Методы управления параллельными заданиями суперкомпьютера, требующими развёртывания отдельных программных платформ и виртуализации сетей*

Б.М. Шабанов, А.П. Овсянников, А.В. Баранов, О.С. Аладышев, Е.А. Киселёв, Я.О. Жуков

Межведомственный суперкомпьютерный центр РАН – филиал ФГУ ФНЦ НИИСИ РАН

Статья посвящена проблеме управления потоком параллельных заданий, требующих для своего выполнения развёртывания отдельных программных платформ с реконфигурацией коммуникационных сетей. Программные платформы реализуются набором виртуальных машин, параллельно выполняющихся на динамически выделяемых суперкомпьютерных ресурсах. Виртуальные машины объединяются совокупностью виртуальных сетей, в свою очередь определяемых пользователем. Авторами предложены способы динамической организации виртуальной сетевой среды, в т.ч. способ описания внутренней сетевой структуры и метод её отображения на физическую сетевую инфраструктуру. Рассмотрен способ автоматической настройки сетевого оборудования суперкомпьютера при запуске и завершении заданий.

Ключевые слова: виртуализация в суперкомпьютерах, системы пакетной обработки заданий, виртуальные сети

1. Введение

В настоящей работе под системой высокопроизводительных вычислений будем понимать вычислительную установку типовой кластерной архитектуры, состоящую из нескольких вычислительных модулей, объединённых одной или несколькими высокоскоростными сетями. Вычислительный модуль такой системы представляет собой самостоятельный компьютер, оснащённый, как правило, несколькими центральными процессорами (двумя или более), собственными оперативной и дисковой памятями. Важно, что каждый вычислительный модуль управляется отдельным экземпляром операционной системы (как правило, Linux).

Управление вычислительными ресурсами и пользовательскими заданиями в современных системах высокопроизводительных вычислений осуществляет специальное программное обеспечение – системы пакетной обработки (СПО) [1]. СПО обеспечивает коллективный доступ пользователей к супер-ЭВМ, принимает входной поток различных заданий от разных пользователей, планирует очереди заданий, выделяет необходимые для выполнения задания вычислительные ресурсы и освобождает их после завершения задания.

Современные СПО ориентированы на ведение разных очередей для разных типов так называемых **стандартных заданий**, которые выполняются в развёрнутой на установке программной среде и не требуют внесения изменений в процесс конфигурации вычислительных модулей. В этом случае пользователи попадают в зависимость от установленного на кластере ПО. Задания должны создаваться с учетом набора ПО, вспомогательных библиотек и их версий на конкретном кластере, в связи с чем возникают сложности с переносом заданий на другой кластер. В последние годы в суперкомпьютерных центрах коллективного пользования всё чаще появляются **нестандартные задания**, предъявляющие особые требования к ресурсам. Подобным заданиям могут потребоваться отличная от стандартной операционная система (например, MS Windows), специфическое окружение, особые программные пакеты и лицензии, то есть **собственная программная платформа (среда)**.

Одним из способов создания для задания отдельной программной платформы является представление параллельного задания в виде набора виртуальных машин, развёртываемых перед стартом задания из заранее подготовленного образа (образов).

* Работа выполнена за счёт бюджетных средств в рамках выполнения государственного задания

Для СПО возникает новая задача – обеспечение возможности автоматического обслуживания потока нестандартных заданий, представленных набором виртуальных машин, совместно с потоком обычных стандартных заданий. Настоящая статья отражает работу авторского коллектива над решением этой задачи в МСЦ РАН.

2. Макет облачной среды для высокопроизводительных приложений

Для решения поставленной задачи в МСЦ РАН был создан макет облачной среды для высокопроизводительных приложений, представленный на рисунке 1.

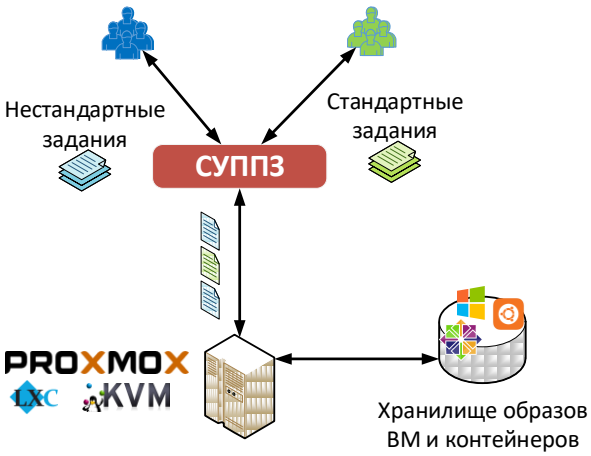

Рис. 1. Макет облачной среды для высокопроизводительных приложений

Макет включает в себя вычислительный кластер, работающий под управлением отечественной СПО – системы управления прохождением параллельных заданий (СУППЗ) [2], дополненной вновь разработанной подсистемой управления виртуальными машинами (ПУВМ). Каждый вычислительный модуль из состава кластера был оснащён гипервизором KVM. В системе хранения данных (СХД) кластера выделена область для хранения образов виртуальных машин (ВМ) и контейнеров. Для обеспечения возможности развёртывания и управления виртуальными машинами применена свободно распространяемая платформа виртуализации Proxmox Virtual Environment (Proxmox VE) [3].

Макет предоставляет возможность представления параллельного задания в виде набора образов виртуальных машин, что позволяет пользователю сформировать нестандартное задание для высокопроизводительных вычислений на базе любой программной платформы (Linux/Windows и т.п.) и направить это задание в СУППЗ. Когда нестандартное задание попадает в СУППЗ, оно проходит через очередь системы наряду со стандартными (традиционными) заданиями, рассчитанными на выполнение в невиртуализованной программной среде, развёрнутой на кластере по умолчанию (как правило, эта стандартная среда представляет собой связку «Linux+MPI+OpenMP»). При запуске нестандартного задания задействуется ПУВМ, которая, в свою очередь, используя платформу Proxmox VE, извлекает из СХД образ виртуальной машины, содержащий необходимую заданию программную платформу, и разворачивает набор виртуальных машин из этого образа на выделенных для задания вычислительных модулях. При этом автоматически конфигурируется виртуальная локальная сеть (VLAN), доступная пользователю для запуска его параллельного приложения на развёрнутой виртуальной программной платформе (рисунок 2).

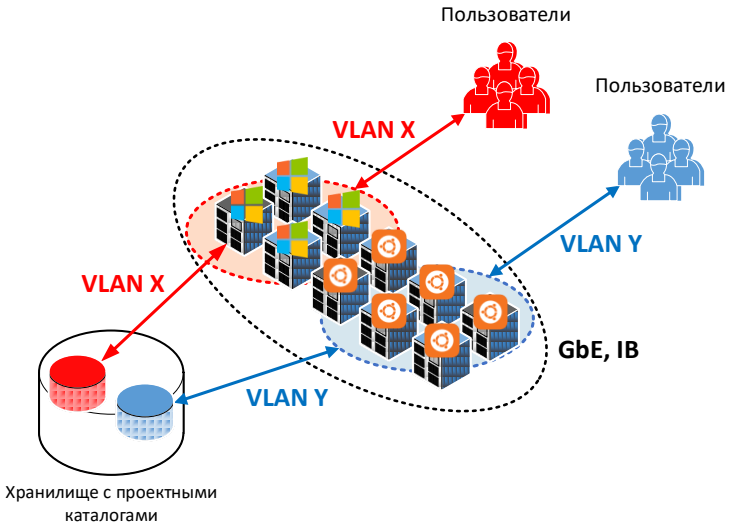

Рис. 2. Виртуальная коммуникационная среда

Процесс подготовки и запуска нестандартного задания (рисунок 3) состоит из следующих этапов. На первом этапе пользователь или администратор подготавливают образ виртуальной машины в формате qcow2, после чего администратор сохраняет подготовленный образ в специальном хранилище образов ВМ и контейнеров. На следующем этапе пользователь формирует паспорт нестандартного задания с включением в него информации о том, какой образ ВМ из хранилища будет использован, какие ресурсы (число ядер, объём оперативной памяти) требуются для функционирования виртуальной машины, развёрнутой из этого образа. Далее нестандартное задание направляется в очередь СУППЗ.

После того, как нестандартное задание пройдёт через очередь, ПУВМ развернёт на выделенных ресурсах виртуальные машины из указанного в паспорте задания образа. По завершении процесса развёртывания и конфигурирования в стандартный вывод задания будут помещены IP-адреса для подключения к запущенным виртуальным машинам. Используя выданные IP-адреса, пользователь (или запущенная им прикладная программная система) подключаются к виртуальным машинам и производят прикладные вычисления.

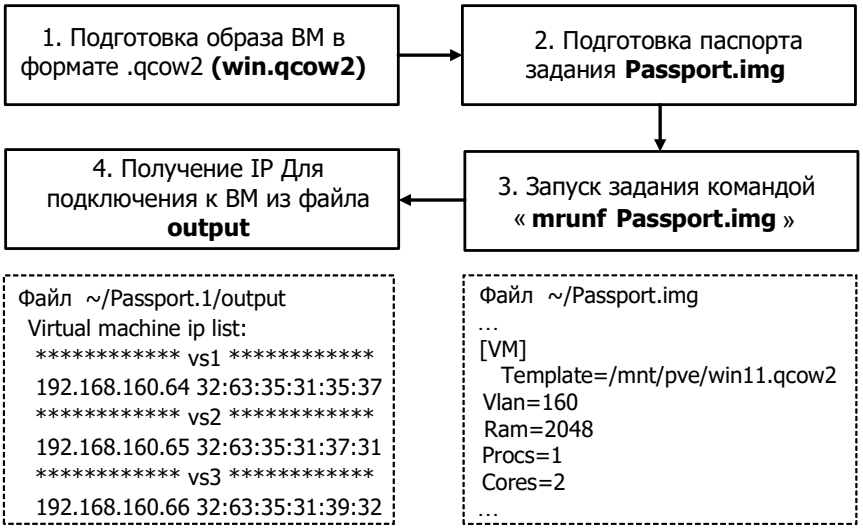

Рис. 3. Процесс подготовки и запуска нестандартного задания

Разработанный макет прошёл опытную эксплуатацию на нескольких прикладных геофизических расчётных задачах, часть из которых рассчитывалась традиционным путём (в связке «Linux+MPI»), часть задач рассчитывалась в гостевой ОС Windows. Гостевая ОС развёртыва-

лась на выделенных через СУППЗ суперкомпьютерных ресурсах из соответствующего образа виртуальной машины.

3. Методы автоматической настройки сетевого оборудования для динамической организации виртуальной сетевой среды

Для возможности подключения к выполняющемуся нестандартному заданию, а также для обеспечения взаимодействия запущенных виртуальных машин из состава задания, необходимо автоматически настроить сетевое оборудование и организовать для задания виртуальную сетевую среду.

Попытки реализации систем с частично схожим назначением предпринимались неоднократно. Так, работа [4] посвящена системе конфигурации VLAN для нового оборудования или при повторном подключении используемого. В контексте нашей задачи интерес представляет настройка виртуальной сетевой среды на уже существующей физической топологии без её изменения. В работе [5] рассматривается изменение конфигурации VLAN при миграции виртуального узла между двумя физическими серверами с соответствующей настройкой сети. В статье [6] тема проброса портов типа Trunk рассмотрена с точки зрения взаимодействия автоматически создаваемой подсети с протоколом STP (Spanning Tree Protocol), что не позволяет решить поставленную нами задачу в полном объёме. Одна из последних работ в этой области [7] в основном посвящена настройке виртуальных узлов, а именно виртуальных сетевых карт, и не в достаточной мере описывает настройки сетевых компонентов общей конфигурации.

Рассмотрим известные на сегодняшний день способы и средства автоматизации настройки сетевого оборудования.

Первым способом является использование специализированного ПО, например, HPE Intelligent Management Center [8]. Как правило, подобное ПО имеет интуитивно понятный дружелюбный интерфейс, но не обладает достаточной гибкостью, позволяя решать лишь набор типовых задач.

Второй способ предполагает использование протокола SNMP, например, утилиты snmp-utils. Способ обладает большей гибкостью, но включает процедуру настройки VLAN по SMTP, эти процедуры существенно отличаются для разного сетевого оборудования.

Третий способ представляется авторам наиболее простым и в то же время универсальным. Способ заключается в автоматизации ввода известных команд настройки сетевого оборудования и подразумевает разработку соответствующих командных сценариев (скриптов). На рисунке 4 приведены примеры возможных сценариев запуска на вычислительном кластере нестандартного задания, состоящего из взаимодействующих виртуальных машин.

В сценарии а) запускается набор виртуальных машин, к которым реализуется на время выполнения задания доступ извне.

В сценарии б) не требуется интерактивной работы с виртуальными машинами, но запущенный набор виртуальных машин должен получить некоторые исходные данные из СХД и записать в СХД результат.

Сценарий в) иллюстрирует задание, в котором используется доступ к СХД, интерактивная работа и некоторые дополнительные коммуникации между виртуальными машинами по отдельной виртуальной сети, например, высокоскоростной типа Infiniband.

В сценарии г) имеется постоянно функционирующая доступная из внешних сетей виртуальная машина, работающая с СХД. Именно по её требованию на вычислительном кластере запускается набор виртуальных машин, с которыми она взаимодействует.

Во всех сценариях необходимо наличие виртуальной локальной сети для внутренних коммуникаций виртуальных машин. В сценарии в) для этого предусмотрены несколько сетей, к которым подключены группы виртуальных машин.

В сценариях а), в), г) для обеспечения интерактивной работы с запущенными виртуальными машинами на одной из машин должен быть интерфейс для взаимодействия с удалённым пользователем через внешние сети, например, Интернет. Для доступа извне для этого интерфейса может быть определён реальный (публичный) IP-адрес или назначен приватный IP-адрес с пробросом портов через NAT. Возможна также инициализация VPN из виртуальной машины

на внешний сервер. Интерфейсы для внешнего доступа могут создаваться не только для одной, но и для нескольких виртуальных машин, кроме того, виртуальная локальная сеть для внутренних коммуникаций может быть использована и для доступа извне ко всем виртуальным машинам.

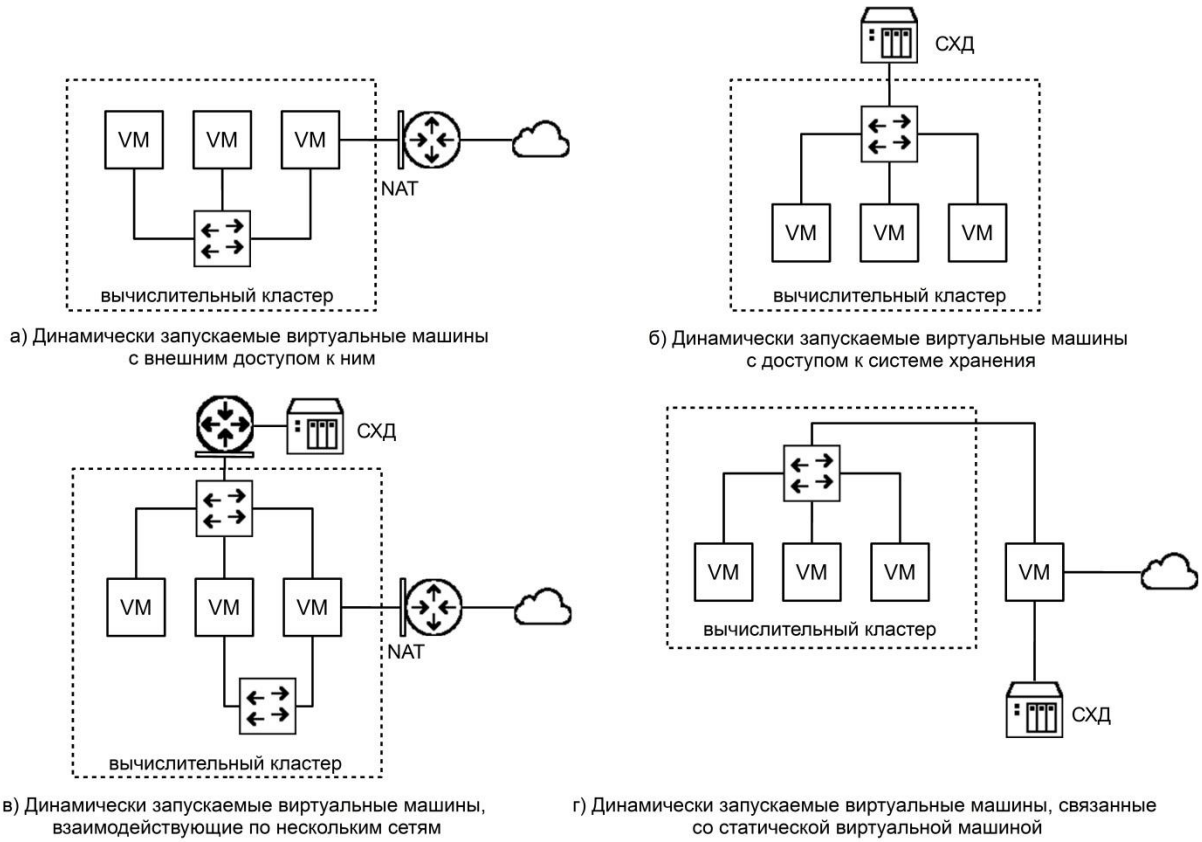

Рис. 4. Примеры возможных сценариев запуска виртуальных машин

Сценарии б), в), г) предусматривают обмен данных между виртуальными машинами и СХД по IP-протоколу. Для этого виртуальные машины и система хранения должны иметь интерфейсы в одной сети (сценарий б)), или должна быть организована маршрутизация между сетью виртуальных локальных машин и СХД (сценарий в)).

Сценарий г) отличается от остальных тем, что внешний интерфейс не требуется поднимать на динамически выделяемых виртуальных машинах, внешнее взаимодействие организуется исключительно с постоянно функционирующей виртуальной машиной.

Для реализации запуска заданий, состоящих из наборов виртуальных машин, на вычислительном кластере необходимо решить следующие задачи:

1) обеспечить нумерацию сетевых ресурсов, выделяемых наборам виртуальных машин;

2) передать информацию о сетевой нумерации в виртуальные машины при их развёртывании на кластере (назначить IP-адреса и виртуальные локальные сети на интерфейсы, задать маршрутизацию и т.д.);

3) определить, как описать информацию о внутренней сетевой организации набора виртуальных машин в описании (паспорте) задания, запускаемого на кластере;

4) отобразить внутреннюю сетевую организацию задания на связывающую выделенные заданию вычислительные модули реальную физическую сетевую инфраструктуру вычислительного кластера, автоматически настроить эту инфраструктуру перед запуском задания.

Нумерация сетевых ресурсов должна исключить пересечение IP-адресов, номеров виртуальных сетей и прочих атрибутов, используемых для взаимодействия с внешним миром, у одновременно выполняемых нестандартных заданий. Следовательно, номера сетевых ресурсов (IP-адреса, номера виртуальных сетей) должны динамически назначаться из некоторого пула свободных ресурсов нумерации и по завершении задания возвращаться в него. Исключение со-

ставляют динамически запускаемые виртуальные машины, взаимодействующие с постоянно работающей (статической) виртуальной машиной.

Выделенные из пула IP-адреса должны быть назначены интерфейсам виртуальных машин, также на них должны быть заданы маршруты по умолчанию (в используемые подсети). Если СХД взаимодействует с виртуальными машинами внутри одной виртуальной локальной сети, как в сценарии б), то на СХД тоже должен быть создан соответствующий виртуальный интерфейс, и назначен IP-адрес. Если СХД подключена через маршрутизатор, виртуальный интерфейс с соответствующим IP-адресом должен быть создан на маршрутизаторе. Аналогичным образом должен настраиваться интерфейс на маршрутизаторе при взаимодействии с внешним миром (сценарии а) и в)).

Настройки интерфейсов СХД и маршрутизаторов могут быть выполнены командными сценариями (скриптами). Настройка виртуальных машин (IP-адресов и маршрутов) может быть выполнена с использованием DHCP-сервера в каждой из виртуальных сетей.

Описание внутренней сетевой организации набора виртуальных машин должно содержать список виртуальных машин и внешних устройств, с которыми взаимодействуют виртуальные машины (системы хранения, маршрутизаторы и т.д.), список их сетевых интерфейсов и сетей, их объединяющих.

Для описания внутренней сетевой организации набора виртуальных машин в паспорте задания может использоваться язык разметки сетей Network Markup Language [4].

При запуске нестандартного задания, состоящего из набора виртуальных машин, на конкретных вычислительных модулях кластера необходимо выполнить настройку сетевой инфраструктуры: организовать виртуальные локальные сети между модулями в соответствии с описанием внутренней сетевой организации набора виртуальных машин в паспорте задания.

Для этого надо поставить в соответствие виртуальные сети разворачиваемых наборов виртуальных машин физической топологии реальной сети вычислительного кластера.

Сеть Ethernet вычислительного кластера представляет собой дерево, в котором листья представлены вычислительными модулями кластера, промежуточные вершины и корень — коммутаторами. Каждая виртуальная машина разворачивается на отдельном вычислительном модуле. Задача построения объединяющей эти машины виртуальной сети сводится к построению минимального поддерева, в которое входят выделенные заданию листья-модули (рисунок 5).

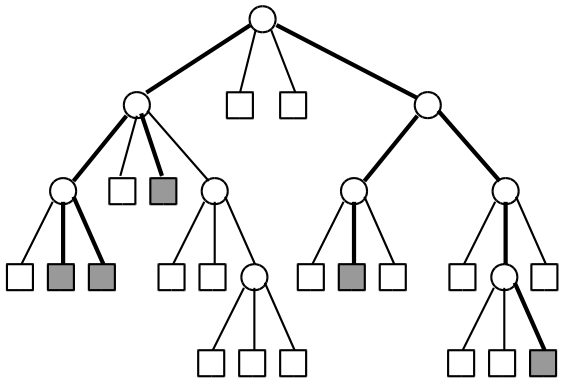

Рис. 5. Представление сети в виде дерева, серым закрашены выделенные нестандартному заданию вычислительные модули

Следующий алгоритм обеспечивает построение искомого поддерева. Выбирается некоторая исходная помеченная вершина (вершина 1 на рисунке 6). Перебором в ширину все рёбра и вершины нумеруются в соответствии с расстоянием от исходной вершины.

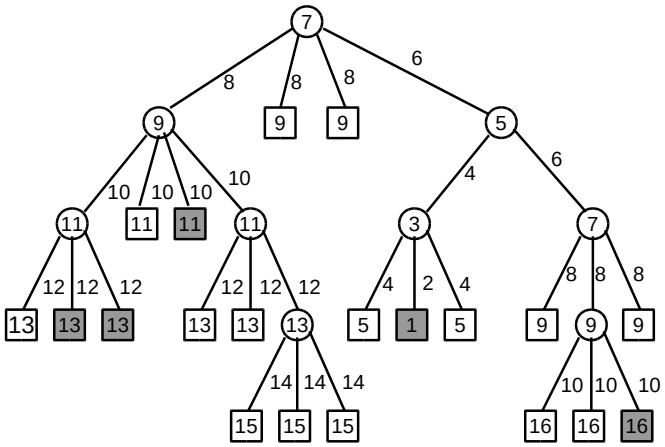

Рис. 6. Алгоритм построения минимального поддерева для виртуальной сети: нумерация вершин

Теперь от каждой помеченной вершины строится путь в сторону исходной вершины 1: в каждой вершине выбирается ребро с номером, на единицу меньшим номера вершины, в каждом ребре – вершина с номером, на единицу меньшим номера ребра. При этом все выбранные на пути вершины и рёбра помечаются (рисунок 7).

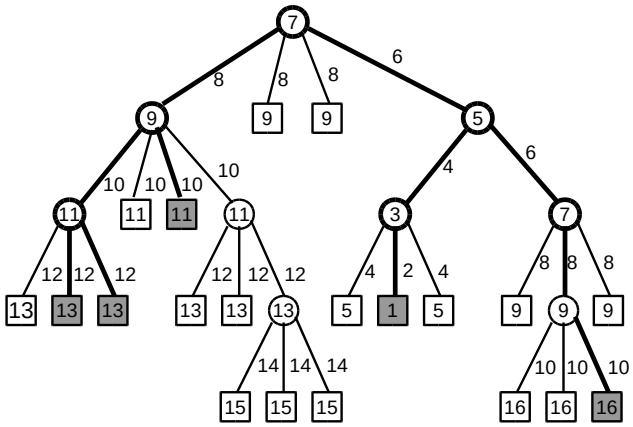

Рис. 7. Алгоритм построения минимального поддерева для виртуальной сети: построение путей к исходной вершине

Помеченные таким образом вершины (коммутаторы и узлы) и рёбра (линии связи) образуют граф требуемой виртуальной сети и определяют список коммутаторов и портов, на которых настраивается виртуальная локальная сеть (VLAN).

Собственно настройка коммутаторов производится соответствующими командными сценариями (скриптами) при запуске и завершении задания.

Если набору виртуальных машин нужна высокоскоростная коммуникационная сеть Infiniband, в качестве аналога виртуальных сетей можно использовать разделы (partition). В настройках Subnet Manager можно задать принадлежность узлов разделам. Кроме того, существует возможность организации прозрачного моста между виртуальными локальными сетями Ethernet (VLAN) и разделами коммуникационной сети Infiniband. Этот метод [5] опробован и с успехом используется в суперкомпьютере МВС-10П в МСЦ РАН.

4. Оценка влияния средств виртуализации на производительность

После реализации макета авторами была произведена оценка влияния средств виртуализации KVM и Proxmox VE, установленных на вычислительных модулях кластера, на производительность заданий, выполняемых на стандартной программной платформе, развёрнутой на ВМ кластера по умолчанию. Стандартная программная платформа включает в себя ОС Linux и коммуникационную библиотеку MPI. Оценка влияния применённых средств виртуализации на

производительность MPI-программ осуществлялась с использованием стандартных тестов NAS Parallel Benchmarks 3.3 (NPB).

Тестирование выполнялось в два этапа. На первом этапе тесты NPB запускались на вычислительных модулях под управлением ОС Linux Debian Jessie. На втором этапе выполнялся запуск тестов NPB на тех же модулях, но с установленными программными компонентами KVM и Proxmox VE. Коммуникационные обмены между MPI-процессами в обоих случаях выполнялись через сеть InfiniBand в режиме IPoIB. Для каждого приложения NPB определялась максимальная величина Mop/s на первом (синий столбец) и втором (красный столбец) этапах тестирования (рисунок 8). Поскольку максимальное значение Mop/s каждого теста NPB одинаково для первого и второго этапов тестирования, можно сделать вывод, что компоненты KVM и Proxmox VE не оказывают существенного влияния на производительность стандартных заданий.

Для оценки влияния виртуальной среды на производительность MPI-программ была проведена серия экспериментов, в ходе которых тесты NPB запускались в виртуальных машинах KVM. В МСЦ РАН уже проводились экспериментальные исследования подобного влияния [6, 7], в настоящей работе ранее полученные результаты были подтверждены в рамках исследуемого макета облачной среды для высокопроизводительных приложений.

На каждом вычислительном модуле макета была запущена только одна виртуальная машина. Коммуникационные обмены между MPI-процессами тестов NPB выполнялись через сеть InfiniBand в режиме «IP over IB» (IPoIB), что согласно результатам работы [7] является самым затратным с точки зрения накладных расходов режимом использования сети Infiniband. Полученные максимальные в виртуальной среде значения Mop/s для каждого теста (серый столбец) сравнивались с результатами выполнения тех же тестов на вычислительных модулях в режиме InfiniBand (синий столбец) и InfiniBand IPoIB (красный столбец) (рисунок 9).

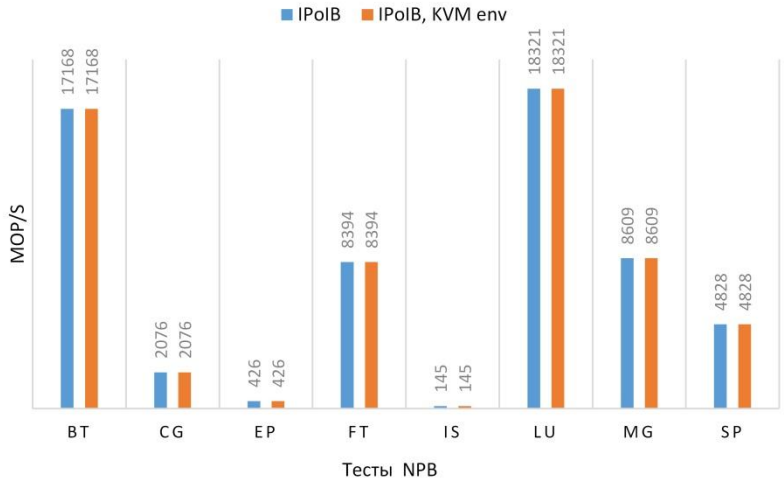

Рис. 8. Оценка влияния компонентов KVM и Proxmox VE на производительность тестов NPB в составе стандартных заданий

На рисунке 10 видно, что в режиме InfiniBand IPoIB максимальные потери производительности, в сравнении со стандартным режимом InfiniBand, достигаются в тестах NPB с преобладанием коммуникационных обменов между MPI-процессами. При сравнении результатов тестирования в режиме работы InfiniBand IPoIB и в виртуальной среде KVM, видно, что наибольшие потери в производительности достигаются на тестах LU (9%), IS (5%), SP (4%) и связаны с накладными расходами виртуализации KVM на коммуникационные обмены. Для остальных тестов потери в среднем не превышают 3%.

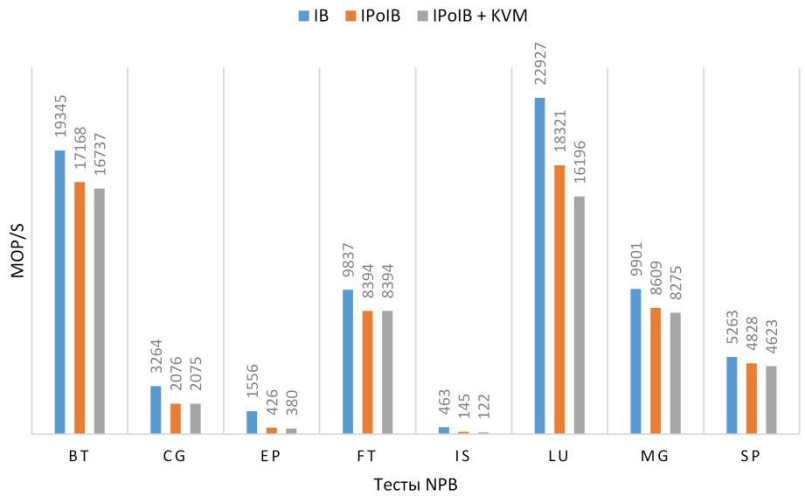

Рис. 9. Оценка влияния виртуальной среды KVM на выполнение тестов NPB

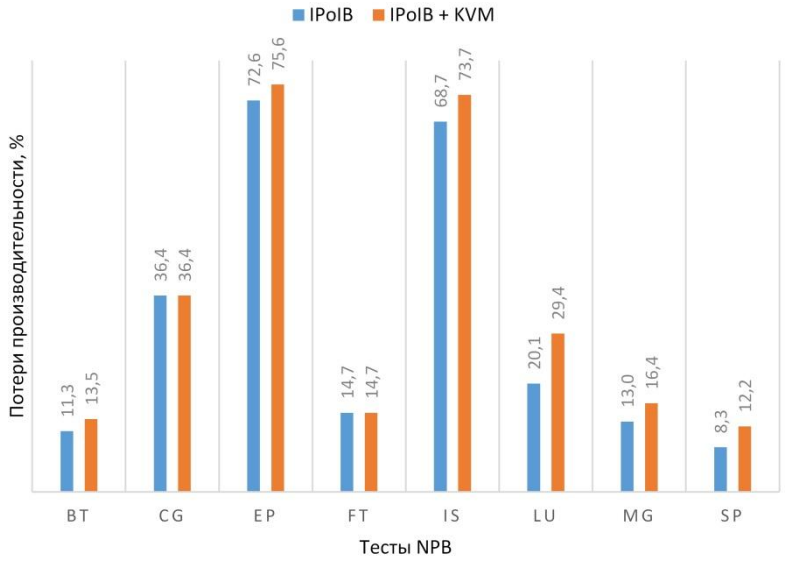

Рис. 10. Потери производительности при использовании InfiniBand IPoIB и виртуальной среды KVM в сравнении с вычислением на модулях в стандартном режиме InfiniBand

5. Заключение

Для современных суперкомпьютерных центров коллективного пользования актуальной является задача обработки входного потока пользовательских заданий, в котором совмещаются так называемые стандартные и нестандартные задания. Стандартные задания выполняются в рамках развёрнутой по умолчанию программной среды, нестандартные задания требуют для своего выполнения отдельной программной платформы, с отдельными операционной системой, прикладным и инструментальным ПО. Созданный в МСЦ РАН макет облачной среды для высокопроизводительных приложений предоставляет возможность смешанного входного потока заданий за счёт представления нестандартного задания в виде набора автоматически разворачиваемых при старте задания виртуальных машин. При этом вычислительные модули для запуска набора виртуальных машин динамически выделяются через систему управления прохождением параллельных заданий (СУППЗ), являющуюся отечественным аналогом таких известных систем, как SLURM, PBS, Moab и т.п. В рамках работ по созданию макета авторами были разработаны методы автоматической настройки сетевого оборудования для динамической организации виртуальной сетевой среды запускаемого на созданном макете нестандартного задания, состоящего из набора виртуальных машин, разворачиваемых на динамически выделяемых

заданию вычислительных модулях. Проведённые авторами эксперименты показали, что применённые при создании макета средства виртуализации KVM и Proxmox VE не оказывают значительного влияния на производительность стандартных заданий, что позволяет успешно совмещать обработку стандартных и нестандартных заданий в одном входном потоке.

Литература

1. Баранов А.В., Киселёв А.В., Старичков В.В., Ионин Р.П., Ляховец Д.С. Сравнение систем пакетной обработки с точки зрения организации промышленного счета. Научный сервис в сети Интернет: поиск новых решений: Труды Международной суперкомпьютерной конференции (17-22 сентября 2012 г., г. Новороссийск). М.: Изд-во МГУ, 2012. С. 506-508. URL: http://agora.guru.ru/abrau2012/pdf/506.pdf (дата обращения: 12.04.2017).
2. Баранов А.В., Ляховец Д.С. Сравнение качества планирования заданий в системах пакетной обработки SLURM и СУППЗ. Научный сервис в сети Интернет: все грани параллелизма: Труды Международной суперкомпьютерной конференции (23-28 сентября 2013 г., г. Новороссийск). М.: Изд-во МГУ, 2013. С. 410-414. URL: http://agora.guru.ru/abrau2013/pdf/410.pdf (дата обращения: 12.04.2017).
3. Ken Hess. Proxmox: the Ultimate Hypervisor, Jul. 2011. URL: http://www.zdnet.com/article/proxmox-the-ultimate-hypervisor (дата обращения: 12.04.2017).
4. Iijima A., Yamamoto Y. System and method for automatically setting VLAN configuration information, 24 Apr. 2001. URL: https://www.google.com/patents/US6223218 (дата обращения: 30.05.2017).
5. Jaiswal, R.K., Kuroda, A., Prissel, L.P., Sealy, C.R., Seth, E. Automated network configuration in a dynamic virtual environment, 07 Feb. 2013. URL: https://www.google.com/patents/US20130034015 (дата обращения: 30.05.2017).
6. Li F., Yang J., An C., Wu J., and Wang X. (2015), Towards centralized and semi-automatic VLAN management, Int. J. Network Mgmt, 25, 52–73, doi: 10.1002/nem.1884
7. Thai, C. Methods and apparatus for automatic configuration of virtual local area network on a switch device, 25 Oct. 2016. URL: https://www.google.com/patents/US9479397 (дата обращения: 30.05.2017).
8. Network Management Systems, Solutions & Services - Information Management Center, May 2017. URL: https://www.hpe.com/us/en/networking/management.html (дата обращения: 30.05.2017).
9. van der Ham J., Dijkstra F., Łapacz R., Zurawski J. Network Markup Language Base Schema version 1. Open Grid Forum, 2013, URL: https://www.ogf.org/documents/GFD.206.pdf (дата обращения: 12.04.2017).
10. Mellanox Virtual Protocol Interconnect® Creates the Ideal Gateway Between InfiniBand and Ethernet. URL: http://www.mellanox.com/related-docs/case_studies/CS_VPI_GW.pdf (дата обращения: 12.04.2017).
11. Аладышев О.С., Баранов А.В., Ионин Р.П., Киселев Е.А., Орлов В.А. Сравнительный анализ вариантов развертывания программных платформ для высокопроизводительных вычислений. Научный сервис в сети Интернет: многообразие суперкомпьютерных миров: Труды Международной суперкомпьютерной конференции (22-27 сентября 2014 г., г. Новороссийск). М.: Изд-во МГУ, 2014. С. 349-354. URL: http://agora.guru.ru/abrau2014/pdf/349.pdf (дата обращения: 12.04.2017).
12. Баранов А.В., Николаев Д.С. Использование контейнерной виртуализации в организации высокопроизводительных вычислений // Программные системы: теория и приложения. 2016. № 7:1(28). С. 117–134. URL: http://psta.psiras.ru/read/psta2016_1_117-134.pdf

Methods of Deployment of Software Platforms and Virtualization of Networks for Running Parallel Jobs

B.M. Shabanov, A.P. Ovsyannikov, A.V. Baranov, O.S. Aladyshev, E.A. Kiselev, Y.O. Zhukov

Joint Supercomputer Center of the Russian Academy of Sciences - Branch of Federal State Institution «Scientific Research Institute for System Analysis of the Russian Academy of Sciences», Moscow, Russia

The article is devoted to the problem of managing the flow of parallel jobs, which demand detached software platforms and reconfiguration of networks. In such environment, software platforms represented by a set of virtual machines running in parallel and dynamically allocated by the high performance cluster batch system. Virtual machines are joined by user configured virtual networks. The authors suggest methods to organize a virtual network subsection of cluster, including way of describing the internal network structure and the method of mapping virtual machines to the physical network infrastructure. A method for automatically configuring the supercomputer's network equipment at the start and completion of tasks is considered.

Keywords: virtualization in HPC, cluster workload management, virtual network

References

1. Baranov A.V., Kiselev A.V., Starichkov V.V., Ionin R.P., Lyakhovets D.S. Sravnenie sistem paketnoj obrabotki s tochki zrenija organizacii promyshlennogo scheta [Comparison of Workload Management Systems from the Point of View of Organizing an Industrial Computing]. Nauchnyj servis v seti Internet: poisk novyh reshenij: Trudy mezhdunarodnoy superkomp'yuternoy konferentsii (Novorossiysk, 17-22 sentyabrya 2012 g.) [Scientific Services & Internet: Search for New Solutions: Proceedings of the International Supercomputing Conference (Novorossiysk, Russia, September, 17–22, 2012)]. Moscow, Publishing of Lomonosov Moscow State University, 2012. P. 506–508. URL: http://agora.guru.ru/abrau2012/pdf/506.pdf (accessed: 12.04.2017).
2. Baranov A.V., Lyakhovets D.S. Sravnenie kachestva planirovanija zadanij v sistemah paketnoj obrabotki SLURM i SUPPZ [Comparison of the Quality of Job Scheduling in Workload Management Systems SLURM and SUPPZ]. Nauchnyj servis v seti Internet: vse grani parallelizma: Trudy mezhdunarodnoy superkomp'yuternoy konferentsii (Novorossiysk, 23-28 sentyabrya 2013 g.) [Scientific Services & Internet: All Facets of Parallelism: Proceedings of the International Supercomputing Conference (Novorossiysk, Russia, September, 23–28, 2013)]. Moscow, Publishing of Lomonosov Moscow State University, 2013. P. 410–414. URL: http://agora.guru.ru/abrau2013/pdf/410.pdf (accessed: 12.04.2017).
3. Ken Hess. Proxmox: the Ultimate Hypervisor, Jul. 2011. URL: http://www.zdnet.com/article/proxmox-the-ultimate-hypervisor (accessed: 12.04.2017).
4. Iijima A., Yamamoto Y. System and method for automatically setting VLAN configuration information, 24 Apr. 2001. URL: https://www.google.com/patents/US6223218 (accessed: 30.05.2017).
5. Jaiswal, R.K., Kuroda, A., Prissel, L.P., Sealy, C.R., Seth, E. Automated network configuration in a dynamic virtual environment, 07 Feb. 2013. URL: https://www.google.com/patents/US20130034015 (accessed: 30.05.2017).
6. Li F., Yang J., An C., Wu J., and Wang X. (2015), Towards centralized and semi-automatic VLAN management, Int. J. Network Mgmt, 25, 52–73, doi: 10.1002/nem.1884

7. Thai, C. Methods and apparatus for automatic configuration of virtual local area network on a switch device, 25 Oct. 2016. URL: https://www.google.com/patents/US9479397 (accessed: 30.05.2017).
8. Network Management Systems, Solutions & Services - Information Management Center, May 2017. URL: https://www.hpe.com/us/en/networking/management.html (accessed: 30.05.2017).
9. van der Ham J., Dijkstra F., Łapacz R., Zurawski J. Network Markup Language Base Schema version 1. Open Grid Forum, 2013, URL: https://www.ogf.org/documents/GFD.206.pdf (accessed: 12.04.2017).
10. Mellanox Virtual Protocol Interconnect® Creates the Ideal Gateway Between InfiniBand and Ethernet. URL: http://www.mellanox.com/related-docs/case_studies/CS_VPI_GW.pdf (accessed: 12.04.2017).
11. Aladyshev O.S., Baranov A.V., Ionin R.P., Kiselev E.A., Orlov V.A. Sravnitel'nyj analiz variantov razvertyvanija programmnyh platform dlja vysokoproizvoditel'nyh vychislenij [Comparative Analysis of Variants of Deployment of Program Platforms for High Performance Computing]. Nauchnyj servis v seti Internet: mnogoobrazie superkomp'juternyh mirov: Trudy mezhdunarodnoy superkomp'yuternoy konferentsii (Novorossiysk, 22-27 sentyabrya 2014 g.) [Scientific Services & Internet: Variety of Supercomputing Worlds: Proceedings of the International Supercomputing Conference (Novorossiysk, Russia, September, 22–27, 2014)]. Moscow, Publishing of Lomonosov Moscow State University, 2014. P. 349–354. URL: http://agora.guru.ru/abrau2014/pdf/349.pdf (accessed: 12.04.2017).
12. Baranov A.V., Nikolaev D.S. Ispol'zovanie kontejnernoj virtualizacii v organizacii vysokoproizvoditel'nyh vychislenij [The Use of Container Virtualization in the Organization of High-Performance Computing] // Program systems: theory and applications. 2016. No. 7:1(28). P. 117–134. (In Russian) URL: http://psta.psiras.ru/read/psta2016_1_117-134.pdf (accessed: 12.04.2017).