多服务级别带宽公平分配算法的研究*)

刘 威 杨宗凯 乐春晖 何建华
(华中科技大学电子与信息工程系 武汉 430074)

摘 要 区分服务网络节点中的多级别的队列输出带宽由权值调度算法保证,固定权值调度在网络负载发生变化时无法继续提供公平的带宽保证。本文提出了一种动态调整权值的调度算法以达到在多服务级别间公平分配带宽。实验仿真表明,该算法可以对负载流数目的变化作出有效的响应,并快速实现调度权值的理想公平值。
关键词 带宽分配,公平性,多服务

Fair Bandwidth Allocation Algorithm for Multiple Service Classes

LIU Wei YANG Zong-Kai LE Chun-Hui HE Jian-Hua
(Dept. of EI Engineering, Huazhong University of Science and Technology, Wuhan 430074)

Abstract Class-based service differentiation is provided in DiffServ networks. However, when it is under dynamic traffic load, this differentiation will be disordered by the fixed weighted bandwidth allocation. In this paper, a fair bandwidth allocation algorithm is proposed for the scheduling among multiple service classes. Simulation results show that the algorithm can quickly response to the changes of the traffic load and adjust the weights to ideal faire values.
Keywords Bandwidth allocation, Fairness, Multiple services

1 引言

多服务级别的网络是解决 QoS 问题的主要途径。以典型的多服务级别的网络区分服务(DiffServ)为例,其可以在传统 IP 网络的尽力服务(BE)以外提供加速型(EF)服务和确保型(AF)服务。区分服务节点上对于多个服务级别的处理队列通常采用权值调度算法,包括加权循环调度(Weighted Round Robin, WRR),加权公平排队调度(Weight Fair Queueing, WFQ)等多种。由于现有的调度算法是依据各服务等级所占用的总带宽的比例权值进行调度的,没有考虑到各服务内负载变化的影响。当网络负载发生变化(如 EF 或 AF 服务的订购带宽、BE 服务的流数目有变,或者有某个服务中出现恶意流量),现有的固定权值的调度方案很难保证在 AF 和 BE 服务中的传输流之间公平地分配带宽资源,可能引发高服务等级 AF 的传输流反而获得较少剩余带宽的情况。

对于该公平性问题,改进调度算法是主要的解决方法。文[1]注意到网络负载发生变化时引发的带宽分配问题,提出对调度权值进行渐进式修正以维持初始比例配置,却忽略了按流分配带宽的公平性问题。文[2,3]以带宽公平性为目标改进调度算法,但在估计用户流数目方面采用了较为复杂的方法,或者通过在分组包头中增加订购带宽信息的方法[2],或者需要在本地维持一个 Zombie list 来计算流数目[3]。

我们将研究问题定于多个 AF 和 BE 服务之间的剩余带宽的合理分配,通过计算本地缓存区中多个队列指标的测量平均值,提出动态调整权值方案以改进带宽分配的公平性。该方案的前期工作[4]取得了良好效果,本文进一步提出更为精确的调度算法。

2 动态权值调度算法

2.1 多服务队列调度系统模型

在区分服务节点输出端的权值调度系统中,存在 1 个 EF、4 个 AF、1 个 BE 构成的多服务队列。其中 EF 服务享有最高优先级并获得固定带宽,多个 AF 服务的优先级次之并至少获得订购带宽,AF 和 BE 服务的传输流共享未订购的剩余带宽。区分服务节点中多服务调度系统的参考模型如图 1 所示。

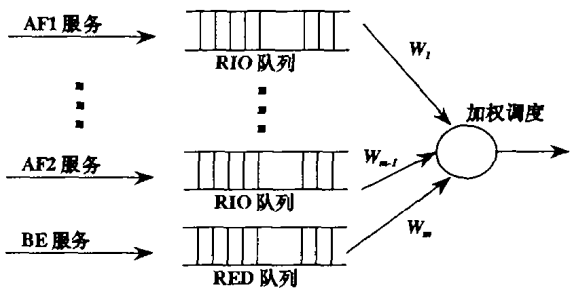


图 1 多服务调度系统模型示意图

各 AF 服务采用 RIO 队列管理算法;对普通 BE 服务采用 RED 队列管理算法;队列输出端采用权值调度算法,如 WRR、WFQ 等。

为简化讨论,我们假设 AF 和 BE 服务的总数即服务队列数是固定的(用 m 表示,$m\leqslant 5$),AF 和 BE 服务所共用的总带宽也是一定的(用 B 表示,单位包/秒)。为便于计算吞吐率,本文中假设同一个服务中通过的传输流都是 TCP 流。我们的前期工作[5]表明,区分服务网络中的 UDP 流量在 TCP 友好的速率控制(TFRC)机制下可以体现类似 TCP 的流量特征,因此本文的研究结果也可以扩展到 UDP 流量情况下。由于 AF 和 BE 服务在竞争剩余带宽时享有同样的优先权,我们把 BE 服务视为订购带宽为零的特殊 AF 服务,因此可以用一组 AF 服务 i ($i=1,\cdots,m$)来表示由多个级别的 AF 和

*)基金资助:国家自然科学基金(6202005),国家教育部骨干教师基金。刘 威 博士研究生,主要研究领域为网络服务质量、区分服务。杨宗凯 教授,博士生导师,主要研究领域为高速通信网络及其应用。乐春晖 博士研究生,主要研究领域为网络服务质量。何建华 副教授,主要研究领域为网络服务质量和多媒体。

BE 服务构成的多服务系统，其在输出链路上对应的调度权值为 w_i。

2.2 多种服务队列公平性的准则

本文采用带宽平均分配的公平性准则，即除了订购带宽之外的剩余带宽按流的数目平均分配给所有可以分享带宽的 AF 和 BE 服务中的传输流。对第 i 个服务而言，S_i 为其订购带宽，B_i 和 B_i' 分别为其实得带宽和应得带宽，N_i 表示该服务中流的数目。则多服务间带宽分配的公平性准则可以表示为：

$$\frac{B_i'-S_i}{B_j'-S_j}=\frac{N_i}{N_j} \tag{1}$$

由式(1)，进一步可得：

$$\frac{B_i'-S_i}{B-\sum_{j=1}^{m}S_j}=\frac{N_i}{\sum_{j=1}^{m}N_j} \tag{2}$$

由式(2)，可得第 i 个服务应得带宽为：

$$B_i'=S_i+\frac{N_i}{\sum_{j=1}^{m}N_j}\cdot(B-\sum_{j=1}^{m}S_j) \tag{3}$$

为了由式(3)计算 B_i'，我们还需要每个服务 j ($j=1,2,\cdots,m$)中的订购带宽 S_j 和流数目 N_j。考虑到在实际网络节点中直接测量这些指标需要大量缓存空间的开销，本文将通过对本地队列缓存区中的一些容易测量的指标间接计算获得 S_j 和 N_j。

2.3 服务队列的订购带宽的估计

在本文的研究场景下，忽略队列处理的时延，通过缓存区队列的速率近似为骨干链路的速率。考虑到 TCP 弹性流量的特点，在大多数情况下其在缓存区中的队列长度不会为零，则可得比例关系如下：

$$\frac{S_i}{B_i}\approx\frac{Q_{i,avg}^{in}}{Q_{i,avg}} \tag{4}$$

其中 $Q_{i,avg}^{in}$ 和 $Q_{i,avg}$ 分别是 IN 队列和总队列的平均长度。以此可以获得对订购带宽的估计值如下：

$$S_i\approx\frac{Q_{i,avg}^{in}}{Q_{i,avg}}\cdot B_i \tag{5}$$

2.4 服务队列中流数目的估计

在本文的研究场景下，网络负载处于未饱和订购，仅仅 OUT 分组有丢失，区分服务节点中第 i 个服务中聚合的 TCP 流与其吞吐量的关系[6]如下：

$$B_i=\frac{3}{4}S_i+\frac{3k}{4}\cdot\sum_{r=1}^{N_i}\frac{1}{RTT_{i,r}}\sqrt{\frac{2}{p_{i,r}^{out}}} \tag{6}$$

其中 k 表示平均 TCP 包长度；$RTT_{i,r}$ 和 $p_{i,r}^{out}$ 分别表示第 i 个服务中第 r 个 TCP 流的平均往返时间和 OUT 分组的丢失率。为简化起见，我们假设第 i 个服务的聚合流中的各 TCP 流都有相同的平均往返时间 RTT_i 和 OUT 分组的丢失率 p_i^{out}，则(6)式可以写为：

$$B_i=\frac{3}{4}S_i+\frac{3kN_i}{4RTT_i}\cdot\sqrt{\frac{2}{p_i^{out}}} \tag{7}$$

而 OUT 分组的丢失率与 RIO 队列的总分组丢失率之间满足以下关系：

$$p_i=p_i^{in}p_m+p_i^{out}(1-p_m) \tag{8}$$

其中 p_i 和 p_i^{in} 分别表示第 i 个服务队列中的总分组丢失率和 IN 分组的丢失率，在本文未饱和订购的场景下 $p_i^{in}=0$；p_m 表示该服务在边缘节点的标记概率，理想情况满足 $p_m=S_i/B_i$ 的关系，故(8)式可以化为：

$$p_i=p_i^{out}(1-S_i/B_i) \tag{9}$$

另外一方面，RTT_i 由链路传输时延(T_i)和队列时延共同构成：

$$RTT_i=T_i+\frac{Q_{i,avg}}{B_i} \tag{10}$$

综合式(7)(9)(10)，可得区分服务节点中第 i 个服务中的流数目 N_i 由 $Q_{i,avg}$ 和 S_i、B_i 等实测量的表达：

$$N_i=\frac{\sqrt{p_i}(B_i\cdot T+Q_{i,avg})(4-3S_i/B_i)}{3k\sqrt{2(1-S_i/B_i)}}$$

$$\approx 2\sqrt{2p_i(1-S_i/B_i)}(B_i\cdot T+Q_{i,avg})/3k \tag{11}$$

在我们的前期工作[4]中，基于对多个 TCP 流在 RED 队列中的吞吐量公式获得了一个较为简单的计算公式。本文基于区分服务中 TCP 聚合流在 RIO 队列中的吞吐量公式进行推导，比之更为精确。

2.5 理想调度权值的计算

考察权值调度中当前权值和达到公平时理想权值之间的关系，设 w_i 和 w_i' 分别表示第 i 个服务的当前调度权值和理想调度权值。则由式(3)，可得：

$$w_i'=\frac{B_i'}{B}=\frac{S_i}{B}+\frac{N_i}{\sum_{j=1}^{m}N_j}\cdot(1-\sum_{j=1}^{m}(\frac{S_j}{B})) \tag{12}$$

其中第 $j(j=1,2,\cdots,m)$ 个服务的 N_j 和 S_j 由式(11)和(5)给出。又因为在理想的权值调度中，各服务在输出链路得到的带宽与其调度权值成比例，即：

$$B_i=w_i\cdot B \tag{13}$$

把式(5)(11)(13)代入式(12)，并假设链路的总带宽 B 和各 TCP 流的往返传播时延 T 已知，可得计算理想调度权值 w_i' 的函数 $f(\cdot)$ 如下：

$$w_i'\approx f(Q_{i,avg}^{in},Q_{i,avg},p_{i,avg},\{w_j\})\approx w_i\cdot Q_{i,avg}^{in}/Q_{i,avg}+$$

$$\frac{\sqrt{p_i(1-Q_{i,avg}^{in}/Q_{i,avg})}(B_i\cdot T+Q_{i,avg})}{\sum_{j=1}^{m}\sqrt{p_j(1-Q_{j,avg}^{in}/Q_{j,avg})}(B_j\cdot T+Q_{j,avg})}\cdot$$

$$(1-\sum_{j=1}^{m}(w_j\cdot Q_{j,avg}^{in}/Q_{j,avg})) \tag{14}$$

由 $f(\cdot)$，我们可以设计一个周期性的对调度权值进行动态调整的方案：在每一个周期结束时，基于当前调度权值 $\{w_i\}$ 计算下一个周期内的理想调度权值 $\{w_i'\}$，下一个周期以新的调度权值进行调度。

2.6 动态调整调度权值的算法

周期性的动态权值调度算法如下：

1) 在每一个动态调整周期 τ 的开始时刻，重置所有的变量；

2) 当第 i 个服务的包丢弃时，用于计算丢包概率的丢包计数器 D 加 1；

3) 按采样时间 $\tau_i(\tau_i<\tau)$ 记录当前各服务 $j(j=1,\cdots,m)$ 的总队列长度 $\{Q_{j,k}\}$ $(k=1,\cdots,\tau/\tau_i)$ 以及 IN 包队列各自的队列长度 $\{Q_{j,k}^{in}\}$；

4) 在每一个动态调整周期的结束时刻，计算该周期内各服务队列的丢包概率 p_j 和平均总队列长度 $Q_{i,avg}=\tau_i/\tau\sum(Q_j,k)$ 以及平均 IN 包队列长度 $Q_{i,avg}^{in}=\tau_i/\tau\sum(Q_{j,k}^{in})$；

5) 权值计算：由式(5)和(11)，计算各服务 j 中的订购量 S_j 和流数目 N_j，然后根据式(14)计算理想的归一化调度权值 $\{w_i'\}$；

6) 加权调度：以权值 $\{w_i'\}$ 进行调度，重新分配各服务占有的输出链路带宽。

显然，该动态调整权值算法与具体权值调度算法的类型无关，因此具有高度的可扩展性。在实现过程中，缓存区队列的长度和队列门限值必须设得足够大。这是因为对流数目的估计是基于平均队列长度的，考虑到当前队列大小的抖动会使计算得到的平均值有一定的误差，过小的缓存区设置会使这种误差相对放大，从而严重影响计算结果。建议缓存区队列长度近似等于带宽时延之乘积，同时 BE 或者 AF 中 OUT 包

队列的最大队列门限值维持在 0.7 倍的缓存区队列长度左右。

3 实验仿真与评估

我们在 ns-2[7]环境中实现了动态权值调度方案。仿真网络拓扑模型如图 2 所示，区分服务域由两个边缘节点(*E*0，*E*1)和一个核心节点(*C*0)构成。从源节点(*S*0，*S*1)到目的节点(*D*0)分别建立了 TCP 的连接并发送 FTP 数据流。该区分服务域的骨干链路的带宽为 50Mbps；各接入链路的带宽为 100Mbps；各链路的传播时延为 10ms，TCP 连接的 RTT 为 80ms。

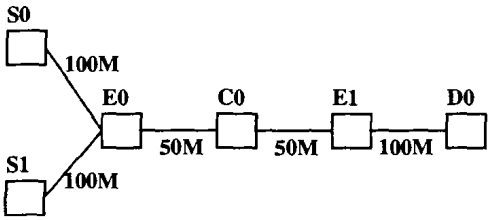


图 2 实验仿真网络拓扑模型示意图

所有从节点 *S*0 到节点 *D*0 的 TCP 流都采用 AF 服务，而从节点 *S*1 到节点 *D* 的 TCP 流都采用 BE 服务。处理 AF 服务的 RIO 算法的参数设置(以"th_{min}，th_{max}，p_{max}"的格式)为 IN 包队列(50，150，0.1)，OUT 包队列(70，200，0.02)；处理 BE 服务的 *RED* 算法的参数设置为(70，200，0.02)。边缘节点 *S*0 和 *S*1 上采用令牌桶的标记算法；核心节点 *C*0 上配合 WRR 调度算法实现动态权值调整方案(称为动态 WRR)，动态调整权值的周期为 10s。实验中计算剩余带宽公平性时采用以下公平性指数定义，其中 e_i 是流 i 获得的剩余带宽，显然该公平指数越接近 1 则越公平：

$$Fairness\ Index=\frac{(\Sigma e_i)^2}{n\cdot\Sigma(e_i^2)} \tag{15}$$

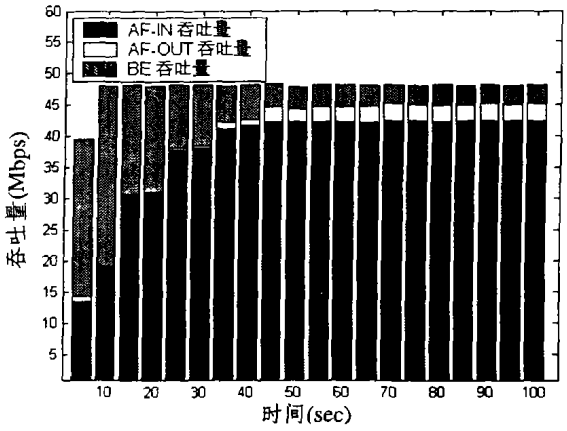
3.1 链路订购比例的影响

由式(3)可知，各服务队列的应得带宽与其订购量和网络负载(即流数目)关系密切。考虑到现实网络配置中，链路的订购带宽可能会重新配置，本组实验考察动态权值调度算法对这种情况的适应能力。我们在实验中保持网络负载不变，其中 AF 和 BE 服务的流数目分别设为 25 和 35，将 AF 和 BE 服务的初始调度权值设为(40:60)，用动态 WRR 算法运行实验运行多次实验，每次实验逐步改变 AF 服务的链路订购比例从 30%到 90%，记录 AF 服务中 IN 流量和 OUT 流量，以及 BE 服务获得的吞吐量，并且计算剩余带宽公平性，作出实验结果图 3。

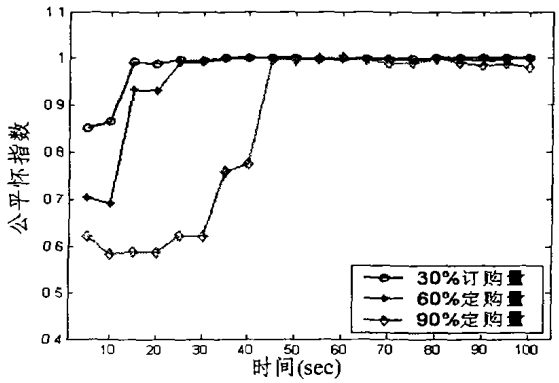
由于篇幅限制，在此仅给出链路订购比例为 90%时的各流量获得调度带宽柱状图(图 3(a))。观察图中结果可知，动态调度算法在 50 秒左右就将调度权值调整到接近理想值。在 t=5 秒时，链路输出带宽还是由初始权值(AF:BE=40:60)进行分配；经过 5 次动态权值调整，在 t=50 秒时，链路输出带宽比例已经接近(AF-IN:AF-OUT:BE=90:4:6)的理想值。观察图 3(b)可知，在大部分情况下，动态调度算法都可以快速地将系统的剩余带宽公平性提高到接近 1 的水平。

3.2 动态网络负载的影响

考虑到现实网络中 BE 流量的数目可能发生未知的变化，本组实验考察动态权值调度算法对这种情况的适应能力。我们在实验中保持链路的订购比例为 60%不变，AF 的流数目为 20，BE 的流数目按照图 4 中所示规律按 50 秒的周期发生变化，将 AF 和 BE 服务的初始调度权值设为(50:50)，分别用传统 WRR 调度算法和动态 WRR 运行实验，记录 AF 服务中 IN 流量和 OUT 流量、以及 BE 服务获得的吞吐量，并且计算剩余带宽公平性，作出实验结果图 5。



(a) 带宽分配情况：动态 WRR，订购量 90%



(b) 公平性指数计算结果

图 3 动态带宽调度克服链路订购比例的影响

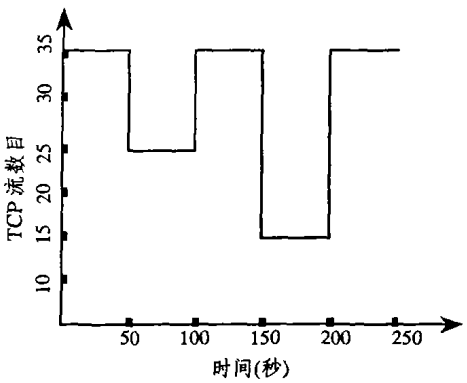
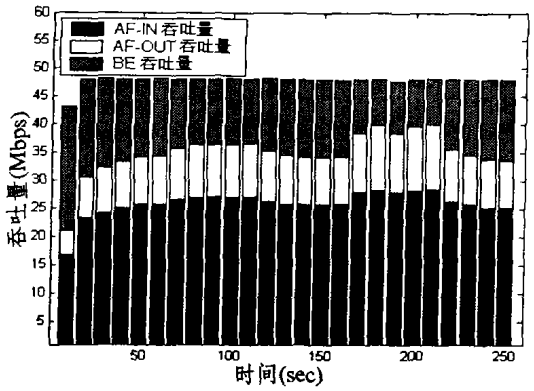


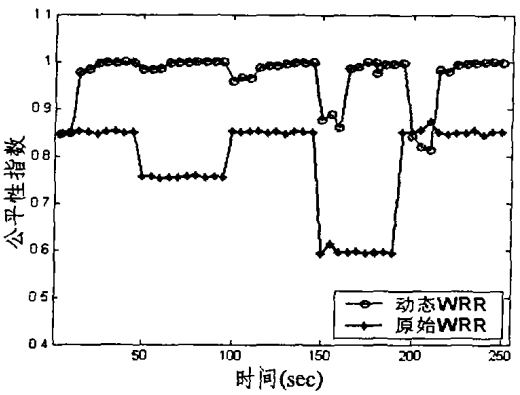
图 4 实验中的动态网络负载

观察图 5(a)可知，动态权值调度算法可以依据负载的动态变化调整各服务队列的输出，比如，在 t=120 秒时，BE 流数目为 35，动态调度的链路输出带宽比例接近于(AF-IN：AF-OUT：BE = 60：15：25)的理想值；在 t=180 秒时，BE 流数目为 15，动态调度的链路输出带宽比例接近于(AF-IN：AF-OUT：BE = 60：23：17)的理想值。由图 5(b)可知，动态权值调度算法在经历短暂的调整之后，可以快速地响应 t=150 秒和 t=200 秒时剧烈的网络负载变化，维持接近于 1 的系统剩余带宽公平指数。

由于本文采用了较为精确的 TCP 聚合流量公式进行推导，所得到的带宽公平指数比文[4]中的实验结果相比平均提高了 0.1 左右的公平指数。



(a) 带宽分配情况：动态 WRR，订购量 60%



(b) 公平性指数计算结果

图 5 动态带宽调度克服动态负载的影响

结论 在本文中，我们针对网络负载动态变化时多级别服务网络中带宽分配公平性的问题，提出了一种周期性的动态权值调度算法。该动态算法与具体的权值调度算法无关，计算过程依赖本地缓存区队列的测量值，计算开销小。实验仿真结果表明该算法不但可以快速地将调度权值调整到实现带宽公平分配的理想值，而且可以对输入流量负载(流数目)的变化作出有效的响应，与普通权值调度算法相比，大幅提高了多服务队列间的带宽分配的公平性。

参 考 文 献

1 Ji L, Arvanitis T N, Wolley S I. Fair weighted round robin scheduling scheme for DiffServ networks. Electronics Letters, 2003,39(3)

2 Kawahara R, Komatsu N. Dynamically Weighted Queueing for Fair Bandwidth Allocation and its Performance Analysis. In: Proc. of IEEE ICC'02, Jun. 2002

3 Shimonishi H, Maki I, Murase T, Murata M. Dynamic Fair Bandwidth Allocation for DiffServ Classes. In: Proc. of IEEE ICC'02, Jun. 2002

4 程文青，刘威，乐春晖，Chou C T. 区分服务网络中自适应加权调度方案的研究. 通信学报，已录用

5 Le C, He J, Liu W, Yang Z. Performance of TFRC for multimedia services with bandwidth guarantee. In: Proc. IEEE TENCON'02, Oct. 2002

6 Yeom I, Reddy ALN. Modeling TCP behavior in a Differentiated Services network. IEEE/ACM Trans. on Networking, 2001, 9 (1): 31～46

7 VINT Project. Network Simulator version 2 (ns-2). http://www.isi.edu/nsnam/ns

(上接第 20 页)

综合上述分析，在分布式并行计算环境下，Krylov 子空间迭代法的并行化策略可归结如下：

(1) 重排迭代法中的运算，使其充分使用局存中的数据，以提高 Cache 的利用率；

(2) 将系数矩阵 A 按行分配到各处理机上，以尽量达到负载平衡；

(3) 分析数据的依赖关系，使数据分布尽量做到只在邻居间通信，避免全局通信；

(4) 通过数学变换，将二个分离的内积用可并行执行的三个连续内积代替，以减少全局通信的次数；

(5) 将校正延迟一个迭代步，使解的校正不必等待内积计算的完成，以达到通信与计算的重叠，从而提高计算效率。

结论 大型稀疏线性方程组经常出现在科学和工程计算中，因此寻找稀疏线性方程组的高效计算方法及其并行算法是提高科学与工程应用问题计算效率的有效途径。本文介绍了 Krylov 子空间方法的概念及其分类，在此基础上，研究了分布式并行计算环境下 Krylov 子空间方法的并行算法，给出了 Krylov 子空间方法的几种并行化策略。

参 考 文 献

1 莫则尧. 大型科学与工程应用程序并行化关键技术及其应用研究：[博士后研究工作报告]. 北京应用物理与计算数学研究所计算物理实验室，1999

2 迟利华. 大型稀疏线性方程组的并行计算：[博士论文]. 湖南长沙国防科技大学，1998

3 谷同祥. 大型稀疏线性代数方程组的并行非定常迭代方法：[博士论文]. 北京应用物理与计算数学研究所计算物理实验室，2001

4 Hestenes M R, Stiefel E. Method of conjugate gradients for solving linear systems. J. Res. Nat. Bur. Stand., 1952,49:409～436

5 Concus P, Golub G H, O'Leary D P. A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations. In: J. R. Bunch and D. J. Rose, eds. Sparse Matrix Computions, Academic Press, 1976. 309～332

6 Saad Y. Krylov Subspace methods for solving large unsymmetric linear systems. Math. Computing, 1981

7 Jea K C, Young D M. Generalized conjugate gradient acceleration of nonsymmetrzable iterative methods. Linear Algera Appl., 1980

8 Saad Y. GMRES: A Generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput, 1986

9 Saad Y, Wu K. dqgmres: A direct quasi-minimal residual algorithm based on incomplete orthogonalization: [Technical Report UMSI-93/131]. Minnesota Supercomputing Institute, Minneapolis, MN, 1993

10 Saad Y. Analysis of augmented krylov subspace techniques: [Technical Report UMSI-95/175]. Minnesota Supercomputing Institute, 1995

11 Eisenstat S C. A note on the Generalized conjugate gradient method. SIAM J. Number Anal., 1983

12 Sturler E. Nested krylov methods based on GCR. J. of computional and applied mathematics, 1996

13 Lanczos C. Solution of systems of linear equations by minimized iteration. J. Res. Nat. Bur. Stand., 1952,49:33～53

14 Sonneveld P. CGS: a fast lanczos-type solver for nonsymmetric linear systems. SIAM J. Sci. Stat. Comput, 1989,10:36～52

15 Fokkema D R, Sleijpen P, Van der vorst H A. Generalized conjugate gradient squared. J. Comput. Appl. Math., 1996,71:125～146

16 Van der vorst H A. Bi-CGSTAB: Afast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. SIAM J. Sci. Stat. Comput, 1992,12:631～644

17 Freund R W, Nachtigal N M. QMR: A quasi-minimal residual method for non-hermitian linear systems, Numer. Math., 1991, 60:315～339

18 Schonauer W. Scientific computing on vector computers. North-Holland, NY, 1987

19 Freund R W. A transpose-free quasi-minimal residual algorithm for non-hermitian linear systems. SIAM J. Sci. Comput., 1993, 14:470～482

20 Duff I S, Meurant G A. The effect of ordering on precondioned conjugate gradient. BIT, 1989,29:635～657

21 Meurant G. The block preconditioned conjugate gradient method on vector computers. BIT, 1984,24:623～633

22 Basermann A. Conjugate gradient and lanczos methods for sparse matrices on distributed memory multiprocessors. J. Parallel Distributed Computing, 1997,45:46～52

23 de Sturler E. Aperformance model for krylov subspace methods on mesh-based parallel computers. Parallel Computing, 1996,22: 57～74