A Privacy-aware Virtual Machine Migration Framework on Hybrid Clouds

Hongli Zhang, Panpan Li, Zhigang Zhou, Junchao Wu, and Xiangzhan Yu
School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
Email: {zhanghongli, yzx}@hit.edu.cn, {lipan, zhouzhigang, wujunchao}@pact518.hit.edu.cn

Abstract—With the proliferation of hybrid clouds in both cost-saving and effectiveness, a growing number of users are building their own private cloud. However, private cloud can only provide limited resource, and always resorts to public cloud in order to meet elastic service requirements. Generally, public cloud is operated by commercial service providers (CSPs) who are not completely honest with users to some extent. So, existing hybrid clouds service models are hampered by privacy concerns, and one of the main reasons is that a majority of users’ workloads involves sensitive contents and therefore can’t be directly outsourced to public cloud without proper protection. In this paper, we present a novel framework that makes such privacy-aware VM migration effectively. At First, all the users’ workloads are uploading and running in private cloud VMs then labeling the sensitive and non-sensitive VMs. Then, we design an effective way focusing on how to get VM migration queue without introducing heavy inter-cloud communication traffic after migration. Finally, non-sensitive VMs are migrated to public cloud with utmost effort by live migration premised on minimal service level agreement (SLA) disruption. Experiment results illustrate that our proposed framework is effective while occurring an acceptable communication overhead.

Index Terms—Privacy-Aware; Hadoop; Xen; VM Migration; Hybrid Clouds

I. INTRODUCTION

As an emerging technology trend, cloud computing is expected to reshape in information technology. Meanwhile, the hybrid clouds are being adopted by most enterprise users and is still undergoing a rapid development, with new techniques mushroomed to enable a smoother inter-cloud coordination. A hybrid clouds that is a cloud computing environment in which an organization provides and manages some resources in-house and has others provided externally can tackle transient and great volume of requests efficiently and effectively. Meanwhile hybrid clouds that integrate private cloud and public cloud is increasingly becoming an important research issue [2, 15].

Actually, hybrid clouds represent a combination of a public cloud with the organization’s private cloud, which aim at efficient distribution of the load among the clouds. Usually, a private cloud is operated by a user itself, while a public cloud is operated by an independent commercial CSP instead of fully controlled by user itself. The expansion of rapid hybrid clouds which fully leverage the benefits of both public and private cloud computing due to some reasons. On the one hand, the hybrid clouds service mode allows enterprise users to improve the utilization of their reduced IT infrastructure and thereby reduce their IT costs. On the other hand, users can manually decide on what data and applications should be reside and run within internal private cloud and which should be moved to public cloud.

Despite more hybrid clouds’ attractive at present, unfortunately, as users no longer physically possess the storage of their data in public cloud, the privacy-preserving concern is still a major obstacle on the road for the enterprise users who are considering adoption of hybrid clouds weather or not, due to the cloud users’ data often involves sensitive data [6, 8, 9]. Therefore, in other words user’s data cannot be directly shared with public CSPs without appropriate protection. In addition, today’s commercial public clouds do not offer high security assurance, a concern that has been significantly aggravated by recent incidents about users’ private data linkage of GMail [34] and users’ sensitive data in Microsoft BPOS platform is accessed by unauthorized users [35]. In more serious cases, public cloud is fully controlled by CSP, therefore malicious or curious CSP even may steal or pry users’ sensitive contents. Consequently, attempts to outsource the application involved sensitive data to public cloud are not encouraged [14].

An intuitive idea to solve the above-mentioned security problems is encryption technique, and there are some research achievements in this direction [2, 12, 13, 16, 36]. Whereas, those existing approaches are still not adapted to the need and demand of mass storage and large-scale computing services due to massive encryption and decryption huge cost now. To satisfy this demand, the latest research proposed a new idea that splits user’s tasks into sensitive tasks and public tasks, and then only public tasks are migrated to public cloud [2]. However, the shortage of this method is that tasks are split too small to get efficient benefit.

The above discussions indicate that, in order to meet such massive service demands requires, we propose a VM migration strategy on hybrid clouds to protect users’ private data. Although conceptually simple, this VMs migration form private cloud to public cloud does bring in a few interesting technical issues [1]. First, we should focus on how to recognize users’ private data efficiently.
Second, how to trigger the migration opportunity about which VMs to migrate to public cloud and when is another challenge. Finally, the computation results of user’s applications which already migrate into public cloud should return to private cloud for later combination and sent back to users. And that leads to an even more worrisome problems: due to destruction of VM after migration, how to recoil the execution results of outsourced public cloud VM which involve sensitive information to VM running private cloud at the reduce phrase. In addition to the concern of above technical challenges, how to guarantee the quality of service (i.e., SLA) from the user’s perspective, such as service delay or disruptions guarantee to the end users is equally important as well.

To the best of our knowledge, we study the problem of VM migration on hybrid clouds with privacy-preserving for the first time. We present a privacy-aware VM migration framework, which causes minimal SLA disruption. Moreover, our proposed framework tries to move computation load to public cloud at its best effort. In addition, VM is used as a migrate-unit other than a task, so that it is more robustness and is suitable for legacy user’s applications without redevelopment. Performance and costs analysis on large-scale dataset in our prototype system shows that our proposed framework indeed introduces lower additional communication and computation overhead.

The rest of the paper is organized as follows. In Section II, we overview the system and design objectives. Section III describes our proposed framework in detail. Furthermore, focusing on privacy-aware VM labeling, we provide a sensitive data recognition and VM labeling method, and propose VM re-mapping table in order to meet the demand of VM attributes changing after migration in Section IV. Section V describes the VM migration policy in detail aiming at minimal SLA disruption during migration. We use experiences to validate the performance and effectiveness of the framework in Section VI. Finally, we overview the related work and give the conclusion of the whole paper in Section VII and Section VIII respectively.

II. OVERVIEW

In this section, we first explain the background and basic technologies of our work and then present objectives in our research design.

A. Background

Figure 1 presents the basic scenario of our work, where hybrid clouds consist of private cloud and public cloud. Usually, the private cloud is built by user itself, that is to say, private cloud is honest as it is fully controlled by user. However, public cloud operated by commercial CSPs is rent by users, which could cause potential security problems by malicious or curious CSPs. In order to protect users’ sensitive data or applications from leaking to public CSPs, the general idea of our proposed framework is that we must recognize VMs which are not involving sensitive data or applications in private cloud first, and then migrate some or all of them to public cloud by a certain migration policy. Besides, the migration policy is another core design in this research work which effectively reduces communication traffic between private and public cloud.

B. Live Migration

In order to meet minimal SLA disruption as possible, so it is necessary that VM migration should not heavily affect running applications, and our framework performs that VM’s local current running status transfer along with the continuing migrated VM operation. Therefore, based on those considerations, we adopt live migration method to achieve our aim. Moreover, live migration allows a server administrator to move a running VM from private cloud to public cloud without disconnecting the user. The reason for this is that when down-time of a VM during a live migration is nearly zero or a few millisecond which is not affecting running application or noticeable by end user. For a successful and complete live migration, the memory, storage, and network connectivity of the VM need to be migrated to the destination. In other words, our framework should design a monitor and collector those information that is essential during live migration.

C. Design Objectives

The original Hadoop and Xen framework do not support migration of VMs with different security levels over the hybrid clouds directly. We develop a novel framework to enhance Xen to make it suitable for performing a privacy-aware VM migration over hybrid clouds. More specifically, our system has been designed to achieve the following objectives:

High privacy assurance. Only the non-sensitive VM, as detected by our system, can be handed over to the public commercial cloud, workload sensitive attribute recognition of VM as well as VM migration launch should be done in private cloud.

Minimal SLA disruption & high efficiency. VM migration over hybrid clouds will naturally cause additional overhead and system “down-time”. In order to guarantee the usability of system, our strategy must assure minimum disruptions of SLA from end user’s perspective.

Moving workload to the public cloud as possible. When non-sensitive VMs migrating to public cloud, we need to move as much computation to the public cloud as possible, under the premise of user’s private data is preserved.

More Generality. The system only can make full use of general service interfaces of commercial public cloud without modifying or rebuilding APIs of public cloud in practice.
III. SYSTEM ARCHITECTURE

In this section, we elaborate our design of the privacy-aware VM migration on hybrid clouds framework over Xen VM monitor.

A. Overview Architecture

Figure 2 shows the overall architecture of the proposed system. For easy presentation, operations in Xen discussed in following section are not shown in this figure. Consistently with typical hybrid clouds scenarios, in this prototype there are two main parts:

Private cloud Private cloud is usually built by user itself. That is to say user has a full control of private cloud. So we can assume it is honest but just having limited computing and storage capacity. Specially, private cloud is based on Xen VMM over Hadoop cluster. As a master/slave architecture framework, a Hadoop system contains a single NameNode, a master server that manages the file system namespace and regulates access to files by users. Moreover, there are some DataNodes, usually one per node in the cluster, which manage storage attached to the nodes that they run on. JobTracker is the daemon service for submitting and tracking MapReduce applications in Hadoop. There is only one JobTracker process runs in Hadoop system. A TaskTracker is a slave node daemon in the cluster that accepts tasks from a JobTracker [6].

Public cloud Public cloud operated by commercial CSP is rent by users in general. Users no longer physically possess the storage of their data outsourced in public cloud. And there do exist various motivations for CSP to behave unfaithfully towards the cloud users due to their sensitive outsourced data. However, on the other hand public cloud possesses massive computing and storage capacity. In general, the public cloud can provide dynamic provision of resources involving the computing and storage capacities [21].

B. VM Status Integration

For demonstration purposes, we only describe an example Xen in private cloud as shown in Figure 2. However, we need to integrate all the Xen status in order to make migration policy decision in large-scale real distributed cloud system (discussed in Section V).

Generally, each Xen system has a single privileged OS called Dom0, which is responsible for starting and managing the other unprivileged OS instances. Dom0 is a unique VM running on the Xen hypervisor that has special rights to access physical I/O resources as well as interact with the other Guest VMs. Meanwhile, Dom0 has the tools necessary to manage other domains. DomU guests have no direct access to physical hardware on the machine as a Dom0 does and is often referred to as unprivileged. Besides Xenstore is design for information storage space shared between domains in Xen [3, 4], and Xenbus as an interface to interact with Xenstore offers bus abstraction for virtual driver such as split device driver. Usually, Xenbus plays the role of interface in reading or writing information such as configuration or state in Xenstore, and xenstore-read displays the value of a XenStore entry. Therefore, we use xenstore-read to read keys and values in Xenstore to get the current status of a specific VM [28, 29].

IV. PRIVACY-AWARE VM LABELING

In this section, we propose a method about how to recognize sensitive data workload and label VMs by workload. Then we utilize VM re-mapping table to fulfill the VM remapping mechanism after migration.

A. Data Labeling and Uploading

Users’ datasets always involve mixture of sensitive and non-sensitive contents in general. Firstly, sensitive and non-sensitive datasets should be labeled respectively by users itself in order to recognize VM security attribute before migration [17]. Inspired by [2], we adopt this method about data labeling strategy, and the detail without the scope of our work. To put is simple, data labeling can be done manually when only a very small amount of sensitive contents is involved or implemented such a batch processing tool as a string scanner that searches a given dataset for the keywords or other text patterns which describe sensitive user information. Similarly, in case of the sensitive contents are detected, a security label is used to record the location of the information: the one built into our prototype is a tuple (filename, offset, length). Once this is all done, then user can outsource data to HDFS within private cloud.
where further non-sensitive VMs replicate and migrate to public cloud.

**B. VM labeling Method**

1) **Sensitive Data Recognition**

Before presenting the details of our sensitive data recognition, we study a few basic techniques used in our proposed framework.

Grant tables are provided by Xen for sharing memory between domains. Entries in the grant table are identified by GR (Grant References), and it acts as a capability which the grantee can use to perform operations on the granter’s memory [5]. Generally, guest OSs access devices via device channel links with Isolated Driver Domains (IDDs). Figure 3 shows a guest OS requesting a data transfer through a device channel. Specifically, P2 page is placed on device channel at first. Then, IDD removes GR and sends pin request to Xen by steps 2 and 3. After that, Xen looks up GR in active grant table, and GR validate against Guest (if necessary) in 4 and 5 steps. Finally, Pinning is acknowledged to IDD which sends Direct Memory Access (DMA) request to device by steps 6, 7 respectively.

![Figure 3. A grant handling process by grant table mechanism](Image)

The Xen front and back-end driver communicate with each other by using two buffers: one for packet transmission and another one for packet reception. The ring buffers are implemented in Xen via grant tables and event channel Xen grant tables is a mechanism to share memory pages between VMs by page mapping or page transfer [19, 20].

Page mapping among domains is mainly achieved by \texttt{GNTTABOP\_map\_grant\_ref} function [30]. This function by given a GR from another domain, map the referred page into the destination domain’s address space. During this process, the key data structure gnptab\_map\_grant\_ref contains two related members named host\_addr, domid. Member variable host\_addr denotes memory space that is will be mapped for destination domain, and domid is indicated the domain ID of destination domain. Hence, host\_addr and domid are used to label VMs with different security levels in following section. Similarity to page mapping mechanism, page transfer process is maintained by \texttt{GNTTAPOP\_transfer}, and ownership of a page frame is transferred to destination domain by this mechanism. Moreover, the gnptab\_transfer structure is mainly focus on the process. In gnptab\_transfer structure, mfn describes the memory space to be transferred to destination domain, and domid denotes the destination domain ID.

We could detect the data security level label during memory page mapping and transferring process. For this, we modify \texttt{GNTTABOP\_map\_grant\_ref} and \texttt{GNTTAPOP\_transfer} by adding a data security label recognition functional module. This proposed model by checking the host\_addr, mfn and domid to detect which VMs are assigned the sensitive contents while which are assigned non-sensitive contents. By this result, then, we label the VM into two categories: sensitive VMs and non-sensitive VMs. It’s important to note that grant tables are generally faster than bounce buffers, as the VM can directly access the memory page by DMA mechanism.

2) **VM labeling**

We can detect and label the VMs which are assigned sensitive data or non-sensitive data by above method. Therefore, all the VMs are divided into two categories: sensitive VMs and non-sensitive VMs. Specifically sensitive VMs’ workload involves sensitive data while non-sensitive VMs’ workload only includes non-sensitive data. For consideration of data privacy, the sensitive VMs should run in private cloud. However, to meet the design objectives, the non-sensitive VM selectively migrate from private cloud to public cloud by a certain policy (discussed in section V).

**C. VM Re-mapping Table**

In Hadoop system, The NameNode makes all decisions by regarding replication of blocks and it periodically receives a Heartbeat and a Blockreport from each of the DataNodes. On the other hand, each DataNode sends a Heartbeat message to the NameNode periodically. This bidirectional heartbeat communication method is down through DatanodeProtocol communication protocols. Similarly, Tasktrackers runs a simple loop that periodically sends heartbeat method calls to the JobTracker by InterTrackerProtocol. Heartbeats tell the JobTracker that a TaskTracker is alive, but they also double as a channel for messages. InterTrackerProtocol protocol is use to support the heartbeat message communication between JobTracker and TaskTracker.

However, DomID and IP of Guest OS will be changed after VM migration from private cloud to public cloud [18, 23]. Therefore, two problems arise with changing of VM key attribute. On the one hand, heartbeat between JobTracker and TaskTracker lost the communication addressing modes. On the other hand, the result of VM which has be migrated to public cloud not recoil to private to accomplish the Reduce phrase. To solve the above-mentioned problems, we introduce re-mapping table in Xen. The item of re-mapping table is a tuple \(<\text{SourceDomID, Destination IP, DestinationDomID}>\), DatanodeProtocol and InterTrackerProtocol are encapsulated TCP/IP protocol respectively in order to act as heartbeat between inter-cloud communications after VM migration.

**V. VM Migration Policy**

In this section, we introduce a migration opportunity policy to achieve above-mentioned minimal SLA disruption objective. The main contribution of this paper lies in this novel technique to reduce inter-cloud communication migration policy. The core design of our prototype is how to determine the VM migration...
opportunity that refers to which non-sensitive VMs to migrate and when. Migration opportunity heavily affects SLA during the migration process. So, non-sensitive VMs which will badly incur inter-cloud traffic communication should run in private cloud.

A. Inter-VM Network Traffic

In order to formulate the minimal SLA policy migration policy, getting the network traffic of each VM in private cloud is a first requirement, then to decide which to migrate. As stated before, network communication between Dom0 and DomU mainly relies on the memory page transfer that uses a zero-copy page remap mechanism to implement the data transfer in memory [10, 31]. So, we can capture the network traffic of VM by counting mfn in gnttab_transfer. The network traffic of every VM can be easily got through assembling a small database to record the total mfn of each VM.

B. Migration Policy

Definition 1: \( \text{bandwidth}_{VM}^{\text{available}}, \text{bandwidth}_{VM}^{\text{actual}} \) mean available bandwidth and actual throughput of a specific VM, and we define heavily traffic of this VM subject to:

\[
\frac{\text{bandwidth}_{VM}^{\text{actual}}}{\text{bandwidth}_{VM}^{\text{available}}} \geq \lambda
\]

(Generally \( \lambda = 0.9 \) [40])

Theoretically, how to select VMs which are to be migrated to public cloud through network traffic factors can be viewed as an undirected graph selection problem. We are given a traffic undirected graph \( G = \langle V, E, l \rangle \). The vertices \( V \) represents the VMs. Edges \( E \) represents the network collection between VMs and the labels \( l \) on them denote the network traffic between them. A traffic load threshold is determined based on the system state. According to this threshold traffic load can be classified into two generic groups: heavy traffic and light traffic. For easy presentation, traffic between sensitive VM is not shown in the framework since all the sensitive VMs only should run in private cloud without considering migration problem. Therefore undirected graph \( G \) ignores the all traffic edges between sensitive VMs. Let non-sensitive VMs be \{\text{non-sensitiveVMs}\}, the traffic undirected graph just shown as in Figure 4, and we set \{\text{non-sensitiveVMs}\} = \{VM_i\}, (i = 1,...,12). And, we use solid and broken lines to denote the heavy traffic and light traffic respectively.

In order to reduce the private cloud load, and make full use of public cloud resources effective, so our proposed policy supposes that non-sensitive VMs migrate to public cloud as possible on causing the least inter-cloud communication overhead premises. Actually, if some non-sensitive VMs do indeed under heavy workload with heavy communication load with VMs running in private cloud, those VMs should not migrate into public cloud. Based on this idea, we give our proposed policy in detail below.

In Figure 4, firstly we select non-sensitive VM set which has heavy traffic communication with sensitive VMs denoted as \{\text{non-sensitiveMV}_{\text{heavy}}\}. For simple description, we assume \{\text{non-sensitiveMV}_{\text{heavy}}\} = \{VM_4, VM_{10}\}. Those VMs should not migration to public cloud on account of heavily traffic with VMs within private cloud. \{VM_4, VM_{10}\} is selected from non-sensitive VMs set as non-migration candidates, then the next step is to find its neighbor which have heavy traffic with them.

Next, as for every VM node in \{\text{non-sensitiveMV}_{\text{heavy}}\}, we employ depth-first walk approach to traverse the traffic undirected graph \( G \) by heavy traffic edges aiming to find heavy traffic linked VMs node. Walking the results by \( VM_4 \) and \( VM_{10} \) are \{VM_5, VM_6, VM_3\} and \{\emptyset\} respectively. Then, Get the union of above VM set. That is

\[
\{VM_4, VM_{10}\} \cup \{VM_5, VM_6, VM_3\} \cup \{\emptyset\}
\]

= \{VM_4, VM_{10}, VM_5, VM_6, VM_3\}

So, the pre-migrate VM set is:

\{\text{non-sensitiveVMs} – \{VM_4, VM_{10}, VM_5, VM_6, VM_3\}\}

= \{VM_1, VM_2, VM_7, VM_8, VM_9, VM_{11}, VM_{12}\}

It is obvious that VM_8 and VM_{12}, VM_9 and VM_{11}, take up massive network traffic as well. However, this massive network traffic is among the pre-migrate VM set, and will become the inner public cloud communication traffic instead of occupying inter-cloud network communication traffic after migrating to public cloud totally. In addition, VM_4 with light traffic communication with sensitive VM which deployment in private cloud is still migrated from private cloud, on account of only causing little traffic load between inter-cloud as well as making full use of public cloud resources.

And there is another related problem after getting the pre-migration VMs is the order of migration. It is obvious that VM migration consumes certain resource. In Figure 4 scenario, there is heavy traffic communication between VM_8 and VM_{12}, and both of them will migrate to public cloud. Let’s suppose that VM_8 is migrate to public cloud, which is the order to be migrate is other VM instead.
of VM_{12}, So, VM_8 and VM_{12} will be split between public cloud and private cloud compulsively before VM_{12} is migrated into public cloud platform. That is the heavy traffic communication will aggravate inter-cloud network traffic overhead until this situation end. Therefore, we try to find a pretty order of migration VM queue and make heavy traffic communication VMs as neighbor for causing further minimal SLA.

If we assume that the random initial pre-migrate VM queue is shown as in Figure 5(a), then the ordering pre-migrate VM queue is shown as in Figure 5(b).

![Figure 5. Ordering the pre-migrate VM queue](image)

Through above-proposed method, VMs which have heavy traffic communication within private cloud VMs were not migrated, so this idea can effectively reduce the inter-cloud communication, in other word, migration policy causes minimal SLA disruption. More generally, we give our algorithm about migration policy below.

**Algorithm 1 VM Migration Policy**

Input: sensitive VM set: \{sensitive VMs\} including M nodes
non-sensitive VM set: \{non-sensitive VMs\} including N nodes

Output: pre-migrate VM array

1: for each VM in M
2: for each VM in N
3: if VM in N has traffic with M, building G=<V,E,l>
4: end for
5: end for
6: for each VM node in N
7: finding every heavy traffic node with M, denoted as \{N_{\text{heavy}}\}
8: end for
9: for each node in \{N_{\text{heavy}}\}
10: depth-first walk in traffic undirected graph, denoted as \{N'_{\text{heavy}}\}
11: end for
12: \{pre-migrate VMs\} = \{N\} - \{N_{\text{heavy}}\} - \{N'_{\text{heavy}}\}
13: ordering pre-migrate array into neighbor node by heavy inter-communication traffic
14: return pre-migrate VM array

Algorithm 1 shows the pseudo-code of migration policy in detail. The algorithm includes four main steps. Line 1-5 builds undirected graph G about VM traffic topology. Based on heavy traffic, line 6-8 extracts VM set which have directly heavy traffic communication with sensitive VMs. In order to get \{N_{\text{heavy}}\}, we start with \{N_{\text{heavy}}\} to do depth-first walk in G, as shown in line 9-11. Line 12-14 is design to get the pre-migrate VM array.

VI. EVALUATION

In this section, we report the evaluation study of our privacy-aware VM migration framework.

A. Experiment Environment

Here, we describe the setting of our experimental study including the datasets we used to evaluate our proposed prototype, and our hybrid clouds computing environment based on Eucalyptus [7]. We implemented our proposed prototype on a serious of physical nodes, each with cores of Xeon E5506, 2.13GHz CPUs, 16G dual-channel 1333GHz memory, 2T 7200 RPM disk and also Ubuntu 12.04 LTS OS with Xen 4.0.3, experiment setting as Figure 6. Our prototype includes the main components: (1) the sensitive/non-sensitive MV recognition and labeling modules, and (2) VM migration policy. Both components adhere to the design presented in Section IV and V.

![Figure 6. Experiment setup](image)

In our study, we ran four Hadoop applications over this prototype to evaluate its performance and effectiveness, which include a data analysis for target marketing, two intrusion detection analyses and two applications for preparing spam detection. The two examples for IDS (intrusion detection systems) were all based on the Cyber Systems and Technology Group Intrusion Detection Evaluation dataset (IDS1 and IDS2 for short) [32]. The 32GB dataset includes the sniffing data from external networks, which was marked as public in our study, as well as that from the internal network, which was considered to be sensitive. Task of IDS1 on the data is design to find all the ports connected by each IP address and IDS2 devices to determine the amount of traffic generated by individual hosts. The other two Hadoop applications are designed to detect email spam about 2.1GB (DES1 and DES2 for short) by a Naive Bayesian classifier for. Specially, DES1 counts the occurrences of a set of words on a spam key-word list and the DES2 counts the total number of words in a large dataset. As [2] discussed, we utilized the published Enron email dataset [38] as private data and a SPAM archive [39] as public data.

In order to record the running status of every physical node and inter-cloud network traffic, Ganglia is deployed on each node to monitor our systems [27]. We evaluate the two aspects of our proposed scheme performance: (1) performance of our migration policy, namely, the outsourced ratio on different percentage of sensitive data, and (2) effectiveness of proposed migration policy, in other words, decreasing degree of inter-cloud communication hybrid clouds after migration.

B. Experiment Result

In our experiment, we evaluated both the performance and effectiveness of our proposed framework.

1) Performance

We employ the above-mentioned four applications that compute outsourced ration to analyze the performance of our execution platform in the presence of different sensitive/non-sensitive data mixtures. Specifically, we implemented the sensitive data was uniformly distributed in data set, since it can represent the worst case scenario with the most negative impact on the performance of our
system. Under this scenario, we gradually increase the proportion of sensitive information from 10% to 50% to evaluate the amount of the outsourced ration of data to public cloud. The outcomes of this study are illustrated by Figure 7.

From Figure 7, what we can see here is that the outsourced ration tends to decrease as percentage of sensitive data rise. However, noted that outsourced ration is always greater than the percentage of sensitive data. The main reason for this result can be accounted that we maintain some non-sensitive VMs in private cloud in order to cause minimal inter-cloud communication traffic during migration process as discussed in section V. Taking IDS1 as an example, the workload of private cloud gradually decreases from about 76% of outsource ratio, when 10% of the dataset was sensitive, to about 32% of outsource ratio, when 40% of the data was private, denotes that performance of proposed framework. In addition, the outsource ratio drops a bit when the ratio of the sensitive data went to 30%~50%, slowly when 20%~30%, and probably due to the randomness in execution. Meanwhile, the different sampling data sets, i.e., different number of sampled data items, are also a variable factor that affects the outsourced ratio. This figure also denotes that those additional cost acceptable. Therefore, we can also conclude that all of them successfully moved a large portion of computation to the public cloud, in accordance with the ratio of the private information within the individual datasets they worked on.

2) Effectiveness

The main goal of our migration policy is to minimal SLA disruption after migration through adding as least inter-cloud communication as possible. We design two experimental scenarios in order to verify the viability and effectiveness of migration policy. The first, we deploy our migration policy in system and the other one is not. In the following, we conduct two sets of experiments and measure the performance of our proposed migration policy. In following experiment, solid lines depict results when employed migration policy, while broken line represent results which all the non-sensitive VMs migrate to public cloud. Similarity to above experiment, we increase the proportion of sensitive information gradually from 10% to 50% to verify of our effectiveness of migration policy.

In the first set, we compare with total execution time of 4 different data sets with different percentage sensitive/non-sensitive data in two experiments scene, with which all non-sensitive MVs migrate into public cloud with which employed our proposed migration policy settings respectively.

Figure 8 shows that total execution time is less when employing our migration policy due to not involving heavy inter-cloud communication. For the consideration of VM migration cost, the number of migrated VM determined by our migration policy is less than the number of all non-sensitive VM. So, we should realize that there is more time is consumed by more migrated VMs themselves in this experiment result. Those experimental results can also be noted that efficiency of our migration policy is relatively high. In other words, our migration policy provides a balance parameter, between migration cost and service capacity, to satisfy minimal SLA disruption. Finally, by this figure we also can see that our strategy is becoming more advantages along with an increasing percentage of sensitive data. Therefore, this would explain our strategy facilitate massive cloud service.

In the second set, we continue to compare network traffic load in above-mentioned two experimental scenarios. The main purpose is to verify the effectiveness of our proposed migration policy. To be sure, it is obvious that VM live migration over private cloud and public cloud generates considerable network traffic. However, in this scenario, we only consider the network communication cost between VM, so we need to omit the network traffic caused by VM migration at first. To this end, the approach we are taking is that for a certain sensitive percentage sensitive/non-sensitive data, we calculate network traffic cost by live migration VMs denoted as \( n_{\text{vm}} \) and total network traffic \( n_{\text{total}} \), so communication network cost denotes as \( n_{\text{com}} = n_{\text{total}} - n_{\text{vm}} \). To sum up, the results are presented in Figure 10.

As we can see from Figure 10, our migration policy not only significantly improves transmission rate, but also shortens total migration time as well. This saving in bandwidth consumption can significantly improve the hybrid clouds’ service capabilities given the fact that inter-cloud bandwidth is typically limited in reality.

Figure 9 also illustrates that all dataset applications achieve similar low additional network traffic at all times.
compared with that all non-sensitive VMs migrate to public cloud, which greatly enhances our service quality target. This saving in inter-cloud communication traffic can significantly improve the performance of our framework. The advantage of proposed migration policy is effective in reducing inter-cloud communication traffic after migration. Therefore, the experimental results offer strong evidences that our speculative migration opportunity strategy indeed work.

Figure 9. Network traffic on different sensitive data ratio

VII. RELATED WORK

Little effort has been made to facilitate security and privacy-preserving on hybrid clouds system. To the best of our knowledge, this work is the first system that privacy-aware VM migration on hybrid clouds.

From the architectural perspective, the privacy-preserving on hybrid clouds can be handled by different approaches. Most previous hybrid clouds security research focus on data security. [8] proposes a data privacy model for cloud computing is provided in which sensitive and non-sensitive data are maintained separately. In addition, an authentication monitor is introduced. In order to maintain data privacy, however, this approach cannot make full use of public cloud because without considering of migrate the heavily load to public cloud.

In terms of the technologies used to data security migration on hybrid clouds, we benefit from the work down in [2]. Sedic is a system that leverages the special features of Mapreduce to automatically partition a computing application according to the security levels of the data it work on by modifying the Mapreduce distributed file system, and arrange the computation across a hybrid clouds. However, this migration process is triggered by task as a migrate-unit, so that this method only own limited availability, because the commercial public loud supporting VMs as service API mostly. In contrast, our solution designs VM as a migration-unit more consistent with actual environment. Other work in this direction (e.g., [11, 22, 24]) can be used to our proposed scheme effectively.

Recently, researches on services migration between hybrid clouds have received much attention [11, 24, 25]. [25] presents an optimization framework for dynamic, cost-minimizing migration of contents distribution service in to a hybrid clouds infrastructure by employing Lyapunov optimization. [25] enables federation between on-and off-premise infrastructure for hosting and Internet-based application. The core design of this scheme proposed a fast frequent data item detection algorithm, which enables factoring incoming requests not only on volume but also on data contents. [11] aims to highlight and emphasize the purpose of the optimization engine component in the context of hybrid clouds scheduling. OpenNebula [26] is effective in managing clustered services, which is used to deploy and manage the back-end nodes of a Sun Grid Engine compute cluster and an Nginx Web server on both local resources and an external cloud. [9] addresses the construction of a collaborative integrity verification mechanism on hybrid clouds to support the scalable service and data migration. However, this scheme user’s sensitive data which should not migrate to public cloud. Different from theme, in this paper, we consider a more complex and practical problem about privacy-aware VM migration scheme on hybrid clouds. Both that recognizing the sensitive data by Xen and the intuitive migration policy makes migration process just causing minimal SLA disruption as well and using public cloud as possible. Perhaps more importantly, in our proposed scheme, we focus on the case that our policy should bring minimal SLA disruption during migration process, instead of all non-sensitive VMs migrating to public cloud.

VIII. CONCLUSION

Commercial public cloud services enable their users to process massive storage and large-scale computing application with a low cost. This benefit, however, brings serious privacy security risks: cloud users’ data often involves sensitive data which cannot upload directly to commercial public clouds without high security assurance. For this reason, in this paper, we presented a privacy-aware VM migration framework on hybrid clouds. We design and implement a data label recognition functional modules to detect the data security level by modifying memory sharing process between domains. As for different security level data assigned to VMs, we label the sensitive VMs and non-sensitive VMs. Then, in order to reduce heavy additional inter-cloud communication incurred by migration to public cloud, an effective VM migration policy is proposed. Extensive analysis shows that our proposed scheme is secure and efficient.

As our future work, we try to extend our migration a more effective one. To this aim, we intend to design quick sensitive data recognition functional module in Xen, and more effective migration policies to guarantee highly usable privacy-aware VM migration.

ACKNOWLEDGMENT

This research was partially supported by the National Basic Research Program of China (973 Program) under grant No. 2011CB302605, the National High Technology Research and Development Program of China (863
Program) under grant No. 2011AA010705, the National Science Foundation of China (NSF) under grants No. 61100188 and No. 61173144. We also thank the anonymous reviewers for their helpful feedback and suggestions.

REFERENCES


Hongli Zhang born in 1973, received her BS degree in Computer Software from the Sichuan University on 1994, and received her MS and PhD degrees in Computer Architecture from the Harbin Institute of Technology on 1996 and 1999, respectively. Her research interests are focused in the area of network security, Internet measurement, and network computing.

Panpan Li born in 1983, Ph. D. candidate in Harbin Institute of Technology, E-mail: lipan@pact518.hit.edu.cn. His research interests include cloud computing, cloud security, network security.

Zhigang Zhou born in 1986, Ph. D. candidate in Harbin Institute of Technology, E-mail: zhouzhigang@pact518.hit.edu.cn. His research interests include cloud computing, cloud security, network security.

Junchao Wu born in 1988, Master candidate in Harbin Institute of Technology, His research interests include cloud computing, cloud security.

Xiangzhan Yu Ph.D., associate professor in Harbin Institute of Technology, His research interests include Computer Network and Information Security, Large-Scale Distributed Storage Technology.

© 2014 ACADEMY PUBLISHER