



(12)发明专利

(10)授权公告号 CN 105843867 B

(45)授权公告日 2019.09.03

(21)申请号 201610154457.7

G06F 16/2457(2019.01)

(22)申请日 2016.03.17

(56)对比文件

(65)同一申请的已公布的文献号  
申请公布号 CN 105843867 A

CN 102317943 A,2012.01.11,  
CN 104850601 A,2015.08.19,  
CN 104750776 A,2015.07.01,  
CN 103744960 A,2014.04.23,  
US 2015310090 A1,2015.10.29,  
梁艳 等.“基于solr的异构数据融合检索技术”.《无线互联科技》.2013,(第5期),  
梁艳 等.“基于solr的异构数据融合检索技术”.《无线互联科技》.2013,(第5期),

(43)申请公布日 2016.08.10

(73)专利权人 畅捷通信息技术股份有限公司  
地址 100094 北京市海淀区北清路68号用友软件园

审查员 李亚楠

(72)发明人 洪光华

(74)专利代理机构 北京友联知识产权代理事务所(普通合伙) 11343

代理人 尚志峰 汪海屏

(51)Int.Cl.

G06F 16/22(2019.01)

G06F 16/23(2019.01)

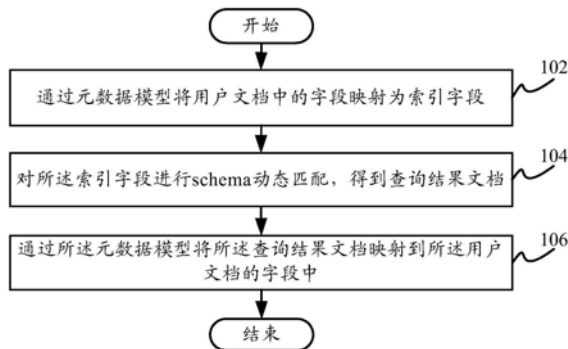
权利要求书2页 说明书8页 附图1页

(54)发明名称

基于元数据模型的检索方法和基于元数据模型的检索装置

(57)摘要

本发明提出了一种基于元数据模型的检索方法和一种基于元数据模型的检索装置,其中,基于元数据模型的检索方法包括:通过元数据模型将用户文档中的字段映射为索引字段;对所述索引字段进行schema动态匹配,得到查询结果文档;通过所述元数据模型将所述查询结果文档映射到所述用户文档的字段中。通过本发明的技术方案,在用户文档类型发生变更后,直接更新元数据模型即可,而不必更新索引服务器中的相关配置并重新加载,减少了更新复杂度,使开发工作更加简单高效,实现了以较少的资源提供更稳定的服务。



1. 一种基于元数据模型的检索方法,其特征在于,包括:  
通过元数据模型将用户文档中的字段映射为索引字段;  
对所述索引字段进行schema动态匹配,得到查询结果文档;  
通过所述元数据模型将所述查询结果文档映射到所述用户文档的字段中;  
所述元数据模型为客户端的应用程序接口与索引服务器之间的元数据层,所述用户文档来自所述客户端,所述schema动态匹配的服务由所述索引服务器提供。

2. 根据权利要求1所述的基于元数据模型的检索方法,其特征在于,所述通过元数据模型将用户文档中的字段映射为索引字段,具体包括:

根据所述元数据模型,确定所述用户文档所需的元数据,并在预设元数据库中为所述用户文档调用所述所需的元数据,其中,所述用户文档所需的元数据为所述用户文档到所述索引服务器的索引文档的映射关系。

3. 根据权利要求2所述的基于元数据模型的检索方法,其特征在于,还包括:

根据来自所述客户端的注解,为所述元数据模型定义全量元数据,并将定义的所述全量元数据同步更新至所述索引服务器;和/或

根据来自所述客户端或所述索引服务器的修改命令,修改所述元数据模型的元数据。

4. 根据权利要求1至3中任一项所述的基于元数据模型的检索方法,其特征在于,所述元数据模型包括文档模型和字段模型,其中,所述文档模型包括:文档类型、文档所属域和文档包含的字段属性集合,所述字段模型包括:所述用户文档的字段名称、索引类型和其他属性,所述其他属性包括:是否保存、是否索引、是否为拼音、是否为多值、是否为唯一和是否为只读。

5. 一种基于元数据模型的检索装置,其特征在于,包括:

索引字段映射单元,通过元数据模型将用户文档中的字段映射为索引字段;

动态匹配单元,对所述索引字段进行schema动态匹配,得到查询结果文档;

查询结果映射单元,通过所述元数据模型将所述查询结果文档映射到所述用户文档的字段中;

所述元数据模型为客户端的应用程序接口与索引服务器之间的元数据层,所述用户文档来自所述客户端,所述schema动态匹配的服务由所述索引服务器提供。

6. 根据权利要求5所述的基于元数据模型的检索装置,其特征在于,所述索引字段映射单元具体用于:

根据所述元数据模型,确定所述用户文档所需的元数据,并在预设元数据库中为所述用户文档调用所述所需的元数据,其中,所述用户文档所需的元数据为所述用户文档到所述索引服务器的索引文档的映射关系。

7. 根据权利要求6所述的基于元数据模型的检索装置,其特征在于,还包括:

全量更新单元,根据来自所述客户端的注解,为所述元数据模型定义全量元数据,并将定义的所述全量元数据同步更新至所述索引服务器;和/或

增量更新单元,根据来自所述客户端或所述索引服务器的修改命令,修改所述元数据模型的元数据。

8. 根据权利要求5至7中任一项所述的基于元数据模型的检索装置,其特征在于,所述元数据模型包括文档模型和字段模型,其中,所述文档模型包括:文档类型、文档所属域和

文档包含的字段属性集合,所述字段模型包括:所述用户文档的字段名称、索引类型和其他属性,所述其他属性包括:是否保存、是否索引、是否为拼音、是否为多值、是否为唯一和是否为只读。

## 基于元数据模型的检索方法和基于元数据模型的检索装置

### 技术领域

[0001] 本发明涉及计算机技术领域,具体而言,涉及一种基于元数据模型的检索方法和一种基于元数据模型的检索装置。

### 背景技术

[0002] 目前,检索的作用就是在用户给定一个关键词的前提下,检索并给出用户最想要的内容。但是随着云应用技术的发展,部署在云端的应用也越来越多,与此同时,伴随着用户数量的增长,应用的数据也越来越多,单纯使用数据库中的检索功能,不仅效果不好,还会影响数据库的其它功能的使用,为此,开发者一般使用独立服务器单独搭建搜索服务。

[0003] 然而,在这种方案中,单个应用的部署和维护成本又太高,开发需要的时间和精力太多,在云端环境中,如果各个应用的开发者都使用自己的服务,撇开部署集群要使用多台服务器不说,在开发过程中,用户需要保持应用的信息与solr(一种独立的企业级搜索应用)的schema(可扩展标记语言架构)定义的字段保持一致,在应用发生变化时,需要同步修改集群中的schema相关信息,并重载加载修改过的配置,这个过程极为复杂,如果漏掉其中的某个步骤,就会导致更新索引和查询失败。

[0004] 因此需要一种新的技术方案,可以减少搜索更新复杂度的同时,使开发工作更加简单高效。

### 发明内容

[0005] 本发明正是基于上述问题,提出了一种新的技术方案,可以减少搜索更新复杂度的同时,使开发工作更加简单高效。

[0006] 有鉴于此,本发明的一方面提出了一种基于元数据模型的检索方法,包括:通过元数据模型将用户文档中的字段映射为索引字段;对所述索引字段进行schema动态匹配,得到查询结果文档;通过所述元数据模型将所述查询结果文档映射到所述用户文档的字段中。

[0007] 在该技术方案中,可以solr(一种独立的企业级搜索应用)服务前端添加一个元数据层,用来管理元数据模型,其接受并保存用户定义的元数据,同时可以通知使用元数据的元数据层去数据库中加载最新的元数据,由于元数据层是在索引集群之前,并且直接面向客户提供搜索和索引服务,故索引和搜索服务都会使用用户定义的元数据来进行验证。

[0008] 这样,通过元数据模型可以将用户用于检索的字段映射为索引字段,再对索引字段进行schema(可扩展标记语言架构)动态匹配,从而完成检索,并且,可以将检索得到的查询结果文档继续经由元数据模型映射到用户文档的字段,完成检索步骤。通过该技术方案,在用户文档类型发生变更后,直接更新元数据模型即可,而不必更新索引服务器中的相关配置并重新加载,减少了更新复杂度,使开发工作更加简单高效,实现了以较少的资源提供更稳定的服务。

[0009] 在上述技术方案中,优选地,所述元数据模型为客户端的应用程序接口与索引服

务器之间的元数据层,所述用户文档来自所述客户端,所述schema动态匹配的服务由所述索引服务器提供。

[0010] 在该技术方案中,用户文档中具有客户端的用户用于检索的字段,solr(一种独立的企业级搜索应用)服务前端添加有一个元数据层,用来管理元数据模型,其接受并保存用户定义的元数据,同时可以通知使用元数据的元数据层去数据库中加载最新的元数据,由于元数据层是在索引集群之前,并且直接面向客户提供搜索和索引服务,故索引和搜索服务都会使用用户定义的元数据来进行验证,从而在用户文档类型发生变更后,直接更新元数据模型即可,而不必更新索引服务器中的相关配置并重新加载,减少了更新复杂度。

[0011] 在上述任一技术方案中,优选地,所述通过元数据模型将用户文档中的字段映射为索引字段,具体包括:根据所述元数据模型,确定所述用户文档所需的元数据,并在预设元数据库中为所述用户文档调用所述所需的元数据,其中,所述用户文档所需的元数据为所述用户文档到所述索引服务器的索引文档的映射关系。

[0012] 在该技术方案中,可以根据元数据层中的用户文档所需的元数据进行映射,使得用户文档中的字段可以映射为索引字段,从而进一步通过索引字段进行schema动态匹配,从而完成检索。

[0013] 在上述任一技术方案中,优选地,还包括:根据来自所述客户端的注解,为所述元数据模型定义全量元数据,并将定义的所述全量元数据同步更新至所述索引服务器;和/或根据来自所述客户端或所述索引服务器的修改命令,修改所述元数据模型的元数据。

[0014] 在该技术方案中,元数据的管理分为查询与更新两种,更新时又可分为全量更新和增量更新两种,其中,全量更新即通过客户端使用注解来定义全量元数据,同时在索引服务器启动时,将元数据更新到索引服务器,增量更新即根据来自客户端的应用程序接口或者服务端的网页界面的修改命令修改元数据中的相关内容。通过对元数据的更新,可不必更新索引服务器中的相关配置并重新加载,减少了更新复杂度,使开发工作更加简单高效,实现了以较少的资源提供更稳定的服务。

[0015] 在上述任一技术方案中,优选地,所述元数据模型包括文档模型和字段模型,其中,所述文档模型包括:文档类型、文档所属域和文档包含的字段属性集合,所述字段模型包括:所述用户文档的字段名称、索引类型和其他属性,所述其他属性包括:是否保存、是否索引、是否为拼音、是否为多值、是否为唯一和是否为只读。

[0016] 在该技术方案中,元数据是关于数据的描述性数据信息,用来定义用户文档到索引文档的一个映射关系,它有文档模型和字段模型两种。其中,文档模型包含了文档类型、文档所属域和文档包含的字段属性集合。字段模型包含了用户文档的字段名称、索引类型、是否保存、是否索引、是否拼音、是否多值、是否唯一和是否只读。其中,索引类型指的是字段在建索引和查询过程中使用的分词器类型,是否只读指的是限制某些字段写属性或者只能由后端服务来进行相应的更新。

[0017] 本发明的另一方面提出了一种基于元数据模型的检索装置,包括:索引字段映射单元,通过元数据模型将用户文档中的字段映射为索引字段;动态匹配单元,对所述索引字段进行schema动态匹配,得到查询结果文档;查询结果映射单元,通过所述元数据模型将所述查询结果文档映射到所述用户文档的字段中。

[0018] 在该技术方案中,可以solr(一种独立的企业级搜索应用)服务前端添加一个元数

据层,用来管理元数据模型,其接受并保存用户定义的元数据,同时可以通知使用元数据的元数据层去数据库中加载最新的元数据,由于元数据层是在索引集群之前,并且直接面向客户提供搜索和索引服务,故索引和搜索服务都会使用用户定义的元数据来进行验证。

[0019] 这样,通过元数据模型可以将用户用于检索的字段映射为索引字段,再对索引字段进行schema(可扩展标记语言架构)动态匹配,从而完成检索,并且,可以将检索得到的查询结果文档继续经由元数据模型映射到用户文档的字段,完成检索步骤。通过该技术方案,在用户文档类型发生变更后,直接更新元数据模型即可,而不必更新索引服务器中的相关配置并重新加载,减少了更新复杂度,使开发工作更加简单高效,实现了以较少的资源提供更稳定的服务。

[0020] 在上述技术方案中,优选地,所述元数据模型为客户端的应用程序接口与索引服务器之间的元数据层,所述用户文档来自所述客户端,所述schema动态匹配的服务由所述索引服务器提供。

[0021] 在该技术方案中,用户文档中具有客户端的用户用于检索的字段,solr(一种独立的企业级搜索应用)服务前端添加有一个元数据层,用来管理元数据模型,其接受并保存用户定义的元数据,同时可以通知使用元数据的元数据层去数据库中加载最新的元数据,由于元数据层是在索引集群之前,并且直接面向客户提供搜索和索引服务,故索引和搜索服务都会使用用户定义的元数据来进行验证,从而在用户文档类型发生变更后,直接更新元数据模型即可,而不必更新索引服务器中的相关配置并重新加载,减少了更新复杂度。

[0022] 在上述任一技术方案中,优选地,所述索引字段映射单元具体用于:根据所述元数据模型,确定所述用户文档所需的元数据,并在预设元数据库中为所述用户文档调用所述所需的元数据,其中,所述用户文档所需的元数据为所述用户文档到所述索引服务器的索引文档的映射关系。

[0023] 在该技术方案中,可以根据元数据层中的用户文档所需的元数据进行映射,使得用户文档中的字段可以映射为索引字段,从而进一步通过索引字段进行schema动态匹配,从而完成检索。

[0024] 在上述任一技术方案中,优选地,还包括:全量更新单元,根据来自所述客户端的注解,为所述元数据模型定义全量元数据,并将定义的所述全量元数据同步更新至所述索引服务器;和/或增量更新单元,根据来自所述客户端或所述索引服务器的修改命令,修改所述元数据模型的元数据。

[0025] 在该技术方案中,元数据的管理分为查询与更新两种,更新时又可分为全量更新和增量更新两种,其中,全量更新即通过客户端使用注解来定义全量元数据,同时在索引服务器启动时,将元数据更新到索引服务器,增量更新即根据来自客户端的应用程序接口或者服务端的网页界面的修改命令修改元数据中的相关内容。通过对元数据的更新,可不必更新索引服务器中的相关配置并重新加载,减少了更新复杂度,使开发工作更加简单高效,实现了以较少的资源提供更稳定的服务。

[0026] 在上述任一技术方案中,优选地,所述元数据模型包括文档模型和字段模型,其中,所述文档模型包括:文档类型、文档所属域和文档包含的字段属性集合,所述字段模型包括:所述用户文档的字段名称、索引类型和其他属性,所述其他属性包括:是否保存、是否索引、是否为拼音、是否为多值、是否为唯一和是否为只读。

[0027] 在该技术方案中,元数据是关于数据的描述性数据信息,用来定义用户文档到索引文档的一个映射关系,它有文档模型和字段模型两种。其中,文档模型包含了文档类型、文档所属域和文档包含的字段属性集合。字段模型包含了用户文档的字段名称、索引类型、是否保存、是否索引、是否拼音、是否多值、是否唯一和是否只读。其中,索引类型指的是字段在建索引和查询过程中使用的分词器类型,是否只读指的是限制某些字段写属性或者只能由后端服务来进行相应的更新。

[0028] 通过以上技术方案,在用户文档类型发生变更后,直接更新元数据模型即可,而不必更新索引服务器中的相关配置并重新加载,减少了更新复杂度,使开发工作更加简单高效,实现了以较少的资源提供更稳定的服务。

### 附图说明

[0029] 图1示出了根据本发明的一个实施例的基于元数据模型的检索方法的流程图;

[0030] 图2示出了根据本发明的一个实施例的基于元数据模型的检索装置的框图;

[0031] 图3示出了根据本发明的一个实施例的客户端、元数据层与索引集群的交互示意图。

### 具体实施方式

[0032] 为了能够更清楚地理解本发明的上述目的、特征和优点,下面结合附图和具体实施方式对本发明进行进一步的详细描述。需要说明的是,在不冲突的情况下,本申请的实施例及实施例中的特征可以相互组合。

[0033] 在下面的描述中阐述了很多具体细节以便于充分理解本发明,但是,本发明还可以采用其他不同于在此描述的方式来实施,因此,本发明的保护范围并不受下面公开的具体实施例的限制。

[0034] 图1示出了根据本发明的一个实施例的基于元数据模型的检索方法的流程图。

[0035] 如图1所示,根据本发明的一个实施例的基于元数据模型的检索方法,包括:

[0036] 步骤102,通过元数据模型将用户文档中的字段映射为索引字段;

[0037] 步骤104,对所述索引字段进行schema动态匹配,得到查询结果文档;

[0038] 步骤106,通过所述元数据模型将所述查询结果文档映射到所述用户文档的字段中。

[0039] 在该技术方案中,可以solr(一种独立的企业级搜索应用)服务前端添加一个元数据层,用来管理元数据模型,其接受并保存用户定义的元数据,同时可以通知使用元数据的元数据层去数据库中加载最新的元数据,由于元数据层是在索引集群之前,并且直接面向客户提供搜索和索引服务,故索引和搜索服务都会使用用户定义的元数据来进行验证。

[0040] 这样,通过元数据模型可以将用户用于检索的字段映射为索引字段,再对索引字段进行schema(可扩展标记语言架构)动态匹配,从而完成检索,并且,可以将检索得到的查询结果文档继续经由元数据模型映射到用户文档的字段,完成检索步骤。通过该技术方案,在用户文档类型发生变更后,直接更新元数据模型即可,而不必更新索引服务器中的相关配置并重新加载,减少了更新复杂度,使开发工作更加简单高效,实现了以较少的资源提供更稳定的服务。

[0041] 在上述技术方案中,优选地,所述元数据模型为客户端的应用程序接口与索引服务器之间的元数据层,所述用户文档来自所述客户端,所述schema动态匹配的服务由所述索引服务器提供。

[0042] 在该技术方案中,用户文档中具有客户端的用户用于检索的字段,solr(一种独立的企业级搜索应用)服务前端添加有一个元数据层,用来管理元数据模型,其接受并保存用户定义的元数据,同时可以通知使用元数据的元数据层去数据库中加载最新的元数据,由于元数据层是在索引集群之前,并且直接面向客户提供搜索和索引服务,故索引和搜索服务都会使用用户定义的元数据来进行验证,从而在用户文档类型发生变更后,直接更新元数据模型即可,而不必更新索引服务器中的相关配置并重新加载,减少了更新复杂度。

[0043] 在上述任一技术方案中,优选地,步骤102具体包括:根据所述元数据模型,确定所述用户文档所需的元数据,并在预设元数据库中为所述用户文档调用所述所需的元数据,其中,所述用户文档所需的元数据为所述用户文档到所述索引服务器的索引文档的映射关系。

[0044] 在该技术方案中,可以根据元数据层中的用户文档所需的元数据进行映射,使得用户文档中的字段可以映射为索引字段,从而进一步通过索引字段进行schema动态匹配,从而完成检索。

[0045] 在上述任一技术方案中,优选地,还包括:根据来自所述客户端的注解,为所述元数据模型定义全量元数据,并将定义的所述全量元数据同步更新至所述索引服务器;和/或根据来自所述客户端或所述索引服务器的修改命令,修改所述元数据模型的元数据。

[0046] 在该技术方案中,元数据的管理分为查询与更新两种,更新时又可分为全量更新和增量更新两种,其中,全量更新即通过客户端使用注解来定义全量元数据,同时在索引服务器启动时,将元数据更新到索引服务器,增量更新即根据来自客户端的应用程序接口或者服务端的网页界面的修改命令修改元数据中的相关内容。通过对元数据的更新,可不必更新索引服务器中的相关配置并重新加载,减少了更新复杂度,使开发工作更加简单高效,实现了以较少的资源提供更稳定的服务。

[0047] 在上述任一技术方案中,优选地,所述元数据模型包括文档模型和字段模型,其中,所述文档模型包括:文档类型、文档所属域和文档包含的字段属性集合,所述字段模型包括:所述用户文档的字段名称、索引类型和其他属性,所述其他属性包括:是否保存、是否索引、是否为拼音、是否为多值、是否为唯一和是否为只读。

[0048] 在该技术方案中,元数据是关于数据的描述性数据信息,用来定义用户文档到索引文档的一个映射关系,它有文档模型和字段模型两种。其中,文档模型包含了文档类型、文档所属域和文档包含的字段属性集合。字段模型包含了用户文档的字段名称、索引类型、是否保存、是否索引、是否拼音、是否多值、是否唯一和是否只读。其中,索引类型指的是字段在建索引和查询过程中使用的分词器类型,是否只读指的是限制某些字段写属性或者只能由后端服务来进行相应的更新。

[0049] 图2示出了根据本发明的一个实施例的基于元数据模型的检索装置的框图。

[0050] 如图2所示,根据本发明的一个实施例的基于元数据模型的检索装置200,包括:索引字段映射单元202、动态匹配单元204和查询结果映射单元206。

[0051] 其中,索引字段映射单元202用于通过元数据模型将用户文档中的字段映射为索



引字段;动态匹配单元204用于对所述索引字段进行schema动态匹配,得到查询结果文档;查询结果映射单元206用于通过所述元数据模型将所述查询结果文档映射到所述用户文档的字段中。

[0052] 在该技术方案中,可以solr(一种独立的企业级搜索应用)服务前端添加一个元数据层,用来管理元数据模型,其接受并保存用户定义的元数据,同时可以通知使用元数据的元数据层去数据库中加载最新的元数据,由于元数据层是在索引集群之前,并且直接面向客户提供搜索和索引服务,故索引和搜索服务都会使用用户定义的元数据来进行验证。

[0053] 这样,通过元数据模型可以将用户用于检索的字段映射为索引字段,再对索引字段进行schema(可扩展标记语言架构)动态匹配,从而完成检索,并且,可以将检索得到的查询结果文档继续经由元数据模型映射到用户文档的字段,完成检索步骤。通过该技术方案,在用户文档类型发生变更后,直接更新元数据模型即可,而不必更新索引服务器中的相关配置并重新加载,减少了更新复杂度,使开发工作更加简单高效,实现了以较少的资源提供更稳定的服务。

[0054] 在上述技术方案中,优选地,所述元数据模型为客户端的应用程序接口与索引服务器之间的元数据层,所述用户文档来自所述客户端,所述schema动态匹配的服务由所述索引服务器提供。

[0055] 在该技术方案中,用户文档中具有客户端的用户用于检索的字段,solr(一种独立的企业级搜索应用)服务前端添加有一个元数据层,用来管理元数据模型,其接受并保存用户定义的元数据,同时可以通知使用元数据的元数据层去数据库中加载最新的元数据,由于元数据层是在索引集群之前,并且直接面向客户提供搜索和索引服务,故索引和搜索服务都会使用用户定义的元数据来进行验证,从而在用户文档类型发生变更后,直接更新元数据模型即可,而不必更新索引服务器中的相关配置并重新加载,减少了更新复杂度。

[0056] 在上述任一技术方案中,优选地,索引字段映射单元202具体用于:根据所述元数据模型,确定所述用户文档所需的元数据,并在预设元数据库中为所述用户文档调用所述所需的元数据,其中,所述用户文档所需的元数据为所述用户文档到所述索引服务器的索引文档的映射关系。

[0057] 在该技术方案中,可以根据元数据层中的用户文档所需的元数据进行映射,使得用户文档中的字段可以映射为索引字段,从而进一步通过索引字段进行schema动态匹配,从而完成检索。

[0058] 在上述任一技术方案中,优选地,还包括:全量更新单元208,根据来自所述客户端的注解,为所述元数据模型定义全量元数据,并将定义的所述全量元数据同步更新至所述索引服务器;增量更新单元210,根据来自所述客户端或所述索引服务器的修改命令,修改所述元数据模型的元数据。

[0059] 在该技术方案中,元数据的管理分为查询与更新两种,更新时又可分为全量更新和增量更新两种,其中,全量更新即通过客户端使用注解来定义全量元数据,同时在索引服务器启动时,将元数据更新到索引服务器,增量更新即根据来自客户端的应用程序接口或者服务端的网页界面的修改命令修改元数据中的相关内容。通过对元数据的更新,可不必更新索引服务器中的相关配置并重新加载,减少了更新复杂度,使开发工作更加简单高效,实现了以较少的资源提供更稳定的服务。

[0060] 在上述任一技术方案中,优选地,所述元数据模型包括文档模型和字段模型,其中,所述文档模型包括:文档类型、文档所属域和文档包含的字段属性集合,所述字段模型包括:所述用户文档的字段名称、索引类型和其他属性,所述其他属性包括:是否保存、是否索引、是否为拼音、是否为多值、是否为唯一和是否为只读。

[0061] 在该技术方案中,元数据是关于数据的描述性数据信息,用来定义用户文档到索引文档的一个映射关系,它有文档模型和字段模型两种。其中,文档模型包含了文档类型、文档所属域和文档包含的字段属性集合。字段模型包含了用户文档的字段名称、索引类型、是否保存、是否索引、是否拼音、是否多值、是否唯一和是否只读。其中,索引类型指的是字段在建索引和查询过程中使用的分词器类型,是否只读指的是限制某些字段写属性或者只能由后端服务来进行相应的更新。

[0062] 图3示出了根据本发明的一个实施例的客户端、元数据层与索引集群的交互示意图。

[0063] 如图3所示,客户端通过应用程序接口与元数据模型进行交互,具体地,客户端通过元数据模型的搜索服务和/或索引服务与索引集群进行交互,从而完成检索操作。

[0064] 元数据模型作为隔离层置于客户端和索引服务器之间,由于有隔离层的存在,索引集群可以更容易地扩展,无论是横向还是纵向,前端的索引服务器都不会感知。元数据管理单元是专门用来管理元数据模型的,其接受并保存用户定义的元数据,同时可以通知使用元数据的隔离层去数据库中加载最新的元数据。

[0065] 由于元数据模型是在索引集群之前,并且直接面向客户提供搜索和索引服务,故索引和搜索服务都会使用用户定义的元数据来进行验证,从而在用户文档类型发生变更后,直接更新元数据模型即可,而不必更新索引服务器中的相关配置并重新加载,减少了更新复杂度。

[0066] 元数据的管理分为查询与更新两种,更新时又可分为全量更新和增量更新两种,其中,全量更新即通过客户端使用注解来定义全量元数据,同时在索引服务器启动时,将元数据更新到索引服务器,增量更新即根据来自客户端的应用程序接口或者服务端的网页界面的修改命令修改元数据中的相关内容。通过对元数据的更新,可不必更新索引服务器中的相关配置并重新加载,减少了更新复杂度,使开发工作更加简单高效,实现了以较少的资源提供更稳定的服务。

[0067] 需要说明的是,在索引节点中,core(处理器)的数量太多,会影响到节点启动速度,为减少core的数量,可以将一部分用户应用的数据映射到同一个core中,这样能提高节点的使用效率。

[0068] 元数据是关于数据的描述性数据信息,用来定义用户文档到索引文档的一个映射关系,它有文档模型和字段模型两种。其中,文档模型包含了文档类型、文档所属域和文档包含的字段属性集合。字段模型包含了用户文档的字段名称、索引类型、是否保存、是否索引、是否拼音、是否多值、是否唯一和是否只读。其中,索引类型指的是字段在建索引和查询过程中使用的分词器类型,是否只读指的是限制某些字段写属性或者只能由后端服务来进行相应的更新。

[0069] 索引集群中,可以根据预先定义的分词类型定义相关的动态字段类型和一些必要的内部使用的字段名称,如\_uid\_、\_type\_、\_domain\_,这些是用来标示文档的唯一ID(地

址)、类型和所属域,为每个文档的都需要的必要字段。

[0070] 其中,\_uid\_是由多个字段合并组成,即应用该类型文档的唯一字段,在与应用的ID组合而成。在建索引时,用户文档通过该用户元数据模型检验,把用户文档字段映射到索引字段,索引字段再跟索引中schema中动态字段匹配。同时,需要将不符合元数据模型的字段从文档中剔除,对相关找不到对应的元数据文档抛出异常。搜索时,索引服务器将接收和解析用户的查询语句,根据用户检索文档的元数据模型,将语句中的字段映射为索引字段,然后进行查询,最后,将查询返回的文档内容重新映射到用户文档字段。

[0071] 例如,用户搜索name:北京,通过元数据映射转换为name\_s\_s,匹配索引schema类型\*\_s\_s;索引服务器返回相关的文档name\_s\_s:北京天安门,再通过元数据映射回用户原有字段名称name:北京天安门。建索引的过程也与之类似。s\_s对应了用户字段的类型和创建索引使用的分词器两种,这样,用户为达到自己想要的搜索效果选择不同的类型,很好地隐藏了索引字段对用户的感知,在用户文档类型发生变更后,直接更新元数据模型,并使之生效即可,而不必更新索引中相关的配置并重新加载,减少了模型变化后更新的复杂度。

[0072] 以上结合附图详细说明了本发明的技术方案,通过本发明的技术方案,在用户文档类型发生变更后,直接更新元数据模型即可,而不必更新索引服务器中的相关配置并重新加载,减少了更新复杂度,使开发工作更加简单高效,实现了以较少的资源提供更稳定的服务。

[0073] 以上所述仅为本发明的优选实施例而已,并不用于限制本发明,对于本领域的技术人员来说,本发明可以有各种更改和变化。凡在本发明的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

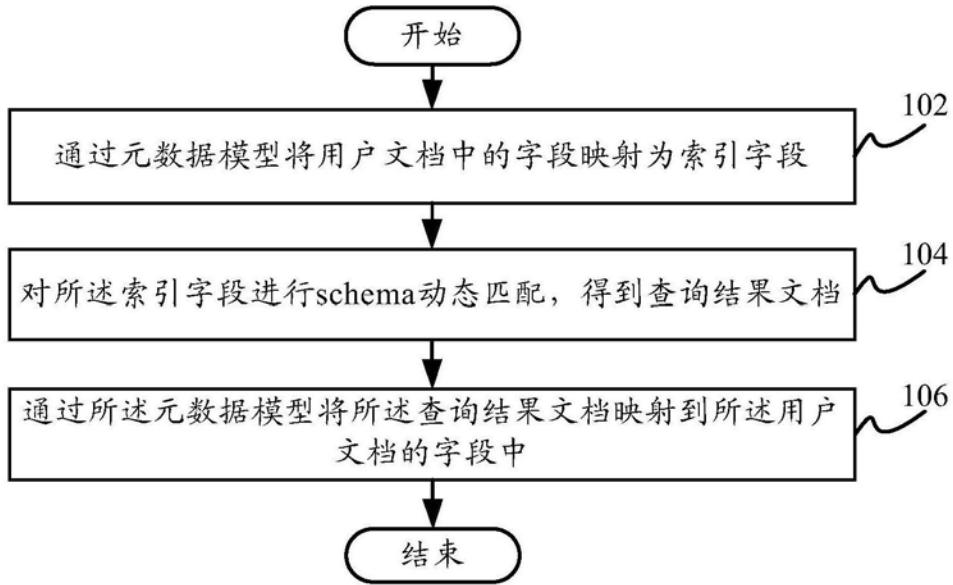


图1



图2

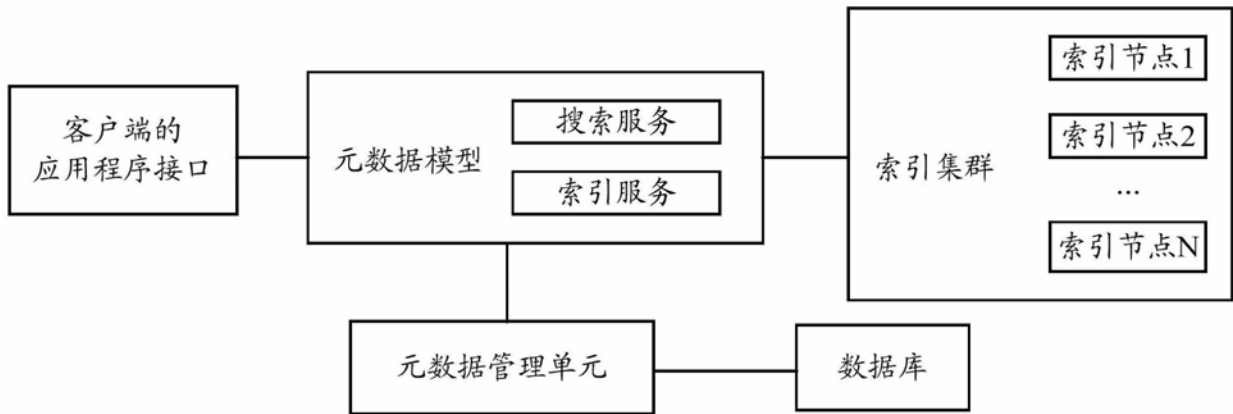


图3