基于结构化 P2P 的发布订阅系统①

沈燕玉，王泽洪，李国宾

(同济大学 信息与通信工程系，上海 200092)

摘　要：从发布/订阅(Pub/Sub)系统的拓扑结构入手，基于结构化 P2P，提出分层次的发布/订阅系统拓扑结构，将节点按处理能力分为超节点和普通节点，超节点组织形成超立方体结构，实现整个网络的广播遍历。层次式分布网络不仅具有一般 P2P 网络的特性，即能够支持大规模、动态的分布式应用，而且更适合于发布订阅系统中对大量事件的传播的要求。仿真结果表明，基于结构化 P2P 的发布订阅系统能显著降低系统负载，提高系统的可扩展性。

关键词：发布/订阅系统；结构化 P2P；超立方体；负载

Publish-Subscribe System Based on Structural P2P

SHEN Yan-Yu, WANG Ze-Hong, LI Guo-Bin

(Information and Communication Engineering, Tongji University, Shanghai 200092, China)

Abstract: Starting from the topology of network, based on structural P2P, this paper proposed a hierarchy of publish - subscribe system. The nodes were separated into super nodes and common nodes according to their ability, and those super nodes were organized in Hypercube structure. According to the broadcasting algorithm of the Hypercube, it realized the traverse in the whole network. The result of the simulation indicated that the system load could be well balanced and bandwidth could be saved in this system.

Key words: publish-subscribe system; structural P2P; Hypercube; load

发布/订阅[1](Publish/Subscribe，Pub/Sub)系统是一种新型的面向应用和服务的通信模式，具有松耦合、多对多、异步以及匿名等通信特征，Pub/Sub 系统直接反映了以信息为导向的应用的内在特征，近年来大规模的 Pub/Sub 系统受到国内外学术界的重视。由于 Pub/Sub 系统的匿名特性，传统的路由通过遍历网络内所有代理，通常采用 Flooding[2]、Gossip[3]等方法，遍历代理网络寻找通信对象。无结构以及节点的动态性，使得事件路由很难维护，系统的可扩展性差。本文从 Pub/Sub 系统的拓扑结构入手，引入 Chord 环[4]和 Hypercube[5-7]结构，提出分层的 Pub/Sub 系统拓扑结构，并设计相应的事件路由算法。研究结果表明：层次分布式路由模型能有效降低系统负载，大大提高系统的可扩展性。

1　相关工作

Pub/Sub 系统由发布者、订阅者和代理网络三部分组成，如图 1 所示。发布者发布(Publish)通告，消费者订阅(Subscription)通告。发布者和订阅者的通信通过与代理网络(Broker Network)的交互实现。订阅者可以有多个有效的订阅，在客户已经签发订阅后，通告服务为客户提供所有匹配的通告，直到客户取消订阅。

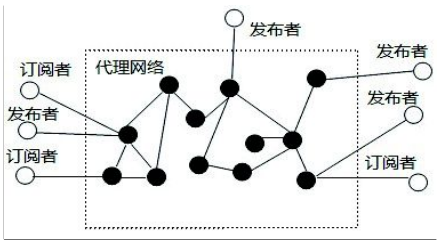

图 1　Pub/Sub 系统通信结构

① 基金项目:国家自然科学基金(60475019)
收稿时间:2011-06-21;收到修改稿时间:2011-08-10

一个典型的 Pub/Sub 系统包括拓扑结构、事件模型、订阅模型、匹配算法和路由算法。事件模型定义了事件的数据结构；订阅模型定义了系统能够支持的订阅条件，指明了订阅者如何表达对事件的兴趣；事件模型和订阅模型共同决定了系统的表达能力。匹配算法负责寻找有共同兴趣的双方。拓扑结构定义了事件代理之间的组织结构，在一定程度上决定了系统的可扩展性；路由算法一般要根据相应的事件代理网络的拓扑结构，选择适当的路径，将一个事件从发布者传送到订阅者。Pub/Sub 系统的表达能力、效率、可扩展性和服务质量是其主要目标，但这些目标往往是相互矛盾的，需要根据具体需求权衡利弊，找到合理的均衡点。

按照拓扑结构的稳定性，Pub/Sub 系统在结构上分为静态模式和 P2P 模式[8]。

P2P 结构又分为纯 P2P 网络和混合的 P2P 网络两种。纯 P2P 网络的 Pub/Sub 系统是指网络中每个节点同时作为事件代理和客户端，如 Scribe[9]系统；混合的 P2P 网络指的是 Pub/Sub 系统中的每个节点只是事件代理，每个代理连接若干个客户端，如 Hermes[10]。

目前，对 P2P 网络的研究进入第三代结构化 P2P 网络，它一般都是基于分布式散列表 (Distributed Hash Table, DHT[11])技术，运用 DHT 技术建立具有一定结构的逻辑拓扑，使节点与资源形成一定的关系，每个节点按照一定规则保存系统中部分其他节点的信息，为搜索提供一定的信息，代表结构有 Chord 环[4]、CAN[12]、Pastry[13]，Tapestry[14]等。尽管采用 DHT 技术能够实现资源的高效定位，但对于节点频繁移动的 Pub/Sub 系统来说需要的维护开销比较大。

2 基于结构化P2P的发布订阅系统拓扑

本文以结构化 P2P 为基础，提出分布式发布订阅系统双层拓扑结构，如图 2 所示。Chord 层位于模型的下层，由一系列 Chord 环组成，每个 Chord 环内节点形成一个节点簇，簇内节点物理位置相对较近，簇首节点 SN 为能力(在线时间，存储空间，带宽等综合能力)最强的节点，其他节点为普通节点 CN。SN 定期检查各 CN 的能力，并从中选出候选簇首节点。Hypercube 层位于模型上层，由底层各 Chord 环上的簇首节点 SN 按照超立方体结构组成。由于 Pub/Sub 系统在本质要求任何一个提交的订阅都要遍历到代理网络中所有已发布的事件，任何一个发布的事件都要遍历到代理网络中所有已提交的订阅。因此各簇首节点 SN 以超立方体结构组织，利用超立方体结构高效的广播特性可以实现将事件(或订阅)遍历到整个代理网络。由于底层每个 Chord 环除了簇首节点外，还有候选簇首节点，因此上层结构即使有 SN 离开，其位置也会由候选簇首节点填补。

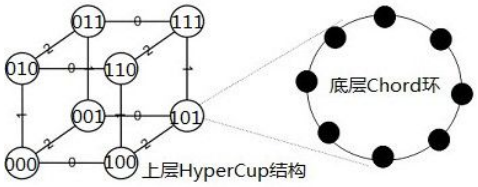

图 2 基于结构化 P2P 的发布订阅系统

2.1 上层 Hypercube 结构

文献[5,6,7]提出了超立方体结构(Hypercube)，这是一种具有高效广播和搜索特性的 P2P 结构化拓扑，它能保证超立方体上的所有节点只接收和转发一次消息，从而使得转发的消息数量最小化。

一个完整的 n 维超立方体结构由 N=2n 个节点组成，每一个节点采用 BRGC(Binary Reflected Gray Code)编码后得到一个长度为 n 的二进制编号，相邻两个节点其编号只有一位不同。每个节点的度为 n，即每个节点有 n 个邻居节点。如果两个节点的编号只有一位不同(假设为第 i 位，)，则这两个节点之间就存在一条边，称为 i 维边，这两个节点互为对方的第 i 个邻居节点。由此可见，超立方体结构是对称的，这对于负载平衡来说很关键，每个节点都可以成为信息的广播源，实现负载均衡。

基于超立方体结构的广播算法如下：某个节点发起一次广播，它先将消息广播到它所有的邻居节点。接收到消息的节点把这一消息向它的邻居节点进行组播，遵循“邻居节点边编号比消息来源的节点边编号大”的原则。该算法保证了每个节点只会接收和转发一次广播的消息，所被广播的消息总和为 N-1 个，并且达到最后一个节点的广播消息经过 n 跳。

2.2 底层 Chord 环

Chord[2]协议算法是由麻省理工学院在 2001 年提出的一种分布式查询协议。Chord 实现的操作是利用相容散列函数 SHA-1[13]给每个节点和资源都分配一个关键字。定义节点 IP 地址为 IP_Address，资源(订阅和

事件)为 Resource，Chord 环节点标识符 nodekey 和资源分配和定位标识符 key 值为 Chord 环的两个键值，并且 nodekey=hash(IP_Address)，Chord 环 key=hash(Resource)。资源在网络中的存放规则是：关键字标识为 key 的资源信息存放在节点标识为 successor(key)的节点上，successor(key)是节点标识大于或等于 key 的第一个节点，称为 key 的后继节点。

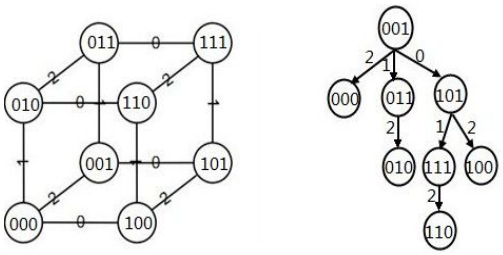

图 3　3 维完整超立方体及节点 1 的组播树

Chord 网络中每个节点维护一张称为 Finger 的路由表。在 Finger 中有 m(每个标识的位数)个表项，其中第 i 个表项为 Chord 环上标识符等于或大于(nodekey+2i-1)mod 2m 的第一个节点。任何一个节点收到查找关键字 key 的请求时，首先检查自身节点是否等于 key，如果是的话就直接回应查找节点，否则，节点将查找它的路由表，找到表中最大但不超过 key 的第一个节点，并将这个查找请求转发给该节点。通过重复这个过程，最终可以定位到 key 的后继节点，即存储有 key 的节点。图 1 显示 m=6 时从节点 8 开始查询 key=54 的关键字的过程，N56 的 IP 地址就是 N8 要查询的目标。

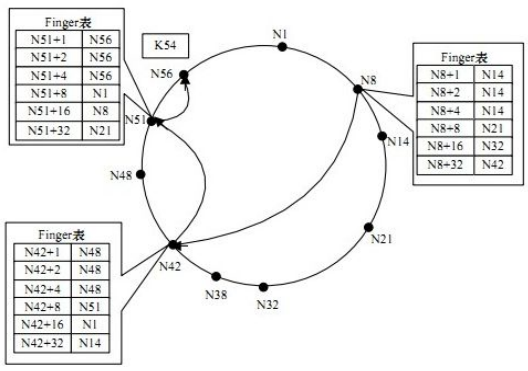

图 4　节点 N8 查找 key=54 的过程

3　Pub/Sub系统事件路由研究

Pub/Sub 系统的消息传递模式，要求任何一个提交的订阅都要遍历到代理网络中所有已发布的事件；任何一个发布的事件都要遍历到代理网络中所有已提交的订阅。由于 Pub/Sub 系统的匿名性，事件发布时，事件的接收者都是不确定的，必须先进行广播以确定事件的接收者。已有的基于内容的路由算法都是以某种广播协议为基础，采用一些优化措施，避免不必要的消息转发，减少网络负载。本文以“基于源转发”的超立方体结构和 Chord 环的散列定位算法为基础，实现匹配优先的路由算法。

3.1 基于结构化 P2P 的 Pub/Sub 系统完整路由算法

对于完整的 Pub/Sub 系统路由，需要解决的问题包括：订阅传播、事件传播、订阅取消等问题。基于结构化 P2P 的 Pub/Sub 系统完整路由算法如下：

3.1.1 订阅传播

当客户向某簇内 CNi 提交订阅时，该普通节点对订阅的关键字进行散列，获取订阅的散列值 S-K(Subscription-Key)。在簇内依据 Chord 的定位算法，把该订阅传递到相应负责的节点并存储在订阅集中；并将该订阅与负责节点中的事件集进行匹配，若成功匹配，则将该事件沿订阅的反向路径传递给 CNi。

CNi 还需将此订阅传递给簇首 SNi，SNi 把该订阅按照以 SNi 为根构造的组播树，高效地传递到其它所有的簇首中，收到订阅的簇首再在本簇内进行 Chord 的定位算法，把该订阅传递到相应负责的节点并存储在订阅集中，同时将该订阅与负责节点中的事件集进行匹配，如有匹配成功的事件，首先把该事件传递给本簇簇首，簇首再按以 SNi 为根构造的组播树反向路由算法传递给 SNi，最后 SNi 将该事件传递给 CNi。

3.1.2 事件传播

当有一个客户向某簇内 CNi 发布事件时，该 CNi 对此事件的关键字进行散列，得到事件的散列值 N-K(Notification-Key)，在本簇内进行 Chord 的定位算法，把该事件传递到相应负责的节点 CNk 并存储在事件集中，同时将该事件与此 CNk 中的订阅集进行匹配，如有匹配成功的事件，并且提交该订阅的 CNj 与 CNi 在同一簇内时，则把该事件按订阅传播的路径原路返回；如果 CNj 与 CNi 不在同一簇内时，CNk 直接将此事件传递给簇首 SNm，SNm 按照订阅反向路径将事件传递给 SNn (SNn 是提交该订阅的 CNj 的簇首)，最后 SNn 将该事件传递给 CNj。

3.1.3 订阅取消

取消订阅信息按照 3.1.1 中所提到的订阅传播方法进行传播。当簇内的节点收到取消订阅消息时，将相应的订阅从订阅集中删除。

3.2 事件路由算法分析

假设 Chord 环中有 N 个节点，key 采用 m 位表示，网络中的订阅总数为 S，事件总数为 Q，基于 Chord 环的性质可以得到如下的算法性能：

(1) 路由表的大小。在 Chord 环中， Finger 表中最多拥有 m 条记录；

(2) 路由跳数。通常 Chord 环中路由跳数为 O(log N)[4]；

(3) 负载大小。由于 Chord 协议使用相容散列，可以把资源均衡的分配到环上的每一个节点，所以簇内每个节点平均负担的订阅为 S/N。同样，若每个簇内的事件数为 Q/N，则簇内每个节点平均负担的事件为 Q/N2；

(4) 维护开销。在 N 个节点的 Chord 环中，当有一个节点加入或离开时，重新建立路由信息和 Finger 表所需的开销为 O(log2 N)条消息，所需要移动的订阅和事件量为 O(log(S/N+Q/N2))。

4 系统仿真分析

簇内订阅和事件路由性能由 Chord 环算法保证，3.2 节给出了 Chord 环算法的性能分析。由于模型上层结构对整个 Pub/Sub 系统有着直接的影响，因此本仿真只验证 Hypercube 高效的广播算法。

实验仿真主要考察以下性能指标：

消息的覆盖率：当网络中一个代理发布消息(事件/订阅)后，能准确接收到此消息的代理节点占整个上层 Hypercube 网络中簇首节点的比率：

$$P_{covering} = D_{receive} / D_{all} \quad (1)$$

其中 $D_{receive}$ 表示收到消息的代理数目， D_{all} 表示上层 Hypercube 网络中簇首节点的数目。

对于本文提出的分层模型的 Pub/Sub 系统而言，只有当 $P_{covering} \approx 100\%$ 时，算法才是合适的，否则将破坏 Pub/Sub 系统的生存性原则。

消息的重复收到率：指网络中某个代理发布一个消息后，其它代理节点重复收到此消息的比率：

$$P_{repeating} = g_{repeating} / g_{new} \quad (2)$$

其中， $g_{repeating}$ 表示单个代理节点重复收到某消息的数目，g_{new} 表示单个节点收到新消息的数目。对于整个网络，其重复收到率则为所有节点重复收到某消息的数目总和与所有节点收到新消息的数目总和的比值。

消息的重复收到率反应了组播树对系统性能的影响，重复收到率少反映系统为满足一定的覆盖率所花费的开销小、网络负载小。

仿真建立在一台 Pentium(R)Dual-Core CPU E5300、内存 2GB 主机，运行在 Windows 7 操作系统的 J-sim 仿真平台中。在仿真过程中，实现了著名的 Flooding 算法、Gossip 算法，并与本文中超立方体结构广播算法进行对比。

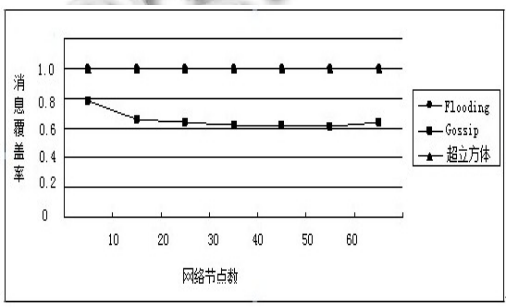

图 5 消息的覆盖率

图 5 显示了在不同网络规模下传统遍历算法及本文提出的超立方体广播算法的消息覆盖率，其中 Flooding 和基于超立方体的广播算法的覆盖率接近 100%，Gossip 算法的覆概率较差，且随着网络规模的增大有恶化的趋势。

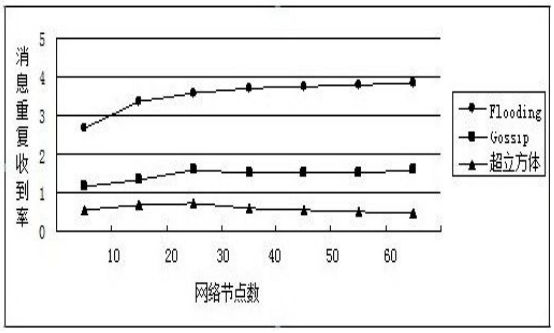

图 6 消息的重复收到率

图 6 显示了在不同网络规模下遍历算法 Flooding、Gossip 和超立方体广播算法的消息重复收到率，可以看出 Flooding 算法要收到 3.5 个重复的构造信息，Gossip 平均收到大概 1.5 个重复构造信息，而超立方体广播算法只有 0.5 个左右。

实验表明，节点以超立方体结构组织，通过超立

方体的广播算法一方面可以实现全网络的遍历，另一方面节点收到的消息冗余度较低，减少网络负载。

5 结语

由于 Pub/Sub 系统的匿名特性，事件发布和事件订阅没有指明具体的接收者，这种不确定性使得其路由算法设计起来比较复杂。所以需要对已有的广播协议进行优化，以抑制不必要消息的转发。同时，由于 Pub/Sub 系统的动态特性，节点频繁加入和退出，因此需要设计一个合理的网络拓扑结构以减少系统维护开销。本文提出基于结构化 P2P 分层次的 Pub/Sub 系统，上层采用超立方体的 P2P 网络，实现整个网络的广播遍历；下层采用 Chord 环结构，节点可动态加入和退出，并通过 Chord 环的高效定位算法实现事件的匹配和路由。本文提出基于结构化 P2P 的 Pub/Sub 系统，需要进一步研究以提高系统性能。例如，通过改进 Chord 算法实现事件的模糊匹配；对于事件或订阅的语义聚集性也有必要进行进一步的研究。另外，Pub/Sub 系统建立在尽力而为的传送机制上，针对实时的发布和订阅关系，保障 Pub/Sub 系统发布方和订阅方之间的 QoS 路由问题，也将受到研究人员的高度关注。

参考文献

1 Oki B, Pfluegl M, Siegel A, Skeen D. The information bus:an architecture for extensible distributed systems. ACM SIGOPS Operating Systems Review,1993,27(5):8–68.

2 Banavar G, Chandra T, Mukherjee B. An efficient multicast protocol for content-based publish-subscribe system. Proceedings of the 19th IEEE International Conference on Distributed Computing Systems. New York:ACM Press, 1999: 262–272.

3 Costa P, Migliavacca M, Picco GP, Cugola G. Epidemic algorithms for reliable content-based publish-subscribe: an evaluation. Proceedings of the 24th International Conference on Distributed Computing Systems. Washington: IEEE Computer Society Press,2004:552–561.

4 Stoica I, Morris R, Karger D, Kaashoek F, Balakrishnan H. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. Proceedings of the ACM SIGCOMM'01,San Diego,CA.2001–08.

5 Schlosser M. Sintek M, Decker S, Nejdl W. Shaping Up Peer-to-Peer Networks. [2007–08–23]. http: //infolab. stanford.edu/~schloss/docs/HyperCuP-DISC2002.

6 Schlosser M, Sintek M, Decker S, Nejdl W. HyPerCuP–HyPercubes, ontologies and Efficient Search on P2P Networks. APZPC, 2002.

7 Schlosser M, Sintek M, Decker S, Nejdl W. Ontology–Based Search and Broadcast in HyPerCup. ISWC,2002.

8 马建刚,黄涛,汪锦岭,徐罡,叶丹.面向大规模分布式计算发布订阅系统核心技术.软件学报,2006,17(1):134–147.

9 Rowstron A, Kermarrec AM, Castro M, Druschel P. SCRIBE: The design of a large-scale event notification infrastructure. In:Proc.of the 3rd Int'1 Workshop on Networked Group Communication.London:Sprthger-Verlag, 2001:30–43.

10 Pietzuch PR, Bacon JM. Hermes:A distributed event–based middleware architecture. 22nd International Conference on Distributed Computing Systems Workshops. Vienna: ICDCS'02,2002.

11 Zhang H, God A, Govindan R. Improving lookup latency in distributed hash table systems using random sampling. IEEE/ACM Transactions on Networking. NJ, USA:IEEE Press Piscataway,2005:1121–1134.

12 Ratnasamy S, Francis P, Handley M, Karp R, Shenker S. A Scalable Content–Addressable Network. Proceedings of the SIGCOMM. New York:ACM,2001:161–172.

13 Rowstron A, Druschel P. Pastry:Scalable distributed object location and routing for large-scale peer-to-peer systems. Proceedings of ACM/IFIP/USENIX Middleware Confere-nce. New York: Middleware 2001,2001:329–350.

14 Zhao B, Kubiatowicz J, Joseph A. Tapestry: An Infrastructure for Fault-Tolerant Wide-area Location and Routing. UCB/ CSD–01–1141:Berkeley,University of California 2001: 726–755.