(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国 际 局

(43) 国际公布日
2020 年 9 月 3 日 (03.09.2020)

WIPO | PCT

(10) 国际公布号
WO 2020/173083 A1

(51) 国际专利分类号:
G06F 9/54 (2006.01) *G06F 21/53* (2013.01)
G06F 12/10 (2016.01) *G06F 9/455* (2006.01)

(21) 国际申请号: PCT/CN2019/106833

(22) 国际申请日: 2019 年 9 月 20 日 (20.09.2019)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
201910151836.4 2019年2月28日 (28.02.2019) CN

(71) 申请人: 上海交通大学 (SHANGHAI JIAO TONG UNIVERSITY) [CN/CN]; 中国上海市闵行区东川路800号, Shanghai 200240 (CN)。

(72) 发明人: 陈海波 (CHEN, Haibo); 中国上海市闵行区东川路800号, Shanghai 200240 (CN)。 糜泽羽 (MI, Zeyu); 中国上海市闵行区东川路800号, Shanghai 200240 (CN)。 臧斌宇 (ZANG, Binyu); 中国上海市闵行区东川路800号, Shanghai 200240 (CN)。 管海兵 (GUAN, Haibing); 中国上海市闵行区东川路800号, Shanghai 200240 (CN)。

(74) 代理人: 上海汉声知识产权代理有限公司 (SHANGHAI HANGSOME INTELLECTUAL PROPERTY LTD.); 中国上海市闵行区银都路3828弄56号307室, Shanghai 201108 (CN)。

(81) 指定国 (除另有指明，要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。

(84) 指定国 (除另有指明，要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ,

(54) **Title:** MICROKERNEL INTERPROCESS COMMUNICATION METHOD AND SYSTEM

(54) **发明名称:** 微内核进程间通讯方法和系统

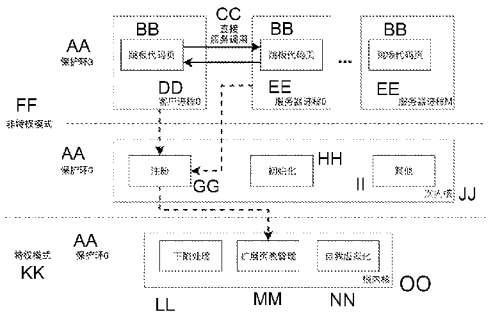

图 1

AA Protection ring
BB Trampoline code page
CC Direct service calling
DD Client process
EE Server process
FF Non-privileged mode
GG Registration
HH Initialization
II Other
JJ Sub-kernel
KK Privileged mode
LL Drop processing
MM Extended page table management
NN Self-virtualization
OO Root-kernel

(57) **Abstract:** Provided are a microkernel interprocess communication method and system. Th microkernel interprocess communication method comprises: activating a virtual environment by means of hardware, configuring a microkernel to be a sub-kernel, and configuring a root-kernel under the sub-kernel, the root-kernel being capable of interacting with the virtual environment; configuring extended page tables corresponding to different processes, the processes being divided into a client process and a server process, and filling a page-table-based address of the client process in the extended page table of the server process; providing a user-mode process-oriented interface, the interface enabling switching between processes in a user-mode address space; and scanning code pages in the processes, and substituting for illegal code jump instructions. The present invention employs hardware virtualization techniques, and uses an extended page table to control content in a page table corresponding to a user-mode process, thereby realizing switching between processes without having to modify page-table-based addresses thereof. The invention significantly enhances communication performance between microkernel processes without requiring any modification of existing hardware architectures.

WO 2020/173083 A1

[见续页]

NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

— 包括国际检索报告(条约第21条(3))。

(57) 摘要: 本发明提供了一种微内核进程间通讯方法和系统，借助硬件启动虚拟化环境，将微内核构造成次内核，所述次内核之下构造根内核，所述根内核能够与虚拟化环境交互；为不同的进程构造对应的扩展页表，所述进程分成客户进程和服务器进程，将客户进程的页表基地址填入服务器进程的扩展页表中；提供面向用户态进程的接口，所述接口能够在用户态的地址空间中进行进程间切换；对进程中的代码页进行扫描，并替换非法的代码跳转指令。本发明利用硬件虚拟化技术，利用扩展页表控制用户态进程的页表内容，实现无需修改进程页表基地址的进程间切换，在对现有硬件架构无需做任何修改的情况下，大幅提升微内核中进程间通信的性能。

微内核进程间通讯方法和系统

技术领域

本发明涉及操作系统微内核技术领域，具体地，涉及一种微内核进程间通讯方法和系统，尤其是涉及一种高效且安全的微内核进程间通讯方法和系统。

背景技术

微内核自上世纪提出以来，经过了 30 年的研究与发展，其关键的设计是将操作系统内核只提供最基础的机制，并将大部分操作系统内核的其他功能从内核态移到用户态服务器进程中。这意味着一个服务器进程内发生的错误不会影响到其他服务器进程，更不会影响到微内核服务器微内核。因此，这样的设计能够增强微内核的鲁棒性。同时，将大部分功能移除内核态，可以有效减小可信计算基（TCB），使得系统更不容易被攻击，也更容易被形式化验证。基于上述优势，微内核被广泛应用于高度依赖于安全与可靠性的领域，例如航空、车载系统、医疗设备。

在一个微内核中，任何两个进程之间的通讯都依赖于进程间通讯机制，但是目前已知该机制是运行时开销的重要来源。一次进程间通讯首先需要调用系统调用陷入微内核，之后微内核查找到目标进程，再将消息拷贝到目标进程，同时还需要两次进程地址空间切换（如果要防御最近的熔断 Meltdown 攻击），最后回到用户态。如果要回到原进程，这样的过程还需要完整的重复一遍。

研究人员一直在探索更加高效的优化方案，以便减少微内核进程间通讯的开销。目前已知软件和硬件两类优化方案。

目前已知性能最佳的微内核进程间通讯方法由 seL4 操作系统实现，该方案完全通过软件实现，目标将所有不必要的操作从进程间通讯的路径中去除。seL4 为 Call 和 ReplyWait 系统调用使用快速通道的技术方法，该方法会直接将消息传送给目标进程，同时不需要调度。所有的传输的数据存放在寄存器中，也消除了数据拷贝的开销。但是 seL4 存在缺陷，首先，快速通道技术依然需要下陷进入内核，因此它的性能开销也较大；其次，快速通道路径只适用于部分系统调用（Call 和 ReplyWait），同时只能传输少量数据，当使用其它系统调用或者传输数据超出一定限制，只能使用性能开销更大的慢速

通道技术，也就是传统的微内核进程间通讯方法。第三，当通讯的两个进程运行在不同的处理器上，一次进程间通讯需要使用跨处理器中断（Inter-Processor Interrupt），该中断会极大的影响进程间通讯的性能。

基于硬件的修改方案可以大大提高微内核进程间通讯的开销，dIPC 项目通过修改硬件的方式将所有进程间通讯参与方放置在同一个虚拟地址空间，之后的进程间通讯完全由硬件实现，允许一个进程直接调用另一个进程的函数，而无需操作系统内核的帮助。进程间的隔离依赖于 dIPC 实现的标签内存，该内存也需要通过修改硬件的方式实现。但是，使用 dIPC 需要对软件做出较大修改，以便利用 dIPC 提出的接口。同时也需要对操作系统内核做出较大修改，以便适应新的进程间通讯方式。修改硬件的技术手段距离真正被大规模使用依然需要长时间的时间检验，同时该方法与直接使用成熟商用硬件的方法相比，较难得到部署。因此该方法很难在短时间内得到接受。

与本申请相关的现有技术是专利文献 CN103425538A，公开了一种进程通讯方法，根据进程通讯请求分配内存空间；将通讯数据存入所述内存空间；将所述内存空间的逻辑地址写入消息队列；通过从所述消息队列中读取的逻辑地址访问所述通讯数据。上述进程通讯方法及系统，在接收到进程通讯请求时分配用于存储通讯数据的内存空间，这将不需要预先划分通讯数据的存储空间，将内存空间的逻辑地址写入消息队列中，通过在消息队列中逻辑地址的读取进行通讯数据的访问，以在消息队列的作用下使得数据访问进程通过对消息队列中逻辑地址的逐一读取有序访问通讯数据，在多个进程之间的通讯过程中不需要进行通讯数据的复制，大大地提高了数据共享的灵活性。

发明内容

针对现有技术中的缺陷，本发明的目的是提供一种微内核进程间通讯方法和系统。

根据本发明提供的一种微内核进程间通讯方法，包括：

轻量级虚拟化步骤：借助硬件启动虚拟化环境，将微内核构造成次内核，所述次内核之下构造根内核，所述根内核能够与虚拟化环境交互；

扩展页表步骤：为不同的进程构造对应的扩展页表，所述进程分成客户进程和服务器进程，将客户进程的页表基地址填入服务器进程的扩展页表中；

快速通讯步骤：提供面向用户态进程的接口，所述接口能够在用户态的地址空间中进行进程间切换；

二进制修改步骤：对进程中的代码页进行扫描，并替换非法的代码跳转指令。

优选地，所述轻量级虚拟化步骤包括：

虚拟层下陷消除步骤：对虚拟层下陷进行处理；

根内核初始化步骤：在次内核启动后对根内核进行初始化，动态将次内核移动到非特权模式下，根内核提供用于管理扩展页表的接口；

进程识别步骤：为进程分配一个内存页用于记录进程的身份信息，记为身份信息页，并将所述身份信息页映射到所述进程扩展页表中以及次内核的虚拟地址空间中，使得次内核能够通过虚拟地址访问当前下陷进程的身份信息页以识别下陷进程的身份信息。所述身份信息页在不同进程扩展页表中拥有相同的客户物理地址。

优选地，所述对虚拟层下陷进行处理中，对于特权指令造成的下陷，根内核通过配置 VMCS 域令特权指令的执行不造成虚拟层下陷；对于硬件事件造成的下陷，根内核允许硬件向非特权模式下的次内核插入外部中断；对于访问扩展页表违规造成的下陷，根内核使用大容量页表将物理内存地址映射给次内核。

优选地，所述扩展页表步骤中包括：

初始化步骤：初始化时对服务器进程注册到次内核，当客户进程进行注册时，次内核通知根内核为客户进程、服务器进程分别复制扩展页表，添加扩展页表映射；

切换进程步骤：客户进程访问跳板代码页进行进程间切换，跳板代码页调用 VMFUNC 指令将扩展页表指针从客户进程的扩展页表改成指向服务器进程的扩展页表，无需修改 CR3 寄存器值。

优选地，所述快速通讯步骤，对于小数据量传输，所述小数据量通过 CPU 寄存器传输，对于大数据量传输，分配共享缓冲区，将共享缓冲区的地址映射到客户进程及服务器进程的扩展页表中。

优选地，所述二进制修改步骤中，对于植入的单条 VMFUNC 指令，将所述单条 VMFUNC 指令替换为三条空指令；对于由相邻指令拼凑的非法指令，令相邻指令之间插入一个空指令；对于存在于一条长指令之中的非法指令，令所述长指令替换成多个等价的指令后再处理。

优选地，所述次内核是运行在非特权模式下的微内核；所述根内核运行在特权模式下的，包括下陷处理单元、扩展表管理单元、自我虚拟化单元；所述下陷处理单元处理次内核引起的下陷，包括访问扩展页表违规和使用特权指令下陷；所述扩展表管理单元动态管理次内核、及次内核内进程的扩展页表；所述自我虚拟化单元

在系统启动时动态将次内核降级为非特权模式并初始化其 VMCS 和扩展页表。

根据本发明提供的一种微内核进程间通讯系统，包括：

轻量级虚拟化模块：借助硬件启动虚拟化环境，将微内核构造成次内核，所述次内核之下构造根内核，所述根内核能够与虚拟化环境交互；

扩展页表模块：为不同的进程构造对应的扩展页表，所述进程分成客户进程和服务器进程，将客户进程的页表基地址填入服务器进程的扩展页表中；

快速通讯模块：提供面向用户态进程的接口，所述接口能够在用户态的地址空间中进行进程间切换；

二进制修改模块：对进程中的代码页进行扫描，并替换非法的代码跳转指令。

与现有技术相比，本发明具有如下的有益效果：

1、对于存在已久的微内核进程间通信性能较差问题，本发明巧妙地利用硬件虚拟化技术，在对现有硬件架构无需做任何修改的情况下，大幅提升微内核中进程间通信的性能。

2、本发明可以应用在各种不同设计的微内核上，并且仅需对微内核的代码进行较小的修改即可带来大幅的性能提升。

3、本发明并未改变原本微内核的强隔离性，仍然对于熔断 Meltdown 等攻击具备防御能力。

附图说明

通过阅读参照以下附图对非限制性实施例所作的详细描述，本发明的其它特征、目的和优点将会变得更明显：

图 1 为本发明通讯装置的实施例示意图；

图 2 为利用扩展页表控制客户页表内容；

图 3 为进程启动注册流程；

图 4 为用户态进程间切换流程；

图 5 为动态二进制修改流程。

具体实施方式

下面结合具体实施例对本发明进行详细说明。以下实施例将有助于本领域的技术人员进一步理解本发明，但不以任何形式限制本发明。应当指出的是，对本领域的普通技

术人员来说，在不脱离本发明构思的前提下，还可以做出若干变化和改进。这些都属于本发明的保护范围。

本发明所提出的方法能够在微内核进程间通讯时，在无需微内核的介入的情况下，允许一个进程直接切换到另一个进程的虚拟地址空间。微内核的介入是进程间通讯开销大的重要原因，如果将微内核从进程间通讯的路径中去除，将极大提高进程间通讯的性能。因此，本发明将微内核从进程间通讯中去除。在无需微内核介入的情况下，保证一个进程不会恶意利用进程间通讯访问其他进程的数据和执行其他进程的代码。传统进程间通讯方法需要由微内核负责检查通讯的合法性，阻止任何可能的攻击。现有技术可以保证在没有操作系统内核介入情况下的进程间通讯安全，但是却需要修改硬件，本发明需要使用成熟的商用硬件来保证安全性。本发明尽可能小的修改应用程序与微内核源码，进程间通讯方法是微内核中的核心机制，对它的修改常常意味着需要对应用程序与微内核源码做出大量修改，这样会带来较大的部署难度。因此，本发明尽可能小的修改应用程序与微内核源码。

根据本发明提供的一种微内核进程间通讯方法，包括：

轻量级虚拟化步骤：借助硬件启动虚拟化环境，将微内核构造成次内核，所述次内核之下构造根内核，所述根内核能够与虚拟化环境交互；

扩展页表步骤：为不同的进程构造对应的扩展页表，所述进程分成客户进程和服务器进程，将客户进程的页表基地址填入服务器进程的扩展页表中；

快速通讯步骤：提供面向用户态进程的接口，所述接口能够在用户态的地址空间中进行进程间切换；

二进制修改步骤：对进程中的代码页进行扫描，并替换非法的代码跳转指令。

具体地，所述轻量级虚拟化步骤包括：

虚拟层下陷消除步骤：对虚拟层下陷进行处理；

根内核初始化步骤：在次内核启动后对根内核进行初始化，动态将次内核移动到非特权模式下，根内核提供用于管理扩展页表的接口；

进程识别步骤：为进程分配一个内存页用于记录进程的身份信息，记为身份信息页，并将所述身份信息页映射到所述进程扩展页表中，以及次内核的地址空间中，使得次内核能够通过虚拟地址访问当前下陷进程的身份信息页以识别下陷进程的身份信息。所述身份信息页在不同进程扩展页表中拥有相同的客户物理地址。

具体地，所述对虚拟层下陷进行处理中，对于特权指令造成的下陷，根内核通

过配置 VMCS 域令特权指令的执行不造成虚拟层下陷；对于硬件事件造成的下陷，根内核允许硬件向非特权模式下的次内核插入外部中断；对于访问扩展页表违规造成的下陷，根内核使用大容量页表将物理内存地址映射给次内核，所述页表的容量采用 1GB 的内存页大小。

具体地，所述扩展页表步骤中包括：

初始化步骤：初始化时对服务器进程注册到次内核，当客户进程进行注册时，次内核通知根内核为客户进程、服务器进程分别复制扩展页表，添加扩展页表映射；

切换进程步骤：客户进程访问跳板代码页进行进程间切换，跳板代码页调用 VMFUNC 指令将扩展页表指针从客户进程的扩展页表改成指向服务器进程的扩展页表，无需修改 CR3 寄存器值。

具体地，所述快速通讯步骤，对于小数据量传输，所述小数据量通过 CPU 寄存器传输，对于大数据量传输，分配共享缓冲区，将共享缓冲区的地址映射到客户进程及服务器进程的扩展页表中。

具体地，所述二进制修改步骤中，对于植入的单条 VMFUNC 指令，将所述单条 VMFUNC 指令替换为三条空指令；对于由相邻指令拼凑的非法指令，令相邻指令之间插入一个空指令；对于存在于一条长指令之中的非法指令，令所述长指令替换成多个等价的指令后再处理。

具体地，所述次内核是运行在非特权模式下的微内核；所述根内核运行在特权模式下的，包括下陷处理单元、扩展表管理单元、自我虚拟化单元；所述下陷处理单元处理次内核引起的下陷，包括访问扩展页表违规和使用特权指令下陷；所述扩展表管理单元动态管理次内核、及次内核内进程的扩展页表；所述自我虚拟化单元在系统启动时动态将次内核降级为非特权模式并初始化其 VMCS 和扩展页表。

根据本发明提供的一种微内核进程间通讯系统，包括：

轻量级虚拟化模块：借助硬件启动虚拟化环境，将微内核构造成次内核，所述次内核之下构造根内核，所述根内核能够与虚拟化环境交互；

扩展页表模块：为不同的进程构造对应的扩展页表，所述进程分成客户进程和服务器进程，将客户进程的页表基地址填入服务器进程的扩展页表中；

快速通讯模块：提供面向用户态进程的接口，所述接口能够在用户态的地址空间中进行进程间切换；

二进制修改模块：对进程中的代码页进行扫描，并替换非法的代码跳转指令。

关于本发明中涉及的术语作如下解释，微内核是一种内核架构，由数量最小化的功能组成，这些功能负责实现一个操作系统依赖的最基础的机制，包括物理地址空间管理，进程管理，进程间通讯（IPC）。微内核进程是运行在微内核之上的应用程序，是微内核进行资源分配和资源调度的基本单位。进程间通讯（IPC）是至少两个进程或线程间传送数据或信号的一些技术或方法。本发明专注于微内核的进程间通讯。客户进程/服务器进程是在微内核上，每一个客户端进程的实例都可以向一个服务器进程发出请求，服务器进程负责提供各种功能性的服务。虚拟地址空间是 CPU 在寻址的时候，是按照虚拟地址来寻址，然后通过内存管理单元(MMU)将虚拟地址转换为物理地址。虚拟机监视器能够在一台物理机器上虚拟出多台客户虚拟机，每台客户虚拟机拥有与真实机器一样的功能。为了提升虚拟地址到物理地址的翻译速度，处理器利用 TLB 缓存部分页表中储存的地址映射。当需要翻译某一虚拟地址时，将首先查询 TLB，如果 TLB 中无对应映射，才访问储存于内存中的页表，完成地址翻译。客户虚拟地址（GVA）/ 客户物理地址（GPA）/ 主机物理地址（HPA）是在虚拟化环境中，客户虚拟机中的程序使用客户虚拟地址访问内存，客户虚拟机的物理内存为客户物理地址。客户虚拟机的内核通过控制客户页表，从而控制客户虚拟地址到客户物理地址的转换。主机物理地址代表物理机器的真实内存，虚拟机监视器通过扩展页表控制客户物理地址到主机物理地址的转换。CR3 控制寄存器：用于控制和确定处理器的操作模式以及当前执行任务的特性，CR3 中含有页目录表物理内存基地址，因此该寄存器也被称为页目录基地址寄存器 PDBR（Page-Directory Base address Register）。VMCS 数据域是一个物理 CPU 通过 VMCS 数据域能够获得每个虚拟 CPU 的各种信息。

本发明利用硬件虚拟化技术，允许一个进程在无需陷入微内核的情况下，直接切换到另一个进程的虚拟地址空间，并调用目标函数。具体来说，本发明依然允许不同进程拥有不同的虚拟地址空间，这样的设计与已有的微内核一致，可以减少对现有系统的修改。同时，本发明通过引入一个极小的虚拟机监控器，为不同的进程构造不同的扩展页表（控制客户物理地址到主机物理地址的映射），并利用硬件的 VMFUNC 指令进行扩展页表的切换，以达到在用户态切换虚拟地址空间的效果。为此，对于两个客户进程（发送者进程）和服务器进程（接收者进程），本发明通过配置接收者进程的扩展页表，将发送者进程的页表基地址（CR3 寄存器的值）映射到接收者进程页表基地址（CR3 寄存器的值）对应的主机物理地址。因此，发送者进程在利用硬件的 VMFUNC 指令进行扩展页

表的切换后，其CR3寄存器的值会直接指向接收者进程的页表。同时，本发明在接收者进程的虚拟地址空间内为其每个线程分别提供了一个栈。另外，为了支持长进程间通信，本发明提供了共享缓冲区用以在传输大量的信息，这些共享缓冲区和接收者进程中的每个线程一一绑定从而保证了本发明在高并发环境下的良好性能。

在虚拟机监控器为进程构造扩展页面的过程中，Intel硬件虚拟化技术允许为每个用户态进程配置一份扩展页表（现阶段最多可支持512份扩展页表）。为了便于对不同扩展页表进行翻译、切换等操作，这些扩展页表的基地址（即指向扩展页表的指针）需要存储在扩展页表指针列表中，该列表的内存分配是在根内核初始化时完成的。在微内核初始化阶段，微内核直接运行在物理机上，通过微内核的页表来直接管理虚拟地址到主机物理地址的映射。在初始化本发明阶段，需要增加一层主扩展页表，微内核原有页表负责客户虚拟地址到客户物理地址的映射，主扩展页表负责客户物理地址到主机物理地址的映射。主扩展页表的基地址默认存储在扩展页表指针列表的第一个位置（即偏移量为0）。

后续每启动一个用户态进程，都会先对主扩展页表进行拷贝，为了节省内存开销采取写时复制（Copy-on-write）机制，仅当后续进程对扩展页表项有修改时，新分配内存并建立新的映射。在客户进程向服务器进程注册自己时，会向服务器进程的扩展页表添加客户进程页表基地址到服务器进程页表基地址对应主机物理地址的映射。

构造扩展页表是每一个用户态进程启动时，在每个用户态进程的启动过程中，虚拟机监控器会直接使用主扩展页表。只有当该进程向服务器注册时，才需要为每一个新注册的服务器进程拷贝一份主扩展页表，同时在这个新的扩展页表中，将该注册进程的页表基地址的客户物理地址指向服务器进程的页表基地址的主机物理地址（采用写时复制机制）。然后将拷贝生成的新扩展页表的第一级内存页的基地址填入到扩展页表指针列表的对应偏移项中，偏移量取决于当前用户进程的进程标识符。

如图 1 所示，本发明总体架构包括四个模块：轻量级虚拟化模块，扩展页表管理模块，进程间快速通信模块以及动态二进制修改模块。

轻量级虚拟化模块负责借助硬件启动一个支持虚拟化技术的环境，将原本的微内核作为次内核，并在其下插入一个根内核用于处理其余模块与虚拟化环境相关的交互。该模块通过仔细的配置以在保证功能正确性的前提下，尽可能的减少虚拟化环境相较于原有环境所带来的性能损失。

具体地，为了使用VMFUNC指令，进程需要运行在虚拟化环境中的非特权模式下。

轻量级虚拟化模块首先将原本的环境转变为拥有特权模式和非特权模式的虚拟化环境，并将各种进程置于非特权模式中。对于原有微内核的虚拟化设计则需考量是否需要将其置于非特权模式下。满足上述需求的已知技术主要分为两大类：1）模拟成熟的虚拟机技术，将原有进程和内核视为一个虚拟机系统，同时运行在非特权模式下。2）将内核置于特权模式下，而保持进程运行在非特权模式下。然而现有技术具有以下缺点：第一类技术可以利用现有的商用虚拟机管理程序（例如 KVM 和 Xen），但是会因为虚拟化层造成较大的性能损失。第二类技术会在进程和内核交互（例如系统调用）时产生大量的虚拟层下陷，而一次虚拟层下陷的开销要比非虚拟化环境下一次系统调用的开销要昂贵的多。

轻量级虚拟化模块提供了一种新的解决方案，相比现有技术，既避免了传统的虚拟化方案带来的性能损失，又消除了大量的虚拟层下陷造成的额外开销：微内核仍然被置于非特权模式下作为次内核，而在特权模式引入了一个轻量级的只包含了必要功能的小型管理程序根内核，仅仅提供了扩展页表管理功能，动态自我虚拟化模块以及一些基础的虚拟层下陷处理逻辑。轻量级虚拟化模块采用以下三种方式，例如：

（1）不必要虚拟层下陷的消除：为了消除昂贵的虚拟层下陷开销，根内核通过仔细地设置 VMCS 域使得大部分的虚拟机行为不会触发任何的虚拟层下陷。总体上看，虚拟层下陷一共可以分为三大类：特权指令造成的下陷、硬件事件造成的下陷以及扩展页表项违规造成的下陷。1）对于执行特权指令造成的下陷（比如更改 CR3 寄存器的值，停机（HLT）指令等），根内核可以配置 VMCS 域令这些特权指令的执行不造成任何虚拟层下陷；2）对于硬件事件造成的下陷（比如外部中断等），传统的虚拟机管理程序会配置硬件在接收到此类事件时触发一个虚拟层下陷，本发明中的根内核也扮演同样的角色以允许硬件向非特权模式下的次内核插入外部中断；3）对于访问扩展页表项违规造成的虚拟层下陷，为了尽可能的减少二级地址翻译造成的性能损失，本发明使根内核使用最大的大型页（在 x86-64 架构下大小为 1GB）来映射大部分的物理内存地址给次内核，既能减少 TLB 未命中后处理逻辑的内存访问次数，又能减少 TLB 未命中的次数。除上述三类下陷之外，根内核仍然保留有部分用于管理的虚拟层下陷处理逻辑，比如 VMCALL 指令会无条件的触发虚拟层下陷而根内核利用这条指令实现了一个和上层次内核通信的接口。

（2）根内核的初始化：根内核的启动方式与传统的随物理机器一同初始化的虚

拟化管理程序不同，为了避免启动过程中执行大量且易错的初始化代码，根内核选择在次内核启动后初始化并动态地将次内核移动到非特权模式下。为了使非特权模式下的次内核能够方便地管理每个进程的扩展页表，根内核通过 CPUID 为上层暴露了一个用于管理扩展页表的接口。

（3）进程误识别问题：当一个发送者进程正在一个接收者进程的虚拟地址空间中执行时，如果此时发送者进程接收到一个中断而导致其下陷到次内核中，则它会试图以接收者进程的身份去调用次内核提供的功能。然而此时的次内核仍然会将调用内核功能的进程识别为原本的发送者进程，这就是所谓的进程误识别问题。为了解决这个问题，本发明为每个进程分配了一个内存页用于记录每个进程的身份信息并将这个页映射到每个进程扩展页表中相同的客户物理地址。同时本发明通过将每个进程的身份信息页映射到次内核的地址空间中使得次内核可以通过一个虚拟地址访问当前下陷进程的身份信息页以正确地确定下陷进程的身份。

扩展页表管理模块负责为不同的进程构造对应的扩展页表，并配合进程间快速通信模块将有关的映射和数据结构填入被调用进程的扩展页表中的正确位置。

具体地，本发明需要同时满足两个需求：1）保证不同进程间的虚拟地址空间的隔离性 2）为这些进程提供一套有效的用户态虚拟地址空间切换的方法。为满足上述需求，现有的技术方案可分为两类：1）将不同的进程放入同一个虚拟地址空间，但是为每一个进程单独分配一个用不同的扩展页表以在相同的虚拟地址空间中提供隔离性，同时也利用VMFUNC指令来绕过内核直接在用户态执行虚拟地址空间的切换。2）利用英特尔提出的硬件特性PKU来切换不同进程在虚拟地址空间中的不同视角。但是现有技术具有以下缺点：第一类技术在进程数量较少时拥有易于实现的优点，但是当进程数量增多时，为了避免不同进程分配到的虚拟地址区域产生冲突，就需要非常仔细地管理虚拟地址空间的划分，导致一系列繁杂的工作并且提高了配置出错的可能性。第二类技术同样无法解决潜在的虚拟地址区域冲突问题。此外该硬件特性仅提供了有限数量的安全域，显然无法满足微内核场景下的需求。扩展页表管理模块采用扩展页表的映射管理进行实现。

针对现有技术的缺陷，扩展页表管理模块提出了一套新的解决方案，对切换前后扩展页表进行映射管理，不需要经过大量的修改即可既保留传统的虚拟内存隔离性又能够快速地在虚拟地址空间之间进行切换。对于不同的进程仍然保留它们各自的页表，将客户进程的CR3 寄存器的值到服务器进程的CR3 寄存器值对应的

主机物理地址的映射添加到服务器进程的扩展页表中，这样就能够使得用户态的进程在利用VMFUNC指令切换扩展页表时无需修改CR3 寄存器中的值，可以直接进行后续的虚拟地址翻译。

如图 2 所示，在虚拟地址空间切换流程中，客户进程和服务器进程拥有它们各自的页表，页表基地址的值分别为客户进程CR3 值和服务器进程CR3 值。在初始化过程中，服务器进程会首先将自己的进程信息（比如CR3 的值等）注册到次内核。当客户进程进行注册时，次内核会通知底层的根内核为两个进程分别复制两份新的扩展页表并建立合适的映射。在执行进程间切换的过程中，主机的CR3 寄存器中的值会保持客户进程CR3 值不变。当客户进程调用相应的接口后，跳板代码会调用VMFUNC指令将扩展页表指针从客户进程扩展页表改指向服务器进程扩展页表，客户进程可以直接访问当前服务器进程虚拟地址空间中的任意虚拟地址。

进程间快速通信模块负责提供一套面向用户态进程的接口，用于快速有效地在用户态地址空间中进行进程间切换。

具体地，进程间通信模块负责在每个进程向次内核注册自己时将跳板代码页映射到该进程的虚拟地址空间中，从而为用户态的进程提供用于快速进程间切换的一套接口。每个客户进程都需要将所有需要调用的服务器进程填入跳板代码页中。当将一个客户进程绑定到一个服务器进程上时，次内核会根据服务器进程注册自己时设定的最大可支持的并行线程数量来分配对应数量的栈，并将这些栈映射到服务器进程的虚拟地址空间中。通常情况下，发送者进程需要通过进程间通信将一些数据传输至接受者进程，本模块根据不同的待传输数据大小提供了两种方式：1）对于数据量较小的传输，本模块按照x86-64 架构下的调用约定将要传输的数据放入CPU的寄存器中。2）对于数据量较大的传输，本模块为每一对客户进程和服务器进程分配了一块共享缓冲区并将缓冲区的地址映射到了两个进程的页表中。

动态二进制修改模块负责扫描每个进程的所有代码页并替换非法的 VMUNFC 指令，由此防止用户恶意利用 VMFUNC 指令非法地跳转至任意代码执行地址。

具体地，动态二进制修改模块能够保证系统安全性，在用户态进程的代码页中，可能会出现因为偶然或恶意等因素拼凑出的非法VMFUNC指令，这些非法的VMFUNC指令的存在有概率被攻击者利用而跳转到非法的代码区域执行，所以本发明引入了动态二进制修改模块来消除这些非法的VMFUNC指令。

当一个进程注册自己时，次内核会调用本模块扫描这个进程的所有代码页，如果

在指定的跳板代码页之外发现了非法的VMFUNC指令则会用功能性等价的一些指令替换这条非法的VMFUNC指令。在代码页被动态修改后，原本的一条指令会变为两条甚至更多条的等价指令，这样原本代码页的空间就容不下这些等价指令了。因此，本模块会将原本的指令位置空间中的内容替换为一条跳转指令用以跳转到另一个用于放置等价指令的代码页。存放等价指令的代码页由次内核负责插入到一个未被使用的虚拟地址处。

其中，本发明采用非法指令替换策略，导致代码页中存在非法VMFUNC指令的因素有几种不同的可能，本模块将其归类为三种情况分别进行处理：1）非法的VMFUNC指令确实是一条被故意植入的VMFUNC指令，这种情况下本模块会将这条非法的VMFUNC指令替换为三条空（NOP）指令（不做任何工作的指令）。2）非法的VMFUNC指令由两条或多条相邻的指令拼凑出来，这种情况下本模块会在这些相邻的指令之间插入一个空指令来打破这个拼凑出的VMFUNC指令。3）非法的VMFUNC指令存在于一条较长的指令中包含了VMFUNC指令的编码，这种情况下本模块会将这条指令替换为其它几条等价的指令从而消除非法指令。

具体的实施操作流程可以参考图 3、图 4、图 5，进程启动注册流程中，如图 2 所示，包括：

步骤 1）当一个用户态进程启动时，首先判断自身进程是否会作为一个服务器进程为其他客户进程提供服务。如果自身是一个服务器进程，则跳转至步骤 2，否则跳转至步骤 4。

步骤 2）作为一个服务器进程需要将自己的CR3 寄存器值等进程信息注册并保存到次内核中。

步骤 3）判断自身进程是否会作为一个客户进程去调用其他服务器进程提供的服务。是则继续步骤 4，否则启动注册流程完成。

步骤 4）作为一个客户进程需要注册通知次内核该进程需要调用的服务器进程信息。

步骤 5）次内核接着会通知根内核为当前客户进程和对应的服务器进程分别复制两个客户进程扩展页表和服务器进程扩展页表。

步骤 6）根内核将从客户进程CR3 值映射到服务器进程CR3 值对应主机物理地址的映射添加到服务器进程的扩展页表中，启动注册流程完成。

通过跳板代码页实现客户进程与服务器进程之间的通讯，首先明确跳板代码页是

一个内存页，其中包含了精简的代码逻辑，用于在不同的扩展页表中切换并正确调用函数功能。所谓跳板，即当进程运行到该代码页的首地址时，将开始执行扩展页表的切换等操作，功能上体现为执行流从客户进程跳转到了服务器进程。

在客户进程与服务器进程的初始化注册过程中，跳板代码页的映射已经被分别插入到二者的页表当中，当客户进程想要与服务器进程通信时：

1)首先客户进程设定好待传输的数据参数并调用相应的接口，开始执行跳板代码页中的代码，

2）跳板代码页中的代码将保存客户进程当前的寄存器等状态信息并配置相应的栈以供后续执行，

3）随后跳板代码页使用VMFUNC指令切换到服务器进程的扩展页表，由于前文所述的配置，此时可以正常调用服务器进程中的函数并获得返回值，

4）跳板代码页使用VMFUNC指令切换回原来的客户进程的扩展页表，恢复寄存器等状态，客户进程最后成功获得来自服务器进程的返回值。

扩展页地址映射是虚拟地址到物理地址的映射关系，通常用于根据一个虚拟地址查询对应的物理地址。在扩展页表中表现为：给定一个虚拟地址，将虚拟地址切分为多个部分（当前为 4 个），每一部分作为扩展页表对应层级的偏移量，逐步翻译到扩展页表的最下层，最终获取到存储在最下层扩展页表项中的内容，该内容即给定虚拟地址对应的物理地址。添加扩展页表的映射，本质是根据给定的虚拟地址，在对应的最下层扩展页表项中，填入目标物理地址。

扩展页地址映射是在客户进程向服务器进程注册自己的过程中：

1）次内核会调用根内核提供的接口，让根内核在扩展页表的层面将客户进程与服务器进程绑定，

2）根内核将通过查询扩展页表的方式，以客户进程页表基地址（CR3 寄存器值）的客户物理地址为索引，在服务器进程的扩展页表中逐层翻译，最终在扩展页表的最下层找到对应的扩展页表项，

3）根内核将服务器进程中的该扩展页表项内容填写为服务器进程页表基地址所对应的主机物理地址。

由此，当客户进程的页表和服务器进程的扩展页表搭配使用时，客户进程的页表基地址最终会翻译为服务器进程的页表所对应的主机物理地址，无需更改CR3 寄存器的值。

如图 3 所示，用户态进程间切换流程中包括：

步骤 1）客户进程设定好目标服务器进程参数，调用本发明提供的用户态进程间切换的接口。

步骤 2）跳板代码收到目标服务器进程参数，并检查待传输的数据大小是否超出寄存器所能容纳的大小。是则跳转至步骤 3，否则跳转至步骤 4。

步骤 3）跳板代码将要传输的数据从客户进程的内部缓冲区复制到与目标服务器进程的共享缓冲区中。

步骤 4）利用VMFUNC指令将扩展页表指针从客户进程扩展页表改指向服务器进程扩展页表。

步骤 5）跳板代码配置好服务器进程执行过程中会用到的栈，然后调用服务器进程提前注册好的功能函数使其开始执行。

如图 4 所示，动态二进制修改流程中包括：

步骤 1）一个进程启动时捕获该进程拥有的所有代码页。

步骤 2）次内核扫描所有的代码页并识别出所有的非法VMFUNC指令。若不存在非法VMFUNC指令则流程结束，若存在则继续步骤 3。

步骤 3）判别每个识别出的非法VMFUNC指令的产生原因，对于每个非法VMFUNC指令在特定的未被使用的虚拟地址空间分配一块用于存放等价替换指令的内存页。

步骤 4）对于每个非法VMFUNC指令采取上文中提到的对应替换策略，将替换后的等价指令放入分配好的内存页中。

步骤 5）在每个放置等价指令的内存页末尾添加一条跳转指令，跳转目标地址为被替换指令地址的下一行指令地址。

步骤 6）将原本非法VMFUNC指令的位置中替换为一条跳转指令，跳转目标地址设为对应内存页的起始地址。动态二进制修改流程结束。

本发明在安全性方面能够防御攻击，对于恶意切换扩展页表攻击，利用本发明提供的动态二进制修改技术，动态地在每个进程启动过程中消除所有非法的VMFUNC指令，能够有效地防御来自用户态恶意利用VMFUNC指令进行非法跳转执行的攻击；对于熔断Meltdown攻击及其变体，本发明并未对微内核原有的页表隔离机制做出修改，由于微内核自身的设计天然将用户态进程和微内核的页表分开隔离，即使用户态进程被攻陷也无法读取内核态的关键数据；对于拒绝服务攻击，本发明提供了一套超时机制，用于在服务器进程长时间未响应时强制将执行流返回给客户进程。因此即便攻击者通过

恶意请求使服务器进程阻塞也不会造成其他进程的执行阻塞；对于恶意服务进程调用攻击，本发明提供了一个记录客户进程调用服务器进程权限的表，用于在每次客户进程调用其他服务器进程之前检查其是否拥有调用的合法权限。因此恶意进程无法任意调用没有权限的服务器进程。

本发明利用硬件虚拟化技术加速微内核的进程间通信，利用扩展页表控制客户物理地址到主机物理地址的映射，从而实现进程间切换前后无需修改进程页表基地址的值，利用新型硬件指令，无需下陷即可在用户态进程之间完成扩展页表切换，实现快速的微内核进程间切换；采用轻量化的虚拟化环境配置方法，利用底层插入的小型管理程序，在仅仅针对原有微内核添加一行代码的情况下，将虚拟化技术应用于现有微内核，配置VMCS域以消除绝大部分由于引入虚拟化层带来的性能损失；利用动态的二进制修改保护方法，运用成熟的二进制修改技术，在新建进程时完全消除掉进程中带有的非法指令，确保恶意的进程间切换无法发生，无需修改或重新编译进程程序源代码即可实现消除恶意指令。

本发明提出轻量化且高效的虚拟化系统，利用扩展页表控制用户态进程的页表内容，从而实现无需修改进程页表基地址的进程间切换，通过虚拟化硬件特性，实现针对微内核应用在用户态的进程间快速切换，结合现有成熟的二进制修改技术，在对进程源代码不做任何修改的情况下，完全消除特定的恶意指令。

与传统依靠软件优化进行微内核中进程间通信优化的方法不同，本发明提出了一套利用硬件虚拟化技术加速进程间通信性能的方法。在既保留了微内核原有特性的情况下，借助硬件大幅提升了进程间通信的性能，并仅仅对原有微内核代码进行了较小的修改，既保证了安全性又提升了整体性能。

同时，本发明提出的动态二进制修改保护技术，也能够被使用在各类需要在不修改程序源代码的情况下替换特定指令的系统之中。

本领域技术人员知道，除了以纯计算机可读程序代码方式实现本发明提供的系统、装置及其各个模块以外，完全可以通过将方法步骤进行逻辑编程来使得本发明提供的系统、装置及其各个模块以逻辑门、开关、专用集成电路、可编程逻辑控制器以及嵌入式微控制器等的形式来实现相同程序。所以，本发明提供的系统、装置及其各个模块可以被认为是一种硬件部件，而对其内包括的用于实现各种程序的模块也可以视为硬件部件内的结构；也可以将用于实现各种功能的模块视为既可以是实现方法的软件程序又可以是硬件部件内的结构。

以上对本发明的具体实施例进行了描述。需要理解的是，本发明并不局限于上述特定实施方式，本领域技术人员可以在权利要求的范围内做出各种变化或修改，这并不影响本发明的实质内容。在不冲突的情况下，本申请的实施例和实施例中的特征可以任意相互组合。

权利要求书

1、一种微内核进程间通讯方法，其特征在于，包括：

轻量级虚拟化步骤：借助硬件启动虚拟化环境，将微内核构造成次内核，所述次内核之下构造根内核，所述根内核能够与虚拟化环境交互；

扩展页表步骤：为不同的进程构造对应的扩展页表，所述进程分成客户进程和服务器进程，将客户进程的页表基地址填入服务器进程的扩展页表中；

快速通讯步骤：提供面向用户态进程的接口，所述接口能够在用户态的地址空间中进行进程间切换；

二进制修改步骤：对进程中的代码页进行扫描，并替换非法的代码跳转指令。

2、根据权利要求 1 所述的微内核进程间通讯方法，其特征在于，所述轻量级虚拟化步骤包括：

虚拟层下陷消除步骤：对虚拟层下陷进行处理；

根内核初始化步骤：在次内核启动后对根内核进行初始化，动态将次内核移动到非特权模式下，根内核提供用于管理扩展页表的接口；

进程识别步骤：为进程分配一个内存页用于记录进程的身份信息，记为身份信息页，并将所述身份信息页映射到所述进程扩展页表中，以及次内核的虚拟地址空间中，使得次内核能够通过虚拟地址访问当前下陷进程的身份信息页以识别下陷进程的身份信息。

3、根据权利要求 2 所述的微内核进程间通讯方法，其特征在于，所述对虚拟层下陷进行处理中，对于特权指令造成的下陷，根内核通过配置 VMCS 域令特权指令的执行不造成虚拟层下陷；对于硬件事件造成的下陷，根内核允许硬件向非特权模式下的次内核插入外部中断；对于访问扩展页表违规造成的下陷，根内核使用页表将物理内存地址映射给次内核。

4、根据权利要求 1 所述的微内核进程间通讯方法，其特征在于，所述扩展页表步骤中包括：

初始化步骤：初始化时将服务器进程注册到次内核，当客户进程进行注册时，次内核通知根内核为客户进程、服务器进程分别复制扩展页表，添加扩展页地址映射；

切换进程步骤：客户进程访问跳板代码页进行进程间切换，跳板代码页调用

VMFUNC 指令将扩展页表指针从客户进程的扩展页表改成指向服务器进程的扩展页表，无需修改 CR3 寄存器值。

5、根据权利要求 1 所述的微内核进程间通讯方法，其特征在于，所述快速通讯步骤，对于小数据量传输，所述小数据量通过 CPU 寄存器传输，对于大数据量传输，分配共享缓冲区，将共享缓冲区的地址映射到客户进程及服务器进程的扩展页表中。

6、根据权利要求 1 所述的微内核进程间通讯方法，其特征在于，所述二进制修改步骤中，对于植入的单条 VMFUNC 指令，将所述单条 VMFUNC 指令替换为三条空指令；对于由相邻指令拼凑的非法指令，令相邻指令之间插入一个空指令；对于存在于一条长指令之中的非法指令，令所述长指令替换成多个等价的指令后再处理。

7、根据权利要求 1 所述的微内核进程间通讯方法，其特征在于，所述次内核是运行在非特权模式下的微内核；

所述根内核运行在特权模式下的，包括下陷处理单元、扩展表管理单元、自我虚拟化单元；

所述下陷处理单元处理次内核引起的下陷，包括访问扩展页表违规和使用特权指令下陷；

所述扩展表管理单元动态管理次内核、及次内核内进程的扩展页表；

所述自我虚拟化单元在系统启动时动态将次内核降级为非特权模式并初始化其 VMCS 和扩展页表。

8、根据权利要求 4 所述的微内核进程间通讯方法，其特征在于，所述跳板代码页是内存页，其中包含的代码逻辑能够在不同的扩展页表中切换，并进行函数调用。

9、根据权利要求 4 所述的微内核进程间通讯方法，其特征在于，所述扩展页地址映射是在客户进程注册过程中，通过次内核调用根内核提供的接口，对客户进程和服务器进程绑定，根内核通过查询扩展页表，能够将客户进程的页表基地址对应到服务器进程的主机物理地址。

10、一种微内核进程间通讯系统，其特征在于，包括：

轻量级虚拟化模块：借助硬件启动虚拟化环境，将微内核构造成次内核，所述次内核之下构造根内核，所述根内核能够与虚拟化环境交互；

扩展页表模块：为不同的进程构造对应的扩展页表，所述进程分成客户进程和服务器进程，将客户进程的页表基地址填入服务器进程的扩展页表中；

快速通讯模块：提供面向用户态进程的接口，所述接口能够在用户态的地址空

间中进行进程间切换；

二进制修改模块：对进程中的代码页进行扫描，并替换非法的代码跳转指令。

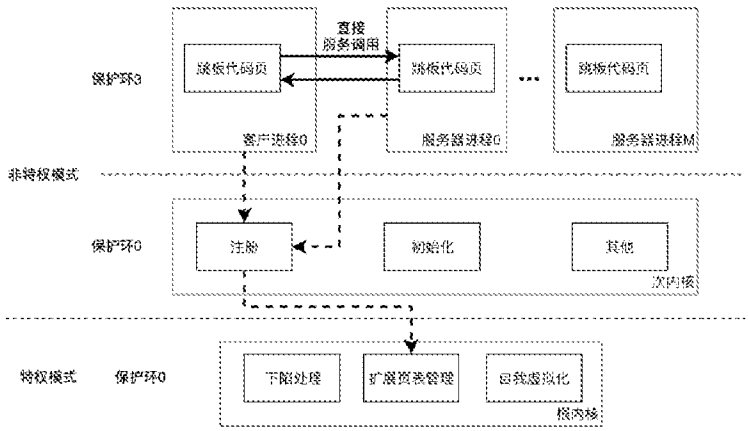
直接
服务调用
跳板代码页
跳板代码页
跳板代码页
保护环3
客户进程0
服务器进程0
服务器进程M
非特权模式
保护环0
注册
初始化
其他
次内核
特权模式
保护环0
下陷处理
扩展页表管理
自我虚拟化
根内核

图 1

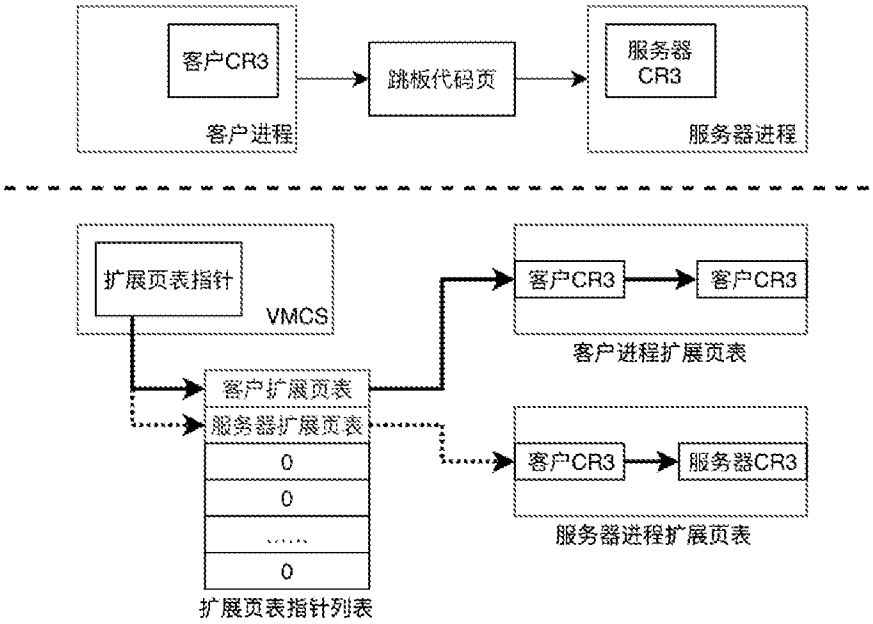
客户CR3
跳板代码页
服务器CR3
客户进程
服务器进程
扩展页表指针
VMCS
客户扩展页表
服务器扩展页表
0
0
.....
0
扩展页表指针列表
客户CR3
客户CR3
客户进程扩展页表
客户CR3
服务器CR3
服务器进程扩展页表

图 2

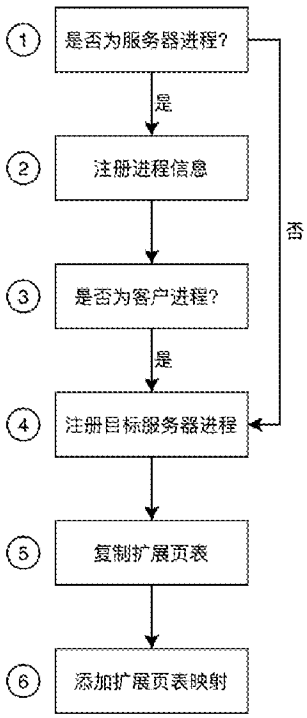
1
是否为服务器进程?
是
否
2
注册进程信息
3
是否为客户进程?
是
4
注册目标服务器进程
5
复制扩展页表
6
添加扩展页表映射

图 3

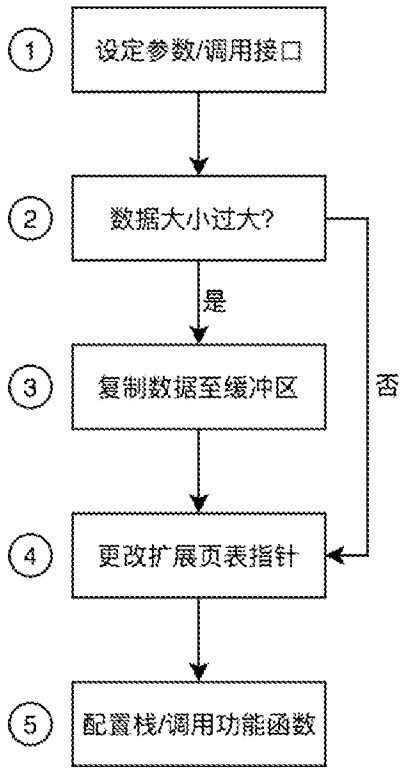
1
设定参数/调用接口
2
数据大小过大?
是
否
3
复制数据至缓冲区
4
更改扩展页表指针
5
配置栈/调用功能函数

图 4

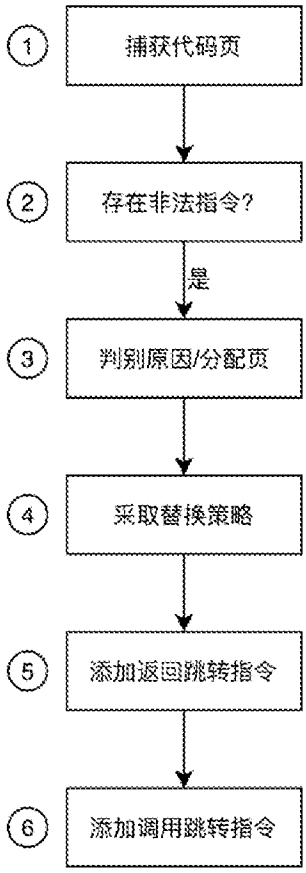
1
捕获代码页
2
存在非法指令?
是
3
判别原因/分配页
4
采取替换策略
5
添加返回跳转指令
6
添加调用跳转指令

图 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2019/106833

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/54(2006.01)i; G06F 12/10(2016.01)i; G06F 21/53(2013.01)i; G06F 9/455(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNABS, CNTXT, CNKI, VEN, USTXT, WOTXT, EPTXT: 进程, 线程, 通信, 通讯, 数据, 客户, 服务, 读, 写, 目标, 虚拟机, 虚拟机监视器, 扩展页表, 虚拟地址, 物理地址, 寄存器, 共享, 微内核, IPC, interpro+ communication?, client, service, read, write, virtual machine, VMM, EPT, extend page table, virtual address, physical address, register, share, micro kernel

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
PX	CN 109933441 A (SHANGHAI JIAO TONG UNIVERSITY) 25 June 2019 (2019-06-25) claims 1-10	1-10
A	CN 107368379 A (CENTRAL SOUTH UNIVERSITY) 21 November 2017 (2017-11-21) entire document	1-10
A	CN 103425538 A (SHENZHEN TENCENT COMPUTER SYSTEMS CO., LTD.) 04 December 2013 (2013-12-04) entire document	1-10
A	CN 104572313 A (HUAWEI TECHNOLOGIES CO., LTD.) 29 April 2015 (2015-04-29) entire document	1-10
A	CN 107667350 A (INTEL CORPORATION) 06 February 2018 (2018-02-06) entire document	1-10
A	US 2005246453 A1 (MICROSOFT CORPORATION) 03 November 2005 (2005-11-03) entire document	1-10

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
06 December 2019	**30 December 2019**

Name and mailing address of the ISA/CN	Authorized officer
China National Intellectual Property Administration (ISA/CN) **No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088** **China**	
Facsimile No. **(86-10)62019451**	Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2019/106833

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	109933441	A	25 June 2019	None			
CN	107368379	A	21 November 2017	None			
CN	103425538	A	04 December 2013	CN	103425538	B	11 May 2016
CN	104572313	A	29 April 2015	CN	104572313	B	13 March 2018
CN	107667350	A	06 February 2018	US	2016364341	A1	15 December 2016
				EP	3308317	A1	18 April 2018
				US	9842065	B2	12 December 2017
				EP	3308317	A4	20 February 2019
				WO	2016204892	A1	22 December 2016
				US	2018113817	A1	26 April 2018
US	2005246453	A1	03 November 2005	KR	20060047639	A	18 May 2006
				EP	1630670	A3	26 December 2007
				EP	1630670	A2	01 March 2006
				CN	1700171	A	23 November 2005
				JP	2005322242	A	17 November 2005

国际检索报告

国际申请号 PCT/CN2019/106833

A. 主题的分类

G06F 9/54(2006.01)i; G06F 12/10(2016.01)i; G06F 21/53(2013.01)i; G06F 9/455(2006.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

G06F

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称，和使用的检索词（如使用）)

CNABS, CNTXT, CNKI, VEN, USTXT, WOTXT, EPTXT:进程，线程，通信，通讯，数据，客户，服务，读，写，目标，虚拟机，虚拟机监视器，扩展页表，虚拟地址，物理地址，寄存器，共享，微内核，IPC, interpro+ communication?, client, service, read, write, virtual machine, VMM, EPT, extend page table, virtual address, physical address, register, share, micro kernel

C. 相关文件

类 型*	引用文件，必要时，指明相关段落	相关的权利要求
PX	CN 109933441 A (上海交通大学) 2019年 6月 25日 (2019-06-25) 权利要求1-10	1-10
A	CN 107368379 A (中南大学) 2017年 11月 21日 (2017-11-21) 全文	1-10
A	CN 103425538 A (深圳市腾讯计算机系统有限公司) 2013年 12月 4日 (2013-12-04) 全文	1-10
A	CN 104572313 A (华为技术有限公司) 2015年 4月 29日 (2015-04-29) 全文	1-10
A	CN 107667350 A (英特尔公司) 2018年 2月 6日 (2018-02-06) 全文	1-10
A	US 2005246453 A1 (MICROSOFT CORPORATION) 2005年 11月 3日 (2005-11-03) 全文	1-10

☐ 其余文件在C栏的续页中列出。

☑ 见同族专利附件。

* 引用文件的具体类型：

"A" 认为不特别相关的表示了现有技术一般状态的文件

"E" 在国际申请日的当天或之后公布的在先申请或专利

"L" 可能对优先权要求构成怀疑的文件，或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件（如具体说明的）

"O" 涉及口头公开、使用、展览或其他方式公开的文件

"P" 公布日先于国际申请日但迟于所要求的优先权日的文件

"T" 在申请日或优先权日之后公布，与申请不相抵触，但为了理解发明之理论或原理的在后文件

"X" 特别相关的文件，单独考虑该文件，认定要求保护的发明不是新颖的或不具有创造性

"Y" 特别相关的文件，当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时，要求保护的发明不具有创造性

"&" 同族专利的文件

国际检索实际完成的日期	国际检索报告邮寄日期
2019年 12月 6日	2019年 12月 30日
ISA/CN的名称和邮寄地址 中国国家知识产权局(ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088 传真号 (86-10)62019451	受权官员 孟文婷 电话号码 86-(010)-62089383

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2019/106833

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	109933441	A	2019年 6月 25日	无			
CN	107368379	A	2017年 11月 21日	无			
CN	103425538	A	2013年 12月 4日	CN	103425538	B	2016年 5月 11日
CN	104572313	A	2015年 4月 29日	CN	104572313	B	2018年 3月 13日
CN	107667350	A	2018年 2月 6日	US	2016364341	A1	2016年 12月 15日
				EP	3308317	A1	2018年 4月 18日
				US	9842065	B2	2017年 12月 12日
				EP	3308317	A4	2019年 2月 20日
				WO	2016204892	A1	2016年 12月 22日
				US	2018113817	A1	2018年 4月 26日
US	2005246453	A1	2005年 11月 3日	KR	20060047639	A	2006年 5月 18日
				EP	1630670	A3	2007年 12月 26日
				EP	1630670	A2	2006年 3月 1日
				CN	1700171	A	2005年 11月 23日
				JP	2005322242	A	2005年 11月 17日