

Московский государственный университет им. М.В. Ломоносова
Факультет вычислительной математики и кибернетики

Курсовая работа

«Разработка модуля в среде Друпал для автоматизации процесса постановки задач на суперкомпьютерах»

**«Module development in the Drupal environment for automation of the
running programs on supercomputer process »**

Студента 3 курса кафедры АСВК
факультета ВМК МГУ
Осадчука Дмитрия Руслановича

Научный руководитель:
к.ф.-м.н. **Сальников Алексей Николаевич**

Москва
2016-2017 гг.

Содержание

1 Введение	3
2 Постановка задачи	4
3 Описание предметной области.....	5
4 Архитектура полной системы	5
5 Архитектура разрабатываемой системы	7
6 Структура системы управления содержимым Drupal	8
7 Специфика создания модулей для системы Drupal	9
8 Сведение задачи к реализации в окружении CMS Drupal	10
9 Характерные моменты при разработке	13
10 Заключение	14
11 Список литературы	15

1 Введение

Стремительное развитие и расширение сфер применения IT-технологий обуславливается современностью. Количество суперкомпьютеров в разных уголках планеты растёт с каждым днём и с течением времени их работа затрагивает всё большую часть человеческой деятельности. Ещё одним немаловажным фактором является использование Интернета в повседневной жизни, так как всемирная сеть и взаимодействие с ней значительно упрощают нашу жизнь.

Не так давно суперкомпьютеры казались чем-то удивительным и непостижимым, однако сегодня практически любой человек может поработать с таковым удалённо, получив разрешение и необходимую для этого информацию от администрации вычислительных комплексов.

Зачастую, для доступа и работы на суперкомпьютере необходимо проделать следующие шаги:

- Обсудить план работ с системным администратором суперкомпьютера для подтверждения адекватности запланированной деятельности на вычислительном комплексе;
- Системный администратор запрашивает разрешение у администрации вычислительного комплекса на предоставление доступа к суперкомпьютеру стороннему лицу;
- При получении положительного ответа от администрации вычислительного комплекса, системный администратор создаёт новую учётную запись, под которой человек будет работать на суперкомпьютере и передаёт её данные заинтересованному лицу;
- Прочитав необходимую документацию, человек может взаимодействовать с суперкомпьютером через заранее определённые интерфейсы под выданной ему учётной записью в рамках тех прав доступа, которые были ей заданы при создании;

На первый взгляд может показаться, что процесс довольно прост, однако временные рамки от начала первого шага до конца последнего могут превышать несколько месяцев, а если возникла необходимость в вычислительных мощностях суперкомпьютера не у одной персоны, а у группы людей, то весь организационный процесс может затянуться на очень длительный срок. Зачастую, такое количество времени просто отсутствует у групп людей, связанных с научной деятельностью.

Чтобы упростить процесс предоставления доступа к вычислительным комплексам без потери надёжности и отказоустойчивости необходимо сократить наименее важные пункты из вышеописанных и отмасштабировать процесс для групп заинтересованных лиц.

Что касается «лишних» пунктов, со стороны администрации вычислительных комплексов, возможно убрать последний (ознакомление с документацией и обучающими материалами по работе на суперкомпьютерах), предоставив такой интуитивно понятный User-friendly пользовательский интерфейс, где пользователь не сможет повлиять на работу суперкомпьютера случайно и, тем более, намеренно.

Для исследовательских групп, нуждающихся в вычислительных мощностях суперкомпьютера, обобщим процесс получения доступа к суперкомпьютерам для одного человека – руководителя группы. Руководитель группы является ответственным лицом за всю деятельность его группы на суперкомпьютерах и учётные записи системы будут

принадлежать ему, что позволит идентифицировать и разбивать пользователей по группам на уровне архитектуры системы. Такое положение вещей позволяет убрать необходимость в прохождении всего процесса каждым членом исследовательской группы и оставляет это обязательство лишь для одного человека – руководителя группы.

2 Постановка задачи

Из пункта выше становится понятно, что возникает необходимость в системе, которая позволит упростить процесс получения доступа к вычислительным комплексам для заинтересованных лиц и исследовательских групп, предоставив надёжный и удобный интерфейс для работы на суперкомпьютере и сделав возможным едино разовое получение доступа к суперкомпьютеру для групп людей, объединённых единой научной задачей.

Рассматривались различные варианты реализации подобной системы:

- Создание приложения для ПК;
- Организация специально отведённых физических терминалов, где пользователи могли бы приходить и работать на суперкомпьютерах;
- Создание единого сайта, где пользователи бы работали через веб-интерфейс.

Проанализировав разные варианты были выявлены преимущества и недостатки каждого из них.

Способ реализации системы через приложение для ПК обладает качеством довольно простой реализации, однако отслеживание некорректной деятельности на суперкомпьютере является чрезмерно сложным, или невозможным механизмом, так же, как и разделение пользователей на научные группы на уровне архитектуры самой системы.

Вариант разработки системы как отдельных физических терминалов для работы на суперкомпьютере является довольно надёжной реализацией описанной системы, однако к недостаткам этой версии можно отнести излишние экономические затраты и невозможность работать на суперкомпьютере удалённо, что противоречит требованиям к разрабатываемой системе.

Разновидность реализации данной системы через единый веб-интерфейс обладает немаловажными качествами, а именно: отсутствие необходимости в дополнительных экономических вложениях, достаточно простой процесс разработки и соответствие всем необходимым требованиям к системе, кроме одного – возможность разделения пользователей на научные группы на уровне архитектуры системы.

Сравнив вышеописанные варианты между собой, было принято решение реализовывать данную систему через веб-интерфейс (последний пункт), однако была внесена некая поправка в предложенную реализацию, чтобы исключить её единственный недостаток. Модификация реализации заключается в том, чтобы разрабатываемый веб-интерфейс был единым не для всех, а только для пользователей отдельно взятой группы. То есть, одной научной группе, получившей доступ к вычислительному комплексу (доступ получает руководитель группы единожды для всей научной группы) соответствует один веб-интерфейс, администратором которого является руководитель группы, получивший доступ к вычислительному комплексу. Множество таких веб-интерфейсов образуют единую систему, которая замыкается единым хранилищем данных (базой данных) операций, файлов, учётных записей системы и задач, расположенной отдельно от узлов с веб-интерфейсами.

Таким образом, необходимо создать веб-интерфейс (сайт), который свободно и легко дублируется (устанавливается), взаимодействует с общим хранилищем данных и

предоставляет членам группы удобный и надёжный интерфейс для работы на суперкомпьютере. При этом руководитель группы является администратором данного ресурса и ему должна предоставляться возможность модифицировать состав группы (добавлять новых членов научной группы). Интерфейс ресурса должен позволять пользователям создавать свои задачи для выполнения на суперкомпьютерах, а так же осуществлять работу с файлами, которые обрабатываются программами в соответствии с поставленными задачами и получаются на выходе при завершении работы программ.

3 Описание предметной области

В соответствии с поставленной задачей, разработка веб-интерфейса с вышеописанным функционалом и соответствием требований к нему с чистого листа привела бы излишним трудозатратам. Учитывая этот фактор было принято решение использовать систему управления контентом (CMS) Drupal. В данной системе управления контентом уже реализован базовый функционал, а именно: регистрация пользователей на сайте, структурированное управление контентом, блоки веб-интерфейса довольно просто модифицируются под нужды администратора и, что немаловажно, ядро (базовая составляющая данной системы) масштабируется. То есть, предусмотрено расширение функционала путём создания и подключения дополнительных модулей, которые выполняют необходимые администратору функции (более подробно данный механизм будет рассмотрен ниже). Так же к преимуществам данной системы управления контентом можно отнести тот факт, что данная система является системой с открытым исходным кодом. Это свидетельствует о том, что система непрерывно обновляется и совершенствуется сообществом людей, которые её используют, а так же о том, что она бесплатна для использования в любых целях.

Многие современные сайты созданы с помощью CMS (Content Management System) – систем управления контентом. Как правило, они используют модульную архитектуру, которая позволяет без труда добавлять новые функции. Таким образом, поставленную задачу можно выполнить с помощью написания необходимого модуля, который дополнит базовый функционал данной системы.

Объектом исследования данной работы является процесс разработки модуля, который позволит пользователям веб-интерфейса создавать и ставить задачи на выполнение на суперкомпьютерах, а так же поддерживает разграничение прав доступа (руководитель группы, член группы и т.д.) и позволяет администратору ресурса управлять составом научной группы. Для разработки подобного модуля потребуется изучить материал связанный с надёжностью аутентификации пользователей в системах, провести предварительный анализ разных вариантов установления соответствия между учётными записями (учётная запись веб-ресурса и учётная запись разрабатываемой системы), изучить методику написания дополнительных модулей для данной CMS, изучить материалы по специфике постановки задач на суперкомпьютерах.

Предметом исследования является использование возможностей CMS Drupal, а также языка HTML, PHP, и язык структурированных запросов MySQL для разработки данного модуля.

В данной работе проведен анализ предметной области и теоретического материала, разработан интерфейс и программный код модуля.

Теоретическая значимость работы заключается в том, что сформированное описание можно использовать в дальнейшем для решения схожих задач.

Практическая значимость работы заключается в возможности использовать разработанный модуль в практической сфере деятельности администратора сайта, который является руководителем научной группы людей, нуждающихся в вычислительных мощностях вычислительных комплексов.

4 Архитектура полной системы

Учитывая специфику поставленной задачи, общая архитектура разрабатываемой системы содержит в себе несколько типов сущностей, а именно: веб-интерфейс (то есть «сайт»), хранилище данных (база данных), отдельная программная система, которая обрабатывает внесённые изменения в хранилище данных через веб-интерфейс и вычислительные комплексы. Необходимо помнить, что каждой научно-исследовательской группе людей соответствует один веб-интерфейс. Это говорит о том, что таких веб-интерфейсов может быть неограниченное количество.

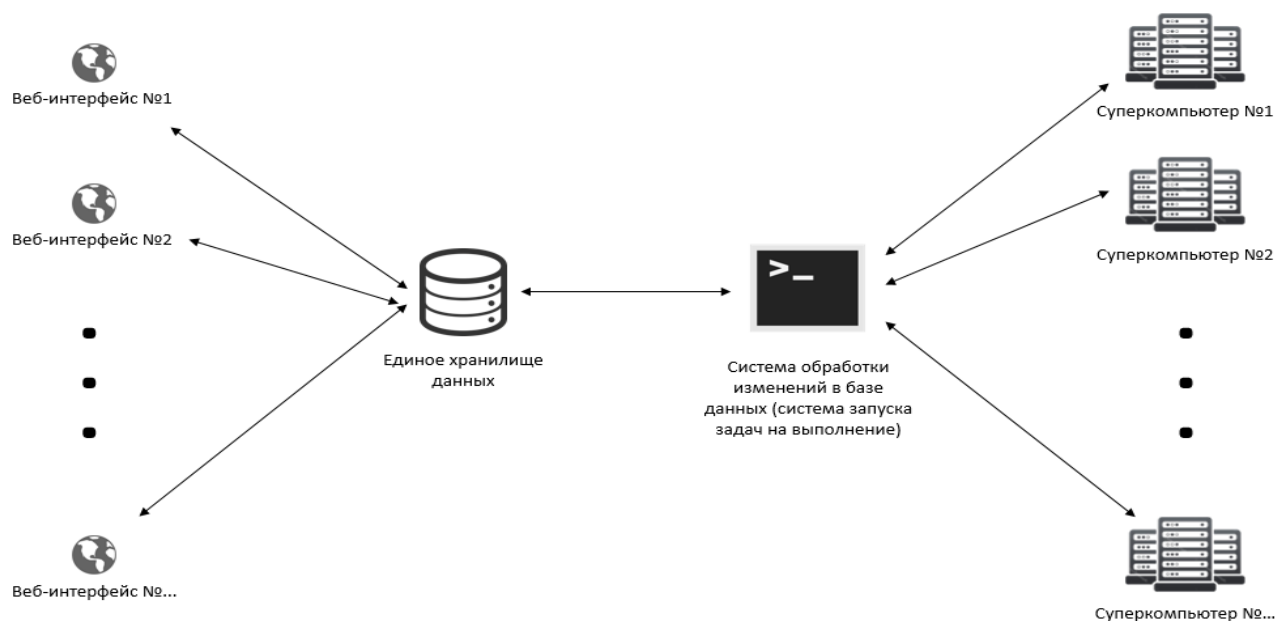


Рисунок 1. Общая архитектура системы в целом

Взаимодействие упомянутых выше сущностей между собой можно описать в несколько этапов:

1. Пользователи взаимодействуют с определённым веб-интерфейсом (загружают файлы, создают новые задачи, убирают уже созданные задачи из очереди и т.д.);
2. Веб-интерфейс заносит новые данные и данные об изменениях в единое хранилище данных;
3. Система запуска задач на выполнение проводит полный жизненный цикл выполнения необходимой задачи от запуска программы на суперкомпьютере с указанными параметрами до завершения работы программы и сохранения результата выполнения в единое хранилище данных;
4. После завершения пункта под номером 3 в веб-интерфейсе отображаются результаты выполнения соответствующей программы, которые загружаются из хранилища данных.

Таким образом, архитектура системы обладает качеством точного распределения обязанностей между сущностями, что определяет конкретные требования к каждой сущности при разработке. Так же, разграничение сущностей и функционала системы в целом между сущностями придаёт всей системе модульную архитектуру, которая в свою

очередь обладает высокой степенью отказоустойчивости, так как при выходе из строя одного модуля вся система продолжает работать в штатном режиме.

5 Архитектура разрабатываемой системы

Данная научная работа посвящена разработке только части всей описанной выше системы, а именно созданию легко переносимой и дублируемой версии веб-интерфейса на базе системы управления контентом Drupal 7, который удовлетворяет требованиям, указанным в пункте «Постановка задачи».

Учитывая данную задачу, рассмотрим архитектуру конкретно разрабатываемой системы в увеличенном масштабе.

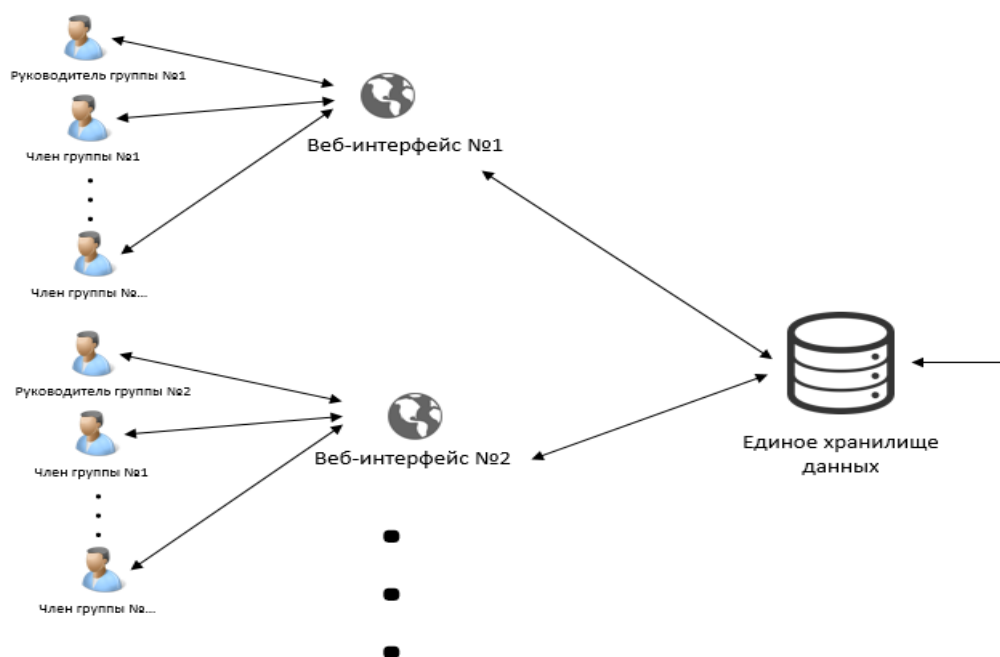


Рисунок 2. Архитектура разрабатываемой системы

Как видно на рисунке 2 и как описывалось ранее, каждый веб-интерфейс соответствует одной научно-исследовательской группе людей и администратором веб-интерфейса является руководитель группы. Такая архитектура позволила перенести ответственность за деятельность разных людей на суперкомпьютерах на руководителя группы, что в свою очередь позволило сократить временные рамки организационного процесса получения доступа к вычислительным комплексам группой людей и избавиться от временных затрат на освоение специфики работы на суперкомпьютерах через терминал, как это было ранее.

6 Структура системы управления содержимым Drupal

В пункте «Описание предметной области» уже оговаривалось, что поставленная задача будет реализовываться на базе популярной системы управления содержимым (CMS) Drupal, а так же были упомянуты преимущества данной системы.

Немаловажно понимать структуру выбранной CMS, чтобы усвоить специфику её работы и заблаговременно предотвратить возможные сложности при разработке модуля для системы Drupal.

Для работы CMS Drupal, как и для любой другой системы управления содержимым, необходимы три «жита»: PHP, MySQL сервер и Apache сервер. Эти компоненты обеспечивают платформу для установки CMS Drupal на низшем уровне и предоставляют необходимое API данной системе для взаимодействия с окружающей аппаратной и программной средой.

Рассматривая более высокий абстрактный уровень необходимо помнить, что данный уровень состоит из Ядра Drupal, модулей ядра Drupal, сторонних модулей и системы тем оформления. Ядро Drupal обеспечивает корректное взаимодействие и является «прослойкой» между модулями ядра Drupal. Модули ядра Drupal – это модули, которые устанавливаются во время установки CMS Drupal. Они предоставляют базовый функционал для пользователей устанавливаемого веб-интерфейса. Что же касается сторонних модулей, это модули, от которых не зависит базовый функционал CMS Drupal, то есть они лишь дополняют уже имеющиеся возможности до необходимых, но не являются обязательными. И, наконец, система тем оформления осуществляет темизацию всего веб-интерфейса, то есть меняет внешний вид веб-страниц не затрагивая реализованный функционал.

Последний уровень – наивысший, это уровень структуры, в котором задаются дополнительные настройки внешнего вида веб-страниц путём задания новых каскадных таблиц стилей (CSS), добавления новых HTML-блоков, не относящихся к Drupal и JavaScript-сценариев, придающих веб-интерфейсу больше интерактивности и динамики.

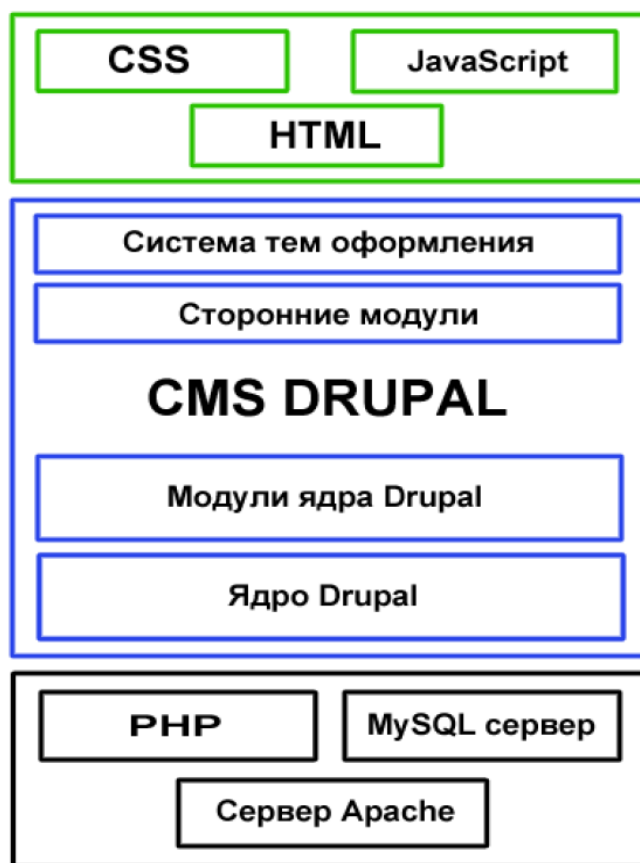


Рисунок 3. Структура CMS Drupal

Нетрудно догадаться, что разрабатываемый модуль относится к категории сторонних модулей, которые дополняют основной функционал веб-сайта. Для планирования и начала разработки модуля необходимо ознакомиться со спецификой написания модулей для данной CMS.

7 Специфика создания модулей для системы Drupal

Для написания сторонних модулей в системе управления содержимым Drupal предусмотрен специальный механизм, который позволяет структурировано взаимодействовать с данной платформой в программном коде разрабатываемого модуля. Этот механизм состоит из таких компонентов, как хуки (hooks), API для работы с базами данных (Database API), API для создания форм на веб-страницах (Form API), глобальных переменных (Drupal Globals) и других, однако для разработки описанного модуля достаточно перечисленных.

Хуки (hooks) – это события в системе Drupal, которым можно назначить php-функции, которые в свою очередь, имеют специфическое название (название функции, обычно, содержит уникальное имя хука). Эти функции вызываются по происшествию событий, которые заранее определены в системе Drupal. Например, php-функция «name_form_submit» будет вызвана при отправки формы пользователем, которая называется «name». Таким образом достаточно просто можно создавать функции-обработчики для необходимых событий.

Database API – это специально предоставляемые функции системой Drupal для защищённой и удобной работы с базами данных. Данное API сокращает количество кода для решения ряда определённых задач и придаёт коду понятную для восприятия структуру. Например, для выборки данных из базы данных предоставляется функция «db_select».

Form API – это уже готовый функционал, предоставляемый системой Drupal для структурированной и удобной работы с формами. Данное API, как и его предшественное, придаёт коду структурированный и читабельный вид, и сокращает количество необходимого кода для реализации поставленных задач. Например, чтобы создать форму, предоставляется функция «name_form», в которой просто необходимо описать ассоциативный php-массив с определёнными параметрами.

Глобальные переменные (Globals) – это переменные, которые доступны в любом участке кода модуля, и которые хранят в себе определённую информацию. Они упрощают работу программиста, придают коду понятность и простоту. Например, чтобы узнать уникальный идентификатор текущего пользователя Drupal, необходимо написать «global \$user; \$user -> uid;».

Благодаря подобным API, которые предоставляет Drupal, чётко прослеживается структура модуля и его взаимодействия с системой в целом, снижается вероятность того, что какие-то блоки в модуле будут реализованы принципиально неверно и сокращается время разработки самого модуля.

8 Сведение задачи к реализации в окружении CMS Drupal

При реализации поставленной задачи по написанию модуля для CMS Drupal возникла необходимость в том, чтобы определиться, какие хуки необходимо использовать в коде данного модуля, ведь таковых свыше 460 штук. Чтобы разрешить эту задачу, был разработан некий сценарий использования данного модуля администратором и пользователями веб-интерфейса.

После установки данного модуля администратором веб-интерфейса (руководителем группы, который получил положительный ответ на запрос доступа к суперкомпьютерам и получивший уникальный ключ), администратор в конфигурациях модуля вводит выданный ему уникальный ключ. Это уникальный ключ сущности Drupal. Нетрудно догадаться, что каждой единице веб-интерфейса выдаётся некий уникальный ключ, по которому данный веб-интерфейс авторизуется в общей системе и может начинать свою

работу. После ввода ключа, администратору веб-ресурса необходимо ввести адрес электронной почты. На этот адрес будут высланы данные учётной записи руководителя группы. Далее под этой учётной записью руководитель группы может войти в систему на странице авторизации, которая создаётся при установке модуля, и ему будут доступна страница создания новой задачи, страница работы с файлами, страница просмотра истории поставленных задач, и, отличительный момент руководителя группы от обычного члена группы, будет доступна страница добавления новых членов научной группы, на которой будет необходимо ввести адрес электронной почты члена группы, и на указанный адрес придёт письмо с данными о новой учётной записи в системе.

Сценарий был продуман в общих чертах, однако этого хватило для того, чтобы определить последовательность реализуемых частей модуля.

Первое, что было реализовано – установление соответствия учётных записей веб-интерфейса и учётных записей всей системы. «Привязка» выполняется в отношении «один-к-одному» по уникальному ключу учётной записи всей системы и уникальному идентификатору пользователя веб-интерфейса. При этом, каждая операция в системе требует правильности уникального ключа пользователя и уникального ключа сущности Drupal. В противном случае, будет выведено сообщение об ошибке, либо запрашиваемые страницы будут вовсе недоступны. Для выполнения этого пункта необходимо было создать страницу авторизации, страницу самого модуля (был использован `hook_menu`), добавить в их содержимое формы авторизации и выхода(отвязки) из учётной записи системы (использовалось Form API) и работа с разными базами данных (единое хранилище данных всей системы и база данных Drupal), для которой использовалось Database API.

Home

Главная

Поиск

Навигация

- ▼ UNICLUST
 - Личный кабинет
 - mtrn
 - Добавить материал

UNICLUST

Войдите в систему UNICLUST

Если вы уже входили, то введите данные, которые использовали в последний раз. Иначе введите данные свободной учётной записи UNICLUST (обычно выдаётся руководителем группы) для её привязки к Вашему аккаунту на сайте.

Логин или Email *

Пароль *

Войти

Рисунок 4. Страница авторизации пользователей системы

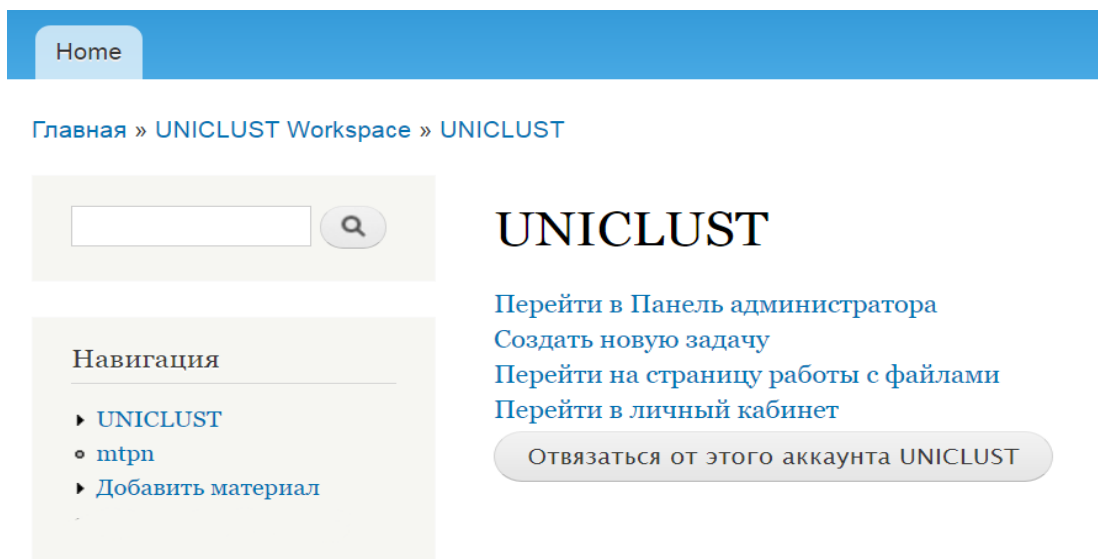


Рисунок 5. Главная страница системы для руководителя группы (корень ссылок)

После реализации всех моментов, связанных с авторизацией членов группы в системе, возникла необходимость в создании панели администратора, а именно панели для добавления новых членов группы руководителем группы, если текущий авторизованный пользователь им является. Для каждой созданной страницы использовался «hook_menu», поэтому в дальнейших пунктах будет подразумеваться, что он был использован без упоминания об этом в тексте. Помимо этого, вновь использовался уже знакомый «hook_form» для создания формы регистрации нового члена научной группы и внесения этой формы в содержимое страницы. Так же использовалось Drupal Database API для работы с базой данных всей системы и базой данных текущей сущности Drupal.

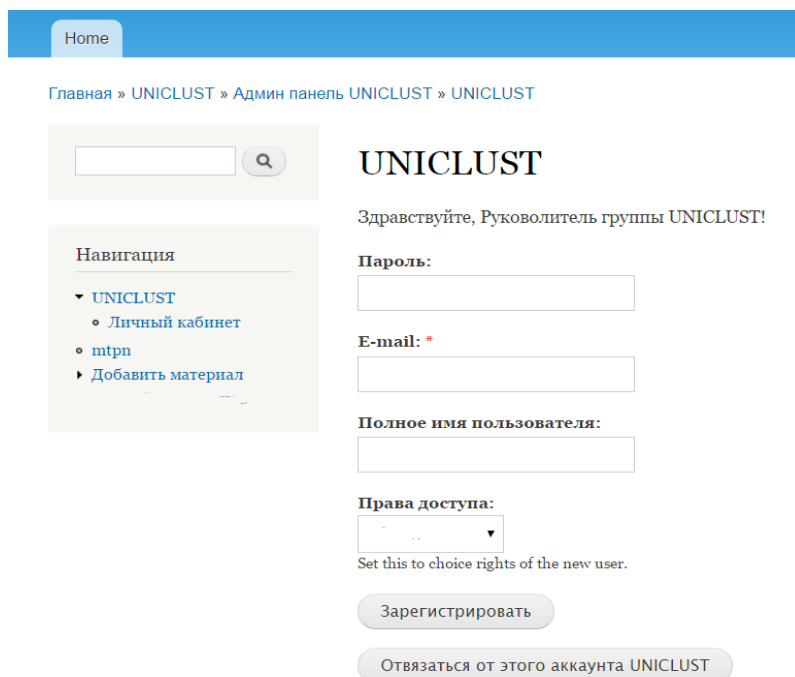


Рисунок 6. Панель администратора (страница регистрации новых членов группы)

В соответствии с общим сценарием, после регистрации учётных записей для новых членов группы, система должна предоставлять возможность работать с файлами. Файлы

можно загружать в систему для дальнейшей их обработки программами, которые указываются в создаваемых задачах членами научной группы. Для этого была создана отдельная веб-страница, на которой можно загружать файлы и просматривать список загруженных ранее файлов. Как и в предыдущих пунктах, для этого были использованы «hook_menu», «hook_form», «hook_form_submit», Database и Form API, которые предоставляются CMS Drupal.

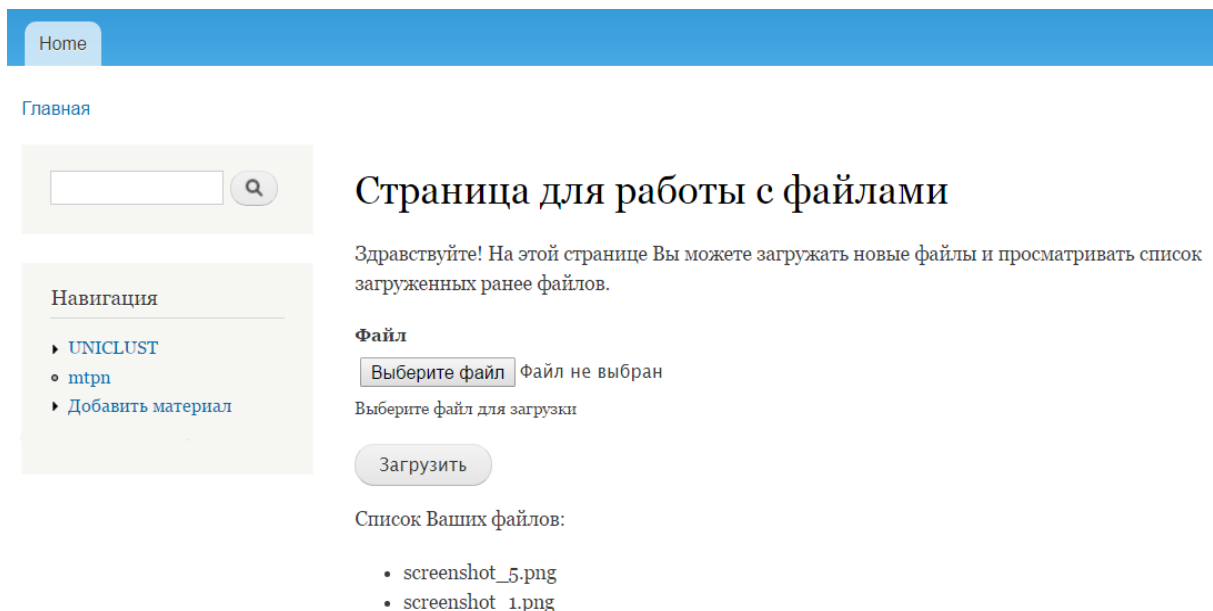


Рисунок 7. Страница для загрузки новых файлов и просмотра списка загруженных ранее файлов

Последний функционал, который необходимо было реализовать и предоставить пользователю – создание новых задач для выполнения на суперкомпьютере. Данный пункт довольно сложен в реализации. Необходимо запрограммировать декодер JSON-файлов, которые описывают разные классы задач, которые доступны для запуска на разных суперкомпьютерах. В этих файлах так же описывается множество всех возможных аргументов и зависимостей между ними, которые необходимо указывать при создании задачи. Далее, с помощью Drupal Form API, необходимо сформировать форму для веб-интерфейса, которая генерируется автоматически при декодировании JSON-файла соответствующего класса задач. После заполнения и отправки формы пользователем, задача состоит в том, чтобы сгенерировать JSON-файл с введенными пользователем аргументами и загрузить данные в единое хранилище данных системы, после чего другая часть системы считывает этот файл и ставит в очередь задачи с соответствующими аргументами на выполнение на суперкомпьютере. По завершению выполнения данной программы, система вносит данные, которые получились «на выходе», в хранилище данных и результаты отображаются в веб-интерфейсе.

Home

Главная

Навигация

- UNICLUST
- Добавить материал

Страница создания новой задачи

Здравствуйте! На этой странице Вы можете создать новую задачу для выполнения на суперкомпьютере!

Название задачи: *

Поворот трёхдиагональной матрицы

Выберите суперкомпьютер: *

IBM pSeries 690 HPC Regatta

Выберите класс программы из списка для выполнения на суперкомпьютере: *

Matrix Processing

Действие: *

Rotate

Угол: *

180

Выберите файл для обработки: *

matrix_16095x16095.txt

Создать задачу

Рисунок 8. Интерфейс для создания новой задачи

Справедливо уточнить, что данный веб-интерфейс можно дополнить большим количеством дополнительных страниц, элементов и виджетов, которые улучшат характеристики эргономики, динамики и наглядности данных, однако это не являлось целью данной научной работы. Реализованы характерные части системы, которые являются обязательными составляющими и предоставляют базовый функционал, который можно масштабировать в разных направлениях и вариациях.

9 Характерные моменты при разработке

В ходе разработки вышеописанного веб-интерфейса возникали нетривиальные технические ситуации, о которых важно упомянуть.

Важным моментом при разработке оказался вопрос надёжности генерации уникальных ключей. Изначально планировалось, что уникальные ключи будут генерироваться в разрабатываемом модуле и затем будут отправляться в хранилище данных. Однако, если рассматривать важный аспект надёжности и безопасности системы, то такое решение привело бы к потенциально возможным вредоносным инъекциям в систему. По этой причине было принято решение перенести ответственность генерации уникальных ключей с разрабатываемого модуля на хранилище данных. То есть, генерация ключей происходит не в модулях отдельно-взятых сущностей Drupal (веб-интерфейсов), а в едином месте – базе данных системы, после чего эти ключи могут быть использованы модулями множества веб-интерфейсов.

Аналогичная проблема возникла и при создании новых пользователей системы руководителями научных групп. Как было сказано выше, руководители групп могут добавлять пользователей в систему, которые будут причислены к научной группе руководителя. При генерации информации о новых учётных записях (включая пароли, ключи и т.д.) во множестве веб-интерфейсов, возникает вероятность того, что данные могут быть перехвачены на сервере, где расположен веб-интерфейс, при отправке данных в хранилище данных системы, либо генерация излишнего количества информации при изменении кодов модуля, что оказывает сильно негативное влияние на всю систему. В соответствии с этим, как и в предыдущем пункте, потенциальная проблема было решена

путём переноса ответственности за генерацию информации о новых пользователях с сущности веб-интерфейса Drupal на хранилище данных. То есть, данные, которые должны быть надёжными и точными генерируются в едином месте, к которому полноценный доступ имеет лишь администратор системы в целом.

Так же, необходимо упомянуть о структуре пользовательской авторизации, ведь именно с её помощью система может идентифицировать научные группы и даже отдельных пользователей. Структура аутентификации устроена таким образом, что одному пользователю веб-интерфейса соответствует только один пользователь разрабатываемой системы. Когда пользователь веб-интерфейса входит в учётную запись системы, происходит «привязывание» этих учётных записей друг ко другу по уникальному ключу, который генерируется при создании учётной записи системы. Таким образом система может однозначно определять пользователя по конкретному уникальному ключу. Аналогично устроена идентификация научных групп. Каждой новой сущности Drupal выдаётся некий уникальный ключ, по которому руководить группы может подключить новую сущность Drupal ко всей системе. Если ключ является неверным (т.е. не существует такого ключа в хранилище системы, либо данный ключ уже занят), то данный веб-ресурс уведомит об этом руководителя группы ошибкой и невозможностью работать с системой в целом.

10 Заключение

Без использования разработанного веб-интерфейса на платформе управления содержимым Drupal работа с суперкомпьютерами осуществляется через командную строку, что является неудобным и небезопасным методом взаимодействия, а так же сам процесс получения доступа к суперкомпьютерам может требовать длительный период времени.

Данная научная работа является решением проблемы, которая возникает перед научно-исследовательскими группами людей, нуждающихся в вычислительных мощностях суперкомпьютеров. Разработанный веб-интерфейс значительно упрощает процесс получения доступа к суперкомпьютерам группой заинтересованных лиц, а так же вносит простоту и понятность во взаимодействие отдельных членов группы с суперкомпьютерами.

Так же, данная научная работа несёт как теоретическую, так и практическую ценность, так как в ней освещена структура системы управления содержимым Drupal, описано устройство взаимодействия разных уровней компонентов данной CMS и рассмотренные практические техники можно использовать для решения схожих практических задач.

11 Список литературы

1. Официальная документация Drupal 7 - [Электронный ресурс]. – URL: <https://www.drupal.org/docs/7>.
2. Форум сообщества разработчиков и пользователей Drupal - [Электронный ресурс]. – URL: <http://drupal.ru/forum/>.
3. Онлайн-сообщество для программистов, поиска решений и ответов на поставленные задачи - [Электронный ресурс]. – URL: <https://stackoverflow.com/>.
4. Официальная документация языка и конструкций языка PHP - [Электронный ресурс]. – URL: <https://php.net/>.
5. Форум программистов и системных администраторов Cyberforum - [Электронный ресурс]. – URL: <http://www.cyberforum.ru/php/>.

6. Чередниченко И. А., Выпускная квалификационная работа, «Разработка модуля отложенного постинга, как анализ способа автоматической публикации контента в социальные сети из CMS Drupal» - [Электронный ресурс]. – URL: https://sibsutis.ru/upload/781/diploma_work.pdf/.
7. Видеокурсы по установке и настройке CMS Drupal - [Электронный ресурс]. – URL: <http://podrupalim.com/>.
8. Блог по настройке и работе с CMS Drupal - [Электронный ресурс]. – URL: <https://drupalfly.ru/>.
9. Подборка статей по созданию сторонних модулей для CMS Drupal - [Электронный ресурс]. – URL: <https://habrahabr.ru/post/200340/>.
10. Портал по изучению структуры и специфике работы CMS Drupal - [Электронный ресурс]. – URL: <http://drupalguide.ru/articles/>.
11. Статья, посвящённая правильному подходу к шифрованию уникальных ключей (ХЭШей) - [Электронный ресурс]. – URL: <https://habrahabr.ru/post/210760/>.
12. Официальная документация по Drupal Database API - [Электронный ресурс]. – URL: <https://www.drupal.org/docs/7/api/database-api/database-api-overview>.
13. Официальная документация по Drupal Form API - [Электронный ресурс]. – URL: <https://www.drupal.org/docs/7/api/form-api>.