УДК 004.946

ВИРТУАЛИЗАЦИЯ КАК ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ

В.Н. Захаров

В статье рассмотрены определения термина «виртуальный» и применения этого термина в сочетании с разными широко используемыми понятиями, в том числе виртуальная реальность, виртуальные миры и пр. Основное внимание уделено трактованию виртуализации как информационной технологии построения виртуальных машин и виртуальных сред. Обсуждаются условия эффективной виртуализуемости архитектуры компьютера, рассмотрены современные аппаратные и программные средства поддержки технологии виртуализации. Приводится пример использования виртуализации в реализации прозрачной отказоустойчивости серверов приложений.

1. Введение

В последние годы в средствах массовой информации стали активно использоваться термины «виртуализация», «виртуальный» в применении к самым разнообразным областям и с вкладыванием различного смысла в эти термины. Наиболее часто встречаются такие словосочетания, как *виртуальная память*, *виртуальный терминал*, *виртуальная машина*, *виртуальная реальность*, *виртуальные пространства*, *виртуальные обсерватории*, *виртуальные правительства*, *виртуальные миры*. В настоящей статье делается попытка некоторого толкования этих понятий и описания примеров их использования, как в широком смысле, так и в более точном, но применительно к информационным компьютерным технологиям.

2. Различные применения термина «виртуальный»

Согласно [1] «виртуальный» (лат. Virtualis) имеет два значения: 1) возможный; такой, который может или должен проявиться при определённых условиях, но в реальности не существующий; 2) созданный на экране компьютера; воспроизводимый компьютерными средствами. В данной статье мы не будем вдаваться в толкование и обсуждение виртуализации как общефилософского понятия в первом из указанных значений, а будем рассматривать этот термин во втором значении, а также его производные в связи с использованием компьютеров.

В [2] прилагательное «виртуальный» входит в несколько определяемых словосочетаний: виртуальные ресурсы, виртуальная база данных, виртуальная память, виртуальная реальность. При этом виртуальные (информационные) ресурсы (virtual (information) resources) определяются как информационные ресурсы других организаций, предприятий, фирм и т.п., доступные пользователям в режиме теледоступа по каналам глобальной связи, например Интернета. Виртуальная база данных (virtual database) трактуется как воображаемое представление данных, в которое может быть преобразована каждая из интегрируемых баз данных произвольной системы управления базами данных.

К наиболее ранним примерам применения термина «виртуальный» можно отнести понятия «виртуальная память» и «виртуальный терминал», появившиеся в 60-х годах прошлого века, а также «виртуальное соединение», получившее хождение в телефонии. Концепция виртуальной памяти стала известной по её использованию в манчестерской вычислительной машине Atlas. В [3] виртуальная память (virtual memory) определяется как система, при которой рабочее пространство процесса частично располагается в быстродействующей памяти и частично в некотором более медленном и более дешёвом внешнем запоминающем устройстве. При обращении пользователя к какой-нибудь области памяти система аппаратными средствами определяет, присутствует или нет физически нужная область в памяти, и, если она отсутствует, генерирует прерывание; это позволяет супервизору системы передать необходимый фрагмент данных из резервной памяти в быструю память.

Там же виртуальный терминал (virtual terminal) определяется как гипотетический терминал, который характеризуется всеми теми же параметрами, что и некоторый класс физических терминалов. Концепция виртуального терминала аналогична определению искусственного языка, на который и с которого можно переводить некоторый набор естественных языков. В целом ряде сетей с коммутацией пакетов сделаны попытки использования концепции виртуального терминала как средства преобразования протоколов между разнородными терминалами. В узле, передаю-

щем сообщение, оно переводится в формат виртуального терминала, а в выходном узле снова переводится на язык протокола приёмного терминала.

Под виртуальным соединением (virtual connection) понимается [3] логическое соединение между двумя конечными точками линии передачи в сети, которое для оконечного оборудования выглядит как физическое соединение. Использование виртуальных соединений нашло применение как в телефонии, так и в сетях передачи данных; такой принцип соединения стал средством увеличения степени использования каналов связи за счёт разделения их физических возможностей между множеством терминальных устройств.

В качестве примера виртуальных информационных ресурсов можно привести виртуальные обсерватории, объединяемые Международным Альянсом Виртуальных Обсерваторий (International Virtual Observatory Alliance (IVOA)), в который входит также Российская виртуальная обсерватория (РВО). Информационная инфраструктура РВО [4] базируется на стандартах IVOA, она является сервис-ориентированной, что означает зависимость компонентов системы от природы запрашиваемых и предоставляемых сервисов. Сервисы архитектуры IVOA делятся на три класса:

— сервисы данных (относительно простые сервисы, предоставляющие доступ к ресурсам астрономических данных, накопленных у разных участников альянса);

— вычислительные сервисы, позволяющие решать разнообразные классы астрономических задач, извлекать и анализировать информацию из различных информационных источников;

— сервисы регистрации, позволяющие осуществлять регистрацию в системе (подключение к ней) новых информационных ресурсов, а также новых вычислительных сервисов.

В РВО в качестве базового принципа, обеспечивающего унифицированный доступ к многочисленным (нескольким тысячам) источникам астрономических данных, предлагается использовать идею построения предметных посредников, позволяющих строить промежуточный слой между конечными пользователями и электронными коллекциями, регистрировать в нём информационные источники, снабжённые определённой метаинформацией. Информационные посредники обеспечивают не только унифицированный доступ к информации, но и выдачу полученных результатов в привычном для зарегистрированного пользователя

виде. То есть пользователь работает в виртуальной обсерватории, обращаясь к данным из всех зарегистрированных информационных ресурсов точно так же, как он работает в отдельной информационной системе.

Одним из наиболее модных и часто используемых в современной прессе терминов является *виртуальная реальность*. Этот термин трактуется достаточно широко. Так, в [2] виртуальная реальность (ВР, virtual reality) определяется как «искусственно созданный мир путём подмены окружающей действительности информацией, генерируемой компьютером». ВР вошла в число десяти самых перспективных (стратегических) направлений современной информатики, названных Национальным научным фондом США в конце XX в.

Виртуальная реальность — это, прежде всего, трёхмерная интерактивная графика, реализуемая в реальном масштабе времени, которая в сочетании с технологией отображения позволяет пользователю погрузиться в мир модели и непосредственно действовать в нём. Наряду с трёхмерной графикой ВР в интерактивном режиме обеспечивается использованием стереозвука и других специальных устройств ввода/вывода данных, имитирующих связь человека с воспроизводимым миром и происходящими в нём процессами. В качестве таких устройств, в частности, могут использоваться:

— шлемы-дисплеи, позволяющие «видеть» стереоскопическое изображение виртуального мира и передающие в ПК данные о положении и ориентации головы для изменения изображения в соответствии с «изменением» точки обзора;

— манипуляторы, манипуляторные устройства — в том числе специальные перчатки, передающие данные о движении рук и пальцев и позволяющие «брать» в руки, управлять положением объектов искусственно созданной среды и т. п.;

— стереоаудиосистемы, способные не только создавать объёмное звучание, но и передавать звуковое давление, например, при моделировании ударов;

— электромагнитные и пневматические устройства, передающие механические воздействия на человека в процессе имитации моделируемых процессов (например, ускорение, давление и т. п.).

Предтечей ВР считаются авиационные тренажёры. В настоящее время основными сферами применения ВР являются: проектирование новых изделий в машиностроении с виртуальным прототипированием, исключающим дорогостоящие макеты, системы

обучения водителей автомобилей, лётчиков, судоводителей, экипажей космических кораблей, диагностика и лечение заболеваний в медицине и здравоохранении, визуализация данных или моделей в научных исследованиях и многие другие.

В [5] в числе информационных технологий выделены *информационные технологии виртуальной реальности* и сформулированы следующие основные направления исследований, которые должны обеспечить дальнейшее развитие этого направления:

— развитие программного обеспечения трёхмерной машинной графики с учётом интерактивного режима работы;

— моделирование и взаимодействие в виртуальных средах, а именно, разработка информационных технологий, позволяющих включать человеческие фигуры в системы ВР, программного обеспечения для регистрации и интерпретации движений человека в виртуальной среде, технологий, позволяющих участникам-людям, автономным или ведомым «актёрам» совместно манипулировать объектами в виртуальном мире;

— разработка новых специальных технических средств ВР;

— исследование эргономики виртуальной среды.

Развитием и расширением использования термина виртуальная реальность является понятие «виртуальный мир». В области использования технологий виртуальных миров активная деятельность ведётся в Санкт-Петербургском государственном университете авиационно-космического приборостроения (ГУАП), где создана специальная лаборатория. В [6] описаны как общефилософские принципы и подходы к построению виртуальных миров, так и представлены основные характеристики и описания технических средств реализации ряда конкретных «виртуальных миров».

Констатируется, что виртуальный мир обеспечивает глубокое погружение человека в определённую среду (например, в искусственно созданный трёхмерный мир с шестью степенями свободы) и взаимодействие с объектами и персонажами этой среды в реальном времени с использованием различных характеристик человека — физических, психических, физиологических, личностных, познавательных и др., адекватных его повседневной жизни и деятельности или существенно расширяющих их. Используемое понятие «виртуальный мир» — это интегрирующее понятие, тесно связанное с понятиями «виртуальная реальность», «мультимодальный интерфейс» и «биокибернетический интерфейс». Виртуальные миры можно рассматривать как средство для просмотра естественных явлений, которые происходят во время коммуникации, независимо от технологической среды, как попытку использовать те средства (визуальные, звуковые, осязательные и т. п.), которые лучше «удовлетворяют» нашу естественную способность общения, чем принуждение нас к адаптации с неестественными средствами, как того требует технология.

Поддерживаемый компьютером виртуальный мир использует сильно связанные петли ощущение-действие для моделирования естественной связи. Мы действуем, это фиксирует компьютер, реагирует, мы получаем ответ, и так далее. Если соответствующая технология реализована хорошо, мы чувствуем эффект присутствия — ощущение того, что мы находимся в виртуальной среде. Это способствует установлению связи между вычислительным миром и нашим умственным миром.

Основными характеристиками виртуального мира являются погружение и интерактивность. *Погружение* — мера информации, окружающей и включающей человека, поступающей через его сенсорные средства. *Интерактивность* — мера предоставляемой человеку возможности свободы действий внутри среды, которые основаны на правилах и поведении среды. Компьютерные виртуальные миры состоят из подсистем: интерфейс пользователя, управления, моделирования среды, объектов и персонажей.

В качестве технических средств реализации виртуальных миров в ГУАП, наряду с современными персональными компьютерами, используются выпускаемые за рубежом средства:

— шлемы-дисплеи, позволяющие видеть подаваемое на управляемые компьютером экраны стереоскопическое изображение виртуального мира, и снабжённые датчиками положения и ориентации головы;

— перчатки, передающие компьютеру информацию о движении руки (её положении, ориентации, углах сгиба пальцев);

— костюмы или их отдельные компоненты (жилеты, штаны, сапоги), передающие компьютеру реакции основных двигательных мышц и формирующие у человека проприоцептивные ощущения, в частности, ощущения усилий;

— устройства отслеживания перемещений и положения человека в пространстве.

ГУАП в дополнение к оборудованию других фирм создал программно-аппаратный комплекс «кибернетический велосипед», который выполняет роль посредника между пользователем и вир-

туальным миром. Он представляет собой велосипед, установленный на специальной платформе, снабжённый целым рядом датчиков, подключённых к интерфейсным блокам компьютера. Физические усилия, затрачиваемые при езде на велосипеде в реальном мире, транспонируются в мир виртуальный, знаменуя соединение активного тела с виртуальным пространством. Педали управляют скоростью, а руль — направлением движения.

Использование кибернетического велосипеда позволяет имитировать поездку на велосипеде по ряду маршрутов. В ГУАП реализованы проекты — «Виртуальный мир университетов Санкт-Петербурга», «Изучение русского языка как иностранного», ведётся работа над проектом «Панорама Второй мировой войны и битвы за Ленинград».

3. Виртуализация и виртуальные машины

Из множества словосочетаний, в состав которых входит прилагательное виртуальный, наиболее близким к точным техническим терминам является понятие *виртуальной машины*, с которым зачастую напрямую и связывается термин виртуализация. В [3] виртуальная машина (ВМ, virtual machine) определялась как совокупность ресурсов, которые эмулируют поведение реальной машины. Концепция виртуальной машины появилась в Кембридже, шт. Массачусетс, в конце 1960-х годов как расширение концепции виртуальной памяти манчестерской вычислительной машины Atlas.

В целом вычислительный процесс определяется в рамках этой концепции содержимым того рабочего пространства памяти, к которому он имеет доступ. При этом процесс не имеет никаких средств для определения того, является ли предоставленный ему ресурс действительно физическим ресурсом этого типа, или же он реализован в результате совместных действий других ресурсов, которые в совокупности приводят к аналогичным изменениям содержимого рабочего пространства процесса. В виртуальной машине ни один процесс не может монопольно использовать никакой ресурс, а все системные ресурсы считаются ресурсами потенциально совместного использования.

Идея виртуальной машины лежит в основе целого ряда коммерческих операционных систем, в частности, систем VM/CMS фирмы IBM и VAX/VM фирмы DEC. Заметим, что приводимые термины и определения были опубликованы более 20 лет назад.

Проблемы виртуализации активно разрабатывались ещё в 70-х годах в архитектурах мэйнфреймов, в первую очередь в фирмах IBM [7] и Siemens [8]. Применение этой технологии диктовалось в первую очередь необходимостью обеспечить поддержку работы различных версий операционных систем (ОС) (в том числе и устаревших версий) в системах коллективного пользования. Появление в 1980-х годах персональных компьютеров, стремительное улучшение их рабочих характеристик, бурный рост их количества, приведший к использованию компьютеров практически во всех областях, на некоторое время пригасили интерес к виртуализации. Создалась иллюзия, что проще для каждой области применения, связанной с одной ОС, иметь свой компьютер. Однако проявилась определённая периодичность в проявлении интереса и внимания к технологии виртуализации (подтверждение верности теории диалектического «развития по спирали»!).

В последние годы ВМ снова набирают популярность, поскольку на смену мэйнфреймам пришли серверы и серверные комплексы, обслуживающие большие группы потребителей. Стремительный рост числа пользователей информационными технологиями, происходящий одновременно с непрекращающимся ростом производительности современных компьютеров, привёл к возобновлению интереса к проблеме виртуализации. Под *виртуализацией* при этом понимается технология, которая позволяет разделить один физический сервер на несколько виртуальных машин (Virtual Machines), на каждой из которых может быть создана своя виртуальная среда, имитирующая для пользователя полную среду вычислительной системы со своей операционной системой. Такой подход заменяет традиционный подход, при котором каждое бизнес-приложение обслуживается одним сервером.

Технология виртуализации корпоративных серверов обеспечивает возможность гибко и надёжно консолидировать использование нескольких разных операционных систем и базирующихся на них приложений на одной аппаратной платформе. При этом увеличивается степень загрузки сервера, тем самым повышается эффективность его использования, упрощается информационная инфраструктура, снижаются операционные расходы. Такие серверные комплексы могут эффективно обслуживать большие группы потребителей.

Виртуализация позволяет также создавать единую логическую среду, в которой создаются виртуальные машины, базирующуюся на комплексе из нескольких физически разных серверов,

в том числе и с разными подсистемами ввода/вывода, тем самым обеспечивается решение проблемы масштабируемости используемых ресурсов. Администраторы такого виртуального центра данных смогут с единой консоли быстро вводить в действие ВМ и управлять большим количеством виртуальных машин, выполняющихся на многих физических серверах. Уйдёт в прошлое взгляд на компьютер как на средство предоставления конкретных услуг. Администраторы будут рассматривать компьютеры просто как часть пула универсальных аппаратных ресурсов [9], пригодных для решения разнообразных возникающих задач.

Виртуализация помогает обеспечить наследование созданного программного обеспечения при переходе на новую платформу — оно может переноситься вместе с операционной системой в качестве виртуальной машины на новую платформу. Выделение отдельной виртуальной машины позволяет проводить разработку и тестирование новых программных систем без приобретения новых аппаратных средств, на имеющейся серверной платформе, без опасений вызвать сбой всей системы [10]. Для каждой новой цели может создаваться собственный набор виртуальных испытательных машин. ВМ являются чрезвычайно удобным средством также и для демонстрации разрабатываемого программного обеспечения — оно может загружаться в ВМ со своей операционной средой.

Виртуальные машины могут эффективно использоваться в качестве стандартных блоков для построения систем с высоким уровнем защиты. На базе технологии виртуализации можно создавать компьютерные среды с различными категориями защиты, тем самым решая проблему обеспечения информационной безопасности и надёжности. Именно эти функции становятся даже более важными, чем организация многозадачности, для чего виртуализация когда-то была задумана.

В последние годы был выполнен ряд успешных проектов, были оформлены патенты на решения в области виртуализации такими компаниями, как VMware, Connectix, IBM, Transitive, Hewlett-Packard, Bull NH Information Systems, Xilinx, Transmeta, Eagle Design Automation, Mentor Graphics Corporation, Intel Corporation, International Meta Systems, российской компанией SWsoft, резко возросло число публикаций в этой области. Современные технологии виртуализации привлекли внимание и отечественной компьютерной прессы — достаточно детальные обзоры современных технологий виртуализации опубликованы

в [11, 12], несколько ранее обзор с упором на рассмотрение проблемы эффективной реализации виртуализации был опубликован в [13].

При реализации виртуальных машин в качестве центрального используется понятие монитора виртуальных машин (МВМ), возникшее ещё в конце 1960-х годов. *Монитор виртуальных машин* (*Virtual Machine Monitor*) — это программный уровень абстракции, который полностью или частично эмулирует аппаратные средства вычислительной машины. Под вычислительной машиной далее понимается не только процессор, но и доступные ему устройства ввода-вывода. Абстракция, созданная МВМ, называется *виртуальной машиной*. МВМ управляет реальными ресурсами вычислительной машины и экспортирует их виртуальным машинам. Рассматривая МВМ, приходится иметь дело с двумя архитектурами:

— *host-архитектурой*, то есть архитектурой реальной вычислительной машины (host-машины), на которой работает сам МВМ. Эта архитектура использует свой набор инструкций (*implementation instruction set architecture*, I-ISA);

— *виртуальной архитектурой*, то есть архитектурой виртуальных машин, которые поддерживает МВМ. Эта архитектура использует свой набор инструкций (*virtual instruction set architecture*, V-ISA).

При реализации МВМ возможны два противоположных подхода. Первый подход опирается на программную или микропрограммную интерпретацию всех инструкций из V-ISA, выполняемых в виртуальной машине. Второй подход исходит из того, что I-ISA и V-ISA для пользовательских приложений совпадают, поэтому инструкции из V-ISA можно выполнять непосредственно на host-машине. Однако режим работы виртуальной машины должен быть таков, чтобы те инструкции V-ISA, с которыми так поступить нельзя, вызывали прерывание, обрабатываемое МВМ, который программно интерпретирует эти инструкции.

МВМ можно разделить на два больших класса:

— МВМ, у которых host-архитектура и виртуальная архитектура не совпадают, причём I-ISA и V-ISA отличаются даже на уровне пользовательских приложений;

— МВМ, у которых host-архитектура и виртуальная архитектура совпадают или в некотором смысле близки (например, I-ISA и V-ISA для пользовательских приложений совпадают).

МВМ класса 1 должен обеспечить такой режим работы host-машины, при котором каждая инструкция из V-ISA, выполняемая в виртуальной машине, программно или микропрограммно интерпретируется. Эта интерпретация может быть достаточно эффективной, если у host-машины имеются средства поддержки, ускоряющие требуемую интерпретацию, или крайне неэффективной, если таких средств нет. Примером могут служить микропроцессоры alpha фирмы DEC, в которых был предусмотрен режим интерпретации инструкций IA32 (Intel) и VAX (правда, только непривилегированного уровня). В таком режиме микропроцессор просто выполнял прямую передачу управления по нужному адресу в программу, находящуюся в памяти микрокода (она называется PAL). Соответствующий адрес зависит от кода операции эмулируемой инструкции. Микрокод эмулирует выполнение инструкции, используя «родные» для alpha RISC-инструкции. Режим интерпретации IA32 применялся, когда на alpha-машине работала Windows/NT (в основном для её приложений). Этот режим за счёт аппаратной поддержки был достаточно эффективен.

У МВМ класса 2 host-архитектура и архитектура виртуальной машины совпадают или близки. Как программный уровень МВМ может опираться непосредственно на интерфейс с аппаратными средствами, который предоставляет ему host-архитектура, или же использовать промежуточный программный уровень базовой операционной системы (*host operating system*, host-OS), который, взаимодействуя сам с аппаратными средствами, предоставляет МВМ интерфейс более высокого уровня.

Ещё в период первой волны работ в области технологии построения виртуальных машин Гольдберг в [14] сформулировал ключевые возможности аппаратуры для машин третьего поколения, без которых невозможна эффективная реализация МВМ класса 2 (*необходимые условия эффективной виртуализации*):

— наличие у процессора двух режимов выполнения инструкций: пользовательского и привилегированного;

— наличие метода, с помощью которого непривилегированная программа могла бы вызвать системную привилегированную подпрограмму;

— наличие механизма назначения памяти или защиты памяти, например, сегментирование или динамическое преобразование адреса;

— асинхронные прерывания, которые позволяют системе ввода-вывода взаимодействовать с процессором.

При этом под эффективной понимается такая реализация, при которой статистически почти все инструкции виртуальной машины выполняются непосредственно host-машиной, то есть время, которое тратит МВМ на имитацию отдельных инструкций, пренебрежимо мало.

Дополнив набор условий, сформулированный Гольдбергом, условием строгой изоляции виртуальных машин, можно сформулировать следующие 3 требования, которым должна удовлетворять host-архитектура, чтобы на ней была возможной реализация МВМ [13].

Требование 1. Выполнение непривилегированных инструкций должно быть строго одинаковым в привилегированном и пользовательском режимах. Например, процессор не должен допускать использование дополнительных разрядов в коде операции или адресной части инструкции, когда он выполняет её в привилегированном режиме.

Требование 2. В host-архитектуре должен быть предусмотрен способ, гарантирующий изоляцию активной виртуальной машины, то есть защищающий от неё сам МВМ и другие виртуальные машины. Этим способом может быть, например, режим работы процессора (привилегированный и пользовательский) или алгоритм трансляции адреса.

Требование 3. В host-архитектуре должен быть предусмотрен способ, который позволяет автоматически оповестить МВМ о попытке выполнить в виртуальной машине инструкцию доступа к ресурсу (*sensitive instruction*), управлением которого заведует МВМ. Это позволяет в МВМ имитировать выполнение этой инструкции.

Упомянутые инструкции доступа к ресурсам включают:

Требование 3А. Инструкции, с помощью которых можно изменить или прочесть состояние виртуальной машины или host-машины.

Требование 3В. Инструкции, с помощью которых можно изменить или прочесть содержимое управляющих регистров таких, как таймер, регистр прерываний и т. д.

Требование 3С. Инструкции, которые используются для управления механизмами защиты памяти, трансляции адреса и т. д. Эти инструкции могли бы разрешить виртуальной машине

доступ к любой ячейке памяти, а не только к памяти этой машины.

Требование 3D. Все инструкции ввода-вывода.

4. Аппаратные средства поддержки виртуализации

Наиболее последовательно проводит техническую политику поддержки виртуализации (вне зависимости от моды) фирма IBM. Решение, которое применяет эта компания, — поддержка MBM на аппаратном уровне. Такое решение предложено IBM ещё в 70-х годах (архитектура System/370, VMM VM/370), она использует его до сих пор (архитектура z/Architecture, VMM z/VM [15]), обеспечивая реализацию высокопроизводительного MBM.

В настоящее время преобладающую долю рынка серверов представляют компьютеры, построенные на базе микропроцессоров компаний Intel и AMD. Изначально в микропроцессорах этого семейства не предусматривались режимы работы, при которых возможна эффективная виртуализация. Поэтому для построения MBM приходилось пользоваться пошаговой эмуляцией инструкций BM или же выполнять бинарную трансляцию исполняемого кода, что резко снижало эффективность работы всей системы. Но уже в конце 2004 года Intel объявила о намерении реализовать в своих продуктах поддержку технологий виртуализации.

В начале 2005 года появились общедоступные спецификации расширения для архитектуры IA-32 под первоначальным названием Vanderpool Technology [16], а также для архитектуры Itanium называемые VT-I (Vanderpool Technology for the Intel Itanium architecture [17]), позволяющие эффективно реализовывать MBM и тем самым обеспечивать возможность одновременной работы нескольких операционных систем и сред на одной аппаратной платформе. В ближайшее время фирма Intel предполагает поддерживать Virtualization Technology (VT) (так её стали называть) на всех вновь выпускаемых семействах микропроцессоров.

В мае 2005 года и компания Advanced Micro Devices (AMD) опубликовала полную спецификации технологии расширения архитектуры AMD64 (под названием Pacifica), которая позволяет реализовать монитор виртуальной машины [18]. Эти расширения в значительной степени совпадают с реализованными фирмой Intel в VT. Планировалось, что чипы с реализованной технологией Pacifica появятся в 2006 году.

5. Программные средства поддержки виртуализации

Среди программных продуктов, поддерживающих виртуализацию, пожалуй, наиболее широко распространённым являются продукты компании VMware. Технология виртуализации для персональных компьютеров, разработанная компанией VMware, базируется на полной эмуляции оборудования на уровне программного обеспечения. MBM VMware позволяет установить несколько операционных систем на одном компьютере и работать с ними без перезагрузки. MBM VMware запускается как программа под управлением host-OS и создаёт набор виртуальных машин, в каждой из которых может быть запущена своя guest-OS.

Первый продукт, VMware Workstation, предназначенный для запуска нескольких операционных систем на рабочей станции, был выпущен компанией в 1999 году. В 2001 году компания VMware выпустила два серверных продукта: VMware GSX Server — для серверов рабочих групп и VMware ESX Server — для серверов уровня предприятия. В 2003 году компания VMware была куплена компанией EMC, производителем систем хранения и одним из крупнейших в мире поставщиков на рынке аппаратных средств. VMware позволяет хорошо решать такие задачи, как: совместный запуск guest-OS Windows и Linux на одном компьютере (как Windows под Linux, так и Linux под Windows); использование виртуальных компьютеров в сложной сетевой инфраструктуре; объединение серверов и построение типовых решений виртуализации для предприятий.

Компания VMware разработала одну из первых реализаций VMM для платформы IA32. Для сокращения количества эмулируемых инструкций и повышения производительности MBM VMware представляет собой гибридный MBM, в котором для достижения максимальной скорости работы используется квази-эмуляция. Квази-эмуляция включает как непосредственную эмуляцию, так и динамическую бинарную компиляцию, причём применяется техника расстановки точек прерывания. Кроме того, VMware учитывает особенности каждой guest-OS и использует для виртуализации адаптированный вариант квази-эмуляции. На архитектурах IA32, IA64 и AMD64, исходно спроектированных без учёта требований эффективной виртуализации, накладные расходы для приложений, связанных с интенсивным вводом-выводом, находятся для MBM VMware Workstation на

уровне десятков процентов. Аппаратная поддержка, реализованная в расширениях Intel VT и AMD Pacifica, должна дать существенное улучшение производительности для технологии, используемой VMware.

Второй компанией в области виртуализации после VMware была компания Connectix, технологию Virtual PC которой в 2003 году купила компания Microsoft. В 2004 году Microsoft выпустила продукт MS Virtual PC 2004 для персональных компьютеров, а затем и MS Virtual Server 2005 для серверов. Изначально MBM Virtual PC проектировался для полной эмуляции guest-архитектуры IA32 на host-архитектуре Apple Macintosh. Поскольку такая эмуляция guest-кода замедляет скорость работы в 100–1000 раз, компания Connectix использовала технику динамической бинарной компиляции для ускорения эмуляции. В Virtual PC для Windows технология виртуализации была усовершенствована за счёт введения элементов квази-эмуляции. Для guest-OS из семейства Windows в Virtual PC используется дополнительное ускорение с помощью перехвата ряда вызовов Win32 API. Только после выхода Virtual PC 5.0 технология виртуализации Connectix догнала по скорости и эффективности своего основного конкурента — VMware. В принципе решения Connectix и VMware достаточно близки, но в настоящее время продукты VMware выглядят более отработанными.

Следует упомянуть также проект с открытым кодом Xen фирмы XenSource — это MBM для компьютеров с архитектурой IA32, разработанный в Кембриджском университете. В MBM Xen применена технология паравиртуализации, требующая предварительной модификации ядер guest-OS, которые должны работать в виртуальных машинах. В настоящий момент силами проектной команды Xen полностью портирован Linux с ядром 2.4 и при участии Microsoft Research завершается портирование Windows XP. Также планируется выполнить портирование FreeBSD 4.8. Основное преимущество технологии Xen — практически полное отсутствие накладных расходов на виртуализацию и высокая производительность по сравнению с другими MBM. Основным недостатком является необходимость выполнять портирование для каждой новой версии guest-OS, что возможно только в случае доступности её исходных текстов и лицензионного соглашения на проведение изменений.

Имеется и другой метод виртуализации — так называемая «высокоуровневая виртуализация», при которой создаются однородные вычислительные пространства на базе одного ядра OS — виртуальные среды (Virtual Environment, VE). Пионером этого подхода является российская компания SWsoft с продуктом Virtuozzo [11, 12]. С помощью Virtuozzo на базе одного ядра OS (имеются реализации для Linux и Windows) можно создать огромное количество VE, каждая из которых имеет своё уникальное изолированное окружение — свои файлы и другие ресурсы, в том числе системные, свои сервисы, свои системные способы связи с внешним миром.

Пользователи могут инсталлировать в VE свои приложения, изменять конфигурационные файлы. В то же время оператор системы имеет возможность выполнять операции над ядром OS одновременно для множества (тысяч) VE. Он может, например, быстро устанавливать приложение во множество VE, динамически управлять распределением ресурсов системы между VE, перемещать отдельные VE между физическими серверами. Подход, опирающийся на построение VE, требует для реализации существенно меньше ресурсов, чем подход с использованием виртуальных машин (как у VMware), но имеет существенное ограничение, поддерживая работу лишь в однородной среде, то есть все запущенные на машине VE работают с одной и той же версией OS.

Применение высокоуровневой виртуализации с использованием VE может быть эффективным для поддержки высокомасштабируемых серверных решений с большим количеством приложений, работающих на однотипных OS, тогда как полная виртуализация (с виртуальными машинами) подходит, прежде всего, для целей разработки, отладки, тестирования.

6. Использование технологии виртуализации в реализации прозрачной отказоустойчивости

Примером и демонстрацией возможности использования технологии виртуализации является решение задачи обеспечения прозрачной отказоустойчивости [19]. В настоящее время существует много областей применения вычислительной техники, требующих длительной безотказной работы. Это прежде всего касается *серверов приложений*, представляющих собой как правило кластеры в сети, предоставляющие своим клиентам определённые ресурсы или услуги, разделяя между ними свои ресурсы.

Кластером мы здесь считаем параллельную или распределённую систему, состоящую из нескольких компьютеров (узлов кластера), связанных необходимыми коммуникационными каналами и используемых как единый унифицированный компьютерный ресурс.

Под *прозрачной отказоустойчивостью* (Transparent Fault Tolerance, TFT) сервера приложений понимается такое его поведение при возникновении аппаратных или программных отказов либо отказов в сети, при котором:

— отказ не вызывает потери или искажения данных, находящихся в базе данных сервера;

— сервер продолжает нормально функционировать, несмотря на имевшие место отказы;

— клиенты сервера «не замечают» произошедших отказов. Единственным допустимым отклонением сервера от нормального поведения с точки зрения клиента является увеличение времени обслуживания (например, на несколько секунд или десятков секунд).

Реализация программной отказоустойчивости серверов приложений является сложной технической задачей. Её решение связано с использованием целого ряда специальных программных технологий, в частности и технологии виртуализации операционной среды, в которой работает серверное приложение. В разработанной системе эта технология реализована в виде отдельного VIRT-компонента. Опишем кратко его функционирование.

VIRT-компонент является неотъемлемой частью компонента создания контрольных точек. Он выполняет виртуализацию операционной среды, в которой работает приложение. Если VIRT-компонент не используется, приложение при создании некоторых ресурсов получает от среды, в которой оно работает, идентификаторы, далее называемые *системными*. Эти идентификаторы среда сама присваивает ресурсам, открываемым приложением.

Далее приложение при обращениях к таким ресурсам, им созданным, ссылается на ранее полученные системные идентификаторы. При переносе приложения в другой экземпляр среды те же ресурсы могут получить в новой среде другие системные идентификаторы. Чтобы сохранить работоспособность приложения после переноса, поступают так. С каждым подобным ресурсом связывается виртуальный идентификатор (*псевдоним*),

который получает приложение, однако операционная среда попрежнему продолжает работать с системным идентификатором, который она назначила. Поэтому при восстановлении работы приложения в другом экземпляре среды необходимо, сохранив псевдонимы, изменить соответствующие им системные идентификаторы. Таким образом, VIRT-компонент предполагает разделение работ, которые выполняет узел, на 3 уровня:

— уровень приложений, функционирующих в виртуальной среде;

— промежуточный уровень, который собственно и представляет собой VIRT-компонент;

— уровень стандартного ядра операционной системы.

VIRT-компонент должен делать следующие работы:

— при открытии (создании) ресурса получить от операционной среды системный идентификатор, сопоставить ему псевдоним и этот псевдоним передать приложению;

— при любом обращении приложения к открытому ресурсу (такое обращение, естественно, ссылается на псевдоним ресурса), передать это обращение в среду, заменив псевдоним системным идентификатором;

— при любом обращении среды к приложению, связанному с открытым ресурсом, выполнить обратную подстановку;

— при восстановлении приложений в другом экземпляре среды при открытии всех ресурсов, которые были открыты в оригинальной среде (этим занимается компонент создания контрольных точек), сохранить их псевдонимы, но сопоставить этим псевдонимам новые системные идентификаторы.

7. Заключение

Современный мир уже невозможно представить себе без компьютеров, используемых буквально везде. Компьютеры сделали возможной техническую реализацию идеи виртуализации, т. е. позволили перейти от чисто философских категорий к практическим применениям. Сложные структуры современного мира компьютерных систем состоят из множества тесно взаимодействующих программных и аппаратных компонентов. В этом мире виртуализация выступает в роли одной из *технологий* организации их взаимосвязи, а именно, *информационной технологии*.

Добавление уровня виртуализации между оборудованием и программным обеспечением фактически строит компьютер, на

котором несовместимые подсистемы могут работать вместе. Кроме того, виртуальная репликация операционной среды позволяет гибко и эффективно задействовать все аппаратные и программные ресурсы.

Сегодня виртуальные машины широко используются для организации взаимодействия между оборудованием, операционными системами и прикладными программами. Учитывая стойкую приверженность компьютерной отрасли стандартам и консолидации, можно ожидать, что все новые архитектуры, операционные системы и языки программирования будут разрабатываться на основе технологий виртуализации и с учётом требований их эффективной реализации.

Таким образом, виртуализация как информационная технология является сегодня одним из важных и перспективных направлений дальнейшего развития систем и средств информатики.

Список литературы

1. Новейший словарь иностранных слов и выражений. — М.: АСТ, 2002.
2. *Воройский Ф.С.* Информатика. Новый систематизированный толковый словарь-справочник (Введение в современные информационные и телекоммуникационные технологии в терминах и фактах). 3-е изд. — М.: Физматлит, 2003.
3. Толковый словарь по вычислительным системам / Под ред. В. Иллингуорта, Э.Л. Глейзера, И.К. Пайла / Пер. с английского. — М.: Машиностроение, 1989. (Oxford University Press, second edition, Oxford, N. Y., Tokyo, 1986.)
4. *Briukhov D.O.*, *Kalinichenko L.A.*, *Zakharov V.N. et al.* Information Infrastructure of the Russian Virtual Observatory (RVO). 2nd ed. — М.: IPI RAN, 2005.
5. *Филинов Е.Н.* Проблемы информатики и информационные технологии // Системы и средства информатики. Вып. 10. — М.: Физматлит, 2000. — С. 11–42.
6. *Игнатьев М.Б.*, *Никитин А.А.*, *Никитин А.В.*, *Решетников Н.Н.* Архитектура виртуальных миров. — СПб., 2005.
7. *Gum P.H.* System/370 Extended Architecture: Facilities for virtual machines // IBM Journal of Research and Development. 1983. V. 27. № 6.
8. Siemens System 4004. Zentralienheiten 4004/150. Beshreibung und Befehlsliste. — Munchen. 70, 1972.

9. Мендель Розенблюм, Тэл Гарфинкель. Мониторы виртуальных машин: современность и тенденции // Открытые системы. 2005. № 05–06.
10. *Shiveley R.* Enhanced Virtualization on Intel Architecture based Servers // Technology @ Intel Magazine. April 2005. P. 3–9.
11. *Колесов А.* Пришло время виртуализации // PC WEEK/RE. 18 июля 2006. № 26. С. 18–20.
12. *Тормасов А.*, *Колесов А.* Виртуализация сегодня: задачи, проблемы, технологии, решения // PC WEEK/RE. 25 июля 2006. № 27. С. 25–27.
13. *Захаров В.Н.* Вопросы эффективной реализации технологии виртуальных машин // Наукоёмкие технологии. 2006. Т. 7, № 2. С. 51–67.
14. *Goldberg R.* Architectural Principles for Virtual Computer Systems. — Ph.D. thesis. — Harvard University, Cambridge, MA, 1972.
15. z/VM built on IBM Virtualization Technology. General Information. Version 4 Release 4.0 `http://www.vm.ibm.com/pubs/pdf/HCSFA60.pdf`.
16. Intel Vanderpool Technology for IA-32 Processors (VT-x) Preliminary Specification, Intel Order Number C. 97063-001, January 2005.
17. Intel Virtualization Technology Specification for the Intel Itanium Architecture (VT-i), Revision 2.0, April 2005. `ftp://download.intel.com/technology/computing/vptech/30594202.pdf`.
18. AMD64 Virtualization Codenamed «Pacifica» Technology. Secure Virtual Machine Architecture Reference Manual. Advanced Micro Devices. 33047 — Rev. 3.01 — May 2005 `http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/33047.pdf`
19. *Захаров В.Н.*, *Козмидиади В.А.* Реализация отказоустойчивости серверов приложений // Наукоёмкие технологии. 2006. № 6. Т. 7. С. 56–62.