

增强型包过滤器的内核态程序的超级调用方法及装置

本申请要求于 2022 年 03 月 14 日提交中国专利局、申请号为 202210249016.0、
申请名称为“增强型包过滤器的内核态程序的超级调用方法及装置”的中国专利申请
5 的优先权，其全部内容通过引用结合在本申请中。

技术领域

本说明书实施例涉及计算机技术领域，尤其涉及一种增强型包过滤器的内核态程
序的超级调用方法及装置。

10 背景技术

增强型包过滤器（extended Berkeley Packet Filter，eBPF）是一种通用执行引
擎。由于增强型包过滤器可以在操作系统内核（如 Linux 内核）中运行特殊程序（如
沙盒程序）而无需更改内核源代码或加载内核模块，因而受到开发人员的普遍使用。

增强型包过滤器的架构至少可以包括用户态程序（或称用户空间程序）和内核态
15 程序（或称内核程序）两部分。其中，用户态程序用于加载指令码（如 BPF 字节码）
至内核，也可以读取内核回传的信息或事件；而内核态程序用于将指令码转化为内核
可执行指令，进而由内核中的指令程序执行。

在虚拟化的环境中（如虚拟机）提供有超级调用（Hypercall）机制，通过调用
Hypercall 可以实现从用户态到内核态的状态切换，从而获取内核态下更高的操作权
20 限。

但是在虚拟化的环境中运行的增强型包过滤器，如果向要获取更高的操作权限，
则需要从内核态程序切换到虚拟机中底层的虚拟机监控器（Hypervisor，或 virtual
machine monitor，VMM）；而现有增强型包过滤器的内核态程序无法调用 Hypercall，
也就无法切换到 Hypervisor 以获取更高的操作权限。

25 发明内容

本说明书实施例提供的一种增强型包过滤器的内核态程序的超级调用方法及装置：
根据本说明书实施例的第一方面，提供一种增强型包过滤器的内核态程序的超级
调用方法，应用于虚拟机监控器，所述方法包括：
响应于内核态程序触发的虚拟化缺页，获取触发所述虚拟化缺页的内存地址；

判断所述虚拟化缺页的内存地址是否和映射表中的缺页键对应的内存地址一致；其中，所述映射表记录了键值对之间的映射关系，所述缺页键为所述内核态程序和所述虚拟机监控器共同在所述映射表中确定的键，所述映射表中的每个键唯一对应一个内存地址；

如果所述虚拟化缺页的内存地址和缺页键对应的内存地址一致，从所述映射表中读取所述缺页键映射的缺页值；其中，所述缺页值包括所述内核态程序在触发所述虚拟化缺页之前写入的超级调用的功能类型；

基于所述缺页值指示的功能类型执行该功能类型的超级调用。

可选的，所述响应于内核态程序的触发的虚拟化缺页，包括：

响应于内核态程序将超级调用的功能类型写入映射表中的缺页键映射的缺页值后触发的虚拟化缺页。

可选的，不同缺页键用于表示不同的参数个数；所述方法还包括：

基于所述缺页键表示的参数个数，从所述映射表中的第一个参数键开始依次读取所述参数个数的参数键，以获取每个参数键映射的参数值；其中，所述参数值为所述内核态程序写入的超级调用的参数，1个参数值对应1个参数；

所述基于所述缺页键指示的功能类型执行该功能类型的超级调用，包括：

基于所述虚拟机监控器所在的虚拟机的身份信息，获取所述虚拟机中的增强型包过滤器维护的映射表；

将所述映射表中的协商键映射的协商值修改为缺页键；以使所述映射表维护的所述协商键映射的协商值指示所述缺页键；其中，所述协商键为所述内核态程序和所述虚拟机监控器共同在所述映射表中确定的键。

根据本说明书实施例的第二方面，提供一种增强型包过滤器的内核态程序的超级调用方法，应用于内核态程序，所述方法包括：

响应于超级调用指令，读取增强型包过滤器维护的映射表中的协商键映射的协商值；其中，所述映射表记录了键值对之间的映射关系，所述协商值包括虚拟机监控器预先写入的用于指示缺页键的值；所述协商键和缺页键为所述内核态程序和所述虚拟机监控器共同在所述映射表中确定的键；

基于所述协商值查询所述映射表中映射的缺页键；

将所述超级调用指令中指定的超级调用的功能类型写入所述缺页键映射的缺页值，从而触发虚拟化缺页；以使所述虚拟机监控器响应于所述虚拟化缺页，基于所述缺页值指示的功能类型执行该功能类型的超级调用。

可选的，所述方法还包括：

在所述超级调用指令中还指定有超级调用的至少1个参数时，从所述映射表中的第一个参数键开始、将所述至少1个参数依次写入到至少1个参数键映射的参数值，以使1个参数值对应1个参数。

根据本说明书实施例的第三方面，提供一种增强型包过滤器的内核态程序的超级

调用装置，应用于虚拟机监控器，所述装置包括：

响应单元，响应于内核态程序触发的虚拟化缺页，获取触发所述虚拟化缺页的内存地址；

判断单元，判断所述虚拟化缺页的内存地址是否和映射表中的缺页键对应的内存地址一致；其中，所述映射表记录了键值对之间的映射关系，所述缺页键为所述内核态程序和所述虚拟机监控器共同在所述映射表中确定的键，所述映射表中的每个键唯一对应一个内存地址；

读取单元，如果所述虚拟化缺页的内存地址和缺页键对应的内存地址一致，从所述映射表中读取所述缺页键映射的缺页值；其中，所述缺页值包括所述内核态程序在触发所述虚拟化缺页之前写入的超级调用的功能类型；

调用单元，基于所述缺页值指示的功能类型执行该功能类型的超级调用。

可选的，所述响应单元，进一步用于响应于内核态程序将超级调用的功能类型写入映射表中的缺页键映射的缺页值后触发的虚拟化缺页。

可选的，不同缺页键用于表示不同的参数个数；

所述读取单元，还用于基于所述缺页键表示的参数个数，从所述映射表中的第一个参数键开始依次读取所述参数个数的参数键，以获取每个参数键映射的参数值；其中，所述参数值为所述内核态程序写入的超级调用的参数，1个参数值对应1个参数；

所述调用单元，进一步用于基于所述参数值指示的参数和所述缺页值指示的功能类型执行该功能类型的超级调用。

可选的，所述装置还包括：

写入单元，基于所述虚拟机监控器所在的虚拟机的身份信息，获取所述虚拟机中的增强型包过滤器维护的映射表；将所述映射表中的协商键映射的协商值修改为缺页键；以使所述映射表维护的所述协商键映射的协商值指示所述缺页键；其中，所述协商键为所述内核态程序和所述虚拟机监控器共同在所述映射表中确定的键。

根据本说明书实施例的第四方面，提供一种增强型包过滤器的内核态程序的超级调用装置，应用于内核态程序，所述装置包括：

响应单元，响应于超级调用指令，读取增强型包过滤器维护的映射表中的协商键映射的协商值；其中，所述映射表记录了键值对之间的映射关系，所述协商值包括虚拟机监控器预先写入的用于指示缺页键的值；所述协商键和缺页键为所述内核态程序和所述虚拟机监控器共同在所述映射表中确定的键；

查询单元，基于所述协商值查询所述映射表中映射的缺页键；

调用单元，将所述超级调用指令中指定的超级调用的功能类型写入所述缺页键映射的缺页值，从而触发虚拟化缺页；以使所述虚拟机监控器响应于所述虚拟化缺页，基于所述缺页值指示的功能类型执行该功能类型的超级调用。

可选的，所述调用单元，还用于在所述超级调用指令中还指定有超级调用的至少1个

参数时，从所述映射表中的第一个参数键开始、将所述至少 1 个参数依次写入到至少 1 个参数键映射的参数值，以使 1 个参数值对应 1 个参数。

根据本说明书实施例的第五方面，提供一种电子设备，包括：

处理器；

5 用于存储处理器可执行指令的存储器；

其中，所述处理器被配置为上述任一项增强型包过滤器的内核态程序的超级调用方法。

根据本说明书实施例的第六方面，提供一种计算机可读存储介质，当所述计算机可读存储介质中的指令由电子设备的处理器执行时，使得所述电子设备能够执行上述任
10 一项增强型包过滤器的内核态程序的超级调用方法。

本说明书实施例，提供了一种增强型包过滤器的内核态程序的超级调用方案，基于虚拟化缺页技术为增强型包过滤器的内核态程序提供超级调用功能；在不修改内核源代码的前提下，通过调用该提供的超级调用功能以实现间接的超级调用。

附图说明

15 图 1 是本说明书一实施例提供的增强型包过滤器的内核态程序的超级调用方法的流程图；

图 2 是本说明书一实施例提供的 Map 的示意图；

图 3 是本说明书一实施例提供的以虚拟机监控器为执行主体的方法实施例的示意图；

图 4 是本说明书一实施例提供的以内核态程序为执行主体的方法实施例的示意图；

20 图 5 是本说明书一实施例提供的增强型包过滤器的内核态程序的超级调用装置的硬件结构图；

图 6 是本说明书一实施例提供的增强型包过滤器的内核态程序的超级调用装置的模块；

25 图 7 是本说明书一实施例提供的增强型包过滤器的内核态程序的超级调用装置的模块。

具体实施方式

这里将详细地对示例性实施例进行说明，其示例表示在附图中。下面的描述涉及附图时，除非另有表示，不同附图中的相同数字表示相同或相似的要素。以下示例性实施例中所述的实施方式并不代表与本说明书相一致的所有实施方式。相反，它们
30 仅是与如所附权利要求书中所详述的、本说明书的一些方面相一致的装置和方法的例子。

在本说明书使用的术语是仅仅出于描述特定实施例的目的，而非旨在限制本说明书。在本说明书和所附权利要求书中所使用的单数形式的“一种”、“所述”和“该”

也旨在包括多数形式，除非上下文清楚地表示其他含义。还应当理解，本文中使用的术语“和/或”是指并包含一个或多个相关联的列出项目的任何或所有可能组合。

应当理解，尽管在本说明书可能采用术语第一、第二、第三等来描述各种信息，但这些信息不应限于这些术语。这些术语仅用来将同一类型的信息彼此区分开。例如，在不脱离本说明书范围的情况下，第一信息也可以被称为第二信息，类似地，第二信息也可以被称为第一信息。取决于语境，如在此所使用的词语“如果”可以被解释成为“在……时”或“当……时”或“响应于确定”。

如前所述，增强型包过滤器（以下简称为 eBPF）的内核态程序无法直接调用超级调用（以下简称为 Hypercall），也就无法从内核态程序切换到虚拟机监控器（以下简称为 Hypervisor）以获取更高的操作权限。

在相关技术中，通过修改内核源代码，可以实现 eBPF 的内核态程序的 Hypercall。具体地，现有内核定义有一系列的 Helper 辅助函数，但其中并没有提供实现 Hypercall 功能的 Helper 辅助函数；为此通过修改内核源代码以添加通过 Helper 辅助函数调用 Hypercall 的实现。然而，这种方式需要修改内核源代码，其适用性较差无法实现大量普及。

本说明书提供了一种增强型包过滤器的内核态程序的超级调用方案，在无需修改内核源代码的情况下，也可以实现 eBPF 内核态程序的 Hypercall 功能。

以下结合图 1 介绍本说明书提供的增强型包过滤器的内核态程序的超级调用方法实施例的流程图，所述方法包括：

步骤 1.2：内核态程序响应于超级调用指令，读取增强型包过滤器维护的映射表中的协商键映射的协商值；其中，所述映射表记录了键值对之间的映射关系，所述协商值包括虚拟机监控器预先写入的用于指示缺页键的值；所述协商键和缺页键为所述内核态程序和所述虚拟机监控器共同在所述映射表中确定的键。

步骤 1.4：所述内核态程序基于所述协商值查询所述映射表中映射的缺页键。

步骤 1.6：所述内核态程序将所述超级调用指令中指定的超级调用的功能类型写入所述缺页键映射的缺页值，从而触发虚拟化缺页。

步骤 2.2：所述虚拟机监控器响应于内核态程序触发的虚拟化缺页，获取触发所述虚拟化缺页的内存地址。

步骤 2.4：所述虚拟机监控器判断所述虚拟化缺页的内存地址是否和映射表中的缺页键对应的内存地址一致；其中，所述映射表中的每个键唯一对应一个内存地址。

步骤 2.6：如果所述虚拟化缺页的内存地址和缺页键对应的内存地址一致，所述虚拟机监控器从所述映射表中读取所述缺页键映射的缺页值。

步骤 2.8：所述虚拟机监控器基于所述缺页值指示的功能类型执行该功能类型的超级调用。

在本说明书中，通过虚拟化技术可以在内核中生成至少 1 个虚拟机，并可以在每

个虚拟机可以运行 eBPF（增强型包过滤器），所述 eBPF 中会维护一个或多个映射表。这样对于虚拟机来说就可以包括 eBPF 的内核态程序、映射表以及虚拟机本身具有的 Hypervisor（虚拟机监控器）。

上述映射表可以理解作为一种记录键值对（即 key-value）之间的映射关系的数据结构，
 5 如 Map。Map 是一种通用的 key-value 存储结构，可以用于存储任意类型的数据。映射表中每个 key 均对应有一个内存地址。

在 eBPF 中，映射表一般用于为用户态程序和内核态程序提供双向通信，以实现用户态程序和内核态程序之间的数据交互。

在本说明书中，通过对上述映射表进行配置或改进还可以用于为 eBPF 和 Hypervisor
 10 提供双向通信，以实现 eBPF 的内核态程序与 Hypervisor 之间的数据交互。具体地对上述映射表进行配置或改进将在后续实施例中进行说明。

在以 Map 这类映射表为例，对于不同场景的需求 Map 还衍生了不同类型的 Map 类型，例如 Array Map（数组表），Hash Map（哈希表）等等；不同的 Map 类型具有不同的特点和功能。

15 以 Array Map 为例，由于 Array Map 具有查询效率高以及占用内存少的特点；因此当采用 Array Map 作为 eBPF Map 时，可以提高 Map 的读写效率。对于本说明书中的实施例来说，提高 Map 的读写效率则意味着提高内核态程序的超级调用时的响应效率。

以下进一步介绍对映射表进行配置或改进，以实现 eBPF 的内核态程序与 Hypervisor 之间的数据交互。

20 在本说明书中，需要在映射表中注册一个或多个的协商键（以下称为协商 key）。每个协商 key 对应的协商值（以下称为协商 value）可以由 Hypervisor 预先写入的缺页键（以下称为缺页 key），即协商 value 可以用于指示缺页 key。协商 key 和缺页 key 可以是内核态程序和 Hypervisor 预先共同在映射表中确定的 key。

需要说明的是，在同一时刻下一个缺页 key 只能存在于一个协商 value，从而避免不同协商 value 同时指向一个缺页 key，导致异常冲突的发生。

在一示例性的实施例中，所述 Hypervisor 写入协商 value 之前需要先获取到映射表，该获取方式可以包括：

30 Hypervisor 可以基于所述虚拟机监控器所在的虚拟机的身份信息，获取所述虚拟机中的增强型包过滤器维护的映射表；进一步，将所述映射表中的协商键映射的协商值修改为缺页键；以使所述映射表维护的所述协商键映射的协商值指示所述缺页键。

在实现时，Hypervisor 可以使用内核分析工具（例如 Crash Utility）获取来获取上述映射表。一般的，内核分析工具可以是基于虚拟机的身份信息作为唯一标识进行定位，并从定位到的虚拟机中获取映射表。其中，所述身份信息可以包括但不限于：虚拟机的内核版本（如 Linux Kernel 版本）、符号表、关键数据结构等信息。

35 需要说明的是，在有的实施例中 Hypervisor 也可以不借助内核分析工具而直接基于

虚拟机的身份信息获取映射表。

在 Hypervisor 将缺页 key 写入协商 value 后，该协商 value 就可以指向映射表中的缺页 key。需要说明的是，所述缺页 key 的缺页 value 最初可以为空，后续该缺页 value 的 value 值需要由内核态程序写入（这一过程将在后面实施例中进行说明）。

- 5 以下结合图 2 所示的 Map 的示意图，假设第一个 key 为协商 key，对应的协商 value 为 X，那么在 Map 中该协商 value 对应的缺页 key 即为 GPA0 处的 key=X。

当 Hypervisor 写入了协商 value 后，Hypervisor 还需要针对缺页 key 构建虚拟化缺页，这样当 eBPF 内核态程序后续在缺页 key 对应的缺页 value 写入任何数据时均可以触发虚拟化缺页。

- 10 上述针对缺页 key 构建虚拟化缺页的方式可以是，将缺页 key 的虚拟地址从页表中去除。所述页表是管理虚拟地址与物理地址的映射关系的表。此外，Hypervisor 在将缺页 key 的虚拟地址从页表中去除后，为了保持虚拟化缺页，需要拦截添加针对缺页 key 的虚拟地址的映射。

- 15 当 Hypervisor 写入了协商 value 后，对于 eBPF 的内核态程序来说就可以使用 Hypercall 功能，以间接实现 Hypercall 的调用。

在实现时，eBPF 的内核态程序可以接收例如来自用户态程序发起的指令，如果该指令需要由 Hypervisor 执行，则内核态程序需要使用 Hypercall。以下将需要使用 Hypercall 的指令称为 Hypercall 指令。

- 20 内核态程序响应于 Hypercall 指令，首先需要读取映射表中的协商 key 对应的协商 value。

然后，以所述协商 value 为索引，查询所述映射表中的所述协商 value 指向的缺页 key。

最后，将所述 Hypercall 指令中指定的 Hypercall 的功能类型作为 value 值写入所述缺页 key 对应的缺页 value 中，从而触发虚拟化缺页。

- 25 相应地，Hypervisor 可以响应于内核态程序触发的虚拟化缺页，获取触发所述虚拟化缺页的内存地址，并判断所述虚拟化缺页的内存地址是否和缺页 key 对应的内存地址一致；

- 30 如果所述虚拟化缺页的内存地址和缺页 key 对应的内存地址一致，则 Hypervisor 可以确定 eBPF 内核态程序的确发起了 Hypercall 调用，进而可以以所述协商 value 为索引，查询所述映射表中的所述协商 value 指向的缺页 key。

由于，此时缺页 key 对应的缺页 value 中已经被内核态程序写入了表示 Hypercall 的功能类型的 value 值；因此，通过读取缺页 key 对应的缺页 value，就可以确定当前需要执行哪种 Hypercall 功能类型的 Hypercall，进而完成 Hypercall 的执行。

- 35 应用上述实施例，eBPF 的内核态程序并非直接调用 Hypercall，而是通过设置映射表中的协商 key 和缺页 key，建立协商 key 对应的协商 value 与缺页 key 的对应关系；进

而利用虚拟化缺页技术将缺页 key 映射的缺页 value 作为 Hypercall 功能类型的数据传递载体, 以实现内核态程序和 Hypervisor 之间的信息交互; 最终使得 Hypervisor 执行传递的 Hypercall 功能类型的 Hypercall; 这其实是一种间接实现 Hypercall 的方式。

一方面, 缺页 value 可以写入任意的 Hypercall 功能类型, 从而可以为 eBPF 的内核态程序提供全功能类型的 Hypercall。

另一方面, 由于上述间接实现 Hypercall 的方式无需修改虚拟机的内核源代码, 因此可以作为通用的 eBPF 的内核态程序的超级调用方案进行普及使用。

在实际应用中, 一些业务需要传输特定参数才能正确执行, 同样的在调用 Hypercall 时也存在传输参数的需求。而由于映射表中缺页 value 已经用于写入 Hypercall 的功能类型, 无法再用于传递 Hypercall 的参数, 因而上述实施例只能提供无参数的 Hypercall; 而无法提供带参数的 Hypercall。

为了解决这一问题, 本说明书还提供了下面的实施例以实现增强型包过滤器中内核态程序的带参数的 Hypercall, 并且这种 Hypercall 可以支持传递任意个数的参数。

具体地, 映射表中可以设置多个缺页 key, 不同缺页 key 可以用于表示不同的个数参数; 相应地, 不同的缺页 key 也需要写入到不同的协商 key 对应的协商 value 中; 以及设置用于写入不同参数的参数 key。与协商 key 和缺页 key 类似的, 参数 key 也可以由内核态程序和 Hypervisor 协商后确定。

在上述步骤 1.6 中, 还可以包括:

在所述超级调用指令中还指定有超级调用的至少 1 个参数时, 所述内核态程序从所述映射表中的第一个参数键开始、将所述至少 1 个参数依次写入到至少 1 个参数键映射的参数值, 以使 1 个参数值对应 1 个参数。

该示例中, 对于内核态程序来说, 需要根据 Hypercall 指令中参数的参数个数, 读取对应的协商 value。此外, 内核态程序除了需要将所述 Hypercall 指令中指定的待执行 Hypercall 的功能类型作为 value 值写入协商 value 指向的缺页 key 映射的缺页 value, 还需要从第一个参数 key 开始、将所述至少 1 个参数作为 value 值依次写入到至少 1 个参数 key 映射的参数 value 中, 以使 1 个参数 value 对应 1 个参数。

相应地, 在上述步骤 2.6 中, 还可以包括:

所述虚拟机监控器基于所述缺页键表示的参数个数, 从所述映射表中的第一个参数键开始依次读取所述参数个数的参数键, 以获取每个参数键映射的参数值。

该示例中, 对于 Hypervisor 来说, 根据虚拟化缺页的内存地址就可以判断是带几个参数的 Hypercall; 然后可以从映射表中第一个参数 key 开始依次读取所述参数个数的参数 key, 以获取每个参数 key 映射的参数 value; 最后基于所述参数 value 指示的参数和所述缺页 value 指示的功能类型执行该功能类型的 Hypercall。

依然结合图 2 所示的 Map 的示意图加以说明。假设缺页 key=X 对应的内存地址 GPA0 表示无参数, 缺页 key=X+1 对应内存地址 GPA1 表示有 1 个参数, 缺页 key=X+2 对应内存

地址 GPA2 表示有 2 个参数;

那么, 如果 Hypercall 指令为没有参数的 Hypercall_type1 功能类型; 那么, eBPF 内核态程序可以读取无参数的协商 key=0 对应的 value=X, 从而向缺页 key=X 对应的缺页 value 中写入 Hypercal_type1, 以触发虚拟化缺页。Hypervisor 获取到虚拟化缺页的内存地址 GPA0, 表明是无参数的 Hypercall, 因此无需读取参数, 只需要基于该协商 value 指示的 Hypercall_type1 功能类型执行无参数的 Hypercall。

再例如, 如果 Hypercall 指令为带一个参数的 Hypercall_type2 功能类型; 那么, eBPF 内核态程序可以读取一个参数对应的协商 key=1 的 value=X+1, 从而向缺页 key=X+1 对应的的缺页 value 中写入 Hypercal_type2, 以及向第一个参数 key=Y 对应的参数 value 中写入参数 1, 以触发虚拟化缺页。Hypervisor 获取到虚拟化缺页的内存地址 GPA1, 表明是一个参数的 Hypercall, 因此需要从参数 key=Y 对应的参数 value 中读取参数 1, 从缺页 key=X+1 对应的缺页 value 中读取 Hypercall_type2 功能类型; 最后基于参数 1 和 Hypercall_type3 功能类型执行 Hypercall。

再例如, 如果 Hypercall 指令为带二个参数的 Hypercall_type3 功能类型; 那么, eBPF 内核态程序可以读取二个参数对应的协商 key=2 的 value=X+2, 从而向缺页 key=X+2 对应的缺页 value 中写入 Hypercal_type3, 以及向第一个参数 key=Y 对应的参数 value 中写入参数 1, 向第二个参数 key=Y+1 对应的参数 value 中写入参数 2, 以触发虚拟化缺页。Hypervisor 获取到虚拟化缺页的内存地址 GPA2, 表明是二个参数的 Hypercall, 因此需要分别从参数 key=Y 和参数 key=Y+1 对应的参数 value 中读取参数 1 和参数 2, 从缺页 key=X+2 对应的缺页 value 中读取 Hypercall_type3 功能类型; 最后基于参数 1、参数 2 和 Hypercall_type3 功能类型执行 Hypercall。

同理, 对于 Hypercall 指令中有三个参数、四个参数、……都做相似处理, 这里不再进行赘述。

应该上述示例, 通过在映射表中设置参数 key, 将参考 key 映射的参考 value 作为 Hypercall 参数的数据传递载体, 配合缺页 value, 不仅可以实现全功能的 Hypercall, 而且可以支持不同参数个数的 Hypercall。

以下参考图 3 所示的以 Hypervisor 为执行主体的方法实施例的示意图, 所述方法包括:

步骤 310, 响应于内核态程序触发的虚拟化缺页, 获取触发所述虚拟化缺页的内存地址;

步骤 320, 判断所述虚拟化缺页的内存地址是否和映射表中的缺页键对应的内存地址一致; 其中, 所述映射表记录了键值对之间的映射关系, 所述缺页键为所述内核态程序和所述虚拟机监控器共同在所述映射表中确定的键, 所述映射表中的每个键唯一对应一个内存地址;

步骤 330, 如果所述虚拟化缺页的内存地址和缺页键对应的内存地址一致, 从所述映

射表中读取所述缺页键映射的缺页值；其中，所述缺页值包括所述内核态程序在触发所述虚拟化缺页之前写入的超级调用的功能类型；

步骤 340，基于所述缺页值指示的功能类型执行该功能类型的超级调用。

上述图 3 所示实施例以及后续可选的实施例都可以对应前述图 1 所示实施例，具体步骤细节可以参考图 1 相关的实施例，这里不再进行赘述。

应用上述实施例，通过设置映射表中的协商 key 和缺页 key，并建立协商 key 对应的协商 value 与缺页 key 的对应关系；进而利用虚拟化缺页技术将缺页 key 映射的缺页 value 作为 Hypercall 功能类型的数据传递载体，以实现内核态程序和 Hypervisor 之间的信息交互。

具体地，在内核态程序将 Hypercall 功能类型写入缺页 key 映射的缺页 value 后，触发虚拟化缺页；而 Hypervisor 响应于虚拟化缺页，读取内核态程序写入缺页 key 映射的缺页 value 中的 Hypercall 功能类型，从而执行该 Hypercall 功能类型的 Hypercall。如此实现 eBPF 的内核态程序间接的调用 Hypercall。

一方面，缺页 value 可以写入任意的 Hypercall 功能类型，从而可以为 eBPF 的内核态程序提供全功能类型的 Hypercall。

另一方面，由于上述间接实现 Hypercall 的方式无需修改虚拟机的内核源代码，因此可以作为通用的增强型包过滤器的内核态程序的超级调用方案使用。

在一示例性可选的实施例中，所述步骤 310 中，响应于内核态程序的触发的虚拟化缺页，可以包括：

响应于内核态程序将超级调用的功能类型写入映射表中的缺页键映射的缺页值后触发的虚拟化缺页。

应该上述示例，通过设置缺页 key，当 eBPF 的内核态程序在缺页 key 映射的缺页 value 写入任何数据时均可以触发虚拟化缺页，从而利用缺页 value 作为 Hypercall 功能类型的数据传递载体，以实现 eBPF 的内核态程序和虚拟机的 Hypervisor 之间的信息交互。

在一示例性可选的实施例中，不同缺页键用于表示不同的参数个数；在图 3 所示实施例的基础上，还可以包括：

基于所述缺页键表示的参数个数，从所述映射表中的第一个参数键开始依次读取所述参数个数的参数键，以获取每个参数键映射的参数值；其中，所述参数值为所述内核态程序写入的超级调用的参数，1 个参数值对应 1 个参数；

所述步骤 340，基于所述缺页键指示的功能类型执行该功能类型的超级调用，可以包括：

基于所述参数值指示的参数和所述缺页值指示的功能类型执行该功能类型的超级调用。

应该上述示例，通过在映射表中设置参数 key，将参考 key 映射的参考 value 作

为 Hypercall 参数的数据传递载体，配合缺页 value，可以实现带参数的 Hypercall，并且支持传递任意个数的参数。

在一示例性可选的实施例中，在上述步骤 310 之前，还可以包括：

5 基于所述虚拟机监控器所在的虚拟机的身份信息，获取所述虚拟机中的增强型包过滤器维护的映射表；

将所述映射表中的协商键映射的协商值修改为缺页键；以使所述映射表维护的所述协商键映射的协商值指示所述缺页键；其中，所述协商键为所述内核态程序和所述虚拟机监控器共同在所述映射表中确定的键。

10 应该上述示例，Hypervisor 通过虚拟机的身份信息来获取 eBPF 的映射表，进而写入用于指向缺页 key 的协商 value；而在协商 value 写入后，内核态程序就可以具备 Hypercall 功能，进而可以实现内核态程序的 Hypercall。

以下参考图 4 所示的以内核态程序为执行主体的方法实施例的示意图，所述方法包括：

15 步骤 410，响应于超级调用指令，读取增强型包过滤器维护的映射表中的协商键映射的协商值；其中，所述映射表记录了键值对之间的映射关系，所述协商值包括虚拟机监控器预先写入的用于指示缺页键的值；所述协商键和缺页键为所述内核态程序和所述虚拟机监控器共同在所述映射表中确定的键；

步骤 420，基于所述协商值查询所述映射表中映射的缺页键；

20 步骤 430，将所述超级调用指令中指定的超级调用的功能类型写入所述缺页键映射的缺页值，从而触发虚拟化缺页；以使所述虚拟机监控器响应于所述虚拟化缺页，基于所述缺页值指示的功能类型执行该功能类型的超级调用。

上述图 4 所示实施例以及后续可选的实施例都可以对应前述图 1 所示实施例，具体步骤细节可以参考图 1 相关的实施例，这里不再进行赘述。

25 应用上述实施例，通过设置映射表中的协商 key 和缺页 key，并建立协商 key 对应的协商 value 与缺页 key 的对应关系；进而利用虚拟化缺页技术将缺页 key 映射的缺页 value 作为 Hypercall 功能类型的数据传递载体，以实现内核态程序和 Hypervisor 之间的信息交互。

30 具体地，在内核态程序将 Hypercall 功能类型写入缺页 key 映射的缺页 value 后，触发虚拟化缺页；而 Hypervisor 响应于虚拟化缺页，读取内核态程序写入缺页 key 映射的缺页 value 中的 Hypercall 功能类型，从而执行该 Hypercall 功能类型的 Hypercall。如此实现 eBPF 的内核态程序间接的调用 Hypercall。

一方面，缺页 value 可以写入任意的 Hypercall 功能类型，从而可以为 eBPF 的内核态程序提供全功能类型的 Hypercall。

35 另一方面，由于上述间接实现 Hypercall 的方式无需修改虚拟机的内核源代码，因此可以作为通用的增强型包过滤器的内核态程序的超级调用方案使用。

在一示例性可选的实施例中，所述方法还包括：

在所述超级调用指令中还指定有超级调用的至少 1 个参数时，从所述 Map 中的第一个参数 key 开始、将所述至少 1 个参数作为 value 值依次写入到至少 1 个参数 key 对应的参数 value 中，以使 1 个参数 value 对应 1 个参数；

- 5 所述以使所述虚拟机监控器基于所述缺页 value 指示的功能类型执行该功能类型的超级调用，包括：

以使所述虚拟机监控器基于所述参数 value 指示的参数和所述缺页 value 指示的功能类型执行该功能类型的超级调用。

- 10 应该上述示例，通过在 Map 中设置参数 key，将参考 key 对应的参考 value 作为 Hypercall 参数的数据传递载体，配合缺页 value，可以实现带参数的 Hypercall，并且支持传递任意个数的参数。

该实施例对应了前述图 1 所述实施例，具体步骤细节可以参考图 1 实施例，在本实施例中不再进行赘述。

- 15 与前述增强型包过滤器的内核态程序的超级调用方法实施例相对应，本说明书还提供了增强型包过滤器的内核态程序的超级调用装置的实施例。所述装置实施例可以通过软件实现，也可以通过硬件或者软硬件结合的方式实现。以软件实现为例，作为一个逻辑意义上的装置，是通过其所在设备的处理器将非易失性存储器中对应的计算机业务程序指令读取到内存中运行形成的。从硬件层面而言，如图 5 所示，为本说明书增强型包过滤器的内核态程序的超级调用装置所在设备的一种硬件结构图，除了图 20 5 所示的处理器、网络接口、内存以及非易失性存储器之外，实施例中装置所在的设备通常根据增强型包过滤器的内核态程序的超级调用实际功能，还可以包括其他硬件，对此不再赘述。

- 25 请参见图 6，为本说明书一实施例提供的增强型包过滤器的内核态程序的超级调用装置的模块图。所述装置对应了图 3 所示实施例，应用于虚拟机监控器，所述装置包括：

响应单元 610，响应于内核态程序触发的虚拟化缺页，获取触发所述虚拟化缺页的内存地址；

- 30 判断单元 620，判断所述虚拟化缺页的内存地址是否和映射表中的缺页键对应的内存地址一致；其中，所述映射表记录了键值对之间的映射关系，所述缺页键为所述内核态程序和所述虚拟机监控器共同在所述映射表中确定的键，所述映射表中的每个键唯一对应一个内存地址；

读取单元 630，如果所述虚拟化缺页的内存地址和缺页键对应的内存地址一致，从所述映射表中读取所述缺页键映射的缺页值；其中，所述缺页值包括所述内核态程序在触发所述虚拟化缺页之前写入的超级调用的功能类型；

- 35 调用单元 640，基于所述缺页值指示的功能类型执行该功能类型的超级调用。

可选的，所述响应单元 610，进一步用于响应于内核态程序将超级调用的功能类型写入映射表中的缺页键映射的缺页值后触发的虚拟化缺页。

可选的，不同缺页键用于表示不同的参数个数；

所述读取单元 630，还用于基于所述缺页键表示的参数个数，从所述映射表中的第一个参数键开始依次读取所述参数个数的参数键，以获取每个参数键映射的参数值；其中，所述参数值为所述内核态程序写入的超级调用的参数，1 个参数值对应 1 个参数；

所述调用单元 640，进一步用于基于所述参数值指示的参数和所述缺页值指示的功能类型执行该功能类型的超级调用。

可选的，所述装置还包括：

10 写入单元，基于所述虚拟机监控器所在的虚拟机的身份信息，获取所述虚拟机中的增强型包过滤器维护的映射表；将所述映射表中的协商键映射的协商值修改为缺页键；以使所述映射表维护的所述协商键映射的协商值指示所述缺页键；其中，所述协商键为所述内核态程序和所述虚拟机监控器共同在所述映射表中确定的键。

15 请参见图 7，为本说明书一实施例提供的增强型包过滤器的内核态程序的超级调用装置的模块图。所述装置对应了图 4 所示实施例，应用于内核态程序，所述装置包括：

响应单元 710，响应于超级调用指令，读取增强型包过滤器维护的映射表中的协商键映射的协商值；其中，所述映射表记录了键值对之间的映射关系，所述协商值包括虚拟机监控器预先写入的用于指示缺页键的值；所述协商键和缺页键为所述内核态程序和所述虚拟机监控器共同在所述映射表中确定的键；

查询单元 720，基于所述协商值查询所述映射表中映射的缺页键；

调用单元 730，将所述超级调用指令中指定的超级调用的功能类型写入所述缺页键映射的缺页值，从而触发虚拟化缺页；以使所述虚拟机监控器响应于所述虚拟化缺页，基于所述缺页值指示的功能类型执行该功能类型的超级调用。

25 可选的，所述调用单元 730，还用于在所述超级调用指令中还指定有超级调用的至少 1 个参数时，从所述映射表中的第一个参数键开始、将所述至少 1 个参数依次写入到至少 1 个参数键映射的参数值，以使 1 个参数值对应 1 个参数。

30 上述实施例阐明的系统、装置、模块或单元，具体可以由计算机芯片或实体实现，或者由具有某种功能的产品来实现。一种典型的实现设备为计算机，计算机的具体形式可以是个人计算机、膝上型计算机、蜂窝电话、相机电话、智能电话、个人数字助理、媒体播放器、导航设备、电子邮件收发设备、游戏控制台、平板计算机、可穿戴设备或者这些设备中的任意几种设备的组合。

上述装置中各个单元的功能和作用的实现过程具体详见上述方法中对应步骤的实现过程，在此不再赘述。

35 对于装置实施例而言，由于其基本对应于方法实施例，所以相关之处参见方法实

5 实施例的部分说明即可。以上所描述的装置实施例仅仅是示意性的，其中所述作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部模块来实现本说明书方案的目的。本领域普通技术人员在不付出创造性劳动的情况下，即可以理解并实施。

以上图 6 和图 7 描述的区块链中智能合约的执行装置的内部功能模块和结构示意图，其实质上的执行主体可以为一种电子设备，包括：

- 处理器；
- 用于存储处理器可执行指令的存储器；
- 10 其中，所述处理器被配置为上述任一增强型包过滤器的内核态程序的超级调用的实施例。

在上述电子设备的实施例中，应理解，该处理器可以是中央处理单元（英文：Central Processing Unit，简称：CPU），还可以是其他通用处理器、数字信号处理器（英文：Digital Signal Processor，简称：DSP）、专用集成电路（英文：Application Specific Integrated Circuit，简称：ASIC）等。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等，而前述的存储器可以是只读存储器（英文：read-only memory，缩写：ROM）、随机存取存储器（英文：random access memory，简称：RAM）、快闪存储器、硬盘或者固态硬盘。结合本发明实施例所公开的方法的步骤可以直接体现为硬件处理器执行完成，或者用处理器中的硬件及软件模块组合执行

20 完成。

本说明书中的各个实施例均采用递进的方式描述，各个实施例之间相同相似的部分互相参见即可，每个实施例重点说明的都是与其他实施例的不同之处。尤其，对于电子设备实施例而言，由于其基本相似于方法实施例，所以描述的比较简单，相关之处参见方法实施例的部分说明即可。

25 本领域技术人员在考虑说明书及实践这里公开的发明后，将容易想到本说明书的其它实施方案。本说明书旨在涵盖本说明书的任何变型、用途或者适应性变化，这些变型、用途或者适应性变化遵循本说明书的一般性原理并包括本说明书未公开的本技术领域中的公知常识或惯用技术手段。说明书和实施例仅被视为示例性的，本说明书的真正范围和精神由下面的权利要求指出。

30 应当理解的是，本说明书并不局限于上面已经描述并在附图中示出的精确结构，并且可以在不脱离其范围进行各种修改和改变。本说明书的范围仅由所附的权利要求来限制。

权 利 要 求 书

1、一种增强型包过滤器的内核态程序的超级调用方法，应用于虚拟机监控器，所述方法包括：

响应于内核态程序触发的虚拟化缺页，获取触发所述虚拟化缺页的内存地址；

5 判断所述虚拟化缺页的内存地址是否和映射表中的缺页键对应的内存地址一致；其中，所述映射表记录了键值对之间的映射关系，所述缺页键为所述内核态程序和所述虚拟机监控器共同在所述映射表中确定的键，所述映射表中的每个键唯一对应一个内存地址；

如果所述虚拟化缺页的内存地址和缺页键对应的内存地址一致，从所述映射表中读取所述缺页键映射的缺页值；其中，所述缺页值包括所述内核态程序在触发所述虚拟化
10 缺页之前写入的超级调用的功能类型；

基于所述缺页值指示的功能类型执行该功能类型的超级调用。

2、根据权利要求 1 所述的方法，所述响应于内核态程序的触发的虚拟化缺页，包括：

响应于内核态程序将超级调用的功能类型写入映射表中的缺页键映射的缺页值后触
15 发的虚拟化缺页。

3、根据权利要求 1 所述的方法，不同缺页键用于表示不同的参数个数；所述方法还包括：

基于所述缺页键表示的参数个数，从所述映射表中的第一个参数键开始依次读取所述参数个数的参数键，以获取每个参数键映射的参数值；其中，所述参数值为所述内
20 核态程序写入的超级调用的参数，1 个参数值对应 1 个参数；

所述基于所述缺页键指示的功能类型执行该功能类型的超级调用，包括：

基于所述参数值指示的参数和所述缺页值指示的功能类型执行该功能类型的超级调用。

4、根据权利要求 1 所述的方法，所述方法还包括：

25 基于所述虚拟机监控器所在的虚拟机的身份信息，获取所述虚拟机中的增强型包过滤器维护的映射表；

将所述映射表中的协商键映射的协商值修改为缺页键；以使所述映射表维护的所述协商键映射的协商值指示所述缺页键；其中，所述协商键为所述内核态程序和所述虚拟机监控器共同在所述映射表中确定的键。

30 5、一种增强型包过滤器的内核态程序的超级调用方法，应用于内核态程序，所述方法包括：

响应于超级调用指令，读取增强型包过滤器维护的映射表中的协商键映射的协商值；其中，所述映射表记录了键值对之间的映射关系，所述协商值包括虚拟机监控器预先写入的用于指示缺页键的值；所述协商键和缺页键为所述内核态程序和所述虚拟机监控器共同
35 在所述映射表中确定的键；

基于所述协商值查询所述映射表中映射的缺页键；

将所述超级调用指令中指定的超级调用的功能类型写入所述缺页键映射的缺页值，从而触发虚拟化缺页；以使所述虚拟机监控器响应于所述虚拟化缺页，基于所述缺页值指示的功能类型执行该功能类型的超级调用。

5 6、根据权利要求5所述的方法，所述方法还包括：

在所述超级调用指令中还指定有超级调用的至少1个参数时，从所述映射表中的第一个参数键开始、将所述至少1个参数依次写入到至少1个参数键映射的参数值，以使1个参数值对应1个参数。

10 7、一种增强型包过滤器的内核态程序的超级调用装置，应用于虚拟机监控器，所述装置包括：

响应单元，响应于内核态程序触发的虚拟化缺页，获取触发所述虚拟化缺页的内存地址；

判断单元，判断所述虚拟化缺页的内存地址是否和映射表中的缺页键对应的内存地址一致；其中，所述映射表记录了键值对之间的映射关系，所述缺页键为所述内核态程序和
15 所述虚拟机监控器共同在所述映射表中确定的键，所述映射表中的每个键唯一对应一个内存地址；

读取单元，如果所述虚拟化缺页的内存地址和缺页键对应的内存地址一致，从所述映射表中读取所述缺页键映射的缺页值；其中，所述缺页值包括所述内核态程序在触发所述虚拟化缺页之前写入的超级调用的功能类型；

20 调用单元，基于所述缺页值指示的功能类型执行该功能类型的超级调用。

8、一种增强型包过滤器的内核态程序的超级调用装置，应用于内核态程序，所述装置包括：

响应单元，响应于超级调用指令，读取增强型包过滤器维护的映射表中的协商键映射的协商值；其中，所述映射表记录了键值对之间的映射关系，所述协商值包括虚拟机监
25 控器预先写入的用于指示缺页键的值；所述协商键和缺页键为所述内核态程序和所述虚拟机监控器共同在所述映射表中确定的键；

查询单元，基于所述协商值查询所述映射表中映射的缺页键；

调用单元，将所述超级调用指令中指定的超级调用的功能类型写入所述缺页键映射的缺页值，从而触发虚拟化缺页；以使所述虚拟机监控器响应于所述虚拟化缺页，基于所述
30 缺页值指示的功能类型执行该功能类型的超级调用。

9、一种电子设备，包括：

处理器；

用于存储处理器可执行指令的存储器；

其中，所述处理器被配置为执行上述权利要求1-6中任一项所述的方法。

35 10、一种计算机可读存储介质，当所述计算机可读存储介质中的指令由电子设备

的处理器执行时，使得所述电子设备能够执行如权利要求 1-6 中任一项所述的方法。