Виртуализация ОС «Эльбрус»

Рыбаков С.А., АО «МЦСТ»

Тезисы

Технология виртуализации операционных систем, используемая в современных вычислительных системах, позволяет запустить на высокопроизводительном сервере несколько изолированных друг от друга виртуальных машин, эмулирующих реальные физические серверы со своими (гостевыми) операционными системами. Виртуализация позволяет значительно повысить степень утилизации вычислительных ресурсов сервера, а также увеличить уровень безопасности системы, в частности, при организации облачных вычислений.

Важнейшим шагом на пути к высокопроизводительным и безопасным системам виртуализации является добавление аппаратной поддержки этой технологии в микропроцессорах. Аппаратная поддержка виртуализации, добавленная в микропроцессорах нового поколения «Эльбрус-16С», направлена на поддержку режима полной виртуализации, улучшение защиты режима паравиртуализации, а также повышение производительности обоих режимов. Аппаратные решения включают: новые режимы исполнения и переходы между ними, механизмы теневой и двухуровневой трансляции виртуальных адресов, двухуровневую трансляцию DMA (Direct Memory Access) адресов, изоляцию устройств ввода-вывода друг от друга, а также изоляцию, маршрутизацию и доставку гостевых прерываний. Представленные решения сравнимы с передовыми разработками архитектур x86 (Intel и AMD), MIPS и ARM.

Введение

Виртуализация операционной системы (ОС) – это технология предоставления набора аппаратных ресурсов (вычислительных, оперативной памяти, устройств ввода-вывода) гостевым операционным системам с обеспечением их логической изоляции друг от друга. Технология виртуализации позволяет значительно повысить степень утилизации вычислительных ресурсов сервера, а также увеличить уровень безопасности системы, в частности, при организации облачных вычислений. В докладе приведено описание основных программных и аппаратных средств поддержки виртуализации архитектуры «Эльбрус».

Паравиртуализация и полная виртуализация

Паравиртуализация – это техника виртуализации, при которой гостевая ОС подготавливается к исполнению в виртуализированной среде, для чего ее ядро незначительно модифицируется. Паравиртуализированная гостевая ОС понимает, что она исполняется внутри виртуальной машины (ВМ), что позволяет применять существенные оптимизации для улучшения производительности системы. Главным недостатком этой техники является необходимость модификации ядра гостевой ОС – это возможно лишь в случае, если коды ОС открыты и есть лицензия на их модификацию.

Полной виртуализацией называется техника виртуализации, при которой гипервизору удается полностью эмулировать поведение аппаратуры, вследствие чего оказывается возможным запуск немодифицированных гостевых ОС. Полная виртуализация может быть достигнута либо с помощью двоичной трансляции, динамически подменяющей привилегированные действия гостевой ОС на эмулирующий их код гипервизора, либо с помощью аппаратной поддержки виртуализации. По производительности полная виртуализация уступает паравиртуализации, особенно в области виртуализации ввода-вывода. Этот недостаток

часто компенсируется установкой дополнительных, паравиртуализированных драйверов устройств в гостевые ОС.

Гипервизор QEMU-KVM

В ОС «Эльбрус» в роли гипервизора выступает связка QEMU-KVM. QEMU (Quick Emulator) [1] – это пользовательское приложение, эмулирующее для гостевой ОС вычислительные ресурсы, оперативную память и устройства ввода-вывода. Для начальной настройки и собственно запуска виртуальной машины QEMU использует модули ядра операционной системы Linux KVM (Kernel Virtual Machine) [2]. Передача управления из QEMU в KVM происходит с помощью специальных системных вызовов IOCTL (Input/Output Control). Поскольку QEMU – пользовательское приложение и запускается поверх ОС, пара QEMU-KVM является гипервизором 2-го типа. Это отличает QEMU-KVM от гипервизоров 1-го типа (например, Xen [3]), запускаемых напрямую на аппаратном обеспечении. Каждому процессу QEMU соответствует одна виртуальная машина. Гостевые вычислительные ядра (vCPU), и виртуальные устройства ввода-вывода эмулируются отдельными потоками процесса QEMU, исполняющимися параллельно.

Аппаратная поддержка виртуализации в микропроцессорах «Эльбрус»

Добавленная в микропроцессорах «Эльбрус-16С» нового поколения аппаратная поддержка виртуализации [4] позволяет достичь полной виртуализации системы. В качестве гостя может быть без изменений запущена любая ОС «Эльбрус», либо с помощью системы динамической двоичной трансляции любая ОС на основе архитектуры Intel x86 (Linux или Windows). Аппаратная поддержка также улучшает защиту и производительность режима паравиртуализации. Рассмотрим подробнее три области, в которых были проведены аппаратные доработки: средства виртуализации вычислительных ресурсов, памяти и ввода-вывода.

Виртуализация вычислительных ресурсов

Рассмотрим процесс запуска гостевой ОС гипервизором. После первичной настройки виртуальной машины QEMU посредством специального вызова IOCTL передает управление в KVM, где происходит программное переключение части рабочего состояния (пассивного контекста) гипервизор-гость. Переключение оставшейся и наиболее критичной части контекста (содержащей в том числе указатели на стеки, базовый адрес таблицы страниц ОС, и другие управляющие регистры) производится атомарно, при выполнении новой инструкции запуска гостя glaunch. Аппаратное переключение контекста достигается за счет дублирования подмножества архитектурных регистров: оригинальный комплект регистров называется активным контекстом, а комплект, вводимый для поддержки виртуализации – теневым. При входе в гостевой режим активный и теневой контекст атомарно меняются ролями. Состав активного контекста определяется необходимостью функционирования гипервизора во время запуска гостя. Реализация переключений контекстов ближе к варианту MIPS VM [5], чем вариантам Intel[6] и AMD[7], производящим атомарное переключение через структуру данных в системной памяти.

В предыдущих поколениях микропроцессоров Эльбрус были определены два режима исполнения: привилегированный (для ядра ОС) и пользовательский (для прикладных программ). В микропроцессорах «Эльбрус-16С» поддержано два новых режима: гостевой привилегированный и гостевой пользовательский. И если гостевой пользовательский режим практически не отличается от обычного, то на гостевой привилегированный режим накладываются дополнительные ограничения: ряд действий в этом режиме приводит к

принудительному возврату в гипервизор, перехвату (аналог VM-Exit в Intel VMX[6], VMEXT в AMD SVM[7] и Hypervisor Trap в ARM[8]).

Типичными причинами перехватов являются попытки гостя получить доступ к эмулируемым гипервизором привилегированным ресурсам (например, регистрам устройства ввода-вывода), отсутствие трансляции виртуальных или физических адресов гостя, а также истечение выделенного гостю планировщиком кванта времени. После возврата в гипервизор, KVM анализирует причину перехвата (сохраненную на специально введенных регистрах), и, если ее удается устранить локально, гость запускается снова. При этом, для улучшения производительности, пассивный контекст переключается лишь частично. Если же необходима эмуляция ввода-вывода, то KVM возвращает управление QEMU, что требует полной смены пассивного контекста. Таким образом, перехваты можно разделить на две группы: легковесные (обрабатываемые в KVM), и тяжеловесные (с выходом в QEMU).

Для поддержки паравиртуализации был также аппаратно поддержан механизм гипервызовов, опирающийся на команды hcall (аналог инструкции VMMCALL в AMD SVM[7]) и hret. Использование новых команд вместо системных вызовов для явного вызова гипервизора из гостевой ОС позволяет полноценно использовать новые гостевые режимы исполнения и повысить защиту паравиртуализации.

Виртуализация подсистемы памяти

Физическая память виртуальной машины эмулируется гипервизором и является виртуальной памятью процесса QEMU. Таким образом, при виртуализации можно различить четыре основных типа адресов: системные физические HPA (Hypervisor Physical Address), системные виртуальные HVA (Hypervisor Virtual Address), гостевые физические GPA (Guest Physical Address) и гостевые виртуальные GVA (Guest Virtual Address). В новом поколении процессоров «Эльбрус» поддержано два механизма трансляции гостевых адресов в системные физические: теневой и двухуровневый.

При теневой трансляции при трансляции гостевых адресов не используется формируемая гостевой ОС таблица страниц (ТС): вместо нее работает формируемая гипервизором теневая таблица страниц (Shadow Page Table, SPT). Поскольку процесс поиска изменений в гостевой ТС может быть достаточно долгим, в ОС «Эльбрус» применяется широко распространенная оптимизация: гостевая ТС полностью закрывается по записи, каждое изменение ТС гостем перехватывается и отображается в теневой таблице. Программный механизм теневой трансляции традиционно применяется для паравиртуализации на различных архитектурах (например, virtual TLB в терминологии Intel[6]), но в архитектуре «Эльбрус» этот механизм поддержан и аппаратно. Это позволяет кэшировать промежуточные и конечные результаты гостевых трансляций, увеличивая производительность паравиртуализации.

При полной виртуализации гипервизор не должен знать ничего об организации гостевой ТС и вмешиватьтся в ее организацию. Механизм двухуровневой трансляции адресов (Two-Dimensional Paging, TDP) позволяет транслировать гостевые адреса в два независимых этапа: трансляция GVA-GPA производится по гостевым таблицам, а трансляция GPA-HPA – по таблицам гипервизора. Схожий принцип используется в механизмах EPT [], Nested Paging [] и Two-Stage Address Translation []. Главным недостатком этого механизма, по сравнению с SPT, является дополнительный этап трансляции, причем для каждого уровня ТС гостя, что значительно увеличивает суммарное число шагов трансляции адреса.

Виртуализация подсистемы ввода-вывода

Традиционный подход к виртуализации ввода-вывода, который использовался и до реализации аппаратной поддержки виртуализации, отделяет виртуальный ввод-вывод от физического. Гипервизор предоставляет гостю набор виртуальных устройств, после чего обращения к ним гостя перехватываются и эмулируются, используя в том числе и реальные устройства. Такой принцип виртуализации ввода-вывода характеризуется как интерпозиция [9].

У этого подхода есть ряд достоинств. Существенно, что при интерпозиции ввода-вывода гипервизору в любой момент времени доступно состояние виртуальной машины целиком, включая все виртуальные устройства. Его можно, например, сохранить в виде файла, и позднее возобновить работу гостя с сохраненного момента, возможно даже на другом физическом сервере. Другим преимуществом интерпозиции является консолидация ввода-вывода. Несколько виртуальных устройств могут обслуживаться одним реальным, что повысит его утилизацию. Есть также и обратная возможность – объединение нескольких реальных устройств в одно виртуальное с лучшими производительностью и отказоустойчивостью. Гипервизор также может поддержать функциональность для виртуальных устройств, не предоставляемую реальным оборудованием.

Метод программной эмуляции ввода-вывода

ОС взаимодействует с устройствами ввода-вывода через их регистры, интерфейс DMA (Direct Memory Access) и прерывания. Все эти способы взаимодействия моделируются внутри эмулятора QEMU. Однако эмуляция устройств часто вносит большие задержки, негативно влияющие на производительность гостевой ОС. Драйверы большинства реальных устройств содержат частые обращения к их регистрам. Поскольку при виртуализации каждое гостевое обращение к регистру устройства приводит к тяжеловесному перехвату, производительность работы с виртуальным устройством оказывается далекой от оптимальной.

Метод паравиртуализации ввода-вывода (virtio)

Для уменьшения задержек эмуляции используются специальные виртуальные устройства, предназначенные для эффективной виртуализации, – virtio [9]. С их применением улучшение производительности по сравнению с полной виртуализацией устройств достигается за счет значительного снижения числа перехватов и гостевых прерываний. Интерфейс virtio реализуется рядом виртуальных устройств и соответствующих им драйверов, включая серийный порт, жесткий диск и сетевой адаптер. Поскольку физических устройств virtio не существует, virtio являются элементами паравиртуализации гостя.

Метод прямого назначения устройства

Метод прямого назначения устройства позволяет отдать гостю в пользование реальное физическое устройство. При этом другие гости и даже гипервизор теряют возможность пользоваться этим устройством. Хотя этот метод обеспечивает наилучшую производительность, он имеет ряд недостатков. Во-первых, при пробросе приходится отказаться от всех преимуществ интерпозиции ввода-вывода. Во-вторых, метод плохо масштабируется. В-третьих, во избежание отказов страниц при DMA, проброс устройства требует от гипервизора резидентирования всего гостевого физического адресного пространства в системной памяти. При этом гипервизор теряет возможность оптимизации этой памяти, а также использования ее для других гостей или приложений.

При передаче гостю контроля над устройством, необходимо обеспечить полную изоляцию гипервизора и других гостей от этого устройства. Например, нельзя позволять проброшенному устройству, по ошибке или намеренно, запускать DMA по чужой системной физической памяти или генерировать MSI-прерывания гипервизору. Защиту от такого поведения гостя в проектах «Эльбрус» обеспечивают блок IOMMU (Input/Output Memory Management Unit) и контроллер прерываний EPIC (Elbrus Programmable Interrupt Controller). IOMMU обеспечивает изоляцию DMA всех PCI устройств в системе друг от друга, за счет использования отдельной ТС для каждого устройства. При этом, по аналогии с MMU (Memory Management Unit), в каждой ТС поддерживается двухуровневая трансляция адресов DMA, GVA-GPA по гостевым таблицам, и GPA-HPA по гипервизорным. EPIC обеспечивает изоляцию, маршрутизацию и прямую доставку прерываний от внешних устройств гостевым ОС.

Реализованный механизм трансляции адресов в IOMMU близок к варианту AMD [7]. В отличие от реализации AMD, позволяющей использовать в IOMMU произвольный размер страниц (кратный 4 Кб), на «Эльбрус» доступно только 3 фиксированных размера страниц (4 Кб, 2 Мб и 1 Гб), что позволило значительно упростить аппаратную реализацию. Кроме того, реализация IOMMU на «Эльбрус» не поддерживает идентификаторы адресного пространства процесса (интерфейс PASID PCI Express): при передаче одной ВМ в пользование нескольких реальных устройств, трансляция их DMA обращений будет выполняться по одной ТС, без изоляции указанных устройств друг от друга.

Виртуализация прерываний

Независимо от выбранного метода виртуализации ввода-вывода, гипервизор обязан доставлять гостевой ОС прерывания от предоставленных ей устройств. Однако в отсутствие аппаратной поддержки программная доставка гостевых прерываний приводит к значительным потерям производительности ВМ. Во-первых, для доставки прерывания гостевому vCPU гипервизор обязан снять его с исполнения. Во-вторых, при обработке прерывания гостевая ОС неоднократно обращается к контроллеру прерываний (в частности, для определения вектора прерывания, и для сигнала об окончании обработки прерывания), что также приводит к перехвату или гипервызову. В ОС «Эльбрус» применяется широко распространенная оптимизация: модель контроллера прерываний реализована в модулях KVM во избежание тяжеловесного переключения контекста при выходе в QEMU. В то же время аппаратная поддержка позволяет еще сильнее увеличить производительность гостевой системы.

В микропроцессорах «Эльбрус» нового поколения регистры контроллера прерываний EPIC были продублированы на каждом вычислительном ядре. Эти регистры доступны виртуальному ядру, исполняющемуся в текущий момент на физическом ядре, без перехватов. При снятии виртуального ядра с исполнения гипервизор сохраняет состояние гостевых регистров EPIC в память. При этом атомарность переключения гостевого контекста EPIC поддержана программно и аппаратно. Гостевые прерывания также отличаются от гипервизорных на аппаратном уровне, и если прерывания гипервизора могут прервать исполнение гостя (вызвать перехват), то гостевые прерывания никак не влияют на работу гипервизора. Если при доставке гостевого прерывания целевой vCPU оказывается снят с исполнения, то прерывание сохраняется в оперативную память и доставляется в момент постановки vCPU на физическое ядро.

Указанный подход к доставке виртуальных прерываний с реализацией полноценного набора регистров контроллера отличается от подхода Intel [7], где гостевые запросы к регистрам контроллера сводятся к обращениям в оперативную память, с аппаратной эмуляцией побочных эффектов для отдельных регистров.

Выводы

В докладе представлены программные решения, основанные на гипервизоре QEMU-KVM, а также аппаратные доработки архитектуры «Эльбрус», направленные на поддержку режима полной виртуализации, улучшение защиты режима паравиртуализации и повышение производительности обоих режимов. Аппаратные решения включают: новые режимы исполнения и переходы между ними, механизмы теневой и двухуровневой трансляции виртуальных адресов, двухуровневую трансляцию DMA адресов, изоляцию устройств ввода-вывода друг от друга, а также изоляцию, маршрутизацию и доставку гостевых прерываний. Совокупный результат представленных новаций соответствует уровню передовых разработок архитектур x86 (Intel и AMD), MIPS и ARM.