



UNIVERSITY
OF WOLLONGONG
AUSTRALIA

University of Wollongong
Research Online

Faculty of Engineering and Information Sciences -
Papers: Part A

Faculty of Engineering and Information Sciences

2013

Towards minimizing cost for composite data-intensive services

Lijuan Wang

University of Wollongong, lw840@uowmail.edu.au

Changyan Di

Lanzhou University

Yan Li

Lanzhou University

Jun Shen

University of Wollongong, jshen@uow.edu.au

Qingguo Zhou

Lanzhou University

Publication Details

Wang, L., Di, C., Li, Y., Shen, J. & Zhou, Q. (2013). Towards minimizing cost for composite data-intensive services. In W. Shen, W. Li, J. Barthes, H. Zhu, J. Yong & X. Li (Eds.), 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design (pp. 293-298). IEEE Xplore: IEEE.

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library:
research-pubs@uow.edu.au

Towards minimizing cost for composite data-intensive services

Abstract

Service-oriented architecture provides a scalable and flexible framework to implement loosely-coupled, standards-based, and protocol-independent distributed computing. One of its goals is to make use of the distributed services with different functions to build powerful composite services. Service composition is an active research area in service computing. Existing research is endeavoring to achieve desirable quality levels of composite services and improve customer satisfaction. The service-oriented approach using Web services is also of great interest for the implementation of data-intensive processes such as data mining, image processing and so on. The applications based on data-intensive services have become the most challenging type of applications in service-oriented architecture. The service-based strategy provides maximal flexibility when designing data-intensive applications. Huge data sets that may each be replicated in different data centers have to be exchanged between several services. The movement of mass data influences the performance of the whole application process. Especially, the price of services will be different when considering the data center's locations and the amount of data transferred. It is desirable to find the cost-minimized service composition solution in service computing. Therefore, how to select appropriate data centers for accessing data replicas and how to select services with lowest associated costs are emerging problems when deploying and executing data-intensive service applications. In this paper, a cost-minimizing service composition model for data-intensive applications is proposed. Furthermore, how bio-inspired algorithms offer advantages to solve such problems will be presented.

Keywords

towards, minimizing, cost, intensive, data, services, composite

Disciplines

Engineering | Science and Technology Studies

Publication Details

Wang, L., Di, C., Li, Y., Shen, J. & Zhou, Q. (2013). Towards minimizing cost for composite data-intensive services. In W. Shen, W. Li, J. Barthes, H. Zhu, J. Yong & X. Li (Eds.), 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design (pp. 293-298). IEEE Xplore: IEEE.

Towards minimizing cost for composite data-intensive services

Lijuan Wang, Jun Shen

School of Information Systems and Technology
University of Wollongong
NSW, Australia

Email: lw840@uowmail.edu.au, jshen@uow.edu.au

Changyan Di, Yan Li, Qingguo Zhou

School of Information Science and Engineering
Lanzhou University
Lanzhou, P. R. China

Email: dizhy@lzu.edu.cn, ynali@lzu.edu.cn, zhouqg@lzu.edu.cn

Abstract—Service-oriented architecture provides a scalable and flexible framework to implement loosely-coupled, standards-based, and protocol-independent distributed computing. One of its goals is to make use of the distributed services with different functions to build powerful composite services. Service composition is an active research area in service computing. Existing research is endeavoring to achieve desirable quality levels of composite services and improve customer satisfaction. The service-oriented approach using Web services is also of great interest for the implementation of data-intensive processes such as data mining, image processing and so on. The applications based on data-intensive services have become the most challenging type of applications in service-oriented architecture. The service-based strategy provides maximal flexibility when designing data-intensive applications. Huge data sets that may each be replicated in different data centers have to be exchanged between several services. The movement of mass data influences the performance of the whole application process. Especially, the price of services will be different when considering the data center's locations and the amount of data transferred. It is desirable to find the cost minimized service composition solution in service computing. Therefore, how to select appropriate data centers for accessing data replicas and how to select services with lowest associated costs are emerging problems when deploying and executing data-intensive service applications. In this paper, a cost minimizing service composition model for data-intensive applications is proposed. Furthermore, how bio-inspired algorithms offer advantages to solve such problems will be presented.

Keywords—*bio-inspired algorithms, service composition, quality of service, data-intensive application.*

I. INTRODUCTION

Service-oriented computing (SOC) refers to a computing paradigm that promotes the development of rapid, low-cost, inter-operable, evolvable, and massively distributed applications by utilizing services. The aim of SOC is to create dynamic business processes and agile applications covering different organizations and computing platforms by assembling application components into a loosely-coupled network of services. To build the service model, SOC relies on the service-oriented architecture (SOA), which is an emerging approach that combining elements of software architecture and enterprise architecture [22]. It is a popular system model to implement loosely-coupled, standards-based, and protocol-independent distributed computing. Being one of the most important technologies of SOA, Web service refers to a software system or platform, which provides a standard communication

method so that a variety of applications can inter-operate over the network. It provides a way to integrate Web-based applications using the common language provided by the extensible markup language (XML). The advent of Web service standards boosted the proliferation of the principles as well as the actual implementations of service orientation. Service orientation is founded on the idea of composing business processes or workflows by discovering and invoking the most suitable services (for example, based on cost or other quality of service attributes) rather than building new applications to satisfy a particular domain or business functional requirements [9].

The service-oriented approach using Web services is also of great interest for the implementation of data-intensive processes such as data mining, image processing and so on. In recent years, huge collections of data have been created by advances in technology areas such as digital sensors, communications, computation, and storage. Every area of the global economy is feeling the effects of this data explosion. This data deluge exhibits not just volume and velocity but also variability and diversity of structure, completeness, and domain [37]. The impact of enormous new sources of data extends to many areas of society, far beyond business, industry, government, science, sports, advertising and public health, with no area being untouched. There is no doubt that the importance of data-intensive computing has been increasing and it becomes the foremost research field in industry and academic communities. As a result, applications based on data-intensive services have become the most challenging type of applications in SOA. Also, data-intensive service composition has become an interesting topic in academia and industry.

A set of operations is often necessary to provide an appropriate solution to complex problems. For example, attempting to understand the regional scale impacts of fires, or the long-range transportation of pollutants on air quality, or even the implications for climate, scientists need to combine data from aircraft and satellites. When using Web services technologies to solve scientific problems, solutions to such complex problems can be designed using workflows of various data-intensive services. A composition of a set of services as a composite service can be re-used by other users. Service composition is an active research area in service computing [3]. If no single service can satisfy users' requests, it is necessary to combine existing services into a new one in order to fulfill users' requirements. During the process of service

composition, supposedly, there are many candidate services with the same functionality but different quality of service (QoS) attributes. How to choose the right services to compose in order to get a optimal solution for the task is a very emerging research problem. On the other hand, the service-based strategy provides maximal flexibility when designing data-intensive applications. Cloud computing provides on-demand data and services to users in a pay-as-you-go model. Mass data sets have to be exchanged between several services. These data sets may be replicated in different data centers. As data plays the dominant role in execution of data-intensive applications, the movement of mass data influences the performance of the whole application process. Especially, the access cost of each data replica on one data center is different from that for other data centers [24], and the price of service relates to the amount of data transferred. It is desirable to find the cost minimized service composition solution in service computing. Based on above motivational discussion, to deploy and execute data-intensive service applications, it is necessary to consider not only the selection of a set of atomic services but also the appropriate data centers for accessing data replicas. In many cases in SOC environments, certain atomic services will be more data-oriented by optimizing locations of data and data replicas.

This paper presents a cost minimizing composition model for data-intensive services. The paper is organized as follows: Section II investigates the related work. Section III presents the problem description and model. Then how bio-inspired algorithms can benefit data-intensive service composition will be investigated in Section IV. Finally, some discussions in Section V and Section VI concludes this study and proposes the future work.

II. RELATED WORK

Various approaches have been proposed to solve service composition problems. The literature has presented two service selection approaches, namely, local optimization and global optimization. In local optimization [1], [2], [31], for each abstract service, the best candidate service in its service candidate set will be selected to execute it. This approach can guarantee only local QoS constraints, which assures the performance of each individual task. For example, the limit of the cost of one operation. The local optimization can not satisfy the QoS constraints and preferences at a global level. For example, the whole response time of the composite service is constrained. The global optimization [2], [8], [11], [31] can overcome the limitations of local optimization. Regarding the computational complexity of service composition, there are two kinds of composite methods based on QoS attributes [33]. One is the exhaustive method which calculates all of the candidate paths to choose the best plan, but this leads to poor scalability and heavy calculation cost. The other is the approximation algorithm which chooses an ideal composition plan which is as close as possible to the best one.

Because of the different QoS attributes of services, plus the dependency constraint and the conflict constraint among services, the service composition problem has been regarded in related work as a multiple-choice knapsack problem [13], a constrained combinatorial optimization problem [23], a weighted directed acyclic graph problem [29], a knapsack

problem [32], and a multi-dimensional and/or multi-objective problem [34]. The paper [30] considered web service composition problem as an optimization problem for the first time. It presented linear programming techniques to select services for composition. The method only considered constraint factors of control structure but did not consider the relationship between concrete services [7]. In addition, service composition problem is also regarded as a workflow design problem [26] and AI planning problem [12], [19], [21], [28], [36].

The paper [18] presented cost control in service composition in order to minimize the cost of service provision whilst maximizing the profits. Other studies [6], [14], [16], [25], [27] also investigated the cost minimum service composition problem from different points of view. Data-intensive applications often deal with large distributed data. The size and the number of data sets increase over time making the costs of communication, storage, and transfer of data even become higher. Cost of service is even related to expenses derived from energy consumption, management, and monitoring on resources. Service providers and service composers always try to get a balance between cost and profit whilst service consumers prefer to get high value service with low cost. Many studies are developing new technology and approaches to find the economical models behind costs in the service provision field.

Although many approaches have been presented in service composition domain, most of them are QoS-based composition techniques which are not well suited for data-intensive service composition. Since the data-intensive service composition has distinguished characteristics and requirements such as economics-based, transactional, constraints of pay-as-you-go data policy and so on. The authors in [15] argued that some research considered the effect of data intensity to service composition [5], [35], but they overlooked the communication cost of mass data transfer and its effects on the whole performance of business processes with different structures. As the traditional quality-based service composition approaches are not suited to solve data-intensive service composition problems, it is necessary to find novel techniques which provide service composition solution with minimized cost.

III. COST MINIMIZING SERVICE COMPOSITION MODEL

In this section, some basic definitions and concepts are explained. Then the problem description and the cost minimizing service composition model are given.

A. Basic definitions

Definition 3.1 (Abstract services): Abstract services represent the functional descriptions of the sought services. These descriptions are extracted from the expression of a user's task which is combined with the user's preferences [10]. Abstract services have standard service interfaces across different service providers. An abstract service typically corresponds to a workflow task. In this paper, we use $\{AS_1, AS_2, \dots, AS_n\}$ to represent a set of abstract services.

Definition 3.2 (Concrete services): Concrete services, also called service instances, are published by service providers. They represent the existing services which are available for possible invocation of their functionality and capabilities [10].

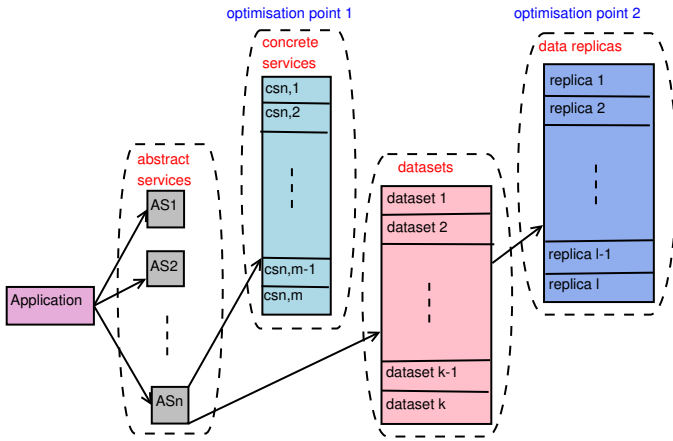


Fig. 1. Service selection and optimization graph

Concrete services may have a similar functionality (for example, checking available tickets, making reservations, planning meetings), but they differ from each other by non-functional qualities, such as response time, service cost, and reliability.

Definition 3.3 (Candidate Relationship): While the function of several concrete services $cs_{i,1}, \dots, cs_{i,m}$ is consistent with the functional description of an abstract service AS_i , we state that concrete services $cs_{i,1}, \dots, cs_{i,m}$ and abstract service AS_i have a candidate relationship. We also deem that $cs_i = \{cs_{i,1}, cs_{i,2}, \dots, cs_{i,m}\}$ is the service candidate set of AS_i .

B. Problem description

Due to the distributed and dynamic nature of data-intensive services, it is important to be able to cope with the changes of the status of services when performing data-intensive service composition. Therefore, optimization is performed at three points in time during the lifetime of a data-intensive service composition. The former two optimization points is described in Fig. 1.

The first optimization phase occurs when the system uses late-binding mechanism to choose concrete services. Since there are many candidate services with the similar functionality but different QoS attributes mapping to an abstract service, a mechanism is needed to select a set of candidate services. This phase is referred as dynamic concrete service selection. The second optimization phase is dynamic data replica selection. Since there can be multiple copies of each data set, the services need to find the best available data replica. The third optimization phase will be dynamic solution optimization. This phase will calculate the score of each solution, then service composers can examine the trade-offs among different composite services and decide one based on their preferences and priorities. If a feasible solution for the service composition problem does not exist, negotiation is performed in this phase in order to determine new QoS offers for services.

C. Cost minimizing model for data-intensive service composition

A data-intensive service composition environment can be considered to consist of a set of z data servers, $D =$

$\{d_1, d_2, \dots, d_z\}$. For an application composed of n tasks (abstract services), denoted as $AS = \{AS_1, AS_2, \dots, AS_n\}$, it is assumed that each abstract service has m concrete services. Their corresponding service candidate sets denoted as $\{cs_1, cs_2, \dots, cs_n\}$, where $cs_i = \{cs_{i,1}, cs_{i,2}, \dots, cs_{i,m}\}$, $i \in \{1, \dots, n\}$. Each task, $AS_i \in AS$, requires a set of k data sets, denoted by DT^i , that are distributed on a subset of D . Each data set $dt \in DT^i$ has l data replicas. Specifically, for a data set $dt \in DT^i$, $D^{dt} \subseteq D$ is the set of data servers (each denoted by d_{dt}) on which dt is replicated and from which it is available. Also, a data server can serve multiple data sets at a time.

A binary decision variable x_{ij} is the constraint used to represent only one concrete service is selected to replace each abstract service during the process of service composition, where x_{ij} is set to 1 if $cs_{i,j}$ is selected to replace abstract service AS_i and 0 otherwise. That is to say, each task has m service providers and only one service provider is chosen by the service composer.

Consider a data-intensive service $cs_{i,j}$ on site y has been chosen to replace AS_i , which is connected by links of different bandwidths with all the data servers. For each data set $dt \in DT^i$, the time to transfer it from d_{dt} to y is denoted by $T_t(dt, d_{dt}, y)$. The estimated transfer time for the task, $T_t(AS_i)$, is the maximum value of time for transferring all the data sets required by the task. If a local copy of data set exists, the transfer time is assumed to be zero. In this case, the estimated execution time for the task, $T_{et}(AS_i)$, can be given by (1).

$$\begin{aligned} T_{et}(AS_i) &= T_{rp}(AS_i) + T_t(AS_i) \\ T_t(AS_i) &= \max_{dt \in DT^i} (T_t(dt, d_{dt}, y)) \\ T_t(dt, d_{dt}, y) &= Rt_{dt} + size(dt)/bw(d_{dt}, y) \end{aligned} \quad (1)$$

where $T_{rp}(AS_i)$ is the response time of the concrete service which used to replace AS_i ; Rt_{dt} is the time span from requesting dt to getting the first byte of dt ; $bw(d_{dt}, y)$ is the transfer rate between data server d_{dt} and y ; $size(dt)$ is the size of dt ; $size(dt)/bw(d_{dt}, y)$ denotes the practical transfer time.

Let's assume all data sets required by each task will be transferred parallel. No data set will be stored during the execution of a service. Therefore, the only data needs to be stored are the input data sets. For example, if a service has two input data sets, the earlier arrived data set needs to be stored temporarily when waiting the second input data set to arrive.

In addition, the data access cost may be pay-as-you-go or pay-per-use or subscription based, as data providers may apply different price policies to different service providers. These policies can also be negotiated prior or after the service composition as far as there are competitions between data/service providers. While one data provider and one service provider agree on a policy for a certain contract period, the service provider may also adjust how to make and distribute data replicas, in order to minimize the overall cost for providing services to service composers and service requesters, as discussed below. For each abstract service AS_i , the data access cost is given by $ac(dt)$. The cost for the task, $Cost(AS_i)$, can

be described by (2).

$$\begin{aligned}
Cost(AS_i) &= C_{vi}(AS_i) + C_{tr}(AS_i) + C_{sr}(AS_i) \\
C_{vi}(AS_i) &= \sum_{dt \in DT^i} ac(dt) \\
C_{tr}(AS_i) &= \sum_{dt \in DT^i} size(dt) * tcost
\end{aligned} \tag{2}$$

where $C_{vi}(AS_i)$ and $C_{tr}(AS_i)$ are the access cost and transfer cost of all data sets required by AS_i respectively; $C_{sr}(AS_i)$ is the cost of the concrete service which used to replace AS_i ; $tcost$ is the cost per unit of transferring data for a link.

Π denotes a feasible service composition solution. Let $TCost(\Pi)$ be the total cost of Π , which given by (3).

$$\begin{aligned}
TCost(\Pi) &= Cost(\Pi) + C_t(\Pi) \\
Cost(\Pi) &= \sum_{i=1}^n (C_{vi}(AS_i) + C_{tr}(AS_i) + C_{sr}(AS_i)) \\
C_t(\Pi) &= \sum_{i=1}^n \mu T_{et}(AS_i)
\end{aligned} \tag{3}$$

where constant μ implies the relative importance of the service composition solution time. Therefore $C_t(\Pi)$ can be referred to as the relative cost of the execution time of the solution. Then the cost minimization service composition problem in this research is formulated as (4). Minimize the total cost $TCost(\Pi)$:

$$\begin{aligned}
TCost(\Pi) &= \sum_{i=1}^n \left(\sum_{dt \in DT^i} (ac(dt) + size(dt) * tcost) \right. \\
&\quad \left. + C_{sr}(AS_i) + \mu (T_{rp}(AS_i) \right. \\
&\quad \left. + \max_{dt \in DT^i} (Rt_{dt} + size(dt)/bw(d_{dt}, y)) \right)
\end{aligned} \tag{4}$$

Subject to:

$$AS_i = \sum_{j=1}^m (cs_{i,j} * x_{ij}); \tag{5}$$

$$\sum_{j=1}^m x_{ij} = 1, x_{ij} \in \{0, 1\}, i \in \{1, \dots, n\}; \tag{6}$$

$$ac(dt) \geq 0; \tag{7}$$

$$size(dt) > 0; \tag{8}$$

$$tcost \geq 0; \tag{9}$$

$$\mu \geq 0; \tag{10}$$

$$Rt_{dt} \geq 0; \tag{11}$$

$$bw(d_{dt}, y) \geq 0. \tag{12}$$

Constraints family (5), (6) guarantee that every task is associated to exactly one service. Constraints (7), (9), (11), (12) are set as zero if a local copy of the data set exists.

IV. BIO-INSPIRED ALGORITHMS-BASED SERVICE COMPOSITION

Bio-inspired algorithms show some advantages to solve data-intensive service composition problems. Biological systems present fascinating features such as autonomy, scalability,

adaptability, and robustness. They are autonomous entities and often self-organize without a central controller. Typical life cycles and behaviors of biological organisms include: birth and death, migration, and replication (or division) of a group of organisms [4]. The biological environment provides a medium that allows biological organisms to interact and mobilize. For example, ants release chemical signals creating chemical gradients in the environment, letting other ants sense the chemical and follow the path to food discovered by other ants. The paper [17] pointed out four key attributes of biological systems which can be applied to biological-inspired systems design: a large number of redundant components, local interactions and collective behavior, stochastic or probabilistic nature, and feedback-based control. A biological system is composed of massive numbers of redundant components. For example, an ant colony may contain millions of ants. A biological system is robust under perturbations or conditions of uncertainty because of the large number of redundant components. An individual ant is not able to find the shortest path to food source, however, a group of ants find the shortest path via interactions among individual ants through the placement of pheromones. The probabilistic nature of biological systems helps them explore large search spaces. For example, mutation in genetic evolution randomly changes genetic information to enable individuals to adapt to the environment. Ants find the shortest path via the probability of choosing different paths. Biological systems frequently use positive and negative feedback control. For example, ants use positive feedback to deposit pheromone and to recruit more ants toward a particular path.

The bio-inspired concepts and mechanisms have been successfully applied to service-oriented systems. While significant progress has been made in recent years in the area of biologically-inspired service-oriented systems research, only a few research is currently being undertaken to apply biological concepts and principles to solve issues of QoS-based service composition. This paper will briefly describe how service composition problems can be solved based on ant colony optimization algorithms.

The field of ‘ant colony algorithms’ studies models derived from the behavior of real ants which is widely used for combinatorial optimization problem. Ant colony algorithms are developed as heuristic methods to identify efficient paths through a graph and have been applied to identify optimal solutions for service composition problems. The two features of the ant colony algorithm are global feedback and local heuristic. When solving service composition problems based on the ant colony algorithm, the problem is modeled as a directed acyclic graph (DAG) with a start point S and a target point T . The start point S as the ants nest, and the target point T as the food source. Thus, the problem is transformed from selecting optimal set of services for composition into selection of the optimal path in the graph. Fig. 2 presents a service selection graph. $cs_{i,j}$ represents the j^{th} concrete service in the corresponding service candidate set cs_i for task i , data sets are the data that concrete services access, and there can be multiple copies of each data set.

The details of the simple ant colony optimization (ACO) algorithm which adapts the real ants’ behavior to the solution of optimal path problems on DAG are presented in the following. Given a DAG $G = \langle V, E, D \rangle$, two nodes

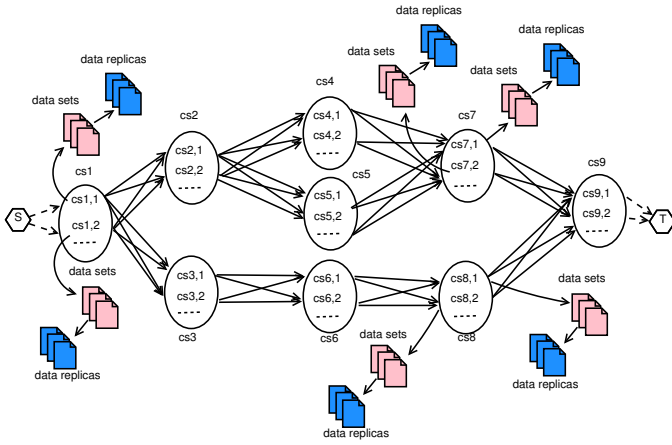


Fig. 2. Services selection graph

$i, j \in V$ are neighbors if there exists an arc $(i, j) \in E$. Each node in G represents one task corresponding to an abstract service. Each abstract service may be implemented by a set of concrete services, which make use of certain data sets, denoted as D . Ants need to find the best set of concrete services which can complete the whole composite service task altogether while also have the best QoS and the least cost for running these services. In data-intensive applications, the major cost are related to data access, data movements (for replica) and data communications. τ_{ij} is the pheromone density on the edge (i, j) . At the beginning of the search process, a constant amount of pheromone is assigned to all the arcs, namely, $\tau_{ij} = C$ (C is a constant, $\forall (i, j) \in E$). In the process of searching, when located at a node i , each ant k uses the pheromone density τ_{ij} and the heuristic information η_{ij} on the path to compute the probability of choosing j as next node:

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta}, & \text{if } j \in N_i^k; \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

where p_{ij}^k represents the probability from point i to point j of ant k , N_i^k is the neighborhood of ant k when in node i , α is a parameter to control the influence of τ_{ij} , β is a parameter to control the influence of η_{ij} . The neighborhood of a node i contains all the nodes directly connected to node i in the graph G , except for the predecessor of node i . Each ant changes the pheromone value τ_{ij} according to (14).

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \Delta\tau^k, \forall (i, j) \in E. \quad (14)$$

where ρ is the pheromone evaporation coefficient, $\Delta\tau^k$ is the amount of pheromone deposited by ant k .

Once the representation and the pheromone as well as heuristic information are defined, the basic operational flow in ACO includes four steps. The first step is to initialize all parameters. The second step is to generate solution from each ant's walk. The third step is to update the pheromone intensities. Then step two and step three are repeated until a termination condition has been satisfied. We have developed a new ACO algorithm based on ant colony system for data-intensive service composition. Also, we have designed a case study to validate and evaluate the proposed ACO algorithm.

Due to the article length restrictions, we cannot present the algorithm and the case study which has been used to test our algorithm.

V. DISCUSSIONS

A number of works have been done to solve service composition problems. QoS-aware optimal service composition and dynamic service composition are two challenging research areas. Since services are in a changing environment and the status of services are also dynamically changed, it requires more adaptation mechanisms to solve service composition problems. Bio-inspired optimization algorithms are proposed to solve service composition problem due to the simplicity of the algorithms and fast speed of convergence to optimal or near-optimal solutions. Current mechanisms for service composition based on bio-inspired algorithms have been used in some non data-intensive distributed scenarios. The paper [20] pointed out that non-evolutionary algorithms was more efficient for small scale environment or locations that business processes were simple or sets of candidates were small, while in environments that the business processes were complex and service candidates were distributed, using methods based on evolutionary algorithms could achieve best solution in a reasonable time.

Since services are in a dynamical and distributed environment, the size and number of data sets are increased over time. Further, new paradigms like Everything-as-a-Service(XaaS) are being explored, and the real strategic value of the data can determine what will happen and what can be discovered in the future, all of these make the service composition for data-intensive applications become more challenging. The economics of data-intensive service composition benefits for both users and service providers. The location, resource utilization and availability of data-intensive services are dynamically adjusted. It needs self-optimizing and self-organizing mechanism to govern these data-intensive services and allow these services to autonomously adapt to dynamic environments. The biological principles such as decentralization, evolution, emergence, and symbiosis can to achieve this goal. Bio-inspired algorithms like ACO are population-based methods with several advantages. They are meta-heuristic approaches to iteratively find near-optimal solutions in large search spaces but without data pre-classification. They can be run in distributed mode and also can deal with different types of data, such as numerical data as well as symbolic and textual data. They have the ability to solve service composition problems for data-intensive applications.

VI. CONCLUSION AND FUTURE WORK

This paper proposes a cost minimization model for data-intensive services composition. After a number of research on service composition are reviewed, the problem description is given. Biological systems are self-optimization and self-organization systems. The bio-inspired concepts and mechanisms have been successfully applied to service-oriented systems. Efforts towards how to solve data-intensive service composition problems with ant colony optimization algorithms is made in this paper. The design of the algorithms is completed and the simulation experiments are under way in our research agenda.

REFERENCES

- [1] D. Ardagna, and B. Pernici, "Global and local QoS guarantee in Web service selection", *Business process management workshops*, LNCS, Vol. 3812, pp. 32-46, 2006.
- [2] D. Ardagna, and B. Pernici, "Adaptive Service Composition in Flexible Processes", *IEEE Transactions on Software Engineering*, Vol. 33, No. 6, pp. 369-384, 2007.
- [3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing", EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28, 2009.
- [4] S. Balasubramaniam, D. Botvich, R. Carroll, J. Mineraud, T. Nakano, T. Suda, and W. Donnelly, "Biologically inspired future service environment", *Computer Networks*, Vol. 55, No. 15, pp. 3423-3440, 2011.
- [5] A. Bucchiarone, and L. Presti, "QoS composition of services for data-intensive application", *International Conference on Internet and Web Applications and Services (ICIW '07)*, pp. 46, 2007.
- [6] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", *Software: Practice and Experience*, Vol. 41, No. 1, pp. 23-50, 2011.
- [7] A. de Campos, A. T. R. Pozo, S. R. Vergilio, and T. Savegnago, "Many-objective evolutionary algorithms in the composition of web services", *2010 Eleventh Brazilian Symposium on Neural Networks*, pp. 152-157, 2010.
- [8] G. Canfora, M. Penta, R. Esposito, and M. L. Villani, "An Approach for QoS-aware Service Composition based on Genetic Algorithms", *GECCO '05 Proceedings of the 2005 conference on Genetic and evolutionary computation*, pp. 1069-1075, 2005.
- [9] S. Dustdar, and M. P. Papazoglou, "Services and service composition- An introduction", *IT-Information Technology*, Vol. 50, No. 2, pp. 86-92, 2008.
- [10] A. Hock-koon, and M. Oussalah, "Many to many service discovery: a first approach", *Proceedings of the 4th European conference on Software architecture ECSA'10*, pp. 449-456, 2010.
- [11] M. C. Jaeger, and G. Muehl, "QoS-based Selection of Services: The Implementation of a Genetic Algorithm", *Communication in Distributed Systems (KiVS)*, ITG-GI Conference, pp. 1-12, 2007.
- [12] M. Kuzu, and N. K. Cicekli, "Dynamic planning approach to automated web service composition", *Applied Intelligence*, Vol. 36, No. 1, pp. 1-28, 2010.
- [13] J. Lee, "Matching Algorithms for Composing Business Process Solutions with Web Services", *Proceedings of the 4th International Conference on ECommerce and Web Technologies (ECWEB03)*, pp. 393-402, Prague, Czechoslovakia, 2003.
- [14] P. Leitner, W. Hummer, and S. Dustdar, "Cost-Based Optimization of Service Compositions", *IEEE Transactions on Services Computing*, Technical Report TUV-1841-2011-1, Vienna University of Technology, Austria, 2011.
- [15] Y. Li, and C. Lin, "QoS-aware service composition for workflow-based data-intensive applications", *IEEE International Conference on Web Services*, pp. 452-459, 2011.
- [16] S. Y. Lin, G. T. Lin, K. M. Chao, and C. C. Lo, "A Cost-Effective Planning Graph Approach for Large-Scale Web Service Composition", *Mathematical Problems in Engineering*, Article ID. 783476, 21 pages, 2012.
- [17] T. Nakano, "Biologically inspired network systems: A review and future prospects", *IEEE Transactions on systems, man, and cybernetics-Part C: Applications and reviews*, Vol. 41, No. 5, 2011.
- [18] J. Niemoller, R. Quinet, R. Levenshteyn, and I. Fikouras, "Cost Control in Service Composition Environments", *Proceedings of The Second International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST '08)*, pp. 25-32, 2008.
- [19] J. Peer, "Web Service Composition as AI Planning—a Survey", University of St.Gallen, St. Gallen, Switzerland, 2005.
- [20] E. Pejman, Y. Rastegari, P. M. Esfahani, and A. Salajegheh, "Web service composition methods: A survey", *Proceedings of the International MultiConference of Engineers and Computer Scientists, IMECS 2012*, Vol. 1, 2012.
- [21] L. Qiu, L. Chang, F. Lin, and Z. Shi, "Context optimization of AI planning for semantic Web services composition", *Service Oriented Computing and Applications*, Vol. 1, No. 2, pp. 117-128, 2007.
- [22] V. Stantchev, "Addressing Dependability throughout the SOA Life Cycle", *IEEE Transaction on Services Computing*, Vol. 4, No. 2, pp. 85-95, 2011.
- [23] M. Tang, and L. Ai, "A hybrid genetic algorithm for the optimal constrained web service selection problem in web service composition", *Proceeding of the 2010 World Congress on Computational Intelligence*, pp. 268-275, 2010.
- [24] S. Venugopal, and R. Buyya, "An SCP-based heuristic approach for scheduling distributed data-intensive applications on global grids", *Journal of Parallel and Distributed Computing*, Vol. 68, No. 4, pp. 471-487, 2008.
- [25] J. Wang, J. P. Wang, B. Chen, and N. Gu, "Minimum cost service composition in service overlay", *World Wide Web*, Vol. 14, Issue. 1, pp. 75-103, 2011.
- [26] S. Wang, W. Shen, and Q. Hao, "Agent based workflow ontology for dynamic business process composition", *Proceedings of 9th International Conference on Computer Supported Cooperative Work in Design (CSCWD '05)*, pp. 452-457, Coventry, UK, 2005.
- [27] X. Wang, Z. Wang, and X. Xu, "Price heuristics for highly efficient profit optimization of service composition", *IEEE International Conference on Services Computing*, pp. 378-385, 2011.
- [28] Y. Yan, and X. Zheng, "A planning graph based algorithm for semantic web service composition", *Proceeding of the IEEE Joint Conference on E-Commerce Technology (CEC '08) and Enterprise Computing, E-Commerce and E-Services (EEE '08)*, pp. 339-342, 2008.
- [29] Z. K. Yang, C. W. Shang, Q. T. Liu, and C. L. Zhao, "A Dynamic Web Services Composition Algorithm Based on the Combination of Ant Colony Algorithm and Genetic Algorithm", *Journal of Computational Information Systems*, Vol. 6, No. 8, pp. 2617-2622, 2010.
- [30] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, and Q. Z. Sheng, "Quality Driven Web Services Composition", *Proceedings of the 12th international conference on World Wide Web (WWW)*, Budapest, Hungary, ACM Press, May 2003.
- [31] L. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "QoS-aware middleware for Web services composition", *IEEE Transactions on Software Engineering*, Vol. 30, No. 5, pp. 311-327, May 2004.
- [32] T. Yu, Y. Zhang, and K. J. Lin, "Efficient Algorithms for Web Services Selection with End-to-End QoS Constraints", *ACM Transactions on the Web*, ISSN 1559-1131, Vol. 1, No. 1, 2007.
- [33] C. W. Zhang, "Adaptive genetic algorithm for QoS-aware service selection", *IEEE Workshops of International Conference on Advanced Information Networking and Applications*, pp. 273-278, 2011.
- [34] W. Zhang, C. K. Chang, T. M. Feng, and H. Y. Jiang, "QoS-Based Dynamic Web Service Composition with Ant Colony Optimization", *IEEE 34th Annual Computer Software and Applications Conference (COMPSAC)*, pp. 493-502, 2010.
- [35] Y. Zhang, X. Zhou, and Y. Gao, "Optimizing the data intensive mediator-based web services composition", *Frontiers of WWW Research and Development - APWeb*, Springer, LNCS Vol. 3841, pp. 708-713, 2006.
- [36] X. Zheng, and Y. Yan, "An efficient syntactic web service composition algorithm based on the planning graph model", *Proceedings of the IEEE International Conference on Web Services (ICWS '08)*, pp. 691-699, 2008.
- [37] Workshop on massive data analytics on scalable systems, at <http://ceng.usc.edu/~simmhan/DataMASS2012/>, accessed on Jan. 1st, 2013.