

Федеральное государственное учреждение «Федеральный научный центр  
Научно-исследовательский институт системных исследований Российской академии наук»  
(ФГУ ФНЦ НИИСИ РАН)

## **ТРУДЫ НИИСИ РАН**

ТОМ 8 № 6

**МАТЕМАТИЧЕСКОЕ И КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ  
СЛОЖНЫХ СИСТЕМ:**

**ТЕОРЕТИЧЕСКИЕ И ПРИКЛАДНЫЕ АСПЕКТЫ**

МОСКВА  
2018

**Редакционный совет ФГУ ФНЦ НИИСИ РАН:**

В.Б. Бетелин (председатель),  
Е.П. Велихов, В.А. Галатенко, В.Б. Демидович (отв. секретарь), Б.В. Крыжановский,  
А.Г. Кушниренко, А.Г. Мадера, М.В. Михайлюк, В.Я. Панченко, В.П. Платонов,  
В.Н. Решетников

**Главный редактор журнала:**

В.Б. Бетелин

**Научный редактор номера:**

А.Г. Мадера

**Тематика номера:**

Исследование физических процессов и их моделирование, суперкомпьютерные  
информационные технологии, информационные технологии и электронные системы,  
информационные технологии в образовании и культуре.

Журнал публикует оригинальные статьи по следующим областям исследований:  
математическое и компьютерное моделирование, обработка изображений, визуализация,  
системный анализ, методы обработки сигналов, информационная безопасность,  
информационные технологии, высокопроизводительные вычисления, оптико-нейронные  
технологии, микро- и нанoeлектроника, математические исследования и вопросы  
численного анализа, история науки и техники.

**The topic of the issue:**

Research of physical processes and their modeling, Supercomputer information technologies,  
Information technologies and electronic systems, Information technologies  
in education and culture.

The Journal publishes novel articles on the following research arias: mathematical and computer  
modeling, image processing, visualization, system analysis, signal processing, information security,  
information technologies, high-performance computing, optical-neural technologies,  
micro- and nanoelectronics, mathematical researches and problems of numerical analysis,  
history of science and of technique.

Заведующий редакцией: Ю.Н.Штейников

Издатель: ФГУ ФНЦ НИИСИ РАН,  
117218, Москва, Нахимовский проспект 36, к. 1

# Организация виртуальной учебной лаборатории параллельного программирования на базе платформы виртуализации Proxmox VE

Ф.А.Аникеев<sup>1</sup>, А.В.Баранов<sup>2</sup>, Ф.С.Зайцев<sup>3</sup>, Е.А.Киселёв<sup>4</sup>, С.А.Лещев<sup>5</sup>

<sup>1,3</sup> ФГУ ФНЦ НИИСИ РАН, Москва, Россия,

<sup>2,4,5</sup> Межведомственный суперкомпьютерный центр РАН – филиал ФГУ ФНЦ НИИСИ РАН, Москва, Россия,

E-mails: <sup>1</sup>snowfed@gmail.com, <sup>2</sup>antbar@mail.ru, <sup>3</sup>fza@mail.ru, <sup>4</sup>kiselev@jscc.ru, <sup>5</sup>sergey.leshchev@jscc.ru

**Аннотация:** В статье рассматривается оптимальный подход к построению надёжной вычислительной инфраструктуры для виртуальной учебной лаборатории. Созданная виртуальная лаборатория успешно апробирована на Межфакультетской кафедре высокопроизводительных вычислений МГУ имени М.В. Ломоносова для эффективного обучения студентов параллельному программированию. Виртуальная инфраструктура лаборатории построена на основе программной платформы виртуализации Proxmox VE, свободно распространяемой в открытых исходных текстах. В статье рассмотрена методика создания виртуальной инфраструктуры, выделены ключевые моменты в подготовке и настройке виртуальной среды и виртуальных машин. Предложенный подход реализован на базе суперкомпьютера МВС-10П МП Petastream Межведомственного суперкомпьютерного центра – филиала НИИСИ РАН.

**Ключевые слова:** виртуальная инфраструктура, параллельное программирование, Proxmox VE, суперкомпьютер

## Введение

Важным аспектом развития суперкомпьютерного образования является постоянное совершенствование и актуализация учебно-методической и элементной базы для проведения всех форм и видов обучения. Возможности использования высокопроизводительных систем для обучения часто ограничены из-за их высокой загруженности и дороговизны. Одним из путей преодоления таких ограничений является создание виртуальных вычислительных систем, обладающих сходным составом программного обеспечения и достаточным для проведения учебных занятий количеством виртуальных вычислителей.

Технология виртуализации позволяет полноценно обучать программированию на суперкомпьютерах и экономить при этом значительные средства. В типичных условиях аппаратное и программное оснащение терминальных классов заметно отстаёт от передового уровня развития вычислительной техники и поэтому не даёт возможности изучения в полной мере новых технологий параллельного программирования. Однако вместо дорогостоящего апгрейда десятков морально устаревших компьютеров их можно использовать в качестве виртуальных рабочих столов, обес-

печивающих доступ к самым новым суперкомпьютерам, программистским средам и параллельным вычислительным методикам. Виртуальный рабочий стол не требует больших вычислительных ресурсов и его легко создать даже на планшете или смартфоне.

Кроме того, виртуализация позволяет существенно снизить накладные расходы на подготовку и поддержку суперкомпьютерного практикума. Так достаточно создать одну виртуальную машину с нужным программным обеспечением и затем размножить её копированием в требуемом числе экземпляров.

Преимущества виртуализации подтверждены авторами статьи совместно с Е.П. Сучковым, А.Г. Шишкиным и С.В. Степановым в реальном учебном процессе по обучению суперкомпьютерным технологиям на Межфакультетской кафедре высокопроизводительных вычислений МГУ имени М.В. Ломоносова. Заведующий кафедрой – научный руководитель НИИСИ РАН, академик В.Б. Бетелин.

Положительный опыт применения виртуализации в учебном процессе имеется и в других учебных заведениях. Например, в Южно-Уральском государственном университете создана облачная образовательная плат-

форма «Персональный виртуальный компьютер», использовавшая для своего выполнения суперкомпьютеры ЮУрГУ «СКИФ-Аврора» и «СКИФ Урал» [1]. Основу этой платформы составили гипервизоры Microsoft Hyper-V и инфраструктура рабочих столов Citrix XenDesktop VDI Edition.

Однако авторами настоящей статьи для создания виртуальной инфраструктуры, в отличие от работы [1], использована другая методика виртуализации. Её выбор основан на проведённых в Межведомственном суперкомпьютерном центре Российской академии наук (МСЦ РАН) научно-исследовательских работах [2-4] по вопросам построения виртуальных вычислительных систем для организации высокопроизводительных вычислений. Наиболее подходящей для виртуальной лаборатории по обучению в области параллельного программирования представляется свободно распространяемая платформа виртуализации Proxmox VE [5].

Платформа Proxmox уже нашла применение для учебных целей. Так, в работе [6] Proxmox используется как часть реконфигурируемой и отказоустойчивой платформы для создания открытых онлайн-курсов и проведения лабораторных работ. В [6] рассмотрено взаимодействие Интернет-ресурса и виртуального окружения Proxmox VE, однако не раскрыты процесс подготовки виртуального окружения Proxmox VE, его состав и характеристики используемого оборудования. Именно этим вопросам уделено основное внимание авторов настоящей статьи. Здесь представлены основные этапы и методика построения инфраструктуры для организации виртуальной учебной лаборатории на базе суперкомпьютерных ресурсов МСЦ НИИСИ РАН, описаны особенности функционирования построенной инфраструктуры.

## Выбор платформы виртуализации

Основными инструментами для создания виртуальных инфраструктур являются гипервизоры и контейнеры. Преимуществом гипервизоров по сравнению с контейнерами является возможность одновременного выполнения на одном вычислительном узле виртуальных машин с несколькими отличными друг от друга гостевыми операционными системами (ОС). Поскольку создаваемая инфраструктура для виртуальной учебной лаборатории предполагает универсальное использование, в т.ч. виртуальных машин (ВМ) с ОС как семейства Linux, так и семейства Microsoft

Windows, в качестве технологии для создания инфраструктуры была выбрана гипервизорная виртуализация.

Поскольку бюджеты научных и образовательных учреждений обычно сильно ограничены, использование платного программного обеспечения для построения виртуальных сред нецелесообразно. Создаваемая авторами виртуальная инфраструктура предназначена для использования исключительно в образовательных целях, её функционирование не требует регулярных обновлений, а уровень квалификации сотрудников МСЦ РАН позволяет администрировать виртуальную инфраструктуру без обращения в службу технической поддержки производителей программного обеспечения. С учётом перечисленных особенностей, для построения виртуальной инфраструктуры было решено выбрать гипервизор KVM.

Следует отметить, что выбор гипервизора всегда сопровождается выбором программных средств управления виртуальной вычислительной средой: виртуальными машинами, серверами виртуализации, гипервизорами, учётными записями пользователей, файловым хранилищем.

Для работы с гипервизором KVM используют следующие системы управления виртуальной средой, распространяемые в открытых исходных текстах: Virtual Machine Manager [7], Proxmox Virtual Environment [8], AQEMU [9], GKVM [10], Cloudmin [11], oVirt [12].

Virtual Machine Manager входит в состав всех репозиториях ОС семейства Linux и представляет собой приложение, позволяющее управлять гипервизорами KVM и Xen, а также виртуальными машинами на одном или нескольких серверах виртуализации.

Proxmox Virtual Environment (VE) – комплекс программного обеспечения для построения отказоустойчивых виртуальных сред с централизованной моделью управления через web-интерфейс. Комплекс Proxmox VE распространяется в виде отдельного дистрибутива на основе ОС Debian Linux, но может быть установлен и на другие ОС семейства Linux.

AQEMU – графическая оболочка управления виртуальными машинами QEMU и KVM. AQEMU разрабатывалась как более удобная и простая в использовании альтернатива Virtual Machine Manager. Её отличительной особенностью является возможность генерации командных сценариев (скриптов) для автоматизации запуска

виртуальных машин с требуемыми настройками.

GKVM – графическая оболочка гипервизора KVM для ОС семейства Linux. Её отличительной особенностью является минимальный объём занимаемого дискового пространства после установки.

Cloudmin – web-интерфейс для централизованного управления виртуальными машинами и гипервизорами Xen, OpenVZ, KVM, VServers, Solaris Zones, Amazon EC2. Бесплатная версия данного программного обеспечения позволяет управлять гипервизором Xen или KVM на одном сервере. В платной версии реализована возможность одновременного управления несколькими гипервизорами Xen, OpenVZ, KVM, Vservers, EC2 и Solaris Zone.

OVirt – web-интерфейс для централизованного управления виртуальными машинами и гипервизорами KVM, развивающийся открытым сообществом разработчиков как альтернатива VMware vSphere. Отличительной особенностью этого программного обеспечения является возможность установки и запуска с флеш-накопителя объёмом 64 Мбайта.

Для работы всех перечисленных программных средств, за исключением Proxmox VE, Cloudmin и oVirt, необходимо наличие графической оболочки. Программные средства AQEMU, GKVM, QtEmu обладают функционалом в большей степени ориентированным на обычные рабочие станции пользователей, что сильно затрудняет их использование для реализации масштабируемых виртуальных систем. При выборе системы управления виртуальной средой из Proxmox VE, Cloudmin и oVirt особое внимание обращалось на совместимость с различными видами дистрибутивов ОС семейства Linux и наличием в бесплатной версии программного продукта необходимого функционала для развёртывания и управления виртуальной вычислительной средой на нескольких физических серверах. Среди перечисленных программных средств указанными свойствами обладал только комплекс программного обеспечения Proxmox VE, который и был выбран в качестве основного средства построения инфраструктуры для виртуальной учебной лаборатории.

В Proxmox VE каждая виртуальная машина представлена двумя видами файлов: файлом с настройками ВМ и файлами виртуального диска. Файл с настройками ВМ может создаваться отдельно и использоваться для подключения уже имеющихся образов диска. Образ виртуального диска является

файлом или группой файлов, в которых виртуальная машина хранит свои данные. В Proxmox VE могут использоваться несколько различных форматов образов виртуальных дисков: raw, qcow2 и vmdk.

Для построения инфраструктуры виртуальной учебной лаборатории авторами был выбран формат qcow2 – формат гипервизора KVM, основным преимуществом которого является возможность автоматического увеличения размера файла-образа ВМ по мере его заполнения, а также поддержка механизма снапшотов. По сравнению с остальными альтернативами формат qcow2 позволяет экономнее расходовать дисковое пространство системы хранения, но при этом обладает достаточной производительностью для решения задач обучения. Кроме этого, формат qcow2 является одним из рекомендованных для ВМ в среде Proxmox.

## Порядок установки и настройки виртуальной инфраструктуры

При создании виртуальной инфраструктуры для проведения учебных занятий в качестве базового использовался подход [3] с рядом изменений, отражающих специфику настоящего проекта. Для построения виртуальной инфраструктуры было выделено:

- 4 вычислительных узла (hps05-hps08) из состава суперкомпьютера МБС-10П МП Petastream [13];
- дисковое пространство подсистемы хранения данных на базе NetApp FAS под управлением ОС 7-mode Data ONTAP (далее – 7-mode);
- дисковое пространство подсистемы хранения данных на базе NetApp FAS под управлением ОС Clustered Data ONTAP (далее – CDOT);
- часть ресурсов VPN-концентратора существующей сетевой инфраструктуры МСЦ РАН.

Для организации сетевого взаимодействия и управления виртуальной средой было создано две виртуальных локальных (VLAN-сегмента) сети:

- сеть управления и доступа для удалённого управления через сеть Интернет виртуальными машинами и гипервизорами, организации сетевого взаимодействия ВМ между собой и доступа студентов к удалённым рабочим столам ВМ;
- сеть хранения и передачи данных для организации доступа гипервизоров к

хранилищам виртуальных машин, размещённых на подсистемах хранения данных.

Общая структура инфраструктуры для виртуальной учебной лаборатории представлена на рисунке 1.

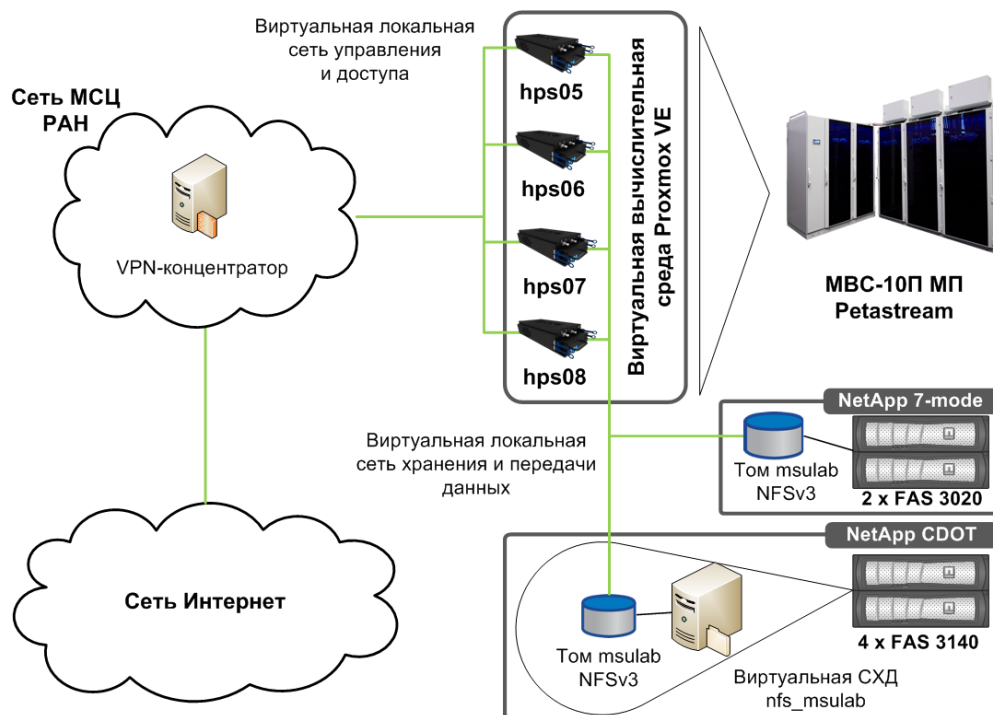


Рисунок 1. Общая структура виртуальной среды

Обязательным требованием создания отказоустойчивой виртуальной среды является наличие дискового пространства для хранения виртуальных машин, разделяемого между всеми гипервизорами. Дисковое пространство для проекта в размере 2 TB первоначально было выделено на подсистеме хранения данных 7-mode NetApp FAS 3020, доступ к сетевому каталогу осуществлялся по протоколу NFSv3. Рост требований к производительности и объёму дискового пространства во время эксплуатации виртуальной среды потребовал изменения подхода к выделению дискового пространства, поэтому тела виртуальных машин были перемещены в сетевой каталог, организованный на более производительной подсистеме CDOT NetApp FAS 3140 с тем же файловым протоколом доступа NFSv3. Для более эффективной утилизации дискового пространства и работы механизма кэширования чтения на выделенном томе была задействована блочная дедупликация, выполняемая по расписанию еженедельно в выходной день. Дедупликация обеспечила экономию дискового пространства в 79% (1,29 TB).

На каждом вычислительном узле суперкомпьютера MBC-10П МП Petastream была установлена ОС Proxmox VE 5.1 и

выполнена процедура объединения узлов в кластер. Объединение вычислительных узлов в кластер ОС Proxmox необходимо для повышения отказоустойчивости ВМ и организации балансировки нагрузки.

Необходимое число вычислительных узлов кластера было определено исходя из требований сохранения работоспособности кластера при отказе либо временном выводе из эксплуатации для обслуживания одного из узлов.

Для каждой ВМ через графический интерфейс ОС Proxmox можно задать порядок загрузки ВМ, правила миграции ВМ при сбоях и активировать или деактивировать балансировку нагрузки.

Поскольку балансировка нагрузки может приводить к деградации производительности, увеличению нагрузки на процессорные ядра, дисковую подсистему и сеть при миграции ВМ с одного на другой вычислительный узел, данная настройка была отключена для всех виртуальных машин, кроме сервисных.

Для построения виртуальной инфраструктуры были созданы два типа виртуальных машин:

– сервисные – для организации сетевого доступа и функционирования ВМ;

– рабочие станции – виртуальные рабочие станции для проведения лабораторного практикума.

К сервисным VM были отнесены DHCP- и DNS-сервер под управлением ОС Linux, файловый сервер для обмена данными между пользователями VM, совмещённый с ActiveDirectory, сервер лицензий для ПО Microsoft Visual Studio 2015 и Intel Parallel Studio XE 2018 под управлением ОС Windows Server. Для сервисных VM были разрешена

миграция и выставлен наивысший приоритет для загрузки. Для каждой сервисной VM под управлением ОС Linux были выделены 1 виртуальное ядро и 1 ГБ оперативной памяти, для VM под управлением ОС Windows Server – 2 виртуальных ядра и 4 ГБ оперативной памяти. Сетевые интерфейсы всех сервисных VM были размещены внутри сети управления и доступа. Компонентная схема созданной виртуальной инфраструктуры представлена на рисунке 2.

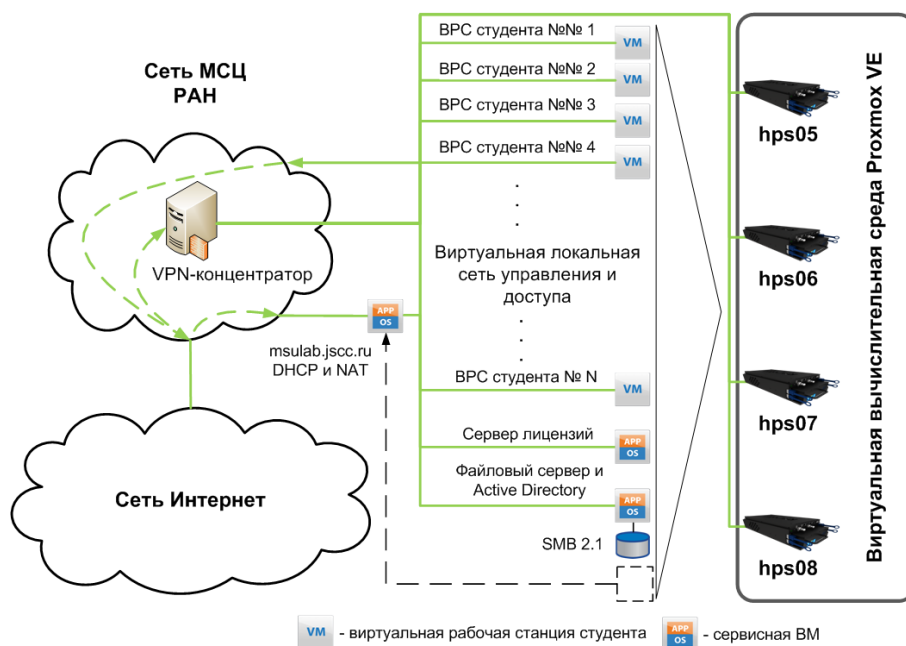


Рисунок 2. Компоненты виртуальной инфраструктуры

При создании виртуальной среды для организации и проведения лабораторного практикума у преподавателя должна быть возможность раздавать студентам учебные материалы на занятиях. Для этих целей внутри сети управления и доступа средствами сервисной VM файлового сервера под управлением ОС Microsoft Windows Server был создан сетевой каталог с правами доступа на чтение для студентов и полным доступом для преподавателей.

В роли рабочих станций студентов выступили VM под управлением ОС Windows 10. Эталонный образ виртуальной рабочей станции был создан и настроен в ПО VMware Player. Подготовленный образ VM был конвертирован в формат qcow2 с использованием штатного программного средства qemu-img, входящего в состав ОС Proxmox, и размножен в 25-ти экземплярах – по числу студентов и преподавателей, работающих через web-интерфейс ОС Proxmox.

Для каждой виртуальной рабочей станции выделено по 2 виртуальных ядра и 4

ГБ оперативной памяти. Сетевые интерфейсы всех виртуальных рабочих станций размещены внутри сети управления и доступа. Для каждого студента выделена собственная VM и создана персональная локальная учётная запись. В целях обеспечения отказоустойчивости для каждой VM средствами Proxmox VE настроено расписание автоматического создания резервной копии после окончания занятий.

Сетевой доступ к виртуальным рабочим станциям был организован посредством технологии NAT через сервисную VM, выполняющую функции DHCP- и DNS-сервера, один из сетевых интерфейсов которой размещён в сети Интернет, а второй – внутри сети управления и доступа. Доступ к виртуальным рабочим станциям организован по протоколу RDP (Remote Desktop Protocol) и может быть осуществлён с любой рабочей станции, планшета или смартфона.

Для организации обратного сетевого маршрута при доступе к виртуальным рабочим станциям и возможности выхода в

Интернет при удаленной работе на них для сети управления и доступа была организована маршрутизация трафика через существующую сетевую инфраструктуру МСЦ РАН.

Управление элементами виртуальной инфраструктуры осуществляется через web-интерфейс Proxmox с использованием отдельных учётных записей сервера LDAP МСЦ РАН. Каждой учётной записи средствами Proxmox VE назначена одна из ролей: администратор виртуальной инфраструктуры или администратор виртуальных машин. Роль администратора виртуальной инфраструктуры подразумевает полный доступ ко всем элементам управления виртуальной средой Proxmox VE и назначается специально выделенным сотрудникам МСЦ РАН. Роль администратора виртуальных машин назначается учётным записям преподавателей и позволяет управлять набором виртуальных машин, заданным администратором виртуальной инфраструктуры. Для всех администраторов средствами VPN-концентратора сетевой инфраструктуры МСЦ РАН был организован VPN-доступ в сеть управления и доступа.

## Порядок настройки виртуальных рабочих станций

Для проведения лабораторного практикума на виртуальные рабочие станции студентов установлено следующее программное обеспечение:

- Notepad++;
- Microsoft Visual Studio 2015 Community Edition;
- Intel Parallel Studio XE 2018 Cluster Edition;
- Microsoft Visual Studio 2008 Professional Edition;
- NVIDIA CUDA Toolkit 2.3;
- NVIDIA SDK 2.3.

Последние три пункта нужны для использования CUDA-эмулятора на компьютерах без видеокарт NVIDIA.

Intel Parallel Studio предоставляется с лицензией на ограниченное число пользователей, лицензирование проводится через сервер лицензий Intel Software License Manager и лицензионный ключ, установленные на специально созданной для этой цели сервисной ВМ.

Отдельным важным вопросом подготовки виртуальных рабочих станций является настройка автоматических обновлений ОС. Автоматические обновления повышают безопасность ОС, но могут приводить к частым перезагрузкам ОС, в том числе при проведении занятий. В ОС Windows 10 не предусмотрено полное отключение обновлений, но су-

ществует возможность настройки интервалов между ними.

Другим не менее важным пунктом настройки виртуальной рабочей станции для проведения лабораторного практикума является ограничение доступа обучающихся к сети Интернет. Доступ к сети Интернет может быть ограничен конкретным приложениям с использованием стандартных компонентов ОС Windows 10. Закрыть доступ в Интернет рекомендуется, по крайней мере, следующим приложениям: Internet Explorer, Microsoft Edge, Microsoft Store, Microsoft OneNote, Skype, Windows Messenger, Windows Store.

Следует отметить, что у студентов остаётся возможность выхода в сеть Интернет с помощью протокола RDP, который позволяет копировать файлы, в том числе исполняемые, с локального диска на удалённый компьютер. Студент может скопировать на удалённый компьютер портативный браузер, для которого доступ в Интернет не будет ограничен настроенными правилами. Для того чтобы предотвратить подобное копирование, необходимо запретить перенаправление дисков для службы удалённых рабочих столов рабочей станции. При этом запрет будет действовать на всех пользователей, включая администраторов.

## Организация доступа к виртуальным машинам на учебных занятиях

Доступ к виртуальным рабочим столам организуется по протоколу RDP с использованием любой совместимой программы-клиента. При первом подключении к ВМ студент должен зайти в магазин приложений на своем устройстве и установить приложение Remote Desktop от Microsoft («Удалённый рабочий стол»). В ОС Windows для рабочих станций по умолчанию уже установлен стандартный RDP-клиент для подключения к удалённому рабочему столу. В ОС семейства Linux может быть использован RDP-клиент Remmina.

Доступ к удалённым рабочим столам организован через специальную шлюзовую машину, доступную через Интернет. При подключении в адресе должен быть указан специальный номер порта, по которому шлюзовая машина перенаправляет соединение на соответствующую номеру порта ВМ.

Стандартными средствами ОС Windows на ВМ учётным записям студентов разрешён вход только на время занятий. Через минуту после окончания занятия для пользо-

вателя-студента принудительно выполняется выход из системы без возможности повторного подключения до следующего занятия. У учётных записей преподавателей доступ не ограничивается по времени. В часы проведения занятий студент может также подключаться к машинам фактически из любого места, где есть доступ к сети Интернет. Другими словами, инфраструктура пригодна и для удалённого обучения. Кроме того, построенная инфраструктура позволяет, в случае необходимости (болезнь, семейные обстоятельства и др.), по заявкам студентов в течение семестра предоставлять дополнительные интервалы доступа.

## Методика построения инфраструктуры для организации виртуальной учебной лаборатории

Опыт реализации рассматриваемого проекта позволил определить следующую последовательность этапов построения виртуальной инфраструктуры для организации и проведения процесса обучения.

1. Выделение требуемого количества однородных вычислительных узлов, которое может быть вычислено по формуле (1):

$$\max \left\{ \left\lceil \frac{M_{vm} \cdot C_{vm}}{M_n - k} \right\rceil; \left\lceil \frac{C_{vm} \cdot N_{vm}}{N_n} \right\rceil \right\} + n \quad (1)$$

$C_{vm}$  – общее количество VM, необходимое для проведения занятий;

$M_{vm}$  – требуемый объём оперативной памяти для 1 VM (определяется на основании требований ОС и устанавливаемого программного обеспечения);

$M_n$  – объём оперативной памяти вычислительного узла;

– объём оперативной памяти на каждом узле, резервируемой под нужды гипервизора и загрузочной ОС;

– количество резервируемых вычислительных узлов;

$N_n$  – количество процессорных ядер в вычислительном узле;

$N_{vm}$  – требуемое количество ядер в VM.

2. Выделение требуемого объёма дискового пространства в системе хранения данных для размещения файлов VM и служебной информации. Объём выделяемого дискового пространства может быть вычислен по формуле (2):

$$S = V_{HDD} \cdot C_{vm} + \sum S_{HDD} \quad (2)$$

$V_{HDD}$  – объём дискового пространства для VM рабочей станции (как правило, определяется на сайте разработчика гостевой ОС);

$C_{vm}$  – общее количество VM рабочих станций, необходимое для проведения занятий;

$S_{HDD}$  – объём дискового пространства для сервисной VM.

3. Объединение вычислительных узлов коммуникационной средой, состоящей из двух сегментов сети:

– сеть управления и доступа (маршрутизируемая сеть для организации доступа к VM и вычислительным узлам, в том числе через VPN-соединения);

– сеть хранения и передачи данных (изолированная сеть для доступа вычислительных узлов к системам хранения данных).

4. Установка и настройка гипервизоров на выделенных вычислительных узлах.

– установка Proxmox VE на все узлы;

– объединение вычислительных узлов в кластер виртуализации средствами ОС Proxmox VE с использованием сети управления и доступа;

– подключение к кластеру виртуализации через сеть хранения и передачи данных сетевых хранилищ для размещения VM и дистрибутивов ПО.

5. Подготовка и настройка сервисных VM в формате qcow2 с использованием средств ОС Proxmox VE.

5.1. Создание и настройка сервисных VM:

– для организации сетевого доступа (DNS, DHCP, VPN) с двумя сетевыми интерфейсами к сети управления и доступа и к сети Интернет;

– для управления пользователями (LDAP, ActiveDirectory) с сетевым интерфейсом к сети управления и доступа;

– для файлового сервера обмена данными между пользователями VM с сетевым интерфейсом к сети управления и доступа;

– для функционирования специального ПО (серверы лицензий) с сетевым интерфейсом к сети управления и доступа.

5.2. Настройка правил миграции сервисных VM при выходе из строя вычислительного узла через web-интерфейс Proxmox VE.

6. Подготовка образа VM (в формате qcow2) с рабочими столами для проведения

лабораторного практикума с использованием средств ОС Proxmox VE.

6.1. Создание и настройка эталонного образа ВМ с сетевым интерфейсом к сети управления и доступа.

6.2. Создание через интерфейс Proxmox VE из подготовленного эталонного образа требуемого для проведения лабораторного практикума числа копий ВМ.

6.3. Назначение на DHCP-сервере развёрнутым ВМ IP-адресов в соответствии с MAC-адресами их сетевых интерфейсов.

7. Тестирование созданной виртуальной инфраструктуры методом белого ящика.

## Заключение

В представленной работе авторами решена фундаментальная задача организации виртуальной учебной лаборатории параллельного программирования. Виртуальная инфраструктура построена на базе свободно распространяемой платформы виртуализации Proxmox VE и вычислительных ресурсах суперкомпьютера МВС-10П МП Petasream МСЦ НИИСИ РАН. Определены оптимальный состав и компоненты подобной инфраструктуры,

разработана эффективная методика её построения и отказоустойчивого функционирования.

Важно подчеркнуть, что настройка виртуальной инфраструктуры лаборатории проводится один раз. Далее она функционирует автономно практически без вмешательства специалиста по виртуализации.

Реализованная с помощью предложенной методики виртуальная учебная лаборатория успешно апробирована в весеннем семестре 2018 года для проведения учебных занятий (лабораторного практикума) по параллельному программированию на супер-ЭВМ и графических картах NVIDIA для студентов «Межфакультетской кафедры высокопроизводительных вычислений» МГУ имени М.В. Ломоносова, заведующий кафедрой – научный руководитель НИИСИ РАН, академик В.Б. Бетелин.

Публикация выполнена в рамках государственного задания по проведению фундаментальных научных исследований по теме (проекту) «Разработка архитектур, системных решений и методов для создания вычислительных комплексов и распределенных сред мультитепетафлопсного диапазона производительности, в том числе нетрадиционных архитектур микропроцессоров» (0065-2018-0409).

# The virtual educational laboratory based on the Proxmox VE virtualization platform

**F.A.Anikeev, A.V.Baranov, F.S.Zaitsev, E.A.Kiselev, S.A.Leschev**

**Abstract:** The article discusses methods and approaches to build a virtual educational laboratory based on the Proxmox VE virtualization platform. The structure and components of the virtual infrastructure that is built on the base of the Proxmox system are considered. Virtual infrastructure designed by the authors was deployed on the MVS-10P supercomputer at Joint supercomputer center of RAS. The infrastructure was used in Moscow state university for teaching parallel programming.

**Keywords:** proxmox, education, parallel programming, HPC, supercomputer

## Литература

1. А.Л.Шестаков, О.Н.Иванова, Л.Б.Соколинский, П.С.Костенецкий. Использование «облачной» образовательной платформы «Персональный виртуальный компьютер» в обучении программированию студентов ИТ-направлений. Преподавание информационных технологий в Российской Федерации: материалы Одиннадцатой открытой Всероссийской конференции. 2013. С. 56-58.
2. Б.М.Шабанов, А.П.Овсянников, А.В.Баранов, О.С.Аладышев, Е.А.Киселёв, Я.О.Жуков. Методы управления параллельными заданиями суперкомпьютера, требующими развёртывания отдельных программных платформ и виртуализации сетей. Суперкомпьютерные дни в России: Труды международной конференции. 2017. С. 616-627.
3. Б.М.Шабанов, А.П.Овсянников, А.В.Баранов, Е.А.Киселёв, С.А.Лещев, Б.В.Долгов, Д.Г.Гуменный, Д.Л.Шуров. Решение задач фотореалистичной компьютерной графики на базе защищённой инфраструктуры суперкомпьютерного центра коллективного пользования. Суперкомпьютерные дни в России: Труды международной конференции. 2017. С. 733-741.
4. О.С.Аладышев, А.В.Баранов, Р.П.Ионин, Е.А.Киселёв, В.А.Орлов. Сравнительный анализ вариантов развёртывания программных платформ для высокопроизводительных вычислений // Вестник Уфимского государственного авиационного технического университета. 2014. Т. 18. № 3 (64). С. 295-300.
5. A.Kovari, P.Dukan. KVM & OpenVZ virtualization based IaaS open source cloud virtualization platforms: OpenNode, Proxmox VE. 2012 IEEE 10th Jubilee International Symposium on Intelligent Systems and Informatics, Subotica, 2012, pp. 335-339. DOI: 10.1109/SISY.2012.6339540
6. L.Chen, W.Huang, F.Sui, D.Chen, and C.Sun. The online education platform using Proxmox and noVNC technology based on Laravel framework. 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS), Wuhan, 2017, pp. 487-491. DOI: 10.1109/ICIS.2017.7960041
7. Manage virtual machines with virt-manager. URL:<https://virt-manager.org> (дата обращения: 02.11.2018).
8. Proxmox Virtual Environment. URL:[https://pve.proxmox.com/wiki/Main\\_Page](https://pve.proxmox.com/wiki/Main_Page) (дата обращения: 02.11.2018).
9. AQEMU a GUI for virtual machines using QEMU as the backend. URL:<https://sourceforge.net/projects/aqemu> (дата обращения: 02.11.2018).
10. GKVM. A Gnome user interface for KVM. URL:<http://gkvm.sourceforge.net> (дата обращения: 02.11.2018).
11. Cloudmin. URL:<http://www.webmin.com/cloudmin.html> (дата обращения: 02.11.2018).
12. Ovirt. Powerful open source virtualization. URL:<https://ovirt.org> (дата обращения: 02.11.2018).
13. MBC-10П МП PETASTREAM. URL:<http://www.jscc.ru/resources/hpc/#item1587> (дата обращения: 02.11.2018).