Knowl Inf Syst (2014) 41:423–447
DOI 10.1007/s10115-014-0755-x

REGULAR PAPER

Security-aware intermediate data placement strategy in scientific cloud workflows

Wei Liu · Su Peng · Wei Du · Wei Wang · Guo Sun Zeng

Received: 6 May 2013 / Revised: 29 December 2013 / Accepted: 13 May 2014 /
Published online: 3 June 2014

Abstract Massive computation power and storage capacity of cloud computing systems allow scientists to deploy data-intensive applications without the infrastructure investment, where large application datasets can be stored in the cloud. Based on the pay-as-you-go model, data placement strategies have been developed to cost-effectively store large volumes of generated datasets in the scientific cloud workflows. As promising as it is, this paradigm also introduces many new challenges for data security when the users outsource sensitive data for sharing on the cloud servers, which are not within the same trusted domain as the data owners. This challenge is further complicated by the security constraints on the potential sensitive data for the scientific workflows in the cloud. To effectively address this problem, we propose a security-aware intermediate data placement strategy. First, we build a security overhead model to reasonably measure the security overheads incurred by the sensitive data. Second, we develop a data placement strategy to dynamically place the intermediate data for the scientific workflows. Finally, our experimental results show that our strategy can effectively improve the intermediate data security while ensuring the data transfer time during the execution of scientific workflows.

Keywords Cloud computing · Data security · Data placement · Scientific workflow

W. Liu · S. Peng · W. Du (✉)
College of Computer Science and Technology, Wuhan University of Technology, Wuhan 430063, China
e-mail: whutduwei@whut.edu.cn

W. Liu · W. Wang · G. S. Zeng
Department of Computer Science and Technology, Tongji University, Shanghai 200092, China

W. Liu · W. Du · W. Wang · G. S. Zeng
Key Laboratory of Embedded System and Service Computing Ministry of Education, Tongji University, Shanghai 200092, China

W. Liu
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210046, China

W. Liu · W. Du
State Key Lab of Software Engineering, Wuhan University, Wuhan 430072, China

1 Introduction

Modern e-Science infrastructure allows researchers to address new large-scale problems whose solution was not available before, e.g. genome, climate, global warming [1,2]. E-Science typically produces a large amount of data that must be supported by a new type of e-Infrastructure, which can store, distribute, process, preserve, and curate these data. We refer to this new infrastructure as the Scientific Data e-Infrastructure(SDI) [1,3]. In e-Science, the scientific data are complex multifaceted objects with complex internal relations: they become an infrastructure of their own, which must be supported by corresponding physical or logical infrastructures to store, access and manage these data [4–7].

Typically, scientists must analyse terabytes and even petabytes of scientific data that are collected from the existing input data, the intermediate data generated in the process, and the resulting data [8]. As reported by Szalay et al. in [9], the total amount of scientific data will be doubly increased in the next decade. The production of scientific datasets involves large numbers of computationally intensive tasks, e.g. the scientific workflows [10]. These generated datasets contain important intermediate or final results of the computation and must be stored as valuable resources [11]. Currently, most popular e-Science applications are deployed in grid systems, which can offer high computational capacity and massive storage. However, building a grid system is extremely expensive and even those existing grid systems are devoted to their own specific applications, thus failing to satisfy the more general scenarios [12].

Recently, the emergence of cloud computing technologies offers a new method to develop scientific workflow systems [13]. This method has been successfully employed in many areas [14]. Foster et al. make a thorough comparison between grid computing and cloud computing [15]. As a conclusion of their work, cloud computing has several advantages over grid computing. First, cloud computing can provide the necessary high performance and massive storage for data-intensive applications such as the scientific workflows like the grid system, but with a relatively low infrastructure construction cost [16]. Then, it creates a new paradigm to gather worldwide scientists for collaborations [13]. The scientists can upload their data and set up their applications on the scientific workflow systems through the Internet [17].

With the development of cloud computing, the widely used scientific workflow systems can be adapted into cloud computing for even more promising applications. Cloud computing offers customers a more flexible way to obtain computation and storage resources on demand [18,19]. Migrating organisational services, data and application in cloud is an important strategic decision for organisations due to the large number of benefits introduced by the usage of cloud computing, such as cost reduction and on-demand resources [20]. However, there are challenges and risks for cloud adaptation related to data security when dynamically placing data among the globally distributed data centres, while minimising the user-perceived latency [20,21]. Because the data centres are connected with limited bandwidth in cloud, some scientific data may be insecure when they are transmitted from one data centre to another. In addition, this challenge is further complicated by the security constraints on those potentially sensitive data. What is more, new threats may also arise since the resources are shared with others [22]. In this scenario, one's sensitive data are more likely to be exposed to others. Moreover, data information stored in data centres may even be leaked out by some malicious cloud providers. The information loss of sensitive data may cause serious loss of interest, property and personal safety. Consequently, the problem of data security has been widely recognised as one of major technical obstacles when deploying data into openly distributed systems, e.g. clouds [23,24].

In this paper, we propose a security-aware intermediate data placement strategy for scientific workflows in cloud. Disregarding the placement of the input data, which is decided by the users, we mainly focus on the placement of the intermediate data. In our strategy, we attempt to ensure data security based on three aspects: data confidentiality, data integrity and authentication access. A security model is used to quantitatively measure the security services provided by the data centres. Then, we use an ACO (ant colony optimisation)-based algorithm to dynamically select the appropriate data centres for the intermediate data to improve the data security. This strategy is notably effective at improving data security during the execution of the scientific workflows.

The remainder of the paper is organised as follows. Section 2 describes the related work. Section 3 introduces the security problem of scientific workflows. Section 4 proposes the security model of scientific workflow systems in cloud. Section 5 presents the data placement strategy. Section 6 demonstrates the experimental results and the analysis. Finally, we draw our conclusions.

2 Related work

The growing trend towards cloud computing has provoked new data management systems such as Google File System [25] and Hadoop [26]. These systems act as data storage infrastructures in cloud that also provide some basic data management functions, e.g. unauthorised access, data block storage, and disaster recovery. The technologies in big data management [3,27] have become an important issue. Currently, several works have begun to concentrate on the data placement of scientific workflows. Guo et al. [28] propose a model for multi-objective data placement and use a particle swarm optimisation algorithm to optimise the time and the cost in cloud computing. To process the resources more effectively, Guo et al. [29] devise an optimal data placement strategy that minimises the processing cost and the transforming time. For the data-intensive scientific workflows, the data movement time seriously affects the efficiency of the data-intensive applications. Ma et al. [30] propose a data placement method based on the Bayesian network for data-intensive scientific workflows, which could effectively reduce the data movement time among different data centres. And Er-Dum et al. [31] propose a data placement strategy based on a heuristic genetic algorithm to reduce the data movements among the data centres while balancing the loads of data centres. Moreover, Shao-Wei et al. propose a two-stage data placement strategy and a task scheduling strategy for efficient workflow execution with data dependencies [32]. These data placement strategies mainly consider the data transfer time among data centres. However, when some sensitive data are transferred across the data centres, data security threats may arise during the transmission. Consequently, data security is a notably significant and challenging problem for data placement.

Because the resources are shared among the users in cloud, the data information is exposed to multi-users and cloud providers when we place our data into data centres. This paradigm introduces some security risks with the existence of some intensive data. Thus, the data security is widely regarded as a major barrier in cloud. Several recent works have considered data security. Xi et al. [33] prove that the compression is asymptotically lossless because the aggregated estimator deviates from the true model for data-intensive computing. Xiong et al. [21] briefly introduce the security problems in cloud, particularly when deploying data. In [34], Peng et al. propose a data placement approach that places the data on the cloud side or the client side according to the privacy requirements and adjusts the data placement according to the control flow to minimise the data transfer time.

However, most of the latest works fail to describe a strategy for solving the data security problem.

The closest studies to ours are [16] and [35]. In [16], Yuan and Yang propose a two-phase data placement strategy based on k-means clustering for scientific workflows to reduce the data movements. The strategy contains two algorithms that group the existing datasets into k data centres during the build-time stage and dynamically clusters newly generated datasets based on their dependencies to the most appropriate data centres during the runtime stage. However, the reduction of data movements does not imply that the performance improves because this strategy does not consider the size of the datasets and the bandwidths among the data centres. This problem has been well addressed. In [35], Zheng Pai and Cui Li-Zhen propose a genetic algorithm-based data placement strategy to address three problems: reducing the time cost of data movements across the data centres, addressing the data dependencies, and maintaining a relative load balancing of the data centres. This strategy adopts a genetic algorithm to obtain some solutions that can reduce the data transmission in the first stage. In the second stage, those solutions are readjusted using the data dependencies. Finally, the most appropriate strategy is selected according to the load balancing performance in the third stage. However, this strategy may fail to obtain the optimal solution. Because the final strategy is selected by different goals in each stage, some appropriate solutions may have been missed. Unfortunately, these works only focus on how to reduce the data transmission and fail to consider the data security. In fact, the security problem is more important in the network construction in cloud.

Therefore, it is necessary to deploy security services to protect the sensitive intermediate data in the scientific workflows. Because snooping, alteration, and spoofing are three common attacks in cloud computing, in this paper, we consider three security services (authentication service, integrity service, and confidentiality service) to guard against the common threats to the sensitive data. Snooping, which is an unauthorised interception of information, can be countered using confidentiality services. Alteration, which is an unauthorised change of information, can be handled using integrity services [36]. Spoofing, which is an impersonation of one entity by another, can be well handled using an authentication service [37]. With the three security services, the users can flexibly select the security service to form an integrated security protection against a diversity of threats and attacks in cloud. In this paper, first, we build a security model to measure the security overhead incurred by deploying three services for the sensitive data. Then, we propose a security-aware data placement strategy that can improve data security while guaranteeing the data transfer time.

3 Problem analysis of the data security for scientific workflows

3.1 Scientific workflow model

Similar to the literature [16], we first describe the scientific workflow applications as follows.

Definition 1 The scientific workflow applications can be expressed as a triple $w = <T, C, DS>$, where T is the set of all tasks in w, and C is the set of the control flow among the tasks. In this paper, the control flow is reflected through the data flow among the tasks, andDS is the collection of all datasets in w.

Our work mainly relates to the input and output datasets of the scientific workflows tasks. Thus, the tasks are described as in Definition 2:

Definition 2 The task is defined as $T_i = <IDS_i,\ ODS_i,\ s_{Ti}, s_{Twi}>$, where IDS_i indicates the input datasets, ODS_i refers to the output datasets, s_{Ti} represents the task security service requirements, and s_{Twi} denotes the importance of the security services. These parameters will be introduced in Sect. 4.2.

In this paper, we divide the datasets into fixed-location and flexible-location datasets according to the different locations where the data are stored. Because of the limitations and constraints on the intellectual property, equipment and processing capacity, the fixed-location datasets are stored in specific data centres. Furthermore, these fixed-location datasets will consume some storage resources because they have fixed storage locations. Thus, the data layout of the overall strategy will be affected. There will be a detailed description and analysis in Sect. 5.

3.2 Example analysis

It is known that the data centres are connected together through the Internet. The users must use computing resources and storage resources from different data centres during the execution of the scientific workflows.

From Fig. 1a, we learn that the scientific workflow includes tasks, input datasets, generated datasets and fixed-location datasets. These data are placed in DC_A and DC_B in Fig. 1b. When executing the scientific workflow, it is necessary to access other data centres to obtain the required data.

The data, particularly sensitive data, may be subjected to security threats in data centres because of the sharing feature of the cloud. As Fig. 1b shows, the sub-tasks $\{T_1, T_2\}$, input data $\{d_1, d_3\}$ and intermediate data $\{dT_1, dT_2\}$ are placed in data centre DC_A. Then, the sub-tasks $\{T_3, T_4\}$, input data d_2, fixed-location data d_{4fix} and the intermediate dT_3 are placed in data centre DC_B. For example, T_3 must call the input data $\{d_2, dT_1\}$ during the execution. However, the input datasets are located in different data centres. We must obtain dT_1 in DC_A. Furthermore, $\{d_1, d_3\}$ is also placed in DC_A. The information is likely to be leaked while acquiring dT_1. If these data are strongly sensitive data, their disclosure will cause irreparable consequences. In conclusion, data security is an important issue that should not be overlooked when running scientific workflows in the clouds.

3.3 Problem description

With the increasing demand for computing resources and storage resources, the cloud computing model emerges. However, it also incurs an increasing challenge for placing data in globally distributed data centres. When deploying a scientific workflow in cloud, the data security is seriously threatened. The security problems may arise from two sources: the cloud providers and the other multi-users.

First, this paper analyses the problem of the cloud providers who offer the security services. In Fig. 1b, for example, the data $\{d_1, d_3, dT_1, dT_2\}$ are located in DC_A. Thus, the cloud providers are certainly aware of the related information of the users' data. These data may be leaked or even sold to others by some malicious cloud providers. The information loss of some intensive data may cause loss of interest, property or personal safety. In addition, there are data for other users of the same data centre. Because the resources are shared with other users, the users' data could easily be stolen or tampered with by the multi-users, which ultimately results in the losses of the user. The details are given in Sect. 3.2.

To address this security issue, we should protect the security of the users' data. The users can request different data security services according to the degree of sensitivity of

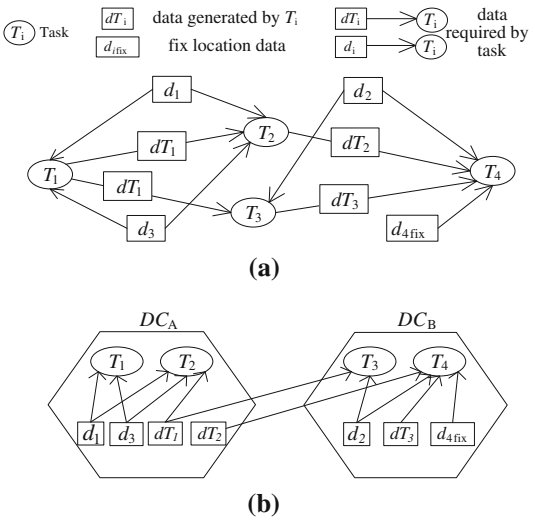


Fig. 1 Example of a scientific workflow

the data when running scientific workflows in cloud. Accordingly, the data centres mainly adopt some measures such as certification services, encrypted data storage, data recovery, security management, security logs, and audit services to provide users with data security requirements.

In addition, we must consider whether the data centre can satisfy the data security services requirements when placing the data. If the data centre cannot satisfy the requirements of the data security services, one should deploy the data in other data centres that can provide the services.

The next section will introduce the security model of the scientific workflow systems.

4 Security model of scientific workflow systems

In this section, we illustrate the security model for scientific workflow systems from service providers, service consumers and service evaluation. This model includes the security service model of data centres, security service model for task and data, the degree of data security deficiency (DDSD), the degree of task security deficiency (DTSD) and Security Deficiency Degree of Scientific Workflow (SDDSW).

4.1 Security service model for data centres

First, we introduce the security services of the data centres from the perspective of the service providers. It is known that the distributed data centres are connected together through the Internet. These data centres provide the certification service, encryption for data storage, data recovery, security management, security logging, auditing and other measures to guarantee data security [22]. Confidentiality, integrity, and authentication are the three basic services that are used to ensure data security. Thus, we use the following definition to denote the security services provided by the data centres.

Definition 3 Let the matrix $p_j = \left\{ p_j^1 , p_j^2 , p_j^3 \right\}$ represent the security service capability of data centres dc_j, where p_j^1 represents the confidential service coefficient, p_j^2 represents the

Table 1 Confidential service parameters

Encryption algorithm	Service factor
SEAL	0.08
RC4	0.14
Blowfish	0.36
Kunfu/Khafe	0.40
RC5	0.46
Rijndael	1.00
DES	0.64
IDEA	0.90

Table 2 Integrity service parameters

Hash function	Service factor
MD4	0.18
MD5	0.26
RIFDMD	0.36
RIFDMD-128	0.45
SHA-1	0.63
RIFDMD-160	0.77
TIGER	1.00

Table 3 Authentication service parameters

Authentication	Service service factor
HMAC-MD5	0.3
HMAC-SHA-1	0.6
CBC-MAC-AES	0.9

integrity service coefficient, and p_j^3 represents the authentication service coefficient. Those coefficients represent different service factors of the security services.

Each service can be implemented using several strategies, as shown in Tables 1, 2, and 3, according to the literature [36,38,39].

From these parameters in the tables, we can see that different security service factors demonstrate different capabilities of providing security services. A greater parameter corresponds to a higher level of security services. Let us take the confidential service as an example. In Table 1, both DES and IDEA can do encryption, but IDEA has a larger coefficient than DES, which implies that IDEA can better guarantee the data confidentiality than DES.

The service factors in those tables are calculated using different algorithms. For example, the encryption efficiencies of SEAL, DES and IDEA in a 90 MHz processor are 168.75, 15 and 13.5 KB/ms, respectively [36]. The IDEA algorithm has the maximum security factor and the lowest efficiency [38]. Therefore, the efficiency of the IDEA encryption algorithm should be divided by the encryption efficiency of SEAL, DES and IDEA. Then, we will obtain the corresponding values of the confidential security service parameters. The security service factors are only reference values.

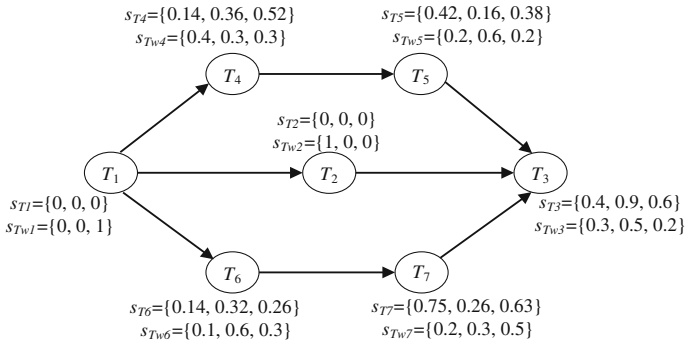


Fig. 2 Security service requirement factors of tasks

4.2 Security service requirements of tasks

We describe the security service requirement factors with respect to the consumers. When deploying the scientific workflows in cloud, different security service requirements are required by the tasks. We first introduce the security service requirements for a task in the following definition.

Definition 4 The security service requirement for a task is represented as $s_{Ti} = \{s^1_{Ti}, s^2_{Ti}, s^3_{Ti}\}$, where s^1_{Ti} denotes the confidential service factor, s^2_{Ti} denotes the integrity service factor, and s^3_{Ti} denotes the authentication service factor.

Different factors of the task security service requirements show different weights of the three services. If one security service s^k_{Twi} is more important, the service factor will be larger. The three components satisfy the formula $\sum_{k=1}^{3} s^k_{Twi} = 1$.

The task security model for a scientific workflow is shown in Fig. 2.

As shown in Fig. 2, the security service requirements of each task are indicated by s_{Ti} and s_{Twi}, where s_{Ti} represents the security service demands for the task, and s_{Twi} denotes the importance of the security services. Let us take T_5 as an example: 0.42, 0.16 and 0.38 are the coefficients of the confidential service, the integrity service and the authentication service, respectively. Additionally, 0.2, 0.6 and 0.2 represent the weights of the task security services, and their sum equals 1.

4.3 Security service requirements of data

According to Definition 2, the security service requirements of the generated data are determined by the task that produces it. For example, we know that the security service requirements of T_5 are $s_{T5} = \{0.42, 0.16, 0.38\}$ and $s_{Tw5} = \{0.2, 0.6, 0.2\}$. Thus, the security service requirements of the data generated by T_5 are identical to that of T_5. Then, the security service requirements for the intermediate data are described as follows.

Definition 5 The data security service requirements are represented by $s_i = \{s^1_i, s^2_i, s^3_i\}$, where s^1_i, s^2_i, and s^3_i represent the confidential service coefficient, the integrity service coefficient and the authentication service coefficient of the data, respectively.

Similarly, the importance of the data security services s_{wi} is defined as s_{Twi}. The security service factors satisfy the formula $\sum_{k=1}^{3} s^k_{wi} = 1$.

Table 4 Security service factors of data centres

DC name	Confidential service	Integrity service	Authentication service
DC_A	0.64	0.36	0.3
DC_B	0.36	0.45	0.9
DC_C	0.9	0.77	0.6
DC_D	0.46	1.00	0.6
DC_E	1.00	0.63	0.9

In this paper, we mainly consider the security service demands for intermediate data. The requirements of the data security service are used to describe the required security services when moving or storing the data. We must decide whether the distributed data centres can provide security services when we schedule tasks. If not, the tasks cannot be scheduled to these data centres.

Table 4 describes the security services that are provided by several data centres in cloud.

Table 4 illustrates the highest level of security services that the data centres can provide. For example, the maximum confidential service factor that DC_A can offer is 0.64. Hence, DC_A can provide all security service coefficients that are less than or equal to 0.64. If the confidential service coefficient that a task requires is 0.9, this data centre obviously cannot meet their requirements, and the other data centres should be selected for this task. The same principle applies for the integrity service and the authentication service.

4.4 Degree of data security deficiency

To evaluate the security services, we propose the *DDSD*. The *DDSD* is regarded as a quantitative indicator to describe the deficiency degree for data security.

In this paper, we define the *DDSD* as the ratio of the *VDSD* (value of data security deficiency) and the *BVDSD* (base value of data security deficiency). We know that *DSD* (degree of security deficiency) has been purposed in architecture [38], but the size of the datasets can affect the transfer time because the data centres are connected with a limited bandwidth. The transmission time increases if the size of the datasets grows, and a security problem is more likely to happen. Thus, the *VDSD is* defined in formula (1). In this formula, d_{si} represents the size of d_i. When the security services p_j provided by the data centre can satisfy the data security service requirement s_i, the value of $g(s_i^k, p_j^k)$ is 0. Otherwise, $g(s_i^k, p_j^k)$ is the absolute value for the difference between s_i^k and p_j^k.

$$\begin{aligned} VDSD(d_i) &= DSD(s_i) * d_{si} \\ &= \sum_{k=1}^{3} s_{wi}^k * g(s_i^k, p_j^k) * d_{si}, 0 \le s_{wi}^k \le 1 \\ \sum_{k=1}^{3} s_{wi}^k &= 1\ ,\ g(s_i^k, p_j^k) = \begin{cases} 0, & \text{if } s_i^k \le p_j^k \\ s_i^k - p_j^k, & \text{otherwise} \end{cases} \end{aligned} \tag{1}$$

BVDSD denotes the value of the data security deficiency when $p_j = \{0, 0, 0\}$. This relationship can be described using the following formula.

$$\begin{aligned} BVDSD(d_i) &= VDSD(d_i)|_{p_j=\{0,0,0\}} \\ &= DSD(s_i) * d_{si}|_{p_j=\{0,0,0\}} \end{aligned}$$

$$= \sum_{k=1}^{3} s_{wi}^{k} * s_{i}^{k} * d_{si}$$

$$0 \le s_{wi}^{k} \le 1, \sum_{k=1}^{3} s_{wi}^{k} = 1 \tag{2}$$

Then, the *DDSD* is shown in formula (3), where the value of $DDSD(d_i)$ ranges from 0 to 1. If that value equals 0, the data security service requirements can be completely satisfied. Otherwise, a smaller value corresponds to more secure data.

$$\begin{aligned} DDSD(d_i) &= \frac{VDSD(d_i)}{BVDSD(d_i)} \\ &= \frac{VDSD(d_i)}{V} DSD(d_i)|_{p_j=\{0,0,0\}} \\ &= \frac{\sum_{k=1}^{3} s_{wi}^{k} * g(s_i^k, p_j^k) * d_{si}}{\sum_{k=1}^{3} s_{wi}^{k} * s_i^k * d_{si}}, \ 0 \le s_{wi}^{k} \le 1 \\ \sum_{k=1}^{3} s_{wi}^{k} &= 1, g(s_i^k, p_j^k) = \begin{cases} 0, & \text{if } s_i^k \le p_j^k \\ s_i^k - p_j^k, & \text{otherwise} \end{cases} \end{aligned} \tag{3}$$

For example, in Fig. 1, we assume that the size of the generated data dT_1 is 80 GB. The security service demand is $s_i = \{0.63, 0.4, 0.6\}$, and the importance of the security services is $s_{wi} = \{0.2, 0.5, 0.3\}$. During the execution of a scientific workflow, dT_1 is moved to the other data centre DC_B, which can provide the security service $p_B = \{0.46, 0.46, 0.9\}$. Thus, we can obtain the following value of $DDSD(d_i)$.

$$\begin{aligned} DDSD(d_i) &= [0.2*(0.63 - 0.46) + 0.5*0 + 0.3*0]*80/[0.2*0.63 + 0.5*0.4 + 0.3*0.6]*80 \\ &= 2.72/40.48 \\ &= 0.067 \end{aligned}$$

4.5 Degree of task security deficiency

In this part, we obtain the *SDDSW* from the above discussion. For the task $T_i = <IDS_i, ODS_i, s_{Ti}, s_{Twi}>$ that was described in Sect. 3.1, the *DTSD* is calculated as the sum of *DDSD*s for the input and output datasets. In this paper, we only consider the security of the generated datasets during the execution. The security requirements of the intermediate datasets are described using the tasks that generate them. During the execution, the data centre should attempt to satisfy the maximum coefficient of the security services for all input datasets and the task itself.

According to the definition of $DDSD(d_i)$, the $DTSD$ is defined by formula (4).

$$\begin{aligned} DTSD(T_i) &= DDSD(IDS_i) + DDSD(ODS_i) \\ &= \sum_{i=1}^{|IDS_i|} DDSD(id_i) + \sum_{i=1}^{|ODS_i|} \sum_{l=1}^{t} DDSD(od_i) \\ where\ id_i \in IDS_i, od_i \in ODS_i, t &= \begin{cases} 1, & dc = dc(d_o) \\ 2, & dc! = dc(d_o) \end{cases} \end{aligned} \tag{4}$$

The *DDSD*s for the input and output datasets should be calculated differently. When the output datasets are at the data centre where the task runs, we can only calculate the *DDSD*s

Table 5 Security service requirements for the tasks and the generated data

Task name	T_1	T_2	T_4
Security service requirements	$s_{T1} = \{0.63, 0.4, 0.6\}$	$s_{T2} = \{0.36, 0.46, 0.7\}$	$s_{T4} = \{0.4, 0.2, 0.3\}$
Importance of security services	$s_{Tw1} = \{0.2, 0.5, 0.3\}$	$s_{Tw2} = \{0.3, 0.4, 0.3\}$	$s_{Tw4} = \{0.5, 0.2, 0.3\}$
Size of generated data	80 GB	100 GB	120 GB
Location of generated data	DC_B	DC_B	DC_B
Position of task execution	DC_A	DC_A	DC_B

Table 6 Security service capability of the data centres

DC name	DC_A	DC_B
Security service capability of data centres	$p_A = \{0.36, 0.46.0.6\}$	$p_A = \{0.46, 0.46.0.9\}$

for the output datasets. Otherwise, we must calculate the *DDSD*s for the generated datasets and the stored datasets; then, we sum them. Because the data centres where the task runs and the data are placed may be different, their *DDSD* values are not necessarily identical.

We use the scientific workflow in Fig. 1 as an example.

The security service requirements for the tasks and the generated datasets are shown in Tables 5 and 6 describes the security service that is provided by the data centres. To calculate the *DTSD*, we use T_2 as an example. From the above tables, we know that the data location and the position of the execution of T_2 are different. Thus, we must calculate the *DDSD* for the datasets that are generated and located in different data centres. During the execution, DC_A must satisfy the maximum coefficient of the security services for dT_1 and T_2.

First, we calculate the *DDSD* of the input data of T_2, which is named $DDSD(i)$.

$$\begin{aligned} DDSD(i) &= [0.2 * (0.63 - 0.36) + 0.5 * 0 + 0.3 * 0.1] * 80/[0.2 * 0.63 \\ &\quad + 0.5 * 0.46 + 0.3 * 0.7] * 80 \\ &= 0.084/0.566 \\ &= 0.15 \end{aligned}$$

Then, the *DDSD* for data generated by T_2 in DC_A is called *DDSD*(g):

$$\begin{aligned} DDSD(g) &= (0.3 * 0 + 0.4 * 0 + 0.3 * 0.1) * 100/[0.3 * 0.36 + 0.4 * 0.46 + 0.3 * 0.7] * 100 \\ &= 0.03/0.502 \\ &= 0.06 \end{aligned}$$

However, the generated data are located in another data centre DC_B, so the *DDSD* in DC_B is calculated as follows:

$$\begin{aligned} DDSD(gs) &= (0.3 * 0 + 0.4 * 0 + 0.3 * 0) * 100/[0.3 * 0.36 + 0.4 * 0.46 + 0.3 * 0.7] * 100 \\ &= 0 \\ DTSD(T_2) &= DDSD(i) + DDSD(g) + DDSD(gs) \\ &= 0.15 + 0.06 + 0 \\ &= 0.21 \end{aligned}$$

The *DTSD*s for the other tasks can be calculated using the same method.

4.6 Security deficiency degree of scientific workflow

A scientific workflow is a collection of interdependent tasks, so the definition of *SDDSW* is:

$$SDDSW(w) = \sum_{i=1}^{|T|} DTSD(T_i), T_i \in T \tag{5}$$

From (5), we know that the $SDDSW(w)$ can be calculated according to the example in Sect. 4.5.

When scheduling the tasks of a scientific workflow [40], we should attempt to guarantee the security services for each task. Thus, one of the goals of scientific workflow scheduling is to make a smaller *SDDSW*.

5 Security-aware data placement strategy

Because the architecture and the technology adopted in data centres are different, the data centres of different organisations show heterogeneous characteristics in some aspects such as security heterogeneity. In this article, this heterogeneity characteristic is manifested mainly by implementing the same security services with encryption algorithms and heterogeneous technologies. For example, some data centres use SSL to ensure the security of data transmission, whereas others use VPN. To improve data security, we fully consider this heterogeneity and the *DDSD* when placing the data. We use the following formula to describe the problem, which is a combinatorial optimisation under multi-dimensional constraints.

Formula (6) describes the minimum *SDDSW*, where d_{sid} and d_{sod} denote the size of the input and the output datasets, respectively. Formula (7) expresses the minimum data transfer time, where the number of data is n, dc_{orig} is the original data centre where d_i is located, and dc_{des} is the destination data centre of d_i. Then, the transfer time of a single dataset is described in (8), where d_{si} represents the size of d_i, and the bandwidth between dc_{orig} and dc_{des} is b. If dc_{orig} is identical to dc_{des}, the transmission time is 0. In addition, (9) indicates that the maximum available space of the data centres cannot exceed *cs*, which is the maximum storage capacity.

$$\begin{aligned} Min(SDDSW(w)) &= Min\left(\sum_{i=1}^{|T|} DTSD(T_i)\right) \\ &= Min\left(\sum_{i=1}^{|T|}\left(\sum_{i=1}^{|IDS_i|} DDSD(id_i) + \sum_{i=1}^{|ODS_i|}\sum_{l=1}^{t} DDSD(od_i)\right)\right) \\ &= Min\left(\sum_{i=1}^{|T|}\left(\sum_{i=1}^{|IDS_i|}\left(\frac{\sum_{k=1}^{3} s_{wid}^k * g(s_{id}^k, p_j^k) * d_{sid}}{\sum_{k=1}^{3} s_{wid}^k * s_{id}^k * d_{sid}}\right)\right.\right. \\ &\quad \left.\left. + \sum_{i=1}^{|ODSi|}\sum_{l=1}^{t}\left(\frac{\sum_{k=1}^{3} s_{wod}^k * g(s_{od}^k, p_m^k) * d_{sod}}{\sum_{k=1}^{3} s_{wod}^k * s_{od}^k * d_{sod}}\right)\right)\right) \end{aligned} \tag{6}$$

$$Min(TimeCost(w)) = Min\left(\sum_{i=1}^{n} TimeCost(d_i, dc_{orig}, dc_{des})\right) \tag{7}$$

$$TimeCost(d_i, dc_{orig}, dc_{des}) = d_{si}/b \tag{8}$$

$$ConstrainSpace(dc_{ava}) \leq cs \tag{9}$$

We obtain the data placement strategy from (6) and select the optimal strategy using (7). Thus, our strategy can improve the intermediate data security while guaranteeing the data transfer time.

Because the ACO algorithm is suitable to solve these combinatorial and optimisation problems, we transfer the restrictions into a heuristic and fitness function; then, we solve the data placement problem using ACO.

5.1 Initial setup

In this strategy, we set the number of ants to be the number of datasets. Assuming that the number of data centres is n and the number of datasets is m, the initial value of this pheromone is set to be 1/(m* n).

5.2 Heuristic function

We select the data centres according to formula (10) for datasets

$$P_{ij}(t) = \begin{cases} \frac{\tau_{ij}^{\alpha}(t)*\eta_{ij}^{\beta}(t)}{\sum_{l\in DC}\tau_{il}^{\alpha}(t)*\eta_{il}^{\beta}(t)} & \text{where } j \in DC \\ \tau_{ij}^{\alpha}(t)*\eta_{ij}^{\beta}(t), & \text{if } p < p_0 \end{cases} \tag{10}$$

where DC represents the collection of data centres in cloud, $P_{ij}(t)$ is the probability in the t-th iteration when the ants choose dc_j, $\tau_{ij}(t)$ is the pheromone between d_i and dc_j in the t-th iterations, η_{ij} is the value of the heuristic function $f(i, j)$, α is the importance of the residual pheromone, and β is the importance of the value for the heuristic function. P is a random value in the interval [0,1], and P_0 is the preset value in the interval [0, 1]. The two values are set to avoid an unwanted early convergence at the local optimal solution in the search process.

The heuristic function $f(i, j)$ can be represented by formula (11)

$$f(i, j) = c(d_i)*DDSD(d_i) \tag{11}$$

where $c(d_i)$ represents the number of tasks that need d_i or the usage frequency of d_i. In this paper, we assume that the frequently used datasets are important, and we prioritise the security requirements of these datasets. In (11), $DDSD(d_i)$ is the $DDSD$ of d_i in dc_j.

5.3 Pheromone update

When every ant completes a search, an evaluation function is used to evaluate the data transfer time in every iterative process and to select the optimal solution. Then, we use (12) to update the pheromone on the optimal path.

$$\tau_{ij}(t+1) = (1-\rho)*\tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{12}$$

In (12), $\tau_{ij}(t+1)$ represents the pheromone between dc_i and dc_j after the t-th iteration, and ρ is the decay parameter of the pheromone. In this experiment, $i\Delta\tau_{ij}(t)$ is the amount of pheromone increase. We adopt a pheromone with a fixed value as shown in (13).

$$\Delta\tau_{ij}(t) = 1/(m*n) \tag{13}$$

where m is the number of data centres, n is the number of datasets, and $\tau_{ij}(t)$ is the total amount of pheromone before updating.

5.4 Data placement strategy

Prerequisites: The number of data centres is m, n represents the number of datasets, and n_{fix} denotes the number of fixed datasets. Each data centre dc_j can provide the security services p_j. The security service requirement for d_i is s_i, and the importance of the security services is s_{wi}. The conditions to end the algorithm: the iterations have completed, or the optimal solution no longer changes. S_{Final} is used to store the optimal solution at this stage. In addition, we first place the data with higher requirements when we select the datasets. First, we rank the *DDSD*s of the datasets in descending order using the vector $p_j = \{0, 0, 0\}$ in the iteration process.

Procedures of the SAI Strategy

Input: D: set of datasets d_1, d_2, ...d_n

DC: set of data centres dc_1, dc_2, ...dc_m

Output: optimal solution S_{Final}:

$S_{Final}= \emptyset$; $S= \emptyset$; $n_{ant}=n$;
allocate n_{fix}
for(every d_i in D)
  for(every dc_i in DC)
    τ_{ij} = 1 / (m * n)
  end for
end for
while(iterations have completed or optimal solution no more changes) do
  for(each ant called Ant_i)
    sort d_i & add to S
    for(allocate d_i)
      calculate $P_{ij}(t)$
    end for
    add S_{Ant} to $S_{strategy}$.
  end for
  for(every solution S_i in $S_{strategy}$)
    find the S_{min} *in* $S_{strategy}$
  end for
  update τ_{ij} in S_{min}
  if(F(DC,B,W,S_{min})) < F(DC,B,W,S_{Final}))
    $S_{Final} = S_{min}$
  end if
end while
return S_{Final}

5.5 Complexity analysis

Theorem 1 *The time complexity of the data placement strategy security-aware intermediate (SAI) is* $O(m*n)+O(t*(c*m*n+c))$*, which sets the number of ants as c, the number of datasets as n, the number of data centres as m and the maximum number of iterations as t.*

Proof At the initial stage, the pheromone should be initialled among the datasets and data centres. Then, the time complexity is $O(m*n)$; the data centres should be chosen according to formula (10) for each dataset.

The ants must traverse all data centres to obtain the location of the datasets in a searching procedure, so the time complexity to complete one iteration for one ant is $O(m*n)$. The time complexity of all t iterations for all ants is $O(t*(c*m*n))$. Furthermore, each searching procedure must use the evaluation function to choose the best solution to update the pheromone. The time complexity of each procedure is $O(c)$. Thus, the total time complexity of SAI is $O(m*n)+O(t*(c*m*n+c))$.

5.6 Convergence analysis

This paper adopts the ACO-based strategy, which can largely avoid premature stagnation and speed up the search [41]. To validate the effectiveness and the feasibility of the strategy, the proof of the convergence algorithm is given as follows:

Theorem 2 *The ACO converges to the optimal solution at the probability of approximately 1, that is, for an arbitrarily small* $\varepsilon > 0$ *and a sufficiently large iteration number of t, the probability of finding the optimal solution at least one time in the first t iterations is:*

$$\lim_{t\to\infty} p^*(t) = 1 \tag{14}$$

To prove the convergence, we must first ensure that the optimal solution can be obtained during the n iterations. Theorem 2 [41] can fully prove that. Therefore, we must only demonstrate that there is more pheromone on the optimal path than on the non-optimal path in the subsequent iteration. The formal description is as follows:

Suppose $\tau_{ij}(t) > \tau_{kl}(t)$ when $\forall(i,j) \in s^*, \forall(k,l) \notin s^*$, and

$$\lim_{t\to\infty} \tau_{kl}(t) = \tau_{\min} \tag{15}$$

where s^* is the optimal solution, $\tau_{ij}(t)$ is the pheromone between d_i and dc_j, and $\tau_{kl}(t)$ is the pheromone between d_k and dc_l in the t-th iteration. In (15), τ_{min} is the minimum pheromone.

The following is the proof:

In the worst-case scenario, we assume that $(i\ ,\ j) \in s^*$, $\tau^*_{ij}(t^*) = \tau_{\min}$ and $(k,l) \notin s^*$, $\tau_{kl}(t^*) = \tau_{\max}$. Starting from the t^*-th generation, the pheromone of (k,l) becomes:

$$\begin{aligned}
&\tau_{kl}(t^*+1) = \max(\tau_{\min}, (1-\rho)*\tau_{\max})\\
&\tau_{kl}(t^*+2) = \max(\tau_{\min}, (1-\rho)^2*\tau_{\max})\\
&\ldots\ldots\\
&\tau_{kl}(t^*+t') = \max(\tau_{\min}, (1-\rho)^{t'}*\tau_{\max})\\
&\lim_{t\to\infty} \tau_{kl}(t^*+t') = \max\left\{\lim_{t\to\infty}\tau_{\min}, \lim_{t\to\infty}[(1-\rho)^{t'}*\tau_{\max}]\right\} = \tau_{\min}
\end{aligned}$$

□

Hence, in the subsequent generation, there is more pheromone on the optimal path than any other non-optimal path, and the pheromone on the non-optimal path gradually decreases. Therefore, an approximate optimal solution can be obtained using the ACO-based strategy.

6 Experimental results and analysis

6.1 Simulation parameters

In this paper, the randomly generated scientific workflow is applied as the input datasets of the simulation. The size of a single dataset is randomly distributed in the interval [80 and 300 GB], there are 40–120 datasets, the proportion of fixed datasets for input datasets ranges from 0.1 to 0.5, and there are 10–30 data centres. The storage capacity of a data centre is set up based on the sizes of all datasets. To simulate the heterogeneity of the resources of the data centres, we find an average storage capacity and make the variation randomly distribute from 10 to 30 % of the total capacity.

For the experiment, we simulate the proposed strategy from the proportion of fixed datasets, the number of data centres, the number of datasets and the maximum usage of the datasets. Then, we assess the experimental results from the *SDDSW*, the total time cost of data transfer and the data movements to verify the validity of the SAI strategy.

6.2 Experimental results

Based on the above four different inputs, we analyse and compare the experimental results primarily from the following three indicators: the *SDDSW*, the total data transfer time, and the data movements. In the analysis of the experimental results, the SAI represents the security-aware intermediate data placement strategy. The TS (Three Stages) represents the data deployment strategy that, respectively, addresses three problems: reducing the time cost of the data movements across the data centres, addressing with the data dependencies, and maintaining a relative load balancing of data centres in three stages [35]. BR(Build-time stage and Run-time stage) denotes the data placement strategy, which contains two algorithms that group the existing datasets in k data centres during the build-time stage and dynamically clusters newly generated datasets based on dependencies to the most appropriate data centres during the runtime stage [16].

6.3 Effect of the proportion of fixed datasets

As shown in Fig. 3a, the *SDDSW* in the SAI is much lower than those in the TS and BR layout strategies. However, the *SDDSW* is not a fixed value because the proportion of datasets varies. Sometimes, TS and BR fluctuate in terms of the *SDDSW*. Different proportions of fixed datasets may have different effects on the performance of our strategy. Thus, our strategy can guarantee better data security than the other two strategies when the proportion of fixed datasets changes. In Fig. 3b, the SAI strategy caused 15,666.33 Sim. Units (a quantitative unit of time in CloudSim [42]) of transmission on average for different circumstances. Compared with the BR, our strategy reduces the transmission time by 5.6 %. TS is superior to BR in transmission time, mainly because the TS strategy attempts to minimise the data transfer time as the optimisation objective. In Fig. 3c, when the proportion of fixed datasets increases, more datasets cannot be moved. Therefore, there are generally more movements of the flexible datasets. As shown in Fig. 3c, BR has fewer data movements than SAI and TS in most cases

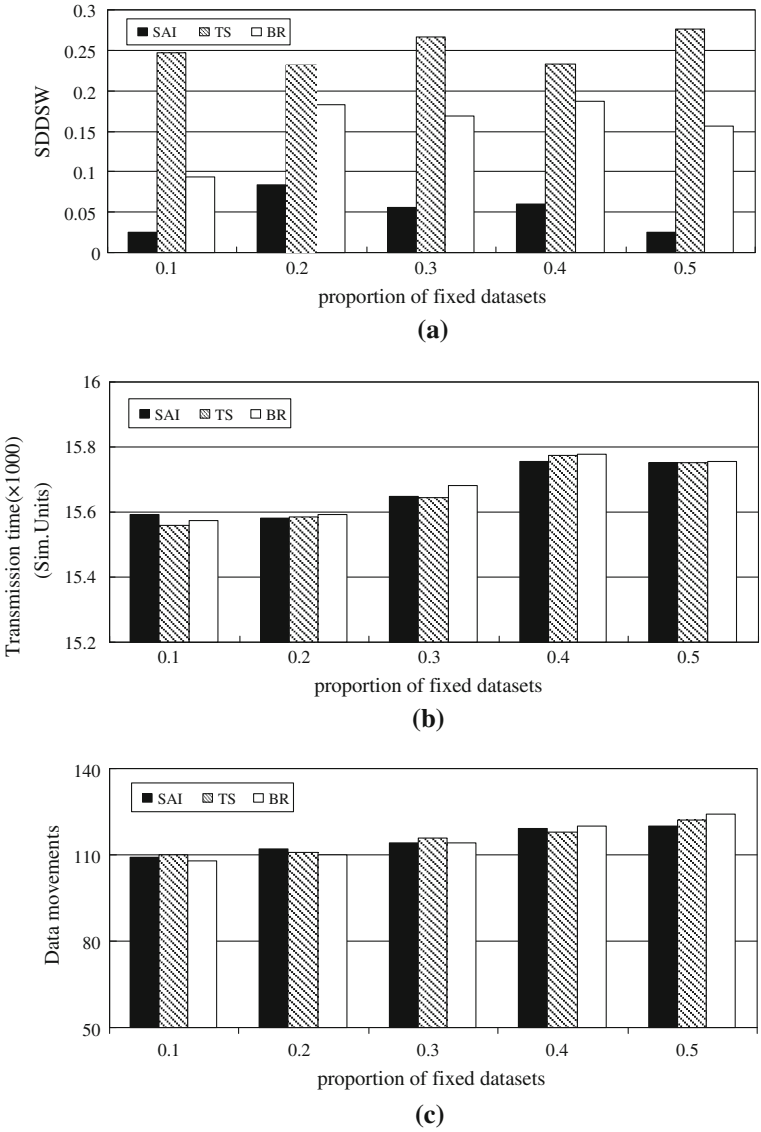


Fig. 3 Effect of the proportion of fixed datasets. **a** SDDSW, **b** transmission time, **c** data movements

because the main purpose of BR is to reduce the data movements. Thus, our strategy is better than the other two strategies in data security when the proportion of fixed datasets changes.

6.4 Effect of the number of data centres

Figure 4 shows the performance of the three strategies in different data centres for the *SDDSW*, the total transfer time and the data movements. In Fig. 4a, there is no stationary variation trend, and the SAI has a lower *SDDSW* than the TS and BR strategy mainly because that the number of data centres that can provide security services increases when the number of data centres

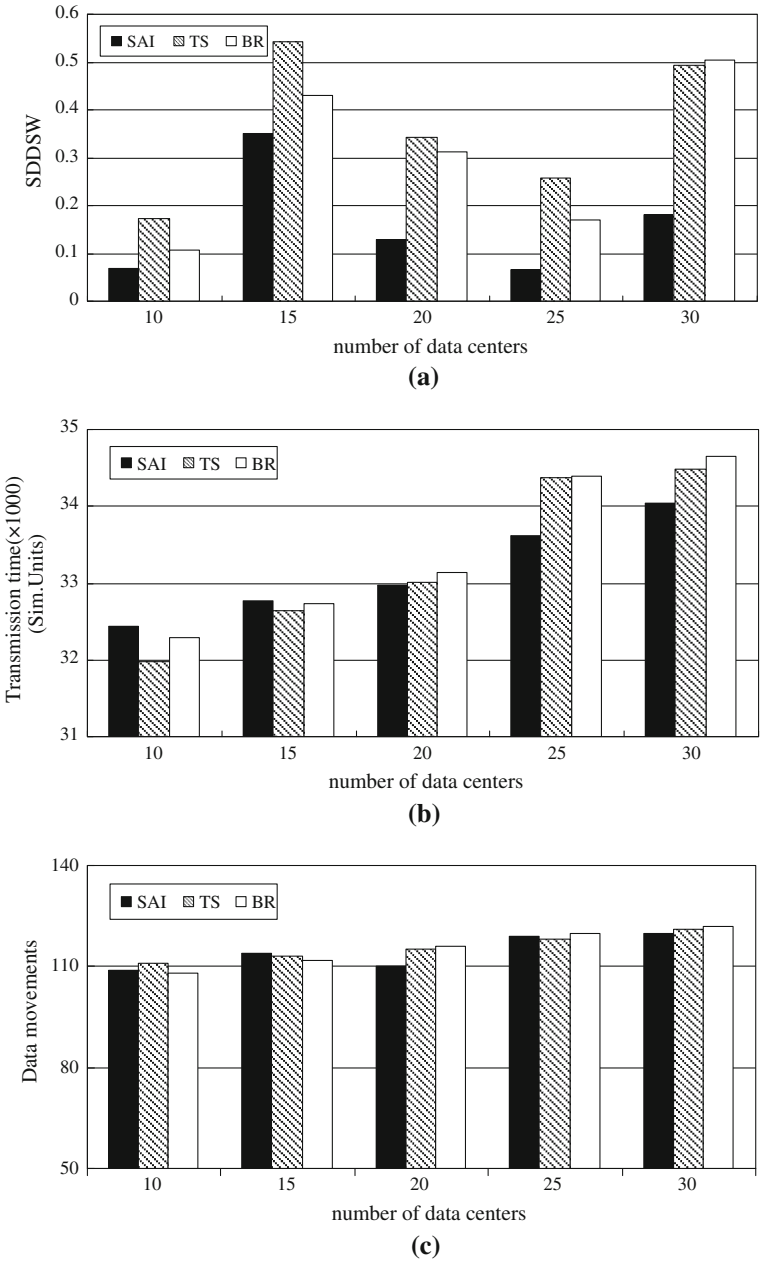


Fig. 4 Effect of the number of data centres. **a** *SDDSW*, **b** transmission time, **c** data movements

goes up. The security services are randomly generated. Thus, the *SDDSW* does not show a special trend. In general, SAI is better than BR and TS in terms of the*SDDSW*. In Fig. 4b, the SAI strategy reduces the transfer time by 3.8 % compared with the TS strategy, whereas the transfer time of the BR strategy is lower than that of the SAI strategy when the number

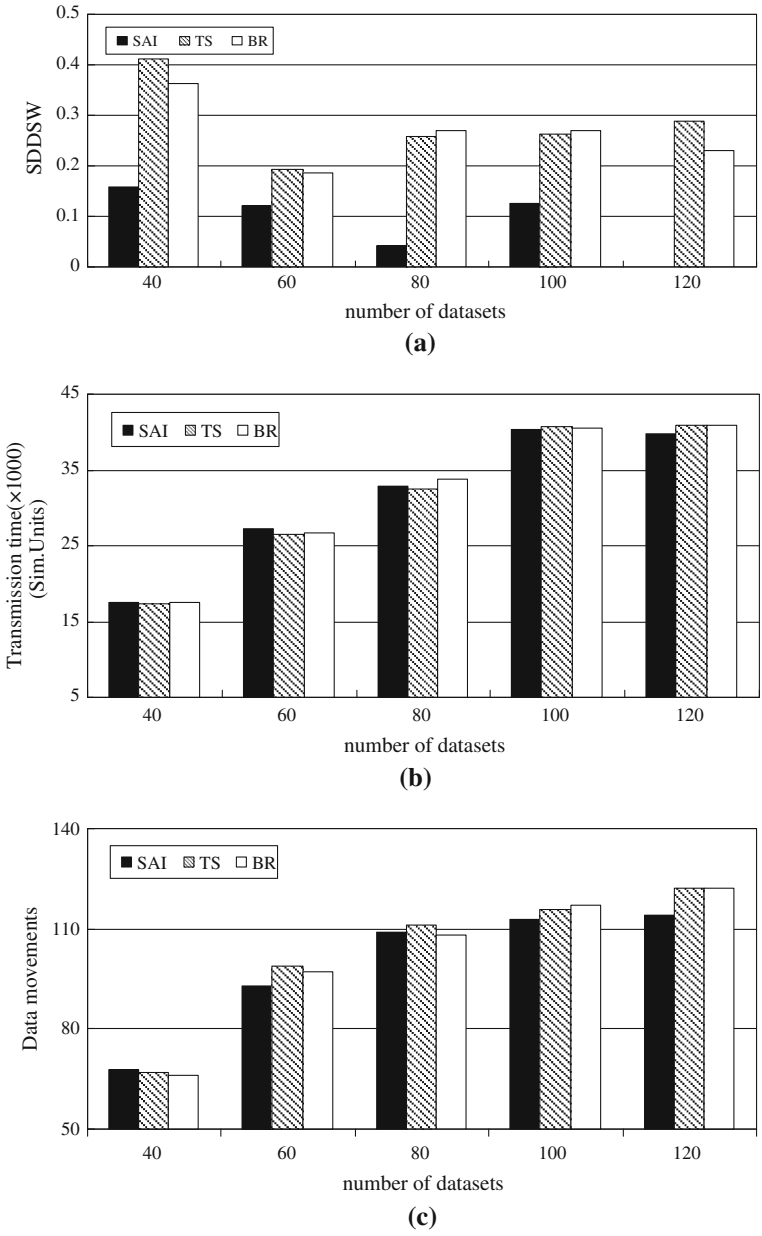


Fig. 5 Effect of the number of datasets. **a** SDDSW, **b** transmission time, **c** data movements

of data centres becomes 10. Because SAI regards the security-based scheduling as the main objective to compensate for the narrower bandwidth of the workflow, the transmission time increases. Overall, SAI is superior to BR and TS in terms of the total transfer time. However, when the number of data centres is 10, SAI is slightly worse than BR. In Fig. 4c, SAI does not show any advantages regarding the data movements compared with BR because SAI and

TS are intending to reduce the data transfer time instead of the data movements. Thus, we can see that our strategy is better than the other two strategies in terms of the security and the performance of scientific workflows.

6.5 Effect of the number of datasets

Figure 5 presents the statistical results of the *SDDSW*, the total transmission time and the data movements for these three data placement strategies using different sizes of datasets. Regarding the *SDDSW*, SAI has obvious advantages, particularly when there are 120 datasets. In that case, the SAI strategy can guarantee the security of scientific workflows compared with other strategies. As shown in Fig. 5b, when the number of datasets increases, the transmission time correspondingly becomes more erratic. More datasets result in more data transfer time. Our strategy is better than TS when the number of datasets is 100 and 120. In Fig. 5c, BR has fewer data movements than TS and SAI when there are 40 datasets. When there are more datasets, SAI gradually has fewer data movements than TS and BR, particularly when the number of datasets is 120. Thus, our strategy is more suitable for data-intensive applications.

6.6 Effect of the maximum data usage

From Fig. 6a, we observe that the *SDDSW* changes according to the maximum usage of the datasets. As the maximum data usage increases, the *SDDSW* correspondingly grows because the probability of transmitting data across data centres increases. Apparently, the results of our strategy are notably lower than those of the other two strategies, particularly when the maximum data usage is 2. In this case, the scientific workflow is not too complex. Thus, our strategy is more effective to safely deploy data. Similarly, the data transfer time increases when the maximum data usage increases, as shown in Fig. 6b. When the scientific workflow becomes more complex, more data must be transmitted to the data centres. Therefore, the data transfer turns more frequently, which results in more transmission time. Our strategy produces less transmission time than the other two strategies in some cases. However, SAI cannot effectively reduce the data movements in Fig. 6c because our strategy mainly aims to optimise the data security.

Therefore, by ensuring the data transfer time, we can conclude that our strategy improves the data security for scientific workflows in cloud. Compared with BR and TS, SAI better ensures the data security. Furthermore, the data transfer time of SAI is guaranteed. In some cases, SAI costs more than BR in data transfer time mainly because SAI satisfies the security service requirements when placing the data.

7 Conclusions

In this paper, we proposed a data placement strategy that can automatically allocate datasets to data centres to improve the data security. The strategy is executed by analysing the security model for scientific workflow systems from the service providers, service consumers and service evaluation. A security model is introduced to quantitatively measure the security services that are provided by the data centres. Then, we utilise an ACO-based algorithm to dynamically select the appropriate data centres for intermediate data to improve the data security while considering the data transfer time. This strategy is effective at improving data security, and the data transfer time is guaranteed during the execution of the scientific workflows.

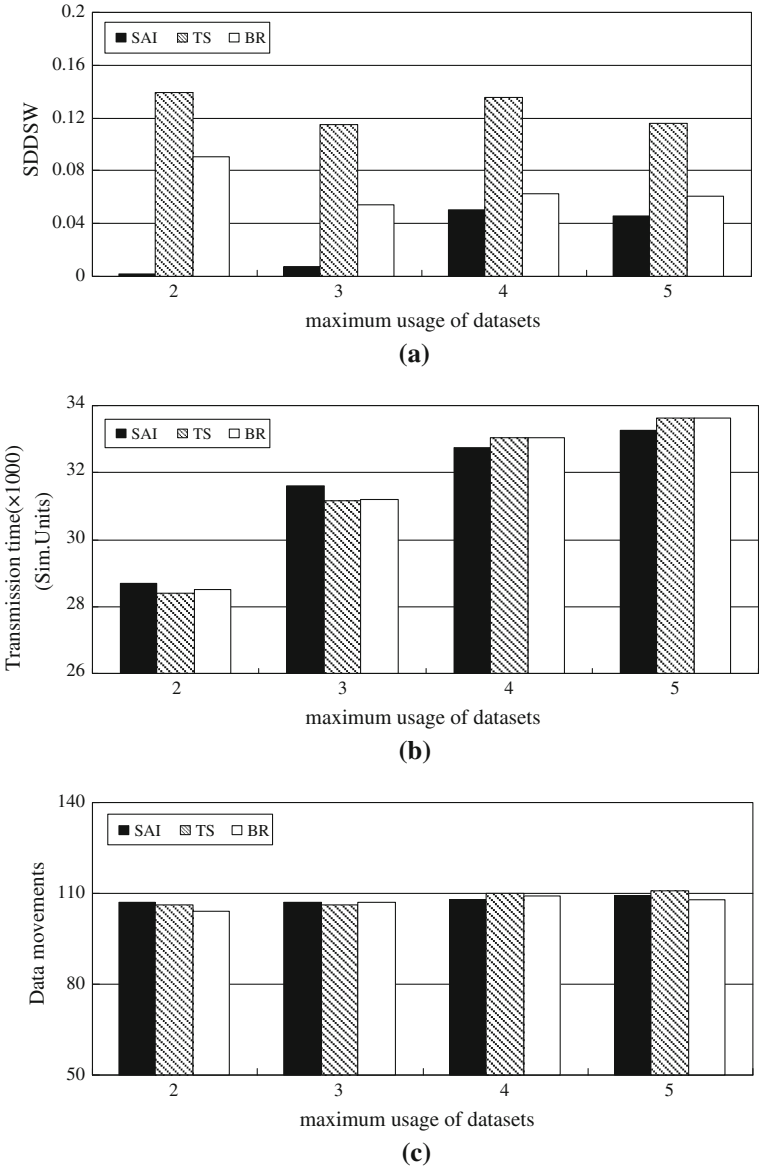


Fig. 6 Effect of the maximum usage of datasets. **a** SDDSW, **b** transmission time, **c** data movements

Currently, there are only a few studies on the data placement in scientific workflows. Furthermore, these studies mainly focus on how to reduce the data transmission across the data centres and rarely consider the security requirements, cost, and other factors. It is known that cloud computing provides a pay-demand computing model where the users can access applications and data from anywhere in the world and pay for what they use. Therefore, in the predictable future, we will exploit more efficient tactics to reduce the cost.

Acknowledgments This work was supported by the National Natural Science Foundation of China under Grant Nos. 61272107, 61202173, and 61103068; the Open Foundation of State Key Lab of Software Engineering of Wuhan University (SKLSE20080720 and SKLSE2012-09-29); the Open Foundation of Key Laboratory of Embedded System and Service Computing Ministry of Education of Tongji University; the Open Foundation of State Key Laboratory for Novel Software Technology of Nanjing University (KFKT2013B21); the Fundamental Research Funds for the Central Universities (WUT: 2013-IV-022 and WUT: 2014-IV-107) and the Ph.D. Programs Foundation of Ministry of Education under Grant No. (20090072110035 and 20110072120017).

References

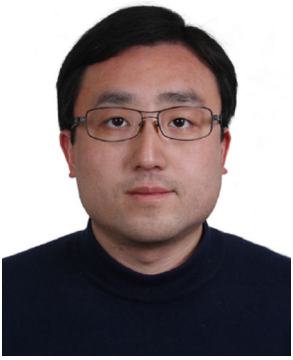
1. Costantino Thanos (2012) Global research data infrastructures: towards a 10-year vision for global research data infrastructures. http://www.grdi2020.eu/Repository/FileScaricati/6bdc07fb-b21d-4b90-81d4-d909fdb96b87.pdf
2. European Commission High Level Expert Group on Scientific Data (2010) Riding the wave: how Europe can gain from the rising tide of scientific data. http://cordis.europa.eu/fp7/ict/e-infrastructure/docs/hlg-sdi-report.pdf
3. Demchenko Y, Grosso P, de Laat C et al. (2013) Addressing big data issues in scientific data infrastructure. In: Proceedings of the international conference on collaboration technologies and systems, pp 48–55
4. Sagiroglu S, Sinanc D (2013) Big data: a review. In: Proceedings of the international conference on collaboration technologies and systems, pp 42–47
5. Acar UA, Chen Y (2013) Streaming big data with self-adjusting computation. In: Proceedings of the workshop on data driven functional programming, pp 15–18
6. Baru C, Bhandarkar M, Nambiar R et al (2013) Setting the direction for big data benchmark standards. In: Nambiar R, Poess M (eds) Selected topics in performance evaluation and benchmarking. Lecture notes in computer science, Springer, Heidelberg, pp 197–208
7. Srivastava D, Dong XL (2013) Big data integration. In: Proceedings of the international conference on data engineering, pp 1245–1248
8. Fei X, Lu S (2012) A dataflow-based scientific workflow composition framework. IEEE Trans Serv Comput 5(1):45–58
9. Szalay A, Gray J (2006) 2020 Computing: science in an exponential world. Nature 440(7083):413–414
10. Deelman E, Gannon D, Shields M et al (2009) Workflows and e-Science: an overview of workflow system features and capabilities. Future Gener Comput Syst 25(5):528–540
11. Yuan D, Yang Y, Liu X et al (2013) A highly practical approach towards achieving minimum datasets storage cost in the cloud. IEEE Trans Parallel Distrib Syst 24(6):1234–1244
12. Yuan D, Yang Y, Liu X et al (2012) A data dependency based strategy for intermediate data storage in scientific cloud workflow systems. Concurr Comput Pract Exp 24(9):956–976
13. Bertram L, Ilkay A, Chad B et al (2006) Scientific workflow management and the Kepler system. Concurr Comput Pract Exp 18(10):1039–1065
14. Weiss A (2007) Computing in the clouds. ACM Netw 11(4):16–25
15. Foster I, Yong Z, Raicu I et al (2008) Cloud computing and grid computing 360-degree compared. In: Proceedings of the grid computing environments workshop, pp 1–10
16. Yuan D, Yang Y, Liu X et al (2010) A data placement strategy in scientific cloud workflows. Future Gener Comput Syst 26(8):1200–1214
17. Wan C, Wang C, Pei J (2012) A QoS-awared scientific workflow scheduling schema in cloud computing. In: Proceedings of international conference on information science and technology, pp 634–639
18. Wei L, Zhu H, Cao Z et al (2014) Security and privacy for storage and computation in cloud computing. Inf Sci 258(2):371–386
19. Chu CK, Zhu WT, Han J et al (2013) Security concerns in popular cloud storage services. IEEE Pervasive Comput 12(4):50–57
20. Kalloniatis C, Mouratidis H, Islam S (2013) Evaluating cloud deployment scenarios based on security and privacy requirements. Requir Eng 18(4):299–319
21. Xiong L, Goryczka S, Sunderam V (2011) Adaptive, secure, and scalable distributed data outsourcing: a vision paper. In: Proceedings of workshop on dynamic distributed data-intensive applications, pp 1–6
22. Mohamed EM, Abdelkader HS, El-Etriby S (2012) Enhanced data security model for cloud computing. In: Proceedings of 8th international conference on informatics and systems, pp 12–17
23. Kaufman LM (2009) Data security in the world of cloud computing. IEEE Secur Priv 7(4):61–64
24. Armbrust M, Fox A, Griffith R et al (2010) A view of cloud computing. Commun ACM 53(4):50–58
25. Saritha S (2010) Google File System. Dissertation, Cochin University of Science and Technology

26. Hadoop (2011) http://hadoop.apache.org/
27. Natarajan A (2013) User-oriented modeling of scientific workflows for high frequency event data analysis. In: Proceedings of the 29th IEEE international conference on data engineering workshops, pp 306–309
28. Guo L, He Z, Zhao S et al (2012) Multi-objective optimization for data placement strategy in cloud computing. In: Liu C, Wang L, Yang A (eds) Information computing and applications. Communications in computer and information science. Springer, Heidelberg, pp 119–126
29. Guo L, Zhao S, Shen S et al (2012) A particle swarm optimization for data placement strategy in cloud computing. In: Zhu R, Ma Y (eds) Information engineering and applications. Lecture notes in electrical engineering, vol 154. Springer, London, pp 946–953
30. Ma F, Yang Y, Li T (2012) A data placement method based on Bayesian network for data-intensive scientific workflows. In: Proceedings of the international conference on computer science and service system, pp 1811–1814
31. Er-Dum Z, Yong-Qiang Q, Xing-Xing X et al (2012) A data placement strategy based on genetic algorithm for scientific workflows. In: Proceedings of the 8th international conference on computational intelligence and security, pp 146–149
32. Liu S-W, Kong L-M, Ren K-J et al (2011) A two-step data placement and task scheduling strategy for optimizing scientific workflow performance on cloud computing platform. Chin J Comput 34(11):2121–2130
33. Xi R, Lin N, Chen Y et al (2011) Compression and aggregation of Bayesian estimates for data intensive computing. Knowl Inf Syst 33(1):191–212
34. Peng Z, Guiling W, Xu X (2013) A data placement approach for workflow in cloud. J Comput Res Dev 50(3):636–647
35. Zeng P, Cui L-Z, Wang H-Y et al (2010) A data placement strategy for data-intensive applications in cloud. Chin J Comput 33(8):1472–1480
36. Xie T, Qin X (2006) Scheduling security-critical real-time applications on clusters. IEEE Trans Comput 55(7):864–879
37. Bishop M (2003) What is computer security? IEEE Secur Priv 1(1):67–69
38. Xie T, Qin X (2007) Performance evaluation of a new scheduling algorithm for distributed systems with security heterogeneity. J Parallel Distrib Comput 67(10):1067–1081
39. Zhu X, Lu P (2009) A two-phase scheduling strategy for real-time applications with security requirements on heterogeneous clusters. Comput Electr Eng 35(6):980–993
40. Zhu X, Qin X, Qiu M (2011) QoS-aware fault-tolerant scheduling for real-time tasks on heterogeneous clusters. IEEE Trans Comput 60(6):800–812
41. Stutzle T, Dorigo M (2002) A short convergence proof for a class of ant colony optimization algorithms. IEEE Trans Evol Comput 6(4):358–365
42. Calheiros RN, Ranjan R, Beloglazov A et al (2011) CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Softw Pract Exp 41(1):23–50

Wei Liu was born in 1978 and received his Ph.D. in computer architecture from the Department of Computer Science and Engineering, Fudan University. Currently, he is working in Wuhan University of Technology as an associate professor of the college of the computer science and technology. His research interests include cloud computing and green computing. He has published more than 10 papers in national or international key journals.

Su Peng was born in 1988 and received his bachelor's degree from Hubei University of Technology in 2011. She is currently a master candidate in School of Computer Science and Technology, Wuhan University of Technology.

Wei Du was born in 1978 and received her Ph.D. in computer science from the Department of Computer Science and Engineering, Huazhong University of Science and Technology. Currently, she is working in Wuhan University of Technology as a lecture of the college of the computer science and technology. Her research interests include information security and service computing. She has published more than 10 papers in national or international key journals.



Wei Wang was born in 1979 and received his Ph.D. in computer software and theory from the Department of Computer Science and Technology, Tongji University. Currently, he is working in Tongji University as a lecture of the Department of Computer Science and Technology. His research interests include service management, service computing, and information security. He got R. L. Zhang Scholarship and HP Chinese Best Student Scholarship in 2006. In 2007, he was granted IBM Ph.D. Fellowship. He has published more than 30 papers in national or international key journals.

Guo Sun Zeng was born in 1964 and received his BS, MS, and Ph.D. in computer software and application from the Department of Computer Science and Engineering, Shanghai Jiao Tong University. Currently, he is working in Tongji University as the vice dean of the department of the computer science and technology, and a supervisor of Ph.D. candidates in computer software and theory. He is an expert of 863 Program in information branch, the senior member of IEEE and China Computer Federation (CCF). He has made great achievements in the fields of heterogeneous network computing and heterogeneous information security. He has more than 90 papers published in national or international key journals.