

A Publishing/Subscription System Based on the Hybrid Two-Layer Topology

Liming Zheng¹, Weidong Sun²

¹Department of Electronics Technology, Armed Police Officer Academy, Chengdu Sichuan

²National Key Laboratory for Parallel and Distributed Processing, School of Computer, National University of Defense Technology, Changsha Hunan

Email: ygrsun@163.com

Received: Feb. 7th, 2018; accepted: Feb. 19th, 2018; published: Feb. 26th, 2018

Abstract

Data distribution is a key technology for resources convergence and sharing in distributed environment. To better meet the requirement for data distribution real-time dynamic network environment, we put forward a publishing/subscription system based on the hybrid two-layer topology. It contains a two-layer topology CBDLO based on content; the lower is an unstructured topology structure; the upper is multiple corresponding to different attributes of distributed balanced binary tree; CRAWL forwards the data on the lower unstructured topology in the form of random walk, so that the data can be sent to the corresponding upper topology entry; it also includes a matching algorithm named CDM for the parallel matching data properties on the upper distributed and balanced binary tree and transmitting the matched data to the nodes that are interested in the data. The experimental results show that the system can effectively support large-scale data distribution in a dynamic network, reducing distribution costs and matching delays.

Keywords

Data Distribution, Publishing/Subscribing Message, Routing, Data Matching

一种基于混合式双层拓扑的发布/订阅系统

郑力明¹, 孙伟东²

¹武警警官学院电子技术系, 四川 成都

²国防科技大学计算机学院并行与分布处理国防科技重点实验室, 湖南 长沙

Email: ygrsun@163.com

收稿日期: 2018年2月7日; 录用日期: 2018年2月19日; 发布日期: 2018年2月26日

摘要

基于内容的快速数据分发是分布式环境中资源汇聚和共享的一项关键技术。为更好的满足动态网络环境中对数据分发实时性的要求,提出一种基于混合式双层拓扑的发布/订阅系统,它包含一种基于内容的双层拓扑结构CBDLO,下层是一个非结构化的拓扑结构,上层是多个对应不同属性的分布式平衡二叉树;一种基于内容的带踪迹路由算法CRAWL,以随机行走的方式在下层非结构化拓扑上转发数据,同时为对应数据寻找相应的上层属性拓扑的入口;一种基于属性计数的分布式匹配算法CDM,在上层的分布式平衡二叉树上并行的匹配数据的各个属性,并把匹配的数据发送到对数据感兴趣的节点上。实验结果表明,该系统能够有效支持大规模动态网络中的数据分发,降低分发开销和匹配延迟。

关键词

数据分发, 发布/订阅消息, 路由, 数据匹配

Copyright © 2018 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

数据分发是指将一组数据从一个或者多个数据源节点发送到多个目标节点的过程,其关键在于数据需要由源节点出发经由动态异构的网络环境传送到地理上分散的目标节点集合。数据分发已成为众多分布式应用的基本组成部分,应急响应系统的不断涌现对数据分发的实时性提出较高的要求。例如网络中心战[1]中,军事情报必须在尽可能短的时间内从源节点准确的传送到感兴趣的目标节点上。再如地震发生后,美国地质勘探局的 Shake Cast [2]系统将震后公共设施损毁图发送给感兴趣的用户,以帮助受灾人群快速选择逃生路线。

发布/订阅模型是用于紧急应用的异构数据分发的一种关键技术。在发布/订阅的模式中,数据的生产者(数据源,又称为发布者)与消费者(目标节点,又称为订阅者)之间所传输的数据称为事件,订阅者通过发出订阅事件来声明自己对数据的兴趣,通过取消订阅事件来声明不再感兴趣的数据;发布者根据之前的订阅条件将其产生的事件(数据)经由一些中间节点分发给相应的订阅者。通过将发送者和接收者在空间,时间和同步上解耦[3],发布/订阅系统能够很容易的扩展至较大规模。

发布/订阅系统根据订阅模型的不同可分为基于通道、基于主题、基于类型和基于内容四类[3]。基于通道的发布/订阅系统的表达能力很弱,最初的发布/订阅系统一般是基于主题[4] [5] [6]的。基于类型的发布/订阅系统将面向对象的类型模型引入到系统中,根据事件类型来过滤事件,但系统效率较低,现有应用很少。基于内容的发布/订阅系统[7] [8] [9] [10]提供丰富的表达能力,其订阅是多个维度上的布尔表达式。为了满足日益增加的复杂订阅的需求,基于内容的发布/订阅是本文的研究对象。

已有的基于内容的发布/订阅系统难以满足应急响应系统对数据分发实时性的要求。在基于非结构化拓扑的发布/订阅系统中,节点之间采用随机拓扑以保证链路的可靠性,然而非结构化拓扑中节点之间不具有稳定的连接关系,消息的转发往往是随机的,这导致较高的通信冗余[11] [12]。在基于结构化拓扑的发布/订阅系统中,节点间的邻居关系通常由确定性的算法(如 DHT)严格控制,资源(或资源的元信息)的放置也是由确定性的算法精确发布到特定的节点上,结构化拓扑通常具有较低的节点定位延迟,然而拓

扑的维护开销较高[8] [13] [14]。

本文提出一种基于混合式双层拓扑的发布/订阅系统以支持快速数据分发。首先, 提出一种混合式双层拓扑 CBDLO。CBDLO 融合了非结构化拓扑的灵活性和可扩展性优点, 以及结构化拓扑的准确性和低消息开销的特点, 为动态环境下的快速数据分发提供鲁棒的、低开销、高效率的路由和匹配支持; 其次, 为降低 CBDLO 中下层拓扑的数据路由的消息开销, 提出一种基于内容的带踪迹路由算法 CRAWL; 再次, 为降低 CBDLO 中上层拓扑的数据匹配的延迟, 提出一个基于计数的分布式匹配算法 CDM。基于 Peersim 模拟器的实验结果表明, CBDLO 能够有效支持大规模动态网络中的数据分发, CRAWL 算法能够有效降低分发开销, CDM 算法在数据匹配时能够有效降低匹配延迟。

2. 系统结构

为了更好的发挥结构化拓扑和非结构化拓扑的优点, 摒弃它们的缺点, 针对动态网络环境中基于内容的快速数据分发的拓扑构建问题, 提出了一种基于内容的双层拓扑结构 CBDLO (Content-based Double-Layered Overlay)。CBDLO 融合了非结构化拓扑的灵活性和可扩展性优点, 以及结构化拓扑的准确性和低消息开销的特点。

CBDLO 的下层是一个非结构化的网络拓扑以维护网络的连通性, 其逻辑结构类似于随机图(Random Graph)。每个节点保存一定数量的其它节点作为其邻居—称为视图(View), 并定期的更新其视图, 以便拓扑结构能够及时的反映网络的连通性和动态性。在 CBDLO 中采用随机抽样的方式定期的选择节点并更新节点的视图, 这种方式使得拓扑基本上维持了随机图的结构, 也基本保持了网络的“小世界模型”特性和幂律特性。每个节点同时还拥有一张属性拓扑入口点查询表 APT (Access Point Lookup Table), 缓存有一定数量的上层属性拓扑的信息, 以支持其它节点的加入和数据转发时的路由。

上层是构建在这个非结构化网络拓扑之上的属性拓扑, 为高效的分发路由和匹配提供支持。CBDLO 依据每个节点的订阅属性把节点分簇, 让拥有相同订阅属性的节点构成一个簇, 每个簇即是一个属性拓扑。CBDLO 依据订阅中属性值的分布, 把整个属性值空间分割成若干个相连却互不相交的子空间, 每个子空间对应一个虚拟节点, 所有的虚拟节点又被映射到实际节点上, 并组织成一棵分布式的平衡二叉树结构。

在 CBDLO 中, 当有数据产生以后, 首先通过随机行走的方式在下层的拓扑结构上转发, 匹配到相应的属性拓扑以后, 进入到上层属性拓扑中, 开始以确定的方式在属性拓扑中快速的分发数据。图 1 表示了 CBDLO 的双层网络拓扑结构。

3. CRAWL 算法描述

在 CBDLO 下层的拓扑中, 数据以随机行走的方式转发, 并在转发的每个节点上通过属性拓扑入口点查询表 APT 查找与数据匹配的属性拓扑。为了避免在随机行走的过程中形成回路, 提出了一种基于内容的带踪迹路由算法 CRAWL (Content-based Routing Algorithm with Labeled Trace), 在转发的数据上携带了一个路径踪迹(Trace)缓冲区, 用来记录最近经过的路径。为了防止踪迹带来的“路径阻塞”, 引入了路径的恢复机制。

3.1. 算法过程

源节点产生数据以后, 分解数据的各个属性, 然后为每个属性启动一个随机行走的过程向外散布数据。因此, 属于一个数据的多个属性之间以并行的方式向外散布数据, 加快了数据的分发过程。每个节点在收到数据以后在它的属性拓扑入口点查询表中查找对应的属性拓扑, 如果找到则把数据直接转发到

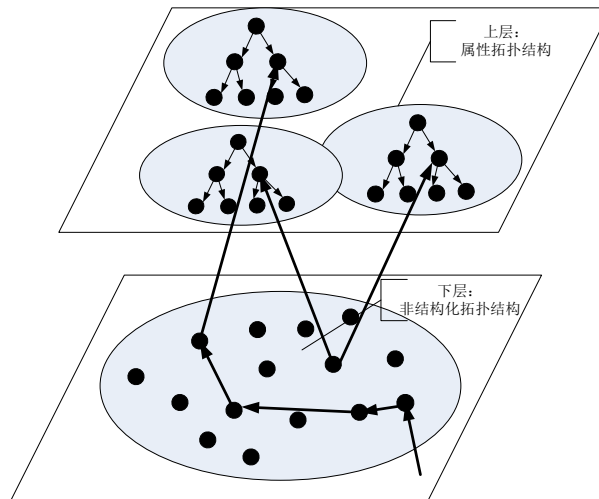


Figure 1. Two-layered net topology structure
图 1. CBDLO 的双层网络拓扑结构

下层的随机转发路由算法

```
//假定节点 A 发布一个新数据 D<MetadataList,Trace,Content>;
//则从节点 A 开始有如下的执行过程
step1. for int i = 0 to k do
begin
    data = <MetadataList[i],Trace,Content>
    data.trace.add(A);
    N' = select(A.view)
    send(data,N');
end;
step2. //当节点 N 收到数据 data<metadata,trace,content>之后
step2.1 for int j = 0 to N.APT.count do
begin
    if N.APT[j].attribute = data.metadata.attribute then
begin
    发送数据 data 到节点 N.APT[j].subscriber;
    break;
end;
end;
step2.2 if data.tti > k then
return;
step2.3 if j = N.APT.count then
begin
    data.trace.add(N);
    N' = select(view,N), 且 N'不属于 N.trace
    N = N';
    send(data,N');
    goto step3;
end;
end;
step3. end.
```

Figure 2. The random transmission routing of the lower topology
图 2. 属性拓扑间的路由算法

相应的属性拓扑上, 否则就从它的邻居节点中随机的选择一个节点转发数据。为了防止短期内数据回到刚刚已经收到过的该数据的节点上形成回路, 造成冗余消息, 在分发的数据上附加了一个长度为 k 的踪迹列表(Trace), 用以记录该数据最近刚刚走过的节点。

下层拓扑的随机转发路由过程如图 2 所示。step 1 把数据分解为单一属性-值对的形式, 并把源节点

加入到 Trace 表中, 然后从邻居节点中随机选择一个节点把数据发送出去, 从 step 1 可以看出, 多个属性的发送过程是并行的; step 2 给出的是当某个节点收到一个属性数据以后的处理过程, step 2.1 表示它先在自己的入口点查询表中查找, 找到就直接发送到该节点, 并终止路由过程, 否则在 step 2.2 中判断是否达到路由的最大跳数, 没有达到就进入 step 2.2, 首先把当前节点加入到 Trace 表中, 然后从邻居节点中随机的选择一个不在 Trace 表中的节点并且把数据发送到该节点。

3.2. 路径的阻塞与恢复

Trace 表虽然能够防止数据形成回路, 减小冗余的消息, 但是它可能会引起“路径阻塞”。路径阻塞是指在随机行走的过程中到达的某个节点的邻居全部位于 Trace 表中时, 将无路可走。这种情况可以用图 3 来解释。在图 3 中节点 A 启动一个随机行走过, 沿着箭头所指的路径, 经过 B → C → D → E → F, 但是节点 F 位于网络的边缘, 只有一个邻居节点, 因为 Trace 表的原因, 虽然此时没有找到匹配的属性拓扑, 也没有达到最大跳步数, 但是已经“无路可走”。路径恢复是指放生路径阻塞以后, 从 Trace 表中随机的选择一个节点, 重新启动一个随机行走的过程, 但是为了避免陷入不断的路径阻塞、路径恢复的循环中, 每次重新启动的随机行走的最大跳步数应减为原来最大跳步数的一半。

4. CDM 算法描述

数据经过 CBDLO 下层拓扑的随机转发找到上层的属性拓扑以后, 开始在树上快速的匹配分发。针对上层的平衡二叉树拓扑上的数据匹配分发过程, 提出了一个基于计数的分布式匹配算法 CDM (Counter-based Distributed Matching Algorithm)。CDM 算法在上层的分布式平衡二叉树上并行的匹配数据的各个属性, 并把匹配的数据发送到对数据感兴趣的节点上, 订阅节点对匹配的属性个数进行计数, 当匹配的属性个数超过订阅的属性个数时接受数据。

数据到达属性拓扑以后, 只需要找到匹配属性值的节点, 并把数据转发出去。属性拓扑是一个平衡二叉树结构, 因此匹配的过程是一个在二叉树上传递的过程, 但是在树上传递之前和结束以后需要执行两次转换, 因为属性拓扑的入口点是实际的订阅节点, 所以在树上传递以前需要把实际的订阅者转化为相应属性拓扑上的虚拟节点; 当找到匹配的虚拟节点以后, 需要把数据发送到相应的实际节点。

当属性拓扑中的入口节点收到一个数据以后会从其虚拟节点列表中查找分配给它的所有虚拟节点, 并查找和数据属性相匹配的虚拟节点, 完成第一次转换, 然后开始匹配过程。当找到匹配数据值的虚拟节点以后, 会把数据发送到虚拟节点上存储的实际节点列表(Node List)中的所有实际节点, 完成第二次转换。

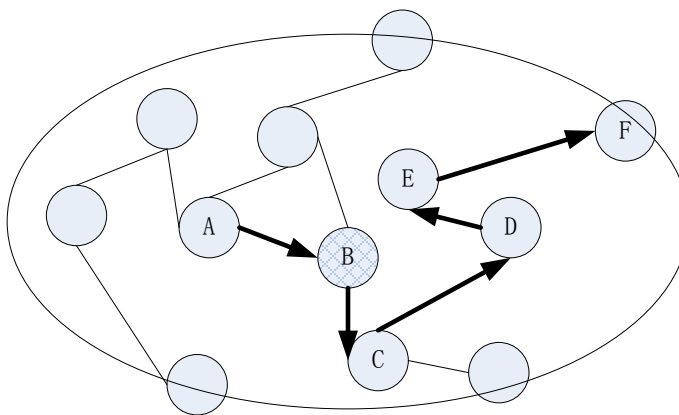


Figure 3. Schematic diagram of path blocking
图 3. 路径阻塞情况示意图

具体的匹配过程如图 4 所示, 其中 step 1 收到数据的节点首先在它的虚拟节点列表中查找是否有与当前收到的数据相匹配的属性, 如果找到, 则从找到的虚拟节点查找包含此属性值的虚拟节点; 如果没有找到, 则可能是该订阅者已经离开, 所以需要重新在下层拓扑中重新启动一个属性拓扑查找的随机行走过程。为了防止每次都找不到而陷入死循环, 每次启动的随机行走过程会把最大的跳步数减半。

5. 测试结果与分析

5.1. 方法实现

Peersim [15]是一款 P2P 系统的模拟器, 它以一种组件开发的方式来支持 P2P 系统模拟, 因此使得协议的开发变得容易。Peersim 是用 Java 语言开发的模拟器, 它支持两种开发模式, 周期驱动(Cycle Driven)的模式和事件驱动(Event Driven)的模式。Peersim 支持节点规模巨大的模拟, 适应并支持大规模网络的可扩展性和动态性。因此在 CBDLO 的模拟测试中, 选用 Peersim 模拟器。为了更好的测试 CBDLO 的性能, 以事件驱动的方式在 Peersim 上实现了拓扑结构, 并对测试结果进行了分析。

5.2. 参数设置

假定属性在订阅中的出现服从 Zip-f 分布, 即对任何一个订阅, 属性 i 在其中出现的概率为 $\mu \times i^{-\alpha}$, α, μ 是 Zip-f 分布的参数, 且 $0 < \alpha, \mu < 1$ 。也就是说常见的属性大多数订阅都拥有, 一些特殊的属性只被一些特殊的订阅拥有。对于 CBDLO 来说, 虚拟节点的数目最多等于该属性的值空间被订阅所分割产生

```

属性拓扑匹配算法
//节点 A 收到被标识为匹配的数据 data
//data=<subject,metadata,trace,content>
step1. bool Found = false;
      for i = 0 to A.VirtualNodeList.count do
      begin
        if (metadata.attribute = A.VirtualNodeList[i].Attribute) then
        begin
          //找到匹配的虚拟节点
          step1.1 Found = true;
          step1.2 VirtualNode VN =
                    FindTargetNode(A.VirtualNodeList[i],data);
          step1.3 for j = 0 to VN.NodeList.count do
          begin
            Send(data,VN.NodeList[j]);
          end
          break;
        end;
      end;
step2. if (not Found) then
      being
        //没有找到匹配的虚拟节点
        data.MaxHop = data.MaxHop/2;
        从 A 开始启动一轮新的 Gossip 发送过程,
        最大跳步数是 MaxHop;
      end;
step3. end.

```

Figure 4. Property topology routing algorithm

图 4. 属性拓扑内的路由算法

的最大分割数, 不超过 $\max(VN) = \text{subscriptions} + 2$ 。因为属性拓扑是一颗平衡二叉树, 所以树的高度等于 $h = \log_2 N$, N 是属性拓扑中虚拟节点的总数。

5.3. 路由的消息开销

为了说明 CBDLO 性能, 在 Peersim 上实现了另外一种简单的协议 Simple 以及 TERA 来和 CBDLO 做比较。Simple 是简单的基于内容的洪泛式分发协议。在这个简单的协议中, 拓扑结构是一个类似 CBDLO 下层拓扑的非结构化随机图拓扑, 而数据转发过程则首先在本地进行属性的匹配, 如果匹配则表明当前节点订阅了该数据, 所以接收数据, 然后再把数据转发出去, 直到到达消息的最大时限(跳步数)。这个协议无疑是很简单的, 但是消息的开销也很大, 而且也不能保证能够覆盖所有对数据感兴趣的节点。

首先假定系统的规模为 3000 个节点, 视图的大小是 10, 本地的最大订阅缓存数是 100, 转发的最大跳步数是 10, 则有如下的测试结果。

一个有效的系统不仅要求消息冗余的开销低, 而且要求能够把数据准确的发送到相应的需求节点上, 为了刻画系统的有效性, 我们首先给出有效分发的消息开销的定义:

$$\text{有效分发的消息开销} = \frac{\text{系统的所有消息数目}}{\frac{\text{系统实际成功匹配的消息}}{\text{理论上所有可能的匹配}}}$$

在图 5 中, Simple 协议的第一个点表示系统的所有消息总数是 68,605 个, 理论上应该匹配的发布消息数应该是 408, 但是实际匹配的数目只有 68 个, 所以有效分发的消息开销为 $68,605 / (68 / 408) = 411,630$ 。

从图 5 可以看出, CRAWL 算法极大的降低了系统的有效分发开销。因为随着订阅数目的增加, 在随机转发的过程中有匹配的概率增加, 所以随着订阅的增加 Simple 协议的有效分发开销也逐渐降低。但

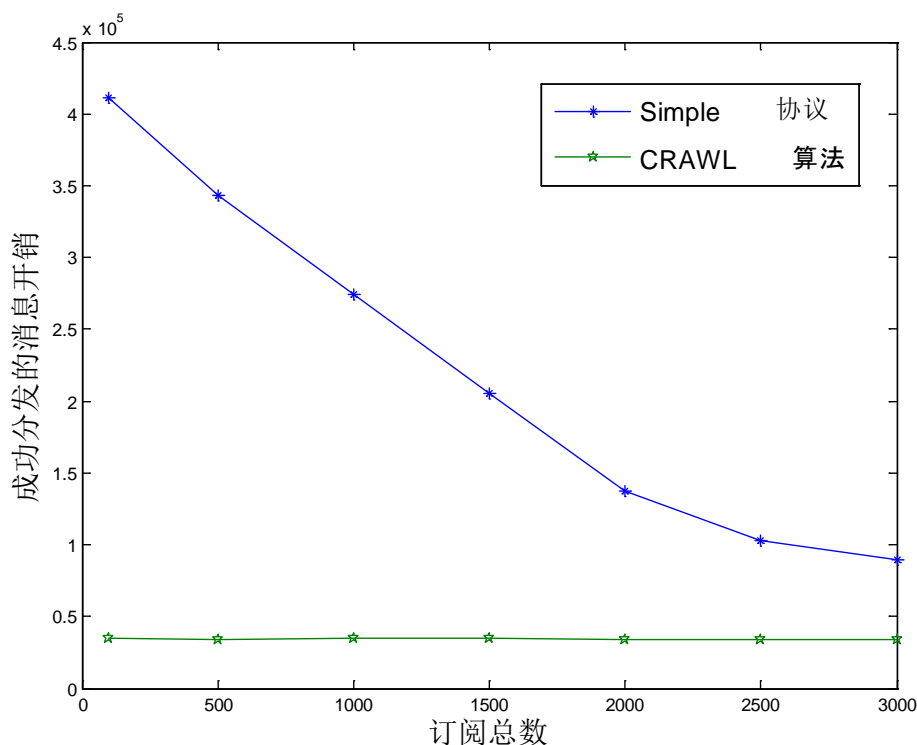


Figure 5. Effective distribution cost changing with the total number of subscribing

图 5. 有效分发的开销随订阅总数的变化

是订阅的增加对 CBDLO 的影响极小, 因为模拟的过程中订阅数目一定之后, 一定时间内产生的消息总数是固定的, 而 CBDLO 的 CRAWL 路由算法只有第一阶段受其影响, 第二阶段的开销是固定的, 等于属性拓扑树的高度, 但是对第一阶段产生影响的实际上是属性的数目, 而且 CBDLO 能够把数据准确的发送到所有对数据的需求节点上, 因此, CRAWL 算法受订阅数目的影响很小。

图 6 展示了有效分发的消息开销与最大跳步数的关系。与图 5 比较可以发现, 与订阅数相比, 当跳步数增加的时候, CRAWL 算法的有效分发的开销降低要明显一点。但是和简单的随机算法相比较, 仍然是很稳定的。Simple 算法开始的时候受跳步数影响而有效分发的消息开销迅速降低, 但是当达到一定程度之后, 却不再降低, 甚至有所回升。原因是 Simple 算法随着跳步数的增加消息会不断的增加, 但是跳步数越大, 对它分发准确性的影响就越小, 反而是消息本身成了主要开销。

图 7 比较了 TERA 和 CBDLO 在数据分发时的消息开销随节点变化的情况。在实验中节点的数目从

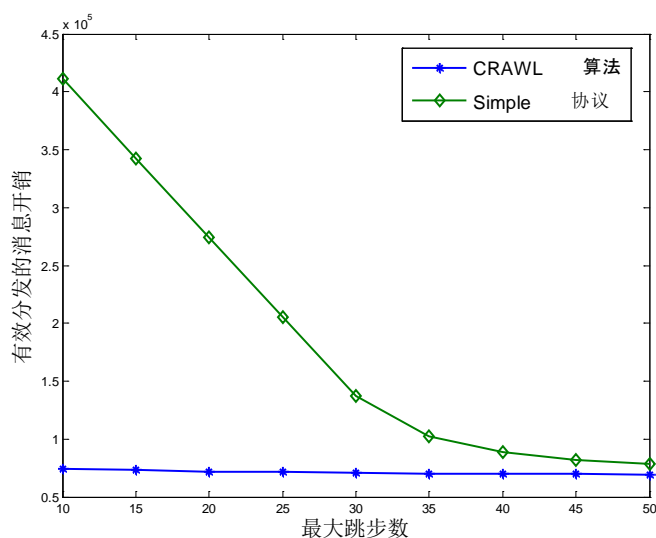


Figure 6. The relationship between overhead of effective distributed messages and the max hops

图 6. 有效分发的消息开销随最大跳步数的变化

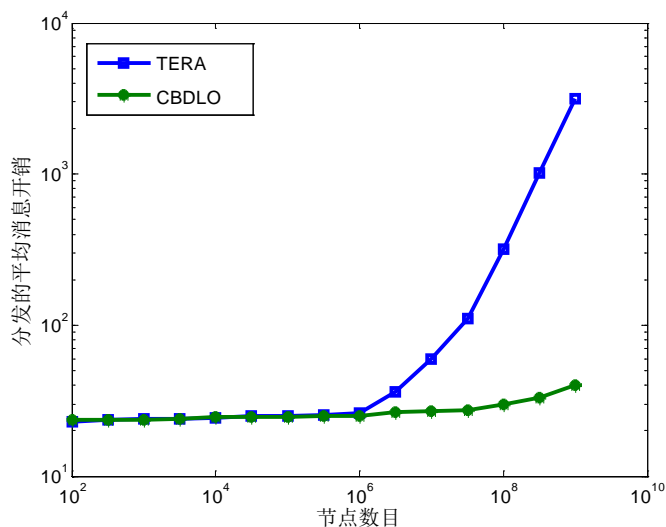


Figure 7. Overhead of distributed messages comparison between CBDLO and TERA

图 7. CBDLO 与 TERA 分发的消息开销比较

100 到 1,000,000,000 变化。TERA 中有 100 个主题, 文献[16]的实验表明在 TERA 的下层拓扑上进行随机转发的命中率随着入口点查询表大小的增加而提高, 当入口点查询表的大小为 50、随机行走的最大跳步数为 10 时, 命中率已经能够达到 90%, 在规模较大的网络中提高入口点查询表的大小时, 命中率并不会特别显著提高, 因此在综合考虑空间和效率的情况下, TERA 的入口点查询表的大小宜定为 50。CBDLO 在下层拓扑的命中率也随着入口点查询表的大小的增加而提高, 但是要在最大跳步数为 10 的情况下让命中率达到 90%, 则入口点查询表的大小不能低于 100, 因此 CBDLO 的入口点查询表大小被设定为 100。在上面的条件下, 得到如图 7 所示的结果, 实验结果表明, 随着节点数目的增加, 在网络规模较小时, TERA 和 CBDLO 的消息开销相差不大, 这是因为网络规模较小时 TERA 的广播消息开销并不大, 而 CBDLO 却较多的下层消息开销; 但是随着网络规模增大, TERA 的消息开销迅速增大, 而 CBDLO 增加的更加缓慢, 表明 CBDLO 的上层结构化拓扑有效的降低了消息的开销。因此 CBDLO 比 TERA 表现出更大的优势, 更加适应大规模网络的数据分发。

5.4. 内容匹配的开销

传统的基于内容的分发往往把属性的匹配过程集中到一个或少数几个节点上, 使得属性的匹配过程成了快速分发的一个瓶颈, 而且使得少数节点的负载过重, 严重影响了系统的效率, 而且损害了系统的可扩展性。CBDLO 把匹配的过程分布到网络上不同的节点上, 使得匹配的过程得以并行分布的进行, 不但提高了匹配的执行速度, 而且降低了单个节点的开销。CRAWL 通过两个机制实现了数据匹配过滤的分布化, 第一个机制是分解一个数据的多个属性, 让不同的属性在不同的拓扑上进行, 即使是在下层的拓扑上, 多个属性之间也是并行的; 第二个机制是通过分割属性的值空间为相连但互不相交的子空间, 把属于不同范围的属性值分布到不同的子空间去执行。

假设属性的值空间范围 $[a, b]$, 被 m 个节点产生的 n 个订阅分割, 则产生的子空间的个数, 也就是虚拟节点的个数满足: $|Virtual Node| \leq n + 2$, 内容匹配的开销取决于每个属性数据分发需要经过的总跳数 Hop 以及每一跳的匹配开销。在 Simple 算法中, 一个节点对一个订阅需要执行的匹配操作数等于属性的个数, 因此如果用 m 表示订阅中属性的平均数目, s 表示本地节点发布的订阅数目, 则每个节点需要 $m \times s$ 个属性的匹配开销。

在 CRAWL 算法中, 开销取决于两个阶段, 第一个阶段是下层拓扑上的随机转发, 第二个阶段是上层拓扑上的确定转发。在第一个阶段, 每一跳的开销等于执行 $|APT|$ 个属性的匹配开销, 因为在 CRAWL 算法中每一步只对一个属性进行匹配; 在第二个阶段执行每一步的开销是 1。CBDLO 中数据转发经过的跳步数 H 由两个部分组成, 第一部分是随机转发的跳步数, 第二部分是树的高度。因为多个属性之间是完全并行的, 所以数据的分发经过的跳步数取决于最慢的属性的跳步数。

$$H = H_{stage1} + H_{stage2}$$

因此 CBDLO 的匹配开销为:

$$Cost(CBDLO) = |APT| \times H_{stage1} + 1 \times H_{stage2}$$

实验中, 假定订阅服从 Zip-f 分布, 邻居节点的视图为 20, APT 的大小为 20 建立模拟实验, 最大跳步数是 30。

图 8 显示了匹配的开销随着订阅数目的变化情况, 因为 Simple 协议是泛洪的协议, 所以在它的纵坐标值上乘了一个系数 0.1, 而在 CBDLO 的纵坐标轴上乘了一个系数 1/8, 因为属性的平均数目是 8, 这样我们看到的实际就是源节点启动的单个转发中产生的匹配开销, 虽然 Simple 所乘的系数比 CBDLO 的小, 但是 CBDLO 依然展示了巨大的优势。从图中还可以发现, 当订阅数目较小(<3000)时, CBDLO 随着订

阅数目的增大反而减小, 这是因为订阅数目比较小时, 随着订阅的增大 APT 的命中率会不断提高, 因而第一阶段的匹配开销减小, 但第二阶段的开销等于树的高度, 当数目较小时, 取对数得到的结果变化很小。

图 9 显示了匹配开销随属性总数的变化情况。图中纵坐标的处理和图 8 的处理相同。从图中可以看出 CBDLO 依然展示了巨大的优势。属性数目的变化对 CBDLO 的影响很小, 充分展示了并行匹配的效果。

6. 结语

基于内容的快速数据分发是大规模分布式系统构建的关键技术, 对于各种以数据为中心的应用意义重大。然而, 广域网络的动态性与异构性、数据集的多样性与动态型, 以及客户请求的多样性, 都给构

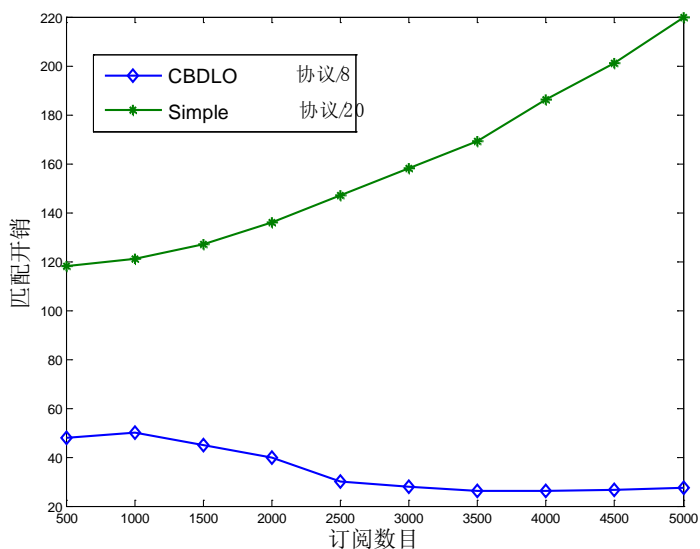


Figure 8. The relationship between overhead of matching and subscription number
图 8. 匹配开销与订阅数目之间的关系

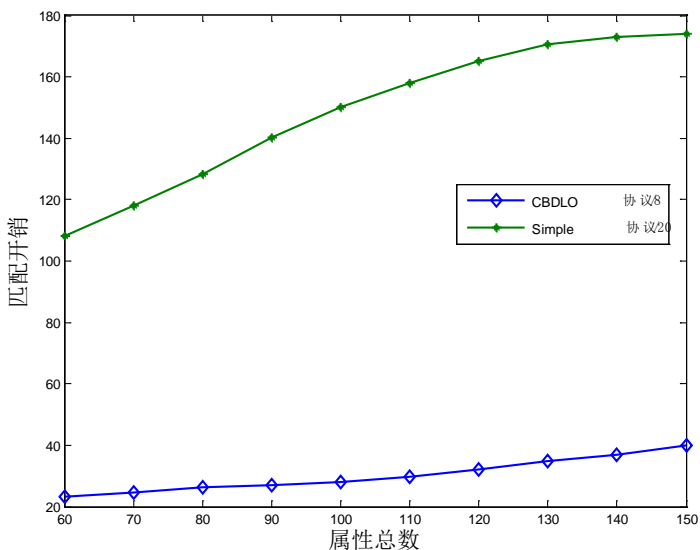


Figure 9. The relationship between matching overhead and property number
图 9. 匹配开销与属性总数之间的关系

建高效鲁棒的快速数据分发系统带来了很大的挑战。本文针对动态网络环境下的快速数据分发技术进行了深入研究, 提出一种基于混合式双层拓扑的发布/订阅系统, 它包含一种基于内容的双层拓扑结构 CBDLO, 一种基于内容的带踪迹路由算法 CRAWL 和一种基于属性计数的分布式匹配算法 CDM。实验结果表明, 该系统对动态网络有很强的自适应性, 能够有效的避免回路, 降低冗余的消息开销和匹配延迟, 提高了分发的效率。

基金项目

本文受四川省科技厅重点研发项目《基于物联网的实时数字营区安全监控系统》(2017FZ0011)资助。

参考文献 (References)

- [1] Dell-Imagine, R. (1976) JTIDS—An Overview of the System Design and Implementation (Joint Tactical Information Distribution System). *Position Location and Navigation Symposium*, San Diego, Calif, 16-21 April 1976, 212-215.
- [2] ShakeCast. (2014) <https://earthquake.usgs.gov/earthquakes/>
- [3] Eugster, P.T., Felber, P., Guerraoui, R., *et al.* (2003) The Many Faces of Publish/Subscribe. *ACM Computing Surveys (CSUR)*, **35**, 114-131. <https://doi.org/10.1145/857076.857078>
- [4] Castro, M., Druschel, P., Kermarrec, A., *et al.* (2002) SCRIBE: A Large-Scale and Decentralized Application-Level Multicast Infrastructure. *IEEE Journal on Selected Areas in Communications*, **20**, 1489-1499. <https://doi.org/10.1109/JSAC.2002.803069>
- [5] Rahimian, F., Girdzijauskas, S., Payberah, A.H., *et al.* (2011) Vitis: A Gossip-Based Hybrid Overlay for Internet-Scale Publish/Subscribe Enabling Rendezvous Routing in Unstructured Overlay Networks. *IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, Anchorage, AK, 16-20 May 2011, 746-757.
- [6] Zheng, Z., Wang, Y. and Ma, X. (2011) PeerChatter: A Peer-to-Peer Architecture for Data Distribution over Social Networks. *Information—An International Interdisciplinary Journal*, **15**, 259-266.
- [7] Pietzuch, P. and Hermes, B.J. (2002) A Distributed Event-Based Middleware Architecture. *22nd International Conference on Distributed Computing Systems Workshops*, Vienna, 2-5 July 2002, 611-618. <https://doi.org/10.1109/ICDCSW.2002.1030837>
- [8] Gupta, A., Sahin, O.D., Agrawal, D., *et al.* (2004) Meghdoot: Content-Based Publish/Subscribe over P2P Networks. *Proceedings of the 5th ACM/IFIP/USENIX International Conference on Middleware (Middleware)*, Toronto, 18-22 October 2004, 254-273.
- [9] Grummt, E. (2011) Fine-Grained Parallel XML Filtering for Content-Based Publish/Subscribe Systems. *Proceedings of the 5th ACM International Conference on Distributed Event-Based System*, New York, 11-15 July 2011, 219-228. <https://doi.org/10.1145/2002259.2002288>
- [10] Li, M., Ye, F., Kim, M., *et al.* (2011) A Scalable and Elastic Publish/Subscribe Service. *IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, Alaska, 16-20 May 2011, 1254-1265.
- [11] Cao, F. and Singh, J.P. (2005) MEDYM: Match-Early with Dynamic Multicast for Content-Based Publish-Subscribe Networks. *Proceedings of the ACM/IFIP/USENIX 2005 International Conference on Middleware (Middleware)*, New York, 1 November 2005, 292-313. https://doi.org/10.1007/11587552_15
- [12] Wong, B. and Guha, S. (2008) Quasar: A Probabilistic Publish-Subscribe System for Social Networks. *Proceedings of the 7th International Workshop on Peer-to-Peer Systems*, Tampa Bay, 25-26 February 2008, 2-7.
- [13] Aekaterinidis, I. and Triantafillou, P. (2006) Pastry Strings: A Comprehensive Content-Based Publish/Subscribe DHT Network. *IEEE 26th International Conference on Distributed Computing Systems (ICDCS)*, Lisboa, 4-7 July 2006, 23-32. <https://doi.org/10.1109/ICDCS.2006.63>
- [14] Baldoni, R., Marchetti, C., Virgillito, A., *et al.* (2005) Content-Based Publish-Subscribe over Structured Overlay Networks. *IEEE 25th International Conference on Distributed Computing Systems (ICDCS)*, Columbus, 6-10 June 2005, 437-446. <https://doi.org/10.1109/ICDCS.2005.19>
- [15] Peersim. (2014) <http://peersim.sourceforge.net/#docs>
- [16] Baldoni, R., Beraldi, R. and Quema, V. (2007) TERA: Topic-Based Event Routing for Peer-to-Peer Architectures. *Proceedings of the 2007 Inaugural International Conference on Distributed Event-Based Systems*, Toronto, 20-22 June 2007, 157-164. <https://doi.org/10.1145/1266894.1266898>

知网检索的两种方式：

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>
下拉列表框选择：[ISSN]，输入期刊 ISSN：2325-2286，即可查询
2. 打开知网首页 <http://cnki.net/>
左侧“国际文献总库”进入，输入文章标题，即可查询

投稿请点击：<http://www.hanspub.org/Submission.aspx>

期刊邮箱：sea@hanspub.org