实例绑定,说明该语音数据包是这个 UID 对应的语音请求所请求的第一个语音数据包,此时,则找一个空闲的语音识别实例和该 UID 进行绑定,绑定成功则意味着该语音识别实例仅为所绑定的语音请求服务。

[0116] 步骤 3:取得待识别语音数据包,和其对应的语音识别实例后,该语音数据包携带的 UID 被从等待处理的数据包队列中移动到正在处理的数据包列表中。

[0117] 将 UID 从等待处理的数据包队列中里移动到正在处理的数据包列表中,表示这个 UID 对应的请求所请求的语音数据包正在被处理。

[0118] 步骤 4:语音识别线程处理完该 UID 对应的语音数据包后,将该 UID 从正在处理的数据包列表中移动到处理完毕的数据包列表中。

[0119] 将该 UID 从正在处理的数据包列表中移动到处理完毕的数据包列表中以表示该语音数据包处理完毕,识别的反馈被放到语音数据包的反馈区域中。等待识别反馈的语音连接不断的扫描处理完毕的数据包列表。一旦发现其对应的 UID,就会把识别结果取走。并把该 UID 从“处理完毕的数据包队列中移除。如果当前处理的是一个语音请求的最后一个语音包,除了取走识别结果,还需要把 UID 和之前绑定的语音识别实例解除绑定,以表示该识别实例的状态为空闲。

[0120] 需要说明的是,上述仅是以单个语音处理线程处理一个语音数据包为例进行的说明,在具体实现过程中,各语音处理线程均从等待处理的数据包队列中获取待处理的语音数据包进行处理,并且,将处理过程中的语音数据包携带的 UID 移动到正在处理的数据包列表中,将处理完成的语音数据包携带的 UID 移动到处理完毕的数据包列表中。当各语音处理线程空闲时,则会从等待处理的数据包队列中获取待处理的语音数据包进行处理,以实现并行语音识别。

[0121] 当需要对服务器中用于进行语音识别的语音模型进行更新时,则需要对服务器进行外部中断,本具体实例中,当接收到 S7 外部中断请求后,则执行 S8。

[0122] S8:关闭服务连接。

[0123] 本具体实例中,在关闭服务连接前,必须处理完已建立的各语音识别任务,并返回最终识别结果(即需要处理完以建立的语音连接所请求处理的全部语音数据包)后才关闭所有语音连接,释放语音识别实例,释放全局的模型资源并退出。具体地,在处理完已建立各语音识别任务后依次执行 S9:停止并释放语音识别线程;S10:释放任务队列;S11:释放语音识别实例;S12:释放识别资源。

[0124] 本具体实例提供的语音处理方法,一方面,由于所创建的语音识别线程的个数等于服务器包含的物理核的个数,因此,语音识别线程不会轮流或随意争抢处理器资源,从而提高语音识别系统的吞吐量。再一方面,本具体实例中松绑了服务器计算资源和语音识别实例,一个语音识别线程不止服务于一个语音请求(即一个语音识别线程不止处理一个语音连接所请求的处理的语音数据包),从而提高了语音识别系统的吞吐量。

[0125] 实施例三

[0126] 参照图 8,示出了本发明实施例三的一种语音处理装置的结构框图。

[0127] 本发明实施例的语音处理装置包括:获取模块 802,用于供当前语音识别线程获

取按照设定规则存储的语音数据包；处理模块 804，用于调用语音识别实例对获取的语音数据包进行语音识别处理；判断模块 806，用于在对获取的语音数据包处理完成后、判断是否还存在待处理的其它语音数据包；返回模块 808，用于若存在，则返回获取模块继续对其它语音数据包进行语音识别处理，其中，获取的语音数据包与其它语音数据包属于不同的语音连接。

[0128] 通过本发明实施例提供的语音处理装置，当前语音识别线程并非仅为一个语音连接提供语音处理服务，在当前语音连接对应的语音数据包处理完成后即当前语音识别线程空闲时，当前语音识别线程主动判断是否还存在待处理的其它请求的语音数据包，若存在，则直接获取一个待处理的其他请求的语音数据包进行处理。本发明实施例提供的语音处理装置，一方面，当语音识别线程空闲时主动获取其他待处理的语音数据包进行语音处理，因此，语音识别线程并不存在现有技术中所存在的语音识别线程等待语音连接对应的下一个语音数据包而造成资源浪费的问题。另一方面，本发明实施例中的语音识别线程空闲时是通过主动获取其他待处理的语音数据包，来改变其对应的物理核的空闲状态，以使物理核的资源能够被充分利用。整个过程中都不存在语音识别线程之间的切换，语音识别线程对应的物理核的全部资源都利用于处理语音数据包，因此，能够有效避免现有的处理方法中存在的语音识别线程之间的轮转而消耗物理核资源的问题。再一方面，语音线程之间的轮转是由处理器按照自身设定的时钟周期进行的，即在处理同一个语音数据包时，可能暂时中断一段时间，来处理另外的一个语音识别线程对应的语音数据包，因此，语音识别线程之间的轮转将导致单个语音请求处理速度的下降。而本发明实施例中，并不存在语音线程之间的轮转，因此，能够避免语音识别线程轮转而导致的单个语音请求处理速度下降的问题。

[0129] 实施例四

[0130] 参照图 9，示出了本发明实施例四的一种语音处理装置的结构框图。

[0131] 本发明实施例是对实施例三中的语音处理装置的进一步优化，优化后的语音处理装置包括：获取模块 902，用于供当前语音识别线程获取按照设定规则存储的语音数据包；处理模块 904，用于调用语音识别实例对获取的语音数据包进行语音识别处理；判断模块 906，用于在对获取的语音数据包处理完成后、判断是否还存在待处理的其它语音数据包；返回模块 908，用于若存在，则返回获取模块继续对其它语音数据包进行语音识别处理，其中，获取的语音数据包与其它语音数据包属于不同的语音连接。

[0132] 优选地，所述处理模块 904 包括：语音识别实例调用模块 9042，用于调用与所述语音数据包对应的语音识别实例；语音识别模块 9044，用于采用调用的所述语音识别实例对所述语音数据包进行语音识别处理，并将处理完成的所述语音数据包压入处理完毕的语音数据包存储空间中，以供所述语音数据包所属的语音连接获取。

[0133] 优选地，语音识别实例调用模块 9042 调用与所述语音数据包对应的语音识别实例时：判断识别实例列表中是否存储有与所述语音数据包所属的语音连接对应的语音识别实例；若存在，则直接调用与所述语音连接对应的语音识别实例；若不存在，则从所述语音识别实例列表中调用一个当前在状态为空闲的语音识别实例，并建立调用的所述语音识别实例与所述语音数据包所属的语音连接之间的对应关系，其中，所述空闲的语音识别实例为未与语音连接建立对应关系的语音识别实例。

[0134] 优选地，按照设定规则存储的语音数据包通过以下方式存储：所述语音数据包所

属的语音连接判断待处理的语音数据包存储空间中、当前正在处理的语音数据包存储空间中、以及处理完毕的语音数据包存储空间中是否存储有本语音连接对应的语音数据包；若均未存储有本语音连接对应的语音数据包，则将所述语音数据包存储至所述待处理的语音数据包存储空间中。

[0135] 优选地，本发明实施例的语音处理装置还包括：标识判断模块 910，用于在所述语音识别模块 904 将处理完成完毕的所述语音数据包压入处理完毕的语音数据包队列中之后，判断所述语音数据包中是否包含有结束标识，其中，所述结束标识用于指示当前语音数据包为所属的语音连接对应的最后一个语音数据包；语音识别实例释放模块 912，用于若所述标识判断模块的判断结果为是时，则将所述语音数据包对应的语音识别实例的释放，并将释放后的所述语音识别实例的状态更新为空闲。

[0136] 优选地，本发明实施例的语音处理装置还包括：创建模块 914，用于在所述获取模块 902 供当前语音识别线程获取按照设定规则存储的语音数据包之前，供服务器主线程依据服务器包含的物理核数量，分别为每个物理核创建一个语音识别线程。

[0137] 优选地，本发明实施例的语音处理装置还包括：生产模块 916，用于当服务器主线程接收到对服务器的外部中断请求时，生成中断标识；连接处理模块 918，用于根据所述中断标识，针对所述服务器中已建立的语音连接，继续对所述语音连接对应的语音数据包进行语音识别处理，待所述语音连接对应的全部语音数据包进行语音识别处理后，中断所述语音连接；针对未与所述服务器建立连接的语音连接，直接取消与所述服务器建立连接请求，不再建立于所述服务器之间的语音连接。

[0138] 本实施例的语音处理装置用于实现前述实施例一以及实施例二中相应的语音处理方法，并且具有相应的方法实施例的有益效果，在此不再赘述。

[0139] 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单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。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实施例方案的目的。本领域普通技术人员在不付出创造性的劳动的情况下，即可以理解并实施。

[0140] 通过以上的实施方式的描述，本领域的技术人员可以清楚地了解到各实施方式可借助软件加必需的通用硬件平台的方式来实现，当然也可以通过硬件。基于这样的理解，上述技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来，该计算机软件产品可以存储在计算机可读存储介质中，如 ROM/RAM、磁碟、光盘等，包括若干指令用以使得一台计算机设备（可以是个人计算机，服务器，或者网络设备等等）执行各个实施例或者实施例的某些部分所述的方法。

[0141] 最后应说明的是：以上实施例仅用以说明本发明的技术方案，而非对其限制；尽管参照前述实施例对本发明进行了详细的说明，本领域的普通技术人员应当理解：其依然可以对前述各实施例所记载的技术方案进行修改，或者对其中部分技术特征进行等同替换；而这些修改或者替换，并不使相应技术方案的本质脱离本发明各实施例技术方案的精神和范围。

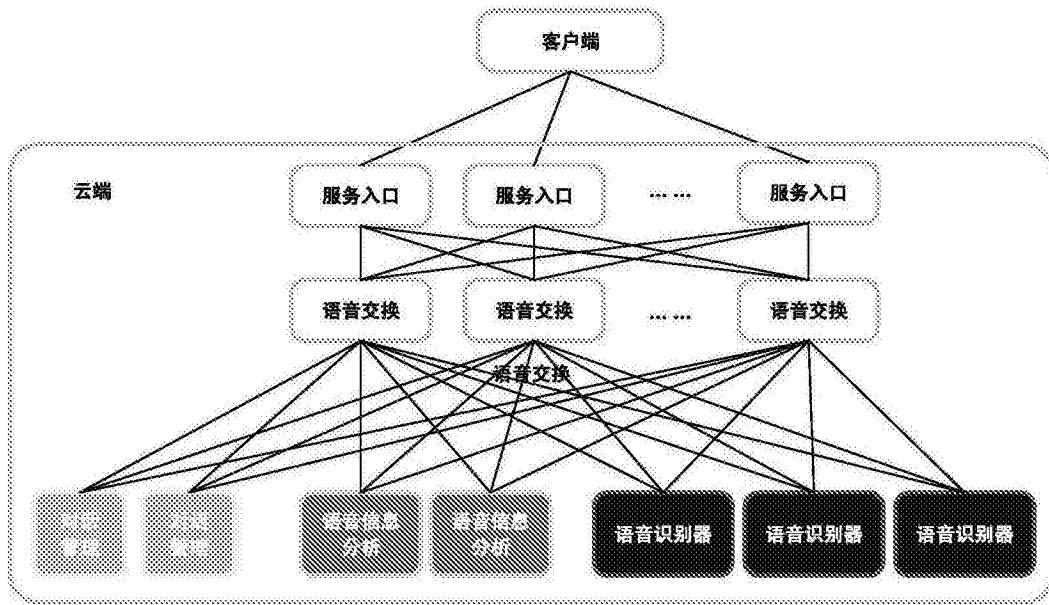


图 1

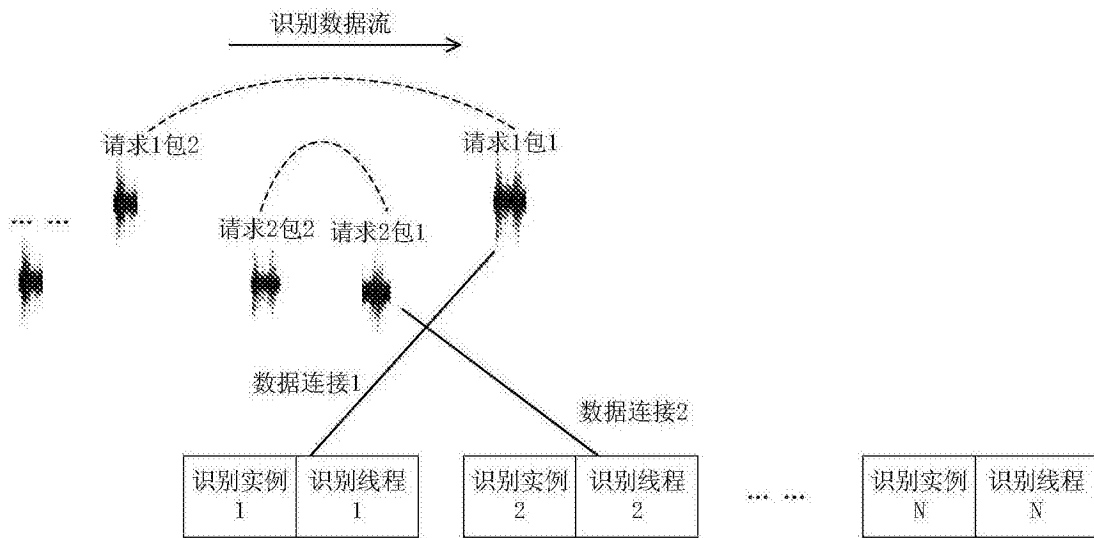


图 2

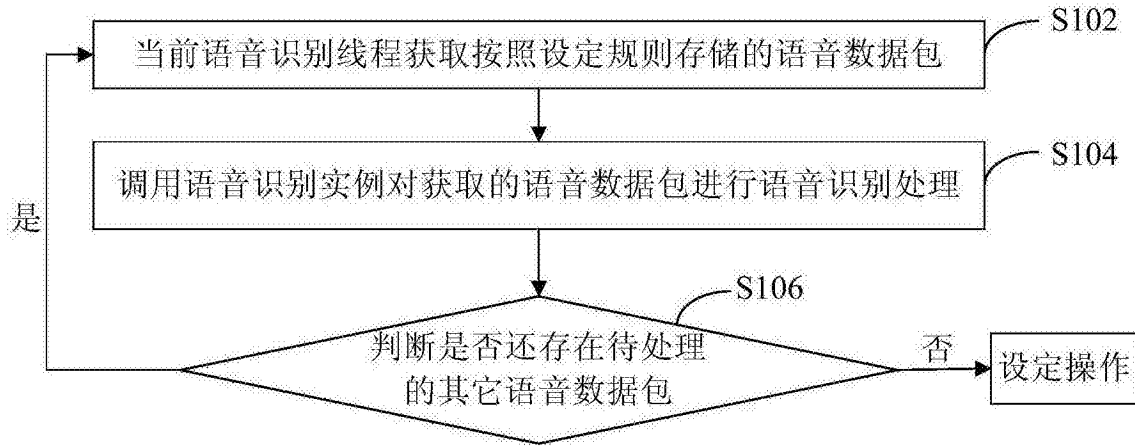


图 3

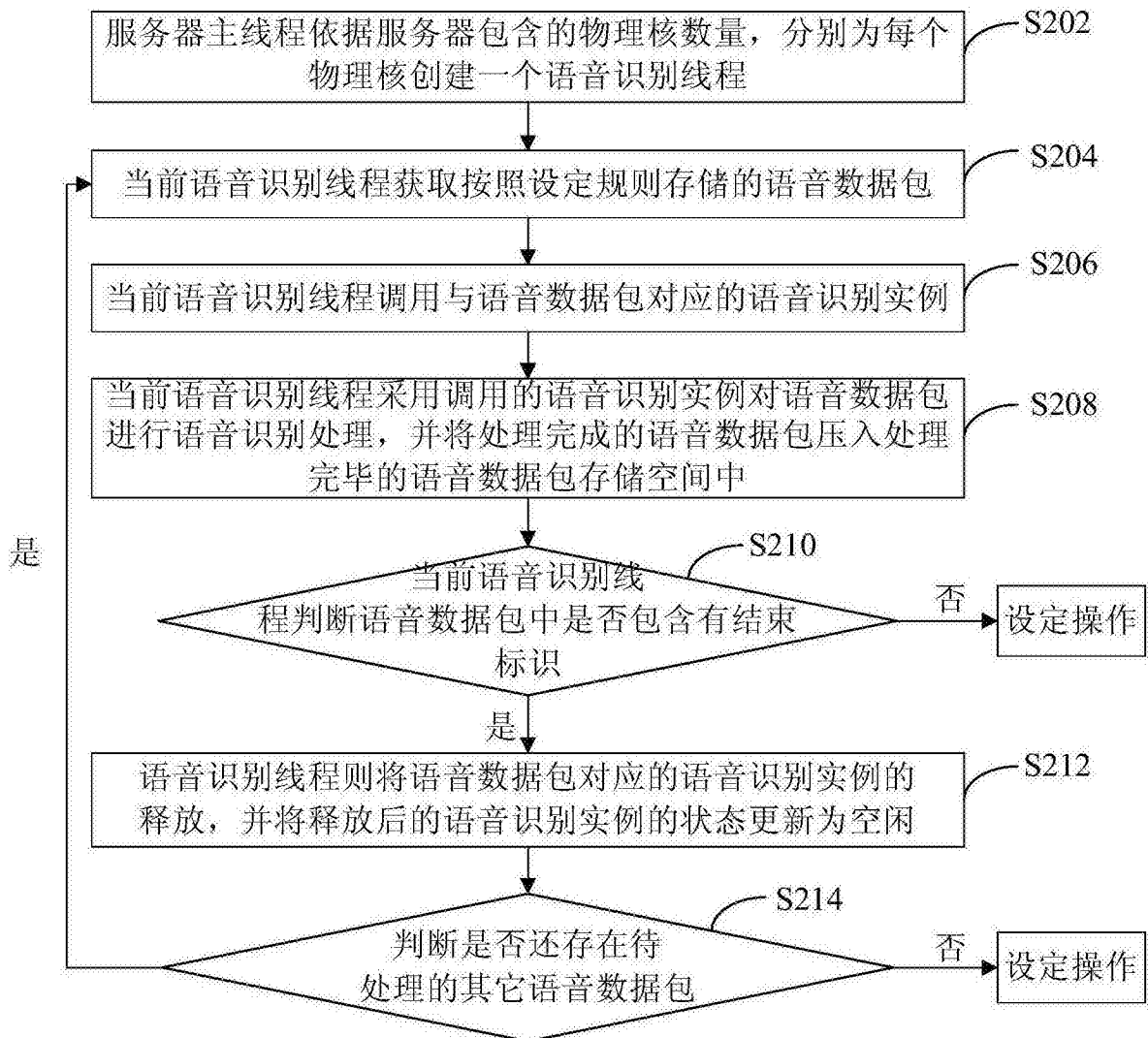


图 4

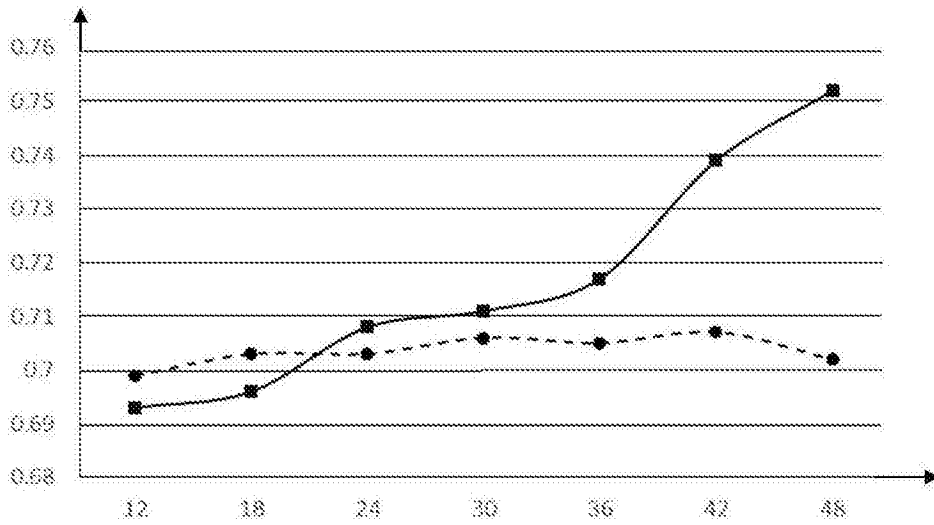


图 5

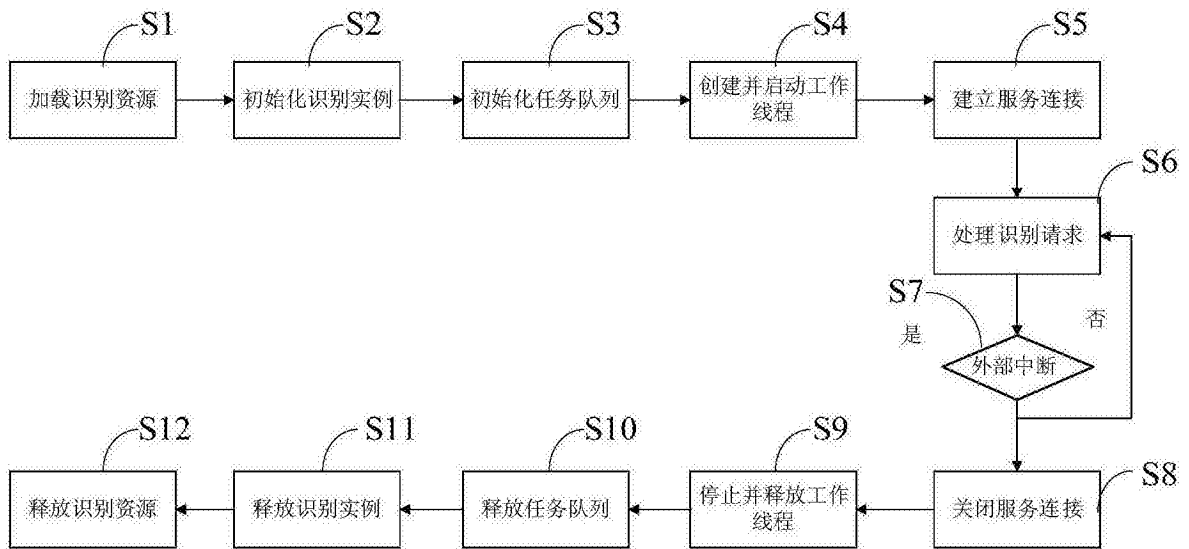


图 6

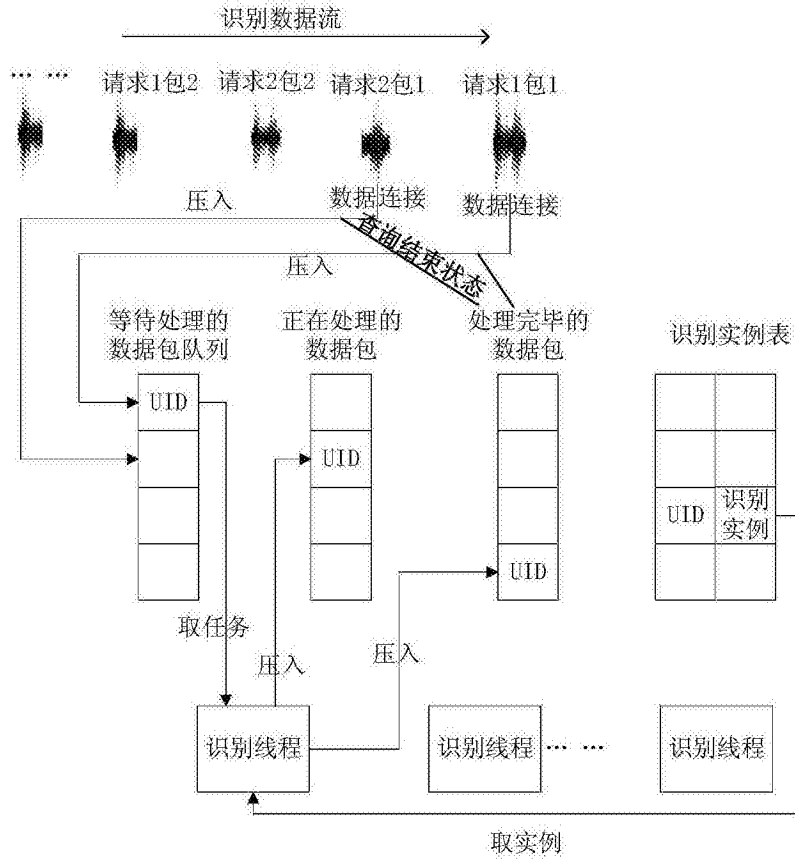


图 7



图 8

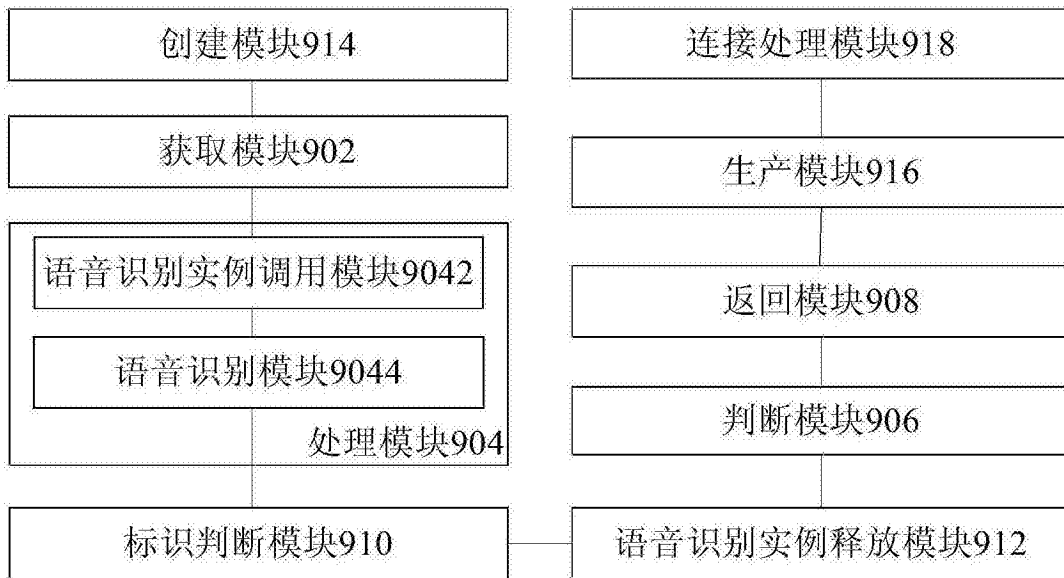


图 9