

Заключение

Рассмотрены новые средства организации мультиагентного управления высокопроизводительными вычислениями. Экспериментальные результаты показывают высокую масштабируемость и эффективность расчетов с помощью научных сервисов, разработанных на основе предложенного подхода. Эти результаты обусловлены учетом специфики предметных областей решаемых задач и мониторингом ГРВС.

1. Гергель В.П., Линев А.В. Проблемы и перспективы достижения экзаслопного уровня производительности суперкомпьютерных систем // Вестник Нижегородского университета им. Н.И. Лобачевского. – 2012. – № 3-1. – С. 189 – 198.
2. Gil Y., Deelman E., Ellisman M. et al. Examining the challenges of scientific workflows. Computer. – 2007. – Vol. 40. – № 12. – P. 24 – 32.
3. Toporkov, V.V., Yemelyanov, D.M. Economic Model of Scheduling and Fair Resource Sharing in Distributed Computations // Programming and Computer Software. – Vol. 40. – № 1. – P. 35 – 42.
4. Каляев А.И., Каляев И.А., Коровин Я.С. Метод мультиагентного диспетчирования ресурсов в гетерогенной облачной среде при выполнении потока задач // Вестник компьютерных и информационных технологий. – 2015. – № 11. – С. 31 – 40.
5. Kravari K., Bassiliades N. A Survey of Agent Platforms. Journal of Artificial Societies and Social Simulation. – 2015. – Vol. 18(1). – № 11. URL: <http://jasss.soc.surrey.ac.uk/18/1/11.html> (дата обращения: 19.04.2016 г.)
6. Бычков И.В., Опарин Г.А., Феоктистов А.Г., Богданова В.Г., Пашинин А.А. Сервис-ориентированное управление распределенными вычислениями на основе мультиагентных технологий // Известия ЮФУ. Технические науки. – 2014. – № 12. – С. 17 – 27.
7. Bychkov I.V., Oparin G.A., Feoktistov A.G., Bogdanova V.G., Pashinin A.A. Service-oriented multiagent control of distributed computations // Automation and Remote Control. – 2015. – Vol. 76. – № 11. – P. 2000 – 2010.
8. Bychkov I., Oparin G., Novopashin A., Sidorov I. Agent-Based Approach to Monitoring and Control of Distributed Computing Environment // In the Proc. of 13th International Conference on Parallel Computing Technologies. Lecture Notes in Computer Science. – 2015. – Vol. 9251. – P. 253 – 257.
9. Поляк Б.Г., Щербаков П.С. Трудные задачи линейной теории управления. Некоторые подходы к решению // Автоматика и телемеханика. – 2005. – № 5. – С. 7-46.
10. Nemirovskii A.A. Several NP-hard problems arising in robust stability analysis // Mathematics of Control, Signals, and Systems. – 1994. – Vol. 6. – P. 99 – 105.
11. Баландин Д.В., Коган М.Н. Синтез законов управления на основе линейных матричных неравенств. – М.: Физматлит, 2007. – 280 с.

A.B. Васюков, A.C. Ермаков, К.А. Беклемышева

ИСПОЛЬЗОВАНИЕ КОНТЕЙНЕРОВ В ИНФРАСТРУКТУРЕ РАЗРАБОТКИ, ЗАПУСКА И АВТОМАТИЧЕСКОГО ТЕСТИРОВАНИЯ НАУЧНЫХ ПРИЛОЖЕНИЙ*

*Московский физико-технический институт, г. Долгопрудный,
a.vasyukov@phystech.edu*

Контейнеры для разработки и запуска научных приложений

На сегодняшний день перспективность запуска высокопроизводительных научных приложений в модели облачных вычислений можно считать общепризнанным фактом, существуют успешные примеры реализации таких инфраструктур. Стандартом де-факто на настоящий момент для построения облачной инфраструктуры для научных приложений можно считать использование KVM и OpenStack, что позволяет по запросу пользователя запускать виртуальные машины с преднастроенными программными пакетами, а также формировать вычислительные кластеры из таких машин [1]. Однако накладные расходы при запуске приложений в виртуальных машинах могут быть достаточно высоки, а формирование единого образа для различных сред виртуализации является достаточно сложной задачей [2]. В связи с этим представляет интерес рассмотреть возможность использования в рамках той же модели технологий контейнерной виртуализации [3].

Одной из традиционных сложностей при совместном использовании множества прикладных вычислительных пакетов в рамках единого вычислительного кластера являются их разнородные и зачастую конфликтующие требования к инструментарию сборки и программному окружению для работы. В результате установка практически любого большого программного пакета на кластер вызывает необходимость предварительно собрать из исходных текстов большое количество библиотек конкретных версий. К тому же эту задачу зачастую приходится решать заново каждому пользователю кластера, которому потребовалось конкретное приложение. Авторам приходилось сталкиваться с ситуацией, когда для установки приложения сначала требовалось собрать из исходных текстов новую версию компилятора с помощью имеющейся на кластере более старой.

В работе на примере открытых пакетов OpenFOAM, deal.II, Gmsh, ParaView, GCM-3D был апробирован подход использования контейнеров Docker для решения указанной проблемы. Выполнялась

* Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 15-29-07096 офи_м

сборка приложения из исходных текстов в наиболее удобной для этого среде (Fedora 21, Fedora 22, Ubuntu 15.10 – в зависимости от прикладного пакета), после чего формировался контейнер, содержащий собранный прикладной пакет и всё требуемое для его работы программное окружение, такое как библиотеки конкретных версий и необходимые переменные окружения. Готовые контейнеры успешно запускались на модельной кластерной инфраструктуре на базе CentOS 7 без необходимости выполнять какие-либо операции по сборке из исходных текстов. Без применения контейнеров сборка и запуск некоторых из рассмотренных приложений в указанной инфраструктуре оказались бы нетривиальной задачей, требующей хорошей подготовки системного программиста. Таким образом, использование контейнерной виртуализации позволило разделить среды сборки и исполнения прикладных приложений и обеспечить для пользователя возможность быстрой подготовки и запуска собственного расчётного модуля, не требуя при этом от него ни квалификации системного программиста, ни прав администратора кластера.

Вопросы производительности при использовании контейнеров

В работе было выполнено тестирование производительности реальных прикладных вычислительных пакетов при работе в контейнерах. Основные тесты были выполнены на примере OpenFOAM и GCM-3D. В силу архитектуры контейнерной виртуализации накладных расходов оперативной памяти не наблюдается. Замеры времени решения модельных задач показали низкий уровень накладных расходов на работу системы контейнерной виртуализации как для однопоточных, так и для многопоточных версий – во всех замерах деградация производительности относительно запуска на физическом оборудовании не превышала 1 – 2 %. Данный результат показывает высокий потенциал облачных технологий на базе контейнерной виртуализации для запуска ресурсоёмких приложений.

Контейнеры в задаче автоматизации тестирования

Отдельно стоит отметить потенциал технологий контейнерной виртуализации при их тесной интеграции непосредственно в процессе разработки прикладных вычислительных пакетов, основанный на активном использовании облачных сервисов. Данная работа была выполнена на примере открытого расчётного модуля GCM-3D. Исходные тексты GCM-3D хранятся в репозитории облачного сервиса Github. Инфраструктура репозитория была настроена таким образом, что при внесении изменения в исходные тексты, выполняется автоматическая сборка бинарных модулей и обновление контейнера, содержащего GCM-3D и

требуемое программное окружение. Сборка выполняется один раз, формируя единый образ для всех сред выполнения.

В том числе собранные контейнеры с GCM-3D используются для автоматического тестирования корректности прикладного кода после внесения в него изменений. Для тестирования используется облачный сервис Travis CI. Существенной особенностью тестирования научных приложений является ресурсоёмкость тестов – для проверки корректности требуется выполнить расчёты большого количества модельных постановок, для которых имеются аналитические решения или экспериментальные данные, после чего сравнить в той или иной метрике полученное численное решение с эталонным. Для разумного покрытия тестами функционала прикладного пакета требуется провести тесты для множества солверов с различными входными параметрами, в силу чего прогон тестов вполне может занимать несколько часов времени. Использование облачных сервисов, таких как Travis CI, позволяет значительно облегчить тестирование при совместной работе над кодом команды разработчиков. Использование же контейнеров позволяет обеспечить простой запуск каждой новой сборки в инфраструктуре стороннего сервиса, в которой иначе было бы затруднительно выполнить хотя бы компиляцию расчётного модуля.

Заключение

Таким образом, в работе показана перспективность применения технологий контейнерной виртуализации при разработке и запуске научных приложений в модели облачных вычислений. К перспективным направлениям дальнейшей работы следует отнести интеграцию рассмотренных технологий с системами управления заданиями кластера, такими как SLURM, а также с планировщиками для грид-систем.

1. Running Virtual Machines in a Slurm Batch System, Ulf Markwardt, Technische Universität Dresden, Center for Information Services and High Performance Computing (ZIH), 2015 - <http://slurm.schedmd.com/SLUG15/SlurmVM.pdf>
2. Performance Comparison of Virtual Machines and Linux Containers, IBM Research, 2014 – [http://domino.research.ibm.com/library/cyberdig.nsf/papers/0929052195DD819C85257D2300681E7B/\\$File/rc25482.pdf](http://domino.research.ibm.com/library/cyberdig.nsf/papers/0929052195DD819C85257D2300681E7B/$File/rc25482.pdf)
3. Containerization of High Performance Compute Workloads using Docker, QNIB Solutions, 2014 – http://doc.qnib.org/2014-11-05_Whitepaper_Docker-MPI-workload.pdf