УДК 681.324

М.А. Волк, Т.В. Филимончук, Ал Шиблак Муаз

Харьковский национальный университет радиоэлектроники, Харьков

АНАЛИЗ СОВРЕМЕННОГО СОСТОЯНИЯ И РАЗВИТИЯ GRID-ТЕХНОЛОГИЙ И ЯЗЫКОВ ОПИСАНИЯ ЗАДАНИЙ

Аннотация: в работе представлен анализ современного состояния GRID-технологий, проанализированы достоинства и недостатки предложенного рынком прикладного программного обеспечения, главной функцией которого является предоставление пользователю удобного интерфейса между приложением и необходимыми вычислительными ресурсами. Рассмотрена эволюция языков описания заданий, отмечено отсутствие в GRID-системах стандарта, который позволяет описывать задания, что приводит к возникновению ряда проблем и ошибок при попытке запуска их на выполнение.

Ключевые слова: *промежуточное программное обеспечение, GRID-полигоны, распределенные среды, языки описания заданий.*

Введение

Постановка задачи. Первые работы в области GRID-вычислений начались в 90-х годах прошлого века. За это время наработано множество решений в научном и прикладном сегментах GRID.

Такое многообразие требует собственной классификации для облегчения понимания темы GRID-вычислений специалистами, которые могли бы воспользоваться GRID технологиями, но не имеют базовых знаний в данной области. Особенно важно потенциальным потребителям GRID-ресурсов разбираться в способах описания заданий для размещения своих задач на разнообразных GRID-системах, доступных широкому кругу пользователей.

Цель статьи. Провести анализ современного состояния и перспектив развития GRID-технологий и выполнить обзор языков описания заданий с целью облегчения принятия решения инженерными и научными сотрудникам по использованию в своих исследованиях и проектах распределенных ресурсов GRID-систем.

Анализ исследований и публикаций. В 2002 году в своей работе [1] Ян Фостер с помощью трех критериев вводит разграничение GRID-систем по научным и коммерческим результатам, которые получены в ходе решения ряда прикладных задач. В качестве критериев выступают определения, в соответствии с которыми GRID – это система, которая:

– координирует использование ресурсов при отсутствии централизованного управления этими ресурсами;

– использует стандартные, открытые, универсальные протоколы и интерфейсы;

– нетривиальным образом обеспечивает высококачественное обслуживание.

Для работы в GRID-системах используется прикладное программное обеспечение (ППО), которое выполняет две функции:

– облегчает доступ приложений к вычислительным ресурсам, что позволяет разработчикам информационных систем сделать акцент на бизнес логике, а не на разработке механизмов доступа к ресурсам;

– отвечает за ускорение процессов взаимодействия, т.к. обладает лучшей производительностью по сравнению с неспециализированным решением.

В данной статье введем негласное разделение ППО по трем направлениям. К первому отнесем программное обеспечение (ПО), которое соответствует критериям, перечисленным в работе [1] – это Globus toolkit, NurduGrid ARC, gLite. Перечисленное проекты – это основополагающее ППО, на основе которого строят свои исследования ученые, в том числе и ученые Украины.

Второе направление представляют системы Condor, Unicore, Legion. Представленные проекты – это доступные и стабильные среды, которые используются для организации распределенных вычислений, однако выбор определенной среды зависит от функций, которые она будет выполнять.

К третьему направлению отнесем GRID-полигоны, которые создаются для решения конкретной задачи (ряда задач) на определенном типе аппаратного и программного обеспечения. В настоящее к GRID-полигонам можно отнести следующие проекты: EGEE [2], DOE Science Grid [3], NorduGrid [4,5], Grid3 [6], BIRN [7].

Украинские ученые также принимали участие в совместных международных проектах и, начиная с 2004 года, отечественная академическая GRID-инфраструктура преобразуется в украинский национальный грид (УНГ). В мае 2011 года УНГ [8] стал участником NorduGrid, а в декабре 2011 года был подписан Меморандум о взаимодействиях между EGI и УНГ, т.е. продолжается интеграция украинской GRID-инфраструктуры в GRID инфраструктуру EGI.

Основная часть

Проект Globus

Проект Globus ориентирован на предоставление инфраструктуры для GRID-вычислений [9]. Целью его создания являлась предоставление пользователям возможности работать с распределенными вычислительными ресурсами как с единой системой. Проект Globus является продолжением развития проекта I-WAY, в ходе которого основной акцент был перенесен с поддержки высокопроизводительных вычислений в сторону сервисов поддержки виртуальных организаций [10].

Основным элементом проекта Globus выступает Globus Toolkit, который в 1998 году компания Globus Alliance впервые предложила сообществу как ППО. Последняя версия этого пакета (GT5) включает набор сервисов: Execution Management (управление заданиями), Information Services (информационные сервисы) и Data Management (управление данными).

Execution Management отвечает за управление заданиями и используется для управления, мониторинга и координации удаленного выполнения заданий, Information Services используется для описания служб и ресурсов, Data Management позволяет пользователям иметь доступ, передавать и управлять распределенными данными.

Globus Toolkit содержит набор модулей, которые используются для построения виртуальной организации при распределенных вычислениях. Каждый модуль определяет интерфейс, используемый высокоуровневыми компонентами, и имеет реализацию для различных сред выполнения.

На рис. 1 представлены группы модулей Globus Toolkit.

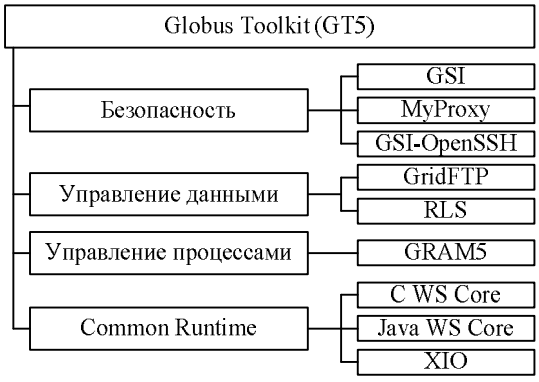

Рис. 1. Группы модулей Globus Toolkit

Безопасность в Globus Toolkit (GT5) достигается за счет авторизации и сертификации (X.509), которые реализуются с помощью компонентов защиты. Данные компоненты позволяют контролировать действия всех пользователей в GRID, защищать передаваемые данные, а также позволяют редактирование полномочий пользователей, с сохранением информации о членстве в группах доступа.

Grid Security Infrastructure (GSI) – это компонент, который предоставляет API для аутентификации, авторизации и сертификации. В качестве идентификаторов пользователей и ресурсов в GSI используются цифровые сертификаты X.509.

Для управления сертификатами X.509 и полномочиями в GT5 используется компонент MyProxy, который сочетает в себе онлайн репозиторий учетных записей и онлайн Центр Сертификации. Используя данную информацию, пользователи получают доступ к своим учетным записям не зависимо от времени и места.

Для работы с прокси-сертификатами X.509, обеспечивающим доступ к удаленным системам передачи данных без ввода паролей, служит модифицированный OpenSSH (GSI-OpenSSH).

Протокол передачи файлов GridFTP является высокопроизводительным, безопасным и надежным протоколом обмена данными, между ресурсами в GRID-системах. Протокол был разработан и оптимизирован для глобальных сетей с высокой пропускной способностью. GridFTP основан на протоколе FTP, однако в отличии от FTP использует несколько каналов для повышения скорости передачи.

RLS (Replica Location Service) – это сервер, обеспечивающий регистрацию и поиск ответной информации. Данный сервис позволяет отслеживать расположение дубликатов данных, т.е. участвует в определении физического расположения файла. Эти действия позволяют получить доступ к различным копиям одного и того же файла (например, выбирать файл, расположенный на ближайшем сервере).

Программный компонент GRAM5 (Globus Toolkit Resource Allocation Manager) используется для распределения (размещения) заданий, контроля за их выполнением, а также при необходимости для отмены заданий, поступивших на вычислительные ресурсы GRID-системы.

Common Runtime – это набор компонентов, которые используются для написания preweb- и web-сервисов. Данные компоненты содержат набор библиотек и инструментов, которые позволяют сделать эти сервисы платформенно-независимыми. Компоненты XIO используют набор расширяемых библиотек ввода/вывода, написанных на языке Си и обеспечивают простой интуитивный API. Компоненты C WS Core содержат библиотеки на языке Си, используемые для написания web-сервисов, а Java WS Core – библиотеки на языке Java.

Запросы к GRAM формируются с помощью специального языка RSL (Resource Specification Language), который определяет унифицированные для всей глобальной среды формы задания ресурсов и служит для реализации связи между компонентами GRID-среды, обслуживающими запросы. Построение запросов осуществляется на основании

спецификации параметров ресурсов, которые соединены между собой логическими операторами (И-ИЛИ-НЕ).

Язык RSL содержит 2 типа параметров, которые различаются по смыслу и способу обработки:

– ограничения на ресурсы формируются при помощи задания имен атрибутов (например, memory>=128, network=atm);

– локальные параметры, отображающие информацию о задании (count – число запрашиваемых процессоров, max_time – время выполнения и т.д.).

Далее по заданным атрибутам осуществляется подбор соответствующих ресурсов с помощью специализированного брокера, который опирается на данные о текущем статусе ресурсов, полученные от информационного сервиса. Результатом работы брокера является множество локальных менеджеров ресурсов, которые подходят для запуска определенного задания.

Ведущие производители ППО приняли Globus Toolkit в качестве стандарта «де-факто» для GRID-систем и с помощью его инструментария развертываются проекты по всему миру.

Сообщество NorduGrid ARC

Основная цель сообщества NorduGrid – создание и поддержка надёжного, легко масштабируемого, переносимого и функционально завершённого программного продукта для гетерогенных GRID-систем. ППО Advanced Resource Connector (ARC) предназначено для управления выполнением заданий в распределенной среде и выступает посредником между пользователем и доступными вычислительными ресурсами.

В ARC реализованы следующие функции:

– информационные (сбор информации о ресурсах GRID-системы);

– динамическое подключение ресурсов в GRID-систему;

– распределение заданий по ресурсам;

– пересылка заданий на выполнение и управление этими заданиями;

– мониторинг GRID-системы.

Все функции ARC реализованы в виде отдельных служб, каждая из которых основывается на известные программные средства с открытым кодом: OpenLDAP, OpenSSL, SASL.

ARC – это коннектор для проекта стран Северной Европы, который ориентирован на поддержку приложений физики высоких энергий. Его реализация выполнена с помощью библиотек Globus Toolkit 2 (GT2), безопасность достигается путем использования протоколов и инфраструктурных решений GSI.

Для описания запроса на выполнение задания в NorduGrid ARC используется расширенный язык запроса ресурсов XRSL (eXtended Resource Specification Language).

Язык xRSL, как и язык RSL, используется для передачи информации и требований на удаленный ресурс. В качестве требований могут выступать следующие параметры: архитектура, оперативная память, количество процессоров и т.д. Версии v1.0 и v.1.1 языка RSL используются в GRAM5 на платформе Globus Toolkit, а xRSL (расширенная версия RSL) применяется в платформе ARC. Отличие языков RSL и xRSL заключается не только в наличии новых атрибутов, но и в уровнях спецификации: User-side RSL – набор атрибутов, задаваемых пользователем в файле-спецификации; GridManager-side RSL – набор атрибутов из пользовательского файла-спецификации, переработанных по определенным правилам для передачи непосредственно в GridManager.

К достоинствам NorduGrid ARC можно отнести простоту в использовании (клиентские компоненты пакета легко устанавливаются, а серверные компоненты не требуют реконфигурирования сайта), пакет является свободно распространяемым, с открытым исходным кодом. Отличительная особенность NorduGrid ARC – собственный набор сервисов, заменяющих GT2. Однако многие функции GT2 в ARC реализуются собственными средствами. Кроме того, расширен язык описания ресурсов xRSL.

ППО gLite

ППО gLite опирается на опыт ряда крупных европейских проектов: EDG (European Data Grid), LCG, Alien, Nordugrid. В 2004 году стартовал проект EGEE (Enabling Grids for E-sciencE) на ППО LCG-2, используемом в проекте LCG. Параллельно происходила разработка ПО gLite и в 2006 году вышла версия 1.5 gLite, которая обеспечивала полный набор основных GRID-служб. В дальнейшем ППО gLite было использовано в EGI (The European Grid Infrastructure).

Высокоуровневые сервисы gLite подразделяются на следующие группы:

– сервисы безопасности;

– информационные службы и мониторинг;

– сервисы управления заданиями.

Для запуска задания в GRID, используя gLite, необходимо создать файл, который содержит описание данного задания на языке JDL (Job Description Language).

Все атрибуты языка JDL разделены на 2 группы: обязательные и второстепенные. В случае отсутствия обязательных атрибутов в JDL файле WMS не сможет выполнить запрос на поиск и размещение задания. Для второстепенных атрибутов система может самостоятельно определить ряд значений по умолчанию для обработки запроса.

Язык JDL поддерживает следующие типы запросов для системы WMS:

– простая задача (Job);

– набор зависимых задач (DAG);

– набор независимых задач (Collection).

Тип запроса задается атрибутом Type, по умолчанию значением атрибута Type является Job, для которого имеется возможность определения типа задачи:

– обычное задание (Normal);

– интерактивное задание (Interactive);

– параметрические задания (Parametric);

– параллельное приложение, использующее протокол распараллеливания процессов MPI (MPICH);

– задания с контрольными точками (Checkpointable);

– серийные задания (Partitionable).

Запрос типа DAG – это набор задач, в котором вход, выход или выполнение одних задач может зависеть от других. Такие задания представляются в виде графа, где в качестве вершин выступают задачи, а ребра определяют зависимости между ними, которые не могут быть циклическими.

Структура описания DAG включает ряд вложенных секций в следующей иерархии:

– корневая секция (root), которая содержит полное описание задания с рядом атрибутов, которые наследуются всеми задачами задания;

– дочерняя секция (nodes) содержит описание всех задач, которые включены в задание, а также дополнительно атрибут Dependencies, который определяет зависимости между всеми задачами задания;

– набор секций, каждая из которых соответствует определенной задаче задания и содержит ее полное описание.

Запрос Collection позволяет осуществлять мониторинг и управление заданиями как через уникальный ID всего задания, так и через назначенные ID для отдельных задач задания.

Интерактивные задания позволяют пользователю взаимодействовать с заданием в реальном времени.

Параметрические задания – это задания, где один или более его атрибутов параметризируемы и для каждого значения параметра создается отдельная задача. Атрибуты параметрического задания могут изменять свои значения в соответствии с атрибутом Parameters, который определяется в описании определенного задания. Запуск параметрического задания приводит к запуску набора задач, имеющих одинаковые описания кроме значений параметрических атрибутов. Параметрическое задание, а также все созданные задачи, получают уникальные ID, что позволяет осуществлять мониторинг и управлять как каждой задачей по отдельности, так и всем заданием в целом.

Message Passing Interface (MPI) – это общепризнанный стандарт в параллельном программировании с использованием механизма передачи сообщений. Если при описании задания в атрибуте JobType выбрано значение MPICH, то это значит что атрибут NodeNumber определяет число необходимых процессоров для распараллеливания данного задания.

Для заданий с контрольной точкой в WMS имеется возможность сохранения состояния, что позволяет перезапускать его в случае сбоя не с начала, а с какого-то промежуточного этапа. Это свойство представляется необходимым при долговременном счете. Обязательными атрибутами для данных заданий являются атрибуты JobSteps и CurrentStep. Первый из них содержит имена контрольных точек или задает их максимальное количество, а второй определяет, с какой точки должен быть выполнен запуск.

При запуске серийного задания происходит разбиение его на подзадания с одним и тем же исполняемым файлом, но с различным исходным набором данных. В JDL-файле описывается одна сериализуемая составляющая и составляющие для пред- и пост- обработки. Задание предобработки выполняется первым и от него зависят все серийные заданий, задание постобработки зависит от всех серийных, и в его функцию входит сборка полученных результатов.

Пакет gLite – это решение для GRID-систем, которое включает в свой состав базовые низкоуровневые программы и службы высокого уровня, т.к. в него были интегрированы компоненты многих проектов ППО (Condor, Globus Toolkit, LCG). Одно из основных достоинств gLite заключается в том, что помимо разработанных собственных компонентов, он позволяет объединять ряд решений, разработанных вне проекта, например WMS, VINE, CREAM. Т.к. архитектура ППО gLite является сервис-ориентированной, то сервисы в ней могут работать как объединённо (выполнять задачу конечного пользователя), так и изолировано (в контексте самостоятельной задачи).

Система Condor

Широкое распространение получила система Condor [11], созданная в университете штата Висконсин (США) для объединения компьютеров университета. Она используется для решения ряда задач, которые требуют большое количество вычислительных ресурсов. Система Condor позволяет распределять задания как на отчуждаемые, так и не неотчуждаемые ресурсы, что позволяет эффективно использовать их вычислительные мощности. ПО системы Condor доступно для пользователей бесплатно.

Condor обладает гибкой системой распределения ресурсов, которая реализована с помощью языка ClassAd (Classified Advertisement). ClassAd один из первых языков описания заданий, который разработан для систем управления рабочими потоками, основанных на Condor. ClassAd является функцио-

нальным языком, который позволяет пользователю создавать описания ресурсов и заданий, поступающих на выполнение. Данный язык легко расширяем, что позволяет легко наращивать требуемый список атрибутов. В Condor на основе языка ClassAd реализован механизм matchmaking, который осуществляет поиск ресурсов для каждого задания из очереди путём сопоставления информации о ресурсах и запросов заданий. Основной единицей языка являются выражения (expression), т.к. выполнение задания влечет за собой вычисление всех выражений. Язык ClassAd в дальнейшем был использован как основа для построения языка JDL.

Condor поддерживает механизмы миграции заданий, создание контрольных точек, удалённый вызов процедур. В случае возрастания активности владельца исполнительного компьютера механизм миграции совместно с механизмом создания контрольных точек позволяют продолжить обработку задания на другом компьютере с того места, на котором она приостановилась.

Механизмы безопасности и поддержания целостности данных отвечают требованиям безопасности, обеспечивают надежную передачу информации между компонентами системы, гарантируют защиту данных пользователя (владельца задания) и владельца ресурса. Одним из возможных способов аутентификации в системе является GSI аутентификация на основе публичных ключей с использованием сертификатов.

Платформа Unicore

Платформа UNICORE (UNiform Interface to COmputing REsources) предоставляет сообществу ученых вычислительные ресурсы, которые объединены в GRID-системы. Платформа UNICORE используется в международных и локальных проектах: DEISA (European Distributed Supercomputing Infrastructure), National German Supercomputing Center NIC, PRACE (European PetaFlop HPC Infrastructure).

Пользовательский интерфейс платформы UNICORE позволяет пользователю создавать, запускать, а также управлять заданием с любой рабочей станции или ПК в интернете.

Платформа реализована в виде трехуровневой архитектуры: клиент-шлюз-сервер. Первый уровень содержит приложения (графический интерфейс, монитор заданий), при помощи которых пользователь может подготовить или внести изменения в существующие задания, а также получить результаты их выполнения. Второй уровень – это шлюз, который контролирует авторизацию пользователей и идентификацию заданий. Т.к. все задания в UNICORE строятся на наборе взаимосвязанных задач, которые связаны между собой временными зависимостями, то для каждого задания определяется исполнительная система и требования по ресурсам для каждой задачи. На данном уровне происходит перевод заданий в специальные команды, с которыми работает вычислительный ресурс. На третьем уровне (уровень сервера) авторизированное задание подтверждается (имеется ли ресурс для запуска задания) и отправляется на выполнение.

Для описания заданий в UNICORE используется язык JSON (JavaScript Object Notation). JSON – это текстовый формат обмена данными, который основан на языке JavaScript, имеется интуитивно-понятный графический интерфейс. Язык строится на основе двух универсальных структур данных: наборе пар ключ/значение и пронумерованном наборе значений, которые реализованы с помощью следующих форм: объект, одномерный массив, значение, строка.

В языке JSON принят следующий синтаксис:

– объект – это неупорядоченное множество пар имя/значение, которое заключено в фигурные скобки { }. Между именем и значением ставят символ ":", а пары имя/значение разделяются запятыми;

– одномерный массив – множество значений, которые имеют порядковые номера (индексы), заключенные в квадратные скобки [ ], значения массива отделяются между собой запятыми;

– значение может быть строкой в двойных кавычках, числом, значением true или false, объектом, массивом, или значением null (эти структуры могут быть вложены друг в друга);

– строка – это упорядоченное множество из нуля или более символов юникода, которое заключено в двойные кавычки, с использованием escape-последовательностей начинающихся с обратной косой черты (backslash). Символы представляются простой строкой.

Особенностью платформы UNICORE является то, что она не использовала Globus Toolkit и была ориентирована на доступ к вычислительным центрам стран Центральной Европы. Безопасность в системе UNICORE, как и в GT5, соответствует сертификату X.509.

Проект Legion

Проект Legion впервые был представлен в университете в 1993 году. Основная его цель состояла в построения метакомпьютерной среды, независимо от масштаба, географического положения, языка или операционной системы [12]. В архитектуре системы используется подход, который основан на принципах объектно-ориентированного программирования. Все сущности в системе являются объектами (включая файлы, компьютеры, сеть) со специфическими процедурами доступа и исполняющимися на виртуальной машине. Однако данный подход не получил широкого распространения в следствии того, что приложения для GRID не являются объектно-ориентированными.

Legion предоставляет пользователю набор объектов, предоставляющих базовые сервисы:

– объекты вычислителей – абстракции, реализующие базовые принципы работы с вычислительными ресурсами;

– объекты систем хранения данных – абстракции, предоставляющие базовые методы работы с системами хранения данных;

– объекты связывания – объекты, обеспечивающие связи между абстрактным идентификатором объекта и его физическим адресом;

– объекты контекста – объекты, реализующие проекцию пользовательских имен объектов на абстрактные идентификаторы объектов в системе Legion.

Legion – это проект, во многом разделяющий с Globus Toolkit базовые принципы построения GRID-систем, однако в отличие от Globus Toolkit, в котором акцент делается на стандартизацию удаленного взаимодействия, ориентирован на объектно-ориентированную модель.

В настоящее время проект Legion развивается в двух направлениях: исследовательском и коммерческом. Компания Avaki поставляет коммерческие решения на основе модели системы Legion. Имеется виртуальная машина для Legion-объектов – грид Centurion.

GRID-полигоны

Проект Enabling Grids for E-sciencE (EGEE) финансировался Европейским Сообществом и странами-участницами. В результате его действия появилась высокопроизводительная всемирная инфраструктура, превосходящая по своим возможностям локальные кластеры и отдельные центры. Проект EGEE завершен в апреле 2010 года.

Worldwide LHC Computing GRID (WLCG) был принят в 2001 году в ЦЕРНе. Основной целью проекта WLCG было создание глобальной информационно-вычислительной инфраструктуры для обработки, хранения и анализа данных, которые получены во время экспериментов, проводимых на Большом адронном коллайдере. Для реализации этого проекта была развернута масштабная глобальная GRID-инфраструктура на основе региональных центров различного уровня.

После завершения проекта EGEE с апреля 2010 года начал функционировать проект European Grid Infrastructure (EGI). EGI является преемником EGEE-III и занимается физикой высоких энергий, и самым ярким ее проектом является Большой адронный коллайдер.

Distributed European Infrastructure for Supercomputing Applications (DEISA) представляет собой консорциум ведущих национальных суперкомпьютерных центров, который был создан для содействия важнейшим мировым научным исследованиям и включающий в себя 18 компьютерных центров различной архитектуры, расположенных по всей Европе.

The Partnership for Advanced Computing in Europe (PRACE) представляет собой инфраструктуру, схожую с консорциумом DEISA, которая объединяет три суперкомпьютерных центра суммарной производительностью 4 PFLOPS.

Open Science Grid (OSG) – открытый научный GRID или сообщество GRID, которое было создано в США с целью проведения научных исследований, которые требуют большие вычислительные мощности. В настоящее время OSG обслуживает один из популярных порталов по моделированию в нанотехнологиях (nanoHUB.org).

Multiscale APPlications on European e-infRastructures (MAPPER) – проект по созданию общей среды для распределенных вычислений, который объединяет ряд научных организаций.

Прект TeraGrid является самой большой в мире распределенной вычислительной системой для научных исследований, которая объединяет суперкомпьютеры и исследовательские центры по все стране. В списке проектов, в которых используется TeraGrid присутствуют расшифровка генома человека, изучение механизма работы мозга, прогнозы погоды и моделирование торнадо в реальном времени.

GridPP – это объединение физиков ядерщиков и компьютерных ученых из Великобритании и CERN. Они построили распределенную GRID-инфраструктуру на территории Великобритании. На данный момент она работает на базе 17 институтов, помогая обрабатывать данные из самого большого в мире ускорителя частиц Большого адронного коллайдера. Эта GRID добавляет дополнительно в мировую (LHC Computing Grid) и европейскую (European Grid Infrastructure) GRID-инфраструктуры 40 тыс. компьютеров, что позволяет 20 тыс. специалистам использовать данные мощности.

Курс на стандартизацию описания заданий для GRID-систем

Важное место в организации GRID-систем занимает функция управления заданиями, которую реализует система управления рабочими потоками (Workflow Management System). В ее состав входит набор программных компонентов, которые выполняют распределение поступающих заданий в систему по имеющимся ресурсам GRID, а также управление ими. Реализация данной системы использует свой язык для описания заданий и управления рабочими потоками. Как правило, любое описание задания содержит следующую информацию: описание поставщика заданий, перечень требований задания к ресурсам, цели распределения.

Так как файлы описания заданий не стандартизированы, в различных GRID-системах могут ис-

пользоваться различные форматы описания, что вносит дополнительные проблемы в ходе запуска заданий на выполнение. В связи с этим, рядом ученых был предложен стандарт описания заданий, в качестве основы был взят язык описания задач JSDL (Job Submission Description Language), разработанный в 2008 г в рамках работы Open Grid Forum.

Предложенный стандарт GFD.136 Rec [13], основан на языке XML и содержит в себе описание элементов (и их допустимых значений), при помощи которых можно формировать как собственно задание, так и требования на программно-аппаратную платформу, на которой это задание может быть выполнено. Для высокопроизводительных систем было разработано расширение JSDL-WG, главным преимуществом которого является возможность проверки структуры и типов данных на соответствие заданной схеме. Языки JSDL/JSDL-WG хорошо зарекомендовали себя в мировой практике и используется на платформах UNICORE и ARC.

Еще одним направлением в области промышленного программирования является использование языка XML [14] как основы для разработки предметно-ориентированных языков, в том числе и для распределенного и параллельного программирования. Данный подход позволяет автоматизировать лексический, синтаксический и частично семантический анализ предметно-ориентированного языка. Кроме этого, подход к проектированию языка на основе XML позволяет воспользоваться готовым визуальным редактором или разработать собственный редактор языка.

Выводы

В статье приведены результаты анализа современного состояния и перспектив развития GRID-технологий. Представленный обзор может быть полезен инженерным и научным сотрудникам, которые планируют использовать в своих исследованиях и проектах распределенные ресурсы GRID-системы. Работа также может быть использована студентами и аспирантами для изучения существующих реальных систем, предоставляющих доступ к ресурсам, позволяющим решать задачи большой размерности в различных областях науки и техники.

Список литературы

1. Foster I. What is the Grid? A Three Point Checklist / I. Foster // GRIDToday. – 2002. – Vol. 1. – № 6. – P. 24-27.

2. Enabling Grids for E-sciencE project [Электронный ресурс]. – Режим доступа: <http://www.eu-egee.org>.

3. DOE Science Grid [Электронный ресурс]. – Режим доступа: <http://doesciencegrid.org>.

4. Eerola P. The NorduGrid production Grid infrastructure, status and plans / P. Eerola // Proc. 4th Intel Workshop on Grid Computing (GRID'03). IEEE CS Press. – 2003. P. 158-165.

5. NorduGrid [Электронный ресурс]. – Режим доступа: <http://www.nordugrid.org>.

6. The Grid3 collaboration [Электронный ресурс]. – Режим доступа: <http://doesciencegrid.org>.

7. The Biomedical Informatics Research Network (BIRN) Portal [Электронный ресурс]. – Режим доступа: <https://legacy-portal.nbirn.net/BIRN>.

8. Украинский национальный грид [Электронный ресурс]. – Режим доступа: <http://infrastructure.kiev.ua>.

9. Foster I. Globus Toolkit Version 4: Software for Service-Oriented Systems / I. Foster // IFIP International Conf. on Network and Parallel Computing. – 2005. – P. 2-13.

10. Радченко Г.И. Распределенные вычислительные системы / Г.И. Радченко. – Челябинск: Фотохудожник, 2012. – 184 с.

11. Condor [Электронный ресурс]. – Режим доступа:< http://research.cs.wisc.edu/htcondor>.

12. Grimshaw A.S. The Legion Vision of a Worldwide Virtual Computer / A.S. Grimshaw, W.A. Wulf // Comm. of the ACM. – 1997. – Vol. 40. – № 1.

13. Журавлёв Е.Е. Разработка первого национального стандарта для обеспечения интероперабельности в Грид-среде / Е.Е. Журавлёв, В.Н. Корниенко, А.Я. Олейников, Т.Д. Широбокова // «Журнал радиоэлектроники» – №2. – 2011.

14. Extensible Markup Language (XML) [Электронный ресурс]. – Режим доступа: <http://www.w3.org/XML>.

Поступила в редколлегию 19.04.2013

Рецензент: д-р техн. наук, проф. О.Ф. Михаль, Харьковский национальный университет радиоэлектроники, Харьков.

АНАЛІЗ СУЧАСНОГО СТАНУ І РОЗВИТКУ GRID-ТЕХНОЛОГИЙ І МОВ ОПИСУ ЗАВДАНЬ

М.О. Волк, Т.В. Філімончук, Ал Шиблак Муаз

В роботі подано аналіз сучасного стану GRID-технологій, проаналізовані переваги і недоліки прикладного програмного забезпечення, що запропоновано ринком. Головною функцією цього забезпечення є надання користувачу зручного інтерфейсу між додатком і необхідними обчислюваними ресурсами. Розглянуто еволюцію мов опису завдань, відмічена відсутність у GRID-системах стандарту, який дозволяє описувати завдання, що призводить до виникнення ряду проблем та помилок при спробі запуску їх на виконання.

Ключові слова: *проміжне програмне забезпечення, GRID-полігони, розподілені середовища, мови опису завдань.*

ANALYSIS OF THE CURRENT STATUS AND DEVELOPMENT OF GRID-TECHNOLOGIES AND JOB DESCRIPTION LANGUAGES

M.O. Volk, T.V, Filimonchuk, Al Shiblak Muaz

In the article the analysis of the current state of GRID-technologies, the advantages and disadvantages of application middleware. The main function of the middleware is to provide a user-friendly interface between the application and computing resources. Considered by the evolution of the job description languages, noted the lack of GRID-systems standard job description. This leads to a number of problems and errors when trying to start them up.

Keywords: *Middleware, GRID-polygons, distributed environments, job description languages.*