

УДК 004.2; 004.77.

Ю.С. Яковлев

О виртуализации рабочих мест и их компонентов при построении компьютерных систем и сетей

Рассмотрен круг вопросов, связанных с виртуализацией рабочих мест и их компонентов – преимущества виртуального рабочего места перед обычными системами, основные этапы построения виртуальных машин и рабочих мест, а также даны понятия виртуального процессора, виртуальной памяти и виртуальных средств ввода/вывода. Приведены примеры структурной схемы виртуального офиса и виртуальной компьютерной системы, а также пример основных сред программ виртуальных машин.

Ключевые слова: виртуальное рабочее место, виртуальный процессор, виртуальная память, виртуальный интерфейс, этапы построения виртуальных машин.

Розглянуто питання, пов'язані з віртуалізацією робочих місць та їх компонентів – переваги віртуального робочого місця перед звичайними системами, основні етапи побудови віртуальних машин і робочих місць, а також дано поняття віртуального процесора, віртуальної пам'яті та віртуальних засобів вводу/виводу. Подано приклади структурної схеми віртуального офісу, віртуальної комп'ютерної системи, а також приклад основних середовищ програм віртуальних машин.

Ключові слова: віртуальне робоче місце, віртуальний процесор, віртуальна пам'ять, віртуальний інтерфейс, етапи побудови віртуальних машин.

Введение. Виртуализация стала важным инструментом разработки компьютерных систем, а виртуальные машины используются в самых разных областях – от операционных систем до языков программирования и архитектуры процессоров [1]. Освобождая разработчиков и пользователей от ресурсных ограничений и недостатков интерфейса, виртуальные машины снижают уязвимость системы, повышают интероперабельность программного обеспечения и эксплуатационную гибкость аппаратной платформы. Виртуальные машины разрабатываются множеством специалистов, преследующих самые разные цели, и в этой области существует не так уж много общепринятых концепций. Сегодня виртуальные машины переживают второе рождение. Их возрождение объясняется двумя основными причинами:

- появлением большого количества разных операционных систем (ОС), предъявляющих специфические требования к параметрам используемых аппаратных компонентов компьютера;
- большими затратами на администрирование и сложностью обслуживания компьюте-

ров, на которых установлено несколько различных операционных систем, в том числе в плане обеспечения требуемой надежности и безопасности работы.

Постановка задачи

Актуальность статьи продиктована, прежде всего, целым рядом преимуществ виртуального рабочего места (стола) *VDI (virtual desktop infrastructure)* перед обычными системами, предоставляющими пользователям *VDI* как инфраструктуру, так и называемых виртуальных рабочих станций (виртуальных машин). Суть данной технологии заключается в полном замещении физических ЭВМ массивом виртуальных компьютеров (установкой виртуальных машин), используемых для цифровой обработки данных (ЦОД). К таким преимуществам, в частности относятся:

- возможность установки на одном компьютере нескольких ОС без необходимости соответствующего конфигурирования физических жестких дисков;
- работа с несколькими ОС одновременно с возможностью динамического переключения между ними без перезагрузки системы;

- сокращение времени изменения состава установленных ОС;

- изоляция реального оборудования от нежелательного влияния программного обеспечения, работающего в среде виртуальной машины;

- возможность моделирования вычислительной сети на единственном автономном компьютере.

Все это приводит к повышению безопасности, так как подключение к серверам *VDI* осуществляется через защищенный канал, а корпоративные данные никогда не покидают пределы ЦОД предприятия [2]. Кроме того, *VDI* позволяет реализовать концепции «*any place, any time, any device*» (возможность работать из любого места, в любое время и с любого устройства) и есть первым шагом к *BYOD (Bring Your Own Device)*, а также появляется возможность работать через медленные каналы связи, несмотря на то, что некоторые компании считают, что *VDI* не применима к их системе из-за наличия распределенных офисов и медленных каналов связи между ними. И, наконец, *VDI* – это снижение затрат на централизованное обновление программного обеспечения (ПО) и миграцию на новые ОС.

Внедрение инфраструктуры виртуальных рабочих мест приобретает характер доминирующей тенденции в модернизации инфраструктур корпоративных информационных технологий (ИТ). На основе современных ИТ создаются новые системы существующих организаций и принципиально новые типы организаций – виртуальные. Поэтому полезно рассмотреть понятие виртуализации применительно к компьютерным системам и их компонентам, а также некоторые способы построения архитектуры виртуальных машин и их компонентов в единой перспективе.

Проблемы виртуальности компьютерных систем

Основными платформами, обеспечивающими сервис виртуальных рабочих мест, считаются [3]: *XenDesktop* от компании *Citrix*, *Horizon View* от компании *VMware*, *Virtual Desktop Interface* от *Microsoft*, *vWorkspace* от *Quest* и др.

Один из основных критериев при выборе платформы – объем требуемых ресурсов для ее реализации. Ключевыми параметрами являются ЦП (центральный процессор), оперативное запоминающее устройство (ОЗУ), *i/o* (дисковая подсистема ввода/вывода) и графическая система. Примерная конфигурация виртуальной машины включает: операционную систему – *Windows 7 x64* и др.; оперативную память – 1.5 Гб; процессор – 1 *vCPU*; жесткий диск – 25 Гб; программное обеспечение – *Microsoft Office*, *Mozilla Firefox* и др. Появление многоядерных процессоров даст мощный дополнительный импульс массовому внедрению технологий виртуализации на различных уровнях. В настоящее время даже в рамках одноядерных и однопроцессорных систем быстро развиваются технологии и средства виртуализации. Виртуализация охватывает широкий круг областей применения, среди которых можно обозначить виртуальные:

- корпорации как высшую форму сети делового сотрудничества и развития информационных технологий;

- организации, предприятия и виртуальные офисы;

- лаборатории;

- рабочие места (виртуальные столы);

- ЭВМ, системы и сети;

- компоненты виртуальных ЭВМ.

Далее будут кратко рассмотрены виртуальные рабочие места и их виртуальные компоненты.

Виртуальные рабочие места. С учетом ИТ, конечно же, первые ассоциации будут с рабочим компьютером и рабочим столом как физическим, так и *виртуальным*, который вам предлагает операционная система. Виртуальный рабочий стол системы – пример того, как виртуализируется рабочее место. На рабочем столе находятся текущие документы, программы, с которыми вы работаете в данный момент, а многозадачность ОС и удобная файловая система упрощают вашу работу. Виртуализация рабочего места сегодня – это перенос вашего компьютера (и всех его приложений) в *виртуальное* состоя-

ние. Виртуализация делится на две больших категории, кардинально отличающиеся по способу взаимодействия пользователя с сервером: *Server-Based Computing* – сервер в облаке и *Central-Based Computing* – сервер в офисе.

При этом в качестве инструментов доступа используют две основные технологии [4]: виртуализация рабочих мест, известная как *VDI* (*virtual desktop infrastructure*), и удаленный рабочий стол, сокращенно – *RDP* (*remote desktop protocol*) или *RDS* (*remote desktop services*).

VDI – программно-аппаратный комплекс, содержащий сервер с серверной ОС, на котором реализуются образы с клиентскими операционными системами. Часть файлов у образов общая, часть – раздельная, но суть заключается в том, что у каждого из пользователей – свой образ ОС, а доступ к нему осуществляется через тонкие клиенты.

RDP/RDS – программно-аппаратный комплекс с главным отличием: серверная ОС – ваша рабочая среда, при этом все программы запускаются прямо на сервере, и по протоколу *RDP* вы получаете доступ к конкретным приложениям или пользовательскому рабочему столу.

Существует еще один тип виртуализации приложений – *BYOD* («bring your own device»), реализация связанного с покупкой оборудования, на котором вы хотите работать (ноутбук, моноблок и др.).

Для построения виртуальной машины (ВМ) к реальному компьютеру добавляется программное обеспечение, поддерживающее желаемую архитектуру. Таким путем можно обойти проблему совместимости реальных машин и ресурсные ограничения оборудования. В применении к компьютерным системам архитектура означает формальную спецификацию системных интерфейсов, в том числе логические модели ресурсов, управляемых через эти интерфейсы [1]. Три таких интерфейса – особенно важны для построения ВМ: это архитектура системы команд (*ISA*), двоичный интерфейс приложений (*application binary interface, ABI*) и интерфейс прикладного программирования (*application programming interface, API*). При

этом реализация описывает фактическое воплощение архитектуры. Как в оборудовании, так и в программном обеспечении уровни абстракции соответствуют уровням реализации, каждый из которых связан со своим собственным интерфейсом или архитектурой.

Сложные структуры компьютерных систем состоят из множества тесно взаимодействующих программных и аппаратных компонентов. Виртуализация выступает в роли одной из технологий организации их взаимосвязи. Добавляя уровень виртуализации между оборудованием и программным обеспечением, мы получаем машину, на которой несовместимые подсистемы могут работать вместе. Кроме того, виртуальная репликация операционной среды позволяет более гибко и эффективно задействовать аппаратные ресурсы. Таким образом, должно рассматриваться построение виртуальных машин как комплексную дисциплину, связанную с разработкой оборудования, операционных систем и прикладных программ. Ниже приведен пример структуры виртуальной компьютерной системы цифровой обработки данных (ЦОД) *Cisco VXI* компании *Cisco* [5].

Инфраструктура *Cisco VXI* спроектирована в расчете на максимальную гибкость выбора оконечного устройства. *Cisco VXI* основывается на принципе экосистемы с открытым подходом к выбору клиентской части без привязки к конкретной технологии.

Cisco VXI – комплексная архитектура виртуализации. Она позволяет оперативно развертывать настольные рабочие места и улучшает управляемость и безопасность с контролируемостью на уровне виртуальной машины. *Cisco VXI* объединяет мультимедийные и сетевые службы для повышения производительности и оперативности приложений. Модульная архитектура обладает гибкостью адаптации для заказчиков и помогает обеспечить соответствие потребностям отрасли в долгосрочной перспективе.

Центр обработки данных – ядро этих систем, поскольку настольная виртуализация в значительной мере опирается на эффективную централизацию вычислительных мощностей,

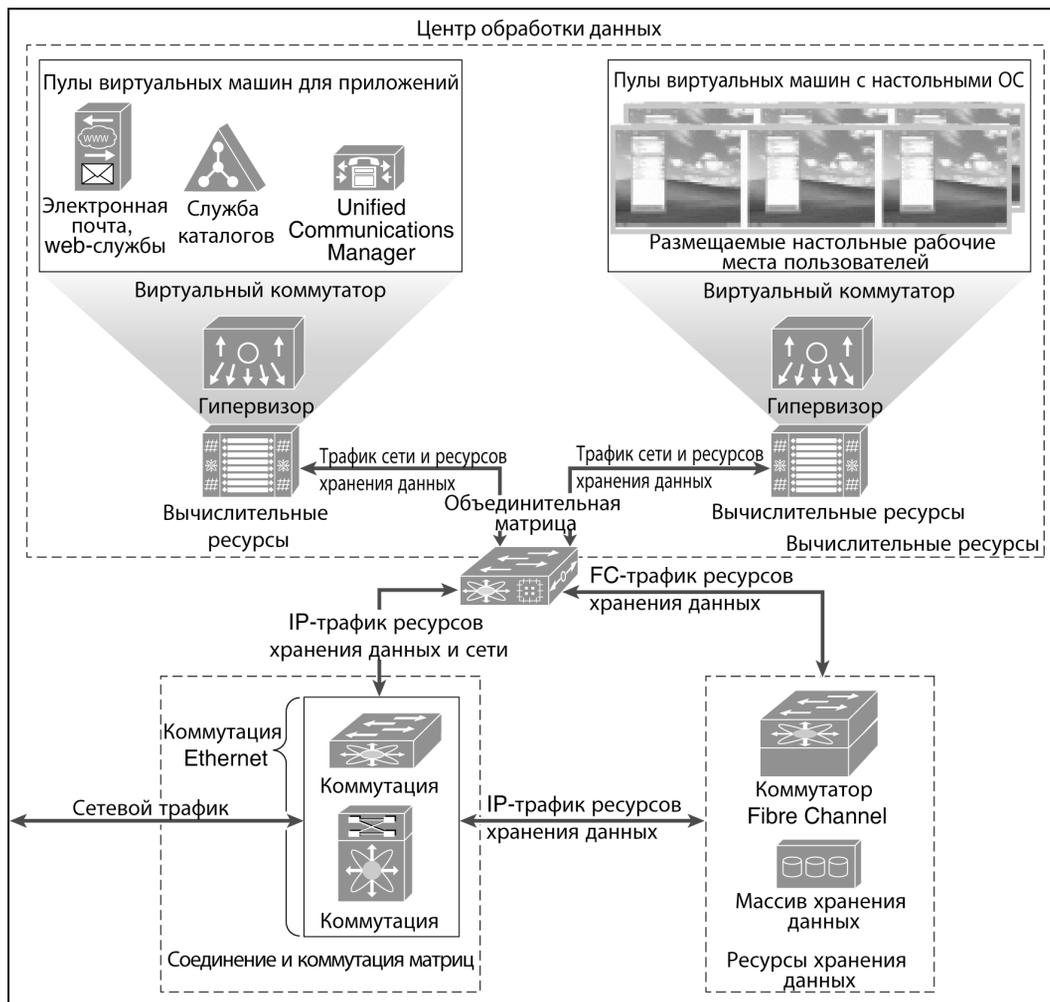


Рис. 1. Структура виртуальной компьютерной системы ЦОД Cisco VXI

приложений и данных. Оптимальный центр обработки данных должен обеспечивать мощную, масштабируемую, интегрированную и управляемую среду для размещения виртуальных настольных систем.

Блок вычислительных мощностей обеспечивает процессорные ресурсы для размещения виртуальных настольных систем и приложений, например *Cisco UCS Manager*. В архитектуре *Cisco VXI* мощные вычислительные ресурсы обеспечиваются средой унифицированных вычислений *Cisco (Unified Computing System™, UCS)*. Архитектура расширенной памяти *UCS* обеспечивает более высокий уровень масштабируемости и плотности виртуальных машин на серверных блейд-модулях, а профили служб позволяют быстро выделять ресурсы сервера, легко объединять ресурсы и управлять политиками.

Среда унифицированных вычислений *Cisco* поддерживает конвергентные сетевые адаптеры (*CNA*), способные обслуживать как сетевые ресурсы хранения данных (*NAS*), так и сети *SAN* на основе *Fibre Channel*. На каждом сервере *Cisco UCS* установлен гипервизор, позволяющий виртуализированным настольным системам и серверам функционировать в качестве независимых виртуальных машин. Гипервизоры упрощают создание, развертывание и эксплуатацию виртуальных машин, обеспечивая получение всеми виртуальными машинами достаточной доли ресурсов центрального процессора, памяти и ресурсов ввода-вывода.

Блок соединительной структуры матриц обслуживает обмен данными блока вычислительных мощностей с сетью и устройствами хранения данных, а также передачу трафика между

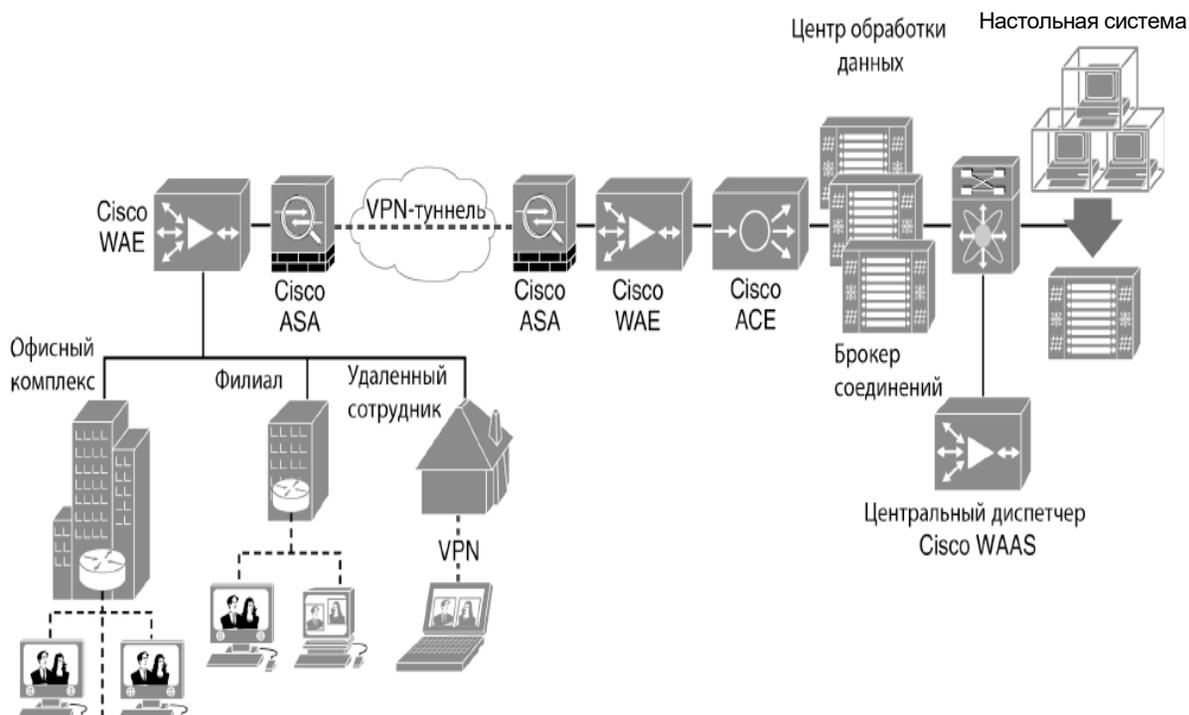


Рис. 2. Детальная структура сети Cisco VXi с поддержкой виртуализации

серверными блейд-модулями. Файловые системы различных виртуальных настольных систем объединяются путем внешних ресурсов хранения данных.

Информация, которую было принято хранить на локальном ПК пользователя, теперь хранится в общем массиве в центре обработки данных. Каждый общий массив поделен на разделы. Хотя эти разделы могут использоваться разными серверами совместно, большинство организаций изолируют виртуальные машины от других виртуализированных задач для укрепления безопасности и выравнивания нагрузок от трафика.

Сеть играет важную роль в виртуализации настольных систем, поскольку конечные пользователи подключаются к виртуализированному центру обработки данных, в том числе и для доступа к приложениям и информации, которые прежде постоянно находились на их ноутбуках. На рис. 2 приведена детальная структура сети Cisco VXi с поддержкой виртуализации [5].

Критический компонент многих комплексов виртуализации настольных систем – брокер соединений. Брокеры соединений выполняют

аутентификацию и переадресуют запросы клиентов конечных пользователей к соответствующим виртуальным настольным системам. Брокеры соединений также выделяют ресурсы для новых виртуальных настольных систем по требованию и ретранслируют реквизиты учетной записи пользователя в размещенную настольную систему.

Виртуализация настольных систем, по своей природе, защищена лучше, поскольку данные и приложения фактически постоянно находятся в центре обработки данных, а не на устройстве конечного пользователя. Виртуализация таким образом обеспечивает защиту от утечек и потерь данных в результате умышленных хищений или аварий. Вместе с тем, настольная виртуализация также поднимает новые вопросы безопасности, потенциально затрагивающие и центр обработки данных, и конечного пользователя.

Виртуальные компоненты. Далее представлены виртуальные компоненты виртуальных компьютерных систем и основные реализуемые ими функции.

В работе [6] представлено решение для создания инфраструктуры виртуальных рабочих

мест (столов) на базе открытого программного обеспечения. Оно отличается всем необходимым функционалом при использовании традиционных ПК, тонких и нулевых клиентов.

Преимущество предложенного решения – простота и прозрачность интеграции с открытой облачной платформой *OpenStack*. Это создает реальные условия для дальнейшей минимизации совокупной стоимости владения корпоративной информационной инфраструктурой университетов и других организаций.

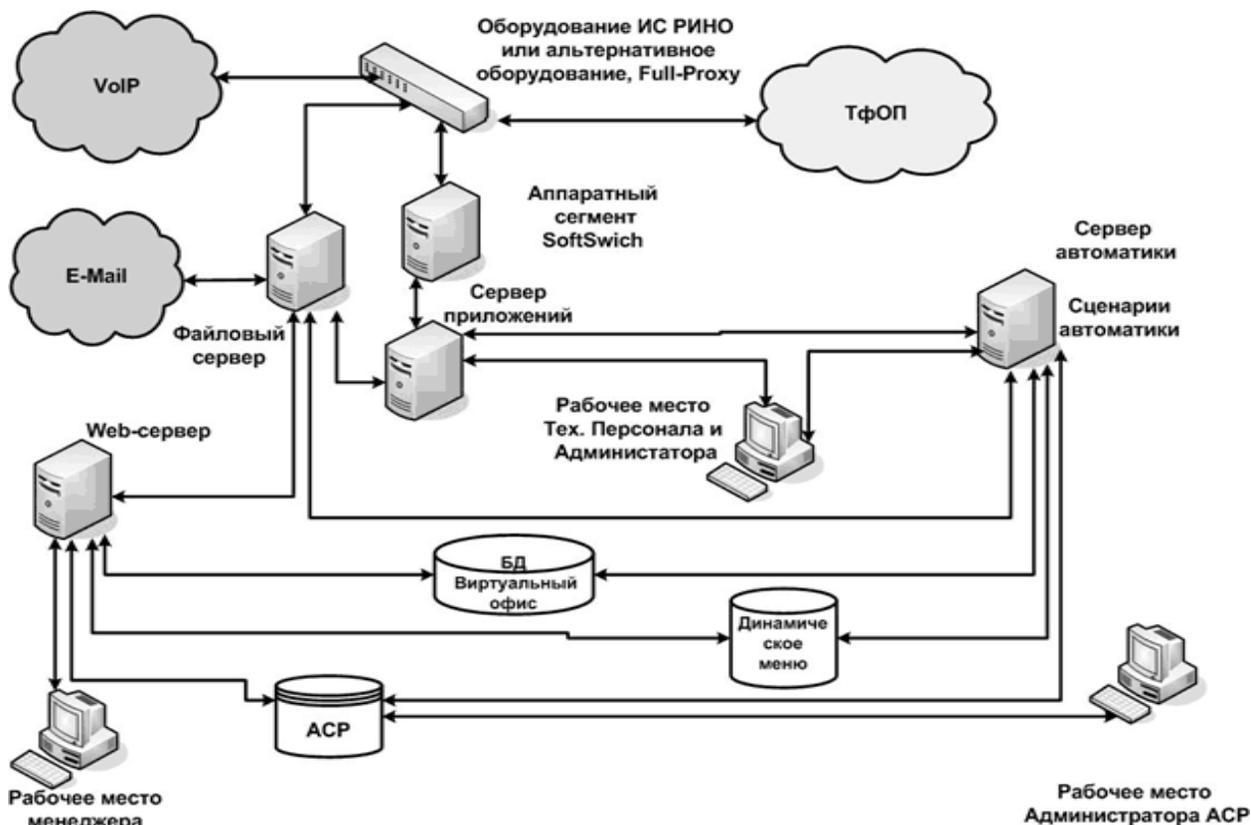
В работе [7] показана структурная схема варианта виртуального офиса (рис. 3).

В качестве аппаратных средств такого офиса может быть использовано оборудование «ИС РИНО» или других фирм (например, *Cisco*), а программные средства *Web*-интерфейса устанавливаются на отдельной серверной платформе.

Особенностью данной схемы офиса есть то, что предусмотрен необходимый набор мер против попыток несанкционированного доступа и злонамеренных действий.

Средства администрирования комплекса предоставляют администратору следующие возможности:

- внесение изменений и добавления/удаления владельцев услуг;
- создание сценариев обслуживания на основе имеющихся шаблонов (имеющихся сценариев владельцев услуг);
- средства работы со сценариями;
- средства для загрузки, удаления, редактирования голосовых сообщений, преобразования к требуемому формату;
- средства безопасного удаленного доступа для администрирования;
- средства детальной трассировки наблюдаемого вызова;
- аварийная сигнализация о неполадках;
- средства администрирования *Web*-интерфейса;
- средства формирования отчетов;
- средства для экспорта данных тарификации;



Примечание: обозначение компонентов схемы и реализуемые ими функции пояснены на рис.3.

Рис. 3. Структурная схема варианта виртуального офиса

- средства для настройки тарификации.

В базе данных (БД) сохраняется информация о том, какие услуги использовались в течение данного вызова: весь *DTMF*-набор, который набирали в отдельности Заказчик и Клиент. В БД сохраняется информация о дате/времени начала и продолжительности каждой фазы установления соединения, время прослушивания сообщения автоинформатора, процесс установления соединения, фаза разговора или время прослушивания сообщения автоинформатора.

Подсистема тарификации включает в себя программные средства автоматического формирования стандартных журналов тарификации и их отправки файлами посредством *FTP*-сервиса через устанавливаемый администратором интервал времени по заданным *IP*-адресам.

Программные среды (системы) виртуальных машин

Часто программные продукты этого класса называют эмуляторами операционных систем [8]. На сегодняшний день двумя самыми популярными программными средами виртуальных машин есть *Microsoft Virtual PC* (www.microsoft.com) и группа продуктов *VMware* (www.vmware.com) компании *VMWare Inc.* Суть ПО системы виртуальных машин состоит в том, что на компьютере, работающем под управлением основной (базовой) ОС, создаются один или несколько виртуальных компьютеров, на каждом из которых можно запустить собственную гостевую ОС. Так *VMWare Workstation* позволяет запускать несколько ОС: *Microsoft Windows*, *Linux* и *Novell NetWare*. Технология виртуализации *Intel (VT)* есть компонентой многоядерной технологии и обеспечивает поддержку виртуализации на аппаратном уровне. По своей сути *VT* обеспечивает поддержку виртуальных машин на уровне процессора, повышая как надежность и производительность работы приложений, так и общую безопасность. Архитектурой *VT* поддерживаются два класса ПО: монитор виртуальной машины (*VMM*) и гостевое ПО. Поддержка виртуализации на уровне процессора обеспечивается новым режимом работы чипа (*VMX*), кото-

рый поддерживается 10 новыми командами: *VMPTRLD*, *VMPTRST*, *VMCLEAR*, *VMREAD*, *VMWRITE*, *VMCALL*, *VMLAUCH*, *VMRESUME*, *VMXOFF* и *VMXON*. Выделяются два типа работы в режиме *VMX*: *VMX root operation* и *VMX non-root operation*. Как правило, монитор виртуальной машины *VMM* работает в режиме *root*, а гостевое ПО – в режиме *non-root*. Технология виртуализации *Intel* будет поддерживаться различными производителями операционных систем, например компаниями–поставщиками *Linux – RedHat*, *SuSe* и *Mont* (более подробную информацию о программном обеспечении виртуальных многоуровневых систем можно получить из [8]).

Виртуализация компонентов компьютерных систем для построения виртуальных рабочих мест

Виртуальный процессор – это не что иное, как процесс, обработку которого запланировала операционная система. Это процессы сервера баз данных. Они работают аналогично процессорам в компьютере и точно так, как и они, выполняют несколько процессов операционной системы, обслуживая нескольких пользователей, а также несколько потоков, обслуживая несколько приложений–клиентов *SQL*.

Можно создать несколько классов виртуальных процессоров, среди которых следует выделить [9]:

– *виртуальные процессоры CPU* выполняют все потоки сеансов – потоки, обрабатывающие требования от приложений–клиентов *SQL* и некоторые внутренние потоки, при этом внутренние потоки обеспечивают выполнение внутренних служб сервера баз данных;

– *виртуальные процессоры Java – Java-UDR* и *Java*-приложения выполняются на специализированных виртуальных процессорах, именуемых виртуальными процессорами *Java (Java Virtual Processor – JVP)*, при этом в код *JVP* встраивается виртуальная машина *Java (JVM)*. У *JVP* те же возможности, что и у *VP CPU*, в том смысле, что *JVP* может обрабатывать полные запросы *SQL*;

– *виртуальные процессоры дискового ввода–вывода* осуществляют обработку с помо-

стью следующих классов виртуальных процессоров:

- *PIO (Physical-Log I/O)* – ввод–вывод физического журнала;

- *LIO (Logical-Log I/O)* – ввод–вывод логического журнала;

- *AIO* – (Асинхронный ввод–вывод);

- *CPU* – (Асинхронный ввод–вывод ядра);

– *сетевые виртуальные процессоры* (связь между клиентом и сервером) – клиент может соединиться с сервером баз данных следующими способами:

- путем использования сетевого соединения;

- путем использования канала;

- через совместную память;

– *класс виртуальных процессоров модуля поддержки связи (Communications Support Module – CSM)* обеспечивает функции службы поддержки связи и модуля поддержки связи;

– *виртуальные процессоры шифрования*, если в файле конфигурации *ONCONFIG* не задана опция *encrypt* параметра *VPCLASS*, сервер баз данных запустит один виртуальный процессор шифрования при первом вызове любой из функций шифрования или расшифровки, заданной для шифрования на уровне столбцов, при этом, если потребуется, вы сможете задать несколько виртуальных процессоров шифрования, а чтобы сконфигурировать виртуальные процессоры шифрования, требуется использовать параметр конфигурации *VPCLASS* в сочетании с ключевым словом *encrypt*.;

– *виртуальный процессор оптической подсистемы* – класс виртуальных процессоров оптической подсистемы используется только внутри оптической подсистемы, при этом оптическая подсистема запускает виртуальный процессор в классе, если присутствует параметр конфигурации *STAGEBLOB*;

– *виртуальный процессор аудита*, – если вы включите режим аудита, задав для параметра *ADTMODE* в файле *ONCONFIG* значение 1, сервер баз данных запустит один виртуальный процессор в классе аудита (*ADT*);

– *виртуальные процессоры в группе «Разное»* – виртуальные процессоры в группе «Разное» (*MSC*) обслуживают требования систем-

ных вызовов, для выполнения которых может потребоваться очень большой стек, при этом на таком виртуальном процессоре выполняется только один поток; он выполняется с использованием стека соответствующего размера (например, 128 КБ);

– *виртуальные процессоры базового текстового поиска (Basic Text Search, BTS)* – этот процессор требуется для выполнения соответствующих запросов, при этом, одновременно он обрабатывает не более одной операции индекса, а чтобы обрабатывать несколько операций индекса *BTS* и запросов одновременно, необходимо создать дополнительные виртуальные процессоры *BTS*.

В таблице представлены классы виртуальных процессоров и типы выполняемой ими обработки. Количество виртуальных процессоров в каждом сконфигурированном классе зависит от наличия физических процессоров (*CPU*), аппаратной памяти и используемых приложений БД.

В сравнении с процессом сервера БД, который обслуживает одно приложение–клиент, динамический многопоточный виртуальный процессор сервера баз данных обеспечивает следующие преимущества [9]:

- виртуальные процессоры способны совместно с процессором сервера выполнять обработку, позволяют экономить память и ресурсы;

- выполнять параллельную обработку между виртуальными процессорами;

- создавать дополнительные виртуальные процессоры и прекращать работу активных виртуальных процессоров *CPU*, когда работает сервер БД, а также их можно привязывать к *CPU*.

Любой виртуальный процессор в классе может выполнять любой поток, относящийся к конкретному классу. Совместная обработка в классе виртуальных процессоров происходит автоматически и прозрачна для пользователя БД. Все потоки, запускаемые виртуальным процессором, имеют одинаковый доступ к памяти виртуального процессора, портам связи и файлам. Виртуальный процессор координирует доступ потоков к ресурсам. В общем случае, вир-

Т а б л и ц а. Классы виртуальных процессоров и типы выполняемой ими обработки [9]

Класс виртуальных процессоров	Категория	Назначение
1	2	3
<i>ADM</i>	Административное выполнение	Выполнение административных функций
<i>ADT</i>	Аудит	Выполнение аудита
<i>AIO</i>	Дисковый ввод-вывод	Выполнение дискового ввода-вывода, не связанного с ведением журнала. Если используется <i>KAIO</i> , виртуальные процессоры <i>AIO</i> выполняют операции ввода-вывода для буферизованных дисковых пространств
<i>BTS</i>	Базовый текстовый поиск	Выполнение операций индекса базового текстового поиска и обработка запросов
<i>CPU</i>	Централизованная обработка	Выполнение всех потоков сеансов и некоторых системных потоков. Выполнение потока для асинхронного ввода-вывода ядра (<i>Kernel Asynchronous I/O - KAIO</i>), если это возможно. Возможно выполнение одного потока опроса в зависимости от конфигурации
<i>CSM</i>	Модуль поддержки связи	Выполнение операций служб поддержки связи
<i>Encrypt</i>	Шифрование	Используется сервером баз данных при вызове функций шифрования и расшифровки
<i>Java VP (JVP)</i>	<i>Java UDR</i>	Выполнение <i>Java UDR</i> . Содержит <i>Java Virtual Machine (JVM)</i>
<i>LIO</i>	Дисковый ввод-вывод	Запись в файлы логического журнала (внутренний класс), если они находятся в буферизованном дисковом пространстве
<i>MSC</i>	Разное	Требования служб для системных вызовов, которым требуется очень большой стек
<i>OPT (UNIX)</i>	Оптическая подсистема	Выполнение операций ввода-вывода оптического диска
<i>PIO</i>	Дисковый ввод-вывод	Запись в файл физического журнала (внутренний класс), если он находится в буферизованном дисковом пространстве
<i>SHM</i>	Сеть	Осуществление связи через совместную память
<i>SOC</i>	Сеть	Осуществление сетевых взаимодействий с использованием гнезд
<i>TLI</i>	Сеть	Осуществление сетевых взаимодействий с использованием интерфейса <i>Transport Layer Interface (TLI)</i>
Имя класса	Пользовательские программы	Выполнение пользовательских подпрограмм в режиме безопасного потока, чтобы возможный сбой подпрограммы не повлиял на сервер баз данных. Определяется с использованием параметра конфигурации <i>VPCLASS</i> . Вы должны задать имя класса

туальный процессор способен переключаться с одного потока на другой быстрее, чем операционная система может переключаться с одного процесса на другой. Ниже перечислены случаи, когда виртуальные процессоры в классе *CPU* могут выполнять несколько сеансов потоков, работая параллельно для одного клиента: построение индекса; сортировка; восстановление; сканирование; объединение; сводка; группа; выполнение пользовательских подпрограмм (*UDR*).

Виртуальная память. Разработчикам программного обеспечения часто приходится решать проблему размещения в памяти больших программ, размер которых превышает объем доступной оперативной памяти [10]. Суть концепции виртуальной памяти заключается в следующем. Информация, с которой работает активный процесс, должна располагаться в оперативной памяти. В схемах виртуальной памяти

у процесса создается иллюзия того, что вся необходимая ему информация имеется в основной памяти. Для этого, во-первых, занимаемая процессом память разбивается на несколько частей, например страниц. Во-вторых, логический адрес (логическая страница), к которому обращается процесс, динамически транслируется в физический адрес (физическую страницу). И, наконец, в тех случаях, когда страница, к которой обращается процесс, не находится в физической памяти, нужно организовать ее подкачку с диска. Для контроля наличия страницы в памяти вводится специальный бит присутствия, входящий в состав атрибутов страницы в таблице страниц. Введение виртуальной памяти позволяет решать другую, не менее важную задачу – обеспечение контроля доступа к отдельным сегментам памяти и, в частности, защиту пользовательских программ друг от

друга и защиту ОС от пользовательских программ. Каждый процесс работает со своими виртуальными адресами, трансляцию которых выполняет аппаратура компьютера. В системах с виртуальной памятью те адреса, которые генерирует программа (логические адреса), называются виртуальными, и они формируют виртуальное адресное пространство. Термин *виртуальная память* означает, что программист имеет дело с памятью, отличной от реальной, размер которой потенциально больше, чем размер оперативной памяти.

Любая из рассмотренных схем управления памятью – страничной, сегментной и сегментно-страничной – пригодна для организации виртуальной памяти. Чаще всего используется сегментно-страничная модель, которая является синтезом страничной модели и идеи сегментации. Причем для тех архитектур, в которых сегменты не поддерживаются аппаратно, их реализация – задача архитектурно-независимого компонента менеджера памяти.

Виртуальные интерфейсы обеспечивают механизм разделения абстрактных моделей и тестовых программ от реальных сигналов, составляющих проект. Виртуальный интерфейс позволяет одной и той же подпрограмме работать в разных частях устройства и динамически управлять набором сигналов, связанных с подпрограммой. К преимуществам виртуального интерфейса в сравнении с другими типами интерфейсов можно отметить следующее [11]:

- Виртуальный интерфейс может использоваться, чтобы сделать *TestBench* независимым от физического интерфейса. Он позволяет разрабатывать тестовый компонент независимо от порта *DUT* при работе с многопортовым протоколом.

- С виртуальным интерфейсом можно динамически изменять ссылки на физический интерфейс. Без виртуальных интерфейсов вся связь определяется во время компиляции, и поэтому не может быть рандомизирована или переконфигурирована.

- В многопортовой среде он позволяет получить доступ к физическим интерфейсам с помощью индекса массива.

- Физические интерфейсы не допускаются в объектно-ориентированном программировании, так как физический интерфейс выделяется во время компиляции. Виртуальный интерфейс, заданный во время выполнения, позволяет делать объектно-ориентированное программирование с помощью сигналов, а не только с помощью переменных.

- Переменные виртуального интерфейса могут передаваться как аргументы в задачи, функции или методы.

Виртуальный интерфейс должен быть инициализирован до его использования, по умолчанию он указывает на *null*. Попытка использовать неинициализированный виртуальный интерфейс приведет к ошибке во время выполнения.

Основные этапы и задачи проектирования виртуальных рабочих мест

Далее представлены основные этапы и задачи, которые помогут спроектировать структуру виртуализации, наиболее полно отвечающую требованиям. В качестве основных этапов целесообразно отметить [12]:

Этап 1. Определение требований виртуальных машин к ресурсам.

Этап 2. Планирование конфигурации виртуальных машин.

Этап 3. Планирование групп узлов серверов виртуализации.

Этап 4. Планирование узлов серверов виртуализации.

Этап 5. Планирование особенностей архитектуры структуры виртуализации.

Этап 6. Планирование начальных характеристик.

Далее сочетание операционной системы и приложений, выполняющихся в виртуальной машине, определяется как рабочая нагрузка. Так что на *этапе 1* необходимо предварительно обозначить требования рабочих нагрузок к ресурсам виртуальной системы. При этом предварительно необходимо определить:

- быстродействие, архитектуру и количество требуемых процессоров;

- требуемую пропускную способность сети (Гбит/с) для входящего и исходящего трафика, а также максимальную задержку сети для выполнения рабочей нагрузки;

- емкость памяти хранилища (ГБ) для хранения файлов приложений, операционной системы и данных рабочей нагрузки, а также скорость ввода–вывода хранилища (в операциях в секунду) для рабочей нагрузки;

- объем памяти (ГБ), которая требуется для рабочей нагрузки, а также возможность поддержки архитектуры *NUMA*.

Этап 2. На этом этапе определяют типы виртуальных машин, которые потребуются для выполнения требований к ресурсам и обеспечения характеристик рабочих нагрузок, в том числе – поколение виртуальных машин, их функциональные возможности и типы поддерживаемых операционных систем. При этом дополнительно необходимо определить:

- объем памяти, характеристики и количество процессоров для каждой виртуальной машины;

- конфигурации сети и тип сетевого трафика, его производительность, доступность и параметры безопасности, а также стратегию назначения IP-адресов.

Этап 3. На данном этапе определяется группа узлов серверов. Группа узлов – это именованный набор серверов, которые выполняют общие задачи. При этом должны быть определены:

- физические расположения аппаратных ресурсов и управления ими;

- типы групп узлов;

- необходимость в кластеризации членов группы узлов.

Этап 4. На этом этапе определяются типы узлов, требуемые для размещения виртуальных машин, планируемых запускать в структуре виртуализации. При этом должно быть обозначено:

- конфигурации вычислительных ресурсов, в рамках конкретной задачи определяется объем памяти, количество процессоров и версия *Windows Server* для каждого узла;

- характеристики процессоров, параметры памяти и серии операционной системы;

- конфигурации сети, в том числе – особенности сетевого взаимодействия узла, включающего процедуры:

- определения типов сетевого трафика;

- определения производительности для сетевого трафика;

- определения стратегии обеспечения высокой доступности сети и агрегирования пропускной способности;

- определения виртуальных сетевых адаптеров;

- определения виртуальных коммутаторов;

- определения стратегии изоляции и защиты сетевого трафика;

- обеспечения качества обслуживания сети;

- обеспечения избыточности сети;

- изоляция трафика между сетями;

- конфигурация хранилища; чтобы определить конфигурацию хранилища, необходимо определить типы данных, которые будут сохраняться виртуальными машинами, и нужный им тип хранилища;

- уточнение типов данных; пример типов данных: файлы операционной системы; файл подкачки *Windows*; программные файлы приложений; данные конфигурации приложений; данные приложений; кластерные общие тома (*CSV*); кластеризации гостевых виртуальных машин; файлы аварийного дампа;

- определение типов хранилищ.

Ниже перечислены типы хранилищ, которые могут использоваться для типов данных, определенных ранее, на этапе 2: виртуальный диск *IDE*; виртуальный диск *SCSI*; инициатор *iSCSI* в виртуальной машине; виртуальный адаптер *Fibre Channel*; *SMB 3.0*;

- определение формата и типа виртуального жесткого диска: *VHD*, *VHDX*, общий диск *VHDX*;

- определение типа хранилища для каждого типа данных: определив типы данных, сохраняемых виртуальными машинами, и типы хранилищ, можно решить, какой тип хранилища и какие форматы и тип виртуального диска будут использоваться для каждого типа данных.

- определение стратегии доступности виртуальных машин.

Этап 5. На этом этапе разрабатывается логическая концепция, в соответствии с которой будет строиться архитектура структуры. При этом должны быть определены:

- домены обслуживания;
- физические домены сбоя;
- резервная мощность.

Этап 6. На этом этапе можно определить начальные затраты на размещение виртуальных машин и хранилища в структуре, а также исходные уровни качества обслуживания, которые может обеспечить структура. При этом должны быть определены:

- начальные показатели соглашения об уровне обслуживания для хранилища и виртуальных машин;
- начальные затраты на размещение хранилища и виртуальных машин;

Спроектировав структуру, можно рассчитать следующие значения:

- затраты на оборудование, помещения, электропитание и охлаждение структуры;
- емкость размещения структуры.

Вполне очевидно, что при оптимизации структуры и функций виртуальной машины и тем более – виртуальной системы потребуются неоднократное возвращение к соответствующим этапам. Более подробная информация о выполнении каждого этапа можно найти в [12]

Заключение. Итак, будущее ИТ принадлежит современным средствам виртуализации серверов и пользовательских рабочих мест, особенно для компаний, где работает большое количество сотрудников (например, тысяча и более). При этом развитие системы *VDI* даст возможность вообще отказаться от использования в работе пользователей мощных персональных компьютеров и значительно сократить штат сотрудников, осуществляющих техническую поддержку, перейдя на удаленное администрирование или ИТ аутсорсинг. Помимо всего прочего, развитие *VDI* в последующем позволит создать оптимальные условия для работающих удаленно пользователей.

В связи с массовым переходом на многоядерные решения, основные поставщики аппаратных и программных средств организовали ассоциацию *Multicore* (www.multicore-assotiation.org) [8]. Ее цель – разработка промышленных стандартов для многоядерных систем. Сейчас в рамках ассоциации ведется работа над четырьмя самостоятельными, но взаимосвязанными стандартами для многоядерных систем: *Resource Management (RAPI)*, *Communication API (CAPI)*, *Debug API* и *Transparent Interprocess Communication (TIPC)*; проведены рабочие встречи, в которых приняли участие такие крупнейшие ИТ-компании, как *Xilinx*, *Express Logic*, *Wind River*, *Freescale*, *ARC Int.*, *MIPS Technologies*, *Synopsys*, *PolyCore Software* и др.

При проектировании виртуальных рабочих мест важное значение следует придавать предпроектному исследованию предметной области применения виртуальных средств и решаемым этими средствами задачам, что позволит сократить количество итераций при выполнении указанных этапов и исключить значительное количество ошибок при проектировании.

1. Джеймс Смит, Рави Наур. Архитектура виртуальных машин – <https://www.osp.ru/os/2005/05-06/185586/>
2. Как построить современное виртуальное рабочее место? – <http://www.tadviser.ru/index.php>
3. Сравнение технологий для виртуализации рабочих станций (*VDI*) и классических ПК. – <http://efsol.ru/articles/desktop-virtualization.html>
4. Виртуализация рабочих мест. Что это? – <https://habrahabr.ru/company/hpel/blog/175213/>
5. Эталонная архитектура инфраструктуры интерактивной виртуализации *Cisco (VXI)*. Корпорация *Cisco Systems*, 2010. – [Cisco_interactive_virtualization_architecture.pdf](https://www.cisco.com/c/enr/whitepapers/cisco_interactive_virtualization_architecture.pdf)
6. Инфраструктура виртуальных рабочих столов на открытых программных продуктах / Д.В. Двоглазов, И.П. Дешко, К. Г. Кряженков и др. // Интернет-журнал «НАУКОВЕДЕНИЕ» Т. 7. – 2012. – № 4. – С. 1–11. – <http://naukovedenie.ru/PDF/37TVN415.pdf>
7. Виртуальный офис. – http://www.rinotel.com/products_virtualoffice.php
8. Золотарев С.В., Рыбаков А.Н. Программное обеспечение для многоядерных компьютерных систем: возможности и проблемы. – <http://www.rtsoft.ru/press/articles/detail.php?ID=1489>

9. *VP (Виртуальные процессоры) / Национальная библиотека им. Н.Э. Баумана.* – [http://ru.bmstu.wiki/index.php?title=VP_\(Virtual_Processor\)&mobileaction=togle_view_mobile](http://ru.bmstu.wiki/index.php?title=VP_(Virtual_Processor)&mobileaction=togle_view_mobile). – «[http://ru.bmstu.wiki/index.php?title=VP_\(Virtual_Processor\)&oldid=35318](http://ru.bmstu.wiki/index.php?title=VP_(Virtual_Processor)&oldid=35318)»
10. *Виртуальная память. Архитектурные средства поддержки виртуальной памяти.* – <http://www.intuit.ru/studies/courses/2192/31/lecture/984>
11. *Виртуальные интерфейсы.* – https://translate.google.com.ua/translate?hl=ru&sl=en&u=http://www.testbench.in/IF_05_VIRTUAL_INTERFACE.html
12. *Руководство по вопросам разработки структуры виртуализации.* – [https://technet.microsoft.com/ru-ru/library/dn800994\(v=ws.11\).aspx](https://technet.microsoft.com/ru-ru/library/dn800994(v=ws.11).aspx)

Поступила 07.03.2017
Тел. для справок: +38 044 526-3207 (Киев)
E-mail: yakyurlen@ukr.net
© Ю.С. Яковлев, 2017

UDC 004.2; 004.77.

Yu.S. Yakovlev¹

About the Workplaces Virtualization and Its Components while Construction the Computer Systems and Networks

¹ Doctor of Technical Sciences, V.M. Glushkov Institute of Cybernetics of National Academy of Sciences of Ukraine, Kyiv, Glushkov ave., 40, Kyiv, 03680, E-mail: yakyurlen@ukr.net

Keywords: virtual workplace, the virtual processor, virtual memory, the virtual interface, stages of construction of virtual cars.

Purpose. Introduction of the virtual workplaces infrastructure is becoming the dominating tendency. On the basis of the modern information technologies, essentially new types of the organizations – virtual which in comparison with the usual physical systems have a number of advantages are created. The main are the following: installation possibility on one computer of several OS; working with several OS simultaneously without system reboot; saving time for structure changing and establishing new OS; isolation of the real equipment from undesirable influence of the software; increasing of the system safety, etc. Besides, it gives the possibility to realize the concept “any place, any time, any device” (possibility to work from any place, at any time and from any device). It is confirm that the occurrence of the multinuclear processors and the creation on their basis the virtual systems will give a powerful additional impulse to mass introduction of the virtual technologies at the various levels.

Methods. In spite of the fact that the sources of the publication are foreign, its complex approach to the problem of virtualization is useful for developers of the virtual workplaces and systems as it mentions the set of the aspects connected with the designing of new information technology on the new base. Thus, the basic stages of the virtual cars and virtual workplaces construction and the definition of the virtual cars requirements are designated.

Results. The examples of the block diagramme of the virtual office and virtual computer system are presented, and also the example of the basic environment (systems of programmes of the virtual cars is allocated. Thus, it is possible to consider the construction of virtual workplaces as the complex discipline connected with working out of the equipment, operational systems and applied programs.

Conclusion. The concepts of a virtual workplace, the virtual processor, virtual memory and virtual means of input–output are given. 9 types and 13 classes of virtual processors are resulted, the detailed information on virtual memory is given, the advantages of the virtual interface of input–output comparing with the other types of interfaces are shown.

