一种新的基于预拷贝的 VM-vTPM 动态迁移方法研究

齐能[1]，孟胜杰[2]

（1. 四川师范大学 计算机科学学院，四川 成都 610101；

2.郑州大学体育学院，河南 郑州，450000）

摘　要：针对目前动态迁移算法无法适用于最新的 vTPM 架构，不能保证虚拟机和对应 vTPM 虚拟机同时迁移的问题，提出了一种基于预拷贝动态迁移算法的 VM-vTPM 迁移方法。该方法对经典的动态迁移算法进行改进，增加了一类存储虚拟机和其对应 vTPM 相关迁移信息的数据结构，并设计虚拟机和 vTPM 迁移前后的联合拷贝和分离拷贝算法，以及在虚拟机管理器上增加 VM-vTPM 动态迁移时的错误处理机制。基于 Xen 的实验结果表明，在保证虚拟机和 vTPM 共同迁移的同时，可以有效的保证 vTPM 功能的使用；对性能实验的结果表明，增加的时间开销仅为 10%左右。

关键词：动态迁移；预拷贝；vTPM；云计算

中图分类号：TP319.1　　**文献标识码：**A

中文引用格式：

英文引用格式：

A New vTPM-VM Live Migration Scheme based on the Pre-copy Method

QI Neng[1],MENG Shengjie[2]

(1. College of Computer Science, Sichuan Normal University, Chengdu, Sichuan, 610068

2.Physical Education College of Zhengzhou University ,Zhengzhou,Henan,450000)

Abstract: To deal with the problem that the present live migration algorithm cannot apply to the latest vTPM architecture,which cannot guarantee the virtual machine and the corresponding vTPM virtual machine migration at the same time,this paper proposes a new VM-vTPM migration method based the pre-copy method.This method improves the classic live migration algorithm,which mainly is showed by the following aspects.First this paper increases a data structure to store the virtual machine the corresponding vTPM migration related information.Then this paper designs the virtual machine and vTPM migration before and after the joint copy and separation algorithm, as well as increases the VM-vTPM error handling on the virtual machine manager.The experiment results show that the method can effectively guarantee the vTPM function,when the virtual machine and the migration of vTPM at the same time.And the performance experiment results show that the increase of time only hold on about 10%.

Key words: Live migration；Pre-copy method；vTPM；Cloud computing

0 引言

云计算的发展给人们的生活带来了巨大的变化，虚拟化技术作为云计算中计算资源利用的一个关键技术，得到了学术界和工业界的广泛重视[1-3]。在云计算数据中心的虚拟机动态迁移是指虚拟机在不中断运行的情况下从某物理服务器节点迁移到另一个物理服务器节点的过程。利用虚拟机的动态迁移技术可以有效保证云租户在感知不到虚拟机变化的情况下实现对虚拟机的物理节点的更换，可以有效解决虚拟机对物理硬件平台的依赖[4,5]。

可信计算的虚拟化作为保证云计算环境安全的一种重要技术也得到了广大学者的研究[6,7]。通过在云计算平台上引入虚拟可信平台模块（vTPM）的概念，可以有效的利用物理可信平台模块（TPM）的功能实现对虚拟机的安全保护[8-11]。虚拟机可以利用虚拟机管理器VMM 提供的 vTPM 前端驱动实现信息的安全存储、完整性报告、远程证明、数据状态等。在虚拟机进行动态迁移时，就会出现如何保证虚拟机对应的 vTPM 实例在虚拟机迁移之后还依然能够正确的使用 vTPM 功能的问题[12-19]。

但是目前针对最新的 vTPM 架构，vTPM 作为云计算平台之上一个轻量级虚拟机而存在，相比在特权域建立 vTPM 实例数据文件相比，最新的 vTPM 架构可以利用 VMM 提供的安全机制实现更安全的服务[15,17]。针对最新的 vTPM 架构，目前针对 VM-vTPM 迁移的方法均不能适用于最新的云计算平台。本文针对目前的 vTPM 架构，提出了一种新的 VM-vTPM 迁移方法。该方法基于预拷贝动态迁移算法，在 VM 及其 vTPM 进行迁移时的迭代拷贝阶段进行改进，并设计了一种迁移时的 VM-vTPM 脏页拷贝策略。

与目前的 VM-vTPM 迁移方法相比，该方法有以下优点：

1）基于最新的 vTPM 虚拟化架构，较好的解决云计算环境下的 VM-vTPM 迁移问题；

2）设计了一种 VM-vTPM 同步迁移策略，该策略能够保证在 VM 和 vTPM 同步进行迁移，相比目前在 vTPM 迁移后迁移 VM 或者 VM 迁移之后迁移 vTPM 的方法更加高效；

3）设计了一种 VM-vTPM 脏页拷贝策略，该策略能够很好的保证虚拟机在迁移前后无中断的使用 vTPM 功能；

4）采用预拷贝动态迁移算法，能够很好的与当前主流的虚拟化平台进行衔接，比如 VMware 的 VMotion[21]，Xen 的 Live migration[22]。

1 基于预拷贝的 VM-vTPM 动态迁移

1.1 相关概念

1）新型 vTPM 体系架构

云计算的不断发展，基于虚拟化平台的 vTPM 会更加方便管理和使用。比如，在 Xen 的最新版本中对基于 TPM2.0 的 vTPM 架构中，vTPM 管理器作为一个单独的域运行在 VMM 之上，在 vTPM 实例创建机制上与传统 vTPM 架构创建 vTPM 实例的过程相似。但是此时的 vTPM 实例也被单独生成一个域，这种新型的 vTPM 架构可以减轻虚拟化平台管理域对 vTPM 的管理，把对 vTPM 管理的工作分给独立的 vTPM 管理域，使整个虚拟化平台的完整性和安全性更加得到保证。并且在 VM 和 vTPM 迁移上，新型 vTPM 架构会通过 VMM 的迁移机制对 vTPM 实例和对应的虚拟机实现迁移。

新型 vTPM 架构在 Xen[22]的具体实现过程如下三个步骤：

Step1：vTPM 管理域的配置以及启动；

vTPM 管理域可以通过配置文件由 VMM 进行启动，其仅仅使用部分较少的空间和内存。vTPM 管理域管理 vTPM 实例的创建、销毁等，但是虚拟化平台运行期间，只能有一个 vTPM 管理域。并且运行在 VMM 之上的 vTPM 管理域通过调用 TPM2.0 设备驱动接口，对 vTPM 实例域 TPM 功能的请求进行回复。

Step2：vTPM 管理域启动之后，对虚拟机 vTPM 的请求进行监听并创建相应的 vTPM 实例域；在虚拟机进行请求 vTPM 请求后，vTPM 管理域通过设备标识号进行 vTPM 实例域的重启或者创建，然后提供相应的后端驱动，由虚拟机连接 vTPM 实例域。

Step3：虚拟机进行 vTPM 实例的使用；

虚拟机通过 VMM 提供的 vTPM 前段驱动向 vTPM 域发起功能使用请求。vTPM 实例域通过 vTPM 管理域对 TPM 功能进行使用实现一次完整的 TPM 操作。

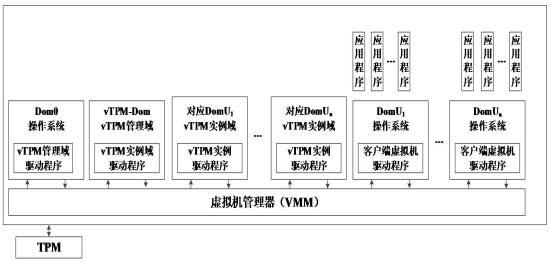

图 1　新型松耦合型 vTPM 架构

2）预拷贝动态迁移算法

预拷贝动态迁移算法（Pre-Copy）是当前主流的虚拟机动态迁移机制之一，并且当前的主流的云计算虚拟化平台都提供了预拷贝动态迁移的实现，比如 Xen，KVM 等，并且在广域网环境中不同物理主机之间的虚拟机迁移得到了广泛的应用。预拷贝算法的主要思想是讲内存拷贝分为三个步骤。

Step1：Push 阶段

在当前源服务器中运行的虚拟机的全部内存复制

到待迁移目的主机；

Step2：Stop-and-copy 阶段，通过内存脏页拷贝机制，迭代复制虚拟机运行时的内存脏页直到达到虚拟机运行时对内存的访问极少或者几乎没有的社会停止拷贝；

Step3：Push 阶段，源服务器中的虚拟机停止运行，并将最后一次迭代拷贝后的剩余内存脏页复制到目的机，然后在目的主机启动迁移后的虚拟机，完成动态迁移。

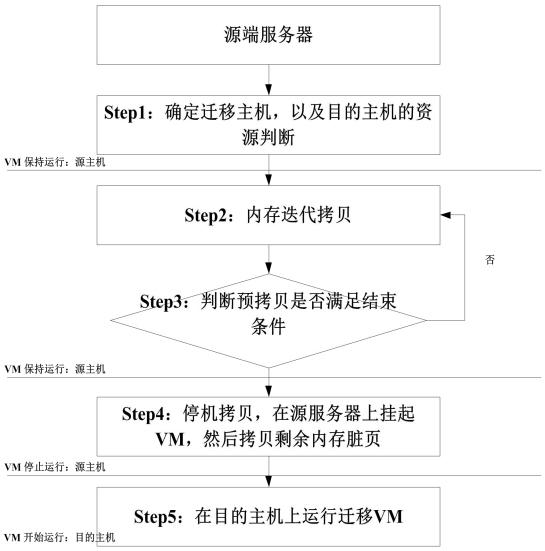

图 2 预拷贝动态迁移算法流程

1.2 VM-vTPM 迁移新方法

在目前的 VM-vTPM 迁移的方法中，由于 vTPM 实例是和 VM 一一对应的，所以在迁移 VM 到目的主机成功之前要把提供给 VM 对应 vTPM 服务的实例迁移到目的主机。目前最普遍的做法便是在 VM 动态迁移的 Push 阶段开始时迁移 vTPM。但是这种方法显然不适合于最新的 vTPM 架构。在虚拟机动态迁移过程中涉及到内存、存储、网络状态等众多资源的迁移，但是内存迁移是其中最重要的部分，本文仅对方法中的内存迁移进行重点描述。

定义 1：伪联合内存页（Camouflage Union Memory Page，CUMP）。在原有的影子页表的基础之上添加的数据结构，提供在 VM-vTPM 共同迁移时虚拟机虚拟地址—>虚拟机物理地址—>真实机器地址的映射，并且记录并更新 VM、vTPM 的 to.send 位图、to.skip 位图、及综合的 CUMP 的 to.send 位图、toskip 位图信息。在 VM、vTPM 虚拟机进行迁移时，由虚拟机监视器 VMM 提供 CUMP 的缺页中断机制。

定义 2：CUMP 缺页中断机制。当 VMM 解析到 VM-vTPM 迁移命令时，首先会判断 VM-vTPM 是否存在 CUMP 与其对应，当然在正常的情况下是不存在 CUMP 的，然后会根据 VM、vTPM 的影子页表等信息新建 CUMP，当在迁移过程中需要更新 CUMP 时，由 VMM 提供更新入口函数。

定义 3:VM、vTPM 关联机制。动态迁移过程中，在目的主机建立新的 VM 及对应 vTPM 虚拟机之后，需要对源主机及目的主机存在的 VM、vTPM 对应关系进行检验，定义数据结构和算法保证每次迁移均为同一关联的 VM 和 vTPM。

本文设计了一种新的 VM-vTPM 迁移方法，具体的迁移流程如下图：

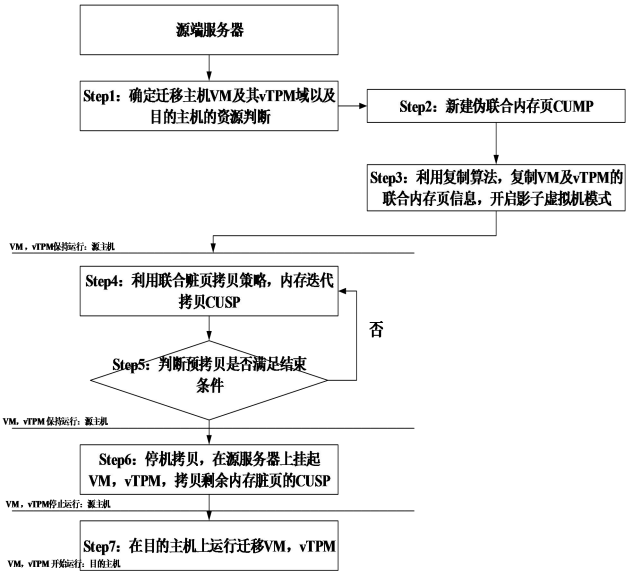

图 3 VM-vTPM 新方法流程

具体步骤描述如下：

Step1：预拷贝迁移准备，首先虚拟机 VM 及其 vTPM 实例域在源主机 A 运行时，选定主机 B 作为迁移目的主机，并向目的主机发出需要进行动态迁移的请求，确保目的主机 B 可以预留足够的资源（内存、CPU 等）。若目的主机不能满足源主机发出的迁移请求，则迁移失败；

Step2：由 VMM 对 VM，vTPM 迁移请求进行响应，提供 VM-vTPM 迁移方法；新建 VM-vTPM 联合内存页 CUMP（Camouflage Union Memory page），CUMP 用来存放 VM 和 vTPM 域的内存赃页信息，并提供 VM 和 vTPM 内存标识，以及脏页拷贝逻辑中的 to_send/to_skip 位图等信息；

Step3：复制 VM 和 vTPM 中所有的内存页信息、VM 及 vTPM 的相关标识到 CUMP 中；

Step4：利用联合内存页拷贝算法，对 CUMP 进行内存迭代拷贝，具体的内存迭代拷贝和预拷贝动态迁移相似，具体的内存页赃页拷贝策略在后文进行详细描述，此部分内存迭代拷贝可以看做是伪虚拟机的内存迭代拷贝，CUMP 代表的仅为拷贝的外在形式，具体的内存脏页数据依然存放在原有的位置；

Step5:如果根据内存拷贝策略决定 CUMP 需要传

送，则共同拷贝 CUMP、VM 脏页、vTPM 脏页到目的主机；由 CUMP 分离拷贝算法对 VM、vTPM 脏页进行处理，分别拷贝到对应的迁移的 VM、vTPM 脏页中；

Step6：判断预拷贝是否满足结束条件；

Step7：停机拷贝，在源服务器上挂起 VM，vTPM，拷贝剩余内存脏页的 CUMP；

Step8：在目的主机上运行迁移 VM，vTPM。

从以上可以看出，本文提出的 VM-vTPM 迁移方法可以对 VM-vTPM 进行同时迁移，在源主机以及目的主机上的阶段都是状态同步的。

1.3 CUMP 相关算法

如下图所示，VM-vTPM 迁移方法整个迁移过程中源主机与目的主机的主要动作。

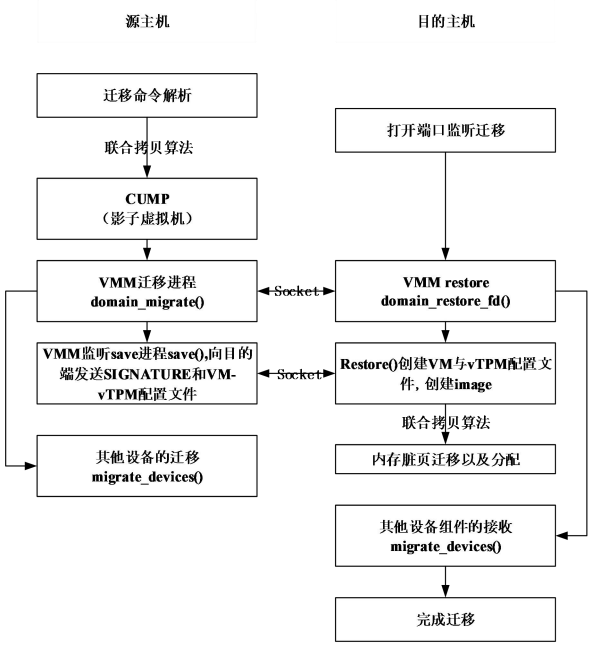

图 4 源主机与目的主机的主要动作

1）联合拷贝算法 VM-vTPM union copy

根据上文的分析，NMVVM 在利用 VMM 的管理工具进行 VM-vTPM 同时迁移时，首先要对源主机上的迁移命令进行解析，若出现 VM-vTPM 迁移命令，则对 VM 与 vTPM 之间的绑定关系进行认证。若 VM-vTPM 为一一对应关系，则新建 CUMP 页表，复制带有标识信息 VM 及 vTPM 的内存页到 CUMP。

算法 1.联合拷贝算法 VM-vTPM Union Copy

输入：VMM 管理工具迁移命令;

输出：VM-vTPM 联合内存页 CUMP.

1.Is VM or vTPM;//判断命令中同时存在一个 VM，一个 vTPM

2.GetUuid(domu.id,vTPM.id);//得到 VM 与 vTPM 的 uuid

3.IF(migrate.domu.vTPM.uuid != migrate.vTPM.uuid) THEN

return error(‘domu | vTPM don’t match’)//判断待迁移 VM 与 vTPM 是否绑定关系

4.Else

{create CUMP;

copy (VM’Shadow page | uuid) into CUMP;//复制 VM 相关信息到 CUMP

copy (vTPM’Shadow page | uuid) into CUMP;//复制 vTPM 相关信息到 CUMP

……

}

5.ENDIF

2）分离拷贝算法

在 CUMP 在源主机一端进行联合拷贝之后，进行正常的迁移过程，到达目的主机后，需要利用分离恢复算法对 VM-vTPM 各自的内存脏页进行恢复，是联合拷贝算法的逆过程。

算法 1.联合拷贝算法 VM-vTPM separation Copy

输入：VM-vTPM 联合内存页 CUMP.;

输出：VM-vTPM 各自的内存页;

1.Get domu’id ,vTPM.id

2.IF(domu’id.cfg or vTPM.idcfg not exit) THEN

return error(‘migration false’);

3.Else

4.Separate CUMP’VM’part to new VM;

5.Separate CUMP’vtom part to new vTPM;

……

6.EndIF.

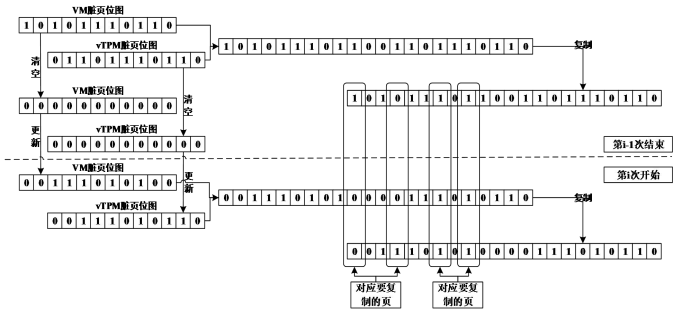

图 5 VM-vTPM 脏页拷贝策略

1.4 VM-vTPM 脏页拷贝策略

如图所示，需要说明的是，当 VM，vTPM 的相关信息存入 CUMP 中时，它们的相关内存依然保存在原有位置，图中表示的为伪内存脏页拷贝，真实的数据没有发生改变。当源主机的 VM，vTPM 进行迁移时，已有的动态迁移提供的脏页拷贝策略只能保证单个的虚拟机进行迁移，本文在目前的脏页拷贝策略上进行了针对 VM，vTPM 的脏页拷贝策略的改进，其中具体的脏页更新逻辑如下:

表 1 CUMPto send 位图随 VM,vTPM to send 位图逻辑

VM.to.send	0	0	1	1
vTPM.to.send	0	1	0	1
CUMP.to.send	0	1	1	1

表 2 CUMPto skip 位图随 VM，vTPM to skip 位图逻辑

VM.to.skip	0	0	1	1
vTPM.to.skip	0	1	0	1
CUMP.to.skip	0	1	1	1

表 3 CUMP 脏页更新逻辑

CUMP.to.send	0	0	1	1
CUMP.to.skip	0	1	0	1
是否传送	否	否	是	否

由表可知，VM,vTPM 的 to send ,to skip 位图的变化对 CUMP 对应位图的影响较大，当表中的位图满足一下条件时，CUMP 脏页不发生传送。

1）CUMP.to.send 位图为 0，CUMP.to.skip 位图为 0，即 VM.to.send 和 vTPM.to.send 同时为 0，并且 VM.to.skip 和 vTPM.to.skip 位图同时为 0，此时 VM、vTPM 的内存页表在上一轮复制过程以及本轮复制过程均没有被修改，CUMP-n 在两次的复制过程也没有被修改，故 CUMP-n 在本轮不会被复制传送，也不会发生在目的主机的分离复制；

2）CUMP.to.send 位图为 0，CUMP.to.skip 位图为 1，即 VM.to.send 和 vTPM.to.send 同时为 0，并且（VM.to.skip 位图为 0，vTPM.to.skip 位图为 1）、（VM.to.skip 位图为 1，vTPM.to.skip 位图为 0）、（VM.to.skip 位图为 1，vTPM.to.skip 位图为 1）的情况下，由于 VM、vTPM 的复制影响，CUMP 的 CUMP.to.skip 位图为 1，CUMP 在上一轮的复制过程中没有发生改变，而在本轮的复制过程中发生了修改，但是并不确定修改是否结束。所以 CUMP-n 在本轮也不会被复制传送。

3）CUMP.to.send 位图为 1，CUMP.to.skip 位图为 0，即（VM.to.send 位图为 0，vTPM.to.send 位图为 1）、（VM.to.send 位图为 1，vTPM.to.send 位图为 0）、（VM.to.send 位图为 1，vTPM.to.send 位图为 1），并且 VM.to.skip 位图为 0，vTPM.to.skip 位图为 1）、（VM.to.skip 位图为 1，vTPM.to.skip 位图为 0）、（VM.to.skip 位图为 1，vTPM.to.skip 位图为 1）的位图同时为 0 的情况下，在上一轮复制过程和本轮的复制过程都在修改，并且并不确定是否会即将修改结束，故 CUMP-n，在此轮并不会被复制传送。

综上，只有 VM，vTPM 的 to.send 和 to.skip 的位图满足一下条件时，CUMP-n 才会被复制传送，即 VM.to.send 位图为 1，vTPM.to.send 位图为 0）、（VM.to.send 位图为 1，vTPM.to.send 位图为 1）时 VM.to.skip 和 vTPM.to.skip 的位图同时为 0 的情况下发生复制传送。在目的主机同时发生分离复制。

2 基于预拷贝的 VM-vTPM 动态迁移

2.1 实验环境

实验采用 3 台主机，其中一台为 ISCSI[23]共享存储服务器，另外两台采用 Xen 4.5 作为 VMM 搭建 Xen 虚拟化平台，在其中一台虚拟化平台创建内存为 512M 的虚拟机，采用 TPM_Emulator 对 TPM 进行仿真，并创建虚拟机对应 vTPM 域。以上实验的操作系统均为 Ubuntu14.04，链路速度为 100Mb/s。

在功能测试实验中，为测试新的 VM-vTPM 方法能够保证在实现迁移的过程中不中断用户使用虚拟机及 vTPM 功能，本文编写脚本模拟用户不停使用 vTPM 功能，此脚本具体的功能在每个时间段调用 vTPM 对数据进行 SHA1 加密操作。

2.2 实验环境

1）CUMP 相关数据结构

为了准确的在 VM-vTPM 迁移时，VMM 能够正确的捕获 VM 及 vTPM 内存的变化以及脏页拷贝逻辑的更新，需要在 CUMP 中添加 VM 及 vTPM 相关数据结构，保证 VM-vTPM 联合迁移。

具体的数据结构表示如下：

```
Struct cumpInfo{
int vm_to_send;//记录 vm tosend 位图信息;
int vtpm_to_send;//记录 vtpm to_send 位图信息;
int vm_to_skip;//记录 vm to_skip 位图信息;
int vtpm_to_skip;//记录 vtpm to_skip 位图信息;
struct vm_pageInfo * next;
struct vtpm_pageInfo * next;
……
}
```

2）其他模块

在方案实现过程中，还必须在 Xen VMM 中添加 VM-vTPM 迁移错误处理机制，以及线程处理等方法，在此不再赘述。

2.3 实验及分析

1）VM-vTPM 迁移

源主机迁移前；

```
root@cs-sicnu:/home/t1-cs-sicnu# xl list
Name                                        ID      Mem  VCPUs      State  Time(s)
Domain-0                                     0    16384      1     r-----    710.1
mig-vm1                                      3      512      1     r-----    134.1
mig-vm1-vtpm                                 5        2      1     r-----     10.3
root@cs-sicnu:/home/t1-cs-sicnu#
```

图 6 源主机迁移前状态

目的主机迁移前：

```
root@cs-sicnu:/home/t2-cs-sicnu# xl list
Name                                        ID      Mem  VCPUs      State  Time(s)
Domain-0                                     0    16384      1     r-----    563.1
root@cs-sicnu:/home/t2-cs-sicnu#
```

图 7 目的主机迁移前状态

源主机迁移后：

```
root@cs-sicnu:/home/t1-cs-sicnu# xl list
Name                                        ID     Mem  VCPUs      State  Time(s)
Domain-0                                     0   16384      1     r-----      710.1
root@cs-sicnu:/home/t1-cs-sicnu#
```

图 8 目的主机迁移后状态

目的主机迁移后：

```
root@cs-sicnu:/home/t2-cs-sicnu# xl list
Name                                        ID     Mem  VCPUs      State  Time(s)
Domain-0                                     0   16384      1     r-----      563.1
mig-vm1                                      3     512      1     r-----      121.1
mig-vm1-vtpm                                 5       2      1     r-----        6.3
root@cs-sicnu:/home/t2-cs-sicnu#
```

图 9 目的主机迁移后状态

2）vTPM 迁移过程功能测试

动态迁移过中 vTPM 运行状态；

```
root@cs-sicnu:/home/tmig# ./vtpmtest

Take_SHA1_BASH
1--time:2017-04-27:14-47
The SHA1 Result
eb059e86122fc9b97e789d015410e37920cdf34e

2--time:2017-04-27:14-47
The SHA1 Result
217a7168f2f72d4faac1b9f300cc9aff111a23cd
```

图 10 vTPM 运行状态 1

动态迁移后 vTPM 运行状态；

```
13--time:2017-04-27:14-48
The SHA1 Result
a8018d6c4084c74b47cd458c9e1bfcedc27632bd

14--time:2017-04-27:14-48
The SHA1 Result
19cc4326e34a950f911674e3d24de1e4bf6b479d
```

图 10 vTPM 运行状态 2

3）性能测试

本文提出的方法在一定程度上会对 VM 和 vTPM 迁移的总时间产生影响，本文选取了正常情况下对 VM 和 vTPM 进行迁移五次，迁移过程中仅仅保证 VM 和 vTPM 的正常迁移，不对 VM 和 vTPM 进行绑定操作。并采用 VM-vTPM 新方法对 VM-vTPM 进行五次迁移。

下表为迁移时间对比；

阶段	第一次	第二次	第三次	第四次	第五次
VM 迁移	58s	62s	60s	62s	59s
vTPM 迁移	10s	8s	9s	8s	8s
总迁移时间	68s	70s	69s	70s	67s
VM-vTPM	75s	72s	76s	73s	70s

表 4 迁移时间对比

由上表可知，本文提出的 VM-vTPM 迁移新方法在迁移总时间上会增加时间的开销，大约在 10%左右。但是相比于迁移总时间以及对 vTPM 使用功能的影响来看，这些多余的时间开销在可接受的范围内。

3.结束语

本文针对目前动态迁移算法不适用于最新 vTPM 架构、无法保证 VM-vTPM 同时迁移的问题，基于原有的预拷贝动态迁移算法，提出了一种新的 VM-vTPM 迁移方法。实验证明本文提出的方法可以在保证VM-vTPM 同时迁移的同时能够保证vTPM功能的有效的使用。

本方案仅仅是实现了 VM-vTPM 同时迁移，在共享存储的基础上实现动态迁移，迁移过程中实现了基本的 vTPM 功能，对 vTPM 远程证明、信任度量等功能未做处理。本文的下一步工作将重点研究迁移过程中远程证明、信任链扩展等。

参考文献

[1]. 林闯,苏文博,孟坤,刘渠,刘卫东. 云计算安全:架构、机制与模型评价[J]. 计算机学报,2013,(09):1765-1784.

[2]. 王斌锋,苏金树,陈琳. 云计算数据中心网络设计综述[J]. 计算机研究与发展,2016,(09):2085-2106.

[3]. 王于丁,杨家海,徐聪,凌晓,杨洋. 云计算访问控制技术研究综述[J]. 软件学报,2015,(05):1129-1150.

[4]. 郭军,闫永明,马安香,张斌. 云环境下基于冷点虚拟机迁移的热点消除方法[J]. 清华大学学报(自然科学版),2016,(11):1232-1236.

[5]. 程虹锡,谭良. 一种高效的虚拟机动态内存迁移方法[J]. 计算机科学,2016,(04):111-114.

[6]. 冯登国,秦宇,汪丹,初晓博. 可信计算技术研究[J]. 计算机研究与发展,2011,(08):1332-1349.

[7]. 沈昌祥,张焕国,王怀民,王戟,赵波,严飞,余发江,张立强,徐明迪. 可信计算的研究与发展[J]. 中国科学:信息科学,2010,(02):139-166.

[8]. 杨健,汪海航,王剑,俞定国. 云计算安全问题研究综述[J]. 小型微型计算机系统,2012,(03):472-479.

[9]. 蒋华,闫一凡,鞠磊. 可信服务链安全架构研究[J]. 计算机应用研究,2018,(04).

[10]. 黄强,孔志印,张德华,常乐. 一种云计算适用的虚拟可信报告根构建机制[J]. 工程科学与技术,2017,(02):140-144.

[11]. 吴吉义. 云计算:从云安全到可信云[A]. 中国计算机学会信息存储技术专业委员会.2010 年第 16 届全国信息存储技术大会（IST2010）论文集[C].中国计算机学会信息存储技术专业委员会:,2010:5.

[12]. 孔斌,张珠君,王婷婷,张杰,黄伟庆. KVM 虚拟化动态迁移技术的安全防护模型[J]. 软件学报,2016,(06):1402-1416.

[13]. 吴军,张轶君,白光伟. Xen 下虚拟机动态迁移优化策略的研究[J]. 电子技术应用,2015,(11):128-131.

[14]. 熊安萍,徐晓龙. 基于内存迭代拷贝的 Xen 虚拟机动态迁移机制研究[J]. 计算机科学,2013,(08):63-65+99.

[15]. 严飞,于钊,张立强,赵波. vTSE:一种基于SGX的vTPM安全增强方案[J]. 工程科学与技术,2017,(02):133-139.

[16]. 于颖超,刘了,陈左宁. 一种安全 VM-vTPM 迁移协议的设计与实现[J]. 电子技术应用,2012,(04):130-133.0
[17]. 杨永娇,严飞,毛军鹏,张焕国. Ng-vTPM:新一代 TPM 虚拟化框架设计[J]. 武汉大学学报(理学版),2015,(02):103-111.
[18]. 黄宇晴,赵波,肖钰. 一种基于 KVM 的 vTPM 虚拟机动态迁移方案[J]. 山东大学学报(理学版),,:1-6.
[19]. 刘明芳,李文锋,赵阳. 一种基于 XEN 平台的可信虚拟机迁移协议[J]. 计算机安全,2013,(03):13-18.
[20]. 王因传,杨林,孙伟峰. IBM vTPM 的 Xen 实现研究[J]. 军事通信技术,2010,(03):67-71.
[21]. VMwareInc.[EB/OL].https://www.vmware.com/cn.html.2017
[22]. Xen.[EB/OL].https://en.wikipedia.org/wiki/Xen.2017
[23]. iSCSI.[EB/OL].https://en.wikipedia.org/wiki/ISCSI.2017

作者简介：

齐能（通讯作者），1993 年生，河南商丘人，男，硕士，主要研究领域为可信计算，云计算安全，电子邮箱：qihuaneng@163.com，电话：18482212820；
孟胜杰，1992 年生，河南驻马店人，女，学士，主要研究领域为信息技术，电子邮箱：qihuaneng@163.com。