УДК 004.051:004.725

РАЗРАБОТКА ВИРТУАЛЬНОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ С ПОДДЕРЖКОЙ БАЛАНСИРОВКИ НАГРУЗКИ ДЛЯ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ РАСЧЕТА

[1]Богданов А.В., [1]Тхуреин Киав Лвин, [2]Чжо За, [2]Пья Сон Ко Ко

[1]Санкт-Петербургский государственный университет, Петергоф, e-mail: bogdanov@csa.ru, trkl.mm@mail.ru;

[2]Санкт-Петербургский государственный морской технический университет, Санкт-Петербург, e-mail: kyawzaya4436@gmail.com, pyaesonekoko@gmail.com

Предложен метод разработки и построения виртуальной вычислительной среды, обеспечивающей эффективную вычислительную мощность за счет динамического распределения нагрузки, управления ресурсами, миграции процессов на узлах вычислительной среды с использованием технологии виртуализации и единого образа операционной системы. Во-первых, мы проиллюстрировали методы интеграции многоуровневых приложений для эффективного управления ресурсами компьютерной системы. А также представлена возможность получения значительного ускорения при выполнении параллельных и многозадачных приложений с использованием миграции процессов для повышения эффективности вычисления. В данной статье анализируется оптимизация метакомпьютинга с помощью единого образа операционной системы на основе четырехузлового кластера с различной аппаратной процессора и программным обеспечением, обеспечивающим единый образ операционной системы для кластера. Во-вторых, мы проанализировали MPI программирование с MOSIX и без него, чтобы оценить возможности параллельных вычислений. Эти испытания проводились под контролем операционной среды MOSIX, с использованием и без использования схемы миграции процессов. Результаты этих тестов наглядно демонстрируют преимущества использования миграции процессов. После всех этих исследований мы запустили научные приложения в нашей разработанной вычислительной среде для проверки эффективности нашей системы.

Ключевые слова: виртуализация, процесс миграции, единый образ операционной системы (SSI), Mosix

DEVELOPMENT OF A VIRTUAL COMPUTING SYSTEM WITH SUPPORT OF LOAD BALANCING TO INCREASE CALCULATION EFFICIENCY

[1]Bogdanov A.V., [1]Tkhurein Kyaw Lwin, [2]Kyaw Zaya, [2]Pyae Sone Ko Ko

[1]Saint-Petersburg State University, Peterhof, e-mail: bogdanov@csa.ru; trkl.mm@mail.ru;

[2]Saint-Petersburg state marine technical university, Saint Petersburg, e-mail: kyawzaya4436@gmail.com; pyaesonekoko@gmail.com

A method for developing and building a virtual computing environment that provides efficient computing power due to dynamic load distribution, resource management, migration of processes on the nodes of the computing environment using virtualization technology and a single operating system image is proposed. First, we have illustrated methods for integrating multi-level applications for effective management of computer system resources. It also presents the possibility of getting significant acceleration when running parallel and multitasking applications using process migration to improve the efficiency of computing. This article analyzes how to optimize meta-computing using a single operating system image based on a four-node cluster with different hardware and software that provides a single operating system image for the cluster. Second, we analyzed MPI programming with and without MOSIX to evaluate the capabilities of parallel computing. These tests were performed under the control of the MOSIX operating environment, with and without the use of a process migration scheme. The results of these tests clearly demonstrate the advantages of using process migration. After all this research, we launched scientific applications in our developed computing environment to test the effectiveness of our system.

Keywords: virtualization, process migration, single system image (SSI), Mosix

Современные компьютерные технологии позволяют создать дешевые многомашинные комплексы с общими вычислительными ресурсами. Такие системы обеспечивают низкую стоимость для вычислений, хорошо масштабируемы, имеют высокий уровень надежности, имеют апробированные инструментальные средства для конструирования, отладки и анализа параллельных программ. В данной статье предлагается новый инструментарий для этих целей, включающий методики построения эффективной вычислительной среды и интеграцию стека программ для повышения общей производительности вычислений. Для построения эффективных вычислительных сред необходимы эффективные системы балансировки нагрузки, системы управления ресурсами и эффективные системы обработки данных. Балансировка нагрузки позволяет распределить избыточную локальную нагрузку по всем узлам вычислительной системы и оптимизировать использование имеющихся ресурсов, тем самым минимизируя расход ресурсов. А также помогает реализовать отказоустой-

чивость, обеспечивая масштабируемость, избегая узких мест и чрезмерной подготовки и сокращая время отклика [1, 2].

Целью данной работы является увеличение вычислительной мощности при объединении гетерогенных вычислительных ресурсов в единый вычислительный комплекс для эффективного управления вычислительными ресурсами и запуск приложений в распределенной виртуальной среде для анализа производительности. Для достижения поставленной цели необходимо решение следующих задач:

– разработка нового подхода к построению операционного окружения, обеспечивающего увеличение вычислительной мощности за счет динамической балансировки нагрузки и миграции процессов на узлах виртуальной вычислительной среды;

– интеграция комплекса программ и инструментария для оптимизации виртуальной гетерогенной системы и запуск специализированных задач для анализа производительности вычислительных сред.

Виртуализация

Благодаря поддержке технологий виртуализации физический сервер можно разделить на несколько изолированных сред выполнения путем развертывания уровня (т.е. виртуальной машины с диспетчером (VMM) или гипервизором) поверх аппаратных ресурсов или операционной системы (ОС). Среды выполнения на сервере, то есть виртуальные машины (ВМ), работают без взаимного прерывания друг друга [3].

Каждая виртуальная машина имеет свою собственную ОС и приложения. Это дает нам возможность консолидировать и объединять ресурсы таким образом, они могут быть распространены на различные приложения, чтобы компенсировать ограничения, связанные с сокращением ресурсов и растущими потребностями бизнеса. Виртуальные машины позволяют пользователям создавать, копировать, сохранять (контрольная точка), читать и изменять, совместно использовать, переносить и откатывать состояние выполнения машин со всей легкостью манипуляции с файлом. Эта гибкость обеспечивает значительную ценность для пользователей и администраторов. Следовательно, виртуальные машины быстро внедряются во многие вычислительные среды [4].

Монитор распределенной виртуальной машины

Поскольку монитор распределенной виртуальной машины обеспечивает сильную изоляцию между виртуальными машинами и физическими машинами, то администратор может передать полный контроль виртуализированного оборудования пользователям виртуальных машин, без раскрытия критических ресурсов. Эта функциональность значительно усиливает достижение безопасного и удобного виртуального хостинга. Кроме того, механизм резервного копирования виртуальных машин позволяет уменьшить последствия системных сбоев и взломов. С использованием распределенного монитора виртуальных машин (DVMM) на основе аппаратной поддержки технология виртуализации обеспечивает поддержку нативной операционной системы и работает прозрачно на распределенных узлах кластера [5, 6].

Единый образ операционной системы (SSI)

SSI означает, что все распределенные ресурсы объединены для пользователей в единый комплекс с универсальным интерфейсом, пользователям не нужно управлять каждым узлом вычислительной среды. SSI включает в себя некоторые атрибуты, такие как единое пространство памяти, одна система ввода-вывода, одна файловая система, система управления задачами и так далее. Ключевые атрибуты SSI реализованы в едином пространстве памяти и едином пространстве для вычислений. Кластеры SSI могут быть реализованы на аппаратном, промежуточном и прикладном уровнях. В принципе, архитектуры для многопроцессорных систем можно разделить на две группы: с когерентным распределенной памяти, таких как SMP и ccNUMA системы, и без таковых, типа сетей рабочих станций, соединенных Ethernet. Общие системы памяти обеспечивают простые модели программирования, совместимые с большой базой существующих приложений и операционных систем. Большинство существующих операционных систем поддерживают такие архитектуры; это относительно простая задача, даже если и не с оптимальной производительностью. Кроме того, они, естественно, реализуют единый образ операционной системы (SSI), где есть один экземпляр операционной системы с одним пространством ресурсов. Единый ресурс означает, что вычисления могут прозрачно мигрировать между процессорами, чтобы сбалансировать нагрузку [7].

Система MOSIX

MOSIX – это мультикомпьютерная операционная система с децентрализованным управлением. Система Mosix основана на Unix и предоставляет единый образ, как будто используется один компьютер с несколькими процессорами. Её основные характеристики сосредоточены на простоте

использования, создавая впечатление работы на одном компьютере с несколькими процессорами. Уникальные возможности системы Mosix включают автоматическое обнаружение ресурсов, динамическое распределение рабочей нагрузки с помощью миграции процессов и приоритетный метод, позволяющий процессам мигрировать между узлами в мультикластерах. Преимущество доступных ресурсов за пределами выделенных узлов в любом частном кластере [8]. Mosix создает виртуальный компьютер, который обеспечивает автоматическую балансировку нагрузки путем переноса процессов с сильно загруженного узла на менее используемый. Благодаря децентрализованному контролю каждый узел самостоятельно принимает свои собственные управляющие решения. Каждый узел способен работать как независимая система. Поскольку это не требует централизованного управления, узлы могут присоединяться или покидать ферму с минимальными нарушениями [9].

Файловая система MOSIX

Ключевым компонентом любой распределенной системы является файловая система. MOSIX использует свою собственную файловую систему MFS, чтобы сделать все каталоги и обычные файлы в кластере MOSIX доступными со всех узлов, как если бы они находились в пределах одной файловой системы. Одним из преимуществ MFS является то, что он обеспечивает согласованность кэша для файлов, просматриваемых с разных узлов, поддерживая один кэш на узле диска сервера.

MFS соответствует стандартам прямого доступа к файловой системе (DFSA), что расширяет возможности перенесенного процесса для выполнения некоторых операций ввода-вывода локально, в текущем узле. Это положение уменьшает потребность связанных с вводом-выводом процессов взаимодействовать с их домашним узлом, что позволяет таким процессам более свободно мигрировать между узлами кластера для балансировки нагрузки и параллельных операций ввода-вывода файлов. Это также позволяет осуществлять параллельный доступ к файлам путем правильного распределения файлов, где каждый процесс мигрирует на узел, имеющий свои файлы [9, 10].

Интеграция виртуализации и единого образа операционной системы (SSI)

Система SSI реализована в целях упрощения управления и программирования кластеров. Единый образ операционной системы – это вычислительная парадигма, в которой несколько распределенных вычислительных ресурсов объединяются и представляются через интерфейс, поддерживающий иллюзию взаимодействия с одной системой. Система SSI может быть реализована на нескольких уровнях абстракции, от пользовательского оборудования и распределенных гипервизоров до специализированных ядер операционной системы и инструментов пользовательского уровня [2, 5, 9]. На рис. 2 проиллюстрированный метод проектирования разработанной вычислительной системы с использованием виртуализации и SSI.

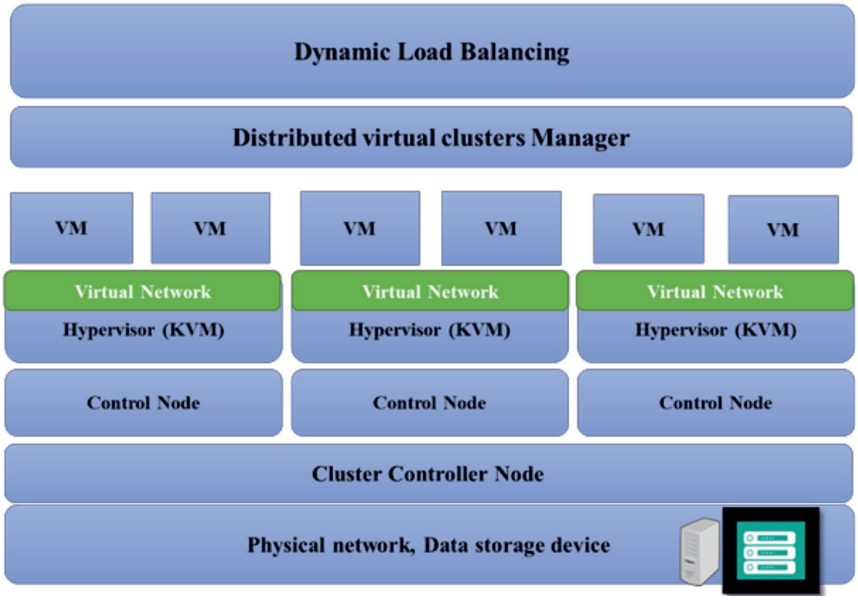

Рис. 1. Разработанная виртуальная вычислительная система

Рис. 2. Комбинация виртуализации и единого образа операционной системы (SSI) для разработки виртуальной вычислительной среды

Балансировка нагрузки

Балансировка нагрузки – это распределение рабочей нагрузки между несколькими компьютерами, вычислительными узлами кластера, центральными процессорами, дисковыми накопителями или другими ресурсами для достижения оптимального использования ресурсов, максимизации пропускной способности, минимизации времени отклика и предотвращения перегрузки системы. Использование нескольких компонентов с балансировкой нагрузки вместо одного компонента может повысить надежность за счет резервирования. Центральный процессор передает информацию с сильно загруженных серверов на недоиспользуемые серверы.

Для эффективной системы балансировки нагрузки в разработанном комплексе был создан виртуальный SMP-кластер с единым образом операционной системы, который скрывает гетерогенный и распределенный характер ресурсов и представляет их для пользователей и приложений как единый вычислительный ресурс. Чтобы повышать производительность, авторы разработали файловую систему MOSIX (MFS). С её помощью перенесенный процесс может более эффективно выполнять некоторые операции ввода-вывода на текущем узле. Таким образом, это позволяет процессам более свободно мигрировать между узлами кластера для удобной планировки нагрузки [10, 11]. Процесс миграции выполняется с использованием программного обеспечения и инструментов MOSIX:

– Запуск демон Mosix с помощью инструмента mosd.

– Получение информации о загрузке, отображаемую в виде графика, используя команду mon.

– Запуск процесса с помощью команды,

- # mosrun. /server

– Информация о процессе отображается с помощью инструмента MOSIX,

- # mosps -AMn

– Выполнение миграции процесса с помощью команды migrate,

- # migrate

Экспериментальная установка

Наша экспериментальная среда состоит из четырех узлов кластера, включая узел управления. Узел управления – это физическая машина с восьмиядерным процессором Intel i7-3770 с частотой 3,40 ГГц, 8 ГБ памяти и 1 ТБ диска. Остальные тамошние машины оснащены четырехъядерным процессором Intel Core i7-3770 с частотой 3,4 ГГц, 8 ГБ оперативной памяти и 1 ТБ дисков соответственно. 64-разрядная версия Ubuntu 19.10 (Eoan Ermine) работает как ОС на четырех машинах, а также во всех виртуальных машинах. KVM 2.5.0 используются для управления виртуальными машинами. Каждая из четырех физических машин оснащена сетевой картой Realtek RTL8111, которая подключается через сетевой коммутатор TP-LINK TL-SG1024DT.

Анализ производительности

В нашем эксперименте кластер MOSIX был создан в виртуальной среде. Там были установлены программы MPI и MOSIX, а также запущено приложение для определения времени задержки при синхронизации процессов MPI [12]. В этом примере выполняется тест связи MPI (время задержки сообщения). MPI-0 отправляет 1-байтовое сообщение в MPI-1, тратя время на ожидание ответа между ними. После этого выполня-

ется синхронизация для каждого повторения, а в конце вычисляется среднее время ожидания. Эти испытания проводились под контролем операционной среды MOSIX, с использованием и без использования схемы упреждающей миграции процессов. Сетевая задержка – это время, которое требуется для того, чтобы что-то, отправленное с исходного хоста, достигло конечного хоста. Время задержки в пути туда и обратно – это то, сколько времени требуется для запроса, отправленного из источника в пункт назначения, и для ответа, чтобы вернуться к исходному источнику. В принципе, задержка в каждом направлении плюс время обработки [13, 14]. На рис. 3 приведены результаты тестирования времени задержки в пути туда и обратно.

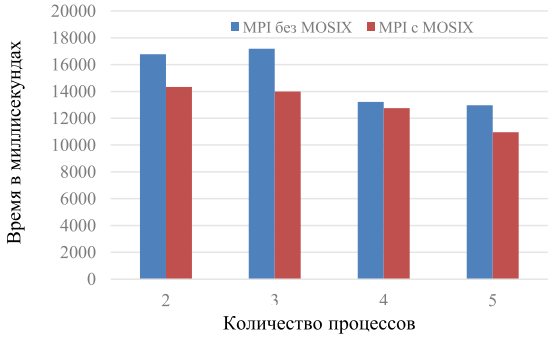

Рис. 3. Тестирование времени задержки в пути туда и обратно с и без MOSIX

После описанной процедуры конфигурации была проанализирована производительность нашей виртуальной распределенной среды. В системе с разнородными ресурсами создаются виртуальные вычислительные кластеры под контролем гипервизора и вычислительные ресурсы объединяются в единую вычислительную систему. Наша система работает с реальными задачами. Были проанализированы тестовые примеры, такие как тест LU из пакета NAS, научные ресурсоемкие приложения, такие как OpenFOAM (IcoFoam) и PCrystal. Результаты тестирования для анализа производительности разработанной среды показаны в таблице и на рис. 4. Эти результаты показывают, что очень сложные задачи с сильным взаимодействием между параллельными процессами могут быть решены достаточно эффективно с использованием разработанной вычислительной среды.

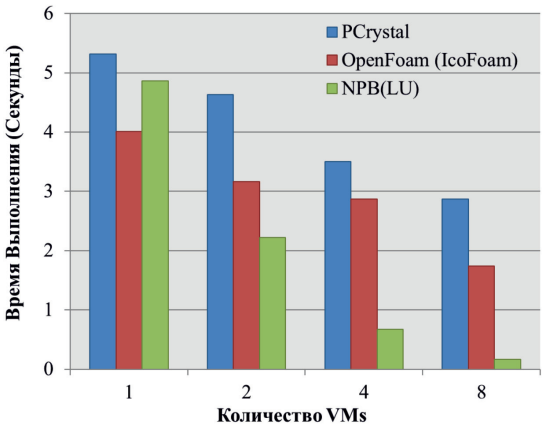

Рис. 4. Результаты тестов производительности ресурсоемких приложений

Заключение

Подход, представленный в данной работе, показал, что наша система позволяет запускать ресурсоемкие научные приложения в системе среднего уровня без каких-либо серьезных проблем и методы организации оптимальной виртуальной вычислительной среды. Технологии виртуализации очень популярны для запуска приложений и служб в различных ситуациях. В данной работе мы используем преимущества интеграции технологий SSI и виртуализации для построения эффективных виртуальных распределенных вычислительных сред с надежной миграцией процессов, управлением ресурсами и динамической балансировкой нагрузки. А также в данной работе были одобрены преимущества схемы миграции процессов для улучшения использования вычислительных ресурсов и возможности получения значительного ускорения при запуске параллельных и многозадачных приложений в разработанной вычислительной среде.

Результаты тестов времени выполнения

	Время выполнения (секунды)		
Количество VMs	PCrystal	OpenFOAM(IcoFoam)	NPB(LU)
1	5,32	4,01	4,86
2	4,63	3,16	2,22
4	3,51	2,87	0,67
8	2,87	1,74	0,17

Список литературы

1. Kiran Singh, Manoj Kumar Yadav. Comparative Study of Virtualization Management with Virtual Migration in Cloud Computing. IJSTE – International Journal of Scienc Technology & Engineering. 2017. Vol. 4. Issue 4. P. 37–44.

2. Богданов А.В., Чжо За, Пуае Сон Ко Ко. Усовершенствование вычислительных возможностей в вычислительной среде с помощью технологий виртуализации // Компьютерные исследования и моделирование. 2015. Т. 7. № 3. С. 499–504.

3. Bogdanov A., Kyaw Zaya, Pyae Sone Ko Ko. Approach to effectiveness and efficiency Cloud Computing environment. CSIT-2013 International Conference. Armenia, 2013. P. 319–322.

4. Yu L., Chen L., Cai Z., Shen H., Liang Y., Pan Y. Stochastic Load Balancing for Virtual Resource Management in Datacenters. IEEE Transactions on Cloud Computing. 2016. Vol. PP. Is. 99. P. 1–14.

5. Чжо За, Пья Сон Ко Ко. Разработка эффективности гетерогенной среды с комбинацией единого образа операционной системы и виртуализации // Естественные и технические науки. 2014. № 7. P. 93–95.

6. Nagarajan A.B., Mueller F., Engelmann Ch., Scott S.L. Proactive fault tolerance for HPC with Xen virtualization. ICS. 2007. P. 23–32.

7. Rajkumar Buyya T.C. Single system image (SSI). The International Journal of High-Performance Computing Applications. 2001. Vol. 15. P. 124–135. DOI: 10.1145/1274971.1274978.

8. Rahul Rajkumar Pahlajani, Dr. G.R. Bamnote. Mosix the Operating System that Support Multiple Cluster Environment with its Advancements & Features. International Journal of Computer Science and Mobile Computing. 2014. Vol. 3. Issue 4. P. 590–596.

9. Ibrahim F. Haddad. MOSIX: A Cluster Load-Balancing Solution for Linux. 2001. [Electronic resource]. URL: https://www.linuxjournal.com/article/4546 (date of access: 17.05.2020).

10. Shakti Mishra, D.S. Kushwaha, A.K. Misra. Hybrid reliable load balancing with MOSIX as middleware and its formal verification using process algebra. Future Generation Computer Systems. 2011. № 27. P. 506–526.

11. Divya Chaudhary, Rajender Singh Chhillar. A New Load Balancing Technique for Virtual Machine Cloud Computing Environment. International Journal of Computer Applications. 2013. Vol. 69. № 23. P. 37–40.

12. Mann V., Gupta A., Dutta P., Vishnoi A., Bhattacharya P., Poddar R., et al. Remedy: Network-aware steady state VM management for data centers. In Proceedings of the international IFIP TC 6 conference on networking, Berlin, Germany: Springer-Verlag, 2012. P. 190–204.

13. Ouyang X., Rajachandrasekar R., Besseron X., Panda D.K. RDMA-based job migration framework for MPI over Infiniband. Proceeding 2010 IEEE Int'l Conference on Cluster Computing (CLUSTER-2010). 2010. P. 116–125.

14. Shea R., Wang F., Wang H., Liu J. A deep investigation into network performance in virtual machine based cloud environments. INFOCOM 2014. Proceedings IEEE. 2014. P. 1285–1293.