ГИФ-ВЕБ-ПОРТАЛ ДЛЯ РАЗРАБОТКИ И ИСПОЛЬЗОВАНИЯ ГРИДОВСКИХ ПРИЛОЖЕНИЙ

В.А. Вознесенский

РЕЗЮМЕ

В статье описывается новый комплекс программного обеспечения, предоставляющий интернетовский сервис для конечных пользователей и разработчиков приложений, использующих в качестве основной вычислительной платформы инфраструктуру gLite/EGEE [1, 2]. Процесс разработки приложений состоит в описании управляющих сценариев на алгоритмическом языке Python [3] с использованием абстрагированного от конкретной вычислительной платформы интерфейса программирования, загрузке в систему файлов с данными и исполняемым кодом, а так же отладке сценариев и кода. Управляющие сценарии приложения инициализируют расчёт, декларируют и проверяют параметры расчёта, манипулируют необходимыми для управления данными, запускают и прерывают атомарные вычислительные задания, реагируют на завершение или сбой атомарных вычислительных заданий. Процессы разработки и использования приложений производятся пользователями через стандартные веб-броузеры, например [4].

ВВЕДЕНИЕ

Решение научных и практических задач, требующих проведения большого объёма вычислений, подразумевает, в общем случае, итерационный процесс, состоящий, как минимум, из двух фаз: описания конкретных заданий для автоматических вычислителей и реализации этих заданий на параллельных и/или распределённых вычислительных системах. Автоматизация распределения вычислительных заданий по вычислительным ресурсам имеет давнюю историю [5], проработанную теорию [6] и реализована в огромном количестве систем (например, [7, 8, 9]).

До проблемы автоматического описания топологий распределённых расчётов (перечней атомарных вычислительных заданий и потоков данных между ними) исследовательские умы добрались значительно позднее, потому ей посвящено заметно меньше литературы и разработок (например, [10]), хотя практически каждый расчёт, распределённый по большому количеству слабо связанных друг с другом вычислительных ресурсов, требует такого описания.

Другая проблема, препятствующая развитию отрасли больших вычислений - предоставление потенциальным пользователям, не имеющим специальных знаний в области высокопроизводительных вычислительных систем или доступа к прикладным проприетарным вычислительным кодам, возможности производить вычисления на таких платформах без непосредственного взаимодействия с кодом приложения.

В данной работе описывается попытка решить две последние обозначенные проблемы путём создания программного агента, с одной стороны предоставляющего пользователям интерфейс абстрактного описания как можно более широкого круга приложений и контроля за ходом расчётов и, с другой стороны, управляющего запуском автоматически продекларированных заданий на конкретной вычислительной инфраструктуре, в данном случае основанной на промежуточном программном обеспечении gLite[1], управляющем вычислительными ресурсами, собранными в рамках проекта EGEE[2].

ТРЕБОВАНИЯ И ПОДХОДЫ К ОПИСАНИЮ ЗАДАНИЙ И ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ

Управляющая расчётами логика приложений в наиболее широкой своей постановке, встречавшейся автору, требует от системы управления расчётами следующих возможностей:

1. Декларации новых вычислительных заданий с произвольно заданными входными и выходными данными, в том числе на основании данных уже успешно завершившихся заданий;
2. Декларации перечня входных параметров и результатов расчёта;
3. Изменения пользователем входных параметров расчёта, в том числе влияющих на перечень запущенных заданий;
4. Прерывания отдельных вычислительных заданий расчёта;
5. Доступности перечисленных возможностей не только до, но и после запуска первых заданий расчёта.

К сожалению, системы, позволяющие задавать статический ацикличный однонаправленный граф описания расчёта (например, [10, 11]), не удовлетворяют совокупности требований 1) и 5). Это, в частности, не позволяет организоввывать вариативные вычисления, меняющие параметры расчёта и перечень новых заданий по мере завершения старых заданий, в частности вычисления адаптивными методами типа Монте-Карло [12] и численную опти-

мизацию генетическими алгоритмами [13]. Кроме того, авторы данных систем уделяют слишком мало внимания автоматической генерации графов, что уменьшает и без того ограниченные практические возможности этих систем.

Кроме того, система управления не будет пользоваться успехом среди разработчиков и, в результате, конечных пользователей приложений без следующих свойств:

6. простой в обучении и применении высокоуровневый язык описания логики управления расчётом;
7. необременительный для разработчика способ сохранения и загрузки необходимых для управления данных;
8. обратная совместимость интерфейса программирования (API) управляющих сценариев с целью защиты инвестиций в разработку приложений.

На данный момент наиболее популярным представляется способ описания логики управления расчётом с помощью программ на специальных графических алгоритмических языках [14], реализованный, например, в [15, 16].

ПРЕДЛАГАЕМЫЙ ПОДХОД

По мнению автора, для описания логики управления расчётом предпочтительными являются текстовые языки программирования, такие как Python [3], созданные сообществами разработчиков и доказавшие свою эффективность в десятках тысяч приложений.

Использование Python позволяет описывать не только вычислительные задания, но и входные параметры и результаты расчёта, а использование объектной базы данных, например ZODB [17], снимает с разработчика бремя заботы о хранении необходимой для управления промежуточной информации.

Приложение в предлагаемой автором системе ГИФ (Гридовский ИнтерФейс) описывается пятью сценариями:

1. Инициализации;
2. Декларации пользовательского интерфейса, в том числе входных параметров расчёта;
3. Проверки и сохранения входных параметров по вводу пользователя;
4. Обработки успешного завершения вычислительного задания;
5. Опционально - обработки события сбоя запущенной задачи.

Объектная база данных доступна в каждом из сценариев. Для сохранения значения переменной между запусками сценариев её следует объявить принадлежащей глобальному пространству имён (global).

Для каждого из сценариев, кроме второго, доступны интерфейсные вызовы запуска и прерывания вычислительных заданий, а так же выдачи именованных результатов расчётов. Результаты расчётов могут быть впоследствии выгружены из системы пользователем, а так же использованы им в качестве файлов при задании входных параметров новых расчётов и описании новых приложений.

У разработчика приложений имеется возможность задавать в приложении один или несколько файлов для добавления их в глобальное пространство имён перед запуском инициализационного сценария расчёта в качестве текстовых строк. Это, в частности, полезно для загрузки в систему исполняемого кода приложения.

В программном вызове запуска задания передаётся перечень входных и выходных файлов задания. Входные файлы задаются как ассоциативный массив, привязывающий имена файлов к конкретным строковым значениям. Выходные файлы задаются как ассоциативный массив, привязывающий имена файлов к именам переменных, которые будут фигурировать в локальном пространстве имён четвёртого сценария при обработке успешного завершения данного задания.

Из каждого сценария можно специальными интерфейсными вызовами выдать сообщения автору и пользователю приложения.

Второй сценарий позволяет с помощью интерфейсных вызовов декларировать файловые и строковые параметры, кнопки отправки значений параметров в систему, текст для выдачи его пользователю. Кроме того, таким же образом можно опционально запросить у пользователя идентификатор прокси-сертификата X.509 [18], предварительно загруженного пользователем в систему и необходимого расчёту для запуска его вычислительных заданий.

Значения продекларированных входных параметров, заданные пользователем, обрабатываются в третьем сценарии. Передача этих значений в третий сценарий осуществляется путём их привязки к именам параметров в локальном пространстве имён сценария, аналогично привязке выходных файлов к именам переменных в четвёртом сценарии. Таким же образом, для идентификации нажатой пользователем интерфейсной кнопки отправки формы с параметрами расчёта, передаётся величина, предварительно заданная в декларации этой кнопки.

ПОРЯДОК И ТЕКУЩИЕ ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ СИСТЕМЫ

Перед началом работы с системой пользователь должен получить через один из регистрационных центров [19] X.509 сертификат [20]. Данный сертификат должен быть загружен в броузер пользователя для дальнейшей аутентификации в системе.

Для запуска расчётов пользователю понадобится загрузить в систему свой прокси-сертификат с VOMS-расширением [21] с помощью специальной программы [22]. На данный момент система ГИФ не имеет функциональности по обновлению VOMS-расширений, так что прокси-сертификат приходится обновлять один-два раза в сутки.

Вся остальная работа осуществляется через броузер пользователя. Пользователь может создать или отредактировать созданное им приложение, а так же создать расчёт путём выбора одного из приложений, предоставленных другим пользователем или им самим, из списка.

Задание сценариев приложений, параметров расчётов и т.д. осуществляется через соответствующие веб-страницы. Веб-страница с параметрами расчёта генерируется системой на основе деклараций во втором сценарии соответствующего приложения.

На данный момент авторы приложений могут выдавать доступ к ним конкретным пользователям. По мере развития системы будет добавлена возможность выдавать доступ представителям определённых виртуальных организаций, групп и ролей в них на основании VOMS-расширений прокси-сертификатов.

На данный момент в системе отсутствует возможность обновления прокси-сертификатов уже запущенных задач, так что нет возможности успешно запускать задания с временем выполнения на ресурсе большим, чем несколько часов. Данное ограничение будет снято реализацией в системе сервиса MyProxy [23].

Система реализована на языке Python. Для изоляции запущенных управляющих сценариев от вызовов операционной системы, внутренних объектов ГИФ и других расчётов используется программный пакет Restricted Python [24]. Информация хранится исключительно в объектной СУБД ZODB. Базовые сетевые сервисы и клиенты, а так же реактор обработки событий предоставлен пакетом Twisted [25]. Криптографическая функциональность реализована подправленным пакетом PyOpenSSL [26].

ЗАКЛЮЧЕНИЕ

Описанная система ГИФ предоставляет потенциальным разработчикам научных и прикладных вычислительных кодов широкие возможности по их портированию на инфраструктуру gLite/EGEE и организации доступа к ним со стороны потенциальных пользователей.

Существующие неудобства с прокси-сертификатами обусловлены новизной системы и будут сняты в следующих версиях. Дальнейшей проработки требуют вопросы производительности и манипуляции большими объёмами данных. Для этого требуется привлечение ряда реальных приложений.

Абстрактный интерфейс программирования сценариев управления, не привязанный к конкретному промежуточному программному обеспечению, даёт разработчику больше времени для решения прикладных проблем, а самой системе - свободу выбора базовой инфраструктуры.

Проблема необходимости привязывать исполняемый код приложений к архитектуре конкретных ресурсов (x86, AMD64 и др.) на данный момент не имеет общего решения и должна, по-видимому, решаться на уровне промежуточного между системой и ресурсами программного обеспечения. В частности, использование в расчётах под управлением ГИФ высокопроизводительных параллельных систем возможно при условии добавления таких систем в существующую доступную для автора инфраструктуру, возможно под управлением иного, чем gLite, промежуточного ПО.

Автор признателен сотрудникам ИИС РНЦ КИ В.Ю. Добрецову за инициативу, Е.А. Рябинкину за техническую поддержку и А.А. Солдатову за организационную возможность создания описанной технологии, а так же сотруднику НИИЯФ МГУ Л.В. Шамардину за техническую поддержку.

ЛИТЕРАТУРА:

1. EGEE > gLite // Сайт проекта EGEE на сервере Европейского Центра Атомных Исследований: URL: http://glite.web.cern.ch/glite/default.asp (2008. 31 мая)
2. EGEE Portal: EGEE Portal // Головной сайт проекта EGEE: URL: http://www.eu-egee.org/ (2008. 31 мая)
3. Python Programming Language -- Official Website // Сайт алгоритмического языка Python: URL: http://www.python.org (2008. 31 мая)
4. Mozilla Firefox | Mozilla Россия // Российский сайт фонда Mozilla Foundation: URL: http://www.mozilla-russia.org/products/firefox/ (2008. 31 мая)
5. John F. Shoch, Jon A. Hupp. The "Worm" Programs - Early Experience with a Distributed Computation // Communications of the ACM.- 1982.- Vol. 25.- No. 3.- P.p. 172-180.

6. Miron Livny. The Study of Load Balancing Algorithm for Distributed Processing Systems // Ph.D. thesis. Weizmann Institute of Science. Rehovot, Israel. August, 1983.
7. Condor Project Homepage // Сервер Факультета Вычислительных Наук Висконсинского Университета: URL: http://www.cs.wisc.edu/condor/ (2008. 31 мая)
8. Система X-COM // Сервер Лаборатории Параллельных Информационных Технологий НИВЦ МГУ: URL: http://x-com.parallel.ru/index.html (2008. 31 мая)
9. СКИФ-ГРИД: Главная // Сайт программы СКИФ-ГРИД на сервере ИПС РАН: URL: http://skif-grid.botik.ru/ (2008. 31 мая)
10. Pegasus :: Home // Сайт системы Pegasus на сервере Института Информационных Наук Университета Южной Калифорнии: URL: http://pegasus.isi.edu/ (2008. 31 мая)
11. P-GRADE // Сайт системы P-Grade на сервере Института Компьютерных и Автоматизационных Исследований Венгерской Академии Наук: URL: http://www.p-grade.hu/ (2008. 31 мая)
12. P.J. Smith, M. Shafi, and H. Gao. Quick simulation: A review of importance sampling techniques in communication systems // IEEE J.Select.Areas Commun., vol. 15, pp. 597—613, May 1997.
13. Л.А. Гладков, В.В. Курейчик, В.М. Курейчик. Генетические алгоритмы // Физматлит. Москва, 2006.
14. Grid Workflow Forum // Сервер Института Вычислительных Архитектур и Техники Программного Обеспечения им. Фраунхофера: URL: http://www.gridworkflow.org (2008. 31 мая)
15. Kepler: Kepler Project // Сервер Калифорнийского Университета: URL: http://kepler-project.org (2008. 31 мая)
16. A C++ Workflow Management System for e-scienze // Сайт проекта CppWfMS на сервере итальянского Национального Института Ядерной Физики: URL: http://wfms.forge.cnaf.infn.it/ (2008. 26 мая)
17. Zope.org - Zope Object Database // Сервер Zope Corporation: URL: http://www.zope.org/Products/StandaloneZODB (2008. 31 мая)
18. S. Tuecke, V. Welch, D. Engert, L. Pearlman, M. Thompson. Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile // IETF RFC 3820. June 2004. URL: http://www.ietf.org/rfc/rfc3820.txt (2008. 31 мая)
19. Russian Data-Intensive Grid Certificate Authority // Сервер ИИС РНЦ Курчатовский Институт: URL: http://ca.grid.kiae.ru (2008. 31 мая)
20. R. Housley, W. Ford, W. Polk, D. Solo. Internet X.509 Public Key Infrastructure Certificate and CRL Profile // IETF RFC 2459. January 1999. URL: http://www.ietf.org/rfc/rfc2459.txt (2008. 31 мая)
21. WP6 Authorisation Working Group // Сервер итальянского Национального Института Ядерной Физики: URL: http://grid-auth.infn.it/ (2008. 31 мая)
22. Gruppo GRID Datamat - EGEE - Wiki Pages - WMProxyClient - glite-wms-job-delegate-proxy // Сервер компании Datamat: URL: http://trinity.datamat.it/projects/EGEE/wiki/wiki.php? n=WMProxyClient.Glite-wms-job-delegate-proxy (2008. 26 мая)
23. MyProxy Credential Management Service // Сервер Иллинойсского Университета: URL: http://grid.ncsa.uiuc.edu/myproxy/ (2008. 31 мая)
24. Python Package Index: RestrictedPython 3.4.2 // Сервер фонда Python Software Foundation: URL: http://pypi.python.org/pypi/RestrictedPython (2008. 31 мая)
25. Twisted // Сайт проекта Twisted Python: URL: http://twistedmatrix.com/trac
26. PyOpenSSL - Python Interface to the OpenSSL library // Сайт проекта PyOpenSSL на сервере компании SourceForge Inc.: URL: http://pyopenssl.sourceforge.net/ (2008. 31 мая)