SLA-based Data Integration on Database Grids

Tiezheng Nie [1], Guangqi Wang [1], Derong Shen [1], Meifang Li [1], Ge Yu [1]
[1] *Dept. of Computer Sci. and Eng., Northeastern University, Shenyang, 110004*
nietiezheng@ise.neu.edu.com; shendr@mail.neu.edu.cn; yuge@mail.neu.edu.cn

Abstract

A Database grid can provide a robust distributed environment for accessing and manipulating different database resources. While current challenges focus on managing database resources and ensuring the quality of data integration. This paper describes a Service Level Agreement (SLA) based data integration model, which includes SLA-based Resource Description Model, SLA-based Query Model, SLA-based Matching Algorithm and SLA-based Evaluation Model. The SLAs of a database resource is defined by SLA-based Resource Description Model. The SLA-based Query Model is defined to normalize the various types of queries. The SLA-based Matching Algorithm has been proposed to match schemas of resources for each sub-query, and then the quality of database resources is evaluated by SLA-based Evaluation Model to obtain the final query solutions. Experimental results have shown that SLA-based Data Integration Model can provide more stable database resources and save time cost in data integration applications as well.

1. Introduction

Grids provide both high performance computation and mass information resources shared for scientific communities and business applications. The Open Grid Services Architecture (OGSA) [1] provides a service-based platform for grid applications, and OGSA-DAI [2], a service-based architecture, is also been designed and implemented for accessing grid services wrapped by databases. While in the database grid environment, how to manage database resources and seamlessly integrate these distributed databases with high quality has been a challenging issue. Databases are often autonomic, heterogeneous, dynamic, which increase the complexity of data integration. Meanwhile, applications with various requirements need different data sources, and thus the database grid must provide database resources on demand to ensure the quality of the data integrated. Comparing with traditional Web services, the metadata of databases is more complex. For a user query, more than one database is involved, and dynamic data integration is required. Unfortunately, query optimization techniques for distributed DBMS can not nontrivially and directly be applied to a database gird system, because there are some fundamental differences between them [3]. In a database grid, resources are composed of autonomous and heterogeneous databases, where autonomy indicates database resources are managed completely by their local DBMS.

In this paper, we present a SLA-based data integration model for the purpose of implementing the data integration on demand in a database grid. So, this paper focuses on resolving the following issues in a database grid:

- Describe database resources in the database grid;
- Define a query model;
- Match the capabilities of database resources with the requirements of a application;
- Reduce the cost and guarantee the quality of the data integration.

The remainder of the paper is organized as follows. Section 2 presents related works. Section 3 describes the overview of the database grid for integrating data based on SLA. Section 4 presents SLA-based Resources Description Model, while Section 5 introduces SLA-based Query Model. In Section 6, SLA-based Matching Algorithm is proposed to match schemas of queries with those of resources. Then to satisfy query's requirements, SLA-based Evaluation Model for choosing the database resources with high quality is given in Section 7. Section 8 discusses experimental results on SLA-based Data Integration Model, while Section 9 summarized the conclusions.

2. Related Work

Nowadays, there are many existing projects focusing on data access and integration on a database grid, typically, OGSA-DAI and OGSA-DQP. OGSA-

31st Annual International Computer Software and Applications Conference(COMPSAC 2007)
0-7695-2870-8/07 $25.00

DAI [2] has been defined and implemented for accessing database over a grid system, which as a grid-inspired effort, focusing on accessing data in relation databases and the XML data and aiming to define a standard data service framework based on web service technologies. OGSA-DQP is for static query processing constructed on OGSA-DAI, emphasizing at parallel executions to improve the efficiency of the query processing.

DartGrid [4, 5] is an implemented prototype system whose goal is to provide a semantic solution capable of deployment in grid-settings, while DartGrid II [3] utilized a query optimization approach for query processing, and proposed the query optimization algorithms with heuristic, dynamic, and parallel features. Another typical effort is applying QoS-based methods [11] to meeting application demands. Al-Ali et al. [7] proposed an algorithm to enable the dynamic adjustment according to the pre-defined SLA. Zhao et al. [8] provided a re-allocation mechanism as service resources in failure. Quan[9] proposed assigning the grid resources of a workflow based on SLAs. Thereby, from the prospective of the quality of data integration in database grids, SLA is a necessity and has become a very important factor for satisfying applications' requirements. Hu et al. [10] proposed the methodology that attempted to evaluate and monitor the quality of data and knowledge integrated. Frederick [6] has developed an algorithm focusing on optimizing query execution plans and storage layout simultaneously to meet an SLA.

3. Overview of the SLA-based Data Integration

To support SLA-based data integration, we propose architecture of database grid with SLA-based data integration model; which is shown in Figure 1. In the architecture, if an autonomous database wants to be shared by third applications, its metadata and service levels should be published into Resource Registry. For a query, first of all, Query Manager parses the query and generates a set of sub-query plans. Then Resource Discover invokes Schema Matching Agent for matching sub-query schemas with that of database resources to discover appropriate databases, so as to generate candidate query solutions. Because a substantial of database resources matched will be found on demand, they should be filtered based on SLAs by Resource Evaluator for choosing databases that have the best service levels for the query solutions. Meanwhile, the URIs of these resources chosen is returned to Resource Discover for invoking. Next,

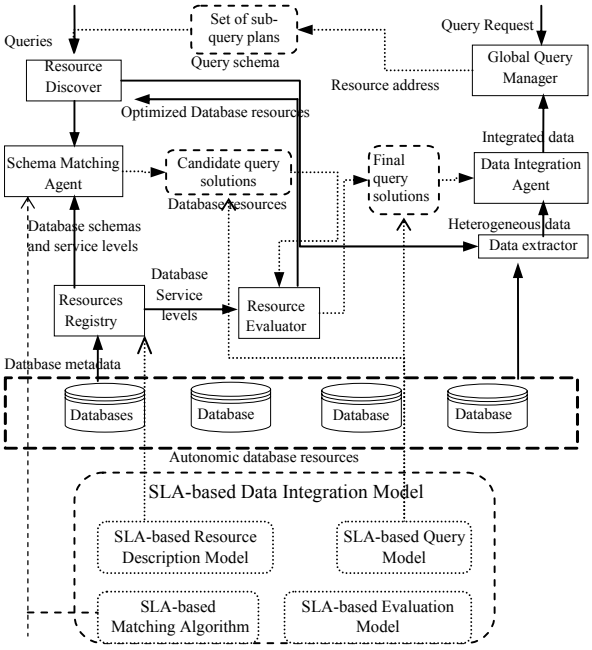

Figure 1. Architecture of Database Grid with a SLA-based Data Integration Model

Data Extractor accesses these databases based on query solutions to obtain the heterogeneous data for Data Integration Agent integrating the data. Finally, as for Global Query Manager, the integrated result are automatically mapped to the format required by the query based on the matching information collected by Resource Discovery, and the final result data be returned to customers.

Since the query optimization techniques [12] are very mature in relational databases, our query parsing makes use of traditional algorithms to take advantages of earlier works. Therefore, we focus on SLA-based Data Integration Model, in which, SLA-based Resources Description Model describes the capability of database resources, SLA-based Query Model normalizes the query execution plans with SLA information, while SLA-based Matching Algorithm and SLA-based Evaluation Model take charge of choosing databases from Resource Registry and evaluating them to generate the final query solution. Since the databases selected must be precise, economical, stable and efficient for the next data integration, service levels used to describe the different quality of a database resource should be provided by the respective provider, and be applied to resources matching and evaluating.

4. SLA-based Resource Description Model

In our SLA-based resource description model, the metadata is the basic information to be submitted to Resource Registry, and the most important part of metadata is the schemas of a database, which provides the data structures of the database. However, the information of schemas can't provide the quality of the data database resource. Therefore, a Database Resource Description Model with SLA for describing database resources is proposed to support data resources negotiation and quality control. For describing database resources, a database resource wrapped as a Grid Service is defined as follows.

Definition 1. Grid Service for Database Resource(*DR*). Grid Service for Database Resource(*DR*) is denoted as:

$$DR = \{M, SL\}$$

Where, M is the metadata of *DR*, which includes accessing method, capability description, and other features of the database. *SL* is the set of service levels of the database resource provided by its database provider. $SL=\{SL_1, SL_2,...,SL_n\}$, and $SL_i=\{S_i, Q_i, Co_i, Au_i\}$, where S_i, Q_i, Co_i, Au_i denote the set of schemas, the QoS, the compensatory rule and the authorization rules included in SL_i respectively.

Schema Information(S_i). S_i is a set of data schemas denoted as $S_i=\{S_{i1}, S_{i2},..., S_{in}\}$, which defines the data structures on the service level SL_i, and consists of multiple sub-schemas of the global schema. The sub-schemas defined in a service level are named as service level schemas. For example, Figure 2 shows the relationship between the service level schemas and the database global schema.

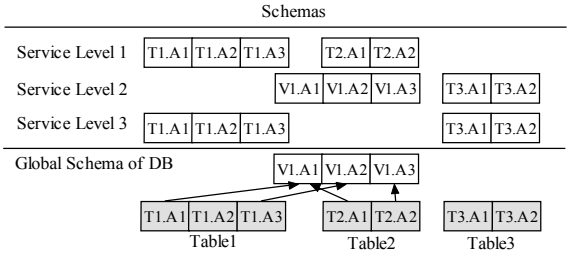

Figure 2. Relationship between the service level schemas and the global schema

Quality Information(Q_i). It declares the quality of service (QoS) on the *i*th service level. The QoS level is composed of quality parameters describing the service level schema, which is used to evaluate the performance of a database resource and estimate the efficiency of the data integration on the service level. Here, the parameters mainly include the query processing cost(C_{proc}), the amount of data(D_{amount}) provided by the resource and the price of using the grid service (C_{price}), which are all provided by database providers. Thus, the quality information (Q_i) is denoted as: $Q_i = (C_{proc}, D_{amount}, C_{price})$, while the other QoS parameters such as the transmitting cost, are excluded at publishing time.

Compensatory rules(Co_i). When a violation of SLA occurs, the grid system can check the compensation rules defined in the service level SL_i, and execute the candidate database resources or the penalization is run.

Authorization(Au_i). To enhance the security shared among the database resources, privileges and constrains control for accessing the database resource are defined, i.e., $Au_i=\{A, C\}$, where, A is the set of the privileges, C the set of the constraints.

The schema S_i and the quality information Q_i are necessary for describing a service level of a database resource, while the compensatory and the authorization setting is optional.

The main advantages of SLA-based Resources Description Model can be summarized as follows:

Supporting QoS-based evaluation. Based on QoS information, resources can be evaluated and the most appropriate resources are chosen to data integration.

Guaranteeing security. Based on the global schemas of a database resource, multiple logic views with multi-service levels are defined for sharing. So the customer only knows the service level schemas (logic view) defined in the current service level.

5. SLA-based Query Model

The SLA-based Query Model is introduced in the remainder of this section, while, SLA-based Matching Algorithm and SLA-based Evaluation Model will be presented in Section 6 and Section 7 respectively.

5.1 Query Classification

In our database grid system, we classify queries into three types based on data integration methods:

Aggregation Query. These queries involve multiple distributed databases with the same or similar schema, the result of which equals the union of these result data.

Join query. In this case, the query is satisfied by joining two relations in different databases.

Complex Query. The query needs both of the union and join operations or includes complex math functions or ordering requirements, in which, the service level of the database resources is an important factor to obtain the optimal integration results with better performance.

5.2 SLA-based Query Model

0-7695-2870-8/07 $25.00

Let a query q contains a set of attributes $A=\{a_1, a_2, \ldots, a_n\}$ in the query schema, where a_i is an attribute of the query q, then SLA-based Query Model QM is denoted as:

$QM=\{q(A,C), DS, QP, SO\}$

Where C is the query constrains on A. $DS=\{DS_1, DS_2,\ldots, DS_m\}$ represents a set of data sources, while a data source DS_i is denoted as $DS_i = \{R_1, R_2, \ldots, R_m\}$, R_i is a relation in the database DS_i and composed of $\{a_{i1}, a_{i2}, \ldots, a_{in}\}$, $a_{ij} \in A$. A data source DS_i can map to one or more service level schemas to satisfy A and C in q. For example, let S_i $(S_{i1}, S_{i2},\ldots, S_{in})$ in SL_i and $(R_{j1}, R_{j2},\ldots, R_{jn})$ in DS_j, if SL_i matches with DS_j, then $(S_{i1}\cup S_{i2}\cup\ldots\cup S_{in}) \cap A \neq \Phi$ and $((S_{i1}\cup S_{i2}\cup\ldots\cup S_{in}) \cap (R_{j1}\cup R_{j2}\cup\ldots\cup R_{jm})) \cap A \neq \Phi$.

$QP=\{(QP_1, DS_1), (QP_2, DS_2), \ldots, (QP_m, DS_m)\}$ represents a set of query plans , each of which runs over one or more database resources. SO is the set of the candidate query solutions, $SO=\{SO_1, SO_2, \ldots, SO_n\}$, $SO_i=\{QP_i, DS_i, S\text{-}SL_i\}$, $S\text{-}SL_i$ is the set of service levels of resources satisfying the data source DS_i in QP_i.

Initially, SO is empty, after the matched database resources are found, the candidate SO becomes valid. Since there are a lot of different solutions for one query, quality criteria are proposed for evaluating these solutions.

Figure.3 shows an example of the query model QM, in which, relation R is equivalent to the schemas in a SL, where $DS=\{DS_1, DS_2\}$, $QP=\{QP_1, QP_2\}$, the data source DS_1 is mapped by R_{config} and R_{price}, DS_2 by $R_{car\ type}$, and the query solutions $SO=\{SO_1, SO_2\}$, where $SO_1=\{QP_1, DS_1, S\text{-}SL_1\}$, while $SO_2=\{QP_2, DS_2, S\text{-}SL_2\}$.

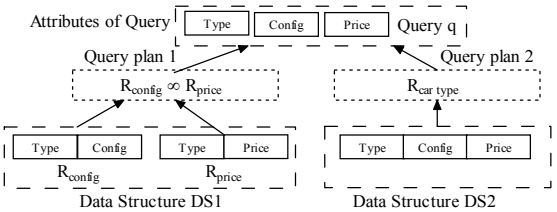

Figure 3. The Example of Query Model

6. SLA-based Matching Algorithm

To discover databases satisfied users' requirements, service level schemas of resources are matched with that of DS_i in each query plan QP_i. Suppose the semantic heterogeneity of attributes has been resolved, and the matching approach is mainly based on the relation schemas in databases. Here, some definitions are presented before the matching algorithm is given.

Definition 2. Full Match. If a schema S_{SLi} of a service level SL_i fully matches with a relation R_{QPj} of a query plan QP_j, then SL_i is full matching with QP_j, which is denoted as $FM(SL_i, QP_j)$.

Definition 3. Overlay Match. If part of data schemas S_{SLi} in SL_i matches with relation R_{QPj} of QP_j, then the overlay match between SL_i and QP_j is denoted as $OM(SL_i, QP_j)$. Here, $FM(SL_i, QP_j)$ is more superior than $OM(SL_i, QP_j)$.

Definition 4. Half Match. For a query plan QP_j with join operation, only one of join relations in QP_j can match with a schema S_{SLi} of the service level SL_i, which is called Half Match, $HM(SL_i, QP_j)$.

We propose an algorithm (Matching Algorithm) to improve the efficiency of finding query solutions. The inverted SL index is used to locate the service levels of databases, and $SL(R_j)$ means the set of service levels that contains relation R_i (equal to data schema S_i). The Matching Algorithm works as follows.

Algorithm 1. SLA-based Matching Algorithm.

Input: the inverted SL tables of schemas;
$QP\{QP_1, QP_2, \ldots, QP_m\}$: the query plans of query q.
$DS\{DS_1, DS_2, \ldots, DS_m\}$: the data sources of query q.

Output: $SO\{SO_1, SO_2, \ldots, SO_n\}$: the set of candidate query solutions for query q.

Begin:
Initial solution set SO of the query q as null;
For i = 1 **to** m
Initial service level caches $SL_i(full)$ and $SL_i(half)$ as null
For each relation $R_j, j = 1\ldots r$, **in** DS_i of QP_i
search $SL(R_j)$ from the inverted tables
filter $SL(R_j)$ by $SL_i(full)$ and $SL_i(half)$
for k = 1 **to** p $\{ SL(R_j) = \{ SL_1, \ldots, SL_p \}\}$
if $FM(SL_k, QP_i)$ = true **or** $OM(SL_k, QP_i)$ = true
add SL_k into $SL_i(full)$
end-if
if $HM(SL_k, QP_i)$ = true **and** R_j in a data join query
add $\{SL_k, R_j\}$ into $SL_i(half)$
end-if
end-for
end-for
optimize $SL_i(full)$ and $SL_i(half)$ by database
assemble $\{\{SL_1, R_1\},\ldots,\{SL_r, R_r\}\}$, $\{SL_k, R_j\} \in SL_i(half) \cup SL_i(full), j = 1\ldots r$
add $\{QP_i, DS_i, \{SL_1,\ldots, SL_r\}\}$ into solution set SO
end-for
end

The basic idea of Algorithm 1 is to match the schemas in service levels of database resources with that of customers' query, and classify the service levels by their matching types. After Algorithm 1, the candidate solutions with better performance are selected from the large set of solutions.

7. SLA-based Evaluation Model

The SLA-based Evaluation Model is proposed for selecting the optimal solutions based on the quality of

service as the final results. In SLA-based Evaluation Model, parameters of evaluating resources include time cost(C_{time}), schema completeness(C_{schema}), data amount(C_{Amount}), data QoS(C_{QoS}) and price (C_{price}). Details of the parameters are defined as follows.

Definition 5. Time cost(C_{time}). It is the execution time of a solution that includes the query processing cost(T_{proc})，the data transfer cost(T_{com}) and the data integration cost(T_{in}). Then, the cost of executing a query q on a service level SL_i of a database ($Cost_q(SL_i)$) is defined as:

$$Cost_q(SL_i) = T_{proc}(q) + T_{com}(q) \quad (1)$$

The total time cost of a solution is denoted as:

$$C_{time}(SO_k) = Max_{i=1}^{p}(Cost_{qi}(SL_i)) + T_{in}(q) \quad (2)$$

Where p is the number of the service levels in the solution SO_k, Max($Cost_{qi}$) is the maximum cost of accessing the distributed databases, since the sub query q_i can runs in parallel. If a sub query includes a join predicate, the join cost of a query plan($Cost_J$) will be included.

Definition 6. Schema completeness (D_{schema}). That is the matching degree between the schema of a query plan and the service level schema of a solution. D_{schema} is calculated as follow:

$$D_{schema}(SO_k) = \frac{1}{p}\sum_{i=1}^{p}(\frac{1}{m_i}\sum_{j=1}^{m_i} Sim(a_{ij}, a'_{ij})) \quad (3)$$

where p is the number of SL in SO_k, m_i is the number of the matched attributes between SL_i and DS, and $Sim(a_{ij},a'_{ij})$ denotes the similarity degree between a_{ij} in SL_i and a'_{ij} in DS.

Definition 7. Data Amount(D_{amount}). Data Amount is the amount of the potential data stored in a database.

Definition 8. Data Quality(D_q). The data quality is the valid data searched from a database, which is defined as:

$$D_q = Count_{valid}(q) / Count_{total}(q) \quad (4)$$

Where $Count_{valid}(q)$ is the number of valid data satisfying the query's request(q) and $Count_{total}(q)$ is the total amount of data returned by accessing the database resource.

Definition 9. Price Cost(C_{price}). It denotes the price for using the service level, and then the total price cost of a solution is defined as:

$$C_{price}(SO_k) = \sum_{j=1}^{p} C_{price}(SL_j) \quad (5)$$

Where $C_{price}(SL_j)$ is the price cost of the service level SL_j in the solution SO_k .

For a solution SO_k, the score of the database resource($Score_{data}$)and the score of integrating cost ($Score_{cost}$)are defined as:

$$Score_{data}(SO_k) = w_s * D^k_{schema} + w_a * D^k_{amount} + w_q * D^k_q \quad (6)$$

$$Score_{cost}(SO_k) = w_t * C^k_{time} + w_p * C^k_{price} \quad (7)$$

Where w_s, w_a *et al,* denote the weights of the evaluation factors.

So the total score of the solution is denoted as:

$$Score_{total}(SO_k) = Score_{data}(SO_k) / Score_{cost}(SO_k) \quad (8)$$

If the solution SO_k needs more databases, the maximum cost of each factor is applied.

Thereby, the most appreciative databases are ranked base on the scores of the solutions to generate the final integrated data results.

8. Experiment Results

In this section, we design a series of experiments for measuring the efficiency of the SAL-based data integration model.

8.1. Experimental Setup

Experiments were conducted on a database grid prototype system, and the experiments are finished in the following environment:

Databases. Multiple databases on Oracle 9i were created and were deployed on the servers with different hardware configurations to simulate different performances. The server machines include an IBM pSeries 630, and some Pentium 4 computers. The database grid we constructed included ten databases. Each of databases is composed of fifteen relations. The scale of relation is from 183,201 records to 16,564,848 records. We created index on part of relations, which would take different performance.

Service levels. For each database service, we setup a set of service levels with different QoS, where the parameters of the QoS include the transferring speed, query processing time, the data amount and so on.

Schemas. In a database resource, more different the service level schemas are defined based on the global schema of the database, and many global schemas are published for applications sharing.

In the following, two types of queries are defined to measure their performance by comparing these queries with SLA to that without SLA.

8.2. Experimental Analysis

(1) The time cost of Aggregation Query

In the first experiment, aggregation queries are executed for data integration, in which, the specified data with the same schema from different database resources are extracted. The time costs of integrating

different amount of data are shown in Figure.4. The time cost of the Aggregation Query is little lower than one with naive method and the difference becomes bigger with the amount of data increasing.

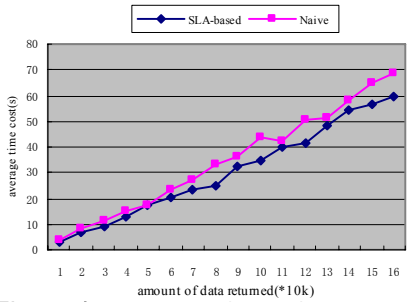

Figure 4. The Time Cost of Integration with Aggregation Query

(2) The time cost of Join query

For queries with join predicates, the SLA-based integration model focuses on finding the most optimal solutions with the least time cost for transferring data from one site to another. The result of the experiment is shown in Figure.5, in which, the performance of the data integrations with SLA-based integration model is the most efficient comparing with that without optimization and with schema-only optimization. In schema-only data integrations, only the completeness of the schems of the final query result is considered, while in SLA-based integration model, many factors, are all considered for estimating the best solution.

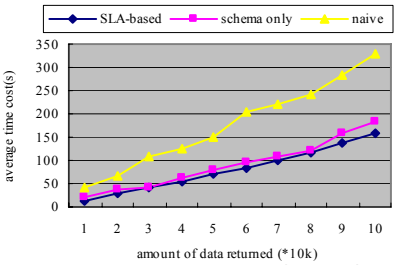

Figure 5. The Time Cost of the data Integration with Join Query

9. Conclusions

In this paper, the architecture of a database grid with an SLA-based data integration model is introduced. To realize the SLA-based data integration effectively, SLA-based Resource Description Model, SLA-based Query Model, SLA-based Matching Algorithm, and SLA-based Evaluation Model are proposed. SLA-based Resource Description Model is used to describe the capability of the traditional database resources with a set of service levels. While three types of queries are described in SLA-based Query Model, and SLA-based Matching Algorithm is used to discover the appropriate databases for each query plan to obtain candidate query solutions. SLA-based Evaluation Model is used to evaluate the candidate solutions by using the QoS information given in service levels to select the set of optimal solutions for data integration. Finally, the experiments have demonstrated the data integration with the SLA-based model worked more efficient.

Acknowledgement

This work was supported in part by the National Science Foundation (60673139, 60473073 and 60573090), and the National Ministry of Education Project of China (GFA060448)

References

[1] I. Foster, C. Kesselman. "The physiology of the grid: an open grid services architecture for distributed systems integration", *Technical report*, Chicago, January 2002.

[2] M. Antonioletti. "The design and implementation of Grid database services in OGSA-DAI", *Grid Performance*, volume 17, 2005, pp 357-376.

[3] X. Zheng, H. Chen, Z. Wu, "Query Optimization in Database Grid", *LNCS 3795*, 2005, pp. 486-497.

[4] Z Wu, H Chen, "DartGrid: Semantic based Database Grid", *LNCS. 3036*, 2004, pp. 59-66.

[5] H Chen, Zh Wu, "RDF-Based Schema Mediation for Database Grid" *Grid Computing 2004*, 2004, pp. 456-460.

[6] F.R. Reiss, T. Kanungo, "Satisfying Database Service Level Agreements while Minimizing Cost through Storage QoS", *International Conference of Services Computing*, 2005, pp. 13–21.

[7] R. Al-Ali, A. Hafid, O.F. Rana, "QoS Adaptation in Service-Oriented Grids", *Concurrency and Computation: Practice and Experience Journal,16(5)*, 2004, pp. 401-412.

[8] X Zhao, B Wang, "Qos-based Algorithm for Job Allocation and Scheduling in Data Grid", *International Conference on Grid and Cooperative Computing Workshops*, 2006, pp. 20–26.

[9] D.M. Quan, O. Kao, "Mapping Workflows onto Grid Resources Within an SLA Context", *EGC 2005*, 2005, pp. 1107-1116.

[10] J. Hu, N. Zhong, "Organizing Multiple Data Sources for Developing Intelligent e-Business Portals", *Data Mining and Knowledge Discovery, vol. 12*, 2006, pp. 127–150.

[11] L. Liu, T. Yu, Y. He, "A QoS-based GFAM Scheduling Approach for Manufacturing Grid", *International Conference on Computer and Information Technology*, 2005, pp. 334-338.

[12] M. Steinbrunn, G. Moerkotte, and A. Kemper, "Heuristic and randomized optimization for the join ordering problem", *VLDB Journal, 6(3)*, 1997, pp. 191–208.