

文章编号: 2095-2163(2021)01-0046-06

中图分类号: TP391.1

文献标志码: A

一种基于空间文本信息流的分布式的发布订阅算法

周泽宁

(哈尔滨工业大学 计算机科学与技术学院, 哈尔滨 150001)

摘要: 发布订阅系统是进行发布的事件和订阅消息之间的匹配系统。首先需要对订阅消息进行聚类操作, 按照聚类结果, 找到事件所属类别, 随后在类别中, 找寻和事件匹配的订阅。本文提出了一个即时的发布订阅的算法, 统筹空间信息和事件属性信息, 不仅可以即时地处理事件和订阅的匹配操作, 也可以在分布式环境下即时地进行订阅的更新和类别的更新。并且可以在没有先验知识的情况下即时地进行聚类操作和匹配操作。设计一个分布式的系统, 将发布订阅算法部署其上, 并且提出了在分布式系统上该算法的负载均衡策略。随后通过自建集群, 使用真实的数据, 实验验证本文提出的发布订阅算法。

关键词: 发布订阅系统; 分布式系统; 聚类算法

A distributed publish-subscribe algorithm based on spatio-text data stream

ZHOU Zening

(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

[Abstract] The publish and subscribe system is a matching system between published events and subscribed messages. First, it is necessary to perform a clustering operation on the subscription messages, find the category to which the event belongs according to the clustering result, and then find the subscription that matches the event in the category. This paper proposes an instant publish and subscribe algorithm, which coordinates the spatial information and event attribute information. It can not only handle the matching operation of events and subscriptions in real time, but also update subscriptions and categories in a distributed environment. In addition, clustering and matching operations can be performed immediately without prior knowledge. A distributed system is designed, the publish-subscribe algorithm is deployed on it, and the load balancing strategy of the algorithm is put forward on the distributed system. Therefore through the self-built cluster, using real data, the publish and subscribe algorithm proposed in this article is verified in the experiments.

[Key words] publish and subscribe system; distributed system; clustering algorithm

0 引言

发布订阅系统主要目的是方便事件发布者和事件订阅者在网上进行信息交换这一过程, 事件的订阅者持续关注某一特定区域内的特定事件, 事件的发布者将事件发生的空间位置和事件发布到网上, 随后这些事件和事件订阅者的订阅信息进行匹配并将符合要求的事件推送给订阅者。

中央服务器上基于空间属性信息的发布订阅算法^[1-8], 使用树状结构按照空间索引订阅数据, 以加速空间属性事件和空间属性订阅的比较。许多分布式系统^[9-12]使用现有的空间索引将数据划分到不同的服务器, 这些系统是基于静态数据的一次性查询。

本文提出了一个没有先验知识、可以即时处理数据流、统筹事件信息和空间信息的发布订阅系统, 满足了对于灵敏度的要求, 并且提出了一个分布式的系统, 将基础的发布订阅系统部署其上, 满足了大

规模的数据量的要求。

1 基础发布订阅系统

发布订阅系统包含发布的事件和订阅这两种数据流。在研究中, t 表示一个发布的事件, 包含至少 2 个信息: 空间位置和属性文本, 可用二元组来表示: $(t.s, t.k)$ 。其中, $t.s$ 表示该事件发生的空间位置, 空间位置采用经纬度的表达方式, 表示空间中的一个点。 $t.k$ 表示该事件的一系列属性。同样, q 表示一个订阅消息, 用二元组来表示: $(q.s, q.k)$ 。其中, $q.s$ 表示该订阅发布者关注的空间范围, 该数据使用的是此次订阅关注范围的最小邻接矩形。 $q.k$ 表示订阅者持续关注的属性信息, 该信息是事件匹配时的属性要求。

发布订阅系统所关注的是事件和订阅的匹配过程。如果满足 $t.s$ 在 $q.s$ 的范围中, 即事件的发生位置在订阅的要求的空间范围之内, 并且 $q.k \subset t.k$,

也就是事件的属性包含了订阅要求的事件属性,则认定事件 t 满足该订阅 q 的要求,就可将事件 t 分配给该查询 q 。

为了节省大量不必要的匹配操作,本文先是将订阅进行聚类操作,将其按照空间信息和事件属性通过聚类操作获得众多类别 $c_i(i = 1, 2, 3 \dots)$, 对到来的事件,将其分配到订阅类别,再在匹配成功的类别中对订阅和事件进行匹配操作,具体如图 1 所示。

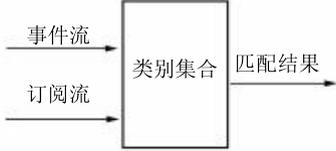


图 1 基础的发布订阅系统示意图

Fig. 1 Schematic diagram of the basic publish and subscribe system

事件流进入类别集合后,首先找到相匹配的类别,再和类别中的订阅一一匹配,找到相匹配的订阅输出。订阅流进入类别集合后,如果进行订阅的增加操作,就通过聚类算法更新类别集合;如果是订阅的删除操作,就找到该订阅所属类别,进行删除操作。类别的集合最初为空,随着订阅的到来,逐渐通过聚类算法形成类别。

2 发布订阅系统的聚类算法

对于发布订阅系统,至关重要的就是订阅的聚类算法,因其直接决定聚类结果的优劣,并最终影响整个算法的反应速度。本文提出一个没有训练集(在进行订阅和事件的匹配过程中,同时进行训练)、即时的聚类算法 Rt-Cluter (RealTime-Cluter),可以对接受到的事件和订阅进行即时的处理。通过对比实验显示了本算法的有效性。

订阅和事件包含 2 种数据,分别是:空间信息和属性信息。本文采取一个融合 2 种策略的混合策略。面对空间信息,文中采取网格的结构来存储空间数据。面对事件的属性信息,文中采取倒排索引的结构。其存储形式如图 2 所示。

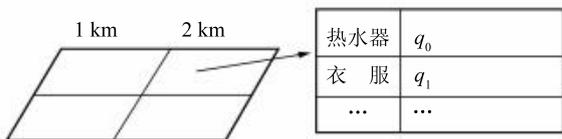


图 2 数据结构

Fig. 2 Data structure

Rt-Cluter 算法使用相似性和相关性作为判断标准。其中,相似性主要解决聚类操作中,对订阅的

聚类判断问题。当一个新的需要添加的订阅 q 到来时,需要按照空间信息和事件的属性信息对其进行聚类操作,将其分配给最相似的一个类别 c 。本文将会分别计算属性相似性和空间相似性,并对这 2 个相似性采用加权和的方式进行融合。属性相似性 $KeySim$ 的计算公式如下:

$$KeySim = \sum_{key \in q, k} key.pro, \quad (1)$$

其中, key 表示属性, $key.Pro$ 表示属性 key 在类别 c 的属性集合 $c.k$ 中所占的比例,即类别 c 中属性 key 的频数和类别 c 中所有属性的频数之比。

相应地,空间相似性 $SpatialSim$ 的计算需要用到如下公式:

$$SpatialSim = \frac{ex}{c.s}, \quad (2)$$

其中, ex 表示如果类别 c 添加订阅 q , $c.s$ 需要扩展的大小, $c.s$ 表示类别 c 在添加订阅 q 之前的最小邻接矩形的面积大小。

综上可得,进行相似性的比较时,总体相似性 Sim 的计算,要同时考虑空间和属性的相似性,其数学公式可写为:

$$Sim = \alpha * KeySim / ((1 - \alpha) * (1 + SpatialSim)). \quad (3)$$

其中, α 表示权值 ($0 \leq \alpha \leq 1$),可以根据对属性或者空间的重视程度进行调整。

对于等待被分配的订阅,将其分配给相似值最大的类别。但是如果最大的相似度仍然足够小(小于一个阈值),本文则为其创造一个新类别,并将该新类别分配到与其最为相似的类别所在的工作节点中,如此一来就保证了最相似的类别将处在同一个工作节点中,在后续数据传输上节省了资源。

在聚类算法中,除了新类别的创建,还需要进行类别融合,使用阈值 $CMaxNumTh$ 来表示所能拥有的最大类别数量,一旦类别数量超过该值,就要进行类别融合操作。本文使用相关性来做类别融合研究。总体而言,如果 2 个订阅 q_0 和 q_1 被同一个发布的事件 t 所匹配并且匹配成功,那么就认为这 2 个订阅具有一点相关性,即: $Correlation(q_0, q_1) = 1$ 。需要进行类别融合操作时,将相关性最大的 2 个类别进行融合。

3 分布式系统设计与部署

网络中数据量不断扩大,单处理器无法满足现实要求,本文提出了一个分布式系统来承载发布订阅算法,将分布式系统下的发布订阅算法命名为

DRt-Cluster(Distributed-Realtime-Cluster)。其逻辑结构如图3所示。

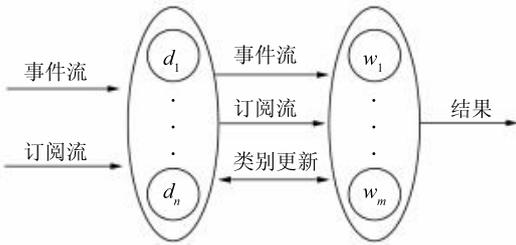


图3 分布式系统逻辑结构

Fig. 3 Logical structure of distributed system

该系统包含有2类节点,分别是聚类节点 $d_i(0 < i \leq n)$ 和匹配节点 $w_j(0 < j \leq m)$ 。其中,聚类节点使用 $d_i(0 < i \leq n)$ 表示,匹配节点使用 $w_j(0 < j \leq m)$ 表示。研究内容具体如下。

(1) 聚类节点。主要进行聚类操作,功能如下:

① 对于事件 t ,在聚类节点中,同各个类别的特征相比较,找寻可能存在与其相匹配的订阅所在的类别。

② 对于订阅的增加,为其找寻归属的唯一类别,进行类别更新的相应操作。

③ 对于订阅的删除,需要找寻可能包含该事件的类别,并进行删除操作。

(2) 匹配节点。节点中分别保存着各个类别,主要进行事件的匹配和订阅的增删操作。功能如下:

① 对于事件,会根据其所从属的订阅类别,寻找符合匹配条件的订阅。

② 对于订阅的增删操作,根据其所从属的订阅类别,进行类别的更新。

③ 类别的更新,按照聚类节点对类别的更新信息,相应地更新本节点内的类别。

将算法应用到分布式系统上需要注意数据传输问题。此外,在分布式系统中还应注意负载均衡的问题,以避免某一个节点负载过重,成为性能瓶颈。

3.1 类别融合

分布式环境下,需要在满足聚类要求的情况下,尽量减少节点间的数据传输,同时根据相关性,本文将优先进行同一节点内的类别融合。过程中拟用到2个变量: CS (correlation in the same worker) 和 CD (correlation in the different worker)。

研究中可得, $c. cs$ 表示类别 c 和其同一匹配节点中的类别的相关性之和,相应数学公式可写为:

$$c. cs = \sum_{c \in w, c' \in w, c \neq c'} correlation(c, c'), \quad (4)$$

其中, w 表示类别 c 所处的匹配节点。

进一步得到, $c. cd$ 表示类别 c 和其不同匹配节点中的类别的相关性之和,相应数学公式可写为:

$$c. cd = \sum_{c \in w, c' \notin w} correlation(c, c'), \quad (5)$$

在此基础上,本文采用 $c. MergeJudgePara$ 作为类别 c 的判断标准,选择该值最大的类别 c 进行同一匹配节点之内的相关性值最大的类别进行融合操作。相应数学公式可写为:

$$c. MergeJudgePara = \frac{c. cs}{c. cd}. \quad (6)$$

如果一个类别和其同一匹配节点中的类别相关性很差,而和不同匹配节点间的相关性很好,即 $MergeJudgePara$ 很小,在这种情况下,本文提出一个匹配节点间的类别融合策略。首先新增一个阈值 $merge$,该阈值用于判断仅仅使用节点内融合是否合理,如果计算得到的最大的 $MergeJudgePara < merge$,则认为需要被融合的类别和其匹配节点内其他类别的相关性太差,只能进行节点间类别的融合。当使用节点间的类别融合时,本文选择 $MergeJudgePara$ 最小的类别进行节点间的融合,按照待融合类别内的订阅和其他类别的相关性进行融合,和原本节点相关性较大的订阅融合进原本的节点内的类别,和其他节点相关性大的订阅融合进其他的类别。

3.2 负载均衡

负载均衡的主要目的是为了均衡各个节点的负载,避免出现某一节点负载过重,成为性能瓶颈。若经过一定数量的事件和订阅(本文使用经过 λ 数量的事件)后,就要判断节点的负载是否失衡,一旦发生失衡,将尽快修复。

为了判断是否存在负载不均衡的情况,先要对工作负载进行量化。经过分析可知,类别 c 需要处理的数据包括3种,分别是:事件 t 、新添加的订阅 q 、需要删除的订阅 q' ,故其工作负载也包含3部分,具体如下:

$$WorkLoad(c) = \int_1^{t.num} c.size + \sum insert(q) + \sum delete(q'), \quad (7)$$

其中, $t.num$ 表示接收到的事件数量; $c.size$ 表示类别 c 的大小,使用类别中订阅的数量表示。

在公式(7)中,第一部分表示处理事件 t 的匹配操作的工作负载,其中 $c.size$ 会随着订阅的增删而变化;第二部分表示在这一段时间内增加订阅所需

的资源消耗;第三部分表示这一段时间内删除订阅所需的资源消耗。工作节点的总工作负载就是一段时间内节点内所有类别的工作负载之和。本文使用匹配操作的工作负载表示类别的工作负载。

本文采取的负载均衡策略可描述为:当某一个节点负载过大,直接将其中的一部分迁移到另一个节点中。研究中,将所有节点的工作负载之和设为 $TotalLoad$,假设共有 m 个工作节点,如果一个节点 w 的工作负载满足公式(8):

$$WorkLoad(w) \geq \frac{2}{m} * TotalLoad, \quad (8)$$

则认为该节点的工作负载过大,需要进行均衡。在负载均衡过程中,就要从负载最大的节点中选择一部分订阅转移到其他的节点,在选择这一部分订阅时,本文判断转移操作是否应该停止的标准参见公式(8)。

文中采用贪心策略来计算需要迁移的子集,使用 $DiviPara$ 作为选择标准,对其进行运算时需用到公式为:

$$c. DiviPara = (c. cd - c. cs) / c. size. \quad (9)$$

选择 $DiviPara$ 值最大的类别作为需要迁移的子集,直至满足负载的均衡的要求。使用公式(9),对节点内类别进行排序,转移类别至待接收的节点,直至达到负载均衡的要求。对于待接收节点,可以选择和待迁移类别相关性最大的节点。

4 实验

本节将给出基础的发布订阅算法的实验验证结果。实验时使用3台计算机来搭建分布式环境,3台计算机的参数配置为:1台内存为8GB,处理器为Intel(R) Core(TM) i5-8400 2.8 GHz,2台 Intel(R) Celeron(R) CPU 1007U 1.5 GHz,4GB内存。每个处理器为单核CPU。该分布式环境由研究者自行设计,计算机系统使用Ubuntu18.04,分布式系统使用zookeeper和storm组件进行搭建,涉及到的聚类节点仅选择2个,匹配节点为4个。

实验数据采取的是网站 <http://www.pocketgpsworld.com/> 上的数据,选择了网站中大约10万条数据。这些数据仅仅可用作发布的事件信息,本文使用事件信息生成相应的订阅信息,方法如下:首先规定订阅的数量,本文选择为事件数量的0.01倍;紧接着随机生成该数量的订阅,方式为每一个订阅随机选择一个事件,订阅的属性集合取该事件的随机属性子集,订阅的空间位置选择以该事

件为中心,经纬度随机扩大的一定数量为订阅的空间位置。这样做就是希望订阅和事件的匹配结果数尽量多。方便更新相关性。

本文对实验过程中的参数定义如下:公式(3)中,平衡属性相似性和空间相似性的参数 $\alpha = 0.5$ 。节点内类别融合和节点间的类别融合的判断阈值 $merge = 1$,聚类节点中网格的间隔设置为5,匹配节点中网格的间隔设置为2,新类别生成的阈值 $NewTh = 0.5$,类别负载均衡的次数为2,由事件平均分配,即如果总共10万条事件,每进行5万条则转入负载均衡判断。

首先,验证相似性和相关性是否能够优化聚类操作,从而提升事件和订阅的匹配速度。提供3种算法来进行对比,分别为:

- (1) 仅仅使用不包含频数的相似性来进行聚类操作,使用 Sim-NonFre 来表示。
- (2) 使用包含频数的相似性来进行聚类操作,使用 Sim-Fre 表示。
- (3) 同时使用包含频数的相似性和相关性来进行聚类操作,使用 Sim-Fre-Co 来表示。

针对这三种算法,测量的数据为事件经过空间筛选和属性筛选后,被分配的类别平均数量。测量的时机分别为:

- (1) 事件流和订阅流同时存在的过程,该过程从0开始,即从接收到的第一个事件和订阅开始,同时进行数据的训练和匹配操作,结果如图4所示。
- (2) 仅仅存在事件流的过程,此时订阅聚类过程已经结束,数据训练完成,仅仅存在匹配过程,结果如图4所示。

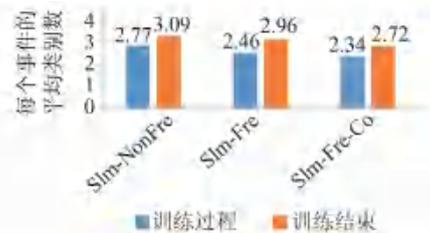


图4 事件被分配的平均类别数

Fig. 4 The average number of categories to which events are assigned

在图4中,训练过程表示事件流和订阅流同时存在,训练结束表示训练完成,仅仅进行事件流的研究。事件需要进行匹配的平均类别数量使用 Sim-NonFre 结果最多,Sim-Fre 结果居中,Sim-Fre-Co 结果最少,即使用带频率的相似性和相关性相比较于普通的相似性有效地减少了事件需要进行匹配操

作的类别数量。对比训练过程和训练结束,训练完成后单纯进行匹配操作时,事件被分配的类别增加,因为同时进行训练和匹配操作,有一些事件发布后,尚未对其给予订阅关注,随后才有订阅关注该事件,使得事件需要进行更多的匹配操作。

随后,验证类别融合策略的数据传输量。本文将提供训练过程中 *DataTra* 的结果,分别给出仅仅使用节点内的类别融合(使用 *InNode* 表示)和同时使用节点内类别融合和节点间的类别融合的数据传输量(*InAndBetNode* 表示)。实验结果如图5所示。同时使用节点内融合和节点间融合的数据传输量 *InAndBetNode* 从训练初期就处于上升阶段,但是此后类别将逐渐下降。仅仅使用节点内类别融合的数据量 *InNode*,由于仅仅使用节点内数据融合,类别的分布不合理,使得类别融合操作次数一直居高不下。

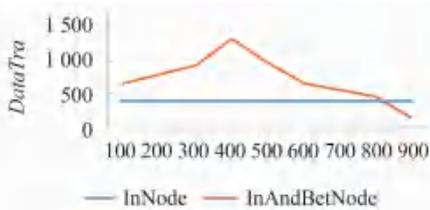


图5 类别融合数据传输量

Fig. 5 Category fusion data transmission volume

接下来,将验证类别融合策略的效果。本节拟给出2种算法进行对比,分别是:本文的 DRt-Cluster 算法,Chen 等人^[13]的发布订阅算法,Chen 等人^[13]的研究使用 kd-tree 作为聚类算法,所以本文使用 kd-tree 表示该算法。本文将提供事件的吞吐量进行对比,对比结果如图6所示。经过事件和订阅的训练之后,单单进行事件的匹配过程和订阅的增删过程来验证类别的聚类结果和在匹配节点上的分布结果,使用 *tran/sec* 表示。由图6可知,本文的 DRt-Cluster 算法的吞吐量略高于 kd-tree 算法。

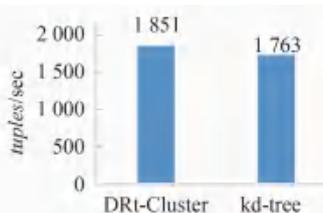


图6 吞吐量对比

Fig. 6 Throughput comparison

最后,本文研究了 DRt-Cluster 算法负载均衡操作前后,数据的吞吐量变化,实验结果如图7所示。经过负载均衡操作后,类别的分布更加均匀,数据的

吞吐量获得一定的提升。

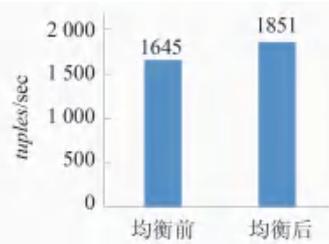


图7 均衡前后吞吐量

Fig. 7 Throughput before and after equalization

5 结束语

本文首先详细介绍了发布订阅系统,采取混合属性的应用方法,提出了相似性和相关性这2种聚类算法中的判断参数,设计并实现了一个单处理器上的即时发布订阅算法,随后设计一个分布式的系统,并且将即时的发布订阅算法部署其上提出了与其相适应的负载均衡策略,最后通过实验验证其效果。

参考文献

- [1] LI G, WANG Y, WANG T, et al. Location-aware publish/subscribe [C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Chicago: ACM SIGMOD Record, 2013:802-810.
- [2] HU Huiqi, LIU Yiqun, LI Guoliang, et al. A location-aware publish/subscribe framework for parameterized spatio-textual subscriptions [C]// IEEE 31st International Conference on Data Engineering. Seoul, South Korea: IEEE, 2015:711-722.
- [3] WANG Xiang, ZHANG Ying, ZHANG Wenjie, et al. AP-Tree: Efficiently support continuous spatial-keyword queries over stream [C]// ICDE Workshops 2015. Seoul, South Korea: IEEE, 2015, 6(1):1107-1118.
- [4] CHEN L, CONG G, CAO X. An efficient query indexing mechanism for filtering geo-textual data [C]// ACM SIGMOD International Conference on Management of Data. New York: ACM, 2013:749-760.
- [5] WANG X, ZHANG Y, ZHANG W, et al. Skype: Top-k spatial-keyword publish/subscribe over sliding window [J]. Vldb Journal, 2017, 26(3):301-326.
- [6] YU Minghe, LI Guoliang, FENG Jianhua. A cost-based method for location-aware publish/subscribe services [C]// Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. New York: ACM, 2015:693-702.
- [7] CHEN Lisi, CONG Gao, CAO Xin, et al. Temporal Spatial-KeyWord Top-k publish/subscribe [C]// IEEE 31st International Conference on Data Engineering. Seoul, South Korea: IEEE, 2015:255-266.
- [8] AJI A, WANG Fusheng, VO H, et al. Hadoop-GIS: A high performance spatial data warehousing system over mapreduce [J]. Proceedings of the VLDB Endowment, 2013, 6(11):1009-1020.

(下转第54页)