

ПРОБЛЕМНО-ОРИЕНТИРОВАННЫЙ ИНТЕРФЕЙС ДЛЯ АДАПТАЦИИ ПАКЕТОВ ПРИКЛАДНЫХ ПРОГРАММ К GRID

А. Г. Тарасов, Я.С. Нестеренкова, С.И. Мальковский

ВЦ ДВО РАН

taleks@as.khb.ru, yana.nesterenkova@ccfebras.ru, sergei@as.khb.ru

Аннотация

Использование распределенных вычислительных систем сопряжено со значительными сложностями, возникающими при использовании современных вычислительных систем специалистами предметной области. Накопленный ранее набор научных программ зачастую неработоспособен на современных архитектурах и широко распространённых операционных системах. Приведено описание подхода, позволяющего преодолеть указанные затруднения.

С развитием распределенных вычислительных систем (РВС) стало возможным решение многих задач, требующих существенных вычислительных затрат. Развитие удаленного доступа и РВС типа Grid позволило специалистам различных научных и образовательных учреждений, не имеющих технических возможностей, инфраструктуры, соответствующих специалистов, пользоваться возможностями вычислительных центров других организаций в рамках единой вычислительной среды [1].

Рост числа суперкомпьютерных систем не был подкреплён достаточным уровнем подготовки специалистов, обладающих знаниями, необходимыми для работы на таких комплексах. Сложилась ситуация, когда достаточные для решения большого класса прикладных задач вычислительные ресурсы есть, а ИТ и специалисты в предметных областях не знают, как в полной мере осуществлять поддержку и использовать такие системы.

Это приводит к тому, что большое число прикладных программных систем и информационно-вычислительных технологий, способных существенным образом повлиять на качество и скорость проводимых научных и прикладных исследований, исключается или используется в минимальном объеме российскими специалистами, что естественно влияет на уровень проводимых работ.

Существуют несколько значимых проблем, затрудняющих проведение вычислений, в том числе:

1. Накоплена значительная библиотека научных программ, которые были написаны в то время, когда основные используемые операционные системы и прикладное ПО отличалось от современного. Такое ПО, для которого нередко уже не доступны исходные файлы, невозможно использовать в целях решения прикладных задач.
2. Сложность организации работы с вычислительными системами растёт с ростом их вычислительных возможностей, а специфика используемого программного обеспечения не позволяет специалистам в прикладных областях, далёких от вычислительных технологий, быстро решать возникающие в работе программного и аппаратного обеспечения проблемы.

В ряде случаев, решить первую проблему возможно с использованием систем

виртуализации либо программно-аппаратной эмуляции. Первое направление наших исследований основано на предоставлении каждому приложению в рамках РВС запрашиваемого при описании задания образа системы. Использование систем виртуализации и программно-аппаратной эмуляции позволяет запускать приложения внутри заранее подготовленных системных окружений, которые и содержат в себе требуемый образ системы, набор библиотек, конкретные версии программного обеспечения. Ранее было показано, что при использовании систем виртуализации существенного падения производительности выполняющихся вычислительных заданий не наблюдается [2].

В связи с этим, базовой системой виртуализации для создаваемого программного комплекса запуска унаследованного ПО на вычислительных кластерах была выбрана система Xen, с которой сотрудники ВЦ ДВО РАН имели достаточный опыт работы. В то же время, жесткая привязка к одной системе исключает гибкость программного комплекса и ограничивает сферу его возможностей. Поэтому было принято решение добиться в какой-то мере независимости от используемой системы программно-аппаратной эмуляции или виртуализации. В настоящее время система строится на основе программного комплекса libvirt, распространяемого в исходном коде под лицензией LGPL. Конечной целью применения libvirt является унификация работы с виртуальными машинами, выполняющихся под управлением различных гипервизоров и систем: KVM, Xen, VMware и др.

На базе имеющихся разработок в ВЦ ДВО РАН ведутся работы над системой управления заданиями, которая будет брать на себя функции создания конкретных виртуальных машин или легковесных контейнеров (зависит от типа задания), выполнения в них задач пользователя на удаленных вычислительных кластерах, получения результатов и другие управляющие функции, скрывая от конечного пользователя работу со средой Grid или локальным менеджером заданий вычислительного кластера.

Другим направлением исследований является адаптация ряда пакетов прикладного программного обеспечения (abinit, FHI98md и др.) для использования в РВС. Во взаимодействии со специалистами предметной области осуществляется создание проблемно-ориентированного интерфейса (ПОИ) работы с abinit (FHI98md, DP и др.).

Архитектура программной системы представлена в виде нескольких изолированных слоев абстракций. Введены такие логические сущности как задание, пользователь, управляемый файл и другие. В основу разрабатываемой программной платформы, обеспечивающей взаимодействие системных компонент РВС и прикладных пакетов программ положена архитектура, состоящая из двух компонентов: серверного и клиентской части, созданной на основе web-технологий. Связь между ними осуществляется с использованием способа организации межпроцессного взаимодействия на базе RESTful запросов, управляющие команды которых представлены JSON объектами. Этот подход дает возможность менять реализации программных компонентов независимо друг от друга.

Серверный компонент реализует необходимые адаптеры к инструментариям управления вычислительной средой. Работа платформы с вычислительной средой осуществляется с помощью утилит командной строки, Pilot-CLI, OpenSSL и т.п. Для поддержки PBS Torque разработан отдельный программный адаптер для запроса состояния вычислительных ресурсов Grid-среды, постановки заданий на выполнение и т.п. Таким образом, обеспечена поддержка двух наиболее широко используемых в научно-образовательной среде инструментариев построения РВС.

Для обеспечения работы с вычислительной средой созданы следующие модули серверного компонента:

- адаптеров для преобразования запросов пользователей в запросы конкретных поддерживаемых платформой вычислительных сред (Pilot-CLI, PBS Torque);
- авторизации web-компонента и пользователей системы;
- формирования конфигурационных и вспомогательных файлов для пакетов прикладных программ на основе системы правил и шаблонов, проверка корректности введения параметров;
- управления файлами, позволяющий получить информацию по всем имеющимся файлам вычислительного задания;
- управления заданиями, хранящий независимые от используемой вычислительной среды идентификаторы задания, с которым связаны различные файлы, позволяющий управлять вычислительным заданием (ставить на выполнение, удалять, получать информацию о состоянии);
- для тестирования разрабатываемой программной платформы.

Серверный компонент предоставляет управляющий интерфейс, с помощью которого с ним взаимодействует клиентская часть платформы. Необходимую функциональность работы с запросами (прием соединений, диспетчеризация) выполняет фреймворк Tornado или любой, поддерживающий WSGI сервер.

Web-компонент основан на WSGI, запросы к которому выполняет web-сервер Apache, который поддерживает SSL соединения. В состав клиентской части платформы входят следующие модули:

- авторизации пользователей;
- создания форм, представляющих набор связанных параметров, зависящих от конкретной предметной задачи;
- просмотра текстовых файлов, представляющих результаты расчетов.
- интерфейс для контроля состояния вычислительного задания;
- модуль, обеспечивающий базовые функции отображения модели данных на веб-интерфейс.

Разработка программных компонент платформы осуществляется на языке программирования Python с использованием СУБД MySQL, ORM SQLAlchemy и ряда стандартных библиотек.

Авторизация пользователей в системе в настоящее время осуществляется с использованием сертификатов x.509, что позволяет интегрировать легко разрабатываемое ПО в инфраструктуру Grid с уже налаженной выдачей сертификатов. Поддерживается запуск заданий от имени серверного компонента (используется единственный сертификат) и запуск с использованием делегированного сертификата. В обоих случаях задания пользователей учитываются независимо.

Введение управляемых файлов позволяет скрыть от пользователя детали реализации и физической структуры/иерархии файлов в хранилище, что в свою очередь повышает надежность системы, исключая умышленный и неумышленный доступ, основанный на предположении о нахождении файлов. Каждый файл имеет информацию о принадлежности к

конкретному вычислительному заданию, типу системы и подсистемы, являющейся владельцем/создателем данного файла.

Для формирования конфигурационных файлов вычислительных заданий используется несколько модулей организации шаблонов файлов, которые на основе входящих параметров формируют итоговый файл для запуска программы. В той или иной мере данный подход свойственен всем системам, связанных с проблемно-ориентированными интерфейсами. Преимуществом разрабатываемой системы является высокая гибкость и автоматизация создания и описания входных файлов задания.

Сами параметры описываются в файлах-описаниях форм ввода, там же указываются и условия, которым должны соответствовать параметры. Это позволяет упростить создание форм ПОИ.

Формируемые программной системой файлы упрощенно делятся на две категории: файлы, относящиеся к заданию (описание требуемых ресурсов в вычислительной среде), и файлы относящиеся к задаче (конфигурационные файлы конкретного пакета прикладных программ). Благодаря этому сравнительно легко обеспечивается перенос одной и той же вычислительной задачи между различными системами планирования заданий в различных видах вычислительных систем (Grid, вычислительный кластер). Такой подход также в явном виде и предлагаемой типизации не встречается в используемых в настоящее время системах.

Модуль управления жизненным циклом задания сопоставляет каждому вычислительному заданию внутренний уникальный идентификатор, который не зависит от конкретного инструментария, используемого для построения вычислительной среды. Схожий механизм используется в Грид-системах, где единый идентификатор скрывает за собой идентификаторы в диспетчерах очереди заданий на конкретном кластере. В разрабатываемой платформе достигается дополнительная абстракция, которая скрывает не только конкретный кластер, но и тип инструментария поддержки вычислительной системы. Пользователи, запускающие задания через разрабатываемую систему, например, на вычислительных кластерах посредством менеджера очередей, имеют прозрачный доступ к данным, полученным от заданий, выполнявшихся в других вычислительных средах.

Используемый подход позволяет создавать интерфейсы для запуска разнообразных программных пакетов как в различных Grid-средах, так и на отдельных вычислительных кластерах. При этом применение технологий виртуализации позволит запускать приложения, не рассчитанные на работу в современных ОС. В настоящее время, создана базовая реализация основных компонент, которая используется для отладки архитектурных решений.

ЛИТЕРАТУРА

- [1]. Смагин С. И., Сапронов А.Ю., Тарасов А.Г., Шаповалов Т.С. Опыт построения GRID-сети ДВО РАН. // Современные информационные технологии для научных исследований. Материалы Всероссийской конференции (Магадан, 20-24 апреля 2008г.). Магадан: СВНЦ ДВО РАН, 2008. С. 23-24
- [2]. Тарасов А.Г. Взаимодействие системы мониторинга со сторонним программным обеспечением. Вычислительные технологии. Т.17, №2. 2012, С. 69-78.