支持多集群数据并行的 On-demand 文件传输算法

魏晓辉 周芹芹 李洪亮
(吉林大学计算机科学与技术学院 长春 130012)

摘 要 数据密集型应用通常需要在广域网分布式共享计算环境中高效地传输海量数据。并行处理中,大量的数据需要在生成集群、存储集群、处理集群间进行传输。针对该传输问题提出了一个支持多集群数据并行传输的按需文件传输算法(On-demand File Transfer),该算法以批量传输请求的整体完成时间最小为目的,根据集群内部快速传输的特点,实现目的端并行,分散单个节点的传输负载;在传输路径上,采用多重路径和多跳路径分割方法实现并行传输。对于批量传输请求,依据每个请求的传输负载,全局按需分配带宽,以解决传输路径的带宽冲突,从而充分利用当前网络带宽,快速传输批量传输请求。

关键词 多集群,目的端并行,按需,文件传输
中图法分类号 TP301 **文献标识码** A

On-demand File Transfer Algorithm Supporting Data-parallel over Multi-cluster

WEI Xiao-hui ZHOU Qin-qin LI Hong-liang
(College of Computer Science and Technology, Jilin University, Changchun 130012, China)

Abstract Data-intensive applications often require the efficient transfer of large amounts of data over wide-area distributed computing environments. In parallel processing, such data transmissions usually happen between data generation clusters and the data storage or processing clusters from time to time. This paper presented an on-demand file transfer algorithm, which aims at minimizing the makespan of overall file transfer requests. Taking advantage of the high transfer performance within a single cluster, the algorithm performs simultaneous transfer of different pieces of files to multiple nodes of a cluster when one node of the cluster requests the files. It employs multi-hop path splitting and multi-pathing methods in [1] to improve the transfer performance of a single source-destination pair. When several requests compete for the physical links, the algorithm evenly allocates the bandwidth resources upon the transmission load of each request, to make full use of the network bandwidth.

Keywords Multi-clusters, Data intensive, On-demand schedule, File transfer

1 概述

近年来,生物工程、高能物理、天气预报、地质研究等数据密集型应用逐渐发展成为高性能计算领域的典型应用。这类应用通常需要大量的计算和存储资源,数据也分布存储在不同的集群中,需要多个跨局域网或管理域的集群协同工作。因此,并行处理中,如何在多集群间快速传输大文件,逐渐成为研究的热点。

并行处理中的文件传输,是一种从多源到多目的的传输,当前的研究重点都是如何提高传输的并行度,缩短传输时间。有些工作使用多文件副本实现源端并行,见文献[2],或者采用分片传输,增加文件副本数,提高源端并行度,见文献[3]。但现有的研究并没有考虑到目的端并行,然而集群内部能够快速传输文件,多个目的节点可同时接收文件的不同分片,可实现目的端并行。对于传输路径,有些工作采用多跳路径分割的方法获得较好的路径,见文献[8];还有些工作采用多重路径和分片传输的方法实现单个源目的对的并行传输,见文献[1,5]。然而,文献[5]参考文献[8]的测试结果,选取跳数不超过3的路径来获取路径分割带来的最大收益,这一限制使得单个源目的对可能找不到连通路径而无法灵活调整。另外,虽然是以批量传输请求的整体完成时间最小作为调度目的,但在实际的调度中,文献[5]采用的是最小作业优先的调度方法,而并没有协调多个请求之间的带宽冲突,使得算法在系统带宽利用率方面存在一定不足,从而影响整个完成时间。

针对上述问题,本文提出了一个支持多集群数据并行的按需文件传输算法(OFT)。首先,OFT利用集群内部快速共享的特点,实现目的端的并行接收与组装。本文将目的节点为同一集群的传输请求合并成一个请求,将该请求分配给集群中的多个节点来分散传输负载。使用多跳路径分割优化传输路径时,对于单个源目的对,选出一条最优路径,最优路径的跳数加上一个可调范围值(如2)作为所有路径跳数的上限,在实现源目的对连通的基础上获得路径分割带来的性能改善。对多个请求间的路径冲突进行处理时,根据每个请求

到稿日期:2012-10-19 返修日期:2012-12-21
魏晓辉(1972-),男,博士,教授,博士生导师,主要研究方向为分布式与网格系统,E-mail:weixh@jlu.edu.cn;周芹芹(1987-),女,硕士,主要研究方向为分布式与网格系统;李洪亮(1983-),男,博士,主要研究方向为分布式与网格系统。

的传输负载按比例为其分配带宽，使得多个请求的传输时间尽可能相同，从而缩短批量请求的整体传输时间。

2 多集群间文件传输模型

2.1 假设

假设1 集群内部的节点通过高速网络互连，并通过相对低速的网络与外部相连。因此本文认为，当文件在集群间传输完成后，集群内部节点的文件传输也基本完成，即在文件的整体传输时间中忽略不计集群内部文件的传输时间。

假设2 在多集群环境中，文件是非集中存储的。数据文件、执行文件等分散在各个集群的不同服务器中。

2.2 网络模型

$G=(V,E)$，V：节点集合 $V=\{v_i\}$，v_i 中保存文件副本 $\{f_1,f_2,\cdots\}$，E：节点间边的集合 $E=\{e_{ij}\}$，$e_{ij}:v_i->v_j$。

Cluster：网络中的集群，Cluster$=\{V_C,E_C\}$，V_C：集群中的节点，E_C 是集群内部的边，用来连接集群内部节点的边。集群中任意节点都可能存在与集群外节点的互连，以下简称Cluster为 c。

边 e_{ij} 的权重 $w_{ij}=$ BW/RTT(带宽/响应时间)。

$R=\{\langle v_i,c_j,f_\ell\rangle\}$，即传输请求集合，$\langle v_i,c_j,f_\ell\rangle$ 表示集群 c_j 中的节点 v_i 请求文件 f_ℓ。

$D=\{\langle v_s,c_j,f_\ell\rangle\}$，即副本集合，$\langle v_s,c_j,f_\ell\rangle$ 表示集群 c_j 中的节点 v_s 保存文件 f_ℓ 的一个副本。

Path$=\langle v_s,v_i,v_j,\cdots,v_d\rangle$，路径由源节点经过若干中间节点到达目的节点，其中任意相邻的两个节点 v_i，v_j 间存在边 e_{ij}，路径的带宽(BW)=Min(BW of e_{ij})；

$r_1=\langle v_2,c_1,f_7\rangle$，即 c_1 中的节点 v_2 请求文件 f_7，存在 $D_1=\langle v_2,c_4,f_7\rangle$、$D_2=\langle v_1,c_3,f_7\rangle$。

$r_2=\langle v_3,c_1,f_8\rangle$，即 c_1 中的节点 v_3 请求文件 f_8，仅存在 $D_4=\langle v_1,c_2,f_8\rangle$。

$r_3=\langle v_2,c_1,f_6\rangle$，即 c_1 中的节点 v_2 请求文件 f_6，存在 $D_3=\langle v_1,c_3,f_6\rangle$。

r_1、r_2、r_3 相关的网络简化为图1。图1中用带宽大小衡量边的优劣，因为大数据量文件传输中，带宽的影响是主要的。

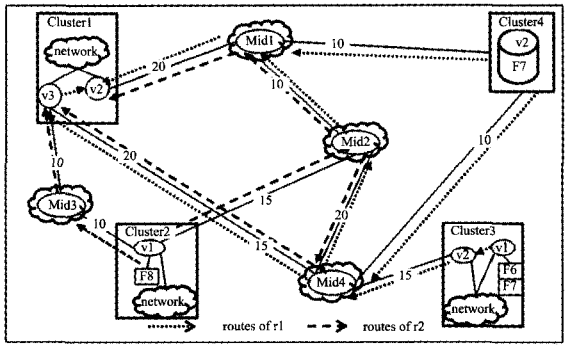

图1 不同传输方式下的传输路径

2.3 传输算法思路

多源传输算法使用多跳路径分割和多重路径的方法，找出所有可能路径，根据跳数限制(如文献[5]中限制为不超过3跳)、路径冲突等选择出一组传输路径集合。本文对多源传输方法进行适度的扩展，主要工作分为以下几步：

第1步 合并请求集合 R，将在同一集群中请求同一文件的请求合并为该集群请求某文件，以避免集群内部与外部网络的多次传输；再将具有相同源节点的文件合并为一个超文件，以避免路径查找算法的多次使用。

$r_1=\langle v_2,c_1,f_7\rangle$，$r_4=\langle v_3,c_1,f_7\rangle=>r_1'=\langle c_1,f_7\rangle$

$r_2=\langle v_3,c_1,f_8\rangle=>r_2'=\langle c_1,f_8\rangle$

$r_3'=\langle c_1,f_1\rangle$，$r_4'=\langle c_1,f_2\rangle$，$sources:f_1=f_2=\{v_m,v_n\}$

$=> r'=\langle c_1,F\rangle$，$F=\{f_1,f_2\}$，$sources$ of $F=\{v_m,v_n\}$

第2步 对合并后的 r' 扩展目的端，此时集群中存在外部连接的节点都可作为目的节点。

r_1' 的目的节点 $Dests=\{v_2$ of c_1,v_3 of $c_1\}$，同样 r_2' 的目的节点也扩展为 $Dests$。

第3步 为 $r'\in R'$ 查找路径。对于每个源目的对，使用多跳路径分割和多重路径的方法查找出多条路径，选取传输性能较好的路径组合。在单个源目的对间，先选出一条最优路径，将最优路径的跳数加上一个可调范围值(本文中为2)作为所有路径跳数的上限。文献[5]参考文献[8]的实验结果，选取跳数不大于3的路径来获得最佳收益。

图1中，对于请求 $r_3=\langle v_2,c_1,f_6\rangle$，只存在一个源文件 $D_3=\langle v_1,c_3,f_6\rangle$，若限制路径跳数(hop)不超过3跳，则无法在源和目的节点之间找到可用路径。

对于 r_1，限制路径不超过3跳时，选出的路径为：$path_1=\langle v_2$ of c_4,mid_1,v_2 of $c_1\rangle$，$hop=2$，$bw=10$；$path_2=\langle v_2$ of c_4,mid_4,v_3 of c_1,v_2 of $c_1\rangle$，$hop=3$，$bw=10$。

使用本文中的方法为 r_1 查找路径：先选出最优路径 $path_1$，$hop=2$，所以路径跳数上限为2+2=4。为 r_1 查找出的路径为：$path_1$，$path_2$，$path_3=\langle v_2$ of $c_4,mid_4,mid_2,mid_1,v_2$ of $c_1\rangle$，$hop=4$，$bw=10$，$path_4=\langle v_1$ of c_3,v_2 of c_3,mid_4,v_3 of c_1,v_2 of $c_1\rangle$，$hop=4$，$bw=15$，$path_2$，$path_4$ 在边 $\langle mid_4,v_3$ of $c_1\rangle$ 上存在冲突，$path_2.bw+path_4.bw=25>15$，因为 $path_2.bw=10<path_4.bw=15$。优先为可获得较大带宽的路径分配资源，$path_2.bw=5$，$path_4$ 不变。

4条路径的带宽和为35，而 $path_1$、$path_2$ 的带宽和仅为20。因此，本文方法通过灵活调整路径跳数限制，获得更多的传输路径和带宽资源，在提高并行度与获得路径分割带来的性能改善间取得平衡。

第4步 当 r' 间竞争物理链路，存在带宽冲突时，根据每个 r' 传输文件的大小，按比例为其分配带宽，使得多个请求的传输时间尽量相同，降低耗时最长请求的传输时间，从而使得整体传输时间最小。

如图1所示，r_1 的4条路径都与 r_2 的路径竞争物理链路的带宽资源。例如，在边 $\langle mid_2,mid_4\rangle$ 中 r_2 和 r_1 期望获得的带宽都为15，而边的实际带宽只有20，即所有通过该边的路径的带宽和不能超过20，所以需要根据 r_3 和 r_1 的传输负载和已获得的带宽为其分配该边的带宽资源。

3 实验结果及分析

本文使用文献[9]中的GT-ITM，生成了5个集群代表中使用的5个实际物理子网络来进行模拟实验。本文测试了在不同的传输模式下文献[5]中的GDS-MHMP-SB(GDS)和本文算法OFT的传输性能。实验结果显示，不同的传输模式下，OFT的传输性能都有很好的改善。有些结果OFT比GDS的吞吐量提高了两倍甚至更多，因为目的端并行很好地分散了传输负载，提高了并行化程度，缩短了传输时间，大大

(下转第94页)

[9] Li Chang-bao, Cheng Bo, Chen Jun-liang, et al. A Web Service Performance Evaluation Approach Based on Users Experience [C]// Proceedings of the IEEE International Conference on Web Services. Washington DC, USA, 2011: 734-735

[10] Li Pei, Comerio M, Maurino A, et al. An Approach to Non-functional Property Evaluation of Web Services [C]// Proceedings of the IEEE International Conference on Web Services. Los Angeles, USA, 2009: 1004-1005

[11] Dou Wan-chun, Lv Chao, Zhang Xu-yun, et al. A QoS-Aware Service Evaluation Method for Co-Selecting a Shared Service[C]// Proceedings of the IEEE International Conference on Web Services. Washington DC, USA, 2011: 145-152

[12] Wang Shang-guang, Zheng Zi-bin, Sun Qi-bo, et al. Cloud Model for Service Selection [C] // Proceedings of the International workshop on Cloud Computing. Kochi, Kerala, India, 2011: 666-671

[13] Wu Jin-hong, Xia Huo-song. Study of Adaptive Qos Evaluation Model for Semantic Web Service[C]// Proceedings of the International Conference on Information Management, Innovation Management and Industrial Engineering. Taipei, Taiwan, China, 2008: 480-483

[14] Han Tao, Guo He-qing, Li Dong, et al. An Evaluation Model for Web Service[C]// Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences. Hangzhou, Zhejiang, China. 2006: 698-701

[15] Chen Li-ping, Ha Wei-tao, Zhang Guo-jun. A new Web Service Evaluation Model with Fuzzy C-Means Artificial Immune Network Memory Classifier[C]// Proceedings of the International Conference on Computational Intelligence and Security. Beijing, China, 2009: 25-29

[16] Lee K-C, Jeon J-H, Lee W-S, et al. QoS for Web Services: Requirements and Possible Approaches[OL]. http:// www. w3c. or. kr/kr-office/TR/2003/ws-qos/, 2003-11-25

[17] 梁保松，曹殿立. 模糊数学及其应用[M]. 北京：科学出版社，2007

[18] 胡运权，等. 运筹学基础及应用(第五版)[M]. 北京：高等教育出版社，2008

[19] Lo Chi-chun, Chen Ding-yuan, Tsai C-F, et al. Service selection based on fuzzy TOPSIS method[C]// Proceedings of the International Conference on Advanced Information Networking and Applications Workshops. Perth, Australia, 2010: 367-372

[20] Zou Hua, Zhang Long-chang, Yang Fang-chun, et al. A Web service composition algorithmic method based on TOPSIS supporting multiple decision-makers[C]// Proceedings of the World Congress on Services. Miami, Florida, USA, 2010: 158-159

[21] Cardoso J, Sheth A, Arnold J, et al. Quality of service for workflows and Web service processes[J]. Web Semantics: Science, Services and Agents on the World Wide Web, 2004, 1(3): 271-308

[22] Zeng L Z, Benatallah B, Ngu A H H, et al. QoS-aware middleware for Web services composition[J]. IEEE Transactions on Software Engineering, 2004, 30(5): 311-327

(上接第 78 页)

提高了吞吐量。有些实验结果显示吞吐量仅提高了 40%左右，其原因在于目的端并行接收受限，只依靠批量传输请求间的按需分配带宽来缩短整体的传输时间。因此 OFT 的吞吐量比 GDS 至少提高 40%。

为了进一步评价算法性能，本文在多到多的传输模式下，测试不同传输负载下 OFT 算法的吞吐量。实验表明，随着传输负载的不断增加，算法的吞吐量也不断提高，当传输负载增长到 20G 时，吞吐量基本稳定，达到 600Mbps。这是由于传输负载相对较小时，文件传输时间相对变小，算法的执行时间所占的比例相对变大，算法吞吐量较小。随着传输负载变大，文件传输时间所占的比例变大，直到算法执行时间所占的比例可以忽略不计，获得稳定的吞吐量 600Mbps。

结束语 本文提出了一个面向并行处理中集群间文件传输的按需文件传输(On-demand File Transfer)算法，该算法以批量传输请求的整体完成时间最小为目的，通过目的端并行，灵活调整路径跳数的上限，依据传输负载按需分配带宽等方法充分利用网络带宽，快速传输批量请求。实验证明，该算法很好地改善了传输性能，比文献[5]中的 GDS-MHMP-SB 算法至少提高了 40%的吞吐量，当文件的传输负载达到 20GB 时，在当前网络获得了最大的吞吐量。本文将在更为复杂的真实网络环境中对算法作进一步的测试和改进。

参 考 文 献

[1] Khanna G, Catalyurek U, Kurc T, et al. Multi-Hop Path Splitting and Multi-Pathing Optimizations for Data Transfers over Shared Wide-Area Networks using GridFTP [C]// Proceedings of the 17th IEEE International Symposium on High-Performance Distributed Computing (HPDC 2008). Cali: Norwell, 2008: 156-168

[2] Khanna G, Catalyurek U, Kurc T, et al. Scheduling file transfers for data-intensive jobs on heterogeneous clusters [C]// Kermarrec A-M, Bouge L, Priol T, eds. Euro-Par, volume 4641 of Lecture Notes in Computer Science. Springer, 2007: 214-223

[3] Rizk P, Kiddle C, Simmonds R. A gridftp overlay network service [C]// Proceedings of the 7th IEEE /ACM International Conference on Grid Computing. Spain: Baercelona, 2007: 282-298

[4] Khanna G, Catalyurek U, Kurc T, et al. Using overlays for efficient data transfer over shared wide-area networks[C]// Proc. of ACM/IEEE SC. 2008

[5] Giersch A, Robert Y, Vivien F. Scheduling tasks sharing files from distributed repositories [C]// Danelutto M, Vanneschi M, Laforenza D, eds. Euro-Par 2004. LNCS, vol. 3149. Springer, Heidelberg, 2004: 246-253

[6] Pucha H, Hu Y C. Overlay tcp: ending end-to-end transport for higher throughput[C]// Poster in ACM SIG-COMM. Philadelphia, PA, 2005

[7] Allcock W, Bresnahan J, Kettimuthu R, et al. The globus striped gridftp framework and server[C]// SC'05: Proceedings of the 2005 ACM/IEEE conference on Supercomputing. Washington, DC, USA, IEEE Computer Society, 2005

[8] Khanna G, Kurc T, Catalyurek U, et al. A dynamic scheduling approach for coordinated wide-area data transfers using gridftp [C]// Proc. of 22th International Parallel and Distributed Processing Symposium(IPDPS). Miami, Florida, 2008

[9] GT-ITM Website[EB/OL]. http://www. cc. gatech. edu/projects/gtitm/, 2007-08-10