

内容中心网络中基于缓存索引的缓存路由机制

王好齐,王 雷,夏 莹,王 格

(中国科学技术大学 自动化系,安徽 合肥 230027)

摘 要: 从内容中心网络内置缓存可用性出发,提出了一种基于缓存索引的缓存路由机制. 对传统 CCN 路由器转发引擎模块进行改进,在保留 CS, PIT 以及 FIB 的同时,添加缓存索引表模块,详细描述了缓存索引表表项的创建以及管理流程,并结合缓存索引表提出一种基于缓存索引的缓存路由机制. 结合该路由机制转发用户产生的兴趣包,可以有效减少服务器的命中率以及转发兴趣包的跳数,仿真结果验证了其有效性.

关键词: 网络内置缓存; 内容中心网络; 缓存路由; 缓存索引表

中图分类号: TP391

文献标识码: A

文章编号: 1000—7180(2016)09—0055—05

Cache-Routing Scheme Based on Cache Index in Content-Centric Networking

WANG Hao-qi, WANG Lei, XIA Ying, WANG Ge

(Department of Automation, University of Science and Technology of China, Hefei 230027, China)

Abstract: Taking availability of in-networking cache in content-centric networking into consideration, a cache-routing scheme based on cache index (CRCI) is proposed in this paper. Forwarding engine module of router in traditional CCN is modified in this paper. Remaining CS, PIT and FIB, a cache index table (CIT) is added to a router. The way to create a CIT entry and manage the entry in CIT is expatiated, as well as, coordinated with CIT, a cache-routing scheme based on cache index is proposed. The scheme proposed can forward interest packet generated by user efficiently, decrease sever hit ratio and hops to forward interest packet. Finally, the simulation prove the feasibility of the scheme by comparing with existing schemes.

Key words: in-network caching; content-centric networking (CCN); cache-routing; cache index table

1 引言

信息中心网络 (Information-Centric Networking, ICN) 采用以信息为中心的通信模型, 以取代传统的以地址为中心的网络通信模型, 通信模式从主机演进到网络, 被认为是未来网络演进的重要方向之一. 在这个方向上, 有很多研究者开始进行研究和探索, 包括 DONA^[1] 和 CCN^[2] 等. 其中 CCN 的网络设计更具代表性, 已成为下一代网络架构的研究热点.

在内容中心网络 CCN 中, 以内容为中心, 每一个内容都用名称来标示, 路由器根据名字进行路由. 数据返回的过程中, 沿路径的路由器会缓存数据, 以

便快速响应下一个请求. 因此网络内置缓存 (in-network caching) 是内容中心网络的一个重要特性. 用户可以在网络内找到多个内容副本, 而不需要到很远的内容源提取内容. 为了提高网络内缓存的性能, 很多的研究将目光和关注点放在了缓存决策上. Sung-Won Lee 等人提出 CC-CCN^[3], 根据全局的信息, 结合每个节点的缓存能力对内容进行选择性缓存. Psaras I 等人提出的 ProbCache^[4] 是一种加权概率的缓存机制, 返回路径中的沿途节点缓存对象的概率与节点和请求者之间的距离成反比.

这些研究仅仅优化了网络内置缓存的缓存放置问题, 还需要对缓存路由进行研究, 提高缓存的可用

性,使得用户可以在网络中较近获取内容副本. L. Zhang 等人提出一种泛洪模型的内容路由方案,但是这种方案效率低,代价昂贵^[5]. Munyoung Lee 等人提出的 SCAN 是一种可扩展的内容路由,在一定程度上改善了扩展性问题. 但是在 SCAN 的设计中,路由器需要同时维护 CRTs 和 IP 路由表,这增加了路由器的负担,另外如果缺乏了 IP 路由的支持,会导致路由负担加重以及不稳定的通信^[6].

本文提出一种基于缓存索引的缓存路由机制 (Cache-Routing Scheme Based on Cache Index, CRCI). 通过对 CCN 路由模块进行修改和增强,并提出缓存路由算法,可以减轻服务器的压力,减少转发兴趣包的跳数.

2 CCN 中兴趣包和数据包处理

在 CCN 的设计中,存在两种包格式,分别为兴趣包 (Interest Packet) 和数据包 (Data Packet). 用户产生兴趣包请求内容,内容源返回数据包响应请求的用户. 兴趣包处理流程如图 1 所示.

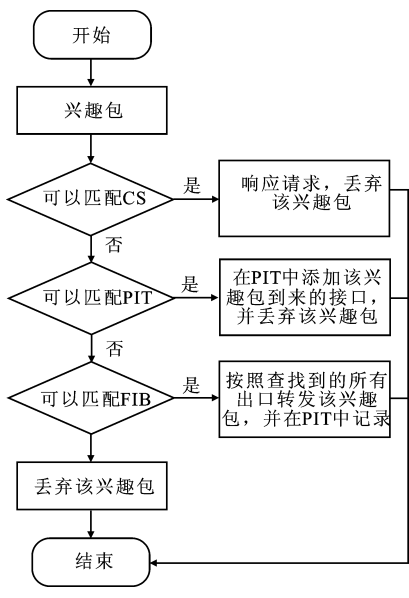


图 1 兴趣包处理流程图

当一个兴趣包 (Interest) 到达一个路由器时, 路由器会根据兴趣包的内容名 (Content Name) 查找 CS, 如果缓存中有被请求的内容, 响应该请求, 并丢弃该兴趣包. 如果没有, 查找 PIT, 如果 PIT 中有该内容名条目, 则在该条目中增加兴趣包到来的端口 (face), 并丢弃兴趣包. 如果 PIT 中没有该内容名条目, 则查找 FIB. 如果在 FIB 中找到, 则按照查找到的所有端口转发兴趣包, 并在 PIT 中记录. 如果 FIB 中也没有该内容名条目, 则直接丢弃该兴趣包.

相较于兴趣包, 数据包处理要简单. 数据包 (Data) 到达路由器, 根据内容名查 CS, 如果 CS 中已经存在, 则丢弃该数据. 如果没有, 则查找 PIT, 如果找到, 则根据查找到的所有端口将数据包转发出去, 然后缓存在 CS 中. 如果在 PIT 中也没有找到, 则直接丢弃该数据包. 如图 2 所示.

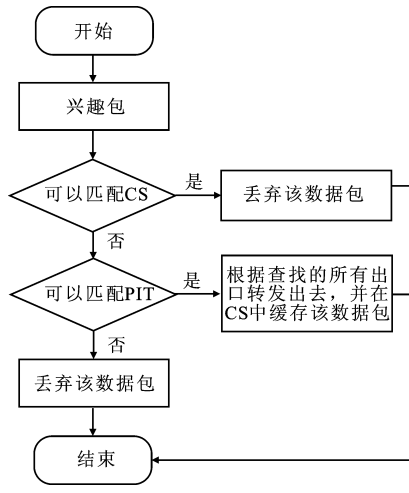


图 2 数据包处理流程

3 缓存索引表及缓存路由算法设计

本文提出的方案对传统的 CCN 路由结构进行修改和增强, 在保留 CS, PIT 和 FIB 的同时, 增加一个缓存索引表 (Cache Index Table, CIT) 模块, 并提出结合 CIT 的缓存路由算法.

3.1 缓存索引表

原始的 CCN 路由结构包含 3 个模块, 内容存储 CS (Content Store), 待定请求表 PIT (Pending Interest Table) 和前向转发表 FIB (Forwarding Interest Table).

CCN 中每个内容都有内容标示 Content Name (CN). CS 中存储的内容名 CN 以及对应的数据. PIT 存储兴趣包所请求的内容名以及兴趣包所到来的端口 Face. PIT 表项中一个内容名可以对应一组端口. FIB 相当于 IP 网络中的 IP 路由表, 存储兴趣所请求的内容名以及可以转发的下一跳出口. 同 PIT 一样, FIB 表项中一个内容名也可以对应一组转发出口.

传统 CCN 对于兴趣包的处理, 是经过 CS, PIT 以及 FIB, 逐步转发, 最终到达内容源. 本文对 CCN 的路由模块进行改进, 添加一个内容索引表 CIT 模块, 由图 3 可以看到, CIT 表项的形式和 PIT 相近, 但两者是不同的, PIT 是根据用户产生的兴趣包生成, 而 CIT 则由返回的数据包生成. 沿途的路由器

在转发数据包的同时,添加 CIT 表项. CIT 表项的形式为内容名和兴趣包到来的端口. 可见 CIT 是指向网络中缓存的一个索引,它使得网络中的缓存可见,使得用户可以就近获取缓存,而不需要到内容源提取内容. 关于 CIT 表项的创建和管理的详细过程,将在后面的部分一一介绍.

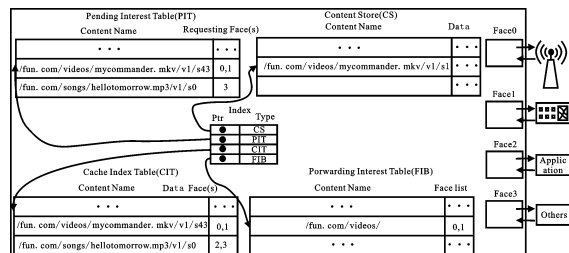


图3 加入 CIT 的转发引擎模型

3.2 创建 CIT 表项

CCN 中默认沿路径的所有节点都进行缓存,这样不仅增加网络的成本,还会造成数据冗余. 本文假设,用户获得请求的数据后,仅在上游节点缓存该内容,转发的路由器建立该缓存的索引信息. 用户发出兴趣包请求内容名为 Content Name 的内容(下文简记为 CN). 转发兴趣包时,路由器统计经过的跳数 HopCount,简记为 HC. 到达内容源后记录总跳数 TotalHops,简记为 TH. 数据返回过程中,沿途的路由器添加缓存索引信息,索引信息的格式是:内容名+兴趣到来的端口.

当 TH 为偶数时,若 $2 \leq HC \leq TH/2$,则相应的路由器建立索引信息. 与 $HC \leq TH/2$ 路由器相连的用户产生的兴趣包将转发到缓存节点,而与 $HC > TH/2$ 路由器相连的用户产生的兴趣包将转发到内容源.

当 TH 为奇数时,若 $2 \leq HC \leq (TH-1)/2$,则相应的路由器添加索引信息,若 $HC \geq (TH+1)/2$,则不需要添加索引信息. 实际上,与 $HC = (TH+2)/2$ 路由器相连的用户距离缓存节点以及内容源的跳数相同,但根据本文的设计, $HC = (TH+1)/2$ 路由器不需要添加索引信息. 因为,内容源的稳定性以及可达性要优于缓存节点,确保用户可以获取内容. 根据前述的方法,不仅可以减少路由器需要维护的索引信息项,还可以使得用户就近获取内容. 图4描述了添加 CIT 条目的详细流程,为了便于阅读和理解,发出兴趣包的用户跳数统计为 0.

(1) 用户节点 h1 发出兴趣包,请求内容名 CN1 的内容,此时 $HC(CN1) = 0$. 路由器将兴趣转发至内容源 Sever1. 转发过程中,经过路由器 S1, S2, S4, S5, S7, S8, 这时, $HC(CN1) = 1$, $HC(CN1) = 2$, HC

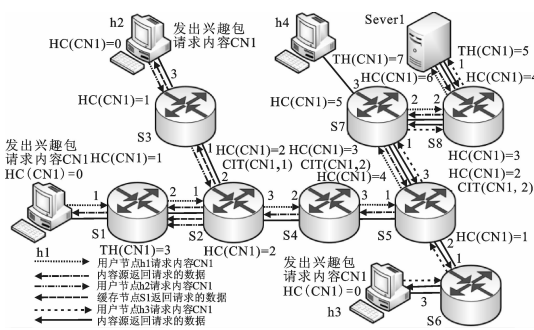


图4 创建 CIT 表项流程

(CN1) = 3, $HC(CN1) = 4$, $HC(CN1) = 4$, $HC(CN1) = 5$, $HC(CN1) = 6$, 到达内容源 Sever1 时, $TH(CN1) = 7$.

Sever1 收到兴趣包后,返回数据包,与用户节点 h1 直接相连的路由器 S1 缓存该数据包. $TH(CN1) = 7$ 为奇数,根据前述设计,当 $2 \leq HC(CN1) \leq (TH(CN1) - 1)/2 = 3$ 时,建立 CN1 的索引信息,因此 S4, S2 添加索引项, $CIT(CN1, 2)$, $CIT(CN1, 1)$, 而 S8, S7, S5, S4, S1 则不需要添加索引项.

(2) 用户节点 h2 发出兴趣包,请求内容名 CN1 的内容,此时 $HC(CN1) = 0$. S3 将兴趣包转发至 S2, 由于 S2 里存有 CN1 的索引信息 $CIT(CN1, 1)$, 因此, S2 将按照查找到的转发端口将兴趣包转发出去,到达 S1. 由前述可知, S1 已经缓存 CN1 的内容,响应用户节点 h2. 转发的过程中,经过 S3, S2, 这时, $HC(CN1) = 1$, $HC(CN1) = 2$, 到达缓存节点即 S1 时, $TH(CN1) = 3$.

S1 收到兴趣包后,返回数据包, S3 缓存该数据包. 当 $2 \leq HC(CN1) \leq (TH(CN1) - 1)/2 = 1$ 时,建立 CN1 的索引信息,因此 S2 不需要添加索引信息.

(3) 用户节点 h3 发出兴趣包,请求内容名 CN1 的内容,此时 $HC(CN1) = 0$. S6 将兴趣包转发至 S5. 由前述可知, S5 并没有添加 CN1 的索引项,因此 S5 将兴趣包转发至 S7, S7 转发至 S8, S8 转发至 Sever1. 转发过程中,经过 S6, S5, S7, S8, 这时, $HC(CN1) = 1$, $HC(CN1) = 2$, $HC(CN1) = 3$, $HC(CN1) = 4$, 到达 Sever1 时, $TH(CN1) = 5$.

Sever1 收到兴趣包后,返回数据包, S6 缓存该数据包. 当 $2 \leq HC(CN1) \leq (TH(CN1) - 1)/2 = 2$ 时,建立 CN1 的缓存索引信息,因此 S5 添加索引项, $CIT(CN1, 2)$.

3.3 管理 CIT 表项

路由器需要对缓存索引表项 CIT 中的条目进行维护,以减少条目的冗余,提高查找条目的效率. 路由器

根据三种情况来对 CIT 条目进行管理和维护.

- (1)缓存节点断开连接.
- (2)缓存节点清除缓存内容.
- (3)路由器给每一个 CIT 条目设置 TTL,定期清除条目.

对于第一种情形,实际的网络节点加入和退出网络是很常见的情况,路由器会定期的发出扫描信息,探测缓存节点是否可达,如果缓存节点已经退出网络,路由器则清除 CIT 条目中对应的端口. 路由器发出扫描信息的过程称为主动扫描过程.

对于第二种情形,由于缓存节点的缓存空间有限,缓存节点会根据替换策略清除缓存,这时缓存节点发出一个不可用(Unavailable)信息给路由器,路由器收到该信息后,清除 CIT 条目中对应的该端口. 缓存节点发出 Unavailable 给路由器的过程称为被动过程.

对于第三种情况,路由器对经过的兴趣包进行统计,进行流行度的分析,对内容进行分级. 内容的等级越高,设置的 TTL 越高;内容的等级越低,设置的 TTL 越低.

3.4 结合 CIT 的缓存路由算法

这一部分,结合 CIT 设计基于缓存索引的缓存路由算法(Cache-Routing Scheme Based on Cache Index,CRCI).

Algorithm 1:Cache-Routing Based on Cache Index Algorithm
A router receive Interest Packet generated by a user

- 1. Search in CS
- 2. If match CS, Then response user, delete the Interest
- 3. Else search in PIT
- 4. If match PIT, Then
- 5. Add the arrival Face of Interest Packet to PIT, delete the Interest
- 6. Else
- 7. If there is CIT in router, Then
- 8. search in CIT
- 9. If match CIT, Then
- 10. forward the Interest to cache node by index and response
- 11. Else search in FIB
- 12. If match FIB, Then
- 13. forward the Interest to content server and response
- 14. Else delete the Interest
- 15. Else search in FIB
- 16. Similar to line 12—14

用户节点产生兴趣包请求内容,路由器依次查询 CS,PIT,如果路由器有 CIT 表项,查找 CIT. 如果 CIT 中有该内容条目,按查找到的端口转发出去,根据索引查找缓存节点. 如果缓存节点无法响应

内容,则查找 FIB. 如果路由器中无 CIT 表项,直接查找 FIB.

4 仿真与性能分析

为了证明上述基于缓存索引的缓存路由机制的有效性,本文进行了仿真实验. 采用的网络共有 50 个节点,5 000 个内容,内容流行度遵循 Zipf 分布. 节点随机发出发起请求,发起请求时间符合泊松分布, $\lambda=5$ Reqs/s. 本文使用缓存大小(Cache Size)来表示缓存节点的缓存能力,节点的缓存大小定义为每个节点所能存储的内容数与网络中所有内容的比值. 分别可设为 1%(0.01),2%(0.02),3%(0.03),...,8%(0.08). 实验结果与 LCD,LCEd 以及 LCE 比较. 根据文章前述的方案,以平均跳数和服务器的命中率来作为评估基于缓存索引的缓存路由机制的性能指标. 平均跳数(AverHop)为请求节点发出的兴趣包到达具有被请求内容节点的平均跳数. 服务器命中率(Server Hit Ratio)为服务器响应的兴趣包占有被响应的兴趣包的比例.

由图 5 可以看出,随着节点缓存大小的增加,各个方案的服务器的命中率都在下降. LCD 方案的服务器命中率最高,然后依次为 LCEd,LCD,CRCI. 采用 LCD 和 LCEd 的方案,一次请求之后网络中存在的副本很少,而且每个节点的缓存能力有限,缓存节点响应兴趣包的能力也是有限的. CCN 默认采用 LCE 方案,沿路径的每个节点都缓存数据包,当有新的兴趣包产生并获得响应时,缓存节点以新的数据替换旧的数据,另外网络中存在较多的冗余数据. 相较于 LCD,LCEd 以及 LCE,文中提出的 CRCI 性能最佳,能提高缓存命中率,减少服务器命中率,减少了服务器的压力.

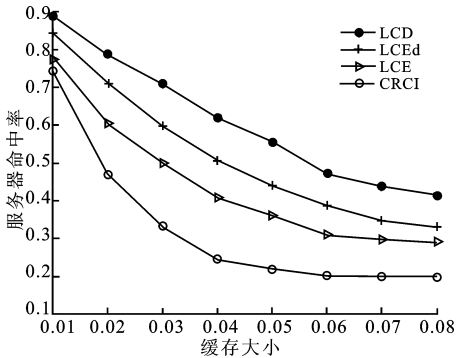


图 5 服务器命中率 VS 缓存大小

由图 6 可以看出,随着缓存大小的增加,各个方案的平均跳数都在下降. 平均跳数最大的是 LCD,然后依次为 LCEd,LCE,CRCI. LCD 方案将内容复

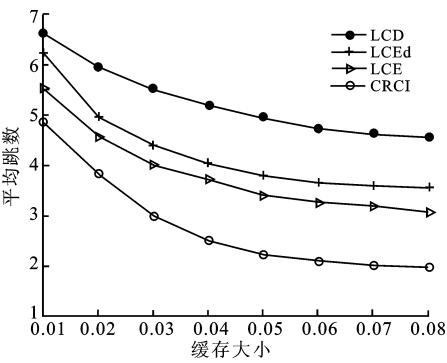


图 6 平均跳数 VS 缓存大小

制到边缘的速度很慢,边缘用户的请求跳数较大.本文提出的 CRCI 可以大大减少兴趣包转发的跳数.收到兴趣包后,如果存在匹配的 CIT 表项,则根据 CIT 转发兴趣包,找到最近的副本放置节点,这样就减少了转发兴趣包的跳数,大大减少了用户获取内容的延迟.

5 结束语

本文从网络内置缓存的可用性出发,提出了一种基于缓存索引的缓存路由机制.通过改进传统 CCN 路由器的转发引擎模块,添加 CIT 模块,详细地介绍了 CIT 表项的创建以及管理流程,并结合 CIT 提出一种基于缓存索引的缓存路由算法 CRCI.仿真实验证明,提出的基于缓存索引的缓存路由机制,可以减少服务器的命中率,减少兴趣包转发的平均跳数.

参考文献:

[1] Koponen T, Chawla M, Chun B-G, et al. A data-ori-

ented (and beyond) network architecture [C]//Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM). New York, NY, USA: ACM, 2007: 181-192.

[2] Jacobson V, Smetters D K, Thornton J D, et al. Networking named content [C]//Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies (CoNEXT). New York, NY, USA: ACM, 2009: 1-12.

[3] Sung-Won Lee, Dabin Kim, Young-Bae Ko, et al. Cache capacity-aware CCN: selective caching and cache-aware routing [C]// Global Communications Conference (GLOBECOM). Atlanta, GA: IEEE, 2013: 2114-2119.

[4] Psaras I, Chai WK, Pavlou G. Probabilistic in-network caching for information-centric networks [J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 25(11): 2920-2931.

[5] L Zhang, D Estrin, J burke, et al, Named data networking (NDN) project, NDN0001 [R]. PARC, 2010.

[6] Munyoung Lee, Kideok Cho, Kunwoo Park, et al. SCAN: Scalable content routing for content-aware networking [C]//IEEE International Conference on Communications (ICC). Kyoto: IEEE, 2011: 1-5.

作者简介:

王好齐 男,(1991-),硕士研究生.研究方向为未来网络.
E-mail: hyhqwang@mail.ustc.edu.cn.

(上接第 54 页)

8 结束语

根据 PCIe 2.0 协议要求设计的 EB,读写时钟可以达到 250 MHz,完全符合协议的全速率和半速率要求.由此可见,基于指针的 SKP 字符的插入和删除操作同样可以完成时钟补偿的操作.

参考文献:

[1] PCI-SIG. PCI Express 2.0 Base Specification Revision 0.9. [EB/OL]. [2015-11-20]. www.pci-sig.com, 2011.

[2] 朱小明,王小利,程曾. USB3.0 物理层中弹性缓冲的设计

与实现[J]. 微电子学与计算机, 2012, 29(6): 117-118.

[3] Cummings CE. Simulation and synthesis techniques for asynchronous FIFO design[R]. SNUG, 2002.

[4] Joe Winkles. Elastic Buffer Implementations in PCI Express Devices[M]. US: MindShare Inc, 2003.

[5] Intel Corporation. PHY Interface for the PCI Express Architecture Version 2.0[P]. 2007.

作者简介:

武桂林 女,(1990-),硕士研究生.研究方向为数字集成电路设计. E-mail: wgl157@mail.ustc.edu.cn.

黄鲁 男,(1961-),副教授,硕士生导师.研究方向为混合信号 IC 芯片设计.