УДК 62-82:658.512.011.56

АЛЕНИН О.И.

СИСТЕМА УПРАВЛЕНИЯ ВИРТУАЛЬНЫМИ КЛАСТЕРАМИ

Рассмотрен способ повышения эффективности использования кластерных систем с помощью виртуализации. Сделан краткий обзор типов виртуализации, выбран наиболее подходящий. Предложена структура системы управления виртуальными кластерами в рамках физического кластера.

The way to increase cluster system efficiency using virtualization is considered. The short review of virtualization types is made, and the most suitable is chosen. The structure of management system for virtual clusters over a physical cluster is proposed.

В настоящее время в связи с расширением круга научно – исследовательских задач требуются значительные вычислительные мощности для расчетов, как правило, большие, чем вычислительные мощности соответствующих организаций. При этом вычислительные ресурсы других организаций, доступные удаленно, не удовлетворяют требованиям задач из-за отсутствия на них необходимого специального программного обеспечения [1]. Часто это программное обеспечение может работать только на определенной платформе.

В настоящее время для решения этой задачи используются различные системы виртуализации. В большинстве случаев они позволяют виртуализировать только вычислительные узлы [2]. Вторая проблема заключается в том, что при этом основное внимание уделяется виртуализации вычислительных ресурсов, а виртуализация дискового пространства и сети выполняется неэффективно или не выполняется вообще. С переходом на Грид технологии возникает новая проблема, связанная с тем, что хотя грид может предоставить доступ к большому числу разнообразных ресурсов, часто эти ресурсы не соответствуют требованиям конкретных приложений или сервисов[3]. В вычислительной среде, где программное обеспечение развивается быстро, это несоответствие может привести к недоиспользованию ресурсов, недовольству пользователей и необходимости затрачивать большие усилия на преодоление несоответствия между ресурсами и приложениями. Эти проблемы могут быть решены путем создания кластеров виртуальных машин, с соответствующим набором программного обеспечения, необходимого для запуска приложений. Анализ подобного показывает эффективность его использования в грид – приложениях. При использовании виртуальных машин для запуска приложений потери производительности из-за использования виртуализации не превышают 5%. В работе [4] показано также, что учитывая время на создание и развертывание виртуальной среды в планировщике, а не оставляя этот процесс пользователю, можно достичь значительно большей эффективности использования ресурсов и более точного соответствия требованиям ко времени выполнения заданий. Учет в планировщике времени на передачу и развертывание образов виртуальных машин имеет два преимущества: во-первых, передачу образа можно запланировать и выполнить заранее, во-вторых, можно кешировать часто используемые образы виртуальных машин.

В статье [5] описан способ выделения ресурсов с помощью виртуальных машин, в котором предлагается модифицировать и расширить функции существующих планировщиков, таких как PBS и Sun Grid Engine, что также приводит к повышению эффективности использования ресурсов. Современные кластерные системы используются для разных типов задач. Некоторые задачи требуют большого объема ресурсов, но не накладывают строгих ограничений на время выполнения, в то время как другие задачи должны выполняться с максимальным приоритетом в заданный момент времени [6]. Использование виртуализации предоставляет возможность приостанавливать и возобновлять выполнение задач, переносить задачи вместе с виртуальными машинами на другие физические узлы, а также предоставлять вычислительные ресурсы (виртуальные машины) с предустановленным программным обеспечением, необходимым для выполнения данной задачи.

Есть несколько факторов, оказывающих существенное влияние на эффективность применения виртуализации для систем с кластерной архитектурой:

- используемый тип виртуализации;
- выбор между виртуализацией отдельных ресурсов и построением виртуальных кластеров;
- эффективность работы системы управления ресурсами;
- класс задач, решаемых на кластере.

Для выполнения параллельных прикладных программ, изначально рассчитанных на работу в системах с кластерной архитектурой, целесообразно виртуализировать не отдельные ресурсы кластера, а создать на основе физических ресурсов виртуальный кластер, наиболее точно соответствующий требованиям задач. Для запуска на кластере множества не параллельных программ виртуализация отдельных ресурсов может оказаться предпочтительнее из-за отсутствия накладных расходов на систему управления виртуальными кластерами. Однако, большинство программ, выполняемых на кластерных системах, являются параллельными, и здесь будет рассматриваться только этот вариант.

Типы виртуализации

Существуют такие типы виртуализации: *эмуляция аппаратуры, полная виртуализация, паравиртуализация, виртуализация уровня операционной системы* и *виртуализация уровня приложений*.

При использовании *эмуляции аппаратуры* полностью эмулируется архитектура целевой машины, фактически происходит интерпретация команд гостевого процессора на хост-процессоре. Это самый медленный тип виртуализации, программы выполняются в сотни раз медленнее, чем на физической машине. Примеры систем – Bochs, QEMU.

Полная виртуализация – самый популярный способ виртуализации, предполагает использование программного обеспечения, получившего название «гипервизор», суть которого заключается в создании уровня абстракции между виртуальными серверами и базовым аппаратным обеспечением. Примерами коммерческих решений, в которых реализован данный подход, могут служить программные продукты VMware и Microsoft Virtual PC, а KVM (Kernel Virtual Machine) – это свободно распространяемое решение для ОС Linux. Гипервизор перехватывает команды центрального процессора и служит посредником для доступа к аппаратным контроллерам и периферии. В результате полная виртуализация позволяет установить на виртуальный сервер практически любую операционную систему без каких-либо изменений, причем сама ОС ничего не будет знать о том, что она работает в виртуализованной среде. Основной недостаток данного подхода связан с накладными расходами, которые несет процессор в связи с работой гипервизора. Эти накладные расходы невелики, но ощутимы [7]. В полностью виртуализованной среде гипервизор взаимодействует непосредственно с аппаратным обеспечением и серверами в качестве хостовой операционной системы. Операционные системы, работающие на виртуальных серверах, которыми управляет гипервизор, называют гостевыми.

Полная виртуализация предполагает серьезное использование ресурсов процессора, обусловленное наличием гипервизора, управляющего различными виртуальными серверами и обеспечивающего независимость этих серверов друг от друга. Уменьшить эту нагрузку можно, например, модифицировав каждую операционную систему таким образом, чтобы она «знала» о том, что она работает в виртуализованной среде, и могла взаимодействовать с гипервизором. Такой подход называют *паравиртуализацией*. Примером свободно распространяемой реализации технологии паравиртуализации может служить Xen. Прежде чем операционная система сможет работать в качестве виртуального сервера в гипервизоре Xen, в нее необходимо внести определенные изменения на уровне ядра.

Существует еще один способ виртуализации – встроенная поддержка виртуальных серверов на уровне операционной системы. Этот подход называется *виртуализацией на уровне операционной системы* и использован, например, в Solaris Containers. Существует также Virtuozzo/OpenVZ для ОС Linux. При виртуализации на уровне операционной системы не существует отдельного слоя гипервизора. Вместо этого сама хостовая операционная система отвечает за разделение аппаратных ресурсов между несколькими виртуальными серверами и поддержку их независимости друг от друга. Отличие этого подхода от

других проявляется, прежде всего, в том, что в этом случае все виртуальные серверы должны работать в одной и той же операционной системе (хотя каждый экземпляр имеет свои собственные приложения и регистрационные записи пользователей). Виртуализация на уровне операционной системы теряет в гибкости, но производительность близка к производительности физического сервера. Кроме того, системой, которая использует одну стандартную ОС для всех виртуальных серверов, намного проще управлять, чем более гетерогенной средой.

Виртуализация прикладных приложений включает в себя рабочую среду для локально выполняемого приложения, использующего локальные ресурсы. Виртуализируемое приложение запускается в небольшом виртуальном окружении, которое включает в себя ключи реестра, файлы и другие компоненты, необходимые для запуска и работы приложения. Такая виртуальная среда работает как прослойка между приложением и операционной системой, что позволяет избежать конфликтов между приложениями.

Каждый тип виртуализации обладает своими достоинствами и недостатками. При применении виртуализации для повышения эффективности использования кластерных систем к выбранному типу виртуализации предъявляются такие основные требования:

- невысокие накладные расходы на виртуализацию;
- возможность запуска в гостевой среде операционных систем с предустановленным программным обеспечением.

Следовательно, нецелесообразно использовать эмуляцию аппаратуры и невозможно использовать виртуализацию уровня приложений. Использование паравиртуализации целесообразно только в тех случаях, когда модификация гостевой ОС не представляет трудностей. Наиболее приемлемым является использование виртуализации уровня ОС, если гостевая ОС близка к базовой (например, различные версии Linux), и полной виртуализации для запуска других гостевых ОС (например, запуск Windows на Linux).

Система управления виртуальными кластерами

Как уже было сказано, одним из факторов, существенно влияющих на эффективность использования виртуализации в кластерных системах, является система эффективная работа системы управления виртуальными кластерами. Она управляет виртуальными узлами, сетями, дисковыми ресурсами, осуществляет создание и удаление виртуальных кластеров, мониторинг состояния узлов и т.д.

Для возможности одновременной работы нескольких виртуальных кластеров с разным набором прикладного ПО на одном физическом кластере, необходимо, чтобы с точки зрения гостевых ОС виртуальные кластера были изолированными друг от друга, использовали разные дисковые и сетевые ресурсы. Такая схема показана на рис. 1

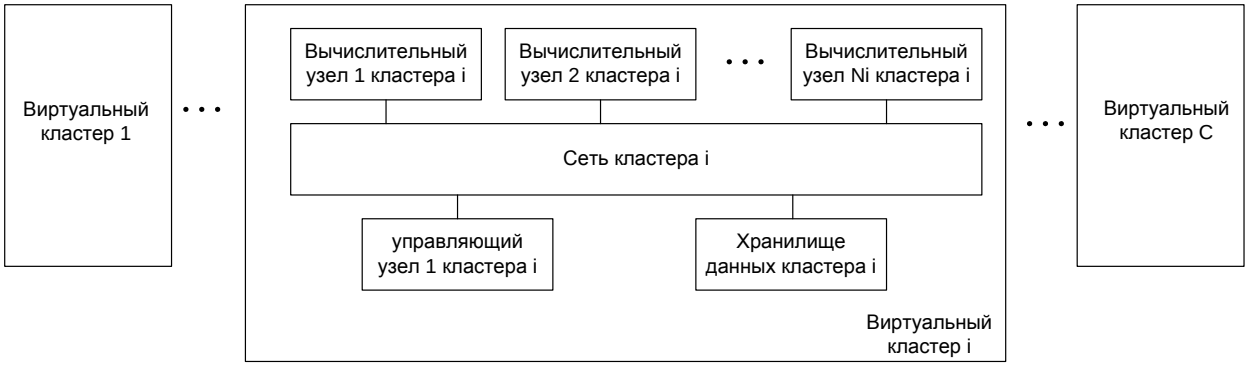

Рис.1 Виртуальные кластера

В соответствии с такой схемой, система управления виртуальными кластерами должна содержать блоки, обеспечивающие управление вычислительными, сетевыми и дисковыми ресурсами. На рис. 2 показана схема системы управления виртуальными кластерами и взаимодействие ее с другими системами, обеспечивающими работу физического кла-

стера, такими как система управления ресурсами физического кластера, интерфейс грузки заданий, система хранения.

Систему можно разделить на несколько блоков. Блок управления вычислительными ресурсами отвечает за создание, удаление и запуск виртуальных узлов на физических узлах. Блок управления системой хранения отвечает за распределение доступного дискового пространства, создание и удаление виртуальных дисковых ресурсов. Блок управления сетевыми службами осуществляет создание, удаление и настройку виртуальных фейсов, распределение адресов, управляет сетевыми сервисами, необходимыми для загрузки виртуальных узлов. Блок обработки очереди заданий связан непосредственно с интерфейсом загрузки заданий (или с мой управления заданиями физического кластера, если физические ресурсы виртуализируются не полностью, и есть возможность запуска заданий в базовой ОС физических узлов). База данных с информацией о ресурсах содержит все записи, необходимые для ты остальных блоков.

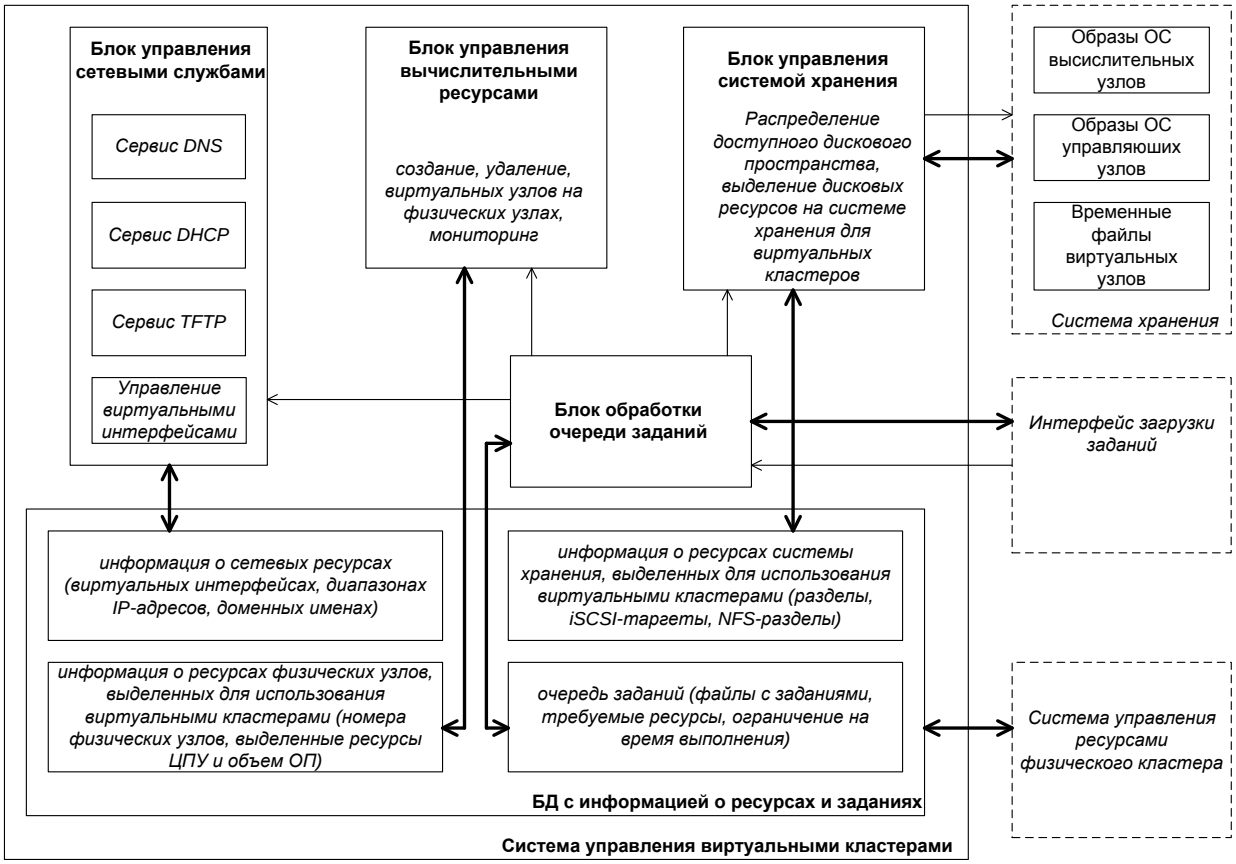

Рис.2 Система управления виртуальными кластерами

Создание/удаление виртуальных кластеров происходит таким образом (рис 3).

Через интерфейс пользователя в блок обработки очереди заданий поступает задание, состоящее из образа виртуального управляющего и вычислительного узлов с предустановленным прикладным программным обеспечением, которое необходимо для решения задачи пользователя, и дополнительных параметров. В дополнительных параметрах задания могут быть указаны такие как необходимый размер оперативной памяти, производительность процессора, количество узлов, требуемое дисковое пространство для хранения временных файлов, требования к сети обмена данными между узлами и т.д. Образы ОС виртуальных узлов передаются на систему хранения и сохраняются там. В базе данных сохраняются ссылки на месторасположение образов ОС узлов в системе хранения с привязкой к данному заданию. При наличии свободных ресурсов в соответствии с дисциплиной обслуживания выбирается задание из очереди. После этого выполняется выделение ресурсов, и загрузка виртуальных узлов на виртуальных машинах в физических узлах. Образ ОС поступает с системы хранения. После загрузки выполняется автоматическая настройка за-

груженных виртуальных узлов, с помощью протокола DHCP задаются настройки сети, выполняется монтирование каталогов или виртуальных дисков для записи временных файлов.

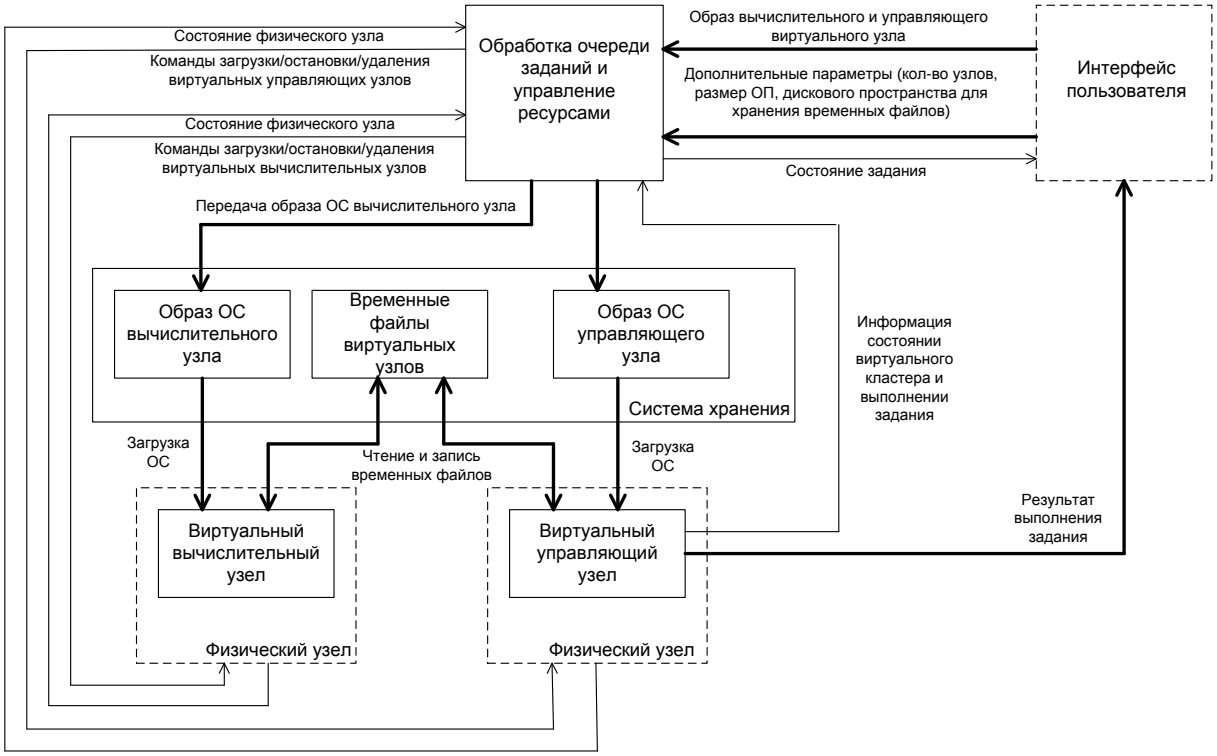

Рис.3 Схема обмена данными и управления при создании и удалении виртуальных кластеров

В процессе работы осуществляется мониторинг состояния физических узлов. При выходе из строя физического узла виртуальные узлы, работавшие на нем, переносятся на работоспособные физические узлы. Это осуществимо потому, что образы ОС и промежуточные результаты работы прикладных программ сохраняются не на локальных дисках физических узлов, а в системе хранения.

После завершения выполнения прикладных программ с помощью интерфейса пользователя подается команда удаления виртуального кластера. Это может выполняться как вручную, так и автоматически при передаче результатов выполнения прикладной программы. При удалении виртуального кластера выполняется размонтирование подключенных дисковых ресурсов, удаление виртуальных сетевых интерфейсов, остановка соответствующих виртуальных машин на физических узлах, и в базе данных о ресурсах занятые ресурсы помечаются как свободные.

Список литературы

1. Keahey, K., T. Freeman, J. Lauret, D. Olson. Virtual Workspaces for Scientific Applications, SciDAC 2007 Conference, Boston, MA. June 2007.
2. NAKADA, H., YOKOI, T., EBARA, T., TANIMURA, Y., OGAWA, H., AND SEKIGUCHI, S. The design and implementation of a virtual cluster management system. In Proceedings of the first IEEE/IFIP International Workshop on End-to-end Virtualization and Grid Management, 2007.
3. Foster, I., T. Freeman, K. Keahey, D. Scheftner, B. Sotomayor, X. Zhang. Virtual Clusters for Grid Communities, CCGRID 2006, Singapore. May 2006.
4. Overhead Matters: A Model for Virtual Resource Management, Sotomayor, B., K. Keahey, I. Foster. VTDC 2006, Tampa, FL. November 2006.
5. Enabling Cost-Effective Resource Leases with Virtual Machines, Sotomayor, B., K. Keahey, I. Foster, T. Freeman. HPDC 2007 Hot Topics session, Monterey Bay, CA. June 2007
6. Combining Batch Execution and Leasing Using Virtual Machines, Sotomayor, B., K. Keahey, I. Foster. HPDC 2008, Boston. June 2008.

7. Jones T. An overview of virtualization methods, architectures, and implementations, IBM, 2006, http://www.ibm.com/developerworks/linux/library/l-linuxvirt.