

# 一种基于本体 KNN 的分布式缓存数据交换策略

王立<sup>1</sup> 王欣<sup>1</sup> 马朝东<sup>2</sup>

(国家开放大学计算机学院 北京 100039)<sup>1</sup> (天津广播电视大学教务处 天津 300191)<sup>2</sup>

**摘要** 以国家开放大学教务管理系统为例,以减少数据资源获取的时间开销以及提高数据质量作为目标,提出了一种基于本体 KNN 的分布式缓存数据交换策略,用于解决分布式系统在不同节点之间进行数据交换时产生的性能优化问题。仿真实验结果表明,该策略具有较为出色的优化访问性能,可以实现数据交换过程的进一步优化,进而提升系统的整体性能,具有一定的实用价值。

**关键词** 分布式缓存, KNN 算法, 本体, 数据交换

中图分类号 TP393 文献标识码 A

## Distributed Caching Strategy for Data Exchange Program Based on Ontology and KNN Algorithm

WANG Li<sup>1</sup> WANG Xin<sup>1</sup> MA Chao-dong<sup>2</sup>

(School of Computer Science, Open University of China, Beijing 100039, China)<sup>1</sup>

(Dean's Office, Tianjin Open University, Tianjin 300191, China)<sup>2</sup>

**Abstract** This paper gave an example of the management system of Open University of China, and proposed a distributed cache strategy for data exchange program based on ontology and KNN algorithm to solve the optimization problems in access time, cost and resources quality when distributed system exchanges data between the different nodes. The correctness and effectiveness of the presented strategy is demonstrated by a simulation application designed in the paper, and results show that the strategy can improve the efficiency of data exchanging in distributed system.

**Keywords** Distributed cache, KNN, Ontology, Data exchange

目前,许多大型信息系统的业务数据量随着业务的发展呈现急剧增长的趋势,但是当前的网络带宽以及 IT 基础设施的建设无法跟上,这导致在多个环节降低了系统的综合工作效能,比如在数据交换环节,主要的现象有:数据交换效率较低,通信两端业务操作速度缓慢,出现通信安全等问题,数据交换的性能成为了影响系统整体性能的主要瓶颈之一。因此,建立通用的、快速而安全的、高效而可靠的数据交换机制是确保当前分布式系统之间有效集成和高效互操作的关键问题。目前分布式缓存技术<sup>[1]</sup>已经广泛应用于大型信息系统的建设中。本文以国家开放大学教务管理系统为例,对分布式系统中数据交换和系统集成问题进行了深入研究,以数据资源获取的时间、开销成本以及数据质量作为目标,设计出基于本体的 KNN(K 最邻近节点算法)匹配算法,提出一种基于分布式缓存技术的数据交换过程优化策略,使现有分布式系统的集成和互操作等工作变得更加容易与高效,为分布式应用中跨平台通信技术的应用提供了可参考的解决方案。

### 1 应用场景

国家开放大学教务管理系统的场景分布结构如图 1 所示。图中的节点分别代表了中央、地方和分校三级管理平台,通过 Internet 实现互连,每个节点上都部署了各自的业务系统,各级平台中的数据既有独立性又有相关性。

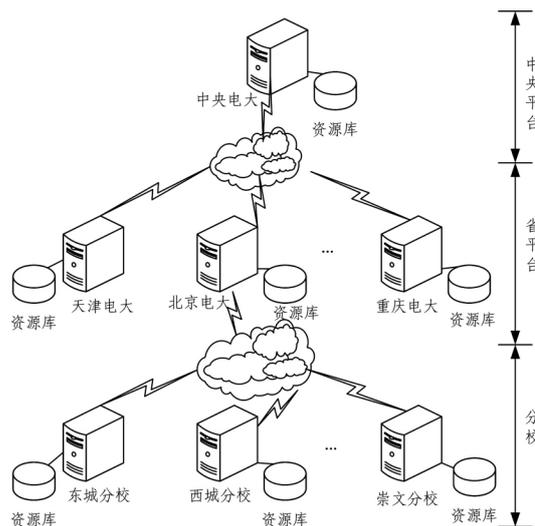


图 1 国家开放大学教务管理系统分布结构图

其中,中央平台的数据库不仅具有本平台自己的数据,同时还是各省平台数据的全集;各省平台的数据库不仅具有本平台自己的数据,同时还有一些全国电大公有的数据,是中央平台数据库的子集。省平台和中央平台之间要进行频繁的数据交换以完成特定的教务任务,例如每年各省都要将各自的学生选课数据传输到中央,中央进行汇总,然后根据学生所选

本文受天津市教育科学规划课题:远程教育环境下基于内容感知的移动学习模式研究(CEYP6001)资助。

王立(1978—),男,博士,讲师,主要研究方向为分布式计算, E-mail: wlpolo@crtvu.edu.cn;王欣(1976—),女,硕士,副教授,主要研究方向为计算机网络;马朝东(1976—),男,硕士,副教授,主要研究方向为分布式计算。

学分制定收费信息再下发到各省。在这个过程中,系统中各平台之间的数据交换非常频繁,数据的交换量很大,且对数据交换的实时性要求较高。为了满足这些要求,本文基于分布式缓存技术设计了数据交换方案。

## 2 分布式缓存数据交换方案的设计

分布式系统间的数据交换是指不同计算机应用系统之间相互发送与传递有意义、有价值的信息[2],其核心问题是通信的方式和信息的标准化。对于端到端的数据交换来说,端操作的效率会直接影响数据交换的效率,如今以效率为首位的现实世界中,涌现出很多提高端操作效率的技术,比如现在已经在IT行业中得到广泛应用的分布式缓存技术。分布式缓存由一个服务端实现管理和控制,有多个客户端节点存储数据,可以进一步提高数据的读取速率。常用的分布式缓存策略一般采用一致性哈希算法来实现[3],确定数据的存储和读取节点。以数据 $D$ 和节点总个数 $N$ 为基础,通过一致性哈希算法计算出数据 $D$ 对应的哈希值(相当于门牌号),根据这个哈希值就可以找到对应的节点了。一致性哈希算法的好处在于节点个数发生变化(减少或增加)时无需重新计算哈希值,可保证数据储存或读取时可以正确、快速地找到对应的节点。

### 2.1 方案设计

在教务管理系统中,通常会用到数据库服务器。由于数据库服务器与业务服务器的距离较远,在系统运行的过程中,数据库的访问通常会成为整个系统性能的瓶颈。数据库每次和服务器之间的数据传输都需要将数据放在网络上进行序列化与反序列化。在这个过程中,网络数据交换的代价是不能忽略的,而且数据库很难像Web服务器或应用服务器那样通过集群轻松地提高可伸缩性。因此需要利用分布式缓存技术将应用程序与数据库的距离拉近,将大量的数据拉到应用程序层,这样大部分的访问在应用程序层就被拦截掉了。在访问经常使用的数据时,缓存技术能够呈数量级地减少潜在的访问时间。这样的结果不仅大大提高了应用程序的速度,也减轻了数据库服务器的负载,使数据库服务器将时间集中在业务处理上,从而大大提高数据库服务器的访问性能。

该本方案中,系统采用主从式进行网络结构的构建,即网络节点中存在一个主节点负责处理收到的请求并同时负责任务的调度分配,从节点负责数据的交换,各个节点中应用分布式缓存技术,将符合传输条件的所有数据一次性保存在缓存服务器中[4]。这样,应用程序在需要数据的时候,所面对的是缓存,方案如图2所示。

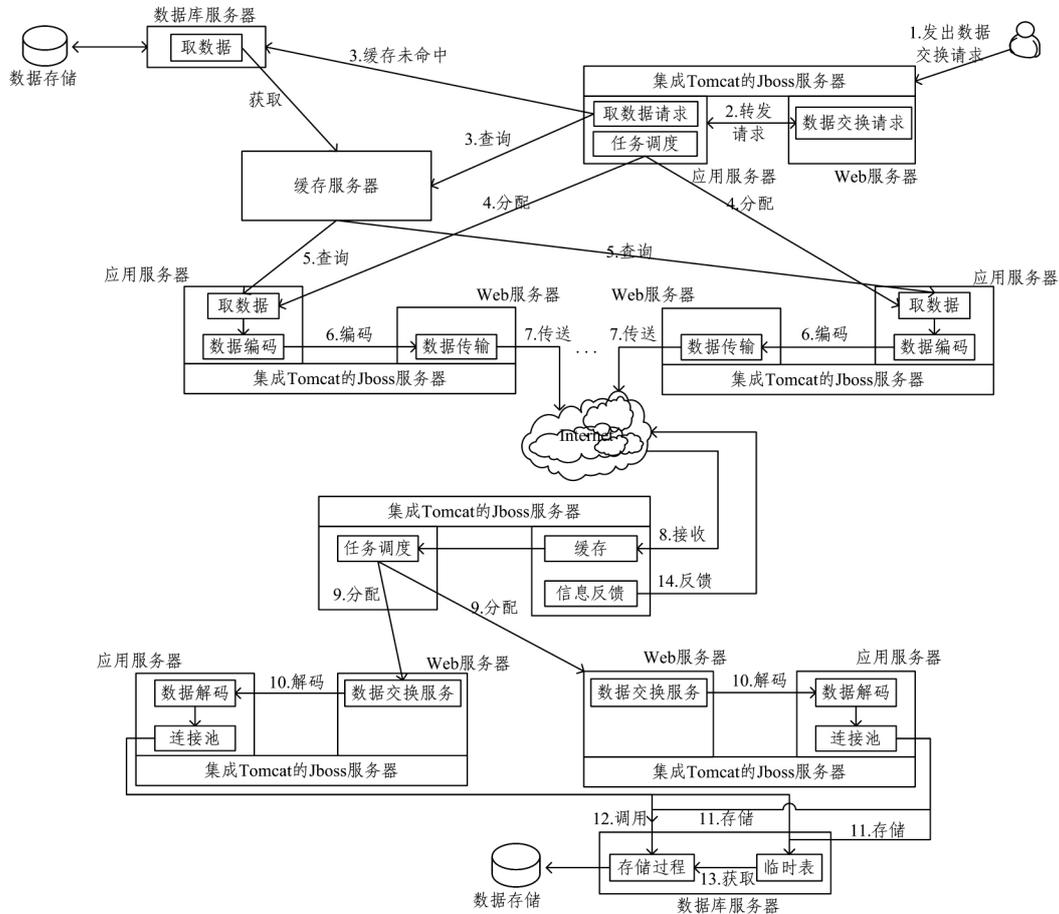


图2 基于分布式数据缓存技术的数据交换方案

一次数据交换的流程为:

- 1) 客户向Web服务器发出数据交换请求。
- 2) Web服务器将请求交与应用服务器处理(主节点)。
- 3) 应用服务器根据用户所选择的传输条件查询缓存是否有相应的数据,如果缓存未命中,则从数据库服务器中获取,将数据存入缓存服务器中。

- 4) 待数据获取完成后,主节点根据各个从节点的负载情况进行任务分配。
- 5) 被分配到任务的从节点根据任务标识,从缓存服务器中取得数据。
- 6) 从节点根据所使用的传输技术(比如Web服务技术)将数据编码成本地格式数据。

- 7) 从节点通过 Internet 传输数据。
- 8) 数据接收端应用服务器(主节点)负责接收每批数据。
- 9) 接收端应用服务器根据从节点的负载情况进行任务分配。
- 10) 被分配到任务的从节点对收到的数据进行解码。
- 11) 从节点将解码后的数据通过数据库连接池存储在临时表中。
- 12) 接着通过连接池调用存储过程。
- 13) 存储过程从临时表获取数据,然后完成数据的筛选和入库操作。
- 14) 操作完成后,通过主节点向发送方发送传输成功标识,发送方在收到传输成功标识后记录完成传输的批数,同时自动传输下一批;如果未收到传输成功标识,视为传输失败,记录失败断点,下次从断点处重传。

## 2.2 基于本体的 KNN 匹配算法

该方案仍然存在数据交换的瓶颈问题,主要压力来自于主节点。如果网络出现拥塞或主节点出现负载过重的情况,就会使得从节点收到来自主节点的信息的速度变慢,在影响整个数据交换的同时也使得从节点服务器 CPU 闲置率较高,同时会影响服务器处理其他用户的请求。因此,本文结合本体技术设计基于 KNN 的匹配算法<sup>[5]</sup>来提高数据交换的效率。

首先,根据节点的信息,比如节点名称、节点标识、IP 地址、MAC 地址、操作系统版本、业务索引信息等,利用本体技术对这些信息进行组织并建立本体模型。模型的建立同样要用到项目(数据交换请求)本身的语义信息,各种应用环境中的项目具有的属性不尽相同。但是,不管提供的是何种类型的属性,在将各类属性进行符号化之后,都可以用向量模型的方式进行表示:

$$t_i = (f_{i_1}, f_{i_2}, f_{i_3}, \dots, f_{i_n}) \quad (1)$$

这样,在分布式环境下,可以有效地组织缓存节点的信息,细化从节点的用户群体,可以提供更具有专指性的匹配和检索服务,使得在分布式环境下数据交换的过程更加高效与合理。本系统采用分布式本体集成框架下虚拟本体构建方法进行节点本体信息的构建<sup>[6]</sup>,基于 KNN 算法从用户(源节点)方面来预测目标用户(目的节点)对目标项目(数据交换)的评分值。

首先利用节点间的整体相似性,选择与目标节点相似性最大的  $K$  个潜在节点作为其  $K$ -近邻,然后根据这  $K$  个邻居节点对数据交换过程的评分来预测目的节点对数据交换请求项目的评分:

$$r_{ij}^u = \bar{r}_i + \frac{\sum_{k \in KNN(i)} Sim_u(i, k) \times (r_{kj} - \bar{r}_k)}{\sum_{k \in KNN(i)} Sim_u(i, k)} \quad (2)$$

其中,  $r_{ij}^u$  表示采用基于节点的协同过滤得到的节点  $i$  对数据交换请求项目  $j$  的预测评分值,  $Sim_u(i, k)$  表示节点  $i$  和节点  $k$  之间的整体相似性,  $KNN(i)$  表示节点  $i$  的  $K$ -近邻集,其它参数同前。融合了节点本体语义信息的协同过滤算法,缓和了传统基于用户的协同过滤算法中的数据稀疏性问题。

其中,  $f$  为表示数据交换请求项目  $i$  的语义向量模型的  $j$  个属性,  $n$  为数据交换请求项目语义向量模型包含的属性个数。例如:在业务报送系统中,信息的属性包括组织机构、关

键词、人物、业务编号、责任者和时间等,形成数据交换的语义向量。

最后,按照各数据交换请求项目语义属性的重要性为其分配适当的权重,采用线性加权合并的方法计算数据交换请求项目间的语义相似性:

$$Sim_i^s(i, j) = \frac{\sum_{k=1}^n \omega_k' \times Sim_k'(i, j)}{\sum_{k=1}^n \omega_k'} \quad (3)$$

其中,  $Sim_i^s(i, j)$  表示数据交换请求项目  $i$  和  $j$  在语义内容上的相似性,  $Sim_k'(i, j)$  表示项目  $i$  和  $j$  在第  $k$  个属性上的相似性,  $\omega_k'$  为第  $k$  个数据交换请求项目语义属性的重要性权重,  $n$  为数据交换请求项目语义向量模型中包含的语义属性的个数。

基于本体的 KNN 匹配算法的流程如下。

- Step1 准备数据,引入本体信息(RDF 三元组)及规则,对数据进行预处理。
- Step2 选用合适的数据结构存储训练数据和测试元组。
- Step3 设定一个本体间的初始关联程度,定义为  $k$ 。
- Step4 然后维护一个大小为  $k$  的按距离由大到小的优先级队列,用于存储最近邻训练元组。随机从训练元组中选取  $k$  个元组作为初始的最近邻元组,分别计算测试元组到这  $k$  个元组的距离,将训练元组标号和距离存入优先级队列。
- Step5 遍历训练元组集,计算当前训练元组与测试元组的距离,将所得距离  $L$  与优先级队列中的最大距离  $L_{max}$  进行比较。若  $L \geq L_{max}$ ,则舍弃该元组,遍历下一个元组;若  $L < L_{max}$ ,则删除优先级队列中最大距离的元组,将当前训练元组存入优先级队列。
- Step6 删除同一类别的元组,减少访问次数。
- Step7 遍历完毕,计算优先级队列中  $k$  个元组的多数类,并将其作为测试元组的类别。
- Step8 测试元组集测试完毕后,计算误差率,继续设定不同的  $k$  值重新进行训练,最后取误差率最小的  $k$  值。
- Step9 找到最佳的匹配节点。

## 3 方案的实现及性能分析

### 3.1 方案的实现

方案中共有两种类型的节点,分别为承担主控功能的管理节点和负责数据交换的计算节点。在这两种类型的节点相互协调工作时,各节点间涉及到通信的问题,同时管理节点在进行任务分配时需要获得各从节点的状态,包括网络连接状态、机器运行信息和负载信息等。节点间采用 JGroups 进行通信。JGroups 是一个用于建立可靠的组播通信的工具包。以下是本文中应用 JGroups 进行通信的关键代码。

```
JChannel channel = new JChannel(); // 创建 channel
MessageDispatcher dispatcher = new MessageDispatcher(channel,
null, null)
new RequestHandler() { // 同步接收并处理消息
public Object handle(Message msg) {
处理代码;
// 引入 KNN 节点匹配算法
matchKNN(sourceIP, TargetIP, channel, session);
}
}
```

```
channel.connect("DataTransfer");//连接到名为 DataTransfer 的集群
//发送消息
RspList rl=dispatcher.castMessage(null,msg,GroupRequest.GET_
NONE,0);
```

从节点发送的消息格式为:[本机 ip & 发送目的标识 & 批数 & 成功标识],其中发送目的标识为 1 代表返回消息,2 代表发送消息,成功标识为 c1 代表发送成功,c2 代表发送失败,批数为数据交换的批号;主节点发送的消息格式为:[本机 ip & 发送目的标识 & 批数 & 目的从机 ip],其中,发送目的标识为 z1 表示发送消息。

### 3.2 性能分析

数据交换客户端和服务端的所有服务器均处于各自的同一局域网内。测试数据中客户端数据库的本地数据量为 78 万条,服务器端数据库本地数据量为 630 万条,数据交换量为 16 万条。

在局部网环境下,使用 5 台服务器搭建分布式缓存测试环境,并通过 VMWARE 软件进行虚拟节点的安装与配置,各个虚拟节点采用桥接的模式通过服务器桥接到局域网环境中。

实验分别采用 3 种方案:

方案 1 原系统所采用的数据交换方案。

方案 2 采用一致哈希算法分布式缓存方案。

方案 3 采用基于本体的 KNN 算法改进的分布式缓存数据交换方案。

具体实验配置见表 1。

实验	节点数	仿真次数
1	10	10
2	100	10

在两个实验中,3 个方案各运行 10 次,每次的数据都经过校验,确认传输完整且成功,然后计算每个方案数据交换的实际时间的平均值。数据交换方案的测试结果如图 3、图 4 所示。

由图 3、图 4 可知,在节点数较少的情况下,方案 2 与方案 3 相差不大,随着节点数的增多,方案 3 在用时方面略大于方案 2。排除网络及测试误差等因素后,本文提出的基于本体 KNN 的分布式缓存数据交换策略能够提升系统的数据交

换效能,具有一定的实用价值。

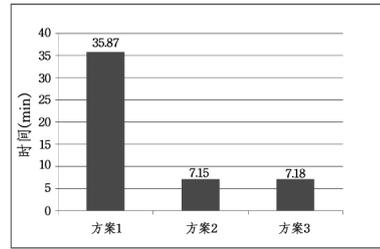


图 3 实验 1 数据交换方案的测试结果

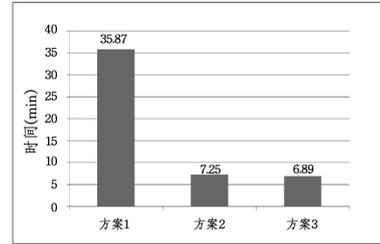


图 4 实验 2 数据交换方案的测试结果

结束语 本文以国家开放大学教务管理系统为例,以数据资源获取的时间开销,以及数据质量作为目标,设计了基于本体 KNN 的分布式缓存数据交换策略能。仿真实验结果表明,该策略可以进一步优化数据交换过程,进而提升系统的整体性能,具有一定的实用价值。

### 参考文献

- [1] 秦秀磊,张文博,魏峻,等. 云计算环境下分布式缓存技术的现状与挑战[J]. 软件学报,2013,24(1):50-66
- [2] 罗军,陈席林,李文生. 高效 Key-Value 持久化缓存系统的实现[J]. 计算机工程,2014,40(3):33-38
- [3] 奉国和,吴敬学. KNN 分类算法改进研究进展[J]. 图书情报工作,2012,56(21):97-100
- [4] Min Jun-ki, Lee M Y. DICE: An Effective Query Result Cache for Distributed Storage Systems [J]. Journal of Computer Science & Technology, 2010, 25(5): 933-944
- [5] 胡丽聪,徐雅静,徐惠民. 基于动态反馈的一致性哈希负载均衡算法[J]. 微电子学与计算机,2012,29(1):177-180
- [6] 王立,邱瑞华. 分布式本体集成框架下虚拟本体集合动态生成算法[J]. 计算机工程与设计,2010,31(24):5183-5186

(上接第 307 页)

- [17] Efe K. The Crossed cube architecture for parallel computation [J]. IEEE Transactions on Parallel and distributed systems, 1998,3(5):513-523
- [18] Hwang, Briggs F A. Computer architecture and Parallel processing[M]. McGrawHill College,1984
- [19] Hsu L H, Lin C K. Graph theory and Interconnection networks [M]. New York;CRC Press,2009
- [20] Bhuyan L N, Agrawal D D. Generalized hypercube and hyperbus structures for a computer network[J]. IEEE Transactions on Computer,1994,33(4):323-333
- [21] 师海忠. 正则图连通圈:多种互连网络的统一模型[C]//中国运筹学会第十届学术交流会论文集. 北京,2010:202-208
- [22] 师海忠,牛攀峰,马继勇,等. 互连网络的向量图模型[J]. 运筹学学报,2011,15(3):115-123

- [23] 师海忠,路建波. 关于互连网络的几个猜想[J]. 计算机工程与应用,2008,44(31):112-115
- [24] 师海忠. 互连网络的新模型:多部群论模型[J]. 计算机科学,2013,40(9):21-24
- [25] 师海忠. 几类新的笛卡尔乘积互连网络[J]. 计算机科学,2013,40(6A):265-270
- [26] Shi Hai-zhong, Shi Yue. A new model for interconnection network: k-hierarchical ring and r-layer graph network[OL]. http://vdisk. weibo. com/s/dliJyfer2-2l
- [27] Shi Hai-zhong, Shi Yue. A hierarchical ring group-theoretic model for interconnection networks[OL]. http://vdisk. weibo. com/s/dlizjyfeBX-2J
- [28] Shi Hai-zhong, Yue Shi. Cell-breeding graph model for interconnection networks [OL]. http://vdisk. weibo. com/s/dlizjyfesb05y