doi:10.11959/j.issn.1000-436x.2017229

TCCL：安全高效的拓展云桌面架构

王斌，李伟民，盛津芳，肖斯诺
（中南大学信息科学与工程学院，湖南 长沙 410083）

摘 要：在传统云桌面系统的基础上，结合透明计算模式，提出了一种安全增强、存储高效的云桌面系统架构：TCCL（transparent computing-based cloud）。将透明计算理论中提出的“算存分离、流式加载”思想应用于云桌面系统，将对安全威胁的防御部署在云主机层次之下，增强了云桌面系统中云主机的系统文件与数据这一层面的安全性，并提升了云主机文件存储效率。
关键词：IaaS；桌面云；透明计算；云安全；云存储
中图分类号：TP301 **文献标识码**：A

TCCL: secure and efficient development of desktop cloud structure

WANG Bin, LI Wei-min, SHENG Jin-fang , XIAO Si-nuo
(School of Information Science and Engineering, Central South University, Changsha 410083, China)

Abstract: A secure and efficient development of desktop cloud structure: TCCL (transparent computing-based cloud), which was designed under the guidance of transparent computing, was proposed. TCCL applied the method, separating calculation and storage, loading in a block streaming way which was proposed in the transparent computing theory, to the cloud desktop system, and deployed the defense module of security threats under the cloud VM (virtual machine). As a result, the TCCL could improve the security level on the cloud VMs' system files and data files, and could optimize the cloud virtual machines' storage efficiency.
Key words: IaaS, desktop cloud, transparent computing, cloud security, cloud storage

1 引言

随着计算机硬件性能的不断提高，成本不断下降，虚拟化技术和基于虚拟化技术的云计算模式开始逐渐成熟。云计算将资源进行虚拟化，并以服务的形式提供给用户。其服务模式可以分为 3 个层面：基础设施即服务（IaaS，infrastructure as a service），为用户按需提供虚拟或实体的计算、存储和网络等资源；平台即服务（PaaS，platform as a service），提供应用程序运行环境部署和管理服务；软件即服务（SaaS，software as a service），提供云计算基础平台之上的应用程序服务。其中，IaaS 服务模式中的云桌面服务受到了越来越多的关注。其采用服务器虚拟化技术在数据中心统一运行并管理所有的虚拟桌面系统，并且把桌面作为一种服务(desktop as a service)提供给用户[1]。云桌面服务在基础的 IaaS 服务架构之上，整合了更多的虚拟化资源，所以也面临着多个层次的安全问题，也更强调数据存储、资源利用的合理性和安全性。

现有的桌面云平台（如阿里云、QingCloud）一般以服务器集群的方式部署，将控制节点、计算节点、存储节点和网络节点分布在不同的服务器上。这种分离式的集群架构可以有效地降低云服务之间的耦合，并支持增量式的拓展。然而，由于桌面云是以云主机的形式封装底层的虚拟

收稿日期：2017-09-15
基金项目：国际科技合作与交流专项基金资助项目（No.2013DFB10070）
Foundation Item: International Science & Technology Cooperation Program of China(No.2013DFB10070)

化资源，云主机的运算和云主机文件的存储都在云平台的计算节点服务器上，这导致了单台服务器可以同时运行的云主机数量比较有限，云主机文件之间也存在大量的冗余。此外，现有桌面云平台的安全机制，是以直接附加在云平台各个层次上实现的，这给云平台自身带来了不小的额外开销，从而会在一定程度上影响云服务的效率和质量。

透明计算（TC，transparent computing）是一种新型的网络计算模式，用户所需的资源，包括操作系统、应用程序和用户数据都集中存储在服务端。客户端仅需保留最底层的 BIOS、极少部分相关协议和管理程序，对服务端资源进行按需加载并流式执行。这种模式保障了客户端设备的安全性，并且可以将安全保障机制下降到操作系统以下的层次，从而能根本性地消除各类安全威胁。

为此，本文基于透明计算的思想，整合现有的云桌面服务架构与透明计算服务架构，以 TC 服务器作为后台服务器，以云桌面服务器之上的云主机作为 TC 的客户端，提出了一种安全可拓展的云桌面架构（TCCL，transparent computing-based cloud）。该架构在后台采用分层存储模型，减少了云主机文件存储冗余，提升了桌面云整体的存储性能。另外，该架构充分发挥透明计算在安全方面的优势，将对安全威胁的防御部署在云主机层次之下，增强了云桌面系统云主机系统文件与数据层面的安全性。

2 相关研究

云计算安全在云计算的研究领域被认为是至关重要的一环，原因在于用户的敏感数据和重要信息都存储在云服务端[2]。因此，云服务端安全性的缺乏会导致用户对于云平台的信任度急剧下降[3]。云计算技术不断快速深入发展的同时也增加了云计算安全保护的挑战，无论是 PaaS、IaaS 还是 SaaS 云计算服务模式，都面临着多租户共享资源所带来的安全威胁[4,5]。虚拟桌面目前受到了越来越多的关注，它统一管理桌面虚拟主机，并把云主机虚拟桌面作为一种服务提供给用户，从而帮助企业集成 IT 资源，降低成本[1,6]。但作为一种 IaaS 服务模式，虚拟桌面服务架构面临着多个层次的安全问题，也更强调数据存储、资源利用的合理性和安全性。

首先是来自云服务器物理主机的安全威胁，多租户环境中的同一台物理主机上可以同时运行不同信任级别的虚拟机，并且虚拟机也会不断地在不同的物理主机之间，甚至是公有云和私有云之间迁移。但是，系统管理员或任何一个被赋予高权限的用户都可能存在恶意行为，进而对同一主机上的虚拟机以及主机本身，甚至是其他虚拟机造成安全威胁[7,8]。其次是来自其他虚拟机的威胁，虽然虚拟化技术能很好地对不可信虚拟机进行隔离或迁移，但云平台上的虚拟机间的通信和协作也是不可缺少的，恶意程序很可能借此渠道侵犯其他虚拟机。此外，恶意虚拟机也可以通过内存碎片访问到其他虚拟机。较常见的拒绝服务（DOS）攻击也通常由其他恶意虚拟机发起，入侵检测系统（IDS）是防御这一类攻击的有效方法[9,10]。最后是来自网络的安全威胁，常见的防御来自网络层面的安全攻击技术手段包括逻辑网络分段、防火墙、流量加密和网络监控等[11,12]。

透明计算的资源远程加载、流式执行的特点使其在安全方面具有独特的优势。文献[13]详细描述了透明计算的安全特征，并通过实例进行了展示。其最大的优势是可以在软件层面将安全威胁防御在操作系统层次以下，从而提高终端的安全性。但透明计算也面临一些安全方面的问题，比如在用户的权限验证与用户账户安全保障方面还有不足[14]，而云计算在这方面的工作却较为完善。

从安全的角度分析，透明计算的优势在于可以解决操作系统之下底层的安全问题，而云计算在 VMM 之上云平台本身的用户账户体系安全、网络安全和云主机防入侵等方面更具优势。综上分析，可以认为云计算和透明计算在安全方面具有一定的互补性。

3 基于透明计算的拓展桌面云架构—TCCL

3.1 TCCL 总体架构

如图 1 所示，TCCL 架构是基于透明计算理论拓展的一种云桌面架构，旨在整合市场上碎片化的异构云数据资源，提升云主机数据文件存储效率，深化异构云之间的数据融合。并通过 TC 后台服务端的访问控制模块在数据块粒度实现对所有云主机的程序和数据的读写服务以及监控，从而提升传

统桌面云架构的安全性。

TCCL 的核心思想是将多个异构的桌面云系统接入统一的 TC 后台服务端，将 TC 后台服务端作为真正存储云主机程序和数据的后台。对于桌面云用户来说，可以继续通过 PC、智能手机、平板电脑等终端使用原有的方式访问桌面云服务，而对云桌面的程序和数据的访问请求将重定向至 TC 后台服务端处理，对于用户来说这个服务过程是透明的。

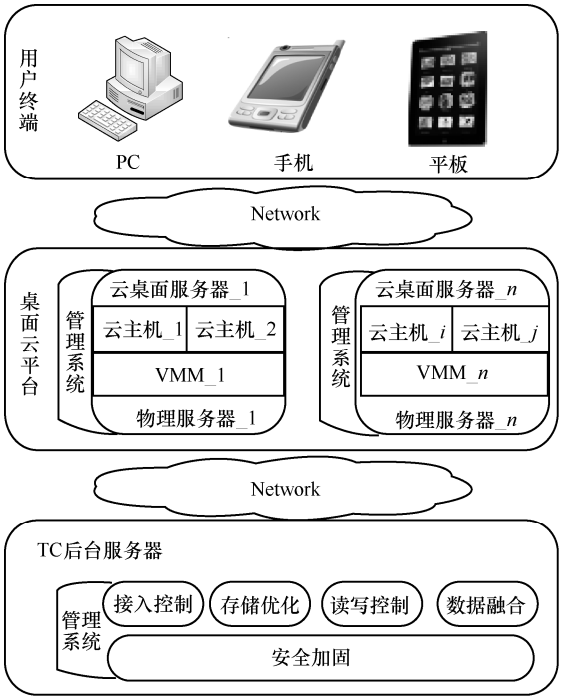

图 1　TCCL 总体架构

TCCL 架构有如下特点。

1）云主机的存储和计算分离。引入透明计算的算储分离的思想，云主机所需的程序和数据均存储在 TCCL 中的 TC 后台服务端并进行集中统一管理，通过网络按照特定的数据访问协议动态按需提供给云主机用户，程序的执行则在云主机的本地端。此外，TC 后台服务端存储的云主机数据采用的是多层链式结构，有效地减少数据的冗余，提高存储的效率。

2）分担和转移云平台的安全服务及其所带来的开销，并增强云数据的安全性。由于云主机的程序和数据不是存储在本地，其安全保护机制也将跟随转移并集中至 TC 后台服务端，从而有效分担和转移了云平台安全服务以及所带来的额外开销，使云平台可以专注将有限的资源用于为用户提供更高效的服务。同时，TCCL 并未对云平台固有的安全技术产生破坏，只是对其安全服务结构略有改变。因此，TCCL 可以在充分保留云平台原有的安全策略的基础上，在 TC 后台服务端进一步构建安全保护机制，增强云数据的安全性。

3）高易用性和扩展性。TCCL 架构是将云桌面平台与透明计算服务平台进行有机整合，不需要对桌面云系统内部接口进行改造，只需为各个桌面云平台提供统一的透明计算接入接口即可随时随地动态获得 TC 后台服务端提供的服务。同时，TC 端也拥有统一的服务管理平台，集中管理资源与安全服务，并且对桌面云用户来说是透明的。因此，TCCL 具有高易用性和扩展性。

3.2　基于资源共享的 TCCL 后台数据存储模型

用户云主机的所有资源，包括操作系统、应用程序和用户个性化数据，都以虚拟磁盘的形式统一存储并管理在 TC 后台服务端。TCCL 存储优化的关键在于尽量共享云主机之间通用的资源，减少 TC 后台服务器端云主机虚拟磁盘存储系统中冗余数据量。具体来看，TC 后台服务端虚拟磁盘中数据资源按资源共享程度及性质划分成如下 3 类。

1）OS 级资源（S_VID）。主要指操作系统以及相关数据，该类资源共享程度最高，基本能被所有用户共享。

2）应用软件群组资源（G_VID）。主要指一种 OS 环境下各种相关的应用软件数据组成的集合，该类资源的共享度次之，主要被同一群组下的用户共享。

3）用户个性化资源（U_VID）。主要指用户区别于其他用户的数据资源，一般包括用户私有数据文件或操作系统、应用软件的个性化配置信息，该资源共享度最低，只能提供给用户自身访问。

TC 后台服务端虚拟磁盘对应的数据存储模型如图 2 所示，它是一种基于树状的按共享程度划分层次的存储模型，不同的 OS 环境及其下方的应用群组和用户数据组成一个存储森林。每棵树的树根存储了支持操作系统启动所必须加载的数据，这一部分可以被每台云主机共享。在树的第二层节点到非叶子节点，针对每个不同的用户群组，为其配置不同的群组虚拟磁盘，群组虚拟磁盘中存储了相关的软件资源，能被群组中的云主机共享。而在树的最底层，为每台云主机配置了存储私有数据和修改

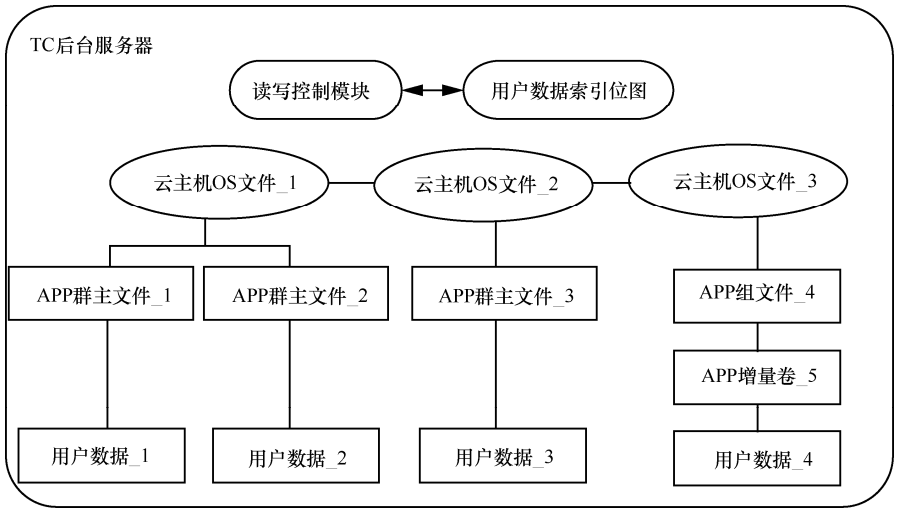

图 2 TC 服务端数据资源组织模型

数据的用户数据虚拟磁盘镜像。由于不同云主机只会修改少量系统和应用数据，因此，针对每个用户虚拟磁盘，只需要分配少量的存储空间。对于用户来说，用户感知到的是一个完整的包含 OS、应用软件和个人私有数据的完整磁盘，而事实上，在服务端上这些数据被划分成多个部分。

4 TCCL 数据访问控制模型

4.1 虚拟磁盘数据访问机制

云主机操作系统文件和 APP 群组文件存储于 TC 后台服务端，被多个云桌面用户共享。本文将写时重定向（ROW，redirect on writing）机制应用于 TCCL 虚拟磁盘存储模型。采用写时重定向机制将云主机对共享的虚拟磁盘镜像 S_VDI 和 G_VDI 的改写块保存于与用户对应的用户虚拟磁盘镜像 U_VDI 中，并利用 Bitmap 来标记各个改写块的位置，从而实现了多个云主机对共享数据的共同读写。

图 3 描述了在 ROW 机制下云主机 *i* 写数据块 1 后再读取数据块 1 和数据块 2 的数据访问流程。在用户 *i* 写数据块 1 时，直接重定向将数据写入至该用户对应的 U_VDI，并且在位图索引中修改相应的比特。如果用户 *i* 再次修改数据块 1 的数据，则直接覆盖 U_VDI 中数据块 1 的旧数据，不用对位图进行修改。

在终端客户 *i* 请求第 1 和第 2 号数据块时，由于其对应的 U_VDI 中存储了数据块 1 的改写块，因此，对数据块 1 的读请求被定位到 U_VDI，而数据块 2 在 U_VDI 的位图中都没有改写标记，因此其请求定位到了 G_VDI，最后将所有读取到的数据块重新排序组合后返回给云主机。

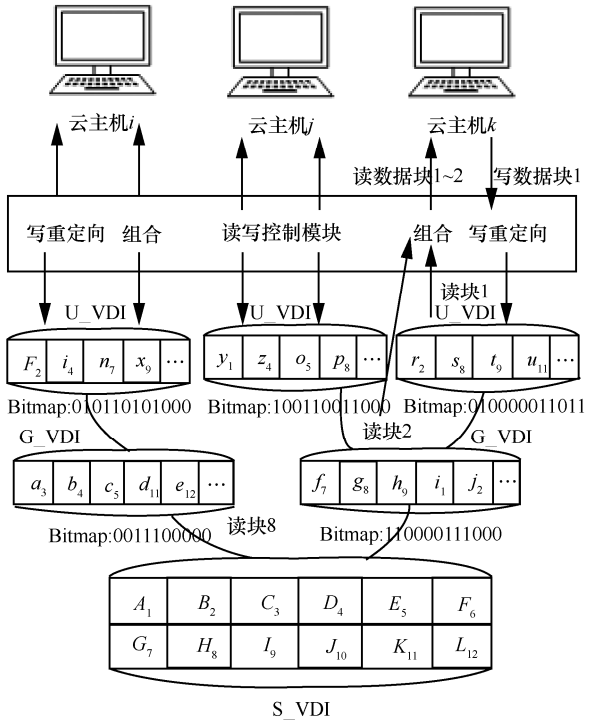

图 3 多用户共享的虚拟磁盘数据访问机制

4.2 虚拟磁盘存储结构

TCCL 的虚拟磁盘存储模型可以解决多用户虚拟磁盘共享数据一致性的问题，但是需要一个合适的虚拟磁盘存储结构来支持该模型，一方面实现高效的访问和共享，另一方面支持用户对虚拟磁盘的安全访问。本文设计了一种基于位图[15]的链式 Vdisk 存储结构，在该结构中，每个虚拟磁盘镜像都被视作一个存储节点，各节点之间通过指向节点的指针相关联，该存储结构能够支撑本文提出的存

储模型及 ROW 机制，为用户提供了较低粒度的高效数据访问。同时，为该架构中的每个节点设计一个表示数据的敏感等级的新型染色位图，控制用户对数据块的安全访问权限。

如图 4 所示，本文提出的虚拟磁盘存储结构主要由 Header、Stor_Bitmap、Col_Bitmap、Q_table、Data 这 5 个部分组成。其中，Header、Stor_Bitmap、Col_Bitmap、Q_table 构成元数据文件，Data 部分构成数据文件，元数据文件和数据文件可连续存储也可分离存储。

其中，Header 区域处于虚拟磁盘镜像的开始部分，它记录整个虚拟磁盘镜像的基础信息，其具体描述如表 1 所示。

存储位图（Stor_Bitmap）。用于记录修改块的位置。镜像节点初始化过程中，在存储位图区域开辟 *Bitmap_size* 大小的空间，且全初始化为 0，位图中每个位都与虚拟磁盘数据块序列中的每个 Block 一一对应，式(1)为 *Bitmap_size* 计算方法。在此后数据访问过程中，若对父节点 Block 序列发生修改时，需要将对应的位置为 1。

$$Bitmap\_size = \frac{size}{B\_unit} \tag{1}$$

染色位图（Col_Bitmap）。用于记录数据块敏感度。镜像节点在初始化过程中，在染色位图区域开辟由式（1）计算出的大小空间。与存储位图不同，染色位图中每 2 个位与虚拟磁盘数据块序列中的每个 Block 一一对应，在初始化值可以为 00、01、10、11，分别表示白色、绿色、黄色和红色，按数据块敏感程度从低到高排列。其中，U_VDI 的染色位图全部初始化为白色，G_VDI 的染色位图根据应用程序对用户的敏感程度初始化为绿色和黄色两类，S_VDI 的染色位图则全部初始化为

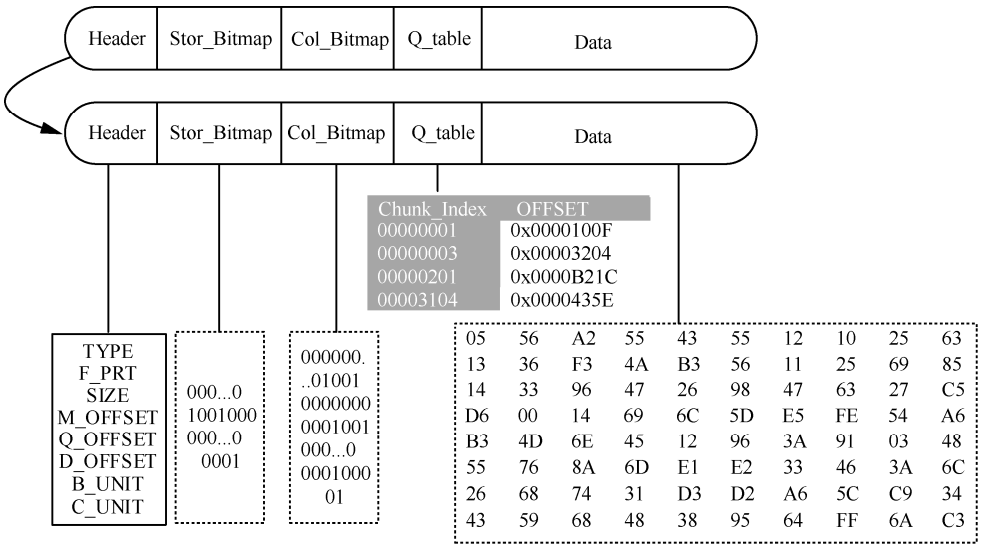

图 4　虚拟磁盘存储结构

表 1　　Header 区域成员描述

成员	描述
vid	64 位整型唯一标识一个虚拟磁盘镜像节点，用于区别其他节点
type	表示节点类型，共包含 4 种类型：系统类型、群组类型、用户类型、快照类型
f_ptr	指向父节点镜像的指针，当某节点指向父节点时，可共享父节点镜像中的数据，并且只保留对父节点数据修改的部分，系统类型节点的 *f_ptr* 为空
size	表示镜像节点大小，即 Bitmap 所能表示的磁盘总大小，单位为 B
M_offset	表示 Bitmap 区域相对于镜像起始位置的偏移量，位图区域紧随 Header 区域，而 Header 大小固定，因此该值是个固定值
Q_offset	表示 Q_table 区域相对于镜像起始位置的偏移量，通过它能快速定位到 Q_table
D_offset	表示 Data 区域相对于镜像起始位置的偏移量，通过 D_offset 和数据块号可计算访问数据的具体位置。
B_unit	划分 Block 的基本单位
C_unit	划分 Chunk 的基本单位，Chunk 由连续多个 Block 组成，大小固定，$C\_unit = n \times B\_unit$，其中，n 为整数

红色，表示敏感等级最高。需要指出的是，所有镜像节点的染色位图一经初始化则用户不允许修改。

查找表（Q_table）。将发生改写 Chunk 的索引号与该 Chunk 在 Data 区域的位置偏移量一一映射，实现加快查询速度的效果。只有发生改写的 Chunk 才会被记录于表中，其大小可依据数据改写量而动态调整。

数据区域（Data）。用于存储每个节点相对于父节点所修改的数据，保存更新后数据以区别于父节点共享数据。

4.3 数据访问控制流程

TC 服务端数据访问控制模块的具体工作时序如图 5 所示（其中，虚线表示返回消息），当访问控制模块（access controller）收到云主机发送的远程数据访问请求时，首先在 TCCL 中融合了各异构云平台用户的数据库（user DB）中查询获取用户设定的安全等级。用户的安全等级分为 0、1、2、3 这 4 个等级，与染色位图敏感等级中的白色、绿色、黄色、红色一一对应，对于没有设定安全等级的用户默认为白色。得到用户安全等级后查询染色位图（colored bitmap），获取请求访问数据块的敏感等级（security level）。如果用户安全等级小于数据块的敏感等级，则要求用户数据密钥进行验证，如果验证失败，启动安全拦截程序。如果验证成功，则查询用户的存储位图（user bitmap），进而确定数据块在的虚拟磁盘区域（user/system data area）中的位置。最终实现对该数据块的访问（读或写）。

算法 1 详细描述了读访问的过程。首先解析出地址偏移量（请求数据中第一块的开始地址）和数据长度，根据和计算该请求的数据块所属的 Block 编号以及 Chunk 编号，计算数据所属 Block 在所属 Chunk 中的起始序号，计算式如下所示。

$$Block\_index = \left\lfloor \frac{Offset}{B\_unit} \right\rfloor \tag{2}$$

$$Chunk\_index = \left\lfloor \frac{Offset}{C\_unit} \right\rfloor \tag{3}$$

$$Start\_no = \left\lfloor \frac{Offset\%C\_unit}{B\_unit} \right\rfloor \tag{4}$$

对比用户安全等级和所访问的数据块敏感等级，如果小于则返回用户重新验证，验证成功则转至步骤 10)，如果失败，启动云平台自身所拥有的安全拦截模块。如果大于，则直接跳转至步骤 10)通过用户节点的存储位图判断所需读取 Block 是否被用户改写。若已被改写，查询用户缓存是否缓存了该 Block，若该 Block 已被缓存则直接在内存中访问，

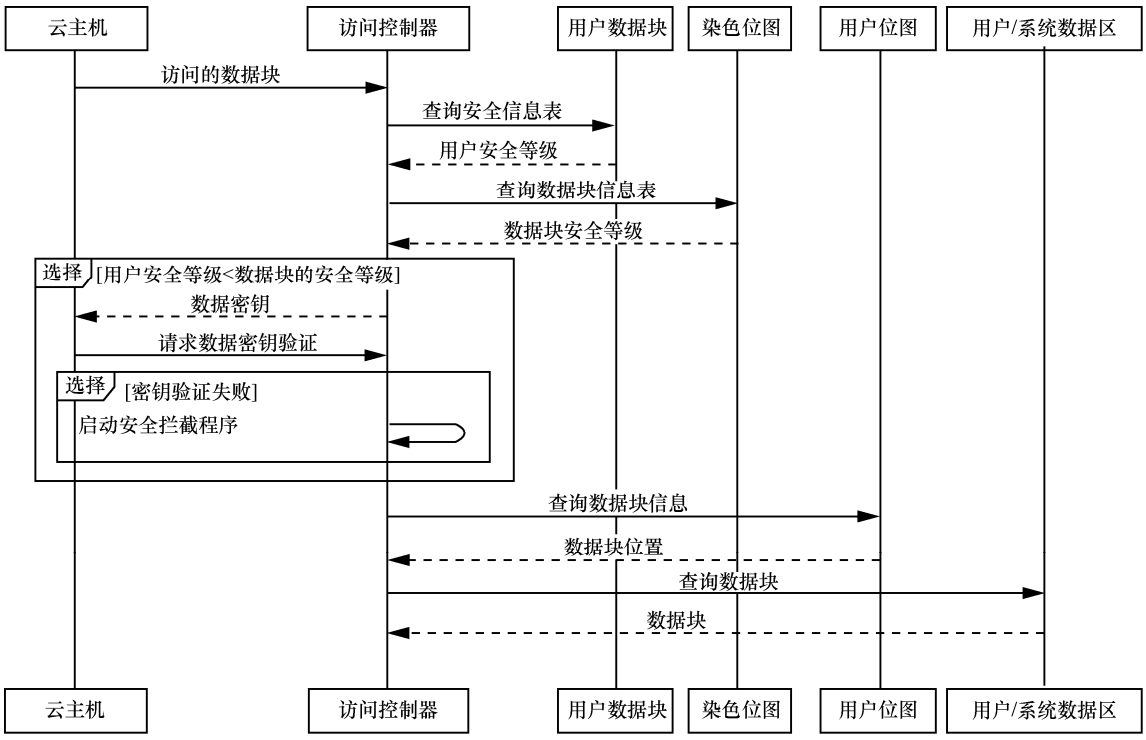

图 5 数据访问控制时序

否则先将数据先载入缓存再进行访问，转至步骤 14)；若查询 Bitmap 判断用户未改写该 Block，则判断当前访问的节点是否是存在父节点。若不存在，程序抛出数据读取异常；若存在，则通过父节点的 Bitmap 判断所需读取 Block 是否在父节点中被改写，如果被改写，则从父节点 Data 区读取数据，转至步骤 14)；若未被改写，则返回继续执行步骤 10)。

判断数据是否全部读取完全，若读取完全继续执行步骤 15)，否则读取下一个 Block 跳转回步骤 3)。将读取的内容合并，去除首尾填充部分，读取流程完毕。

算法 1　TC 后台服务端 Vdisk 数据读取算法

输入

Vdisk 编号：*vid*;

读取数据块的偏移量：*off*;

读取数据块的长度：*len*;

输出　即将读取的数据序列：$D[0,\cdots,n-1]$;

1) 初始化　$D[0,\cdots,n-1]$;

2) for $i = 0; i < len; i{+}{+}$ do

3) 根据偏移量和数据长度计算数据块 $Block_{vid}^{off+i}$ 的 Block_index 和 Chunk_index ;

4) 计算 $Block_{vid}^{off+i}$ 的 Chunk 起始编号;

5) 查询并对比数据块的 Col_Bitmap 和用户的安全级别;

6) if 用户的安全级别小于数据块的 Col_Bitmap 值 do

7) 启动安全验证;

8) if 安全验证失败 do

9) 启动云平台的安全拦截模块;

10) 查询用户的 Stor_Bitmap;

11) if 需读取的数据块被改写 do

12) 在缓存中查询该数据块;

13) if 存在本地缓存中 then

14) $D[i] = Block_{vid}^{off+i}$

else

15) 从远程服务端获取该数据块;

16) 更新本地缓存;

17) else

18) while 当前访问的节点存在父节点 do

19) 查询父节点的 Stor_Bitmap;

20) if 需读取的数据块在父节点中被改写 then

21) 从相应的数据区读取该数据块;

22) break;

if 数据读取完毕 then

23) 合并读取的内容;

else

24) goto 步骤 3);

25) end

由上述的访问控制流程分析可知，TCCL 架构的后台读写控制模块在保留传统云桌面系统的防御功能基础上，同时通过其后台文件读写控制模块对恶意侵入云主机后的文件破坏、用户权限提升等操作进行监控和防御。从而将对云服务端的安全监控下沉到云主机的更底层。

5　原型系统搭建与性能测试

5.1　原型系统搭建

TCCL 原型系统以开源系统为基础：云桌面系统基于 OpenStack 搭建，底层采用 KVM 提供虚拟化支持，云主机与 TC 后台网络存储的构建采用 iSCSI Enter-prise Target(IET)，云主机搭载 Win7 操作系统提供虚拟桌面服务，云桌面服务器和 TC 后台服务器网络配置在同一个局域网内。

TC 后台服务器采用 3 层存储模式，分别存储云主机操作系统文件、APP 群组文件、用户个人数据文件。服务器配置参数如表 2 所示。

表 2　**实验配置参数**

服务器类型	硬件配置	OS 及相关软件
云桌面	Model：Dell OptiPlex	CentOS release 6.5
云服务器	9020MT i7 CPU：Intel(R) Core(TM) i5-3470 @3.20 GHz MEM：8 GB DISK：6 TB Network：千兆网卡	Openstack 全功能组件
TC 后台	Model：Dell OptiPlex	CentOS release 6.5
透明服务器	9020MT i7 CPU：Intel(R) Core(TM) i5-3470 @3.20 GHz MEM：8 GB DISK：6 TB Network：千兆网卡	Jdk1.7、Eclipse 10、Tomcat7.0、iscsi-target-utils
交换机	H3C S5024P	

TCCL 具体实施的集群部署方案如图 6 所示。OpenStack 集群采用 4 节点部署，包含 1 个控制节点 3 个计算节点，与 TC 后台服务器配置在同一个局域网内，构成 TCCL 集群。另外在 OpenStack 集

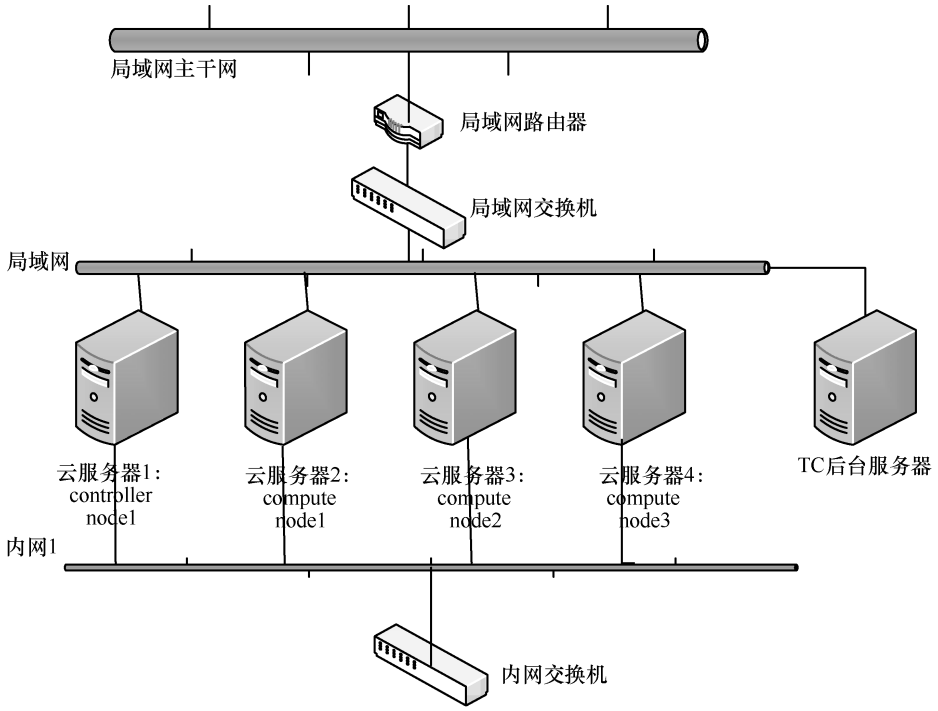

图 6 TCCL 集群部署

群节点之间接入另一个路由器，为 OpenStack 之上的云主机组成内部局域网环境。用户客户端远程访问云桌面采用开源的 virt-viewer 工具。由于本实验侧重于 TCCL 架构的功能验证和性能评估，故不对用户终端访问虚拟桌面的直观效果进行对比分析。实验部分将从 2 个方面展开：TCCL 远程虚拟磁盘读写性能测试以及 TCCL 架构下云主机文件扫描时间测试，重点分析 TCCL 原型系统与传统云桌面系统的云主机读写性能差异和云主机文件存储性能差异。

5.2 单机文件读写性能测试

传统云桌面系统的云主机磁盘读写一般发生在云主机所在的物理服务器上，而在 TCCL 架构下，云主机的磁盘读写都需要访问网络远程的 TC 后台服务器，经由读写控制模块完成对磁盘文件的读写。本实验评估和比较了 TCCL 和传统云桌面下数据读写性能，利用 CrystalDiskMark 工具，读写数据量为 1 GB，重复进行 5 次读写取平均读写速度为最终结果。为了全面测试读写性能，在以下 4 个测试条件下进行了实验。

1) 1 MB seq。单线程单队列 1 MB 单元块顺序读写。

2) 4 KB random。单线程单队列 4 KB 单元块随机读写。

3) 128 KB seq Q32。单线程 32 队列 128 KB 单元块并发顺序读写。

4) 4 KB random Q32。单线程 32 队列 4 KB 单元块并发随机读写。

实验结果如图 7 和图 8 所示。

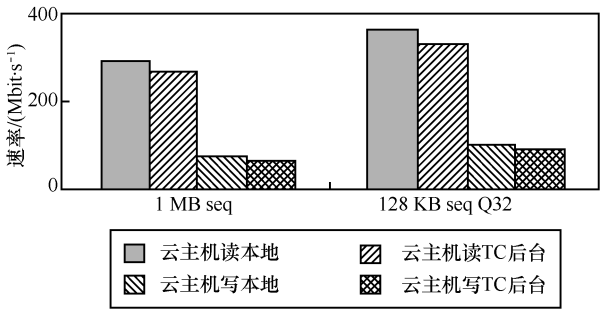

图 7 云主机顺序读写本地和 TC 后台速度对比

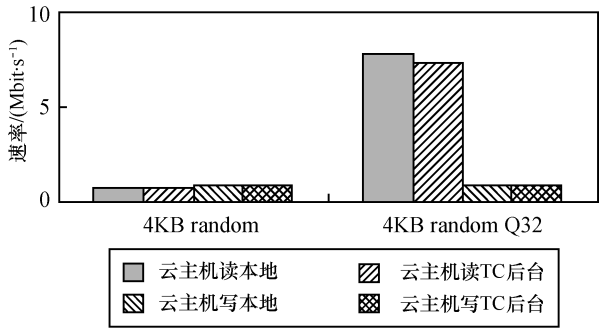

图 8 云主机随机读写本地和 TC 后台速度对比

从图 7 和图 8 中可以看出，在测试工具给出的 4 个测试条件下，对于单机测试来说，无论是读或

写，TCCL 架构下读写存放在远端 TC 后台服务器的虚拟磁盘速度均略低于直接读写云主机本地（即云桌面服务器所在的物理服务器）。从 TCCL 架构上分析可以认为这里的读写性能下降主要来源于网络传输以及 TC 后台服务器上的读写控制模块的读写重定向操作。从数据上分析，相对于传统的云主机本地读写，TCCL 读写性能的损耗基本在 2%~9%。

5.3　云主机实例文件扫描时间测试

为了直观地对比传统云桌面系统中的整个云主机文件扫描与 TCCL 架构下云主机文件扫描的效率，本文在 2 种架构下做了对比实验。本文选取的传统云桌面架构下单个云主机大小为 50 GB，使用 TCCL 架构拓展之后云实例和指定 APP 群组存储空间之和为 40 GB，用户增量卷大小为 10 GB，使用 python 语言编写文件扫描脚本模拟对于云主机文件的安全扫描操作。本文分别在 2 种架构下同时运行 1、2、4、8、10 个虚拟桌面，分别在各自的服务器磁盘存储系统中扫描云主机文件，记录下不同数量的云主机同时运行的情况下各自所需的扫描时间。实验结果如图 9 所示。

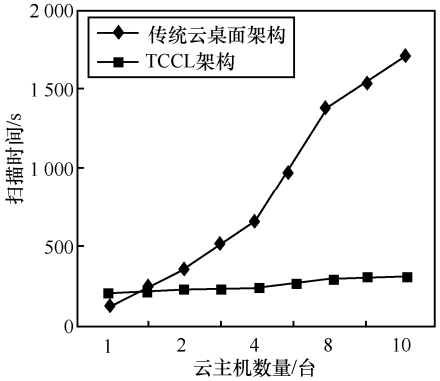

图 9　传统云桌面和 TCCL 下扫描时间对比

从图 9 中可以看出，在传统的云桌面架构下，由于每个云主机都要读取其自身完整的磁盘镜像文件，每个云主机的扫描时间基本保持一致，随着云主机数量的增长，总的扫描时间曲线呈一条延伸线过原点的快速增长的直线。而在 TCCL 架构下，扫描第一台云主机时所花费的时间稍高一点，原因是读写控制模块需要根据位图定位该用户保持不变的云主机实例文件和 App 增量卷文件以及对应的用户增量卷文件。随着云主机数量的增长，总的扫描时间呈一条缓慢增长的直线。这是由于云主机共享的部分文件在扫描第一台云主机时已完成，之后扫描脚本只需扫描用户位图数据所标记修改过的部分数据，大幅度降低了冗余的磁盘文件扫描，大幅度减小了扫描云主机文件中恶意代码等安全性操作所需的服务器磁盘 I/O 开销。

6　结束语

本文提出了一种安全增强、存储高效的云桌面架构，将透明计算整合到云桌面系统中，把云主机系统文件和用户增量文件与云主机服务器在物理上分离。在原型系统之上的实验，验证了 TCCL 架构只需用较低的读写性能损耗即可换取云桌面系统更高的安全性、更强的可拓展性和更高效的存储性能。

参考文献：

[1] DEBOOSERE L, VANKEIRSBILCK B, SIMOENS P, et al. Cloud-based desktop services for thin clients[J]. IEEE Internet Computing, 2012, 16(6): 60-67.

[2] ZISSIS D, LEKKAS D. Addressing cloud computing security issues[J]. Future Generation Computer Systems, 2012, 28(3):583-592.

[3] YU S, WANG C, REN K, et al. Achieving secure, scalable, and fine-grained data access control in cloud computing[C]// Conference on Information Communications. 2010:1-9.

[4] DESHPANDE P, SHARMA S C, KUMAR P S. Security threats in cloud computing[C]// International Conference on Computing, Communication & Automation. 2015.

[5] 林闯, 苏文博, 孟坤,等. 云计算安全:架构、机制与模型评价[J]. 计算机学报, 2013, 36(9):1765-1784.
LIN C, SU W B, MENG K, et al. Cloud computing security: architecture, mechanism and modeling[J]. Chinese Journal of Computers, 2013, 36(9):1765-1784.

[6] DASILVA D A, LIU L, BESSIS N, et al. Enabling green it through building a virtual desktop infrastructure[C]// 2012 Eighth International Conference on Semantics, Knowledge and Grids (SKG). 2012: 32-38.

[7] DAWOUD W, TAKOUNA I, MEINEL C. Infrastructure as a service security: challenges and solutions[C]// 2010 the 7th International Conference on Informatics and Systems (INFOS). 2010:1-8.

[8] JAISWAL P R, ROHANKAR A W. Infrastructure as a service: security issues in cloud computing[J]. International Journal of Computer Science and Mobile Computing, 2014, 3(3): 707-711.

[9] JAISWAL P R, ROHANKAR A W. Infrastructure as a service: security issues in cloud computing[J]. International Journal of Computer Science and Mobile Computing, 2014, 3(3): 707-711.

[10] 项国富, 金海, 邹德清,等. 基于虚拟化的安全监控[J]. 软件学报, 2012, 23(8):2173-2187.
XIANG G F, JIN H, ZHOU D Q, et al. Virtualization-based security monitoring[J]. Journal of Software, 2012, 23(8):2173-2187.

[11] HUSSEIN N H, KHALID A. A survey of cloud computing security

challenges and solutions[J]. International Journal of Computer Science and Information Security, 2016, 14(1): 52.

[12] 王晓聪, 张冉,黄赪东. 渗透测试技术浅析[J]. 计算机科学, 2012, (S1): 86-88.
WANG X C, ZHANG R, HUANG C D. Penetration test techniques shallow[J]. Computer Science, 2012,(S1):86-88.

[13] ZHANG Y, YANG L T, ZHOU Y, et al. Information security underlying transparent computing: impacts, visions and challenges[J]. Web Intelligence & Agent Systems, 2010, 8(2):203-217.

[14] WANG G, LIU Q, XIANG Y, et al. Security from the transparent computing aspect[C]//Networking and Communications International Conference on Computing. 2014:216-220.

[15] AYRES J, FLANNICK J, GEHRKE J, et al. Sequential pattern mining using a bitmap representation[C]// Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2002:429-435.

作者简介：

王斌（1973-），男，山西大同人，博士，中南大学教授，主要研究方向为透明计算、软件工程。

李伟民（1988-），男，湖南邵阳人，中南大学博士生，主要研究方向为云计算和透明计算。

盛津芳（1971-），女，湖南长沙人，中南大学副教授，主要研究方向为透明计算和大数据。

肖斯诺（1993-），男，湖南娄底人，中南大学硕士生，主要研究方向为云计算、透明计算。