

Информационная система ГридННС*

А.П. Крюков¹, Л.В. Шамардин²

Научно-исследовательский институт ядерной физики имени Д.В.Скобельцына Московского государственного университета имени М.В.Ломоносова (НИИЯФ МГУ), 119991, Москва, Ленинские горы, д. 1

Грид-инфраструктура Национальной нанотехнологической сети (ГридННС) предназначена для предоставления ученым, инженерам, аспирантам и студентам, работающим в области нанонаук, унифицированного безопасного удаленного доступа к суперкомпьютерным ресурсам ННС. Одним из главных отличий ГридННС от других грид-инфраструктур является использование архитектурного стиля REST для реализации грид-сервисов. В настоящей работе описана архитектура информационной системы ГридННС, построенная на основе RESTful-веб-сервисов, структура публикуемой информации и ее модель безопасности.

Ключевые слова: высокопроизводительные вычисления, распределенные вычисления, грид, ГридННС, REST, RESTful-веб-сервисы, информационная система.

1. Введение

В 2000 году Р. Филдинг [1] был предложен новый, простой в использовании и гибкий подход к созданию веб-сервисов — архитектурный стиль REST. Важным достоинством RESTful-сервисов по сравнению с подходом основанным на WSRF является простая и ясная семантика запросов, соответствующая интуитивно ясным методам HTTP протокола. Кроме того, RESTful сервисы используют протокол HTTP в своем прямом предназначении как протокол запросов, а не просто транспортный протокол, как это происходит в случае WSRF, где процесс сериализации и десериализации создает дополнительные проблемы, в том числе и с совместимостью реализаций.

В процессе работы на проекте ГридННС [2] авторы использовали архитектурный стиль REST для реализации грид-сервисов в рамках концепции OGSA. В частности, на этой основе был реализован ключевой сервис в ГридННС — сервис управления потоком заданий Pilot [3,4].

Информационная система (ИС) ГридННС первоначально была построена на базе WS-MDS из инструментального набора Globus Toolkit 4. Однако, сложность использования и расширения возможностей WSRF сервисов поставила вопрос о переходе на более удобные в использовании, RESTful-веб-сервисы. Вопрос о новой реализации информационной системы для грида стал еще более актуальным после того как стали известны планы разработчиков Globus Toolkit об отказе от использования WSRF сервисов в последней версии Globus Toolkit 5, а также отсутствия в его составе информационной системы и инструментария для ее создания.

2. Структура ГридННС

Грид-инфраструктура Национальной нанотехнологической сети (ГридННС) предназначена для предоставления ученым, инженерам, аспирантам и студентам, работающим в области нанонаук, унифицированного безопасного удаленного доступа к суперкомпьютерным ресурсам ННС.

* Работа была выполнена при финансовой поддержке Министерства образования и науки РФ (контракт № 07.514.11.4022) и гранта РФФИ № 11-07-00434-а.

¹ kryukov@theory.sinp.msu.ru

² shamardin@theory.sinp.msu.ru

Структуру ГридННС в самом общем виде можно представить в виде трех слоев: слой пользовательских интерфейсов, слой грид-шлюзов к ресурсам и слой общих инфраструктурных сервисов.

Слой пользовательских интерфейсов (ПИ) предназначен для запуска пользователями своих заданий в ГридННС.

Слой общесистемных сервисов отвечает за функционирование и управление ГридННС. Основными общесистемными сервисами являются информационная система, система учета, сервис управления и распределения задач, система регистрации сервисов и ресурсов, служба выдачи и управления цифровыми сертификатами, сервис проверки работоспособности ресурсов.

Слой грид-шлюзов обеспечивает доступ к вычислительным ресурсам, подключенным к ГридННС. Грид-шлюз (ГШ) — это комплекс сервисов, который обеспечивает всю функциональность, необходимую для использования кластера или суперкомпьютера из ГридННС. Одним из компонентов ГШ является сервис информационной системы. Другой компонент информационной системы, агрегатор («hub»), является частью общесистемных сервисов. Далее в статье подробно описывается устройство и особенности функционирования информационной системы ГридННС.

Важный принцип, который был положен в основу проектирования ГридННС — это отказ от установки дополнительного ПО на вычислительном ресурсе. Все промежуточное ПО устанавливается только на специальном сервере — грид-шлюзе.

3. Информационная система ГридННС

Архитектура ИС ГридННС изображена на рисунке 1. Основная задача ИС состоит в том, чтобы обеспечить пользователей и сервисы ГридННС информацией о текущем состоянии ресурсов. В качестве источников информации о ресурсе выступают файл со статическим описанием ресурса, а также провайдер динамической информации о ресурсе. Статическая информация включает, например, название ресурса, его местоположение. Первоисточником динамической информации является локальный менеджер ресурсов (ЛМР), которым является система управления пакетной обработкой (СУПО) вычислительного ресурса.

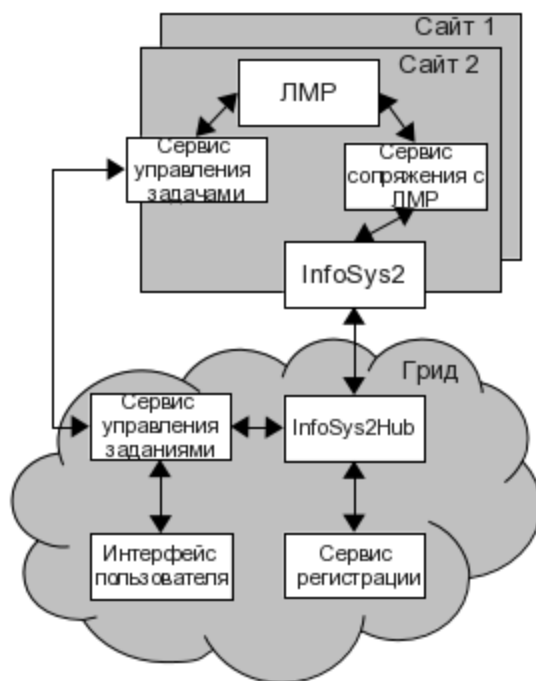


Рис. 1. Архитектура информационной системы ГридННС

Важным моментом построения ИС является вопрос о схеме данных. В качестве прототипа схемы данных использовалась GLUE Scheme 2.0 [5]. В качестве формата для обмена данными используется JSON [6]. Общая структура публикуемой информации по каждому вычислительному ресурсу имеет следующий вид, показанный на рисунке 2.

```
{
  ...
  «Site»: {
    ...
    «Cluster»: [
      ...
      «SubCluster»: [
        ...
        {
          ...
          «Queue»: [...]
          «Host»: [...]
          «Software»: [...]
        }
      ]
    ]
  }
  ...
}
```

Рис. 2. Общая структура публикуемой информации

Таким образом, каждый ресурс (сайт) представляет собой список кластеров, которые в свою очередь содержат подкластеры. Разбиение кластера на подкластеры позволяет учесть неоднородную структуру вычислительного ресурса. Структура подкластера предполагается однородной.

Статическая часть информации о ресурсе содержит общее описание ресурса, такие как его наименование, местоположение, административные контакты и так далее.

Информация о кластерах, зарегистрированных на одном ресурсном центре, помещается в секцию «Cluster», которая включает следующую информацию (см. рис. 3).

```
{
  "SubCluster": [
    {
      "Name": "rc.ru/subcluster0",
      "PhysicalCPUs": 16,
      "LogicalCPUs": 16,
      "Queue": [ ... ],
      "Host": [ ... ],
      "Software": [ ... ],
    },
    ...
  ] ...
}
```

Рис. 3. Структура секции «Cluster»

Структура рабочих узлов (серверов) описывается в секции «Host». Сюда относится информация о объеме оперативной памяти, типа ЦПУ, тип ОС и ее версия.

Доступное пользователям ПО публикуется в секции «Software» (рис.4.).

```

"Software": [
  {
    "LocalID": "hpmpi-2.02.05.01-20070708r",
    "Version": "2.02.05.01-20070708r",
    "Name": "hpmpi",
    "InstalledRoot": "/opt/hpmpi"
  },
  {
    "ModuleName": "lmp",
    "LocalID": "lammps-25Sep11",
    "EnvironmentSetup": [
      {
        "softenv": "+lammps-25sep11"
      }
    ],
    "ACL": {
      "Rule": [
        "VOMS:/sysadmin",
        "VOMS:/gridnnn",
        "VOMS:/education",
        "VOMS:/abinit"
      ]
    },
    "Version": "25Sep11",
    "Name": "lammps",
    "InstalledRoot": "/shared/lammps"
  }
] ...

```

Рис. 4. Структура секции «Software»

В секции, описывающей ПО, важную роль играют права доступа (ACL), с помощью которых системный администратор может сообщить о существовании ограничений на использование тех или иных пакетов прикладного ПО членами конкретных ВО. Таким образом, публикуется локальная политика использования прикладного ПО в зависимости от принадлежности к ВО.

Другой важной частью схемы является секция «EnvironmentSetup», в которой указываются метки программного окружения softenv или другие расширения, по наличию которых программное обеспечение грид-шлюза осуществляет специальную настройку среды выполнения для обеспечения доступа к соответствующему ПО некоторым стандартным образом. Например, этой настройкой может быть добавление необходимых путей в переменные PATH и LD_LIBRARY_PATH. Данная необходимость связана с тем, что прикладное ПО на ресурсах может быть установлено различным способом. Простейший пример — использование различных корневых директорий для прикладных пакетов. Так как пользователь в момент запуска задания не знает на каком ресурсе будет выполнена его задача, необходим механизм «настройки» задачи в момент запуска ее на конкретном ресурсе. Использование механизма SoftEnv позволяет пользователю не знать особенности установки ПО на ресурсах грида, что существенно упрощает запуск задач в такой неоднородной среде.

Другим важным атрибутом ресурса, публикуемым в информационной системе является информация о свойствах очереди, и, в частности, об обслуживаемых ВО. Секция Queue содержит так же динамическую информацию о состоянии очередей СУПО каждого подкластера вычислительного ресурса. На рисунке 5 представлено описание параметров доступных очередей. В этой секции публикуется динамическая информация, которая получается от СУПО.

```

"Queue": [
  {
    "CEInfo": "example.com/batch-A",
    "Feature": [ "mpi", "single" ],
    "ACL": {
      "Rule": [
        "VOMS:/nnn-vo-0",
        "VOMS:/nnn-vo-1",
        "VOMS:/nnn-vo-2/group1",
        "VOMS:/nnn-vo-3/Role=VO-Admin"
      ]
    }
    «MaxWallTime»: 6000,
    «MaxTotalJobs»: 100,
    «MaxRunningJobs»: 50,
    «RunningJobs»: 30,
  }
] ...

```

Рис. 5. Описание параметров доступных очередей

В приведенном примере в качестве динамической информации представлено количество запущенных заданий (RunningJobs).

Данная секция так же как и секция «Software» имеет подсекцию «ACL», которая позволяет указать права доступа к очереди членам ВО. Параметры очередей СУПО устанавливают политику администрации вычислительного ресурса по отношению к потребленным компьютерным ресурсам, таким как время счета, максимальное количество доступных вычислительных ядер. Таким образом, данная секция позволяет информировать пользователей об ограничениях на возможное количество потребляемых ресурсов.

Совместно, секция «Software» и секция «Queue» обеспечивают эффективный и гибкий механизм управления доступом к вычислительным ресурсам.

Информация с каждого ресурса агрегируется специальным RESTful сервисом InfoSys2Hub, который собирает информацию с сайтов, опубликованных в сервисе регистрации грид-сервисов ГридННС, с учетом их режима работы («работает» или «тестируется»). Такая структура позволяет потребителям информационного сервиса иметь дело с единственной «точкой входа» в инфраструктуру. Регистрация нового ресурса автоматически приводит к появлению информации о нем в ИС, изменения состояния сайтов также отражаются в общей информационной системе.

Безопасность ИС построена на использовании цифровых сертификатов стандарта X.509 инфраструктуры публичных ключей (PKI). Доступ к сервису предоставляется конечному набору потребителей в соответствии с конфигурацией сервиса.

4. Заключение

В процессе разработки ГридННС был решен ряд принципиальных вопросов создания грид-инфраструктур на основе RESTful-грид-сервисов.

В частности, была реализована информационная система как RESTful сервис, который обеспечивает публикацию информации о состоянии ресурсов, ее агрегацию со всех доступных для пользователей ресурсов, предоставление ее потребителям, которым она необходима.

В качестве формата обмена информацией был использован JSON. Гибкость данного формата позволила реализовать подмножество публикуемых параметров, которые описаны в GLUE Scheme 2, являющейся стандартом для построения ИС в грид-инфраструктурах.

Использование RESTful сервисов, позволило использовать HTTP протокол не только в качестве транспортного протокола, а в своем прямом назначении, как протокола запросов с ясной

семантикой. Это позволило значительно упростить серелизацию и десерелизацию информации, что улучшило надежность системы.

В настоящее время данная версия ИС успешно прошла тестирование на полигоне ГридННС и в ближайшее время будет внедрена в инфраструктуру в качестве основной.

Литература

1. Fielding R.T. Architectural styles and the design of network-based software architectures // Dotoral dissertation, University of California, Irvine, 2000
2. Грид для Национальной нанотехнологической сети. URL: <http://ngrid.ru/ngrid>, 2008.
3. Демичев А., Ильин В., Крюков А. и Шамардин Л. Реализация программного интерфейса грид-сервиса Pilot на основе архитектурного стиля REST.// Вычисленные методы и программирование, с.62-65, т.11, 2010.
4. Демичев А., Крюков А. и Шамардин Л. Принципы построения грид с использованием Restful-веб-сервисов.// Программные продукты и системы, №4, 2009, г.Тверь
5. GLUE Scheme. URL: <http://forge.ogf.org/sf/projects/glue-wg>, 2007.
6. Zyp K. A JSON Media Type for Describing the Structure and Meaning of JSON Documents. // Techniocal report, IETF Network Workong Group, draft-zyp-json-schema-02, March 2010.