

## **ВИРТУАЛЬНЫЕ ИНСТРУМЕНТЫ ДЛЯ ГРИД-ВЫЧИСЛЕНИЙ**

А.А. Московский<sup>1</sup>, А.Ю. Первин<sup>1</sup>, В. Walker<sup>2</sup>

Институт программных систем РАН, Переславль-Залесский,  
Россия; <sup>2</sup>Hewlett-Packard Laboratories, Пало-Альто, США

*Разработано программное обеспечение для развертывания и управления приложениями работающих внутри виртуальных машин. Созданы и протестированы приложения: вычислительный сервис и веб-сервис. Составлены профили этих приложений и изучены зависимости между производительностью приложений и ресурсами. Представлены промежуточные результаты активного исследования, направленного на изучение вопросов управления, аппаратными ресурсами с использованием математической теории.*

### **Введение**

За последние несколько лет технология виртуальных машин проделала большой путь: от вспомогательного средства для разработчиков и тестировщиков программного обеспечения (ПО) до полноценного средства для организации вычислительной платформы. Современные виртуальные машины обладают целым рядом уникальных свойств. В частности, например, пользователь виртуальной машины может предоставлять ей ровно столько аппаратных ресурсов, сколько необходимо для решения поставленной задачи. Такие возможности позволяют эффективно консолидировать имеющиеся вычислительные ресурсы и тем самым значительно повысить КПД и сократить затраты на их эксплуатацию.

Другим важным достоинством виртуальных машин является возможность создания унифицированных пользовательских окружений. В виду большого разнообразия операционных систем и дистрибутивов (в среде Linux), различных версий программных библиотек и пользовательских приложений значительно усложняется процесс создания нового ПО, поскольку разработчики вынуждены переносить и адаптировать свое программное решение на большое количество различных платформ (аппаратных и программных). Однако в последнее время все большую популярность получает подход, при котором ПО распространяется в виде так называемых виртуальных инструментов (virtual appliance). Такие инструменты представляют собой пару «виртуальная машина» и «пользовательское приложение». При этом приложение, как правило, уже настроено и готово к работе. Для того чтобы воспользоваться им, пользователю достаточно всего лишь запустить виртуальный инструмент.

В контексте грид это обстоятельство принимает решающее значение, поскольку именно грид-среды, служащие вычислительной площадкой одновременно для многих приложений, работают на самых различных аппаратных и программных ресурсах. Использование виртуальных инструментов для грид-вычислений позволяет существенно упростить процесс управления конфигурацией узлов грида и распределения задач. Кроме того, многие исследования виртуальных машин

показывают, что накладные расходы на запуск приложений в виртуальной среде оказываются незначительными. Однако для того, чтобы такие решения были востребованы среди пользователей грида, необходимо создать средства для прозрачной интеграции виртуального инструмента с грид-средой.

В рамках этой работы разрабатывается решение для обеспечения комфортной работы пользователей с виртуальными инструментами в гриде. При управлении виртуальными инструментами применяется концепция соглашений об уровне сервиса (Service Level Agreements - SLA). В качестве грид-среды используется Unicore [1].

## 1. Среда «виртуальные сервисы»

Для решения поставленных задач была реализована простая система, которая позволяет развертывать *виртуальные сервисы* - параллельные виртуальные инструменты. Виртуальные инструменты представляют собой узкоспециализированные приложения, распространяемые в виде образа файловой системы для виртуальной машины, настроенные и готовые к работе. Для управления виртуальными машинами используется Xen [2] -монитор виртуальных машин с открытым исходным кодом, однако принципы, заложенные в системе, могут быть перенесены на любую другую аналогичную по функциональности платформу.

Нормальная эксплуатация системы подразумевает наличие *поставщика сервиса*, который запускает сервисы, и *пользователей*, подключающихся к сервису через так называемую *точку входа* - пары IP-адреса и сетевого порта. Среда предоставляет, в частности, такие функции:

- запуск и остановка сервиса. Запуск сервиса подразумевает запуск одной или нескольких виртуальных машин с соответствующим виртуальным диском (образом файловой системы), установку правил маршрутизации сетевого трафика и создание виртуальной сети из виртуальных машин, предназначенных для этого сервиса;
- выделение и высвобождение ресурсов сервиса. Запросы на ресурсы могут быть инициированы вручную *поставщиком сервиса* или программно при помощи специальной компоненты - *планировщиком сервиса*;
- перенаправление сетевого трафика. Эта функция необходима для обеспечения доступа пользователей к приложениям в виртуальной среде, а также используется для балансировки нагрузки.

Основными отличительными особенностями созданной системы являются:

- использование кластерных установок в качестве аппаратной платформы;
- возможность формирования «виртуального кластера» из виртуальных машин;
- автоматическое управление количеством ресурсов, доступных приложению.

В последнем случае технологии виртуальных машин открывают

новые возможности по управлению качеством обслуживания в системе, на которой запущено несколько приложений, чьи потребности в аппаратных средствах меняются со временем. Отличительной особенностью нашей работы является использование возможности изменять ресурсы, доступные приложениям внутри виртуальной машины мелкими порциями (квантами), что позволяет предполагать функцию уровня сервиса того или иного приложения непрерывной функцией вычислительных ресурсов. В системе реализован механизм, позволяющий оптимизировать количество ресурсов, находящихся в распоряжении того или иного приложения в зависимости от текущей загрузки и потребностей в тех же ресурсах других приложений. Сама функция «уровень сервиса» определяется образом, специфичным для каждого конкретного приложения.

Один из сценариев использования системы описывает веб-сайт, испытывающий резкое увеличение пользовательской активности. Как следствие, возникает отказ в обслуживании пользователей из-за слишком большого числа обращений. В этом случае владелец веб-сайта стремится удерживать время обработки запросов (время отклика) пользователей в некоторых разумных пределах (например, меньше 1 секунды). Такое состояние приложения называется целевым. Для поддержания требуемого качества обслуживания (уровня сервиса) виртуальному инструменту необходимо наращивать вычислительные ресурсы по мере увеличения числа пользователей, работающих с приложением. Среда *виртуальные сервисы* способна автоматически увеличивать количество аппаратных ресурсов, доступных приложению, при увеличении пользовательской нагрузки и, наоборот, уменьшать количество доступного процессорного времени, объема памяти и других ресурсов при снижении нагрузки.

Концепция соглашений об уровне сервиса применима не только для веб-сайтов, а также может быть сравнительно легко обобщена для более широкого класса приложений. Для этого достаточно определить три параметра виртуального инструмента: целевое состояние, описывающее приемлемый уровень сервиса, текущее состояние приложения и некую зависимость, описывающую влияние управляемых переменных (ресурсов) на состояние сервиса.

## 2. Разработанные инструменты

В настоящий момент система поддерживает три виртуальных инструмента. Для каждого инструмента подготовлены соответствующий образ ВМ и средства поддержки (конфигурационные файлы, планировщики и т. д.).

**WebMapServer** [3] позволяет запрашивать различную информацию по географическим картам. В тестах использовались данные по округу Итаска штата Миннесота, полученные через Геологическую службу США. Отображаемые пользователем страницы содержат сгенерированные по запросу фрагменты карты в формате GIF.

**X-Com.** Вычислительный сервис основан на программном обеспечении **X-Com**. X-Com [4] - эта система метакомпьютинга, разработанная в МГУ им. М.В. Ломоносова. X-Com чем-то напоминает систему распределенных вычислений **Condor** [5], однако реализация X-Com значительно более компактна, менее требовательна к ресурсам, проще в установке и эксплуатации. Кроме того, система X-Com может

работать в самых различных окружениях: вычислительные кластеры, федерации кластеров, грид-среды, совокупности гетерогенных процессоров, очереди задач и т. д.

**Виртуальный кластер** позволяет запускать требуемое число ВМ с поддержкой сетевого подключения между ними. В результате запуска этого сервиса создается набор виртуальных узлов, формирующих *виртуальный кластер*. Сетевая поддержка реализуется с помощью механизма bridging. Он позволяет включать ВМ в виртуальный кластер абсолютно прозрачным для пользователя образом. В результате пользователи могут взаимодействовать с узлами виртуального кластера без какой-либо дополнительной настройки, так, как если бы это был обычный компьютер.

По результатам вычислительных экспериментов в виртуальном кластере было показано, что в таком окружении можно работать с полноценными MPI-приложениями, использующими несколько виртуальных узлов кластера одновременно. Кроме того, проведены успешные эксперименты по запуску параллельных программ, созданных с ; помощью средства быстрой разработки параллельных приложений OpenTS [6].

## Заключение

В ходе исследования была разработана платформа для управления приложениями, работающими в виртуальных машинах Xen. С помощью компонент системы можно запускать и останавливать сервисы, динамически увеличивать или уменьшать объем доступных сервисам ресурсов. Представлена простая модель автоматизации управления уровнем сервиса приложений, использующая методы условной оптимизации. На базе платформы был протестирован ряд приложений с целью составления профилей производительности для последующей разработки более точных и эффективных моделей управления ресурсами виртуальных инструментов. Используемый подход не привязан исключительно к *виртуальным сервисам* и может быть использован в других средах, таких, как Virtual Workspaces или Cluster-on-Demand.

В настоящий момент ведется исследование платформы Unicore и проектирование соответствующих расширений системы *виртуальные сервисы* для последующего использования разработанного инструментария на суперкомпьютерах семейства СКИФ.

## Список литературы

- 1 . Unicore, Distributed computing and data resources [Electronic resource]. - 2008. - Mode of access: <http://www.unicore.eu>.
2. Xen hypervisor [Electronic resource]. — 2005. - Mode of access: <http://www.xen.org>.
3. MapServer [Electronic resource]. — 2008. - Mode of access: <http://mapserver.gis.umn.edu>.
4. Воеводин, В. Суперкомпьютер на выходные / В. Воеводин, М. Филамофитский // Открытые системы. — 2003. - Т. 5. — С. 43-48.
5. Thain, D. Distributed computing in practice: the Condor experience / D. Thain, M. Livny // Concurrency and Computation: Practice and Experience. — 2004. - Т. 17, № 2-4. — С. 323-356.
6. Т-Система с открытой архитектурой / С. Абрамов [и др.]

//Суперкомпьютерные системы и приложения. - Минск: ОИПИ НАН Беларуси, 2004. — С. 18-22.