基于Ceph与OpenStack的虚拟机安全迁移

摘 要

由于云计算概念的火热，虚拟化技术作为云计算的核心技术也得到工业界和学术界的关注。在集群中通过部署虚拟化环境可以为用户提供按需分配的计算和数据，在主机中开启虚拟机后，不可避免地碰到容灾备份和负载均衡的问题。为了解决这个问题我们使用了虚拟机迁移技术，然而虚拟机迁移面临的主要问题是迁移时会拷贝虚拟机内存在网络上进行发送，受到严重的安全威胁。因此我们提出了一种基于OpenStack与Ceph的虚拟机安全迁移的方法，通过使用OpenStack的Security Group功能来保证虚拟机安全策略的一致性；通过部署集群中基于PKI身份认证的TLS互信来保证虚拟机迁移过程中迁移内容的机密性和完整性。我们使用上述的方法成功的进行了虚拟机的安全迁移，并通过实验测试了Ceph存储的性能和验证了虚拟机迁移的机密性和完整性。

关键词：虚拟化；虚拟机迁移；OpenStack；Ceph；TLS

Abstract

Due to the hot concept of cloud computing, virtualization has also received the attention of industry and academia. Virtualization is the core part of the cloud computing. In a cluster, we can provide users with on-demand computing and data by deploying virtualization environments. After booting virtual machines on the cluster, we will face the problem of load balancing and disaster recovery. To solve this problem, we use the virtual machine migration technology. But the migration has faced a serious security threat because it is not encrypted. We proposed a virtual machine migration method based on OpenStack and Ceph. OpenStack's Security Group capabilities ensure the consistency of virtual machine security policies. Protect the confidentiality and integrity of migrated content during virtual machine migration by deploying TLS trust for the cluster. We use this method to successfully carry out the security the virtual machine migration. And tested the performance of Ceph storage and verified the confidentiality and integrity of virtual machine migration.

**Key words:** Virtualization; Virtual Machine Migration; OpenStack; Ceph; TLS

目 录

[1 绪 论 4](#_Toc483318496)

[1.1 研究背景及其意义 4](#_Toc483318497)

[1.2 国内外研究现状 5](#_Toc483318498)

[1.3 本文的主要工作 6](#_Toc483318499)

[1.3.1 主要问题分析 6](#_Toc483318500)

[1.3.2 针对问题的研究内容 6](#_Toc483318501)

[2 OpenStack与Ceph分布式存储 7](#_Toc483318502)

[2.1 Ceph分布式存储 7](#_Toc483318503)

[2.1.1 Ceph分布式存储简介 8](#_Toc483318504)

[2.1.2 RADOS存储介绍 10](#_Toc483318505)

[2.1.3 CRUSH介绍 11](#_Toc483318506)

[2.1.4 Ceph与Gluster，Swift的比较 12](#_Toc483318507)

[2.2 OpenStack软件平台 13](#_Toc483318508)

[2.2.1 OpenStack介绍 13](#_Toc483318509)

[2.2.2 OpenStack部分组件介绍 13](#_Toc483318510)

[2.2.3 OpenStack网络服务安全组介绍 15](#_Toc483318511)

[3基于Libvirt的虚拟机迁移技术 15](#_Toc483318512)

[3.1 虚拟机迁移主要应用 15](#_Toc483318513)

[3.2 虚拟机迁移的分类 16](#_Toc483318514)

[3.2.1 在线迁移 16](#_Toc483318515)

[3.2.2 离线迁移 16](#_Toc483318516)

[3.3 迁移流量的传输方式 16](#_Toc483318517)

[3.3.1通过Hypervisor进行传输 16](#_Toc483318518)

[3.3.2 通过Libvirt隧道进行传输 17](#_Toc483318519)

[3.4 控制流量的传输方式 17](#_Toc483318520)

[3.4.1受管理的直接迁移 17](#_Toc483318521)

[3.4.2 受管理的对等迁移 18](#_Toc483318522)

[3.4.3 无管理直接迁移 18](#_Toc483318523)

[3.5 迁移数据安全加密 19](#_Toc483318524)

[3.5.1 TLS介绍 19](#_Toc483318525)

[3.5.2 基于客户端认证的TLS握手过程 20](#_Toc483318526)

[4 基于OpenStack与Ceph的虚拟机安全迁移 22](#_Toc483318527)

[4.1 实验环境 22](#_Toc483318528)

[4.2 Ceph存储性能测试 23](#_Toc483318529)

[4.3 安全迁移机密性完整性分析 23](#_Toc483318530)

[4.4虚拟机安全迁移效率分析 25](#_Toc483318531)

[5 总结与展望 26](#_Toc483318532)

[5.1 论文总结 26](#_Toc483318533)

[5.2 未来工作展望 27](#_Toc483318534)

[参考文献 28](#_Toc483318535)

[致谢 29](#_Toc483318536)

1 绪 论

1.1 研究背景及其意义

随着现代互联网的发展，计算资源，存储资源，数据资源和应用资源都产生着爆炸性的增长。传统的互联网的提出是为了进行不同的主机之间通信，而现代的互联网正在不断地转向以计算为主的平台。云计算是一种基于互联网的计算，可以根据用户的需要向其他设备提供共享的计算机的处理资源和数据。它可以实现按需访问那些可以配置的计算资源，可以最小化的快速配置和发布管理[1]。云计算可以使公司避免花费过多的成本用在基础设施的搭建，可以使企业更快的实现程序和业务的升级，减少维护的成本。

目前的云计算的可伸缩性是依赖虚拟化技术，通过把多台服务器虚拟化之后，构成一个资源池。虚拟化技术可以使资源管理更加灵活，资源利用率更高，且对上层应用透明。通常体现在某一服务器上，创建一个或者多个像具有真正的操作系统的虚拟机，这些虚拟机并行运行，执行的软件和底层的硬件资源完全分离。对于硬件的虚拟话来讲，主机是进行虚拟化的实际存在的机器，虚拟机是通过主机中一种虚拟机管理程序来创建的。它是一个软件层或者子系统，能控制硬件并向客户机操作系统提供访问物理硬件的途径。通过提供虚拟化的硬件，虚拟机管理程序允许多种操作系统在相同的物理机上运行[16]。

当使用集群部署虚拟化环境时，不可避免的要使用虚拟机迁移来进行一些容错备份或者负载均衡。当虚拟机进行跨域迁移时，由于有状态不稳定且在公网中易受攻击的劣势，并且在虚拟机迁移到目的环境时，由于虚拟机安全策略的不一致，会产生额外的安全问题。这种安全问题会严重威胁云租户的个人隐私。本文通过研究虚拟机跨域迁移中虚拟机数据的机密性和完整性，同时确保虚拟机在动态迁移前后，它所拥有的安全策略的一致性，实现了基于Ceph与OpenStack的虚拟机安全迁移。

1.2 国内外研究现状

云计算环境下，用户数量庞大，网络环境复杂。虚拟机迁移时会暴露在公网环境下，所以需要防止虚拟机内存拷贝时的数据被嗅探和被篡改。

王光波提出了一种自主的虚拟机动态迁移框架，该框架不需要中央节点的管理，避免了单点失效，可以提高整个系统的可用性。并对于安全迁移的问题，提出了虚拟机安全迁移模型VMSMM。该模型是基于开源平台Xen的基础上，修改部分安全模块之后，设计了一套全新的vTPM密钥体系来保证虚拟机的安全性[2]。

梁新龙提出了一个vTPM的密钥层次结构，使得不可迁移的vTPM密钥可以迁移，然后虚拟机就能和对应的vTPM一起迁移到目标平台。他使用了源平台和目的平台之间的双重身份认证，还提出了一个安全可靠的VM-vTPM迁移协议来保证vTPM迁移过程中的安全性[3]。

R. Divyambika提出了PALM（Protection Aegis for Live Migration）的安全扩展机制用来在虚拟机管理程序中保护虚拟机系统的安全性。这个系统中所有的元数据都被封装，它总共包括三个模块：Data Protector用来加密解密虚拟机迁移过程中的数据，Metadata Manager负责在源主机和目的主机中传输和重建元数据，Safe Guard模块负责在进程迁移，内存迁移，挂起/继续迁移这三个情境中保证安全性[4]。

虚拟机的迁移完毕后，需要定义符合当前环境的安全策略。所以需要额外建立一条策略同步信道来保证其安全策略的同步更新。孙松儿等提出了一种虚拟机安全策略的迁移方法及装置，应用于数据中心的安全管理服务器上，包括迁移感知单元，定位单元，安全策略管理单元。实现了虚拟机在安全设备上，安全策略自动跟随虚拟机迁移而无缝迁移，免除管理员因为虚拟机迁移而进行的安全设备手动配置工作。能够更好的在维护过程中减少人为手工操作带来的安全性问题[5]。

1.3 本文的主要工作

1.3.1 主要问题分析

本文针对云计算环境下，虚拟机动态迁移时可能产生安全威胁的两个方面做了分析。第一个方面是虚拟机在动态迁移时，采用预拷贝（pre-copy）技术[6]。该技术主要是不断的循环将内存中的数据发送到目的主机，同时源主机在此过程中没有挂起，还可以为用户提供服务。在这个过程中，用户的个人数据会发送到公网上，容易受到中间人的嗅探，修改等攻击。所以在拷贝内存的过程中，我们必须保证数据的机密性与完整性。第二个方面是虚拟机在迁移完成后，虚拟机与目的网络防火墙之间的安全策略并没有同步更新。这个过程如果使用人工手动配置，有可能会由于人员的误操作而导致安全隐患。

1.3.2 针对问题的研究内容

为了增强虚拟机动态迁移中的虚拟集群的自治性，安全性等特征。本文主要研究虚拟机跨域迁移的机密性和完整性的保证方法，虚拟机的安全迁移策略同步方法。通过这两个方面来实现虚拟机的跨域迁移的安全性。

1.3.2.1 虚拟机跨域迁移机密性与完整性的研究

本文主要分析虚拟机迁移时，因为网络环境和执行环境改变而产生安全威胁种类，研究在虚拟机迁移过程中敏感信息的机密性和完整性的保证方法。具体包括：基于虚拟域的跨域迁移安全性分析；目的执行环境和虚拟链路的安全性度量方法；可信的虚拟机迁移密钥协商机制和通信协议。

1.3.2.2 虚拟机安全迁移策略同步方案

本文主要分析虚拟机动态迁移可能导致的虚拟机网络配置、访问控制策略、网络过滤策略、密钥协商策略发生的变化，研究如何保证上述配置与策略的一致性。分析需要同步迁移的安全策略的异同点，然后建立一个用于策略迁移的隐蔽信道，通过一个全局的策略管理服务器来将安全策略分发到目的虚拟机，本地控制节点将消息扩散，解决安全策略的同步性问题。

1.3.2.3 系统架构图

我们的集群包括四台物理主机，每台物理主机上首先运行两个OSD进行基于对象的存储，在此之上是librados，通过它对OSD进行管理操作，再就是librbd，它向上对应用提供块设备存储的接口，向下调用librados进行基于对象的存储。QUEM是一种模拟器，可以模拟出虚拟硬件设备创建虚拟机，它向上提供创建和管理虚拟机和硬件设备的接口，向下可以使用块设备存储创建虚拟的卷。Libvirt用来管理QEMU，并提供丰富强大的API给上层应用进行调用，其中本文重点介绍的是关于迁移的操作。最上层的是OpenStack，它可以使用Libvirt提供的API进行虚拟硬件设备和虚拟机的创建，提供了身份认证，镜像管理，虚拟机创建，网络管理，块设备管理的基本功能。OpenStack还提供了基于Web的用户界面操作，极大的提高了用户体验和降低了用户的使用成本。



图1.1 基于OpenStack与Ceph迁移架构图

2 OpenStack与Ceph分布式存储

2.1 Ceph分布式存储

Ceph是一个分布式的存储系统，可以为用户提供对象，块，文件等多种方式的存储平台[7]。Ceph独一无二的提供一个统一的存储平台来实现上述功能。Ceph将集群内的工作分配给各个客户机以及守护进程来处理，大大提高了它的可伸缩性，可用性。

Ceph在逻辑上大致分为底层和上层应用接口。Ceph在上层为应用程序，操作系统，网络访问等提供了基于文件，块存储，对象存储的接口。然后在底层使用RADOS的分布式对象存储。Ceph在底层是完全的基于对象存储的方式来进行数据管理和分发，是存粹的软件定义存储的方式。软件定义存储兴起于云计算环境下的虚拟化技术，以软件控制的方法来决定数据的存储方式与管理方式，和存储的硬件本身无关。Ceph也支持多种文件系统，而且能够保证没有单点故障的高可用性。

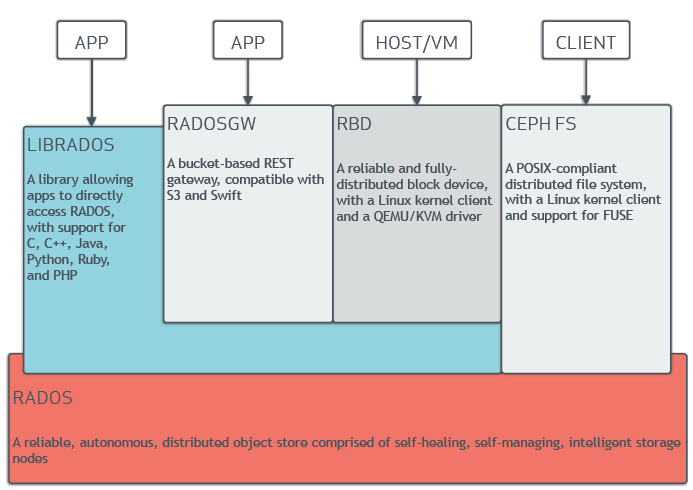


图2.1 Ceph分布式存储架构图

2.1.1 Ceph分布式存储简介

2.1.1.1 Ceph存储集群

Ceph提供的存储集群是基于RADOS技术，是一种可伸缩的，基于PB级存储集群的可靠服务。存储集群具有无限伸缩的特性，可以满足海量用户的存储需求[9]。集群包括两种类型的守护进程，分别是Ceph监视器进程和Ceph OSD 守护进程。

Ceph监视器维护集群映射的主副本，需要有一组监视器来避免由于单一监视器进程失效而带来的集群错误。存储集群的客户端从Ceph监视器请求并获取集群当前映射的最新副本。

Ceph OSD守护进程检查自身的状态和其他OSD的状态，并将状态发送给监视器。存储集群的客户端和每个Ceph OSD守护进程都是用CRUSH算法来有效的计算数据存储位置的信息。这样可以避免从一个中心化的表中拿到数据，从而有效的避免了单点依赖。

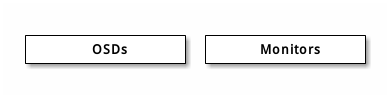


图2.2 Ceph存储集群简图

2.1.1.2 Ceph存储集群协议

Ceph客户端和存储集群之间的交互需要使用原生的Ceph协议。Ceph将所需的协议进行封装，也就是图中的librados，从而可以使用户定制客户端，并访问底层存储集群。Ceph协议可以使用数据条带化的存储方式来提升磁盘的吞吐量和性能。数据条带化可以显著的提高存储设备的性能和吞吐量，Ceph的条带化具有RAID 0级别的吞吐量，Ceph的性能可以因为该特性得到提高，同时它也具有N路RAID的可靠性和更快的恢复性。

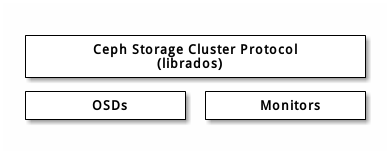


图2.3 Ceph存储集群协议

2.1.1.3 Ceph客户端

Ceph客户端包括也就是Ceph为上层不同的应用程序提供的可以自定义的接口，包括块设备，对象存储，文件存储。Ceph块设备（RBD）提供了大小可调，支持快照和克隆的块设备，Ceph可以将块设备条带化，从而使集群的性能得到提高。而且Ceph支持内核对象和QEMU程序直接使用librbd，避免了内核对象在虚拟系统上的开销。Ceph对象存储提供了RESTful的风格的API，与Amazon S3和OpenStack Swift兼容。Ceph文件系统提供了兼容POSIX的文件系统，可以直接mount或者挂载到用户空间文件系统（FUSE）。本文主要使用了Ceph块设备，通过librbd对librados进行管理。

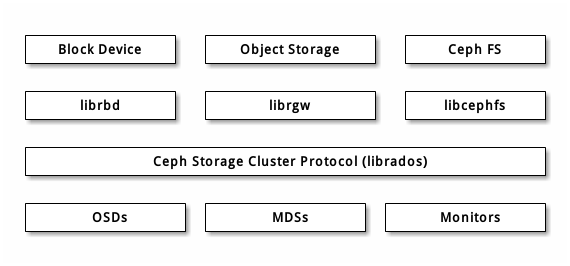


图2.4 Ceph客户端

2.1.2 RADOS存储介绍

RADOS是Ceph实现高可用性，数据自动恢复的框架。在进行读写时，会有以下的步骤来进行读写操作（Ceph块存储设备RBD作为客户端）[9]。

在进行读操作时：客户端发出读请求到RADOS，RADOS接受请求后，将请求发送到一个primary OSD。Primary OSD在本地磁盘上读取数据之后，将数据返回给RADOS,完成读操作。

在进行写操作时：客户端写数据，然后RADOS将数据发送到primary OSD。Primary OSD首先复制出一个replica OSD，然后将数据发送给replica OSD。接着将数据写到本地磁盘。Replica OSD将数据写入本地磁盘，完成数据的备份操作，然后向primary OSD发出完成信号。当primary OSD得到所有replica OSD的信号之后，会向RADOS发送一个请求完成信号

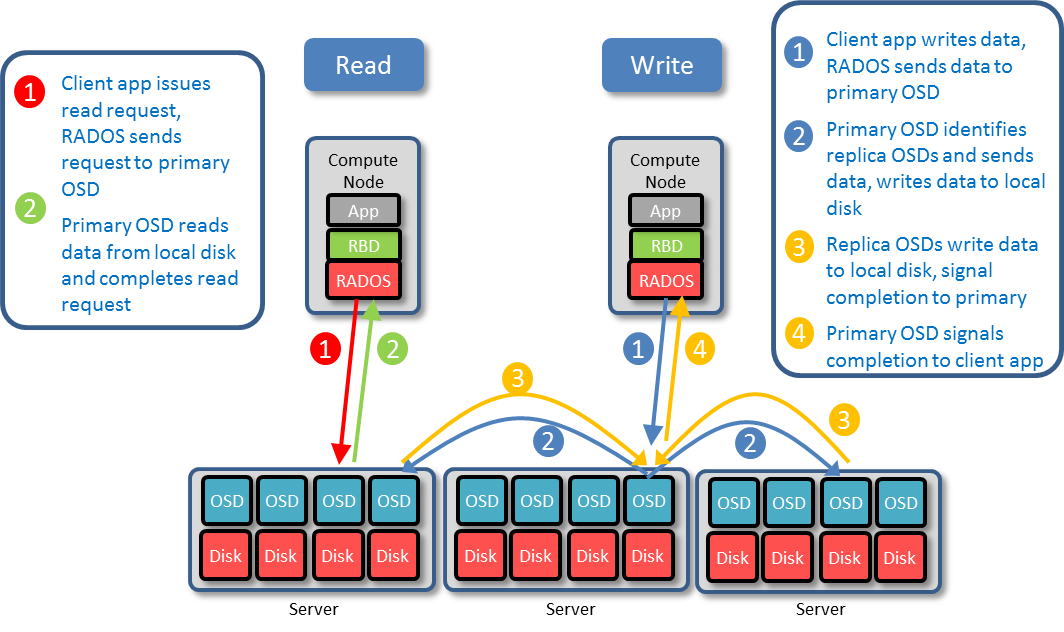


图2.5 RADOS读写操作

2.1.3 CRUSH介绍

CRUSH算法是一种计算数据存放位置的算法，避免了使用类似于文件表的中心化结构。它类似于哈希和一致性哈希。但是哈希算法在使用过程中，如果有新主机加入集群，或者淘汰掉老的主机，就会引起几乎所有数据的位置变动，开销非常的大。一致性哈希通过使用存储节点划分哈希空间进行分区的方式解决了数据迁移量过多的问题，但是它的问题在于不能把空间划分均匀导致某些服务器负载过大，并且在进行添加删除节点时，需要遍历已经写入磁盘的数据来确定是否迁移,这个过程耗费的时间也比较长[8]。

Ceph通过计算数据的哈希值，然后将该值模归置组的数量，得到的数值即为该数据对应的归置组的编号。然后每个归置组使用CRUSH算法映射到一组OSD中，所以在这个过程中首先将数据映射到归置组，然后将归置组映射到OSD进行实际的存储。归置组解决了上文一致性哈希的问题，首先使用数目不变的抽象归置组来划分数据的分区，使数据均匀的分步在归置组中，避免某些服务器负载过大。而且由于归置组的虚拟性，数目不会发生变化，从而保证了分区的数目和虚拟的节点数目的一致，在分区固定的情况下，需要迁移数据时无需遍历所有数据。

CRUSH算法中为归置组选择映射的OSD主要考虑了，归置组在OSD中是否均匀分布的问题和OSD是否被不同的故障域隔离的问题。OSD所占磁盘的大小尽可能相等用来保证它所包含的归置组数量差别不会太大，数据副本与数据本身尽量不要处于同一个主机。Ceph使用的是树形结构来描述OSD的在空间分布中的层次关系。主要包括叶子结点Device和非叶子结点Bucket，叶子节点在层次结构中表示OSD所在的位置，对于非叶子节点来说，从底向上分别为所在的主机，主机所处的机柜，机柜所处的机房。

CRUSH算法包括三个选项，take操作，select操作和emit操作。Take操作选择上述存储结构的一个bucket，并将该bucket分配给一个向量i。select操作对于向量i进行迭代地选择bucket，并可以递归的在选定的bucket的子树进行select操作。通过重复的take操作和select操作，选取出不同的位置来存放数据的副本。选出目的磁盘空间后进行emit操作，将向量i放入到结果中。

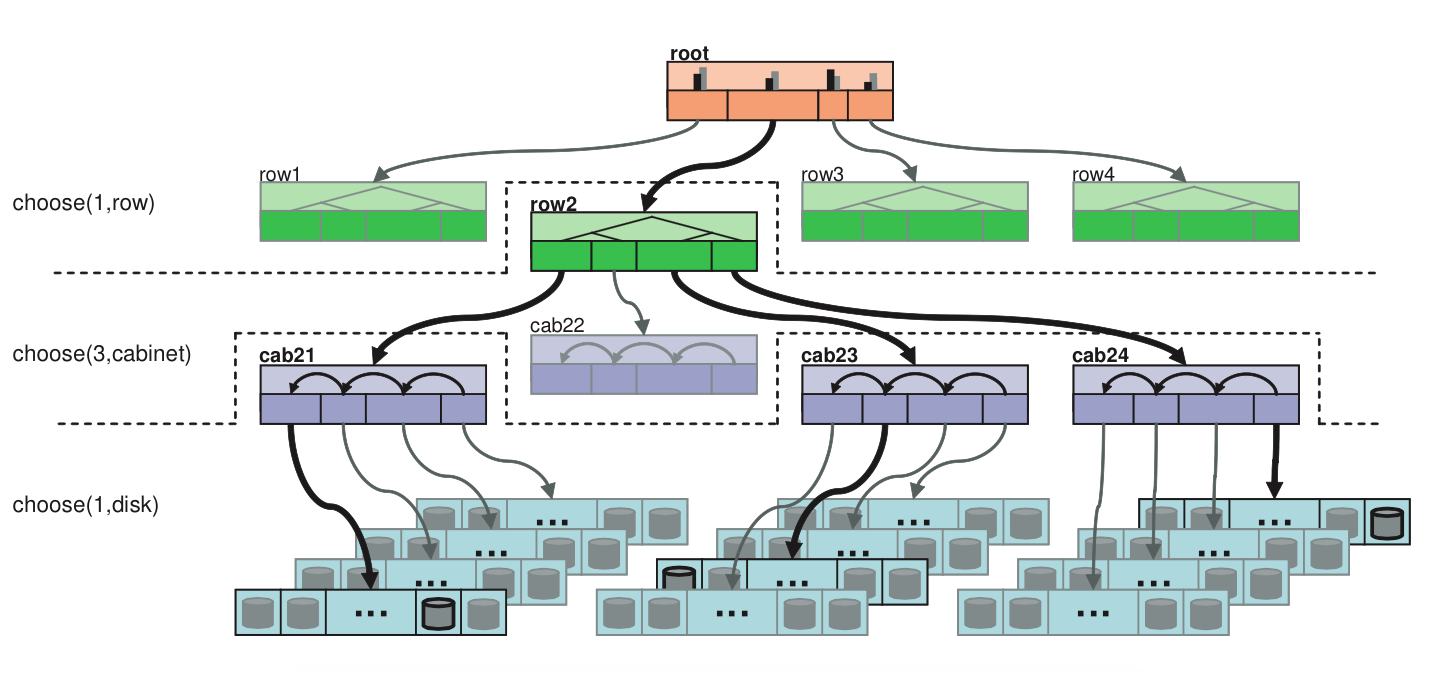


图2.6 CRUSH算法图

2.1.4 Ceph与Gluster，Swift的比较

2.1.4.1 Ceph与Gluster的比较

Ceph与Gluster都是软件定义存储的成功案例，都提供了强大的存储能力和可扩展的存储空间。Gluster是基于文件存储的，并且提供了对象访问的功能。Ceph在底层是基于对象存储并支持上层的文件和块存储扩展。Gluster提供连续的块存储，将一组数据保存在存储空间中。这个特性可以使我们存储诸如日志之类的快速增加的数据，日志数据会随着时间的推移，访问热度不断下降，所占空间不断上升。Ceph使用对象存储，可以将数据存储在跨越计算机的二进制对象中，这个特性可以保证存储的数据可以在很短的时间内为海量的用户提供高性能的访问[11]。对于提供虚拟机的迁移操作而言，我们需要在短时间内提供数据的移动，所以我们选择Ceph而不是Gluster。

2.1.4.2 Ceph与Swift的比较

Ceph与Swift都是开源的对象存储系统，都可以为OpenStack的存储提供支持，在OpenStack默认的对象存储方案是swift。Ceph与Swift都是成熟的产品，并且在不断的开发更新中。Swift是OpenStack一个核心子项目，进出集群的流量都需要经过一个代理服务器，并采用最终一致性的数据更新行为。经过代理服务器会使写入读取的性能降低。Ceph的多地域支持使用了同步更新的策略，当写入时，本地节点和远程节点同步写入，当远程节点写入完成之后才会成功返回。由于我们使用虚拟机迁移的测试环境完全是在单一域中进行，Ceph可以为我们提供更好的性能支持[10]。

2.2 OpenStack软件平台

2.2.1 OpenStack介绍

OpenStack是一系列用来构建和管理云计算平台的软件工具，他可以用于公有云和私有云。全世界有数百家公司或者机构依赖于OpenStack来运行他们的服务，欧洲核子研究中心拥有操作190000个核心来支持世界上最大的OpenStack云。国家电网和中国移动都拥有10个数据中心，他们使用了OpenStack用来削减开支和提升性能[12][13]。

2.2.2 OpenStack部分组件介绍

2.2.2.1 身份认证服务Keystone

身份认证服务提供了一个中心字典，映射了OpenStack能够进行访问的各种服务。它提供了一个通用的认证系统，可以在云操作系统的各个组件之间进行认证操作。它提供了多种形式的认证操作，包括标准的用户名和密码认证，基于令牌的认证和AWS风格的登陆认证。除此之外，它提供了一个包括OpenStack所有服务的列表便于查询，用户和第三方工具能使用这个列表选择他们需要的访问的资源。

2.2.2.2 镜像服务Glance

OpenStack镜像服务提供了磁盘和服务器镜像的发现，注册和分发服务。存储的镜像可以当作一个模板来启动实例，同时实例的备份也可以存储在镜像服务中。镜像服务可以将磁盘和服务器的镜像存储在Ceph中，同时也提供了Rest风格的API用来查询镜像的信息。

2.2.2.3 计算服务Nova

OpenStack计算服务是云计算架构的控制器，是整个基础设施即服务的主要部分。它的目的是管理和自动分配计算资源池，能够很合适地与虚拟化技术和高性能计算配合使用。他可以提供给不同的hypervisor不同风格的API，用来支持不同类型的硬件虚拟化环境。

2.2.2.4 网络服务Neutron

OpenStack网络服务是用来管理虚拟机网络，为虚拟机和虚拟网络设备提供IP地址。它确保了网络不是在OpenStack云环境中的瓶颈或者限制因素，能够使用户进行网络的配置和自定义。

Neutron可以为不同的用户提供不同的网络模型，标准的网络模型包括flat网络或者能划分虚拟子网的VLAN网络。也可以通过用户不同的需求，为实例提供静态IP，或者使用DHCP服务。浮动IP地址可以动态地重定向流量到云环境中地资源，可以在出现故障后进行快速恢复处理。

2.2.2.5 块存储服务Cinder

OpenStack块存储服务提供持久的块存储设备，可以将块设备挂载到实例中进行使用。它可以创建，删除块设备并挂载到实例中。块设备构成的卷已经被完全整合到计算服务和仪表盘服务中，用户可以创建和管理他们自己的存储。它除了使用主机拥有的卷之外，还可以使用上文介绍的Ceph分布式存储。主要用途是提供一个持久化存储的卷来存储实例产生的数据，并且提供强大的快照功能用来备份数据，并从快照中重新启动虚拟机。

2.2.2.6 基于Web的仪表盘Horizon

仪表盘提供了图形化的用户接口，使管理员和用户能够访问，管理，自动部署云计算资源。Horizon提供了三个种类的仪表盘，分别为用户仪表盘，系统仪表盘和设置仪表盘。这三个仪表盘覆盖了上文介绍的OpenStack核心组件，并提供了核心支持。它提供了一组用于核心OpenStack项目的API抽象，为开发人员提供了一致且稳定的可重用方法，这使得OpenStack开发人员不需要熟悉每个项目的API。

2.2.3 OpenStack网络服务安全组介绍

2.2.3.1 安全组简介

OpenStack网络服务提供了安全组的功能，它比之前由Nova提供的安全组功能要更强大，使用更加的方便。安全组是一个包含防火墙安全规则的一个容器，最新版本OpenStack使用的安全组位于网络服务Neutron服务中，它向后兼容了之前版本的Nova安全组。允许对进出实例的流量进行规则定义，并且是在Neutron的端口进行定义，保证了安全策略和实例的解耦。最重要的是，可以在实例运行的时候进行更改安全组的操作[17]。

2.2.3.2 安全组特点介绍

Neutron安全组定义了哪些流量可以进出虚拟机。安全组包括了一组防火墙策略，以网络为单位定义n个安全组，每个安全组内包含n个规则，本网络的实例可以使用n个安全组。

安全组可以控制进出虚拟机的网络包，位于Neutron的L2 agent节点上。它可以通过读取物理机的配置文件开启或者关闭对IPv6的支持，并且它是独立于于虚拟机之外，在物理机中虚拟出的qbr网桥上。它通过Neutron提供的端口进行访问控制，匹配的数据项可以是协议，端口，网段，方向。同时它的使用完全是动态的，即和虚拟机的运行状态无关从而随时更新安全组。当虚拟机进行迁移时，在目的物理主机上创建新的虚拟网桥，并通过安全的内网将安全组的数据进行传输。

3基于Libvirt的虚拟机迁移技术

虚拟机迁移是指，将虚拟机从它所在的物理机中转移到另外一台物理主机中。由于虚拟机是运行在虚拟化环境中，所以虚拟机在运行时不需要直接与底层的硬件进行交互。这给虚拟机迁移提供了基础[14][15]。

3.1 虚拟机迁移主要应用

负载均衡：虚拟机能够在它所在的主机超负荷运行的时候，迁移到另外一个负荷较轻的主机中运行。主机硬件升级：当我们需要对主机的硬件进行升级时候，无论是添加或者删除设备。我们都可以将主机中运行的虚拟机迁移到其它的主机中。这可以保证虚拟机在硬件进行任何更改的时候都不会停止服务。地理位置迁移：虚拟机能通过迁移，到达网络情况更为良好的地区来减少网络延迟。

3.2 虚拟机迁移的分类

3.2.1 在线迁移

在线迁移中，在内存页面被传输的时候，虚拟机在源主机上继续运行。在迁移过程中，有一个监视器会监控已经传输过去的页面中的更改，并在初始页面已经传送过去之后，对这些修改过的页面进行更改。当不断地对内存进行循环地拷贝过程后，最后对网络带宽和剩下的数据进行评估，若在当前带宽下传输剩下的数据小于默认的配置时间（KVM为10ms）就会暂停虚拟机，并将内存数据传输到目的主机中。这样一来，用户几乎感觉不到虚拟机迁移的过程。

3.2.2 离线迁移

离线迁移首先暂停虚拟机，然后创建一个虚拟机内存的镜像，并发送到目的主机。然后虚拟机在目的主机上启动。在源主机中虚拟机占用的内存就可以进行释放，整个过程严重依赖于网络的带宽和虚拟机的大小，如果网络负载严重或者带宽较低，整个迁移的过程将会非常漫长。

3.3 迁移流量的传输方式

迁移流量的传输有两种方式，通过Hypervisor原生的传输，还有一种是通过Libvirt构建的隧道进行传输。默认的传输方式是通过Hypervisor进行传输，当使用这种方式时，迁移流量是否加密取决于Hypervisor是否支持。

3.3.1通过Hypervisor进行传输

原生的数据传输，对加密的支持取决于使用的Hypervisor是否支持。但是不同的Hypervisor都会最小化需要拷贝的数据的大小用来提高性能。原生数据传输需要额外的对Hypervisor的网络进行配置，由于原生传输是多端口的数据传输，需要在防火墙中开放范围比较大的端口数。这种传输方式仅需要虚拟机之间进行数据传输，无需Libvirt的干预。

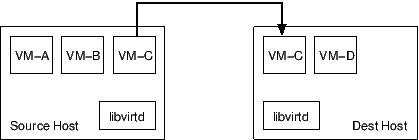


图3.1 通过Hypervisor的native传输

3.3.2 通过Libvirt隧道进行传输

使用隧道进行传输总是要经过Libvirt的RPC协议进行加密来保证安全性。但是由于提高安全性，性能也会降低。在传输过程中，源主机和目的主机都会有额外的数据拷贝。当虚拟机的内存过大，并且虚拟机内存不断被更新时，由于数据的增加会产生非常大的性能开销。在部署过程中，基于隧道的传输不需要开放过多的防火墙端口，只需要一个端口就可以完成迁移操作。这个端口是通过Libvirt进行传输，传输的流量都经过这个守护进程。

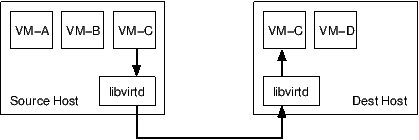


图3.2 通过Libvirt隧道进行传输

3.4 控制流量的传输方式

虚拟机的迁移，需要源主机和目的主机的密切交互。同时也需要有一个管理节点进行迁移控制流量的管理工作

3.4.1受管理的直接迁移

在受管理的直接迁移中，Libvirt客户端进程控制迁移的各个阶段。客户端进程必须能够和源主机和目的主机中运行的Libvirtd守护进程进行热证和连接。这并不需要源主机和目的主机中运行的守护进程进行交互。在迁移过程中，如果客户端应用程序失效或者它与两个守护进程的连接断开，迁移操作就会被中止并暂停虚拟机。

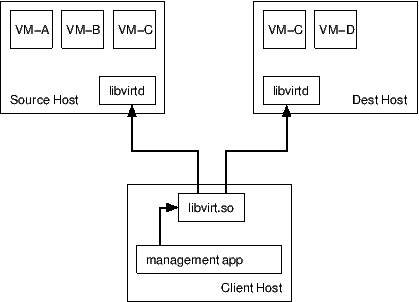


图3.3 受管理的直接迁移

3.4.2 受管理的对等迁移

当使用对等迁移时，Libvirt客户端进程仅仅和源主机上的守护进程进行通信。源主机的守护进程与目的主机的守护进程进行直接连接，从而控制整个迁移过程。如果Libvirt客户端进程在迁移过程中丢失了连接，或者客户端进程出错，迁移还会在源主机和目的主机之间继续进行，不会被中断。在这个情况下，源主机的守护进程使用它自己的身份证书直接与目的主机相连。

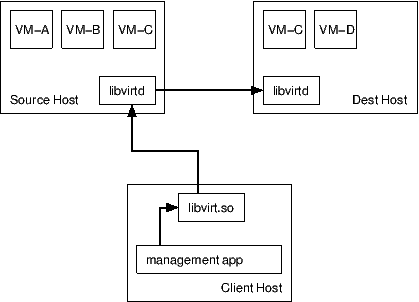


图3.4 受管理的对等迁移

3.4.3 无管理直接迁移

当进行无需管理的直接迁移时，Libvirt客户端和Libvirt守护进程都没有控制迁移过程。由原生的Hypervisor进行迁移过程的控制，Libvirt客户端仅仅只是初始化hypervisor的管理层，如果客户端或者守护进程故障，迁移也继续进行，不会终止。

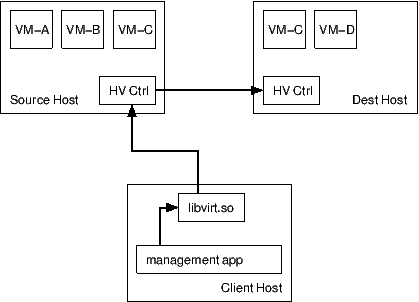


图3.5 无管理的直接迁移

3.5 迁移数据安全加密

3.5.1 TLS介绍

TLS是新版的SSL协议，它是直接在TCP协议上进行加密操作。所以对于应用层的协议无需做任何修改就可以使用安全的连接。当正确使用了TLS之后，即使有攻击者进行嗅探，可以发现数据包使用的端口，发送大约多少数据，使用的加密算法等。攻击者还可以尝试中断数据连接，一旦连接被第三方中断，发送方和接收方两端都能意识到连接受到了攻击。

TLS建立在TCP连接之上，具有以下几个特性：

TLS连接在传输数据的过程中使用对称加密算法对数据进行加密，每个连接的加密密钥都是唯一的，它是基于密钥交换协议生成的。在连接建立之后，进行数据传输之前，服务端和客户端会选择加密算法和使用的密钥。在密钥协商过程中，攻击者无法获得协商的密钥，一旦攻击者截获并更改协商过程的数据包，都会被通信双方检测到。

TLS连接使用公钥加密算法认证身份，在集群中部署CA证书之后，使用服务端证书和客户端证书的方式来验证身份。

TLS连接确保了数据的完整性，在TLS的整个连接过程中，每个数据包都包含一个消息认证码来检查消息的完整性，防止在传输期间有着意外数据的丢失或者更改。

3.5.2 基于客户端认证的TLS握手过程

TLS的握手过程如下[14][18]：

1. 客户端向服务端发送ClientHello消息，消息包括最高支持的TLS版本，一个随机数，建议的密码算法套件列表，压缩方法。

2. 服务端回复客户端一个ServerHello消息，包括选择的TLS协议版本，另一个随机数，密码算法套件，和压缩算法。同时消息还包括一个会话id，用来执行下一步的握手。

3. 服务端发送服务端的证书消息。

4. 服务端发送服务端密钥交换信息，密钥交换算法使用ECDH算法，通过Diffie-Hellman进行密钥交换，通过椭圆曲线对公钥和私钥进行加密。

5. 服务端向客户端请求客户端的证书，用来完成双向验证

6. 服务端向客户端发送ServerHelloDone的消息，表示服务端完成握手的内容

7. 客户端向服务端回复身份证书

8. 客户端向服务器传输客户端密钥交换信息，包括一个PreMasterSecret（同样是一个随机数），公钥，这个PreMasterSceret是由服务器证书的公钥进行加密的。

9. 客户端发送消息验证给服务端，该消息验证使用客户端证书的私钥进行签名，服务端可以使用客户端的公钥进行解密验证。这样服务器知道消息是由客户端发出的。

10. 客户端和服务端使用前面的两个随机数和PreMasterSecret计算出公共的会话密钥MasterSecret。之后所有的应用程序数据都要使用这个密钥进行加密

11. 然后客户端和服务器互相发送编码改变通知，使用当前商定的会话密钥进行加密，并且包含前面所有消息的消息认证码。如果这个消息解密失败，那么认为这个协商过程失效。如果消息解码成功，则后续消息的发送采用事先商定的加密传输。



图3.6 TLS的握手过程

4 基于OpenStack与Ceph的虚拟机安全迁移

4.1 实验环境

测试环境是在四台物理主机上部署Ceph与OpenStack，四台主机hostname分别为blade11，blade12，blade15和blade16。在OpenStack服务中依次为controller，compute1，compute2和compute3。物理机使用的操作系统为Ubuntu 16.04，Ceph版本为Jewel，OpenStack版本为Ocata。

对于Ceph分布式存储系统，blade11作为为管理节点，四个节点中都划分两块500GB的磁盘空间作为OSD存储。使用192.168.0.0/16以太网网段作为OSD交互的网段，使用10.0.22.0/24的InfiniBand网段作为存储数据传输的网段。对于OpenStack系统，使用192.168.0.0/16作为自服务的提供者网络为实例提供外网访问，使用10.0.22.0/24作为管理网络提供计算节点之间的数据传输。虚拟机所在的网络为自服务网络172.16.1.0/24，自服务网络和提供者网络192.168.19.0/24通过虚拟路由器相连。

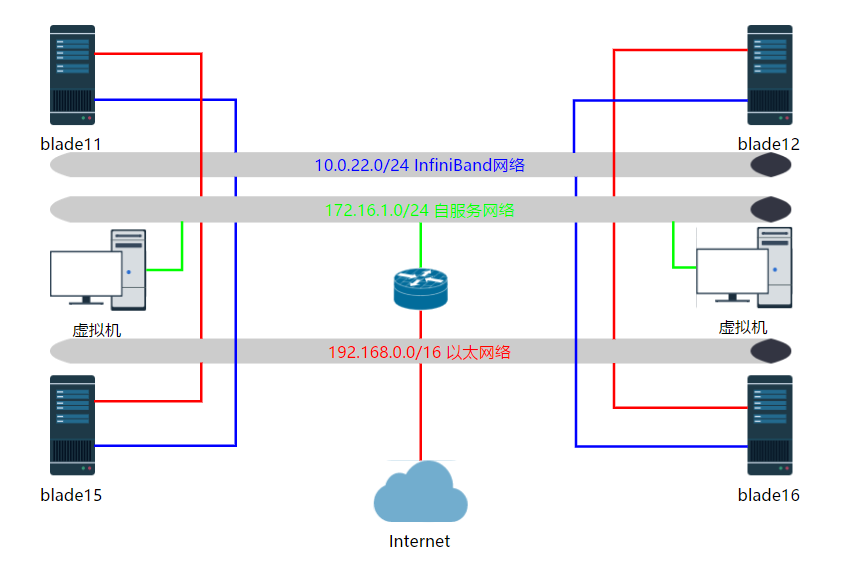


图4.1 实验环境拓扑图

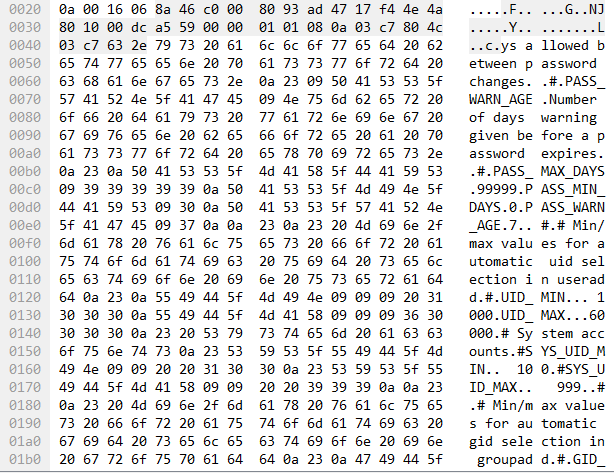
4.2 Ceph存储性能测试

我们进行了存储集群的性能测试，存储集群的性能需要多种因素共同决定，包括网络性能，磁盘性能，Ceph配置等。我们首先对网络性能进行测试，图中InfiniBand折线表示了集群的网络性能，在测试的10s之内总共传输了5.80GBytes的数据，平均带宽为637.44MByte/s。然后我们测试了集群使用的磁盘性能，磁盘在传输1.00GByte的数据,使用了4.37898s，平均速度245Mbyte/s。我们使用部署好的Ceph环境，进行部分Ceph读写测试。对于Ceph读写性能，在我们的测试环境下差别不大。大约在170Mbyte/s左右。Ceph性能和我们预期性能差别较大，可能是因为没有对Ceph进行性能调优。

图4.2 Ceph性能测试

4.3 安全迁移机密性完整性分析

我们在OpenStack与Ceph的环境中进行了虚拟机的迁移，首先我们进行了未加密的迁移测试。然后我们部署集群之间的TLS连接，加密迁移流量。在未进行加密的数据包中，我们能够很容易的找到内存中打开文件的数据。如果迁移流量没有进行加密，攻击者可能截获敏感信息的数据包，然后修改数据包之后再投放到网络上，通过这样的操作来进行敏感信息的更改。（例如图中看到的PASS\_MAX\_DAYS 99999）我们可以看到密码的最大期限和最小期限等，同时我们还可以看到登陆的用户名，id号等用户信息。



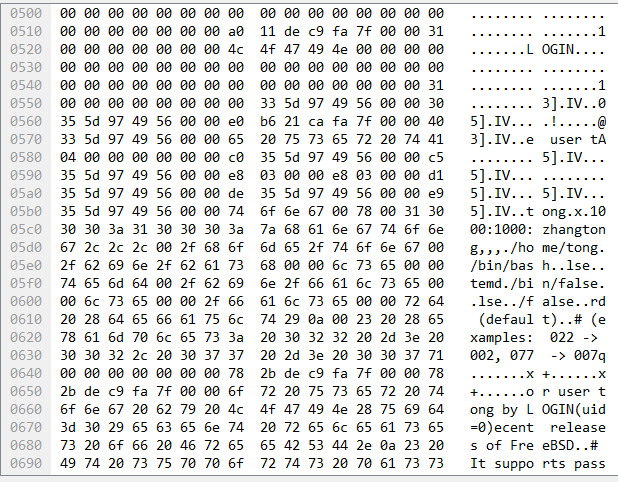


图4.3 未加密迁移流量的流量包内容

我们在启用TLS之后，对数据包进行分析之后发现，连接已经进行了加密，并且我们知道协议版本为TLSv1.2，并且我们可以观察到完整的TLS的握手过程。在传输过程中TLS通过密钥交换协议确定了本次迁移的会话密钥，用AES算法来加密数据保证机密性，使用消息摘要来保证数据的完整性同时使用数字签名来保证是由源节点发送。

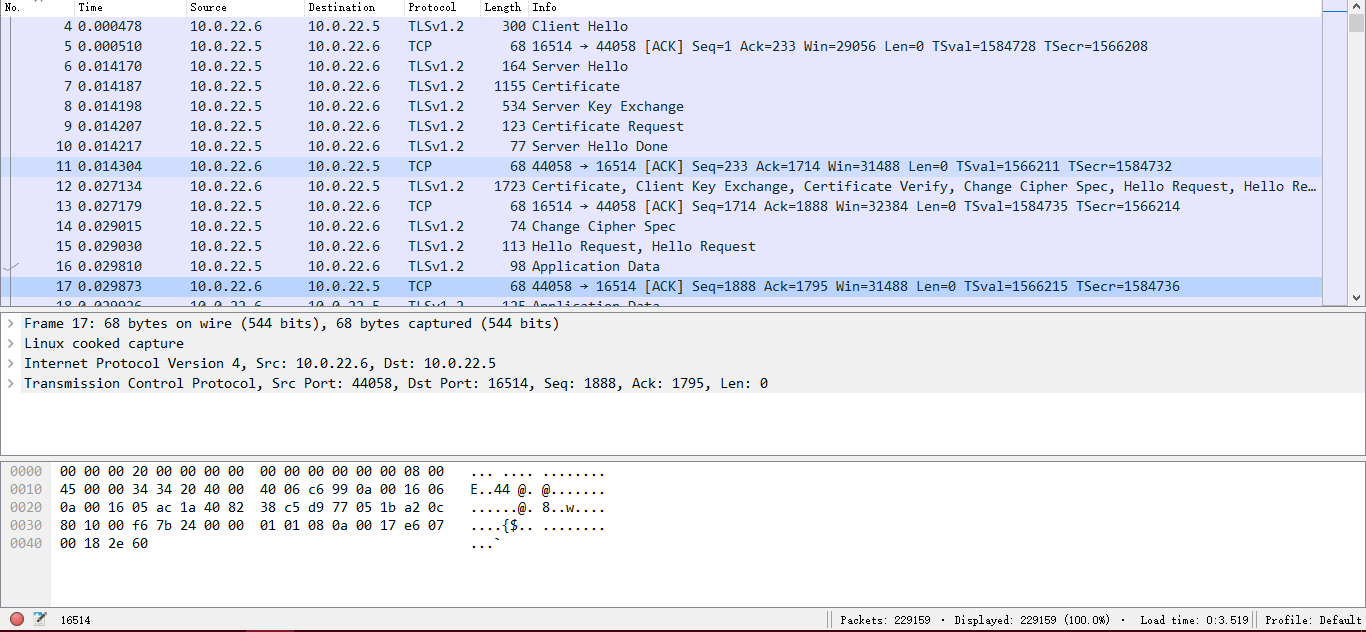


图4.4 经过TLS加密之后的迁移流量和TLS的握手过程

4.4虚拟机安全迁移效率分析

整个迁移过程中，对于未加密迁移，总流量大小为1358Mbyte，总耗时9s，平均速度147Mbyte/s。对于加密过后的迁移，总流量大小为1462Mbyte，总耗时17s，平均速度83Mbyte/s。对于同样环境，加密需要在数据传输开始前进行TLS的握手协议，验证源主机与目的主机身份，并生成会话密钥。未加密迁移直接就可以开始数据传输，所以在0s时刻未加密流量就可以达到比较高的水平。加密传输需要用到16514这个端口，未加密的迁移除了需要开放16514以外，还需要开放49152到49261范围内的端口。加密迁移为了提高安全性，所以取消了多端口的迁移，因此迁移的性能产生了较大的下降。

图4.5 虚拟机安全迁移效率分析

5 总结与展望

5.1 论文总结

随着云计算的发展，虚拟化技术也得到越来越多用户的重视和使用。虚拟化技术中的虚拟机迁移是解决负载均衡，容灾备份，主机硬件升级换代和异地迁移的主要方法。其中虚拟机迁移的安全性也应该得到我们的重视，虚拟机进行在线迁移时，会不断的迭代拷贝虚拟机内存中的内容（脏数据），将内存中的数据发送到目的物理主机中。由于虚拟机数据的流量未经加密，在经过公网时很容易受到安全威胁。本文主要研究了虚拟机迁移时的数据机密性和完整性的问题和虚拟机迁移时安全策略的分发的问题。

本文首先对Ceph分布式对象存储和OpenStack软件平台进行了介绍，并在后面着重介绍了虚拟机迁移的工作方式和虚拟机安全迁移的工作原理。其中我们使用了OpenStack中的Neutron服务提供的Security Group功能来保证虚拟机安全策略的实时迁移。并在最后给出了实验验证虚拟机迁移流量的安全性与机密性。并测试了Ceph分布式对象存储的性能和虚拟机安全迁移的性能。

实验表明，在经过安全配置之后确实保证了虚拟机迁移的机密性与完整性，但是在性能方面还是有较大的可以提升的空间。

5.2 未来工作展望

我们在已经完成基于OpenStack和Ceph的虚拟机安全迁移。但是我们还是需要进一步提高Ceph存储和虚拟机迁移的性能。我们可以尝试用更优化的配置来提高Ceph的性能，并且使用多端口的InfiniBand提高网络性能。

参考文献

1. Neto P. Demystifying cloud computing[C]. Proceeding of Doctoral Symposium on Informatics Engineering. 2011.
2. 王光波.云计算环境下虚拟机迁移机制研究[D].解放军信息工程大学,2013.
3. 梁新龙.云计算环境下数据安全存储及虚拟机安全迁移机制研究[D].东南大学,2015.
4. Divyambika R, Umamakeswari A. Protection of virtual machines during live migration in cloud environment[J]. Indian Journal of Science and Technology, 2015, 8(S9): 333-339.
5. 孙松儿, 吕振峰. 虚拟机安全策略的迁移方法及装置 [P]. 中国: CN 102739645 B,2016年3月16日.
6. Clark C, Fraser K, Hand S, et al. Live migration of virtual machines[C]. Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation-Volume 2. USENIX Association, 2005: 273-286.
7. Weil S A, Brandt S A, Miller E L, et al. Ceph: A scalable, high-performance distributed file system[C]. Proceedings of the 7th symposium on Operating systems design and implementation. USENIX Association, 2006: 307-320.
8. 吴香伟. Ceph剖析：数据分布之CRUSH算法与一致性Hash[EB/OL]. https://www.cnblogs.com/shanno/p/3958298.html?utm\_source=tuicool.
9. Ceph. Ceph Documentation[EB/OL]. http://docs.ceph.org.cn/man/8/rados/.
10. Christian, Huebner. Ceph vs Swift – An Architect’s Perspective[EB/OL]. <https://www.mirantis.com/blog/ceph-vs-swift-architects-perspective/>.
11. TAMARA, SCOTT. Big Data Storage Wars: Ceph vs Gluster[EB/OL]. http://technologyadvice.com/blog/information-technology/ceph-vs-gluster/.
12. OpenStack. The World[EB/OL]. https://www.openstack.org/user-stories/.
13. OpenStack. OpenStack Wiki[EB/OL]. https://wiki.openstack.org/wiki/Main\_Page.
14. Libvirt. TLSSetup[EB/OL]. <https://wiki.libvirt.org/page/TLSSetup>.
15. Libvirt. Guest migration[EB/OL]. <https://libvirt.org/migration.html>.
16. Wikipedia. Hypervisor[EB/OL]. https://zh.wikipedia.org/wiki/Hypervisor.
17. 世民. Neutron 理解 (8): Neutron 是如何实现虚机防火墙的 [How Neutron Implements Security Group][EB/OL]. <http://www.cnblogs.com/sammyliu/p/4658746.html>.
18. Wikipedia. 传输层安全协议[EB/OL]. https://zh.wikipedia.org/wiki/%E5%82%B3%E8%BC%B8%E5%B1%A4%E5%AE%89%E5%85%A8%E5%8D%94%E8%AD%B0.

致谢

这篇论文的完成，首先要感谢我的指导老师李涛老师，在我在武汉科技大学的这段时间的指导。从大一到现在四年的时间。这几年老师对我的教导不止体现在实验和研究上，每当遇到困难和瓶颈的时候，老师总会有耐心的倾听和包容学生的不足，提出宝贵的建议。另外，老师在专业上的知识和对学术严谨的态度深刻影响着我。