ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.19, No.2, February 2008, pp.314−322
DOI: 10.3724/SP.J.1001.2008.00314

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种有效的数据共享环境多数据源选择算法*

汪晓庆[1,2+], 郑彦兴[2], 史美林[1]

[1](清华大学 计算机科学与技术系,北京 100084)

[2](北京系统工程研究所,北京 100101)

An Efficient Multiple Data Sources Selection Algorithm in Data-Sharing Environments

WANG Xiao-Qing[1,2+], ZHENG Yan-Xing[2], SHI Mei-Lin[1]

[1](Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

[2](Beijing Institute of System Engineering, Beijing 100101, China)

+ Corresponding author: Phn: +86-10-66356588, E-mail: wang_xiaoqing@126.com

Wang XQ, Zheng YX, Shi ML. An efficient multiple data sources selection algorithm in data-sharing environments. *Journal of Software*, 2008,19(2):314−322. http://www.jos.org.cn/1000-9825/19/314.htm

Abstract: The problem of multiple data sources selection (MDSS) in DSE (data-sharing environments) is addressed and the algorithm MDSSA (MDSS algorithm) is presented. MDSSA introduces the concept of Pareto optimization which reduces the search space greatly. By means of a novel normal-measure based nonlinear cost function, MDSSA computes approximate Pareto optimal paths to each data source first, and then gives the optimal data source and its corresponding path by comparing the cost of all candidate paths, resulting in finding more effective paths and much shorter response time. Extensive simulations show the efficiency of the algorithm.
Key words: data-sharing environment; multiple data source; quality of service; multiple objective optimization

摘 要: 针对数据共享环境多数据源选择 MDSS(multiple data sources selection)问题,基于 Pareto 最优理论提出了 MDSSA(MDSS algorithm)算法.该算法借助崭新的基于法线测量的非线性路径代价方程计算出到每个数据源的最优路径集合,进而通过代价对比确定实施数据访问的最佳数据源及路径,极大地缩小了搜索空间,在搜索到有效路径的同时,确保了算法的响应时间.大量仿真实验表明,MDSSA 算法是有效的.
关键词: 数据共享环境;多数据源;服务质量;多目标优化
中图法分类号: TP311 **文献标识码**: A

随着全球信息化步伐的不断迈进,信息(数据)在人类社会中所起的作用越来越重要[1].通过构建数据共享环境,在异构的各业务单元之间根据发展中的业务需求实现数据的协同共享、有效管理以及无缝的按需集成,是企业提高生存能力的关键.

在数据共享环境中,数据和网络的冗余是实现有效数据共享、确保数据安全的必要手段[2].数据和网络的冗余在一定程度上缓解了因网络节点单点失效而造成的数据损失和任务失败,但数据使用者却面临如何选择最佳的数据源及路径,以便能够以较小的代价迅速地获得数据的问题.由于在以网络为中心的数据共享环境中,提

* Received 2006-05-22; Accepted 2006-10-10

供冗余数据的多个数据源位于不同的网络节点之上,而且网络往往也是冗余的,即可以有多条路径访问某一特定的数据源网络节点,因此,数据共享环境中面向数据和网络冗余的多数据源选择本质上是涉及网络节点与节点之间时延和通信代价等 QoS 度量的一个多约束路径选择问题.

本文针对数据共享环境多数据源选择 MDSS(multiple data sources selection)问题进行了深入的研究,并基于 Pareto 最优理论提出了 MDSSA(MDSS algorithm)算法.MDSSA 算法的主要任务是首先确定多约束条件下到达每个候选数据源的最优路径,然后通过代价对比最终确定实施数据访问的最佳数据源及路径.本文的主要创新是:(1) 首次实现了基于非线性路径代价的预计算搜索路径,在搜索到有效路径的同时,确保了算法的响应时间;(2) 通过定义非线性路径代价方程,为每个数据源搜索近似 Pareto 最优路径,极大地缩小了搜索空间;(3) 基于非线性路径代价方程的预计算保证了数据使用者能够迅速作出决策,为数据共享提供了必要的服务质量保证.

本文第 1 节给出问题模型及相关研究.第 2 节给出 Pareto 最优的相关概念.第 3 节给出 MDSSA 算法.第 4 节为算法性能评估.第 5 节为结论.

1 问题模型及相关研究

MDSS 问题可以分解为两个子问题:(1) 数据使用者到每个数据源的最优路径问题,称为同源最优路径问题(optimal paths of the same data source,简称 OPSDS);(2) 各数据源到数据使用者之间的路径比较问题,称为异源最优路径问题(optimal paths between various data sources,简称 OPVDS).

OPSDS 子问题本质上是离散的多目标优化问题,可以描述为:

定义 1(OPSDS 问题). 网络 $G(N,E)$,N 是节点集合,E 是边集.每条边对应一个多维度量向量 $\boldsymbol{w}(w_1,w_2,\ldots,w_k)$. w_i 是可加度量,i=1,…,k. P_{sd} 表示从数据使用者 s 到数据源 d 的路径集合,问题是寻找 $\boldsymbol{p}\in P_{sd}$,满足

$$w_i(\boldsymbol{p})=\sum_{e\in \boldsymbol{p}} w_i(e)\le w_i(\boldsymbol{q})=\sum_{e\in \boldsymbol{q}} w_i(e),\quad i=1,\ldots,k,\ \boldsymbol{q}\in P_{sd} \tag{1}$$

路径 $\boldsymbol{p}$ 又可以写成 $\boldsymbol{p}(w_1,w_2,\ldots,w_k)$的形式.满足所有 m 个约束的路径称为一个可行路径,w_i 可以表示为具体的时延或代价.可以看出,OPSDS 是典型的离散多目标优化问题.

解决离散多目标优化问题的自然想法是采用线性路径代价方程,即通过系数$(d_1,d_2,\ldots,d_k)$把每条边 e 上的度量组合成单一度量:$l(e)=\sum_{i=1}^{k} d_i w_i(e)$. 这样就可以基于该单一度量,调用 Dijkstra 算法寻找最短路径.通过调节组合系数可以实现不同的搜索,从而形成了多种基于线性路径代价的算法[3,4].基于 LPLF(linear path length function)算法的最大缺点是,有些最优路径永远也不能被找到.如图 1 所示.

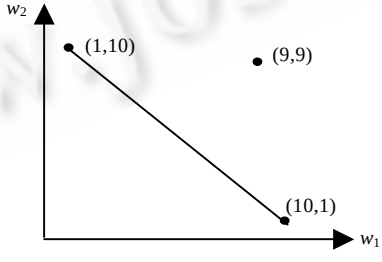

Fig.1 Shortcomings of linear path cost based algorithms

图 1 线性路径代价方程的算法的缺点

图 1 中的源和目的节点之间存在 3 条路径(1,10),(10,1)和(9,9),其中,路径(9,9)不可能被基于线性路径代价方程的算法找到.事实上,基于线性路径代价方程的算法找不到(1,10)和(10,1)两点连线上方的任何点.因此,研究者开始考虑非线性路径代价方程,并提出了一些基于非线性路径代价的算法[5,6].典型的非线性路径代价方程如下:

$$w(\boldsymbol{p})=\left(\sum_{i=1}^{k}\left(\frac{w_i(\boldsymbol{p})}{c_i}\right)^{q}\right)^{1/q} \tag{2}$$

基于非线性路径代价的算法虽然具有更好的搜索效果,但由于非线性路径代价方程需要人为地给出约束信息 c_i,不能用于预计算,因而严重影响了算法的响应时间.

定义 2(OPVDS 问题). 数据使用者 s 与数据源 d_i 之间存在路径 $\boldsymbol{p}_i$,i=1,…,k,寻找路径 $\boldsymbol{q}\in(\boldsymbol{p}_i|i=1,2,\ldots,k)$,满足

$$w_i(\boldsymbol{q})=\sum_{e\in\boldsymbol{q}}w_i(e)\le w_i(\boldsymbol{p}_j)=\sum_{e\in\boldsymbol{p}_j}w_i(e),\ i=1,\ldots,k.$$

2 多目标优化问题解的特性

Pareto 最优是多目标优化问题中的重要概念,解决 OPSDS 需要考虑的一个重要因素是解的 Pareto 最优性.

定义 3(支配(dominance)). 如果 $\forall i\in\{1,\ldots,k\},u_i\le v_i\wedge\exists i\in\{1,\ldots,k\}:u_i<v_i$,则称向量 $\boldsymbol{u}=(u_1,\ldots,u_k)$ 支配向量 $\boldsymbol{v}=(v_1,\ldots,v_k)$,用 $\boldsymbol{u}\prec\boldsymbol{v}$ 表示.

定义 4(Pareto 最优解). 如果不存在路径 $\boldsymbol{p}'(w_1',w_2',...,w_k')\in P_{sd}$ 满足 $(w_1',w_2',...,w_k')\prec(w_1,w_2,...,w_k)$,则称路径 $\boldsymbol{p}(w_1,w_2,\ldots,w_k)\in P_{sd}$ 为
OPSDS 问题的 Pareto 最优解.

定义 5(QoS 度量空间). 对于任意路径 $\boldsymbol{p}(w_1(\boldsymbol{p}),w_2(\boldsymbol{p}),\ldots,w_k(\boldsymbol{p}))\in G(N,E)$,如果 $w_i(\boldsymbol{p})\in W_i$,则称 $(W_1\times W_2\times\ldots\times W_k)$ 为 QoS 度量空间.

定义 6(映射 F). 映射 F 把路径 $\boldsymbol{p}(w_1(\boldsymbol{p}),w_2(\boldsymbol{p}),\ldots,w_k(\boldsymbol{p}))$映射为 QoS 度量空间中的一个点,即

$$F(\boldsymbol{p})=(f_1(\boldsymbol{p}),f_2(\boldsymbol{p}),\ldots,f_k(\boldsymbol{p}))=(w_1(\boldsymbol{p}),w_2(\boldsymbol{p}),\ldots,w_k(\boldsymbol{p})).$$

显然,路径 $\boldsymbol{p}(w_1(\boldsymbol{p}),w_2(\boldsymbol{p}),\ldots,w_k(\boldsymbol{p}))$对应 QoS 度量空间中的点$(w_1(\boldsymbol{p}),w_2(\boldsymbol{p}),\ldots,w_k(\boldsymbol{p}))$.

Pareto 最优解的意义在于,如果所有的 Pareto 最优解都不能满足要求,那么肯定不存在满足要求的其他任何解,因而搜索多目标优化问题的 Pareto 最优解将极大地缩小搜索空间,使得搜索更为有效.和所有的多目标优化问题一样,OPSDS 问题的解实际上是多个 Pareto 最优路径的集合,解决 OPSDS 问题的关键就是找到这个集合.同样,OPVDS 问题实际上是具有多个属性的候选路径之间的对比问题,该问题也可以利用 Pareto 最优理论解决,即对于多个候选路径,只需考虑其中的 Pareto 最优路径.

3 MDSSA算法

针对数据共享环境多数据源选择问题,我们提出了基于法线测量的算法 MDSSA.MDSSA 算法借助非线性路径代价方程,基于 Pareto 概念,成功实现了数据共享环境的多数据源选择.MDSSA 算法分为两大步骤:第 1 步,基于非线性路径代价,以预计算的方式为每个数据源生成候选最优路径集合;第 2 步,根据数据使用者的偏好信息对候选最优路径集合中各路径的代价进行对比,最终确定实施数据访问的最佳数据源及路径.下面首先给出算法描述,再根据算法描述详细解释算法步骤.

3.1 算法描述

算法 1 给出了 MDSSA 算法的主要步骤.MDSSA 算法首先通过 $GCPEDS(s,d_i)$函数为每个数据源 d_i 生成候选路径集合$\{\boldsymbol{p}_j(d_i)|j\ge0\}$.对于某个特定的数据源 d_m,它与数据使用者 s 之间的路径集合$\{\boldsymbol{p}_j(d_m)|j\ge0\}$中可能有多个 Pareto 最优路径.在数据使用者没有任何偏好的情况下,这些路径都是最优路径,即数据使用者可以利用该集合中的任意路径获得数据源 d_m的数据.在数据使用者有偏好的情况下,可根据偏好信息$(\alpha_1,\alpha_2,\ldots,\alpha_k)$决定选择哪条路径.偏好信息$(\alpha_1,\alpha_2,\ldots,\alpha_k)$反映了数据使用者更注重哪个服务质量度量,如时间、代价.利用偏好信息可以确定每个数据源路径的最小代价,进而通过比较各数据源的最小代价确定选择哪个数据源,见算法 1 中的步骤 2~步骤 5.不难看出,算法 1 中的步骤 2~步骤 5 是用来解决 OPVDS 问题的,而步骤 1 中的 GCPEDS 函数用来解决 OPSDS 问题,也是整个 MDSSA 算法的关键.

算法 1. MDSSA 算法.

s:源节点,代表数据使用者

d_i:数据源

(1) For i=1,i≤k

$\{\boldsymbol{p}_j(d_i)|j\geq 0\}=GCPEDS(s,d_i)$. //Generating candidate paths for each data source

为每个数据源 d_i 生成候选路径集合;

(2) 根据用户偏好信息$(\alpha_1,\alpha_2,\ldots,\alpha_m)$,

$\sum_{i=1,m}\alpha_i=1,\alpha_i\geq 0$, 计算候选路径代价

$cost(\boldsymbol{p}_j(d_i))=\prod_{i=1}^{m}\alpha_i\sum_{e\in \boldsymbol{p}_j(d_i)}w_i(e)$;

(3) 数据源 $D=d_1$;

(4) For i=1,i<k

(5) if min($cost(\boldsymbol{p}_j(d_i))$)≥min($cost(\boldsymbol{p}_n(d_{i+1}))$)

$D=d_{i+1}$ //j,n 分别为数据源 d_i 和 d_{i+1} 候选路径的数目

3.2 *GCPEDS*(s,d_i)函数

GCPEDS(s,d_i)函数用来生成每个数据源 d_i 与数据使用者之间的候选路径.*GCPEDS*(s,d_i)函数借助基于法线测量的非线性路径代价方程 NMCF(normal measure cost function),在不需要人为给出路径选择约束的条件下实现对路径代价的测量.因而 *GCPEDS*(s,d_i)函数首次实现了基于非线性路径代价的预计算,提高了算法的响应速度.NMCF 的提出主要受到了法线边界交叉(normal boundary intersection,简称 NBI)方法的启发[7].

3.2.1 法线边界交叉

NBI 方法是生成连续多目标优化问题 Pareto 层的手段之一.通过用户提供的参数$\boldsymbol{\beta}$,NBI 可以生成在 QoS 度量空间均匀分布的 Pareto 最优解.NBI 方法把连续多目标优化问题转化成如下子问题:

$$\text{Minimize } \lambda \tag{3}$$

$$\text{Subject to } \boldsymbol{\phi\beta}+\lambda\hat{n}=\boldsymbol{F}(x)-\boldsymbol{F}^* \tag{4}$$

在这个子问题中,$\boldsymbol{\phi}$是 $m\times m$ 矩阵,该矩阵的第 i 列为 $\boldsymbol{F}(x_i^*)-\boldsymbol{F}^*$.其中,$\boldsymbol{F}(x_i^*)$ 为使得第 i 个目标最优的解在目标空间中对应的向量.$\boldsymbol{F}^*$也是目标空间中的点,它的第 i 个分量对应第 i 个目标的最优值,所以又被称为乌托邦点.$\boldsymbol{\beta}$为一个向量,它满足 $\sum_{i=1}^{k}\beta_i=1$ 及$\beta_i\geq 0$. $\hat{n}=\boldsymbol{\phi e}$,其中,$\boldsymbol{e}\in R^k$ 为元素全为 1 的列向量.$\boldsymbol{\phi\beta}$又被称为各极值点的凸壳(convex hull of individual minima,简称 CHIM).

对于多目标优化问题,用 $\hbar$ 表示可达目标向量$\{F(x)\}$的集合,$\hbar$ 的边界用 $\partial\hbar$ 表示.本质上,NBI 方法就是试图在 $\partial\hbar$ 上找到 Pareto 最优点.

对于一个特定的$\boldsymbol{\beta}$,$\boldsymbol{\phi\beta}$就表示 CHIM 上的一个点.而 $\boldsymbol{\phi\beta}+t\hat{n}$,$t\in\mathbf{R}$ 表示法线 $\hat{n}$ 上的点.问题(3)的解则是法线 $\hat{n}$ 与 $\partial\hbar$ 的交点中最接近原点的一个.问题(4)给出的约束说明了不同的法线 $\hat{n}$ 与 $\partial\hbar$ 有不同的交点.同时也从另一个方面说明,NBI 求得的解一定要在法线 $\hat{n}$ 上.问题(3)又称为 NBI 子问题.记作 $NBI_{\boldsymbol{\beta}}$,通过改变$\boldsymbol{\beta}$的值,就可以找到不同的近似 Pareto 最优解.

3.2.2 法线测量代价方程NMCF

正如前面所讨论的,OPSDS 问题是离散多目标优化问题.NBI 方法是针对连续多目标优化问题提出来的,所以不能用来解决 OPSDS 问题.其原因很简单,由于目标空间是离散的,对于特定的$\boldsymbol{\beta}$,法线 $\hat{n}$ 与 $\partial\hbar$ 可能根本不相交,因而不能实现通过法线对目标点进行测量.

针对 NBI 方法的这一重要缺陷,我们在文献[8]中设计了新的非线性路径代价方程 NMCF,使得在 $\partial\hbar$ 不连续的情况下,仍然可以通过法线度量 $\partial\hbar$ 上的点.因此,NMCF 可以用来生成离散多目标优化问题的近似 Pareto 最优解.本文在文献[8]中研究成果的基础上,首次设计了基于非线性路径代价的预计算算法,并将其应用于解决数据共享环境多数据源选择问题.

首先给出必要的定义及约定.令 $\boldsymbol{p}^{i*}$表示 P_{sd} 中,第 i 个 QoS 度量最优的路径,即 $\boldsymbol{p}^{i*}$满足 $f_i(\boldsymbol{p}^{i*})\le f_i(q),\boldsymbol{p}^{i*},q\in P_{sd}$, $1\le i\le m$.

称 $\boldsymbol{F}^*=[f_1(\boldsymbol{p}^{1*}),f_2(\boldsymbol{p}^{2*}),\ldots,f_k(\boldsymbol{p}^{k*})]^T=[f_1^*,f_2^*,\ldots,f_k^*]^T$ 为乌托邦点;而由 $\boldsymbol{F}(\boldsymbol{p}^{i*}),i=1,2,\ldots,m$ 组成的平面称为乌托邦超平面,记为 U.

同时,定义如下归一化矩阵:

$$\boldsymbol{L}=[l_1,l_2,\ldots,l_k]^T=F^N-F^* \tag{5}$$

其中,

$$F^N\triangleq[f_1^N,f_2^N,\ldots,f_k^N]^T,$$

$$f_i^N=\max[f_i(\boldsymbol{p}^{1*}),f_i(\boldsymbol{p}^{2*}),\ldots,f_i(\boldsymbol{p}^{k*})].$$

对于 QoS 度量空间中的任意点 $F(p)$,归一化后的点 $\bar{F}(\boldsymbol{p})$ 表示为

$$\bar{F}(\boldsymbol{p})=[\overline{f_1}(\boldsymbol{p}),\overline{f_2}(\boldsymbol{p}),\ldots,\overline{f_m}(\boldsymbol{p})]^T,$$

其中

$$\overline{f_i}(\boldsymbol{p})=\frac{f_i(\boldsymbol{p})-f_i^*}{l_i}.$$

根据上述定义,给出如下非线性路径代价方程:

$$len(p)=-\min(\lambda_1,\lambda_2,\ldots,\lambda_m)^T \tag{6}$$

满足

$$\text{s.t. } \bar{\phi}\boldsymbol{\beta}+N=\bar{F}(\boldsymbol{p}) \tag{7}$$

其中

$$N=(\lambda_1n_1,\lambda_2n_2,\ldots,\lambda_mn_m)^T,$$

$$\bar{F}(\boldsymbol{p})=[\overline{f_1}(\boldsymbol{p}),\overline{f_2}(\boldsymbol{p}),\ldots,\overline{f_m}(\boldsymbol{p})]^T.$$

法线 $\hat{n}=(n_1,n_2,\ldots,n_m)^T$ 与公式(4)中的法线具有相同的含义.如果用向量$(\gamma_1,\gamma_2,\ldots,\gamma_m)^T$ 表示 $\bar{\phi}\boldsymbol{\beta}$,,则公式(7)给出的约束可以重写为

$$\lambda_in_i=\overline{f_i}(\boldsymbol{p})-\gamma_i,\quad i=1,2,\ldots,m \tag{8}$$

不难看出,公式(7)给出的约束并没有限制 $\partial\hbar$ 上待测量的点(路径)一定要在法线 $\hat{n}$ 上.实际上,只有当 $\lambda_i=\lambda_j,1\le i,j\le m,i\ne j$ 时,这些路径才位于法线 $\hat{n}$ 上.通过改变β的值,可以得到不同的近似 Pareto 最优点,从而只利用路径在法线上的投影信息就可以实现对路径代价的测量.因而,NMCF 可以用于离散多目标空间.我们在文献[8]中还给出并证明了如下结论:

定理 1. 如果路径 $\boldsymbol{p}(w_1,w_2,\ldots,w_m)$的长度大于 a 约束 $c(c_1,c_2,\ldots,c_m)$的长度,则路径 p 至少有 1 个分量不能满足约束要求.

定理 2. 对于给定的两条路径 $\boldsymbol{p}(w_1,w_2,\ldots,w_m)$及 $\boldsymbol{q}(w_1',w_2',\ldots,w_m')$,如果 $\boldsymbol{F}(\boldsymbol{p})\prec\boldsymbol{F}(\boldsymbol{q})$,则路径 $\boldsymbol{p}$ 的长度不会超过路径 $\boldsymbol{q}$ 的长度.

定理 3. 对于相互 Pareto 最优的两条路径 $\boldsymbol{p},\boldsymbol{q},N$ 为通过 $\boldsymbol{p}$ 的超平面的法线,则在该法线测量下,路径 $\boldsymbol{p}$ 的代价小于路径 $\boldsymbol{q}$ 的代价.

定理 3 进一步说明了虽然路径 $\boldsymbol{p},\boldsymbol{q}$ 的路径代价之间的关系可能会因为法线的不同而不同,但对于任何路径都存在使自己代价最小的超平面法线,这正是与用于解决连续多目标优化问题的 NBI 方法一致之处.对于法线测量长度的直观含义及上述定理的详细证明,参见文献[8].

3.2.3 GCPEDS函数描述

法线测量方法为离散多目标优化问题求解提供了基本工具和重要手段.函数 GCPEDS 灵活地运用了法线测量方法,实现了不同粒度下的可行解搜索.算法 2 给出了函数 GCPEDS 的主要步骤.GCPEDS 首先计算每个 QoS 度量最优路径$\boldsymbol{p}^{i*}$,进而可以确定乌托邦点和超平面,见算法 2 中的步骤 1 和步骤 2.步骤 4 决定了对每个 QoS 度量的考察粒度,b 值越大,考察的粒度越细,搜索的方向就越多,算法的复杂性也就越大.对每组权重系数都可确定超平面上的一个点 $\bar{\phi}\boldsymbol{\beta}$,进而可由 $\bar{\phi}\boldsymbol{\beta}$ 确定超平面的一个法线.步骤 5 调用基于法线测量代价的 Dijkstra 算法进行非线性搜索.

算法 2. *GCPEDS*(s,d_i).

G:网络拓扑;*s*:数据使用者节点;*d*:数据源节点.

(1) For i=1,2,…,k 计算 $\boldsymbol{p}^{i*}$

(2) 确定乌托邦点

$$\boldsymbol{F}^*=[f_1^*, f_2^*,..., f_k^*]^T=[f_1(\boldsymbol{p}^{1*}),f_2(\boldsymbol{p}^{2*}),\ldots,f_k(\boldsymbol{p}^{k*})]^T$$

(3) 计算归一化向量

$$\boldsymbol{L}=(l_1,l_2,\ldots,l_k)^T$$

(4) 对每个 $\boldsymbol{\beta}\in\left\{(a_1/b,a_2/b,...,a_k/b)\,|\,\sum_{i=1}^{k}a_i/b=1,\ 0\le a_i\le b,a_i,b\in\mathbf{Z}\right\}$

(a) $(\gamma_1,\gamma_2,\ldots,\gamma_k)^T\triangleq\bar{\phi}\boldsymbol{\beta}$

(b) 计算经过$(\gamma_1,\gamma_2,\ldots,\gamma_k)^T$的超平面 *U* 的法线

$$\hat{n}=(n_1,n_2,...,n_k)$$

(5) 基于法线测量代价的 Dijkstra 算法

图 2 描述了 GCPEDS 函数的搜索过程,可以看出,通过改变$\boldsymbol{\beta}$的值,可以搜索到不同的路径.另外,通过改变算法 2 中 *b* 的大小,可以实现不同粒度的搜索,*b* 值越大,搜索的粒度越细.

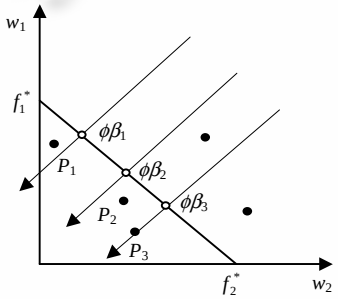

Fig.2 Search procedure of *GCPEDS*(s,d_i)

图 2 *GCPEDS*(s,d_i)的搜索过程

3.2.4 改进的MDSSA算法

MDSSA 算法假定数据使用者具有偏好信息,并根据偏好信息追求 QoS 度量的综合值.当数据使用者对所选路径的 QoS 度量具有量化要求时,如时延应不超过Δd,代价应不超过Δc,MDSSA 算法无能为力.因而,我们进一步改进了 MDSSA 算法.当数据使用者对所选路径的 QoS 度量具有量化要求时,算法在得到所有数据源的候选路径后,首先找出能够满足约束的路径集合,然后根据 Pareto 最优关系确定数据源.如果各数据源的所有候选路径都不能满足约束,改进的 MDSSA 算法可根据约束$\Delta d,\Delta c$ 进行有针对性的搜索.算法 3 描述了改进的 MDSSA 算法.

算法 3. 改进的 MDSSA.

s:源节点,代表数据使用者;d_i:数据源;c_i:QoS 约束.

(1) For i=1,$i\le k$

$\{\boldsymbol{p}_j(d_i)|j\ge0\}$=*GCPEDS*($s,d_i$). //Generating candidate paths for each data source

为每个数据源 d_i 生成候选路径集合;

(2) 如果数据使用者具有偏好信息,按原来算法流程继续执行,参见算法 1.否则:

I. 在候选路径中寻找能够满足约束的路径集合 $\boldsymbol{P}$,即 $P=\{\boldsymbol{p}|w_i(\boldsymbol{p})\le c_i,i=1,2,\ldots,k\}$

II. 在集合 $\boldsymbol{P}$ 中去除被支配的路径,形成新集合 $\boldsymbol{P}'$

III. 如果 $\boldsymbol{P}'$中具有满足$(c_1,c_2,\ldots,c_k)$的路径,则可以直接选取该路径对应的数据源,

否则,i=1:

i. 计算经过$(c_1,c_2,\ldots,c_k)$的超平面的法线 $\hat{n}$;

ii. 法线与超平面的交点为 $\bar{\phi}\boldsymbol{\beta}$,用向量$(\gamma_1,\gamma_2,\ldots,\gamma_k)^T$表示;

iii. $\boldsymbol{p}_i$=*NM_Dijkstra*(G,s,d_i);

iv. 如果 $\boldsymbol{p}_i$ 满足约束,则算法结束;

v. i=i+1;如果 i>k,则算法结束.

4 MDSSA算法性能评估

我们将从以下 3 个方面对 MDSSA 算法进行评估:首先,数据使用者只有偏好信息,没有对 QoS 度量的量化要求时,算法在特定偏好信息时的路径代价;其次,在有量化要求时,算法找到可用数据源的成功率;最后,评价算法在有量化要求时的响应速度.仿真实验采用了 Waxman 随机网络模型,网络节点数为 200 的网络,网络每条边上有两个服从[1,200]间均匀分布的可加 QoS 度量.数据使用者节点和数据源采用随机方式生成,并保证各数据源与数据使用者节点之间至少 2 跳距离,在无特别说明时,仿真过程中假定网络中共有 5 个数据源.

4.1 评价指标

为了充分说明算法的有效性,我们选取两个性能较好的经典算法 MEFPA[4]和 H_MCOP[5]与 MDSSA 算法进行比较.MEFPA 算法是基于线性路径代价的预计算算法,而 H_MCOP 算法则是基于线性路径代价的在线算法.虽然算法 MEFPA 和 H_MCOP 不是针对数据共享空间多数据源选择问题提出来的,但是可用于搜索单个节点对之间的最短路径,因而将两种算法在数据使用者和每个数据源之间应用就可以搜索相应的最优路径,从而与 MDSSA 算法进行比较.MDSSA 和 MEFPA 算法的实际计算代价与它们关注 QoS 度量的粒度相关,即算法 MDSSA 和 MEFPA 中的参数 b 影响算法的实际计算代价(参数 b 意味着算法在预计算阶段将执行 C_{b+k-2}^{k-1} 次 Dijkstra 算法).因此在仿真过程中,我们用 MDSSA(b)和 MEFPA(b)表示算法关注 QoS 度量的粒度为 b.

我们用偏好路径代价,即在给定偏好信息条件下,MDSSA 算法搜索到的数据使用者到各数据源的最小路径代价来评估 MDSSA 算法找到路径的有效性;用成功率,即被满足的约束数目与总的约束数目的比率来评估路径的可行性;用仅仅通过预计算就能够满足约束的数目与约束总数目的比率,即预计算成功率来评估算法的响应时间.在仿真实验过程中,上述性能指标都通过 1 000 次仿真实验获得.

4.2 偏好路径代价

我们首先评估数据使用者具有偏好信息时 MDSSA 算法的偏好路径代价.为了保持较小的计算代价,算法 MEFPA 的参数 b 取 7,MDSSA 算法的参数分别取 3,5 和 7.由于绝对的路径代价没有实际意义,我们用 MDSSA(7)的路径代价归一化其他算法的最小路径代价.图 3 显示了仿真结果.图中的纵坐标为归一化的偏好路径代价(normalized path cost,简称 NPC),横坐标为偏好信息(prefer information,简称 PI),PI_i 表示偏好信息,仿真过程中共使用了 6 种不同的偏好(1.2–0.2i,0.2i–0.2),i=1,2,...,6.从图中不难看出,与算法 MEFPA 和 H_MCOP 相比,MDSSA 算法具有较小的偏好路径代价,尤其是 MDSSA(7)具有最小的路径代价.

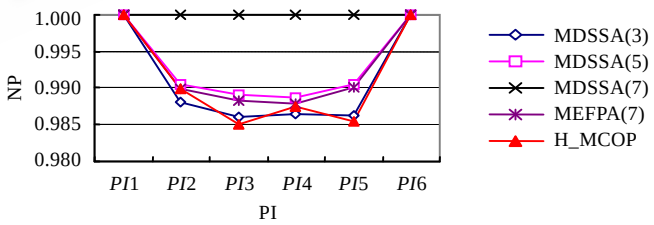

Fig.3 Preferred path cost evaluation of MDSSA

图 3 MDSSA 的偏好路径代价评估

4.3 算法成功率

算法成功率用来评估数据使用者在有 QoS 度量具体要求时,算法找到的路径能够满足要求的成功率.仿真过程中采用了文献[5]中生成约束的方式来模拟数据使用者的 QoS 度量要求.图 4 给出了仿真结果,其中纵坐标表示算法成功率(success rate,简称 SR),横坐标表示数据源的数目(number of data sources,简称 NDS),仿真过程中数据源的数目分别设为 5,10,15.由图 4 可知,与另两种算法相比,在不同数据源数目情况下,MDSSA 算法具有较高的成功率.当 MDSSA 算法的参数 b 为 7 时,具有最高的找到满足要求路径的成功率.

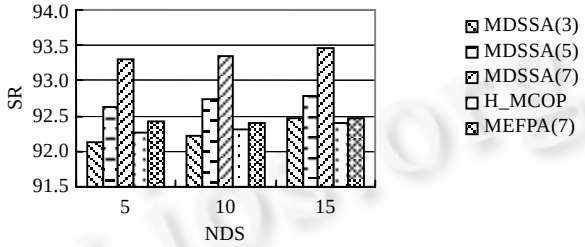

Fig.4 Success rate evaluation of MDSSA

图 4 MDSSA 成功率评估

4.4 算法响应时间

当数据使用者对 QoS 度量具有偏好信息时,MDSSA 算法仅仅通过预计算获得的候选路径即可快速作出数据源选择反应.而当数据使用者有 QoS 度量具体要求时,如果预计算获得的候选路径不能满足要求,MDSSA 算法将针对每个数据源进一步执行在线计算.在线计算在提高算法成功率的同时,降低了算法的响应时间.因而,我们以通过预计算就能够满足约束的数目与约束总数目的比率来评估 MDSSA 算法的响应时间.不难得知,仿真结果与 QoS 度量要求的模拟方式有关.QoS 度量要求越严格,MDSSA 算法执行在线计算就越多,算法的响应也就速度越慢;反之,QoS 度量要求越宽松,MDSSA 算法执行在线计算就越少,算法的响应速度也就越快.我们仍采用文献[5]中的约束生成方式来生成 QoS 度量.图 5 给出了仿真结果,其中纵坐标为算法预计算成功率(precomputation success rate,简称 PSR),横坐标为数据源数目 NDS.从图中不难看出,有超过 92%的 QoS 度量约束能够被预计算阶段获得的路径满足.并且当 NDS 增大时,算法预计算成功率 PSR 也相应增大.从图 5 不难看出,即使 MDSSA 算法需要在线计算,运行 Dijkstra 算法的次数也最多不会超过 NDS 次,因而 MDSSA 算法的响应速度是非常快的.

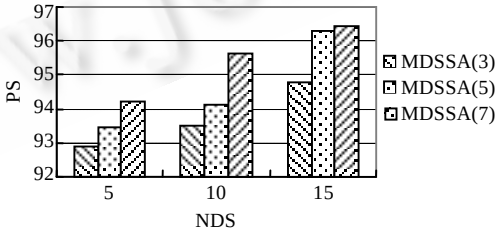

Fig.5 Response speed evaluation of MDSSA

图 5 MDSSA 的响应速度评估

5 结束语

当数据共享环境中有多个数据源能够同时向数据需求者提供数据服务时,如何选择合适的数据源及路径,使得数据需求者能够以较小的代价迅速地获得数据,是数据共享服务质量保证研究所必须面对的挑战.本文首先将数据共享环境多数据源选择问题 MDSS 分解为两个子问题,即同源最优路径问题 OPSDS 和异源最优路径

问题 OPVDS.基于 Pareto 最优理论,分析了与 MDSS 问题相关算法的优、缺点,在此基础上提出了 MDSSA 算法.MDSSA 算法基于法线测量的非线性路径代价方程进行路径搜索,预计算出到每个数据源的候选路径.决策制定者即可根据实际情况利用偏好信息在候选路径中选择合适的路径,进而确定数据源及路径.同时,MDSSA 算法也可以通过数据使用者提出的具体 QoS 度量要求来搜索数据源及路径.基于法线测量获得的路径是近似 Pareto 最优的,极大地缩小了搜索空间.大量仿真实验表明,与相关算法相比,MDSSA 算法在搜索到较小代价路径的同时,具有较快的响应速度.

我们下一步的工作是研究基于非线性路径代价方程的分布式搜索,克服集中式搜索时每个节点都要保存当前的网络链路状态信息的缺点.另外,我们将研究当数据共享环境中各数据源的数据本身在完整性、正确程度、精确程度以及与已有数据一致性等方面存在差异时,如何选择数据源的问题.

References:

[1] Perry W, Signori D, Boon J. Exploring Information Superiority: A Methodology for Measuring the Quality of Information and Its Impact on Shared Awareness. RAND, 2004.

[2] Vego MN. Operational command and control in the information age. Joint Forces Quarterly, 2005,(35):100−107.

[3] Jüttner A, Szviatovszki B, Mecs I, Rajko Z. Lagrange relaxation based method for the QoS routing problem. In: Proc. of the INFOCOM 2001, Vol.2. 2001. 859−868.

[4] Cui Y, Xu K, Wu JP. Precomputation for multi-constrained QoS routing in high-speed networks. In: Proc. of the INFOCOM 2003 Conf. San Francisco, 2003. 1414−1424.

[5] Korkmaz T, Krunz M. Multi-Constrained optimal path selection. In: Proc. of the INFOCOM 2001. 2001. 834−843.

[6] van Mieghem P, Kuipers FA. Concepts of exact QoS routing algorithms. IEEE/ACM Trans. on Networking, 2004,12(5):851−864.

[7] Das I, Dennis JE. Normal-Boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. SIAM Journal on Optimization, 1998,8(3):631−657.

[8] Zheng YX, Korkmaz T, Dou WH, Tian J. Highly responsive and efficient QoS routing using pre-and on-demand computations along with a new normal measure. Computer Networks, 2006,50(18):3743−3762.

汪晓庆(1971－),男,浙江衢州人,博士生,副教授,主要研究领域为数据共享,软件测试.

郑彦兴(1977－),男,博士,助理研究员,主要研究领域为数据共享,软件测试,多目标优化,网络协议.

史美林(1938－),男,教授,博士生导师,主要研究领域为计算机网络,计算机支持的协同工作,分布式系统.