РЕАЛИЗАЦИЯ НОВЫХ ТЕХНОЛОГИЙ РЕШЕНИЯ ЗАДАЧ ВЫЧИСЛИТЕЛЬНОЙ ХИМИИ В ГРИД-СРЕДАХ

В.М. Волохов, Д.А. Варламов, А.В. Волохов, А.В. Пивушков

Для создающихся в России грид-полигонов [1,2] и интегрированных в их состав вычислительных ресурсов настоятельно встают (в первую очередь перед администраторами ресурсных грид-сайтов) проблемы повышения эффективности проведения вычислений в распределенных средах. Сюда могут быть включены увеличение скорости и масштабируемости расчетов, разработка способов решения «объемных» задач, увеличение степени совместимости грид-сред с прикладными программными пакетами (ППП), а также понижение расходов на создание/эксплуатацию грид-ресурсов и снижение трудоемкости администрирования.

Темой статьи является реализация и внедрение в состав ресурсного грид-центра ИПХФ РАН некоторых новых технологий для проведения распределенных вычислений в области квантовой химии. Рассмотрены следующие технологии:

- создание и поддержка виртуальных машин (ВМ) для размещения управляющих и сетевых сервисов ресурсных сайтов грид-полигонов;
- «виртуализация» грид-приложений (особенно сложных ППП) и использование динамически формируемых «виртуальных контейнеров» в роли грид-заданий;
- создание и управление «пулами» грид-заданий для работы с большими задачами на равномерных «сетках» данных или параметров. Задачи при этом могут быть представлены в виде объединения большого количества параллельно выполняемых независимых друг от друга заданий с последующим объединением результатов;
- адаптация для грид-сред GPU-ориентированных версий ППП квантовой химии и молекулярной динамики для повышения масштабирования и эффективности параллельных расчетов.

Виртуализация грид-ресурсов и грид-сервисов на основе виртуальных машин

В 2010-2012 годах в ИПХФ РАН было проведено изучение технологий виртуализации ресурсов для повышения функциональности ресурсных центров грид-полигонов, снижения трудоемкости и затрат на их создание и поддержку [3,4]. Созданный комплекс виртуальных машин переведен в режим постоянной эксплуатации в качестве поставщиков грид-сервисов как для нужд самого узла, так и для нужд поддерживаемых грид-полигонов.

Опыт применения ВМ показал, что виртуальные машины в качестве серверных компонентов (ресурсы, сервисы) должны быть использованы тогда, когда на физическом узле желательно выполнение сервисов, не отличающихся высокими степенью загрузки ресурса и сетевым трафиком, но требующих выполнения следующих условий: (а) наличия различных программных платформ для выполнения специфического системного или прикладного ПО; (б) присутствия несовместимых между собой конфигураций ПО; (в) изоляции сервисов друг от друга (использование одних и тех же портов и сокетов, конфликт сетевых ресурсов и т.п.).

Анализ параметров ресурсных сайтов показал, что отсутствуют значимые технические и программные препятствия для введения виртуальных ресурсов в состав управляющих систем ресурсных сайтов грид-полигонов с последующим переносом или установкой заново на них необходимого набора грид-сервисов. При этом введение виртуальных ресурсов в состав грид-сайтов может быть произведено «прозрачно» для обычных грид-пользователей и практически незаметно для функционирования ресурсного сайта и грид-полигона в целом. Ввод виртуальных ресурсов может производиться как на базе работающих серверных хост-систем (без их останова — менее эффективно с точки зрения производительности) путем размещения на них гипервизоров, так и при замене/вводе в эксплуатацию новых серверных компонент ресурсных центров. На примере ресурсных сайтов ИПХФ авторами показано, что практически все грид-сервисы ресурсных сайтов могут быть размещены на виртуальных ресурсах без ущерба для производительности.

Для эффективной работы грид-сервисов на базе виртуальных ресурсов желательно для их размещения выделение в составе ресурсного сайта физических машин, отвечающих повышенным аппаратным требованиям – 2-4 процессора с поддержкой аппаратной виртуализации на базе Intel VT (Virtualization Technology) либо AMD-V/SVM (Secure Virtual Machine), с достаточным объемом оперативной памяти (4-8 Гб на процессор), с быстрой (SAS/SCSI) дисковой подсистемой с поддержкой RAID. Для сайтов класса «production farm» желательно наличие доступного гипервизорам локального хранилища установочных образов ВМ и «снимков» уже настроенных ВМ с возможностью быстрого (в том числе «на лету») восстановления ВМ и соответствующих сервисов при крахе или останове одной из физических машин. Количество физических машин для размещения пула ВМ и их параметры пропорционально зависят от размера ресурсного сайта и потенциального объема услуг, оказываемых в рамках грид-полигона.

Для заметного повышения производительности и эффективности использования аппаратных ресурсов настоятельно рекомендуется устанавливать гипервизоры ВМ и пул ВМ на «чистые» машины, из оптимизированных дистрибутивов с последующей установкой из образов и настройкой под конкретные грид-сервисы необходимого количества ВМ, а также внешних средств управления пулами ВМ (типа Proxmox (http://www.proxmox.com). При использовании аппаратной виртуализации в этом случае обеспечивается производительность, сравнимая с производительностью невиртуализованной машины.

Установка, настройка и отладка грид-сервисов на виртуальных ресурсах производится аналогично таковым на физических машинах, возможен перенос конфигурационных настроек с работавших физических машин на виртуальные ресурсы с последующим перезапуском грид-сервисов. После успешной отладки грид-сервисов рекомендуется сделать «снимки» (snapshot) ВМ и поместить их на хранение для быстрого аварийного восстановления в случае краха ВМ или физического узла.

В качестве средства виртуализации ресурсов и сервисов нами был выбран гипервизор KVM (Kernel-based Virtual Machine, http://www.linux-kvm.org.). Выбор основан на простоте администрирования (особенно при использовании высокоуровневых front-end типа Proxmox), устойчивости работы под нагрузкой, независимостью от стороннего коммерческого разработчика, интеграцией в Linux ядро (т.е. простотой использования), устойчивостью в работе, достаточно высокой эффективностью использования машинных ресурсов (низкая степень накладных «расходов»).

Для тестирования гипервизора и созданных ВМ на 2-х управляющих машинах ресурсного грид-центра ИПХФ были установлены следующие сервисные ВМ:

- для среды Globus Toolkit 4 (http://www.globus.org, полигон ГридННС): ОС CentOS 5.4, сервисы MDS, GRAM, GridFTP, RFT, User Interface;
- для среды Unicore 6.2 (http://www.unicore.eu, СКИФ-Полигон) – ОС Ubuntu 9.10, сервисы: шлюз (Gateway); серверный контейнер (Unicore/X), интерфейс к целевой системе (TSI), авторизационный сервис и пользовательская база данных – XUUDB$, пользовательский интерфейс (UI);
- для Computing Element среды gLite (http://glite.web.cern.ch, российский сегмент EGI-[RU-NGI]) - ОС ScientificLinux 4.5, сервисы lcg-ce, авторизации и мониторинга (деактивированы в настоящее время).

Выбор ОС для управляющих узлов был обусловлен либо требованиями дистрибутивов распределенного ПО (gLite), либо рекомендациями разработчиков, либо преимуществами администрирования. На управляющие машины были установлены серверные компоненты пакета управления заданиями PBS/Torque (http://www.clusterresources.com), тогда как на расчетные узлы (ОС ScientificLinux 5.4) была установлена клиентская часть данного пакета. Поскольку все распределенные middleware требуют наличия своих собственных очередей PBS (Portable Batch System), были настроены три одновременно работающих экземпляра pbs_mom на расчетных узлах с соответствующим набором очередей заданий. В таком варианте каждая управляющая машина связывается с расчетными узлами по уникальному порту и имеет дело только со своими заданиями.

Недостатком данного подхода является невозможность (пока) правильного учета ресурсов, используемых расчетным узлом, однако, для экспериментальных и исследовательских работ, отладки прикладного и промежуточного ПО и проведения в меру ресурсоемких расчетов это вполне приемлемо, а впрямую касается только использования пула расчетных узлов. Явными преимуществами же являются:

- необходимость однократных установки и настройки ПО для решения входящих задач (особенно прикладных, поскольку часто установка прикладных пакетов оборачивается непредвиденными трудозатратами);
- простота администрирования управляющих и расчетных узлов;
- экономия ресурсов (включая прежде всего электроэнергию);
- повышенный коэффициент загрузки за счет лучшей утилизации CPU.
- повышенная надежность ресурсного сайта. В случае поломки физического узла копия виртуальной машины (размещенная на резервном файл-сервере) может быть запущена в считанные минуты (максимум в первые часы).

При наличии достаточного объема ресурсов все управляющие виртуальные машины могут быть размещены на одном физическом узле. В качестве эксперимента на машине с 8 Гб оперативной памяти были размещены управляющие сервисы всех вышеуказанных ресурсных узлов совместно с машинами, обслуживающими внутрисетевые сервисы ИПХФ, что нисколько не сказалось на качестве выполнения и доступности грид-сервисов. Попутно на управляющих машинах можно также разместить дополнительные виртуальные машины, отвечающие за нересурсоемкие сервисы сети (например, web- или ftp-сервер, клиентские интерфейсы распределенных сетей, сервер баз данных и т.п.), поскольку их влияние на основные сервисы незначительно, а при этом данные службы желательно изолировать от грид-узлов.

Наиболее важной проблемой использования виртуальных машин в качестве ресурсных становится правильный учет доступных для каждой грид среды ресурсов физического узла и мониторинг выполняемых

ресурсным узлом задач (на всех входящих в него ВМ плюс хост-система). Учет ресурсов, востребованных ВМ, пока проводится на уровне гипервизора и не оценивается адекватно мониторингом распределенной среды со стороны агентов мониторинга (собственных и внешних), что может вести к недоучету ресурсов и завышенным ожиданиям со стороны входящих задач.

Использование ВМ позволяет участвовать в работе нескольких грид-полигонов, значительно повышает надежность управляющих узлов ресурсных сайтов, снижает трудоемкость администрирования сайтов, понижает расходы на эксплуатацию и поддержку грид-инфраструктуры.

Виртуализация грид-приложений

Другим вариантом технологий виртуализации является «виртуализация» исполняемых грид-заданий на основе ППП, т.е. создание динамически формируемых «виртуальных контейнеров», позволяющих производить запуски ППП без предустановки и настройки на удаленных узлах распределенных сетей.

Данная технология была ранее успешно апробирована и детально описана авторами ранее [5] в грид-средах gLite и Unicore для бинарных приложений и квантово-химического ППП GAMESS , здесь приводится лишь краткое описание и реализация для ГридННС.

«Виртуальные контейнеры» содержат сам ППП, «персональные» копии необходимых системных файлов и библиотек (в том числе специализированных типа BLAS или FFT) и (при необходимости) параллельных сред исполнения (сокетные модели типа DDI или Charm++ или внешние пакеты типа Mpich-2), скрипты по изменению пользовательских настроек операционной системы и переменных окружения, необходимые для функционирования приложения его собственные базы данных и файловые «деревья», а также пользовательские конфигурационные файлы и файлы данных. Метод позволяет грид-пользователю возможность формировать «виртуальное» (создаваемое на время выполнения) приложение, которое в виде «виртуального контейнера» как обычное грид-задание доставляется на ресурсный сайт и при этом не требует процедуры предварительной установки и настройки. Далее «виртуальный контейнер» собирает информацию о ресурсе, самостоятельно разворачивается на всех доступных узлах грид-ресурса (или ограничивается необходимым количеством), подготавливая среду для исполнения приложения с последующим его запуском (включая параллельные варианты). После выполнения задания происходит возврат результатов штатными средствами грид-среды и «зачистка» расчетных узлов ресурса от файлов «контейнера». Эксперименты в этой области показали, что так могут быть решены многие проблемы установки, настройки, совместимости ППП на грид-узлах, разрешаются конфликты одновременного запуска одинаковых приложений..

В настоящее время этот метод применим для Linux-узлов с 64-битной архитектурой, т.е. типичных компонентов кластеров, интегрированных в грид-среды.

В 2012 году на узлах полигона ГридННС для среды Globus Toolkit был реализован вариант «контейнера» для ППП NAMD, размер «виртуального контейнера» в случае моделирования простых структур составляет около 6,5 мегабайт

Серия первичных запусков была проведена на ресурсном сайте ГридННС ИПХФ РАН (сайт использован как удаленный с запуском задач через инфраструктуру грид-полигона). Дальнейшее успешное тестирование было проведено на ресурсных узлах ГридННС в рамках ВО «NanoChem» (НИИЯФ МГУ и Курчатовский РНЦ), были рассчитаны примеры молекулярных структур белков типа аланина.

Технология формирования «виртуальных контейнеров» для NAMD и GAMESS-US (создан ранее) интегрирована в грид-портал ИПХФ в качестве высокоуровневых веб-интерфейсов, проведены успешные запуски «контейнеров» через портал на узлы ГридННС и аналогичных по функционалу грид-полигонах.

Работа с «пулами» независимых грид-заданий на равномерных «сетках» данных или параметров

Существует огромное количество задач, которые представляют собой совокупность независимых расчетных заданий, решение которых основано на переборе исходных данных или входных параметров, число их зависит от количества параметров или от густоты «сетки» разбиения искомой области данных. Хорошим примером служат многопараметрические задачи вычислительной химии, требующие последовательного перебора большого количества входных параметров. При этом полная задача разбивается на огромное количество (до 10^4-10^7) независимых подзадач (каждая определяется группой значений совокупности параметров). Как вариант – решение задач на больших областях исходных данных, когда результат в каждой решаемой точке не зависит от «соседей». Задача автоматизации процесса разбиения полной задачи на фрагменты («нарезка») во многом аналогична таковой для многих кластерных задач [6] и определяет детальность и полноту получаемого решения и удобство пользования системой. Расчет независимого задания занимает от нескольких минут до первых часов, однако, суммарное время расчета становится нереальным для одиночной вычислительной системы даже при высокой степени параллелизации выполнения задачи. Также стало актуальным использование «тяжелых» прикладных пакетов типа GAMESS для расчета больших массивов точек (например, многомерных энергетических поверхностей), для которых нужно производить вычисления сотен и тысяч «точек» по «сетке» с закономерно изменяющимися параметрами.

Авторами [4,6] описана технология запуска «пулов» («пучков») независимых заданий для использования всех доступных ресурсов распределенной среды с использованием грид технологий. Созданный

авторами программный комплекс обеспечивает разбиение первичной задачи по равномерным «сеткам» областей данных или параметров, формирование пула готовых к исполнению грид-заданий, их параллельного запуска в грид-среду, мониторинг и контроль выполнения участников пула, сборка результатов в конечный обобщающий результат.

Функции комплекса:

- разбиение первичной задачи на равномерных областях данных или входных параметров;
- автоматическое формирование пула независимых друг от друга грид-заданий;
- параллельный запуск независимых заданий из пула в грид-среду;
- мониторинг и контроль выполнения заданий в рамках пула, включая останов и перезапуск;
- сборка результатов расчетов заданий с грид-ресурсов в конечный обобщающий результат.

Выполнение всех работ с пулами задания происходит с использованием базы данных web-портала (MySQL или PostgreSQL) и внешних данных аккаунтинга и мониторинга, предоставляемых средствами грид-полигона. На языке Perl был написан комплекс скриптов для формирования «пулов» заданий, их запуска и получения результатов счета с использованием пользовательских интерфейсов (UI) грид-сред. Для решения многопараметрических задач квантовой химии были разработаны методы формирования «пулов» независимых заданий с варьирующими параметрами – до 10^4, в перспективе до 10^7 «атомарных» заданий на задачу. Авторскими скриптами производится «нарезка» областей данных или входных параметров, формирование пулов независимых грид-заданий, создание очередей запуска и отправки заданий на брокер ресурсов. После запуска периодически запускаемые скрипты UI ведут мониторинг выполнения заданий, контроль тайм-аутов, перезапуск неудачных заданий и сбор результатов (с использованием таблиц базы данных, контролирующих состояние заданий – «ожидание», «запуск», «выполнение»). По окончании расчетов проводится сборка «атомарных» результатов в единый выходной файл или архив. Для части задач (требующих значительного числа параллельных независимых расчетов) в настоящее время создаются механизмы по разбиению областей данных (или расчетов) на большие «подсетки» в форме независимых заданий, передачи всех их интерфейсам распределенных сред с последующим запуском на параллельных узлах и «сборки» финальных результатов из множества полученных независимых. Это позволяет достигнуть разумного баланса между собственно временем счета заданий и накладными расходами по передаче заданий по распределенной среде. Метод был протестирован на распределенных ресурсах ВО «Nanochem» полигона ГридННС, ВО RGSTEST, СКИФ-Полигона.

К отрицательным сторонам метода относится его высокая зависимость от производительности брокеров ресурсов распределенной среды и возможностей их работы в асинхронном режиме.

Вычислительные грид-сервисы на основе прикладных пакетов с поддержкой GPU вычислений.

Применение GPU вкупе со сменой парадигмы программирования для прикладных пакетов ведет к убыстрению расчетов от десятков процентов до первых порядков по времени. В настоящее время резко интенсифицировался перевод программного кода прикладных пакетов (в том числе – квантово-химических) на параллельные технологии с использованием технологий программирования CUDA^{tm} и аналогичных. Это позволяет проводить высокоинтенсивные вычисления с применением гибридных расчетных узлов кластеров. Во вновь создаваемых кластерах (использующихся как грид-ресурсы) большинство расчетных узлов оснащается графическими адаптерами, что ставит задачу по использованию CUDA-ориентированных прикладных пакетов на грид-ресурсах весьма актуальной.

Использование массивно-параллельных архитектур NVIDIA и Radeon дает превосходные результаты при работе с приложениями квантовой химии и молекулярной динамики. По оценкам Nvidia Group выигрыш во времени расчетов (для части методов) составляет (для ППП, реализованных в ИПХФ как грид-сервисы): GAMESS – 50-60 раз, Gaussian – 12-15 раз, VASP – 3-6 раз, NWChem – 3-8 раз, GROMACS – 2-5 раз, LAMMPS – в 6 раз, NAMD – 2-7 раз. Для прикладных пакетов, созданных именно для CUDA вычислений (TeraChem и PetaChem - http://www.petachem.com) выигрыш по времени достигает 50-5000 раз.

В ИПХФ РАН начаты исследования использования GPU-оптимизированных прикладных квантово-химических пакетов как на локальных кластерах ИПХФ (установлены гибридные узлы), так и в рамках грид-сайтов различных грид-сред. Проводится тестирование и оптимизация ППП с поддержкой CUDA вычислений с последующей адаптацией их к грид-средам. Начата разработка методов запуска грид-заданий, ориентированных на поиск и использование ресурсов, поддерживающих использование CUDA технологий.

Работа поддержана грантом РФФИ № 11-07-00686-а.

ЛИТЕРАТУРА:

1. А.П. Крюков, Л.В. Шамардин Мат-лы международной научной конференции «Параллельные вычислительные технологии (ПаВТ’2012)», Новосибирск, март 2012 г., Изд-во ЮУрГУ, 2012. с.553–558
2. С.В. Абламейко, В.В. Анищенко, А.М. Криштофик Труды Международной суперкомпьютерной конференции «Научный сервис в сети Интернет: экзафлопсное будущее», Новороссийск, сентябрь 2011 г., - М.: Изд-во МГУ, 2011. с.147-153

3. В.М. Волохов, А.В. Пивушков, А.В. Волохов, Д.А. Варламов Мат-лы X международной конференции «Высокопроизводительные параллельные вычисления на кластерных системах» HPC-2010, Пермь, т.1, с.119-124
4. А.В. Пивушков, В.М. Волохов, Д.А. Варламов, А.В. Волохов, Н.Ф. Сурков. Труды международной научной конференции “Параллельные вычислительные технологии ПАВТ-2012”, Новосибирск, март 2012 г., изд-во ЮУрГУ, с. 638–644
5. В.М. Волохов, Д.А. Варламов, А.В. Пивушков, Н.Ф. Сурков, А.В. Волохов Динамически формируемые параллельные среды в условиях грид-полигонов, проблемы и решения // «Вычислительные методы и программирование: Новые вычислительные технологии», М.: МГУ, 2011, т.12, № 1, с.39-45
6. В.М. Волохов, Д.А. Варламов, А.В. Пивушков, А.В. Волохов Мат-лы Международной суперкомпьютерной конференция «Научный сервис в сети Интернет: экзафлопсное будущее», 19-24 сентября 2011,Новороссийск – М.: Изд-во МГУ, 2011 с.382-384