Interoperability in the Heterogeneous Cloud Environment:
A Survey of Recent User-centric Approaches

Ibrahim Mansour  Reza Sahandi  Kendra Cooper  Adrian Warman
Bournemouth University Bournemouth University Bournemouth University IBM CDS LEAD
Fern Barrow, Poole Fern Barrow, Poole Bournemouth, UK Hursley Park, Winchester
Bournemouth, UK Bournemouth, UK Bournemouth, UK Hampshire, UK
+447752703995 +441202965258 +441202961126 +441962819176

ABSTRACT
Cloud computing provides users the ability to access shared, online computing resources. However, providers often offer their own proprietary applications, interfaces, APIs and infrastructures, resulting in a heterogeneous cloud environment. This heterogeneous environment makes it difficult for users to change cloud service providers; exploring capabilities to support the automated migration from one provider to another is an active, open research area. Many standards bodies (IEEE, NIST, DMTF and SNIA), industry (middleware) and academia have been pursuing approaches to reduce the impact of vendor lock-in by investigating the cloud migration problem at the level of the VM. However, the migration downtime, decoupling VM from underlying systems and security of live channels remain open issues. This paper focuses on analysing recently proposed live, cloud migration approaches for VMs at the infrastructure level in the cloud architecture. The analysis reveals issues with flexibility, performance, and security of the approaches, including additional loads to the CPU and disk I/O drivers of the physical machine where the VM initially resides. The next steps of this research are to develop and evaluate a new approach LibZam (Libya Zamzem) that will work towards addressing the identified limitations.

Categories and Subject Descriptors

General Terms

Keywords

1. INTRODUCTION
There is an accelerating trend in adopting cloud computing services. According to Gartner [1], cloud infrastructure and services will make up the majority of IT budgets in businesses by 2016. Gartner [1] reports that about 50% of large enterprises will be using hybrid cloud architectures by the end of 2017. Despite the notable upwards trend, cloud computing has security, reliability and interoperability issues [5] [32]. In 2013, for example, Amazon’s US-EAST availability region remained unavailable for 59 minutes, resulting in users in U.S.A. and Canada who could not access Amazon.com and Audible.com. The reported loss was about $1,100 in net sales per second. If customers’ services had been able to rapidly become available by migrating to another provider without paying a substantial cost, then the consequences would have been less disastrous.

Research exploring techniques to migrate from one provider to another remains an active research area. Zhizhong Zhang et al. [3] conducted a survey on the lack of interoperability within the cloud at the IaaS level, open source cloud projects (i.e., OpenStack and OpenNebula), cloud standards, and a user-centric solution called Xen-Blanket [4]. The survey used a criteria aiming for IaaS interoperability, but it was not clear how criteria factors interact with each other. However, it did not include any criteria to assess user-centric approaches at any level, including live migration of VMs to the cloud. Adel Nadjaran et al. [6] conducted a broad survey on cloud interoperability for all levels (IaaS, PaaS and SaaS) within the cloud and related open source projects (i.e. RESERVOIR, mOSAIC and OpenStack [35]). However, the paper did not evaluate any user-centric approaches to facilitate interoperability or approaches that could support live migration of VMs. In addition, an important project, Ubuntu OpenStack Interoperability Lab (OIL), was not included in the analysis. In 2015, OpenStack interoperability press announced that 32 companies signed up to adhere to OIL guidelines. Moreover, OpenStack is one of the widely deployed open source cloud projects, which is supported by about 500 companies and 23,000 individuals across over 150 countries [37].
Much work has been done to provide live migrations of VMs to and within the cloud with minimum service interruption [7] [17] [24]. Live migrations often require the following [26]: memory state transfer between anonymous hosts, access of VMs to the storage at the destination host, without sharing storage between source and destination hosts; and access of the VM to the host’s LAN at the destination without the two sites sharing the LAN.

In this paper a novel survey is presented analysing three recent, user-centric approaches to achieve the live migration of VMs. The comparison criteria span performance, flexibility, and security quality of service (QoS) attributes. For example, a security criterion identifies which encryption algorithm, if any, is to ensure data privacy during the migration; a flexibility criterion assesses the variety of hardware platforms that are supported; and a performance criterion is an assessment of whether or not the migration is imperceptible to the VM and VM users.

The structure of the remainder of this paper is as follows: Section 2 a brief summary of the state-of-the-art in cloud interoperability is presented, highlighting interoperability issues, alternative categories of approaches proposed, and the need for the live migration of VMs across the cloud in the absence of support for cloud interoperability at the IaaS level. Section 3 provides an analysis of recently proposed live, user-centric cloud migration approaches, including a summary discussion of the results. The conclusions and future work are presented in Section 4.

2. STATE-OF-THE-ART IN CLOUD INTEROPERABILITY

One of the greatest challenges facing longer-term adoption of cloud computing services is interoperability, as this is necessary to provide cloud providers’ services such as cloud bursting, cloud federation, servers’ underutilization, maintenance and cease operations [14] [30]. To provide these services, live VMs migration is required within and between the clouds. Cloud interoperability approaches can be viewed as multi-layered models, where every layer has to interoperate with the next layer and with its counterpart in another provider. Cloud interoperability at the Platform as a Service (PaaS) and Software as a Service (SaaS) levels are reliant on the Infrastructure as a Service (IaaS) level, which indicates interoperability at the IaaS level is of key importance [3].

2.1 Cloud interoperability issues and benefits

Cloud computing has been providing considerable capabilities for scalable, highly reliable, and easy-to-deploy environments. However, the potential of interoperable cloud environments is even greater for both providers and users. Some of the benefits may be [6]:

- Previous incidents express the need for disaster recovery using either cloud federation or cloud bursting [3]. To enable cloud providers continue delivering services, even in similar circumstances, interoperability between cloud providers is necessary to continue the provision of resources [14] [30].
- In 2014, Amazon launched a new availability zone in Germany, supporting customers in Europe and the Middle East [11]. Currently, providers cannot support applications to predict users’ geographic locations due to the complexity of machine learning algorithms and its cost. Cloud interoperability can enable utilization of the nearest provider’s datacentre, thereby, reducing latency [6].
- Rules and regulations can be a major impediment to interoperability, for instance, providers might have different policies on how long they keep user’s records. Europe has different rules from the USA. Compatibility between regulations can facilitate reaching a common consensus between providers on legal issues (handle contract) [6].

2.2 Approaches to achieve interoperability

Various approaches have been proposed to improve cloud interoperability for all the three levels (IaaS, PaaS and SaaS) [3] [4] [6]. Figure 1 illustrates a taxonomy organized around provider-centric and user-centric approaches [6].

Provider-centric approaches rely on the provider’s agreement to adopt specific standards to achieve a specified level of interoperability. The development and widespread adoption of a set of standards is a long term vision for the community to support cloud federation, cloud bursting, and hybrid clouds [6]. Cloud federation may be facilitated through network gateways that connect public clouds, private clouds and/or community clouds, creating a hybrid cloud computing environment. Cloud bursting uses a set of public or private cloud-based services as a way to augment and handle peaks in IT system requirements at start-up or during runtime [12]. Hybrid clouds use a combination of private and public clouds [29].

As standardization efforts proceed, alternative user-centric approaches to achieve cloud interoperability are being proposed as more immediate, practical solutions. User-centric approaches do not rely on a provider’s (standards based) agreement, as the users either rely on their own in-house IT personnel or a third
party (cloud broker) to achieve interoperability. There are two main possibilities. The first is a cloud broker, which provides a single interface through which users can access and manage cloud services across multiple providers [13]. The second is a multi-cloud, in which users may develop a separate layer to handle heterogeneity in cloud environments [3]. For example, a user may require deploying an adapter layer to communicate with different APIs or a cloud application may need an abstraction library, such as, jcloud and LibCloud libraries [6]. In the following section, the focus moves to analysing recent user-centric approaches for live, cloud migration of VMs at the IaaS level.

3. ANALYSING USER-CENTRIC LIVE, CLOUD MIGRATION APPROACHES FOR VIRTUAL MACHINES

3.1 Methodology

The purpose of this study is to analyse recent, live, user-centric approaches using QoS comparison criteria in performance, flexibility, and security [15]. The comparison criteria for this study were derived from published requirements on successful live, migration techniques [8] [16] [28]. The comparison criteria are summarized in Table 1 and described below in more detail; the method used to select the approaches for inclusion in the study follows.

Table 1 Comparison Criteria

<table>
<thead>
<tr>
<th>Criterion Identifier</th>
<th>Criterion Description</th>
<th>Values Used in Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Performance</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P1</td>
<td>Migration is imperceptible to VM, VM users</td>
<td>Acceptable, Unacceptable</td>
</tr>
<tr>
<td>P2</td>
<td>Predicting provision of required resources to decide whether or not to proceed with migration.</td>
<td>Estimate resources, Reserve resources, both.</td>
</tr>
<tr>
<td>P3</td>
<td>Monitor resource utilization to avoid overutilization and to predict a potential failure.</td>
<td>CPU overhead, network bandwidth consumption. disk I/O drivers overhead, memory dirty pages, downtime migration and total time migration.</td>
</tr>
<tr>
<td><strong>Flexibility</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1</td>
<td>Support multiple hardware platforms</td>
<td>Wide range of hardware drivers. (CPU architecture and Storage)</td>
</tr>
<tr>
<td>F2</td>
<td>Support multiple O/S</td>
<td>Modified O/S Unmodified O/S</td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1</td>
<td>Privacy (Channel encryption)</td>
<td>Advance Encryption Standard (AES)</td>
</tr>
<tr>
<td>S2</td>
<td>Authentication</td>
<td>Hash-based Message Authentication Code using the SHA1 (HMAC-SHA-1)</td>
</tr>
</tbody>
</table>

**Performance criteria.** The first performance criterion, P1 Migration is imperceptible, is related to the availability of the IaaS to the VM during the migration. The VM and hypervisor should not be exposed to delays, halting or crashing during migration across IaaS; in turn a cloud user should not experience interruption of their applications execution on a VM. To accomplish this, the VM and any connected user must not be aware of the migration process [8] [16]. The second criterion P2 Reserve Provision of Required Resources, assesses whether or not the approach provides estimates before the migration commences for resource requirements (i.e. migration downtime, total migration time, network bandwidth and CPU for both the source and destination cloud [8] [33]. The third criterion P3 Resources are monitored, assesses whether or not resource utilization is monitored by both the source and destination machines. If the consumption of the resources exceeds a certain threshold, it may affect applications and performance of the VM, as well as other VMs running on the physical machine. In such a scenario the process may be stopped and rolled backed the VM to original state [33].

**Flexibility criteria.** The first flexibility criterion, F1 Migration is supported for multiple hardware, assesses the variety of different hardware platforms the migration approach supports (i.e. CPU architecture and Storage heterogeneity); the more hardware platforms that are supported, the broader the approach can be applied. The second criterion, F2 Migration is supported for multiple O/S, assesses the variety of different O/S the migration approach supports (Modified O/S (Linux) and unmodified O/S (Windows)). The more O/S that are supported, the broader the approach can be applied [4] [24].

**Security criteria.** The first security criterion, S1 Encryption Algorithm, is used to assess which algorithm, if any, used to encrypt the channel to ensure the transmission is private. The second criterion, S2 Authentication Algorithm, is used to assess which algorithm, if any, is used to authenticate the user requesting the migration [7] [28]. To select the approaches for inclusion in the analysis, a thorough review of the literature was conducted to identify recent, live user-centric migration approaches that explicitly address one or more of the QoS criteria. The sources used in the literature review included electronic databases (IEEE, ACM Digital Library, USENIX The Advanced Computing Systems Association and Springer). Three approaches were found: 1) Supercloud [7]: Opportunities and Challenges (2015). 2) Kangaroo [17]: A Tenant-Centric Software-Defined Cloud Infrastructure (2015). 3) HVX [24]: Virtualizing the Cloud (2013). These approaches are discussed below.

3.2 Analysis results

3.2.1 Supercloud

The Supercloud [7] was developed using resources from a number of major cloud providers, including Amazon EC2, Rackspace, HP Cloud and other private clouds. Supercloud uses nested virtualization (Xen-Blanket [4]) that overcomes cloud heterogeneity; Xen-Blanket leverages the Para-virtualization (PV-on-HVM) drivers on Xen.

**With respect to Performance, P1** The approach achieved relatively acceptable performance, about 1.4 seconds migration downtime [4] [38].

P2 Disk I/O drivers overhead caused by Xen-Blanket reached 30%, which may affect the physical machine and the other VMs residing on that machine [4] [7] [27]. P3 Due to data size, security, cost saving and load balancing, a shared storage accessible by both source and destination was used during the live migration. This exposes the VM to overhead to access its
disk over the network [17] [33]. The transport protocol used in
the migration is TCP/IP. TCP has a slow start that can affect the
migration process and impose extra overhead on the edge
equipment. Consequently, it affects the application’s
performance [18]. A layer 2 tunnel is used to extend a VM
subnet to multiple geographically distributed datacentres. It is
not efficient due to broadcasting all ARP requests to the two
sites resulting in poor performance [19] [34].

**With respect to Flexibility**, F1 Decoupling VM from
underlying system was achieved by using Xen-Blanket approach
[4]. F2 Xen-para-virtualization cannot run unmodified operating
systems (i.e., Windows) [4].

**With respect to Security**, S1 The approach does not utilize an
encryption algorithm. Also, a security mechanism was not used
during the process, so it opens the system to security attacks.

As a result, the transition channel is insecure and data flow is
vulnerable to attacks, such as, ARP/ DHCP/DNS poisoning and
IP/route hijack [28]. S2 The approach does not utilize an
authentication algorithm. The approach relies on Xen as its
nested virtualization platform, which has a number of issues.

XenSPloit tool was developed to execute man-in-the-middle
attack during VM migration. It was able to modify the sshd
memory segment to circumvent sshd authentication. With such a
tool, VM might be accessed and the system confidentiality and
integrity may be compromised [20] [21] [22].

### 3.2.2 Kangaroo

Kangaroo is an OpenStack-based infrastructure approach that uses
a virtual switch and a Linux container (LXC) to live
migrate nested VMs within the cloud [17].

**With respect to Performance**, P1 The study claims migrating a
running application between the approach’s local deployment
and Amazon within a few minutes and without any downtime
[17]. P2 The nested VMs in the study have a 3.2 GB virtual disk,
which was migrated using OpenStack block migration. The disk
size is not practical and small to run a full Linux or Windows
operating systems [33]. P3 Despite the achieved performance,
the transporting protocol is still TCP/IP. In case of larger virtual
disk, big data and low WAN connection bandwidth, it might be
difficult to achieve the same result with such a protocol and
without any load balancing tools [8] [23].

**With respect to Flexibility**, F1 Decoupling VMs was achieved
by using nested virtualization (QEMU & LXC) [17]. F2 The
approach cannot run on a variety of O/S (i.e., Windows) because
the containers (LXC) are Linux-based [24].

**With respect to Security**, S1 The approach does not utilize an
encryption algorithm. S2 The approach does not utilize an
authentication algorithm. As the approach uses a layer 2
tunnelling technology to connect VMs, it has the same issues as
the Supercloud approach.

### 3.2.3 HVX

HVX is a virtualization platform that enables abstraction of
underlying IaaS. HVX can run unmodified operating systems
(i.e., Windows). HVX is similar to VMware because both
virtualization platforms use binary translation. However, the
lack of a popular open-source binary translation hypervisor has
allowed other approaches (such as para-virtualization) to be
more popular [24] [25].

**With respect to Performance**, P1 There was not a quantitative
evaluation of the approach’s speed, but rather it was mentioned
as robust and reliable [24]. P2 As for the storage migration, the
study introduced a storage abstraction layer that copes with
cloud storage heterogeneity. However, with large data size,
which is most likely to reach a couple of hundreds of gigabytes,
the approach may need optimization techniques, such as data
compression [33]. P3 As the approach leverages binary
translation to achieve a better performance in a nested
virtualization environment, many experts do not agree with
performance statement as this technique imposes extra overhead
on the guest kernel [7] [24]. HVX introduced its own user-
defined L2 overlay network (hSwitch). Yet, the transporting
protocol is UDP, which is a
best effort, connectionless protocol, but unreliable and it is not
clear if the study used a mechanism to recover lost packets due
to use such a protocol [36]. Also, the layer 2 network is subject
cast storm as multiple clouds may span over the network
and L2 has an issue with network scalability and cloud platforms
do not allow multicast and broadcast [18] [19].

**With respect to Flexibility**, F1 the approach managed to
incorporate various virtualization hypervisors, such as, QEMU,
Xen paravirtualization, KVM and VMware ESX, therefore, it
was able to decouple the VM from underlying hardware [24].
F2 this approach is the only one to run on a modified O/S (Linux)
and an unmodified O/S (Windows). Despite, it is seen as a
proprietary product and it cannot be evaluated [25].

**With respect to Security**, S1 The approach does not utilize an
encryption algorithm. S2 The approach does not utilize an
authentication algorithm.

### 3.3 Critical discussion

Despite that the approaches passed the mentioned criteria,
including decoupling VM from underlying system (Flexibility),
yet a number of limitations have been identified, mainly,
security and performance issues. This analysis reveals the
existing gap in those approaches in terms of the migration
downtime (performance), decoupling VMs from underlying
systems (flexibility) and securing live migration channel
(security). Although, all approaches managed to deploy nested
virtualizations (Xen-Blanket, LXC and HVX), these techniques
imposed the system to significant performance degradation and
limit VMs from running different operating systems on them
(i.e., Windows). Even though an IPsec tunnel or tinc VPN may
be used for protecting the migration process, it has not been
deployed due to performance issues [7]. IPsec uses encryption
and authentication algorithms, which expose the CPUs to
intensive overhead. As a consequence, IPsec increases migration
downtime and total migration time [39]. Table 2 provides a
summary of the analysis results.
Overall, the analysis shows that in order to gain a better performance, security mechanisms were not implemented. Despite that, approaches, such as Supercloud proposed tinc VPN as a security mechanism to protect the migration channel because it has less implication on performance [7]. Despite the lack of security criteria (S1 & S2) and some performance criteria (P2&P3), these solutions are still applicable to move VMs hosting publicly visible data (e.g., a Web Server that maintains a catalogue of books for sale). In such a scenario, security (especially, encryption) is not a main concern and in case of a web server migration failure, cloud users might be tolerant of longer time to access the corporate website.

4. CONCLUSIONS AND FUTURE WORK

Through incidents, such as security breaches, natural disasters, scarce resources and licenses costs, there is a demonstrated need to achieve cloud interoperability. Due to the current level of today cloud providers’ interoperability, researchers from industries and academia have been developing various approaches to alleviate the impact of such an issue and achieve live, cloud migration for VMs. Cloud brokerages, provider-centric and user-centric approaches are among the proposed solutions. Three user-centric approaches (Supercloud, Kangaroo and HVX) for VMs live migration across the cloud are analysed in this survey based on performance, flexibility and security QoS attributes.

This analysis reveals the existing gap in those approaches in terms of the migration downtime (performance), decoupling VMs from underlying systems (flexibility) and securing live migration channel (security). Although, all approaches managed to deploy nested virtualizations (Xen-Blanket, LXC and HVX), these techniques impose to significant performance degradation and limit VMs from running different operating systems (i.e., Windows). None of the techniques provide security capabilities. Future work of this study is to address the identified limitations by introducing a new approach, LibZam, which will be designed to minimize downtime migration, properly decouple VMs from underlying hardware, and secure the migration channel. The design of this system is reliant on the mentioned criteria, Performance (P1, P2 & P3), Flexibility (F1 & F2) and Security (S1 & S2). Different technologies, some of which are newly coined, are currently under investigation to realize these challenging QoS attributes. The system will be used in a real scenario; experts in the field will be asked to evaluate the system to enhance the system’s functionality.

Table 2 Summary of the Analysis Results

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Criterion Identifier</strong></td>
<td><strong>Assessment Values</strong></td>
<td><strong>Criterion Identifier</strong></td>
<td><strong>Assessment Values</strong></td>
</tr>
<tr>
<td>P1</td>
<td>✓ Relatively acceptable</td>
<td>P1</td>
<td>✓ Acceptable</td>
</tr>
<tr>
<td>P2</td>
<td>× None</td>
<td>P2</td>
<td>× None</td>
</tr>
<tr>
<td>P3</td>
<td>× None</td>
<td>P3</td>
<td>× None</td>
</tr>
<tr>
<td>F1</td>
<td>✓ Heterogeneous Hardware (CPU architecture i.e. flags) &amp; Disk I/O drivers</td>
<td>F1</td>
<td>✓ Heterogeneous Hardware (CPU architecture i.e. flags) &amp; Disk I/O drivers</td>
</tr>
<tr>
<td>F2</td>
<td>× Only modified O/S (Linux)</td>
<td>F2</td>
<td>× Only modified O/S (Linux)</td>
</tr>
<tr>
<td>S1</td>
<td>× None</td>
<td>S1</td>
<td>× None</td>
</tr>
<tr>
<td>S2</td>
<td>× None</td>
<td>S2</td>
<td>× None</td>
</tr>
</tbody>
</table>

5. Authors’ emails

Ibrahim Mansour: imansour@bournemouh.ac.uk
Reza Sahandi: rsahandi@bournemouh.ac.uk
Kendra Cooper: cooperk@bournemouh.ac.uk
Adrian Warman: Adrian.Warman@uk.ibm.com

6. References


