ISSN 1992-6502 (Print) Вестник УГАТУ ISSN 2225-2789 (Online)
2014. Т. 18, № 3 (64). С. 295–300 http://journal.ugatu.ac.ru

УДК 004.031.2, 004.382.2

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ВАРИАНТОВ РАЗВЕРТЫВАНИЯ ПРОГРАММНЫХ ПЛАТФОРМ ДЛЯ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛЕНИЙ

О. С. Аладышев[1], А. В. Баранов[2], Р. П. Ионин[3], Е. А. Киселев[4], В. А. Орлов[5]

[1] aladyshev@jscc.ru, [2] antbar@mail.ru, [3] rimsk-i@ya.ru, [4] kiselev@jscc.ru, [5] repetitrf@mail.ru

ФГБУН «Межведомственный суперкомпьютерный центр РАН» (МСЦ РАН)

Поступила в редакцию 17 августа 2014 г.

Аннотация. Рассматриваются следующие варианты развертывания программных платформ на ресурсах супер-ЭВМ: из образа диска, сетевая загрузка образа платформы и загрузка в виде виртуальной машины под управлением различных гипервизоров (ESXi, KVM, Hyper-V). Представлены результаты экспериментов по развертыванию программных платформ для каждого из рассмотренных вариантов. В ходе экспериментов измерялись как время развертывания платформ для каждого варианта, так и времена выполнения на развернутой платформе стандартных тестов Linpack и NAS Parallel Benchmarks.

Ключевые слова: высокопроизводительные вычисления; сетевая загрузка; гипервизор; ESXi; KVM; Hyper-V; Linpack; NPB.

Современные системы высокопроизводительных вычислений работают под управлением специального программного обеспечения – систем пакетной обработки (СПО). 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.

Обычно СПО ориентированы на ведение разных очередей для разных типов так называемых **стандартных заданий**. Стандартные задания выполняются в развернутой в СПО программной среде и не требуют внесения изменений в процесс конфигурации вычислительных модулей. Однако в современных суперкомпьютерных центрах в последние годы появляются **нестандартные задания**, предъявляющие особые требования к ресурсам. Например, подобным заданиям могут потребоваться своя аппаратно-программная платформа, программные пакеты и лицензии, то есть своя программная среда. Для СПО возникает новая задача – одновременное планирование и выполнение стандартных и нестандартных заданий разных типов на одной вычислительной установке. Другими словами, необходимы средства управления развертыванием программных платформ для одновременного исполнения стандартных и нестандартных заданий.

Для организации выполнения нестандартных заданий на решающем поле высокопроизводительного вычислительного кластера необходимо выполнить следующие действия.

1. Выделить ресурсы из решающего поля кластера.
2. Развернуть на выделенных модулях соответствующие программные платформы.
3. Выполнить нестандартное задание на развернутой программной платформе.
4. Вернуть выделенные ресурсы обратно.

Развертывание платформ может производиться несколькими способами.

1. Платформа каждый раз устанавливается заново из образа программной платформы для вычислительных модулей, созданного с использованием специального ПО (создан образ жесткого диска, подготовлен сценарий автоматической установки и т. д.).
2. Производится сетевая загрузка программной платформы для вычислительных модулей, при которой корневая файловая система дос-

Рекомендована XVI Международной суперкомпьютерной конференцией «Научный сервис в сети Интернет: многообразие суперкомпьютерных миров» (Абрау, 22–27 сентября 2014 г.).

тупна по сети с некоторого сервера (например, NFS) или передается по сети и располагается в оперативной памяти вычислительного модуля.

3. На вычислительные модули устанавливается гипервизор, и используются виртуальные машины с необходимыми программными платформами.

При развертывании программных платформ из образов жесткого диска на типовом модуле производится установка и настройка программной платформы. С помощью специального программного обеспечения (например, Clonezilla [1]) снимается образ платформы, который можно использовать для ее развертывания на остальных модулях вычислительной системы. Обычно ПО для создания образа жесткого диска содержит в себе и средства для развертывания этих образов. Созданные образы можно сохранять в некоторую библиотеку образов на некотором внешнем хранилище и повторно использовать в дальнейшем.

Сетевая загрузка на основе PXE [2] предполагает загрузки операционных систем вычислительных модулей с помощью сетевой карты без использования дополнительных устройств (жестких дисков, компакт-дисков, устройств на основе флеш-памяти и т. д.). PXE-код, находящийся в памяти сетевой карты, получает загрузчик из сети и передает ему управление. С помощью сетевых служб инфраструктуры класстера (DHCP-сервера, TFTP-сервера и других) определяются параметры загружаемых операционных систем вычислительных модулей. Монтирование корневой файловой системы по сети невозможно для ОС семейства MS Windows.

Технология виртуализации позволяет разместить несколько виртуальных серверов на одном физическом. На одном сервере можно запускать виртуальные машины, которые полностью воспроизводят работу независимых физических серверов. Наиболее распространенным типом является виртуализация с помощью гипервизора. В этом случае часть вычислительных модулей кластера виртуализуется с использованием какой-либо платформы виртуализации (например, Xen [3], KVM [4], vSphere [5]). Развертывание программной среды на виртуализованных ресурсах будет включать в себя подготовку виртуальных машин с необходимым программным обеспечением.

Все три рассматриваемых варианта развертывания программных платформ можно применять в современных центрах высокопроизводительных вычислений для загрузки операционных систем на вычислительные модули кластеров. Необходимо определить, какие накладные расходы привносит применение каждого из рассматриваемых вариантов на выполнение параллельных заданий. Среди существенных характеристик выполнения нестандартных заданий можно выделить следующие:

- время конфигурирования модулей и развертывания программных платформ;
- время выполнения параллельной программы;
- время свертывания программной платформы и возвращения модулей в исходное состояние.

Возвращением вычислительного модуля в исходное состояние для каждого из рассматриваемых вариантов развертывания программных платформ будем считать загрузку исходной программной платформы с использованием этого же варианта развертывания, то есть для варианта с образом жесткого диска – это восстановление исходного образа диска, для сетевой загрузки – сетевая загрузка модуля с исходным корневым каталогом, для платформ виртуализации – запуск виртуальной машины с исходной программной платформой.

Свертывание программной платформы как таковое имеет место только для физических вычислительных модулей. Для сетевой загрузки – это перезагрузка с новым ядром, стартовым RAM-диском и адресом для монтирования корневого каталога (время перезагрузки одного модуля с сетевой загрузкой), для гипервизоров – выключение виртуальной машины и включение новой (время перезагрузки виртуальной машины).

Развертывание программных платформ с использованием образа жесткого диска включает измерение времени следующих операций:

- времени развертывания программной платформы на типовом вычислительном модуле – автоматизированная установка ОС на одном из модулей (при необходимости);
- времени снятия образа жесткого диска модуля с установленной программной платформой;
- времени восстановления сохраненного образа жесткого диска на жесткий диск другого вычислительного модуля;
- времени перезагрузки модуля с программной платформой.

Свертывание (и, следовательно, затрачиваемое время) программных платформ из образов жестких дисков будет состоять из:

- снятия образа текущего жесткого диска (при необходимости);
- восстановления сохраненного исходного образа жесткого диска;
- перезагрузки модуля в исходную программную платформу.

Развертывание программных платформ с использованием сетевой загрузки включает измерение времени следующих операций:

- времени развертывания программной платформы на типовом вычислительном модуле – автоматизированная установка ОС на одном из модулей;
- времени перезагрузки модуля с программной платформой.

Свертывание программных платформ с использованием сетевой загрузки будет состоять только из перезагрузки модуля в исходную программную платформу.

Развертывание программных платформ с использованием платформ виртуализации включает измерение времени следующих операций:

- времени развертывания программной платформы на типовом вычислительном модуле – автоматизированная установка ОС на одном из модулей;
- времени загрузки образа виртуальной машины на гипервизор;
- времени установки гипервизора на вычислительный модуль (при необходимости);
- времени запуска виртуальной машины с программной платформой.

Свертывание программных платформ с использованием платформ виртуализации будет состоять из перезагрузки виртуальной машины с исходной программной платформой.

Для того чтобы определить накладные расходы при выполнении параллельного задания на развернутой программной платформе, целесообразно использовать стандартные тесты для высокопроизводительных вычислительных систем (Linpack [6], NPB [7]). Программы из этих тестов можно запускать с различными параметрами, которые можно подобрать индивидуально для каждой системы высокопроизводительных вычислений в зависимости от ее аппаратной конфигурации.

Экспериментальные сравнения производились на стенде, состоящем из четырех идентичных вычислительных модулей из состава российской супер-ЭВМ МВС-100К [8]. Каждый из четырех вычислительных модулей HP ProLiant BL2x220c G5 включает в себя следующее аппаратное обеспечение:

- два процессора Quad-Core Intel Xeon E5450 (8 ядер);
- 8 Гб оперативной памяти;
- жесткий диск 110 Гб;
- три сетевых порта (два – 1 Гб/с Ethernet, один – 10 Гб/с InfiniBand).

Процессор E5450 не поддерживает технологию VT-d, поэтому передать виртуальным машинам сетевые карты InfiniBand напрямую нельзя. Запуск MPI-программ из тестов осуществлялся через Ethernet. Программная среда была развернута на базе ОС CentOS 6.5 и MPI – mpich2.

Развертывание программных платформ производилось с использованием автоматизированной установки на основе сценария kickstart [9], для создания корневой файловой системы для PXE загрузки использовалась групповая установка базового окружения из менеджера пакетов yum.

В варианте с выделением заданию виртуальных вычислительных модулей были проведены эксперименты с несколькими гипервизорами: ESXi 5.5, Hyper-V 2012 R2 [10], KVM. Установка гипервизора Hyper-V проводилась в ручном режиме, так как автоматизированная установка невозможна в Linux-среде и требует дополнительного настроенного сервера Windows Server. При загрузке виртуальных машин для ESXi и KVM размер передаваемого по сети файла виртуального жесткого диска брался равным максимальному размеру диска, который был задан при создании виртуальной машины. У гипервизора KVM рассматривалась работа виртуальных машин в режиме паравиртуализации (PV) и полной аппаратной виртуализации (HV).

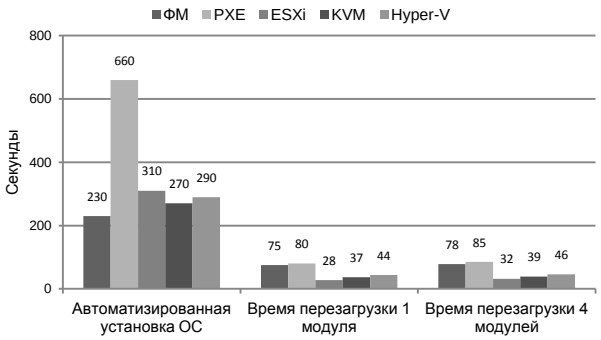

Рис. 1. Время установки ОС и перезагрузки модулей для физических модулей (ФМ), для варианта сетевой загрузки (PXE) и для виртуальных машин под управлением гипервизоров ESXi, KVM и Hyper-V

Для определения возможной зависимости времени загрузки от числа модулей (в частности, для сетевой PXE, в случае которой корневые каталоги модулей находились на одном сервере) измерялось время перезагрузки одного модуля и четырех модулей.

Создание образа жесткого диска модуля с программной платформой, фактически занимающей 1 Гб дискового пространства, с помощью ПО Clonezilla заняло 330 секунд (≈ 5,5 минут). Размер созданного образа – 500 Мб. Время восстановления этого образа на другой вычислительный модуль составило 200 секунд (≈ 3,5 минуты). Результаты загрузки представлены на рис. 1–2.

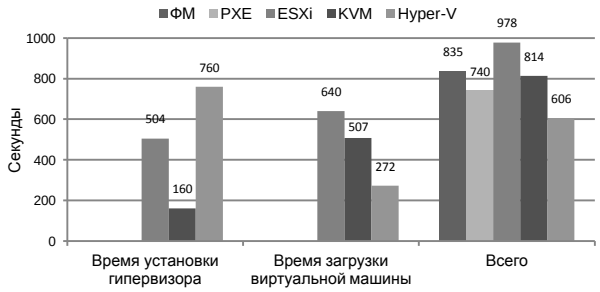

Рис. 2. Время развертывания программной платформы для физических модулей (ФМ), для варианта сетевой загрузки (PXE) и для виртуальных машин под управлением гипервизоров ESXi, KVM и Hyper-V

Разница по времени между загрузкой одного и четырех вычислительных модулей составила не более 5 секунд. Так как в составе стенда находятся только четыре вычислительных модуля, то в данном случае сложно говорить о зависимости между временем загрузки нескольких вычислительных количеством загружаемых модулей.

Время перезагрузки виртуализованных модулей меньше времени перезагрузки физических и сетевых модулей, так как при перезагрузке виртуальных машин не происходит инициализации реального аппаратного обеспечения, а производится только перераспределение ресурсов.

Относительные показатели теста Linpack 2.1 и тестов из пакета NPB 3.3.1 для различных вариантов представлены на рис. 3–5.

Особенности тестов из пакета NPB 3.3.1 с точки зрения их требований к вычислительным возможностям модулей следующие.

Ключевым моментом теста BT является эффективность с точки зрения общего потребления простых арифметических операций.

Ключевым моментом теста CG является оценка скорости передачи данных при отсутствии какой-либо регулярности.

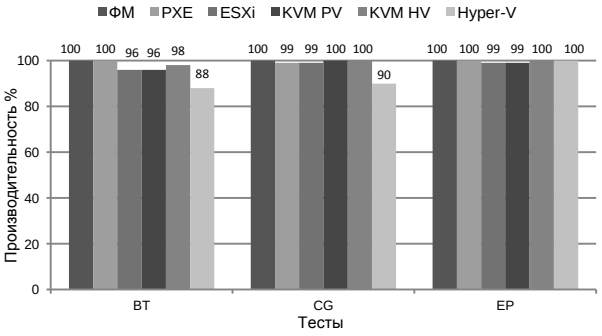

Рис. 3. Время выполнения тестов NPB BT, CG и EP для физических модулей (ФМ), для варианта сетевой загрузки (PXE) и для виртуальных машин под управлением гипервизоров ESXi, KVM и Hyper-V

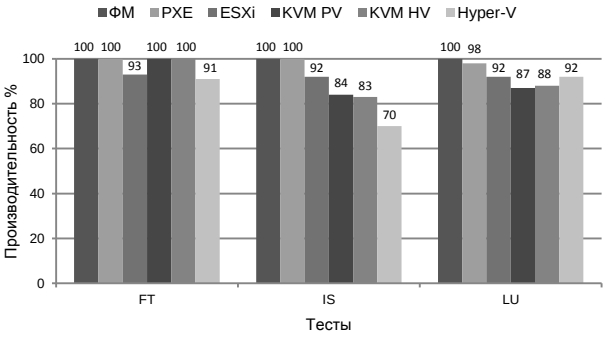

Рис. 4. Время выполнения тестов NPB FT, IS и LU для физических модулей (ФМ), для варианта сетевой загрузки (PXE) и для виртуальных машин под управлением гипервизоров ESXi, KVM и Hyper-V

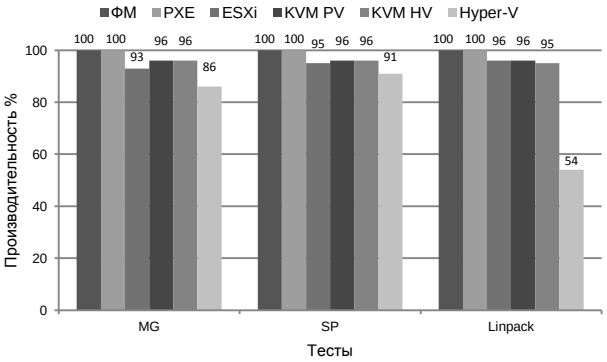

Рис. 5. Время выполнения тестов Linpack и NPB MG, SP для физических модулей (ФМ), для варианта сетевой загрузки (PXE) и для виртуальных машин под управлением гипервизоров ESXi, KVM и Hyper-V

Ключевыми моментами теста EP являются:

- оценка максимальной производительности кластера при операциях с плавающей точкой;
- минимальные межпроцессорные взаимодействия.

Ключевыми моментами теста FT являются:

- большое количество действий, оказывающих большую нагрузку на сеть;
- оценка скорости перемещения массивов данных.

Ключевыми моментами теста IS являются:

- сильное влияние начального распределения чисел в памяти;
- оценка работы с общей памятью.

Ключевым моментом теста LU является критичность ко времени передачи небольших объемов данных между модулями.

Ключевым моментом теста MG является оценка скорости передачи как длинных, так и коротких данных.

Ключевым моментом теста SP является обеспечение оптимальной загрузки сети.

В результате показатели тестов для физических модулей и модулей с сетевой загрузкой примерно одинаковые, то есть с точки зрения производительности параллельных программ использование жесткого диска преимуществ по отношению к сетевой загрузке не дает.

При использовании гипервизора считается, что он уже установлен на каждый из вычислительных модулей. Это позволяет быстро переключаться между имеющимися на нем программными платформами. Если виртуальные машины используются редко, то в процесс развертывания добавляется этап установки гипервизора на вычислительные модули, который для одного модуля занимает до 13 минут в зависимости от типа гипервизора.

Ярко выраженной зависимости между временем загрузки нескольких модулей и числом модулей ни для одного из вариантов не обнаружено, требуется дальнейшее исследование с большим числом модулей. Использование платформ виртуализации (при условии, что модули уже виртуализованы) дает некоторые преимущества перед развертыванием на физических модулях и сетевой загрузкой, так как сокращается время перезагрузки программных платформ, и предоставляются дополнительные возможности, обеспечиваемые средствами платформ виртуализации, например, изоляция виртуальных машин, точное распределение ресурсов, отказоустойчивость и др. Из недостатков использования гипервизоров – возможная долгая удаленная загрузка виртуальных машин (ESXi, KVM) на вычислительные модули, дополнительные требования к инфраструктуре (Hyper-V).

Использование виртуализации привносит накладные расходы при работе с вещественной арифметикой – по результатам тестов, проценты для ESXi и KVM, и существенные для Hyper-V. Виртуализация сети у гипервизоров и выбор сетевых драйверов для виртуальных машин существенно влияют на вычислительный процесс. При минимальном межпроцессорном взаимодействии производительность вычислений примерно одинакова для всех вариантов развертывания программных платформ. При наличии обменов небольшими порциями данных производительность заданий на виртуализованных вычислительных модулях падает, так как появляются дополнительные издержки. При оптимальной загрузке сети разница между физическими модулями и виртуализованными незначительная и составляет от 5 до 9 %.

В целом использование виртуализации для развертывания программных платформ привносит накладные расходы, основным источником которых является виртуализованная сетевая подсистема. Потери составляют от единиц процентов до десятков. Для получения максимальной производительности и минимальных сетевых задержек требуется углубленное знание и настройка конкретного гипервизора.

Минимальные накладные расходы при максимальной производительности у сетевой загрузки, но применение этого варианта невозможно для ОС Windows.

Дальнейшее развитие работы заключается в исследовании других гипервизоров и их различных версий, исследовании рассматриваемых вариантов с использованием сетевого взаимодействия на базе технологий с высокой пропускной способностью (10 Gigabit Ethernet) и минимальными задержками (InfiniBand).

СПИСОК ЛИТЕРАТУРЫ

1. **Clonezilla** Open Source Software for Disk Imaging and Cloning [Электронный ресурс]. URL: http://clonezilla.org (дата обращения 13.07.2014). **[** (2014, Jul. 13). *Clonezilla Open Source Software for Disk Imaging and Cloning* [Online]. Available: http://clonezilla.org **]**

2. **PXE** (Preboot eXecution Environment) [Электронный ресурс]. URL: http://ru.wikipedia.org/wiki/PXE (дата обращения 13.07.2014). **[** (2014, Jul. 13). *PXE (Preboot eXecution Environment)* [Online]. Available: http://wikipedia.org/wiki/PXE **]**

3. **Xen** [Электронный ресурс]. URL: http://ru.wikipedia.org/wiki/Xen (дата обращения 13.07.2014). **[** (2014, Jul. 13). *Xen* [Online]. Available: http://wikipedia.org/wiki/Xen **]**

4. **Kernel-based** Virtual Machine [Электронный ресурс]. URL: http://ru.wikipedia.org/wiki/Kernel-based_Virtual_Machine

(дата обращения 13.07.2014). **[** (2014, Jul. 13). *Kernel-based Virtual Machine* [Online]. Available: http://wikipedia.org/wiki/Kernel-based_Virtual_Machine **]**

5. **Виртуализация** серверов и облачная инфраструктура: VMware vSphere [Электронный ресурс]. URL: http://www.vmware.com/ru/products/vsphere (дата обращения 13.07.2014). **[** (2014, Jul. 13). *VMware vSphere: server virtualization, cloud infrastructure* [Online]. Available: http://www.vmware.com/products/vsphere **]**

6. **Linpack** Benchmark / В. П. Гергель. Технологии построения и использования кластерных систем [Электронный ресурс]: учебн. курс Национального открытого университета «ИНТУИТ» (дата обновления 18.10.2009). URL: http://www.intuit.ru/studies/courses/542/398/lecture/9173?page=2#sect5 (дата обращения 13.07.2014). **[** V. P. Gergel (2014, Jul. 13). *Linpack Benchmark* [Online], (in Russian). Available: http://www.intuit.ru/studies/courses/542/398/lecture/9173?page=2#sect5 **]**

7. **NAS** Parallel Benchmarks / В. П. Гергель. Технологии построения и использования кластерных систем [Электронный ресурс]: учебный курс Национального открытого университета «ИНТУИТ» (дата обновления 18.10.2009). URL: http://www.intuit.ru/studies/courses/542/398/lecture/9173?page=3#sect9 (дата обращения 13.07.2014). **[** V.P. Gergel (2014, Jul. 13). *NAS Parallel Benchmarks* [Online], (in Russian). Available: http://www.intuit.ru/studies/courses/542/398/lecture/9173?page=3#sect9 **]**

8. **Суперкомпьютер** «МВС-100К» [Электронный ресурс]. URL: http://www.jscc.ru/hard/mvs100k.shtml (дата обращения 13.07.2014). **[** (2014, Jul. 13). *Supercomputer "MVS-100K"* [Online], (in Russian). Available: http://www.jscc.ru/hard/mvs100k.shtml **]**

9. **Anaconda/Kickstart** [Электронный ресурс]. URL: http://fedoraproject.org/wiki/Anaconda/Kickstart (дата обращения 13.07.2014). **[** (2014, Jul. 13). *Anaconda/Kickstart* [Online]. Available: http://fedoraproject.org/wiki/Anaconda/Kickstart **]**

10. **Hyper-V** Server 2012 R2 [Электронный ресурс]. URL: https://www.microsoft.com/ru-ru/softmicrosoft/hyperv2012r2.aspx (дата обращения 13.07.2014). **[** (2014, Jul. 13). *Hyper-V Server 2012 R2* [Online], (in Russian). Available: https://www.microsoft.com/ru-ru/softmicrosoft/hyperv2012r2.aspx **]**

ОБ АВТОРАХ

АЛАДЫШЕВ Олег Сергеевич, зав. отд. высокопроизвод. систем и комплексов. Дипл. математик (МГУ, 1992). Канд. техн. наук по парал. системам хранения данных (ИПИ РАН, 2010).

БАРАНОВ Антон Викторович, ст. науч. сотр. Дипл. инж. по экспл. техн. и прогр. средств выч. техн. (МГУПИ, 1994). Канд техн. наук по упр. суперкомп. системами (ИПМ РАН, 2000), доц.

ИОНИН Рэм Павлович, стажер. Дипл. инж. по экспл. техн. и прогр. средств выч. техн. (МГУПИ, 2014).

КИСЕЛЕВ Евгений Андреевич, науч. сотр. Дипл. инж. по экспл. техн. и прогр. средств выч. техн. (МГУПИ, 2009). Канд. техн. наук по размещ. паралл. программ на суперкомп. системах (МГУПИ, 2013).

ОРЛОВ Вячеслав Анатольевич, стажер, студ. МГУПИ.

METADATA

Title: Comparative analysis of variants of deployment of program platforms for high performance computing.

Authors: O. S. Aladyshev[1], A. V. Baranov, R. P. Ionin, E. A. Kiselev, V. A. Orlov

Affiliation:
Joint Supercomputer Center of Russian Academy of Science (JSCC RAS), Russia.

Email: [1] Aladyshev@jscc.ru.

Language: Russian.

Source: Vestnik UGATU (scientific journal of Ufa State Aviation Technical University), vol. 18, no. 3 (64), pp. 295-300, 2014. ISSN 2225-2789 (Online), ISSN 1992-6502 (Print).

Abstract: The following variants of deployment of program platforms on supercomputer resources are considered: from hard disk image, booting platform via network, booting platform as virtual machine under control of various hypervisors (ESXi, KVM, Hyper-V). Results of experiments on deployment of program platforms for each of the considered variants are presented in article. Time of platform deployment and time of Linpack and NAS Parallel Benchmarks execution on deployed platforms were measured during experiments for each variant.

Key words: HPC; booting via network; hypervisor; ESXi; KVM; Hyper-V; Linpack; NPB.

About authors:

ALADYSHEV, Oleg Sergeevich, Head of Department «High Performance Systems and Complexies» JSCC RAS. Dipl. Mathematic (Moscow State University, 1992). Cand. of Tech. Sci.

BARANOV, Anton Victorovich, Senior researcher of JSCC RAS. Dipl. Computer engineer (1994). Cand. of Tech. Sci. (2000).

IONIN, Rem Pavlovich, Intern of JSCC RAS. Dipl. Computer engineer (2014).

KISELEV, Evgeniy Andreevich, Researcher of JSCC RAS. Dipl. Computer engineer (2009). Cand. of Tech. Sci. (2013).

ORLOV, Vyacheslav Anatolievich, Intern of JSCC RAS.