Han YB, Sun JY, Wang GL *et al.* A cloud-based BPM architecture with user-end distribution of non-compute-intensive activities and sensitive data. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 25(6): 1157–1167 Nov. 2010. DOI 10.1007/s11390-010-1092-5

A Cloud-Based BPM Architecture with User-End Distribution of Non-Compute-Intensive Activities and Sensitive Data

Yan-Bo Han (韩燕波)[1], Jun-Yi Sun (孙君意)[1,2], Gui-Ling Wang (王桂玲)[1], and Hou-Fu Li (李厚福)[1]

[1] *Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China*

[2] *Graduate University of Chinese Academy of Sciences, Beijing 100049, China*

E-mail: yhan@ict.ac.cn; {sunjunyi, wangguiling, lhfsday}@software.ict.ac.cn

Received July 15, 2009; revised January 29, 2010.

Abstract While cloud-based BPM (Business Process Management) shows potentials of inherent scalability and expenditure reduction, such issues as user autonomy, privacy protection and efficiency have popped up as major concerns. Users may have their own rudimentary or even full-fledged BPM systems, which may be embodied by local EAI systems, at their end, but still intend to make use of cloud-side infrastructure services and BPM capabilities, which may appear as PaaS (Platform-as-a-Service) services, at the same time. A whole business process may contain a number of non-compute-intensive activities, for which cloud computing is over-provision. Moreover, some users fear data leakage and loss of privacy if their sensitive data is processed in the cloud. This paper proposes and analyzes a novel architecture of cloud-based BPM, which supports user-end distribution of non-compute-intensive activities and sensitive data. An approach to optimal distribution of activities and data for synthetically utilizing both user-end and cloud-side resources is discussed. Experimental results show that with the help of suitable distribution schemes, data privacy can be satisfactorily protected, and resources on both sides can be utilized at lower cost.

Keywords cloud-based BPM, user-end autonomy, data privacy

1 Introduction

With the boom of the Web applications, the Internet services, new business models and innovative computing paradigms (e.g., Software-as-a-Service[1], grid computing[2] and cloud computing[3]), the Internet has evolved into an indispensable social infrastructure and the largest computing platform of the world as well. The Internet-based cyberspace takes shape, and its emergence and development may impact our lives fundamentally. This paper views the buzz-concept cloud as an umbrella term referring to 1) the Internet-based capabilities provided as services in different forms for different tenancies, i.e., SaaS (Software-as-a-Service) for end-users, PaaS (Platform-as-a-Service) and IaaS (Infrastructure-as-a-Service) for application developers. The Internet-based data management and computing facilities are the most popular functionalities delivered as cloud services; 2) a scalable multi-tenancy infrastructure delivering cloud services; 3) various application fabrics, such as virtual communities, which partition and use the Internet-based cyberspace for controlled sharing and collaboration. We believe, service-based and multi-tenant cyberspaces have great potentials in reforming the IT realm.

This paper is related to BPM on cloud. BPM (Business Process Management)[4] resides on top of the IT-related value chain, embodying visible value of IT systems from user's point of view. Recently, BPM has received even more attention since the widely adoption of the SOA (Service-Oriented Architecture)[5] paradigm. In general, BPM can be seen as a major carrier of SOA merits. cloud-based BPM is a natural development in line with the above-stated trends. Our work is motivated by the increasingly representative development of decentralized process management. A typical scenario is upgrading existing EAI systems to adapt to the Internet-based infrastructure services (typically cloud services at present). Business process engines may run on cloud-side, providing a sort of PaaS capabilities. While the main business logic may still be controlled by the user-end, some tricky parts can be executed using

Regular Paper

Supported by the National Basic Research 973 Program of China under Grant No. 2007CB310805, the National Natural Science Foundation of China under Grant Nos. 90412010, 60970131 and 60903048, the National High-Tech Research and Development 863 Program of China under Grant No. 2006AA01A106 and the Beijing Natural Science Foundation under Grant No. 4092046.

the cloud-side capabilities. Such distribution requirements bring impact to the overall architecture. There exist a number of researches that explore how BPM can be more cost-effectively and managed more efficiently in clouds. However, little effort is made to investigate issues of utilizing user-end capability synthetically with cloud and protecting users' data privacy in cloud environment.

The rest of this paper is organized as follows. Section 2 analyzes the benefits and challenges of cloud-based BPM, and identifies the concrete research problems of the paper. In Section 3, after analyzing candidate architectures constellation of cloud-based BPM, we propose a novel architecture supporting user-end distribution of non-compute-intensive activities and sensitive data. Section 4 explores some key issues on an optimal distribution of activities and data in processes in order to utilize resources on both sides. Section 5 makes some assessment with an experimental scenario. Closely related researches are discussed in Section 6. We conclude with our initial findings and some open questions for future exploration in Section 7.

2 Problem Definition

2.1 Sharing BPM Capabilities on Cloud

Nowadays, more and more organizations use BPM systems to improve the effectiveness and efficiencies of enterprise operation. At present, there is a large amount of powerful and expensive commercial BPM software on the market, such as, Oracle BPM[6], Microsoft BizTalk[7], and IBM Websphere Process Management[8]. On the other hand, inexpensive open-source software, such as jBPM[9], Active BPEL[10] and Shark[11], providing another choice of open-source software often suffers from some functional and non-functional shortcomings. Using commercial BPM products means a considerable upfront investment on software procurement and maintenance to users. Users may also have to buy a high performance server to run BPM engine, and employ IT experts to maintain the system.

BPM users are among the people who may profit most from cloud computing. Such burdens as purchasing hardware and software products on-premise, as well as installing, maintaining, or upgrading them, can be relieved. It is particularly attractive to SMEs (Small and Medium Enterprises), as they now can use scalable BPM services with a pay-as-you-go manner[12]. Moreover, multiple business processes which belong to different enterprises may cooperate with each other on the cloud-based BPM platform. Major IT vendors begin to provide BPM services in the cloud, such as IBM's Blue Works[13], Microsoft's SharePoint Online[14] and Vitria's M3O[15]. cloud-based BPM is getting prevalent.

2.2 Challenges of Cloud-Based BPM

Although cloud-based BPMs can help the organizations such as SMEs improve enterprise operational efficiency and cut down their expenditures, there are still some barriers to the adoption of cloud-based BPM. Many users have already deployed a number of legacy applications in their private computing environment. They may have built up their own EAI systems, and have their own rudimentary or even full-fledged BPM systems at their end, but still intend to make use of cloud-side services and BPM capabilities at the same time. It would be helpful to increase efficiency and effectiveness of non-compute-intensive activities on the user side. Moreover, some data involved in business processes may be of business secrets, which users prefer not to leak. They are unwilling to put these sensitive data in cloud because they are afraid of losing control of their data when they release the information into cloud for processing[16].

Only when these barriers are removed, would the cloud-based BPM be widely accepted by the majority of enterprises. In order to utilize both user-end and cloud-side resources synthetically and protect users' data privacy, cloud-based BPM should have a decentralized architecture and support user-end distribution of business processes.

The challenges addressed in this paper in supporting decentralized architecture include the following.

Privacy Protection. Although cloud-side has very secure infrastructure behind solid firewall and has advanced mechanism to keep the isolation of multi-tenancies' data, users may be still unwilling to release their sensitive-data to cloud because they fear losing control of the data. Thus, cloud-based BPM should allow sensitive data to be distributed at the user-end. However, this poses a challenge to the architectural design of cloud-based BPM, because logic integrity of an overall business process has to be well maintained while facilitating user-end data dependency and autonomy.

Optimal Distribution. If user-end distribution is allowed to have decentralized architecture, the cloud-based BPM should choose optimal distribution of activities and data in each process, because different distributions will make different cost according to the charging rates of cloud, capability limitations of user-end and data privacy risk in cloud.

3 Our Approach

To design a decentralized architecture of cloud-based

BPM, some problems should be made clear, for example, where to enact processes, where to execute activities, as well as where to store the data produced and consumed by activities. In this regard, we need to investigate architecture patterns and identify an optimal pattern to utilize resources on both sides synthetically and sufficiently.

3.1 Design Tradeoff

Every coin has two sides. There is a contradiction between cloud computing and user-end distribution. If user-end distribution is allowed, then some advantages of cloud computing such as zero installation may be sacrificed. Thus, design tradeoff is inevitable when building a cloud-based BPM system supporting user-end distribution.

To study the problem of user-end distribution analytically and synthetically, we propose a PAD (Process-enactment, Activity execution and Data storage) model which describes BPM architectures from the distribution of three independent functions. Fig.1 demonstrates the candidate architectures according to the PAD model.

These architectures can be classified into four principal types: traditional standalone BPM, user-end BPM with cloud-side distribution, cloud-based BPM with user-end distribution and exiting cloud-based BPM.

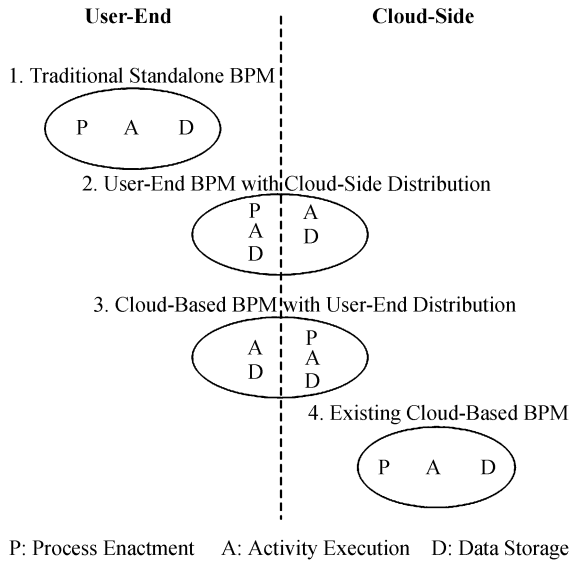

Fig.1. Different patterns of BPM constellation.

Pattern 1 is the traditional architecture, which distributes everything at user-end.

Pattern 4 is adopted by existing cloud-based BPMs, which distributes everything on cloud. The advantage is that users do not have to install anything at the user-end. However, users of this kind of system will lose control of their data and cannot utilize user-end resources.

For users who already have full-fledged BPM engines at their-end, Pattern 2 can be a good choice. They just need to distribute some compute-intensive activities to cloud-side for acquiring stronger capabilities and better performance.

For users who do not have their own full-fledged BPM engines at user-end, the ideal style is Pattern 3. Process engine is on cloud-side, but process designers can specify their distribution requirements of activity execution and data storage. For example, sensitive data and non-compute-intensive activities can be distributed at user-end, and compute-intensive activities and non-sensitive data can be distributed on cloud-side.

3.2 Separation of Control Data and Business Data

In traditional centralized BPM systems, data should be stored in a place, where the process engine can have easy access. Therefore, if we deploy them into cloud directly, all business data related to certain process should be stored on cloud-side, which violates our intention of protecting data privacy. To protect the sensitive data, a novel BPM architecture supporting Pattern 2 or Pattern 3 should be in place to decouple process engine and business data.

Typically, a business process is composed of activities or tasks that involve people, services, and data. An activity may be a service invocation or a human task, and activities are orchestrated to make a process that is often represented as flow chart. There are two primary perspectives of a business process: control-flow perspective and data-flow perspective[17]. The control-flow perspective regulates which activity is being performed, and which is the next activity to be executed. The data-flow perspective regulates how data is forwarded from one activity to another and data mapping issues.

Process engines usually have to handle both control-flow and data-flow. They use control data, such as activity status, switching condition value, and process status to determine the control-flow, and move business data from one activity to another to ensure data-flow. In fact, during process enactment, it is possible for a process engine not to access some business data. In order to protect users' data privacy, we must relieve cloud-side engine from dealing with data-flow. As a result, the cloud-side process engine only focuses on handling control-flow according to the control data. So the business data, which only need to be exchanged between user-end activities, does not have to be accessible for the cloud-side engine.

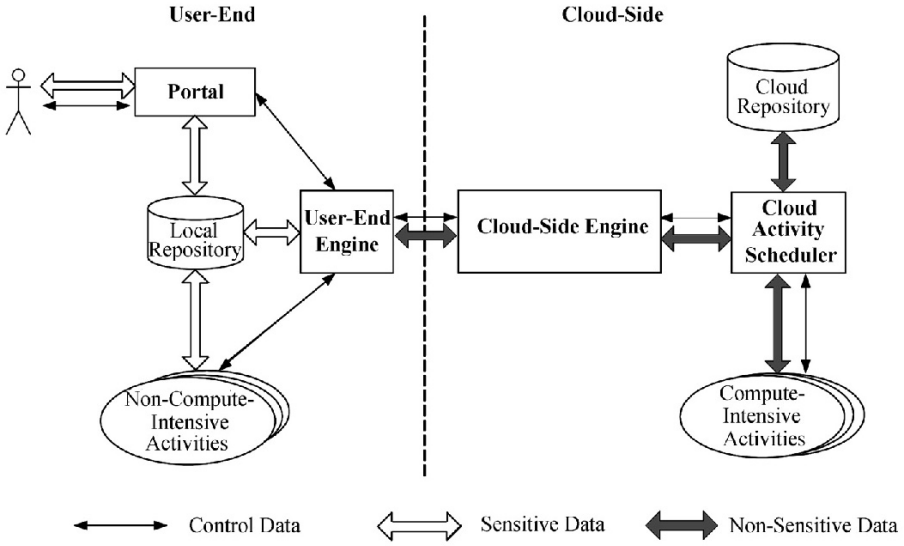

Fig.2. Architecture of cloud-based BPM with user-end distribution.

Dealing separately with control data and business data can also enhance the stability of our system. On the one hand, when user-end crashes, the process instance on the cloud-side can be suspended for the process instance status is maintained on cloud-side. After user-end resumes, the process instance can continue. On the other hand, users' sensitive data will not be influenced when the process engine on cloud-side is unavailable, because the data can be stored in a local repository that is under their own control.

3.3 Architectural Rationales

Our design objectives of cloud-based BPM with user-end distribution are as follows. Firstly, the cloud-side engine handles process enactment by collaborating with the user-end engine. These two engines can handle activity execution and data storage on their own side. Secondly, between cloud-side and user-end, there mainly exists control data such as activity status or service request, which does not contain business data but reference ID. Lastly, business data exchanged between cloud-side and user-end is also allowed but should be under users' surveillance through encrypted tunnel and could be charged based on the amount of data transferred.

Fig.2 illustrates the novel cloud-based BPM, which has an event-driven architecture supporting user-end dependency and autonomy while maintaining logic integrity of an overall business process. As shown in Fig.2, the solid arrow represents control-flow, and the wide arrow represents data-flow. The non-sensitive data is stored in the cloud repository, and users' sensitive data, such as some business documents or confidential financial reports, is stored in local repository under their own control. There are mainly three components (portal, user-end engine, and local repository) installed at user-end, which could be a normal PC. The cloud-side engine with activity scheduler is built on large server clusters, which feature high performance and scalability.

With this architecture, it is no longer necessary for the sensitive data to be accessed by the cloud-side engine especially when those data only need to be exchanged between user-end activities. When activities distributed on cloud-side want to use the data in user-end repository, the cloud-side engine must get authorized by the user-end engine to obtain them.

Users that need the cloud-based BPM just have to deploy these user-end components on their private server, and then get the benefits of full-fledged BPM system without losing control of their sensitive data. Moreover, they can also make some further development on the basis of these components to satisfy their specific needs.

4 Key Issues with Scientific Exploration

In this section, we describe how we can ensure that the cloud-side engine collaborates seamlessly with the user-end components to maintain logic integrity of an overall business process, and also discuss our optimal distribution mechanism and privacy protection issues in more detail.

4.1 Communication Between Cloud-Side and User-End

In the communications between cloud-side and user-end, six types of event are defined as carriers of

communication, i.e., process operation event, activity completion event, service invocation event , work-item creation event, data request event and data response event.

Fig.3 illustrates the schemas of these events. As indicated in the graph, process operation event describes which process will be operated (start, stop, suspend or resume), who is the operator, and when the operation happens; activity completion event represents which process instance it belongs to and its completion time; service invocation event describes where the service is, what the input data, output data, and data mapping relations are; work-item creation event is similar to service invocation event but has more properties, i.e., the form URL and access authority; data request and data response events are used when it is necessary to transfer data between user-end and cloud-side.

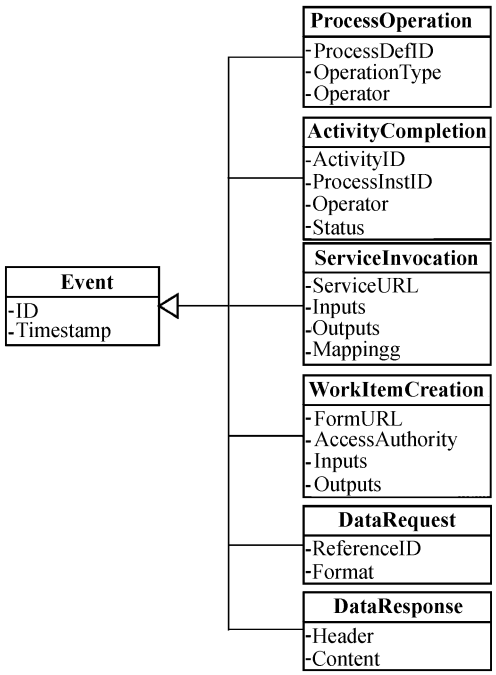

Fig.3. Schemas of communication events.

It should be noted that the "Inputs" and "Outputs" attributes in these events are represented with unique reference IDs for the data in the repository. Although the cloud-side engine can obtain the data ID from these events, it is not able to get the referred data if the data is in the user-end repository, except that the user-end engine gives it authorization.

To better understand the communication and data forwarding mechanism, a simple business process is given in Fig.4. The process involves three sequential activities: the first one is a manual task in which user can configure the input data, the second one is an activity distributed at the user-end, and the third one is an activity distributed on the cloud-side.

In this example, the second activity uses the data entered by user in the first activity, and the third activity uses the output data of the second one. Fig.4 depicts how cloud-side engine and user-end components collaborate to execute this simple business process.

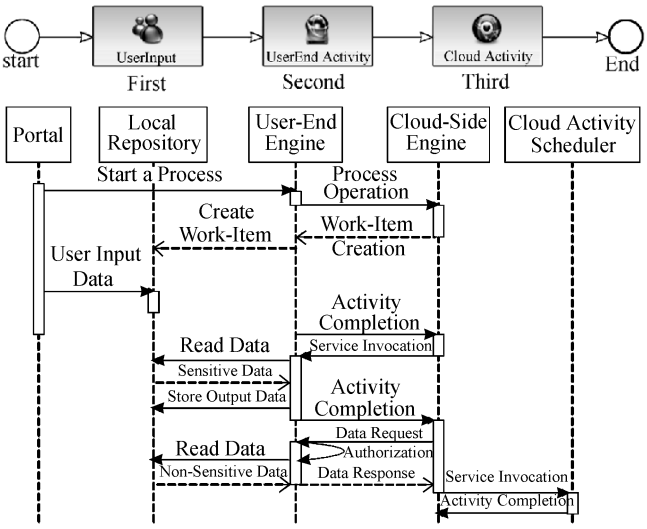

Fig.4. Sequence diagram of a simple business process enactment in cloud-based BPM.

As illustrated in Fig.4, the cloud-side engine communicates with the user-end engine with the aforementioned events. In the example, between cloud-side engine and user-end engine, control-data is mainly transferred. The sensitive business data is only transferred between local repository and the other two user-end components. The non-sensitive data can be transferred to cloud-side only with the authorization of user-end when it is necessary.

In practical applications, the business processes are more complex than the aforementioned example. During a process enactment, the cloud-side engine may sometimes have to evaluate switching condition expressions, which may be partially related with sensitive data. Under this situation, the cloud-side engine will send the expressions encapsulated with communication event to the user-end engine, which can invoke a local service to evaluate the expressions concerned with the sensitive data and send the evaluation value back to the cloud-side engine.

4.2 Optimal Distribution

Avoiding sensitive data being transferred between cloud-side and user-end is a direct and effective method to protect users' data privacy. However, sometimes, users want to utilize resources on both sides synthetically. They want to utilize the high performance

computing and massive storage capability of the cloud, especially when processing data which requires relatively lower privacy level.

There are several schemas about the distribution of an activity and its relative data. We list eight different distribution schemas of an activity and its input and output data with the IAO (Input-Activity-Output) model in a cloud environment. As illustrated in Fig.5, there are 8 rectangles, which represent 8 different distributions. The upper part of each rectangle represents cloud-side distribution, and the lower part represents user-end distribution. The symbol A represents an activity, the symbol I represents its input data, and the symbol O represents its output data.

Fig.5 just gives eight basic examples of the distribution of an activity and its relative data. In fact, an activity may have multiple input data items and output data items, which can be distributed on different sides. One data item can also be used by many activities on different sides.

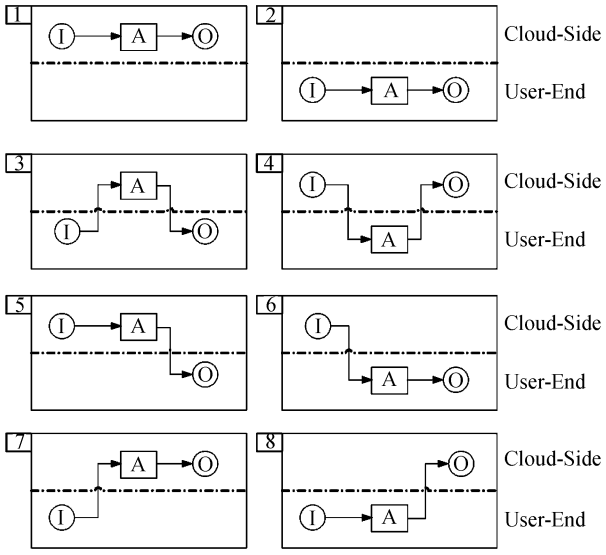

Fig.5. Distribution schemas of activity and data.

In general, user-end has relatively lower performance but higher level of privacy. cloud-side has relatively better performance but lower level of privacy. Furthermore, users have to pay to the cloud operators, such as Amazon, for the transferred data and consumed CPU-hours. However, user-end distribution generally means a fixed monetary cost. Therefore, the cloud-based BPM system should give recommendation about optimal distribution to users according to certain criteria. So we consider the distribution of the activity and data from three perspectives: time cost, monetary cost, and privacy risk cost.

Time Cost. Let us assume there is an activity a. The execution time of a on the cloud-side is $t_c(a)$; the execution time of a at the user-end is $t_u(a)$. Generally speaking, the scalability of cloud-side cluster surpasses that of user-end server. As a consequence, $t_c(a)$ will be less than $t_u(a)$ when handling some compute-intensive task especially under heavy concurrent requests. Although in some circumstance, activity execution on cloud-side costs relatively less time, the data transmission time may offset the performance advantage of cloud-side. Let b be reference bandwidth between cloud-side and user-end in bytes per second. If a involves a set of data items (input or output) $D_R = \{d_1, d_2, \ldots, d_r\}$ to be transferred between cloud-side and user-end, then the transmission time needed is $\sum_{i=1}^{r} \frac{size(d_i)}{b}$, $d_i \in D_R$, where *size* is a function representing the size of data.

Monetary Cost. Other than above performance issues, we should also consider monetary cost. In general, both computing and transferring data into and out of the cloud storage are charged for. Let us assume the charge rate of computing resource is f dollars per CPU-hour, the charge rate of storage resource is g dollars per GB, and the charge rate of transferring data is h dollars per GB. Then the cost of executing an activity a on the cloud-side is $f \cdot t_c(a)$ dollars. Let $D_K = \{d_1, d_2, \ldots, d_k\}$ be the set of activity a's relative data items stored on the cloud-side. The monetary cost of storing its relative data on the cloud-side is $\sum_{i=1}^{k} size(d_i) \cdot g$, $d_i \in D_K$ dollars, and the monetary cost of transferring its input data and output data between user-end and cloud-side is $\sum_{j=1}^{r} size(d_j) \cdot h$, $d_j \in D_R$ dollars.

Privacy Risk Cost. Data privacy is another issue to be considered. Generally speaking, companies will give their important data or documents privacy levels. For an activity a, we suppose users can give a privacy level $p(d_i)$ to its relative data. Then the privacy risk cost is $\sum_{i=1}^{k} p(d_i) + \sum_{j=1}^{r} p(d_j)$, $d_i \in D_K$, $d_j \in D_R$.

A process is usually composed of many activities which may share input data or output data. One data item can be used by many activities, and one activity can also use many data items. Therefore, our system should provide global optimization in order to give recommendation to users about optimal distribution of an overall business process.

Let $\boldsymbol{A} = \{a_1, a_2, \ldots, a_m\}$ be the set of activities of a business process. Let $\boldsymbol{D} = \{d_1, d_2, \ldots, d_n\}$ be the set of business data involved in this process. Let $\boldsymbol{S} = \{s_1, s_2, \ldots, s_m\}$ be the distribution vector of activities, $s_i = 1$ represents that activity a_i is distributed on the cloud-side, and $s_i = 0$ represents that activity a_i is distributed at the user-end. Let $\boldsymbol{Q} = \{q_1, q_2, \ldots, q_n\}$ be the distribution vector of data, $q_j = 1$ represents that data item d_j is distributed at the cloud-side, and $q_j = 0$ represents that data item d_j is distributed at

the user-end. If $|s_i - q_j| = 0$, it represents that activity a_i and data item d_j are on the same side, otherwise, it represents that they are on the different sides. Let $\boldsymbol{R}$ be the relation matrix of activities and data, $\boldsymbol{R}(i,j) = 1$ represents that activity a_i reads or writes data d_j, $\boldsymbol{R}(i,j) = 0$ represents that activity a_i and data item d_j have no direct relation. Then the problem to recommend the optimal distribution of a business process with smallest cost can be described as an integer optimization model described as follows.

Time cost:

$$cost_{\mathrm{t}} = \sum_{i=1}^{m}[t_c(a_i) \cdot s_i + t_u(a_i) \cdot (1 - s_i)] + \sum_{i=1}^{m}\sum_{j=1}^{n} \frac{size(d_j)}{b} \cdot \boldsymbol{R}(i,j) \cdot |s_i - q_j|.$$

Monetary cost:

$$cost_{\mathrm{m}} = \sum_{j=1}^{n}[g \cdot size(d_j) \cdot q_j] + \sum_{i=1}^{m}\sum_{j=1}^{n} h \cdot size(d_j) \cdot \boldsymbol{R}(i,j) \cdot |s_i - q_j| + \sum_{i=1}^{m} f \cdot t_c(a_i) \cdot s_i + \mu.$$

Privacy risk cost:

$$cost_{\mathrm{p}} = \sum_{j=1}^{n}[p(d_j) \cdot q_j] + \sum_{i=1}^{m}\sum_{j=1}^{n} p(d_j) \cdot \boldsymbol{R}(i,j) \cdot |s_i - q_j|.$$

Optimal distribution model:

$$\begin{aligned} &\min cost = w_{\mathrm{t}} \cdot cost_{\mathrm{t}} + w_{\mathrm{m}} \cdot cost_{\mathrm{m}} + w_{\mathrm{p}} \cdot cost_{\mathrm{p}} \\ &\text{s.t.} \quad s_i \in \{0,1\}, \quad q_j \in \{0,1\}. \\ &\qquad constr(s_i, q_j). \end{aligned}$$

In the model above, w_{t} is the weight factor of time cost; w_{m} is the weight factor of monetary cost; w_{p} is the weight factor of privacy risk cost, $constr(s_i, q_j)$ is the constraints specified by users. μ is a fixed monetary cost of maintaining user-end distribution. By figuring out the model, we can get the optimal distribution represented by vectors $\boldsymbol{S}$ and $\boldsymbol{Q}$, which can utilize resources on both sides synthetically with the relatively smallest cost under users' restriction.

It is worth noting that this model is only used to give recommendation to users about how to distribute activities and data in a process, when they deploy it. So some factors in runtime are not included, such as an activity that may be executed several times.

5 Assessment with an Experimental Scenario

As a middleware, cloud-based BPM can support lots of Internet-based applications at back end, such as E-Commerce, CRM, and HIS online. This section illustrates the use of our architecture with an exemplary scenario.

Assume a company intends to sell its products on the Internet. However, it is not easy and too expensive to develop a full-fledged e-shopping application, because it involves many complex business process at back end, such as user authentication process, shipping process, purchase order process. What makes it worse is that processes change frequently. Thus, the company needs a BPM engine to support the e-shopping application. However, BPM products on the market are too expensive for them. Even if the development of the e-shopping application is affordable, it would have already taken a long time. Under this circumstance, this company can register as a user (or tenancy) of our cloud-based BPM, and design the business processes. After a successful registration, the system provides a basic portal and repository for them online. If they worry about losing control of the sensitive business data, then they can download and install user-end components on their own server to protect the sensitive data. They can also distribute some compute-intensive activities on the cloud-side to reduce execution time, and even do some value-added developments on the user-end components such as personalizing Web portal. Therefore, with our cloud-based BPM, this company can save money and time, and protect their sensitive data at the same time.

To describe this scenario vividly, a deployment diagram is given in Fig.6.

In order to validate our ideas, we do some experiments on Google App Engine (GAE)[18], which is an open platform for cloud computing, and supports Python and Java programming language. Therefore, we built a prototype system of cloud-based BPM on GAE. As mentioned in above sections, the cloud-side components and user-end components should communicate with each other to run a whole process, so a high performance communication protocol is needed. XMPP[19] (Extensible Messaging and Presence Protocol) is what we want. It is an open and decentralized XML routing technology that allows any entity to actively send messages to another entity. With the help of XMPP, the cloud-side engine does not need to wait for service invocation result or poll repetitively to check the result,

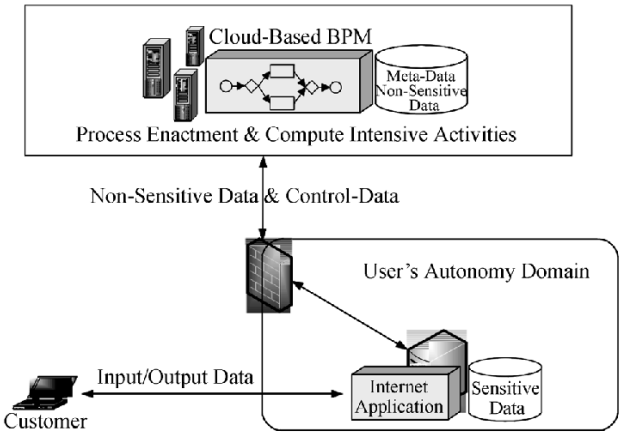

Fig.6. A scenario of cloud-based BPM facilitating practical application.

but the user-end engine sends the result back asynchronously upon completion status.

GAE has a good support for XMPP. Any rich client with a connection to an XMPP server (such as GTalk) can use XMPP to interact with an App Engine in real time, including receiving messages initiated by the application. Each XMPP entity is identified by its Jabber ID (JID)[20]. The cloud-side engine with JID cloudbpm@appspot.com can communicate with the user-end engine through XMPP protocol. Our system adopts multi-tenancy architecture, so each tenancy has its own user-end engine JID. Besides exchanging control data, the non-sensitive data like some files or documents can also be transferred safely between the cloud-side and user-end through the extension protocols such as XEP-0096 and XEP-0244[21].

In order to see the performance difference of activity distribution, some simple experiments are carried out. GAE is used as the cloud-side, and our lab PC is used as the user-end. In the experiment, we used the same code on both sides, and the configuration of the user-end PC is: Intel Core(TM) 2 Duo Processor, 2 GB RAM, Realtek Gigabit Ethernet NIC.

There are two activities in a process, one activity is to sort one million of random integers (compute-intensive), and the other is to sort only one thousand of random integers (non-compute-intensive). We recorded the average time cost of them under different levels of concurrent requests on different sides. Fig.7 depicts the experimental result.

As shown in Fig.7(a), the non-compute-intensive activities will cost less time when they are distributed at user-end, because when an activity is not compute-intensive, it can be processed efficiently at user-end without the data transmission time cost between the cloud-side and user-end. From Fig.7(b), we can see that the compute-intensive activities cost less time when they are distributed on the cloud-side, especially the level of concurrent requests is high. GAE (the cloud platform used in the experiment) is based on Google's server farm, which has great scalability. Therefore, when a compute-intensive activity is confronted with heavy concurrent requests, its performance will decrease smoothly if it is distributed at the cloud-side, but when it is distributed at the user-end, the performance will decrease rapidly if the number of concurrent requests increases.

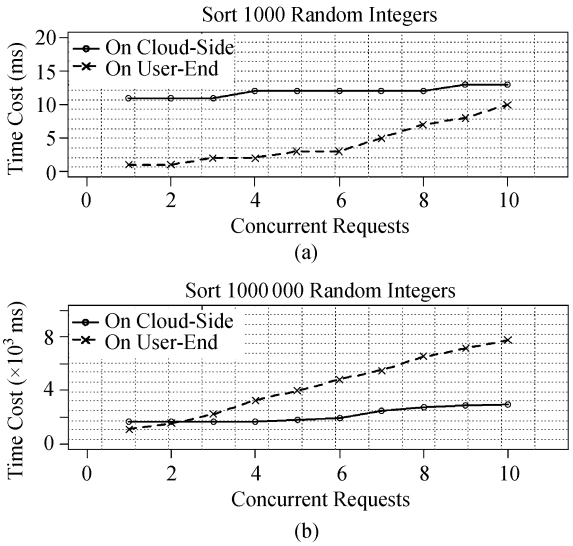

Fig.7. Time cost comparison. (a) Non-compute-intensive activity. (b) Compute-intensive activity.

In view of the different costs under different distributions, we make some effort on the optimal distribution of activities and data to help BPM users to make a better distribution plan. As described in Subsection 4.2, given certain costs of time, money, and privacy for each activity and data, optimization algorithm is used to figure out two sequences which demonstrate the distribution of activities and data respectively. With the help of optimal distribution algorithm, our deploy tools became smarter to give distribution recommendation.

6 Discussion

In this section, we will first discuss the complexity of the optimal distribution problem that we realized in experiments, and then present some related work.

During our experiments, we found that it is not a trivial problem. Suppose there are n activities and m data items in a process, one activity can use several data items and one data item can also be used by several activities, each of them can be distributed either on the cloud-side or at user-end. There should be $2^m \cdot 2^n$ possible distributions of activities and data items totally.

Therefore, if we use Brute-Force method to find the optimal distribution, the time complexity will be $O(2^{m+n})$ due to the large search space.

However, by careful observation, we can find that the cost of each activity only depends on its own distribution and its relevant data's distribution. So, given the data distribution, if the cost made by each activity's distribution is optimal then the total cost is optimal. Thus, we can first arrange all the data with $O(2^m)$ time complexity, and then find the optimal distribution of each activity in n steps. Therefore, the complexity of the problem comes down to $O(2^m \cdot n)$. It is better, but still has exponential complexity with the number of data items in a process. Fig.8 demonstrates the search space of possible distributions for a process with 3 data items and 4 activities.

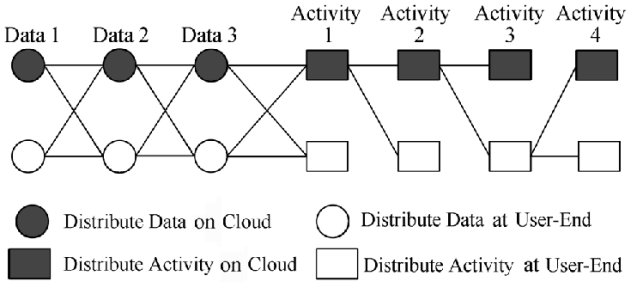

Fig.8. Search space of possible distributions for a sample process.

In practice, most of the business processes only have a number of data items from 1 to 25, so the algorithm is acceptable. Although most of the processes only involve a small set of data, we still have to consider the complicated one. For those complicated cases which have a large number of data items, we try to use some heuristic algorithms to produce a solution which is approximate to optimization because such an integer optimization problem is generally an NP-hard problem.

Genetic algorithms[22] can be used in a computer simulation in which a population of abstract representations of candidate solutions to an optimization problem evolve toward better solutions. According to the characteristic of our problem, genetic algorithm was chosen in the experiment. We use "1" to represent a data item distributed on the cloud-side, and "0" to represent a data item distributed at the user-end. The fitness function is the sum of weighted value of time, privacy risk and money cost as we mentioned in Subsection 4.2. To better integrate our deploying tools which was written in Java, JGAP[23], an open source genetic algorithm package written in Java, was utilized to implement optimal distribution of complicated processes.

Cloud-based BPM is becoming a trend. More and more work on cost-aware cloud-based applications and data privacy in cloud is undertaken. On the one hand, a number of researches explore how business processes can be more cost-effective and more efficiently managed in Grids and clouds[24-26]. These researches mainly focused on how to reduce the cost without any impaction on performance for a cloud-based BPM. On the other hand, with the booming of cloud computing nowadays, the data privacy and security issues in the cloud computing are gaining attentions by the researchers. These researches focused on cloud platform's security and cryptographic protocol issues such as Bertino's work[16] and Krautheim's work[27]. However, with these approaches, user's data still have to be stored on the cloud-side physically, and the process owners still face the critical choices of whether to trust the security protocols or not for fear of losing of control of their sensitive data.

All of the above-discussed studies have not taken into account both the cost optimization problem and the data privacy problem at the same time. We believe only when these two requirements have been satisfied, the cloud-based BPM would be accepted by the majority of enterprises. As far as we know, there has not been any research report on this problem. In order to lower the cost of BPM platform and at the same time let users protect and control their sensitive data, this paper proposes an architecture enabling user-end distribution of non-compute-intensive activities and sensitive data. Also we provide a new approach to giving recommendations for users to best distribute the activities and data in synthetically utilizing both user-end and cloud-side resources. Therefore, compared with the above researches, our approach is not only novel, but also more flexible and safe for users.

7 Conclusion

With cloud-based BPM, users can reduce their IT expenditure and enjoy full-fledged BPM based upon the cloud fabric with high availability and scalability. However, to our knowledge, little effort is given to investigate issues of how to utilize user-end resources and how to protect user's data privacy in cloud-based BPM. In the paper, we first analyzed candidate patterns of cloud-based BPM architecture using the PAD model; then, we proposed a new approach to modeling and synthesizing both cloud-side resources and user-end resources, dealing with control data and business data separately, and ensuring optimal distribution of non-compute-intensive activities and sensitive data; moreover, we clarified some design tradeoffs and implementation issues; and finally, we make an initial assessment with an experimental scenario. The proposed architecture of cloud-based BPM with user-end distribution

of non-compute-intensive activities and sensitive data for synthetically utilizing both user-end and cloud-side resources is considered as the major contribution of the paper. Experiments and analysis verify that our approach can protect data privacy and utilize the resources on both sides at lower cost by the right distribution.

As a new technical term, cloud-based BPM can bring many benefits to business users and independent operators of cloud computing platform. Meanwhile, there are still some problems remaining unsolved. There are many kinds of different legacy BPM systems, and they were deployed with different perspectives, scale, approaches, infrastructure assumption and technologies. Thus, it is crucial to enable the collaboration of the processes in them and the processes on cloud-side. We have done some preliminary work on business-end programming: VINCA[28], client-centric scheduling[29] and business process collaboration[30]. As for future work, we will further investigate how to develop a user-centric, light-weight middleware for business process agile integration and collaboration for cloud computing environment. Besides data privacy, we will focus on multi-process collaboration in cloud environment.

Acknowledgements We thank Kai Sun, Qi-Long Sun, Xiao-Wei Zhao, Hai-Lue Lin, Li-Yong Zhang and Guang Ji for their help and suggestions in writing this paper.

References

[1] Turner M, Budgen D, Brereton P. Turning software into a service. *IEEE Computer*, 2003, 36(10): 38-44.

[2] Foster I, Iamnitchi A. On death, taxes, and the convergence of peer-to-peer and grid computing. In *Proc. the 2nd International Workshop on Peer-to-Peer Systems* (*IPTPS 2003*), Berkeley, USA, Feb. 20-21, 2003, pp.10-17.

[3] Armbrust M, Fox A, Griffith R *et al.* Above the clouds: A Berkeley view of cloud computing. UCB/EECS-2009-28, EECS Department, University of California, Berkeley, 2009, http://www.eecs.berkeley.edu/Pubs/TechRpt s/2009/EECS-2009-28.html.

[4] Wil M. P. van der Aalst, Arthur H. M. ter Hofstede, Mathias Weske. Business process management: A survey. In *Proc. Int. Conf. Business Process Management*, Eindhoven, The Netherlands, Jun. 26-27, 2003, pp.1-12.

[5] Krafzig D, Banke K, Slama D. Enterprise SOA: Service-Oriented Architecture Best Practices. Prentice Hall PTR. Upper Saddle River, NJ, USA, 2004, pp.1-17.

[6] Oracle BPM. August 2009, http://www.oracle.com/technologies/bpm/index.html.

[7] Microsoft BizTalk server business process management. Aug. 2009, http://www.microsoft.com/biztalk/en/us/bpm.aspx.

[8] IBM - WebSphere business process management. Oct. 2009, http://www.ibm.com/software/websphere/products/businessint/.

[9] JBoss jBPM. October 2009, http://jboss.com/products/jbpm.

[10] BPEL Open Source Engine. October 2009, http://www.activevos.com/community-open-source.php.

[11] Open source Java XPDL workflow. Oct. 2009, http://shark.enhydra.org/.

[12] Buyya R, Yeo C S, Venugopal S. Market-oriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities. In *Proc. the 10th Int. Conference on High Performance Computing and Communications* (*HPCC-08*), Los Alamitos, USA, Sept. 25-27, 2008, pp.5-13.

[13] IBM — BPM BlueWorks: BPM working in the cloud. Sept. 2010, https://apps.lotuslive.com/bpmblueworks/.

[14] Microsoft SharePoint online, business productivity online suite. Oct. 2009, http://www.microsoft.com/online/sharepoint-online.mspx.

[15] VitriaCloud M3O in the cloud. October 2009, http://vitria-cloud.com/site/.

[16] Bertino E, Paci F, Ferrini R. Privacy-preserving digital identity management for cloud computing. *IEEE Data Eng. Bull*, 2009, 32(1): 21-27.

[17] Bowers S, Ludascher B, Ngu A H H, Critchlow T. Enabling scientific workflow reuse through structured composition of dataflow and control-flow. In *Proc. the 22nd Int. Data Engineering Workshops* (*ICDEW 2006*), Atlanta, USA, Apr. 3-7, 2006, pp.70-76.

[18] Google App engine. October 2009, http://appengine.google.com/.

[19] Extensible messaging and presence protocol (XMPP): Core. Oct. 2009, http://www.ietf.org/rfc/rfc3920.txt.

[20] Jabber ID. Oct. 2009, http://www.jabber.org/index.php/faq/#jid.

[21] Wagener J, Spjuth O, Willighagen E L, Wikberg J ES. XMPP for cloud computing in bioinformatics supporting discovery and invocation of asynchronous Web services. In *BMC Bioinformatics*, 2009, 10(1): 279.

[22] Genetic Algorithm. Oct. 2009, http://en.wikipedia.org/wiki/Genetic_algorithm.

[23] JGAP: Java genetic algorithms package, Oct. 2009, http://jgap. sourceforge.net/.

[24] Yu J, Buyya R, Tham C K. Cost-based scheduling of scientific workflow application on utility grids. In *Proc. the 1st Int. Conf. e-Science and Grid Computing* (*e-Science 2005*), Melbourne, Australia, Dec. 5-8, 2005, pp.140-147.

[25] Singh G, Kesselman C, Deelman E. A provisioning model and its comparison with best-effort for performance-cost optimization in grids. In *Proc. the 16th Int. High Performance Distributed Computing Symp.* (*HPDC 2007*), Monterey Bay, USA, Jun. 27-29, 2007, pp.117-126.

[26] Deelman E, Singh G, Livny M *et al.* The cost of doing science on the cloud: The Montage example. In *Proc. the ACM/IEEE Conference on High Performance Computing* (*SC 2008*), Austin, USA, Nov. 15-21, 2008, pp.1-12.

[27] F. John Krautheim. Private virtual infrastructure for cloud computing. In *Workshop on Hot Topics in Cloud Computing* (*HotCloud 2009*), San Diego, USA, Jun. 14-19, 2009, pp.10-17.

[28] Han Y, Geng H, Li H, Xiong J *et al.* A visual and personalized business-level composition language for chaining Web-based services. In *Proc. the 1st International Conference on Service Oriented Computing* (*ICSOC 03*), Trento, Italy, Dec. 15-18, pp.165-177.

[29] Wang J, Zhang L Y, Han Y B. Client-centric adaptive scheduling of service-oriented applications. *Journal of Computer Science and Technology*, 2006, 21(4): 537-546.

[30] Li H F, Han Y B, Hu S L. An approach to constructing service-oriented and event-driven application dynamic alliances. *Chinese Journal of Computers*, 2005, 28(4): 739-749. (in Chinese)

Yan-Bo Han is a professor at the Institute of Computing Technology and the Graduate University of Chinese Academy of Sciences in Beijing, China. He also serves as the director of the Institute of Service Engineering, Shandong University of Science and Technology in Qingdao, China. His current research interests include Internet computing, services interoperability and composition, dependable distributed systems, business process collaboration and management.

Jun-Yi Sun received his B.S. degree in electronic engineering from Huazhong Normal University, and B.S degree in computer science from Huazhong University of Science and Technology in 2006. He is a Ph.D. candidate in Graduate School of the Chinese Academy of Sciences. His current research interests include workflow scheduling and cloud-based BPM.

Gui-Ling Wang is currently an assistant professor in the Institute of Computing Technology, Chinese Academy of Sciences (ICT, CAS). She received her Ph.D. degree in computer science from Tsinghua University in 2007. Her research interests include services composition, information integration and Web Mashup technologies.

Hou-Fu Li is an assistant professor at the Institute of Computing Technology. He received his Ph.D. degree in computer science from the Graduate University of Chinese Academy of Sciences in 2008. His research interests include service- and event-based business process management and collaboration technologies, and dependable and high productive Internet-based operating system.