Quality of Service Guarantee in Data Sharing Environment

Xiaoqing Wang,[1,2] *Yanxing Zheng*[2]*, Meilin Shi*[1]

[1] Tsinghua University, Beijing, China; [2]BeiJing Institute of System Engineering, China

Summary
A large variety of Web-based applications demand access and integration of up-to-date information from multiple distributed and heterogeneous information systems. Data-Sharing Environments(DSE) contribute to the achievement of information superiority which will enable decision dominance. The goal of DSE is to help data interaction between distributed and heterogeneous information sources. One challenge in Data Sharing is how to select the right data at the right time and right cost. We call this problem as quality of service(QoS) guarantee in DSE. In this paper, we model the problem and present an efficient mechanism to deal with QoS guarantee in DSE.

Key words:
Data-Sharing Environments, Information interaction, Quality of Service

1. Introduction

A wide variety of information sources are available both in internal networks of organizations and on the Web. Information today resides on a variety of information sources that are increasingly interconnected. File systems, databases, document retrieval systems, workflow systems, ERP (enterprise resource planning) systems, data warehouses, and other sources of valuable information are accessible inside corporate intranets. Moreover, they are also becoming increasingly available to the “outside world,” through extranets or the Internet.

Information sharing succeeds when the right information is provided to the right people at the right time and place so that they can make the right decisions. People make better decisions when they have the right information – data which they understand and which is relevant to the decision at hand. These “five rights” describe the purpose of DSE. Specifically, the “five rights” sets three goals for the activities in DSE: to ensure that the right data exists, is accessible, and is understood and discoverable. The first and most important step is to ensure that data is accessible; that is, made available by those who have it and deliverable to those who need it. The next step is to make the right data discoverable and understandable. Individuals and organizations must be able to obtain all the data they need, but to avoid the problem of data overload, it must be possible for them to receive only the data they need. Finally, the enterprise must take steps to ensure that the right data will exist. The enterprise must develop an understanding of current and anticipated information needs to drive the development and operation of its data resources, so that the data needed by a decider will be collected and made available somewhere in the enterprise.

Exploring information superiority plays a more and more important role in the practices of enterprises[1]. The key to information superiority is constructing an integrative data sharing environments in which information is stored, discovered and retrieved. In data sharing environments, all information or knowledge exist in the form of data and individuals or Community of Interests(COIs) share their data with others. DSE is a collection of data intended to suit the needs of a group of consumers. Data producers post data to one or more information spaces; data consumers pull the data they need from one or more information spaces.

Previous study focuses on unifying their enterprisewide data and designing architectures to maximize the usefulness and accessibility of that data. Even there are some research on quality of service in DSE, they mainly concerns on the accuracy, insistence of data. In this paper, however, we consider a more practical case that causes more difficulties in DSE. It is obvious that, in data sharing environments, distributed information sources are interconnected, which means that there may be more links between two information sources. Furthermore, there might be more information sources that have the same data. In such cases, we have to decide which information source should be selected to exchange data.

In quality of service routing(QoSR), each links in networks is associated with some metrics. Researchers focus on how to find a path that can satisfy the coming routing requests. We know that links in networks are usually redundant so as to make the whole sharing system more robust and improve the survival ability of the sharing system itself[2][3][4]. However, this brings more difficulties to QoSR and also to Quality of service study in DSE, i.e., data and link redundance also brings another embarrassment for decision-makers: which data source and which links should be used so as to make efficient decision?

Manuscript received September 5, 2006.
Manuscript revised September 25, 2006.

2. Problem Formulation

It is obvious that if we know how to select path between data consumer and each data source(SPCS), we have solved the whole problem.

Definition 1. SPCS Problem: Underlying network is given as a G(N;E), where N is the set of nodes and E is the set of links. Each link is associated with a k-dimensional metric vector $w\ (w_1, w_2, ..., w_k)$. w_i is an additive QoS metric, $i = 1,...,k$. P_{sd} denotes the path set between data consumer(also known as source node) and data source d. The problem is to find $p \in P_{sd}$ such that:

$$
\begin{aligned}
& w_i(p) = \sum\nolimits_{e \in P} w_i(e) \le w_i(\mathrm{q}) \\
& w_i(\mathrm{q}) = \sum\nolimits_{e \in \mathrm{q}} w_i(e), i = 1,2,...k \quad (1) \\
& q \in P_{sd}
\end{aligned}
$$

Path p can also be written as $p(w_1, w_2, ..., w_k)$. The path satisfies all k constraints is called as a feasible path. w_i may denote specific delay or cost of the links in networks. It is obvious that SPCS is a typical discrete multi-object optimization problem(MOOP). Now we will give some key concepts in MOOP.

3. Pareto Optimal

First, we give some requisite mathematical preliminaries.

Definition 2. Dominance: Vector $u = (u_1, ..., u_k)$ is said to dominate $v = (v_1, ..., v_k)$ if and only if u is partially less than v, namely $\forall i \in \{1,...,k\}$, $u_i \le v_i \ \Lambda\ \exists i \in \{1,...k\} : u_i < v_i$. We use $u \prec v$ to denote that u dominates v.

Definition 3. Pareto Optimal solution: Path $p(w_1, w_2, ..., w_k) \in P_{sd}$ is a Pareto Optimal solution if and only if there is no path $p'(w_1', w_2'\ , ..., w_k') \in P_{sd}$ such that $(w_1', w_2'\ , ..., w_k') \prec (w_1, w_2, ..., w_k)$.

Definition 4. QoS Metric Space (QoSMS): For any path $p(w_1(p), w_2(p), ..., w_k(p) \in G(N, E)$, if $w_i(p) \in W_i$, then $(W_1 \times W_2 \times ... \times W_k)$ is called QoS Metric Space (QoSMS).

Definition 5. Mapping F: Mapping F is a function that maps path $p(w_1(p), w_2(p), ..., w_k(p)$ to a point in QoSMS, i.e., $F(p)$ = $(f_1(p), f_2(p), ..., f_k(p))$ = $(w_1(p), w_2(p), ..., w_k(p))$.

Pareto optimal is a key concept in MOOP. The significance of Pareto optimal solutions lies in that if none of Pareto optimal paths can satisfy the constraints, then there is no path that can satisfy the constraints. Therefore, we can only consider Pareto optimal solutions of SPCS, which reduces the search space greatly. Like other discrete MOOPs, the solutions to SPCS are a set of Pareto optimal paths.

4. Algorithm SPCSA

Focusing on quality of service guarantee in DSE, we present an efficient algorithm SPCSA. By means of nonlinear path cost function and the concept of Pareto optimal, SPCSA manages selecting optimal data sources in data sharing environments. We first present the description of SPCSA.

4.1. Algorithm Description

Description of SPCSA

s: source node indicating data consumer

$\mathrm{d_i}$: data source

(1) For i=1, i $\le$ k, { $p_j(d_i) \mid j \ge 0$ }=GCPEDS(s, d_i).
//Generating Candidate Paths for Each Data Source d_i

(2) Computing the cost of candidate paths according to preferred information $(\alpha_1, \alpha_2, ..., \alpha_\mathrm{m})$, $\sum_{\mathrm{i=1,m}} \alpha_\mathrm{i} = 1$, $\alpha_\mathrm{i} \ge 0$, $\cos t(p_j(d_i)) = \prod_{\mathrm{i=1}}^{\mathrm{m}} \alpha_\mathrm{i} \sum_{e \in \mathrm{p_j(d_i)}} w_i(e)$

(3) D= $\mathrm{d_1}$;

(4) For i=1, $i < k$

(5) if $\min(\cos t(p_j(d_i))) \ge \min(\cos t(p_n(d_{i+1})))$
D= d_{i+1}
//j, n are the number candidate paths for data source

Fig 1. The description of SPCSA

Fig 1 describes the main steps of SPCSA. Firstly, SPCSA computes candidate paths { $p_j(d_i) \mid j \ge 0$ } for each data source d_i by means of GCPEDS(s, d_i). There may be multiple Pareto optimal paths in the path set $\{ p_j(d_m) \mid j \geq 0 \}$ for data source d_m. These Pareto optimal paths are all optimal if there is no preferred information about the paths and each of them can be used to get the data in d_m. While in the presence of preferred information $(\alpha_1, \alpha_2, ..., \alpha_k)$ about paths, optimal path can be selected according to the combination cost of the candidate paths.

4.2 Function GCPEDS(s, d_i)

Function GCPEDS(s, d_i) is used to generate candidate paths between data consumer and each data source. By means of normal measure based nonlinear path cost function(NMCF), GCPEDS(s, d_i) manages measuring path cost without the need of constraints information. Hence, GCPEDS(s, d_i) performs, for the first time, nonlinear path cost based precomputation. The proposed NMCF is mainly inspired by NBI method[10].

4.2.1 NBI Method

Normal Boundary Intersection(NBI) is proposed to generate approximate Pareto optimal solutions for continuous MOOP problems. By provided manually parameter β, NBI solves the following subproblem to find Pareto optimal solutions evenly distributed in QoSMS:

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

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

In this sub-problem, $\lambda \in R$, ϕ is a $k \times k$ matrix in which the i^{th} column is composed of the vector $F(x_i^*) - F^*$, in which x_i^* is the solution of which the i^{th} objective function has its minimum, $F(x_i^*)$ is the vector of objective functions evaluated at the point x_i^*, i=1,2,…,k and F^* is the vector containing the individual global minima of the objective functions. F^* is also called as Utopia point. β is a vector satisfying that $\sum_{i=1}^{k} \beta_i = 1$ and $\beta_i \geq 0$. $\hat{n} = \phi e$, $e \in R^k$. $\phi\beta$ is also called as Convex Hull of Individual Minima(CHIM).

For a MOOP problem, let $\hbar$ be the set of attainable objective vectors $\{F(x)\}$, and $\partial\hbar$ be the boundary of $\hbar$. In essence, the NBI method tries to find the portion of $\partial\hbar$ which contains the Pareto optimal points.

For a given β, $\phi\beta$ is a point in the CHIM. $\phi\beta + t\hat{n}, t \in R$, represents the set of points on normal $\hat{n}$. The point of intersection of the normal and $\partial\hbar$ closest to the origin is the global solution of (3). The constraints in (4) claim that the solutions to NBI are sure to be on the normal.

4.2.2 Normal Measure Cost Function(NMCF)

Although the NBI method is very efficient for an MOOP in case of continuous objective space, it cannot directly be used for discrete objective space. As clearly stated by Das and Dennis [26], the NBI method may fail if the objective space is discrete. The reason is that there may not be any point of intersection between the normal and the boundary for a particular setting of the NBI parameter.

To deal with the points in discrete space, we used a novel nonlinear path cost function NMCF which can efficiently be used in discrete objective space since it does not require that the point measured lies on the given normal[11]. We introduce the idea of NMCF first and then present GCPEDS(s, d_i).

Let p^{i*} denote the path in P_{sd} whose i^{th} objective achieves its minimum, i.e., for any path q $\in P_{sd}$, $f_i(p^{i^*}) \leq f_i(q)$, $1 \leq i \leq m$. $F^* = [f_1(p^{1*}), f_2(p^{2*}), ..., f_k(p^{k*})]^T$ $[f_1^*, f_2^*, ..., f_k^*]^T$ is called as Utopia point and the plane comprises of $F(p^{i*}), i = 1,2,...,m$ is called as Utopia hyperplane denoted by U. Let define a normalizing vector as $L = [l_1, l_2, ..., l_k]^T = F^N - F^*$, where $F^N \underline{\underline{\Delta}} [f_1^N, f_2^N, ..., f_k^N]^T$ and $f_i^N = \max[f_i(p^{1*}), f_i(p^{2*}), ..., f_i(p^{k*})]$.

We can now define the normalized $F(p)$ as $\overline{F}(p) = [\overline{f_1}(p), \overline{f_2}(p), ..., \overline{f_m}(p)]^T$, where $\overline{f_i}(p) = \dfrac{f_i(p) - f_i^*}{l_i}$.

Using the above definitions, we can define the following nonlinear path cost function:

$$\text{len(p)} = -\min(\lambda_1, \lambda_2, ..., \lambda_m) \tag{5}$$

$$s.t.\quad \overline{\phi}\beta + N = \overline{F}(p) \qquad (6)$$

where $\overline{F}(p) = [\overline{f_1}(p), \overline{f_2}(p), ..., \overline{f_m}(p)]^T$. The meaning of $\hat{n}$ is the same as the one in (4). Let $\gamma = (\gamma_1, \gamma_2, ..., \gamma_m)^T$ denote $\overline{\phi}\beta$ and the constraint condition in (6) can be rewritten as

$$\lambda_i n_i = \overline{f_i}(p) - \gamma_i \quad \text{i=1,2,…,m.} \qquad (7)$$

For the intuitive meaning of λ_i and how we can obtain different Pareto optimal points by varying β, please refer to [11].

4.2.3 Description of GCPEDS

As outlined in Fig 2, GCPEDS function mainly performs normal measure based nonlinear search in the manner of precomputation.

GCPEDS(s, d_i):
G: Network topology; s: Data consumer node;
d: Data source node.
(1)For i=1,2,…,*k*, compute p^{i*}
(2)Determine the Utopia point
$$F^* = [f_1^*, f_2^*, ..., f_k^*]^T$$
$$= [f_1(p^{1*}), f_2(p^{2*}), ..., f_k(p^{k*})]^T$$
(3)Compute normalizing vector
$$L = [l_1, l_2, ..., l_k]^T$$
(4)for each $\beta \in \{(a_1/b, a_2/b, ..., a_k/b) \mid \sum_{i=1}^{k} a_i/b = 1, 0 \le a_i \le b,\ a_i, b \in Z\}$
(a) $(\gamma_1, \gamma_2, ..., \gamma_k)^T \triangleq \overline{\phi}\beta$
(b)compute the normal of hyperplane U that crosses $(\gamma_1, \gamma_2, ..., \gamma_k)^T$:
$\hat{n} = (n_1, n_2, ..., n_k)$
(6)NM_Dijkstra(G, *s*, d_i)

Fig 2. Description of GCPEDS(s, d_i)

Firstly, GCPEDS computes the shortest paths individually w.r.t. each constraint. Then the Utopia point and Utopia hyperplane can be determined, see steps 1~2 in Fig 2. Step 4 indicates the search granularity, i.e., the larger the b, the finer the search granularity. Each point $\overline{\phi}\beta$ on the hyperplane corresponds to a normal that can be used to measure candidate paths. Step 5 calls NMCF based Dijkstra algorithm, or NM_Dijkstra, to perform nonlinear search. The relaxation procedure used by NM_Dijkstra is given in Fig 3.

NM_Dijkstra_Relax(*u*,*v*)
(1) Increase=$(w_1(u,v), w_2(u,v), ..., w_k(u,v))$
(2) Tempcost=cost(*u*)+ Increase
(3) newcost(i)=(Tempcost(i)- f_i^*)/l_i,
for i=1,2,…,k
(4) $\lambda_i = (\gamma_i -$ newcost(i))/n_i, for i=1,2,…,*k*
(5) templen= $-\min(\lambda_i)$
(6) If templen<len(*v*)
(7) len(*v*)=templen
(8) parent(*v*)=*u*
(9) cost(*v*)=Tempcost
(10)end if

Fig 3. Relaxation procedure of NM_Dijkstra

5. Performance Evaluation

Performance evaluation includes three parts. (1) Evaluation of path cost when data consumer only has preferred information. (2) Evaluation of success rate when data consumer has specific constraint requests. (3) Evaluation of response speed when data consumer has specific constraint requests.

5.1 Simulation Model and performance measures

The underlying network topologies used for simulations are randomly generated based on Waxman's model with 200 nodes. Data consumer node and data source nodes are also randomly generated and at least two nodes away. Without explicitly announcement, we assume that there are 5 data sources in the network.

To fully evaluate our algorithm, we compare SPCSA with two typical algorithms with high performances, namely H_MCOP[7][8] and MEFPA[6]. While the former is chosen as a representative for on-demand algorithms, the latter is chosen as a representative for precomputation based algorithms. Although these two algorithms are not proposed to address multiple data source selection problem, they can be used to search the shortest path between two nodes in networks. Viewing data consumer node and each data source as a node pair, we can compare H_MCOP and MEFPA with SPCSA. The real computation cost of SPCSA and MEFPA are related to the extent to which they care about the search granularity, i.e., parameter b reflects the computation cost of these two algorithms(parameter b means that Dijkstra algorithm will be called C_{b+k-2}^{k-1} times in the precomputation phrase).

We use preferred path cost, i.e., the least cost of the path found by SPCSA between data consumer and each data source according to specific preferred information, to evaluate the efficiency of the paths found by SPCSA. As the key performance measure, we use success rate (SR), the ratio of the constraints satisfied by heuristic to the total constraints generated, to evaluate the feasibility of SPCSA. Finally, we use precomputation success rate(PSR), the ratio of the number of constraints satisfied by primary paths to the number of constraints generated, to evaluate the response time of SPCSA.

5.2 Preferred Path Cost

To keep the computation cost at a tolerable level, we let parameter b=7 in MEFPA denoted by MEFPA(7), and b=3,5,7 in SPCSA denoted respectively by SPCSA(3), SPCSA(5), SPCSA(7). Because the real path cost is meaningless for the evaluation of algorithms, we use the cost of paths found by SPCSA(7) to normalized other costs. Fig 4 shows the simulation results. Y-coordinate is the normalized preferred path cost(NPPC). X-coordinate denotes preferred information(PI). There are six kinds of preferred information used in the simulations, i.e., PI_i =(1.2-0.2i, 0.2i-0.2),i=1,2,...,6. We can see from the Fig that, compared with MEFPA and H_MCOP, SPCSA has small path cost and SPCSA(7) has the least path cost.

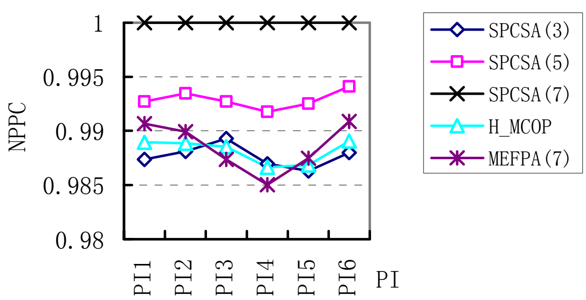

Fig 4. Evaluation of preferred path cost

5.3 Response Time

SPCSA can response at once by precomputation when data consumer has preferred information. But when data consumer has specific QoS request, SPCSA will perform on-demand computation if the paths found by precomputation cannot satisfy the request. Although on-demand computation increases the success rate, it suffers a longer response time. Hence, we use precomputation success rate(PSR) to evaluate the response time of SPCSA. We generate the constraints as that in [8]. Fig 5 shows the simulation results. Y-coordinate is precomputation success rate(PSR) and X-coordinate is the number of data sources(NDS). We can see from the Fig that over 92% constraints can be satisfied by the paths found in precomputation phase and PSR increases with NDS. We also know from Fig 5 that even SPCSA algorithm need on-demand computation, it calls Dijkstra algorithm at most NDS times. So it is reasonable to believe that SPCSA response very quickly.

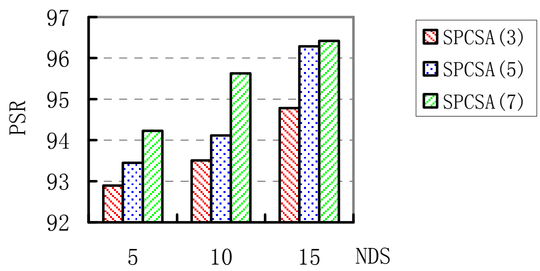

Fig 5. Evaluation of Response Time

6. Conclusions

How to select an appropriate data source and path in data sharing environments with a low cost and short response time is a key challenge that researchers who care about the quality of service. Based on Pareto optimal theory, we analyze the advantages and shortcomings of related algorithms and then we propose SPCSA algorithm. Based on normal measure based path cost function, SPCSA searches nonlinearly approximate Pareto optimal paths between data consumer and each data source, which reduces search space greatly. Extensive simulations show that, compared with related algorithms, SPCSA manages achieving a high success rate and a short response time while keeping computation cost at a low level.

As the future work we plan to further investigate distributed search algorithm so as to avoid the need to maintain all network state information. In addition, we plan to study the problem of data source selection when data itself is different in completeness, correctness and consistence etc.

Refenences:

[1] Exploring information superiority : a methodology for measuring the quality of information and its impact on shared awareness, Walt L Perry, David Signori, John Boon Santa, Monica, RAND, 2004
[2] Operational Command and Control in the Information Age Milan N. Vego. Joint Forces Quarterly, 11 January 2005
[3] Information Operations and Millennium Challenge, Mark W. Maiers and Timothy L. Rahn, Joint Forces Quarterly, 11 January 2005
[4] Performance Analysis of Data Sharing Environments, A. Dan, The MIT Press, 1992
[5] Juttner, A., Szviatovszki, B., Mecs, I., Rajko, Z. “Lagrange relaxation based method for the QoS routing problem”, in Proceedings of the INFOCOM 2001, Vol. 2, 859-868.
[6] Cui, Y., Xu, K., Wu, J. “Precomputation for multi-constrained qos routing in high-speed networks”, in Proceedings of the INFOCOM 2003 Conference, San Francisco.
[7] Korkmaz Turgay, Krunz Marwan, “Multi-constrained optimal path selection”, in Proceedings of the INFOCOM 2001, 834-843.
[8] Korkmaz Turgay, Krunz Marwan, “Routing multimedia traffic with QoS guarantees”, IEEE Transactions on Multimedia, 2003, 429-443.
[9] P. Van Mieghem and F.A. Kuipers, Concepts of Exact QoS Routing Algorithms, IEEE/ACM Transaction on Networking, 2004, Vol 12. 851-864.
[10] Das, I.; Dennis, J.E., “Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems”. SIAM J. Optim. 8, 631-657. 1998
[11] Highly Responsive and Efficient QoS Routing Using Pre- and On-demand Computations along with a New Normal Measure, Yanxing Zheng, Turgay Korkmaz, Wenhua Dou, Jing Tian, to appear in Computer Networks Journal.

XiaoQing Wang received the B.S. and M.S. degrees in Department of Computer Science, National University of Defense Technology, in 1993 and 1998, respectively. He is now a PhD candidate in Tsinghua University, Beijing, and assistant professor in BeiJing Institute of System Engineering. His research interests include Data Sharing, Software Testing and Information Technology.

Yanxing Zheng received PhD degree in 2005, China. He is now with BeiJing Institute of System Engineering. His research interests include Data Sharing, Software Testing, Information Technology and Optimization Algorithms.