МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
федеральное государственное бюджетное образовательное учреждение высшего образования
КРАСНОЯРСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ
им.В.П.АСТАФЬЕВА
(КГПУ им.В.П.Астафьева)

Институт/факультет Институт математики, физики и информатики
(полное наименование института/факультета/филиала)

Выпускающая кафедра Базовая кафедра информатики и информационных технологий в образовании
(полное наименование кафедры)

Денисов Евгений Сергеевич
Иванов Анатолий Юрьевич

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Тема Способы организации временного вычислительного кластера на основе PelicanHPC в компьютерном классе

Направление подготовки 44.03.05 Педагогическое образование
(код и наименование направления)

Профиль Физика и информатика
(наименование профиля для бакалавриата)

ДОПУСКАЮ К ЗАЩИТЕ
Заведующий кафедрой
д.п.н., профессор Пак Н.И.
(ученая степень, ученое звание, фамилия, инициалы)

(дата, подпись)

Руководитель к.ф.-м.н., доцент, Шикунов С.А.
(ученая степень, ученое звание, фамилия, инициалы)

Дата защиты

Обучающиеся Денисов Е.С., Иванов А.Ю.
(фамилия, инициалы)

(дата, подпись)

Оценка
(прописью)

Красноярск 2017

Содержание

Введение

Развитие компьютерной техники на современном этапе привело к обеспечению повышения производительности путём увеличения количества ядер в процессорах и количества используемых процессоров. Потребность в написании программ, использующих несколько процессорных ядер и/или процессоров, впервые возникла в области суперкомпьютерных технологий. Современный же этап развития компьютерной техники сделал эти технологии востребованными практически во всех современных цифровых устройствах. Но, несмотря на это, суперкомпьютерные технологии, и в первую очередь параллельное программирование, продолжают интенсивно развиваться и использоваться для ускорения решения самых различных задач.

Ввиду вышесказанного изучение параллельного программирования и других суперкомпьютерных технологий, как в вузах, так и в школе весьма актуально.

Проблемой при этом является, как правило, отсутствие доступа к реальным суперкомпьютерам в учебных заведениях. Решением этой проблемы может быть построение вычислительного кластера на основе локальной сети учебного класса.

Целью данной работы является разработка методических рекомендаций по быстрому развертыванию и использованию в учебном процессе вычислительного кластера на основе школьного класса посредством пакета PelicanHPC.

Объектом исследования является обучение основам суперкомпьютерных технологий.

Предметом исследования является методика использования пакета PelicanHPC для учебных целей

Для достижения поставленной цели, были определены следующие задачи:

- проанализировать системы автоматического развёртывания вычислительных кластеров
- изучить характеристики, возможности и особенности установки кластера при помощи PelicanHPC
- разработать методику проведения занятий на вычислительном кластере на основе PelicanHPC
- наметить пути модификации пакета PelicanHPC для использования в учебном классе, и провести модификацию, получив новый пакет PelicanHPC-EDU
- выявить особенности проведения занятий на вычислительном кластере на основе PelicanHPC-EDU

Глава 1. Системы автоматического развёртывания вычислительных кластеров

1.1. Вычислительный кластер

Существуют три основных вида кластеров: Fail-over (отказоустойчивые), Load-balancing (балансировочные, распределяющие нагрузку) и High Performance Computing (высокопроизводительные вычислительные).

Отказоустойчивые кластеры представляют собой два или более связанных по сети компьютера с дополнительным выделенным контрольным (heartbeat) соединением между ними. Это выделенное соединение между машинами используется для мониторинга состояния сервисов: как только заданный сервис на одной машине выходит из строя, то другая начинает выполнять её функции.

Концепция балансировочных кластеров заключается в том, на что пришедший запрос к веб-серверу кластер реагирует следующим образом: кластер сначала определяет наименее загруженную машину, а затем направляет к ней запрос. Довольно часто балансировочный кластер выполняет и функции отказоустойчивого кластера, хотя для этого и требуется большее количество узлов.

Вычислительный кластер можно определить как множество рабочих компьютеров (узлов), связанных коммуникационной средой и способных работать как единое целое за счет дополнительного программного обеспечения [21].

Вычислительный кластер используется специально для центров обработки информации, которым необходима максимально возможная производительность. В таких системах также присутствует некоторая функция балансировочных кластеров: для повышения производительности они распределяют процессы на большее количество машин. При такой конфигурации процесс распараллеливается на части, которые выполняются

на многих машинах одновременно, вместо того чтобы выполняться друг за другом по очереди [10].

Существует несколько технологий реализации параллельных вычислений: (N)UMA, DSM, PVM и MPI – всё это различные схемы параллельных вычислений. Некоторые из них уже реализованы аппаратно, другие только в программном, а некоторые – и в том, и в другом виде.

(N)UMA: здесь машины пользуются разделяемым доступом к памяти, в которой они могут выполнять свои программы. В ядре Linux реализована поддержка NUMA, позволяющая получать доступ к разным областям памяти. При этом задача ядра использовать ту память, которая находится ближе к процессору, работающему с ней.

DSM уже реализована не только в программном виде, но и в аппаратном. Основная концепция DSM в организации абстрактного слоя для физически распределённой памяти.

Технологии PVM и MPI наиболее часто упоминаются, когда речь заходит о GNU/Linux Beowulf кластерах.

MPI – это открытая спецификация библиотеки передачи сообщений. Самой популярной реализацией MPI является MPICH. Второй по популярности после MPICH можно назвать LAM, также являющейся свободной реализацией MPI [14].

MPI является наиболее удобной и легко осуществимой процедурой в реализации вычислений параллельных программ. Программу посредством технологии MPI может запустить любой пользователь [1].

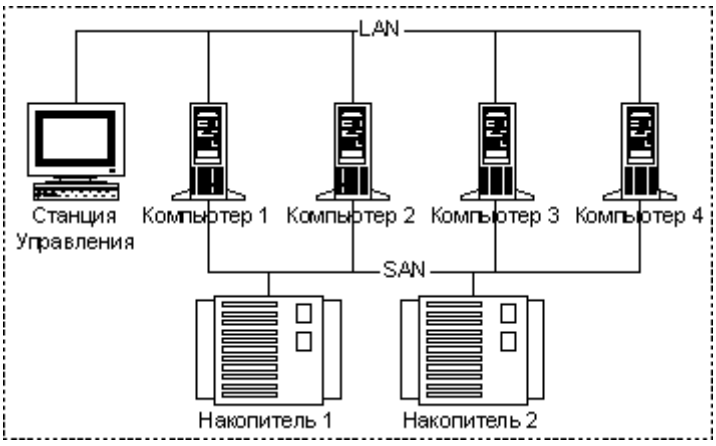


Рисунок 1 - Кластерная система

Кластер функционирует как единая система (Рис. 1), то есть для пользователя или прикладной задачи вся совокупность вычислительной техники выглядит как один компьютер. Именно это и является самым важным при построении кластерной системы.

Первые кластеры компании Digital были построены на машинах VAX. Эти машины уже не производятся, но все еще работают на площадках, где были установлены много лет назад. Эти общие принципы, заложенные при проектировании кластеров, остаются основой при построении таких систем и сегодня [8].

К общим требованиям, предъявляемым к кластерным системам, относятся:

1) Высокая готовность
2) Высокое быстродействие
3) Масштабирование
4) Общий доступ к ресурсам
5) Удобство обслуживания

Общие принципы построения кластерной архитектуры.

Как уже было сказано раньше вычислительный кластер — это совокупность компьютеров, объединенных в рамках некоторой сети для решения одной задачи, которая для пользователя представляется в качестве

единого ресурса. Такую конфигурацию кластера впервые предложила и реализовала в начале 80-х корпорация Digital Equipment, которая и сегодня развивает эту технологию.

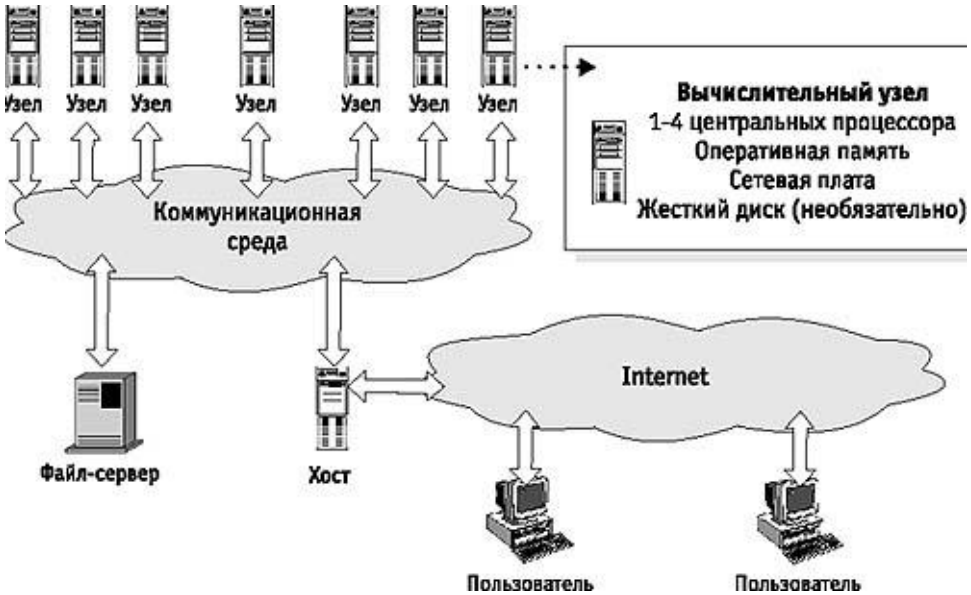


Рисунок 2 – Вычислительный кластер

Понятие «единый ресурс» означает наличие программного обеспечения, дающего возможность пользователям, администраторам и прикладным программам считать, что имеется только одна сущность, с которой они работают - кластер. Например, система пакетной обработки кластера позволяет послать задание на обработку кластеру, а не какому-нибудь отдельному компьютеру. Более сложным примером являются системы баз данных. Практически у всех производителей систем баз данных имеются версии, работающие в параллельном режиме на нескольких машинах кластера. В результате приложения, использующие базу данных, не должны заботиться о том, где выполняется их работа. СУБД отвечает за синхронизацию параллельно выполняемых действий и поддержание целостности базы данных.

Компьютеры, образующие кластер, — узлы кластера всегда относительно независимы, что допускает остановку или выключение любого из них для проведения профилактических работ или установки

дополнительного оборудования без нарушения работоспособности всего кластера.

В качестве вычислительных узлов в кластере обычно используются однопроцессорные персональные компьютеры, двух- или четырехпроцессорные SMP-серверы. Каждый узел работает под управлением своей копии операционной системы, в качестве которой чаще всего используются стандартные операционные системы: Linux, NT, Solaris и т.п. Состав и мощность узлов может меняться даже в рамках одного кластера, давая возможность создавать неоднородные системы. Выбор конкретной коммуникационной среды определяется многими факторами: особенностями класса решаемых задач, необходимостью последующего расширения кластера и т.п. Возможно включение в конфигурацию специализированных компьютеров, например, файл-сервера, и, как правило, предоставлена возможность удаленного доступа на кластер через Internet [6].

Из определения архитектуры кластерных систем следует, что она включает в себя очень широкий спектр систем. Рассматривая крайние точки, кластером можно считать как пару ПК, связанных локальной 10-мегабитной сетью Ethernet, так и вычислительную систему, состоящую из множества вычислительных узлов.

Таким образом, видно, что различных вариантов построения кластеров очень много. При этом в архитектуре кластера большое значение имеют используемые коммуникационные технологии и стандарты. Они во многом определяют круг задач, для решения которых можно использовать кластеры, построенные на основе этих технологий [12].

Применение кластера

Вычислительный кластер можно использовать для решения различных классов задач: большого объема последовательных программ, никак не связанных друг с другом; обработки данных распределенными программами, состоящими из большого числа независимых процессов, имеющих общие данные; вычисление параллельных программ, когда выполнение отдельных

операторов программы распределяется по узлам вычислительной сети, для сокращения времени выполнения программы.

В простом случае, когда на кластере требуется выполнить только одну параллельную программу, вся функциональность, необходимая для запуска, выполнения и завершения программы, может обеспечиваться операционными системами на узлах самой программой. В более сложных случаях, когда требуется выполнить несколько задач, как параллельных, так и последовательных, имеющих различные требования к системным ресурсам в дополнение к среде выполнения (т.е. операционной системе узла и, возможно, дополнительных библиотек, поддерживающих выполнение параллельных программ), необходима некоторая система, управляющая вычислительными ресурсами кластера в целом. Вычислительный кластер можно определить как множество рабочих компьютеров (узлов), связанных коммуникационной средой и способных работать как единое целое за счет дополнительного программного обеспечения [3].

1.2. Популярные системы развёртывания вычислительных кластеров

Система управления кластером в очень ограниченном смысле может рассматриваться как распределенная операционная система для кластера. Она обеспечивает управление над выполняемыми заданиями, пользователями и ресурсами.

Сегодня разработано достаточное количество таких систем управления вычислительным кластером. Рассмотрим наиболее используемые, а также наиболее простые, понятные из них.

В настоящее время для управления вычислительными кластерами используют такие системы как Condor, PBS, Rocks, PelicanHPC.

Системы Condor, а также PBS не являются бесплатными, а значит, для использования в школах малоприменимы [9].

1.3. Пакет Rocks

Rocks определяется как пакет развертывания, управления и поддержки кластера. С его помощью можно установить кластер на месте, имея в наличии одно лишь аппаратное обеспечение. Пакет содержит средства для запуска параллельных программ и программы для поддержки и расширения кластера после его первоначальной установки. Пакет распространяется в виде набора ISO-образов, которые нужно записать на несколько CD или DVD. Машина, которая должна быть главным узлом, устанавливается с CD или DVD. Следуя подсказкам мастера установки, устанавливается все необходимое для работы главного узла. Последним шагом является добавление остальных машин в качестве вычислительных узлов. Чтобы добавить вычислительный узел, нужно загрузить его по сети, и он будет добавлен в кластер и настроен автоматически. После добавления последнего узла, получается функционирующий кластер, пригодный для запуска параллельных программ.

Одно из важных свойств данного дистрибутива является возможность построения кластера с использованием полнокомплектных компьютеров.

Недостатком пакета Rocks может выступать тот факт, что для его развертывания необходимо проинсталлировать операционную систему на компьютер, который будет главным узлом, тогда как в других пакетных системах этого можно избежать, путем загрузки операционных систем с использованием Live CD или Live USB.

1.4. PelicanHPC

Из популярных систем развертывания вычислительных кластеров, использующих Live CD, можно выделить пакет Pelican HPC.

PelicanHPC является ISO-гибридом (CD или USB), что позволяет создать высокопроизводительный вычислительный кластер на несколько минут. Кластер "Пеликан" позволяет выполнять параллельные вычисления с использованием MPI. Можно запустить Пеликан на одной машине с

многоядерным процессором для использования всех ядер, или можно соединить вместе несколько компьютеров в кластер. Центральный узел (либо на реальном компьютере или виртуальной машине) загружается из образа. Вычислительные узлы загрузки по протоколу PXE, используя интерфейс узла в качестве сервера. Все узлы кластера получают свои файловые системы из одного образа, поэтому гарантируется, что все узлы запускают одинаковое программное обеспечение. Загрузочный образ создается путем запуска одного скрипта, который использует инфраструктуру Debian. Это позволяет очень легко создать собственную версию с новыми пакетами, добавив имя нужного пакета в скрипт и запустив его [24].

Система Pelican имеет преимущества перед другими системами в том, что, имея загрузочный CD/DVD диск или USB-накопитель, можно развернуть кластер за несколько минут даже в учебном классе. Узлы при такой конфигурации имеют возможность загружать операционную систему по сети. PelicanHPC хорош тем, что для развертывания кластера на его основе не обязательно выделять отдельную компьютерную аудиторию. Подойдет обычный школьный учебный класс, состоящий из нескольких машин, связанных общей единой сетью.

Однако, данная система однопользовательская, что ограничивает использование ее при проведении лабораторных работ с группой учащихся. Но эта проблема решаема предоставлением удаленного доступа к Frontend-узлу кластера [18].

1.5. Вычислительные кластеры, удовлетворяющие особенностям проведения занятий в учебных заведениях

Чтобы построить вычислительный кластер на основе учебного класса он должен удовлетворять некоторым требованиям.

Одним из них является то, что после установки кластера в компьютерном классе, этот класс мог бы продолжать использоваться в рамках стандартной образовательной программы. Поэтому преимущественно использовать системы развертывания, которые устанавливаются по сети, и не

используют жёсткие диски компьютеров учебного класса. Компьютеры, на которых планируется установить кластер, должны иметь возможность загружать операционную систему по сети.

Вторым требованием является возможность управлять кластером дистанционно, чтобы было можно проводить полноценные занятия с группой учащихся. В такой группе каждый пользователь с отдельного компьютера подключается к кластеру [4].

1.5.1 Характеристика и возможности PelicanHPC

Особенности:

- Управляющий узел может представлять собой реальный компьютер и загружаться с помощью компакт-диска или USB-устройства. В другом варианте возможно использование виртуальной машины, которая загружается с помощью файла образа диска. При этом последний вариант может быть использован в то же время, когда работает хостовая ОС, которая может являться любой из распространенных операционных систем.
- Вычислительные узлы, как правило, состоят из реальных компьютеров для достижения максимальной производительности. Однако для настроек, проверки и компоновки PelicanHPC узлы также могут быть и виртуальными.
- Поддержка параллельных вычислений на основе MPI с использованием Fortran (77, 90), C, C + +, GNU Octave и Python.
- Предлагается Open MPI реализация MPI.
- Кластер может быть изменен в масштабе, чтобы добавить или удалить узлы, используется команда "pelican_restarthpc".
- Легко расширяемый для добавления пакетов. Также легко изменяемый, поскольку образ компакт-диска PelicanHPC создается с помощью одного скрипта, который основывается на Debian Live системе. По этой причине, дистрибутивы довольно простые и легкие.
- Содержит примеры программ: Linpack HPL (в настоящее время v2.0) тесты с многочисленными примерами, которые используют GNU Octave .

Требования и ограничения:

- Вычислительные узлы должны быть загружены по сети. Это вариант поддерживают все современные сетевые карты поставляющиеся с материнскими платами, должна быть включена опция в BIOS Setup. Управляющий узел PelicanHPC работает как DHCP сервер. Нельзя использовать его в локальной сети с уже существующим DHCP-сервером, чтобы избежать возможные конфликты. Это может стать причиной проблем с сетевым администратором.
- Кластер PelicanHPC предназначен для использования одним человеком — то есть только один пользователь, с именем "user".
- Текущие версии только для 64-битных процессоров (Opteron, Turion, Core 2, и т.д.).

При необходимости скрипт make_pelican может быть использован, чтобы сделать 32-битную версию.

- На веб-сайте PelicanHPC перечислены некоторые другие подобные дистрибутивы, которые могут быть более подходящим для определенных целей [24].

Глава 2. Методические рекомендации по организации вычислительного кластера в учебном классе

2.1. Архитектура локальной сети учебных классов

Локальная вычислительная сеть (ЛВС или LAN – Local Area Network) представляет собой коммуникационную систему, позволяющую совместно использовать ресурсы компьютеров, подключенных к сети, таких как принтеры, плоттеры, диски, модемы, приводы CD-ROM и другие периферийные устройства. Локальная сеть обычно расположена в одном или нескольких близко расположенных зданий.

В самом экономичном варианте можно построить одноранговую локальную сеть, в которой все компьютеры выступают как в роли пользователей (клиентов), так и в роли хранителей данных (серверов). При этом все имеют равный доступ ко всем ресурсам других участников сети, будь то файлы или, например, принтеры.

Часто в составе сети выделяют отдельный компьютер – выделенный сервер, который используется для хранения общих ресурсов (архива, дистрибутивов программ и т. д.). Кроме того, сервер может понадобиться для разграничения прав пользователей (если это необходимо) и организации коллективного доступа в Интернет.

Топология сети – это способ связи компьютеров. Для малых сетей самым простым вариантом является топология сети типа «звезда». При таком соединении обмен данными между рабочими станциями происходит через центральный узел по отдельным линиям [11].

Физически каждая линия представляет собой специальный кабель, при помощи которого каждый компьютер в сети соединен с устройством, называемым концентратором (Хабом) или с маршрутизатором.

Маршрутизатор - специализированный сетевой компьютер, имеющий два или более сетевых интерфейсов и пересылающий пакеты данных между различными сегментами сети. Маршрутизатор может связывать разнородные

сети различных архитектур. Для принятия решений о пересылке пакетов используется информация о топологии сети и определённые правила, заданные администратором.

Маршрутизаторы работают на более высоком «сетевом» (третьем) уровне сетевой модели OSI, нежели коммутатор (или сетевой мост) и концентратор (хаб), которые работают соответственно на втором и первом уровнях модели OSI.

Работа такой сети будет нарушена, если только из строя выйдет центральный узел, будь то хаб или выделенный сервер. Выход из строя одной из рабочих станций никак не скажется на работе остальных пользователей. Подключение новых компьютеров также никак не скажется на общей работе.

Каждый компьютер в составе ЛВС должен иметь сетевой адаптер, который собственно и подключается специальным кабелем к хабу и сетевую операционную систему.

Технические средства создания ЛВС

Сетевые карты

Могут быть включены в состав системной платы или вставляться в специальные разъемы на плате (обычно это PCI разъемы). При установке сетевой карты, кроме диска с дистрибутивом Windows, может потребоваться дискета с драйвером сетевой карты (обычно поставляется вместе с сетевой картой). Драйвер сетевой карты – специальная программа для ее подключения.

После установки драйвера в Windows в разделе настроек «Система» появится сетевой адаптер, а в разделе «Сетевые подключения» автоматически установятся несколько протоколов (алгоритмов общения сетевых устройств). Среди них будет присутствовать TCP/IP (Transmission Control Protocol/Internet Protocol), изначально придуманный для работы в сети Internet, но также используемый и для работы в локальных сетях. В большинстве локальных сетей используется именно этот протокол.

Проводка

Подключение сетевой карты к хабу осуществляется кабелем типа «витая пара» – UTP (Unshielded Twisted Pair). Обычно это четыре пары скрученных между собой изолированных проводников, помещенных в оболочку. Каждый проводник окрашен в свой цвет и имеет специальный номер. Витая пара бывает нескольких типов. Чаще всего используют витую пару 5-й категории со скоростью передачи до 100 Мбит в секунду. На каждый конец кабеля устанавливают вилки RJ-45, которые подсоединяют к сетевой карте и хабу. Кабель с двумя вилками иногда называют «патч-корд». Часто кабель, идущий от хаба подсоединяют к розетке, а уже розетку соединяют с компьютером коротким патч-кордом.

Хаб (другое название – Разветвитель)

Служит для коммутации всех рабочих станций сети. Хаб с функциями усилителя сигнала, называют активным концентратором. К нему с помощью соединительных кабелей подключаются компьютеры с сетевыми картами. Задача хаба проста – передавать пакеты данных пришедшие с одного из компьютеров, на все другие. Хаб может иметь 4, 5, 8, 12, 16 и более портов (порт – разъем на хабе для соединения с компьютером). Разъем порта такой же, как и на сетевой карте [2].

Для работы с сетью каждое устройство, подключаемое к этой сети должно иметь определенный электронный идентификатор – IP-адрес, а также настроенные параметры маски, шлюза и данные о DNS-сервере. Этот адрес сетевой интерфейс при подключении сразу пытается получить от специального оборудования, которое занимается автоматической раздачей адресов. Если такое оборудование не находится, то пользователю приходится ввести параметры сети вручную. Если в сети множество компьютеров, то возможны ошибки в адресации и дублирование адресов, ведущие к сбоям в системе и замедлению производительности сети в целом. В домашней сети, при отсутствии автоматической раздачи адресов, приходится прописывать параметры сети вручную на каждом устройстве для подключения, например,

к роутеру. Не каждый пользователь обладает достаточным объемом знаний и опыта для правильного изменения таких настроек. Поэтому в домашних условиях тоже многие пользователи стараются развернуть автоматическую раздачу адресов, то есть развернуть сервер DHCP.

Во время подключения любого клиентского устройства к сети происходит специальная широковещательная отсылка запроса в сеть с целью поиска сервера DHCP, раздающего параметры этой сети. DHCP создает пакет с ответом на запрос клиента, в который могут включаться такие настройки, как адрес IP, маска сети, параметры шлюза, адреса DNS-серверов, название домена и т. д. и отправляет этот пакет клиентскому устройству. Клиент принимает подтверждающий сигнал от сервера DHCP. Сформированный пакет данных является стандартизированным, поэтому его могут расшифровать и использовать практически любые операционные системы.

Параметры, выданные сервером для клиентского устройства, имеют ограниченный настраиваемый срок действия, у которого есть свое название - «время аренды». Адреса, выдаваемые сервером, анализируются на совпадение действующих адресов с неистекшим временем аренды, поэтому дублирование адресов исключается. Обычно ставится срок аренды небольшой - от нескольких часов до 4-6 дней. По истечении этого срока устройство повторяет запрос к серверу и получает от него этот же адрес (если он еще свободен) или любой свободный [11].

2.2. Место временного вычислительного кластера в учебном заведении

На сегодня в образовательных учреждениях суперкомпьютерные системы малодоступны либо недоступны вообще. Решением может послужить создание временного (на период проведения занятий) вычислительного кластера на основе имеющихся аппаратных средств. Для использования кластера на основе PelicanHPC подойдет обычный школьный компьютерный класс, машины которого обладают возможностью загрузки операционной системы (в нашем случае PelicanHPC) по сети. Такое решение позволит использовать компьютерный класс как для занятий по суперкомпьютерным

технологиям и параллельному программированию, так и как обычный класс в остальное время. Установка такого кластера позволит без вмешательства в операционные системы компьютеров, разворачивать кластер в любое необходимое время, например, для проведения олимпиад по параллельному программированию.

2.3. Модификация локальной сети для установки вычислительного кластера в учебном классе

Вычислительный кластер в учебном заведении может быть реализован посредством двух вариантов модификации имеющейся локальной сети.

Изначально PelicanHPC возможно использовать только в однопользовательском режиме, к тому же он имеет собственный DHPC-сервер. Ввиду последнего, необходимо изолировать локальную сеть компьютеров, на основе которых будет создаваться кластер, чтобы избежать конфликтов с DHPC-сервером, который обслуживает этот класс. Сделать это можно отключив коммутатор локальной сети класса от внешней сети, и получив полностью изолированный сегмент локальной сети (Рис 3).

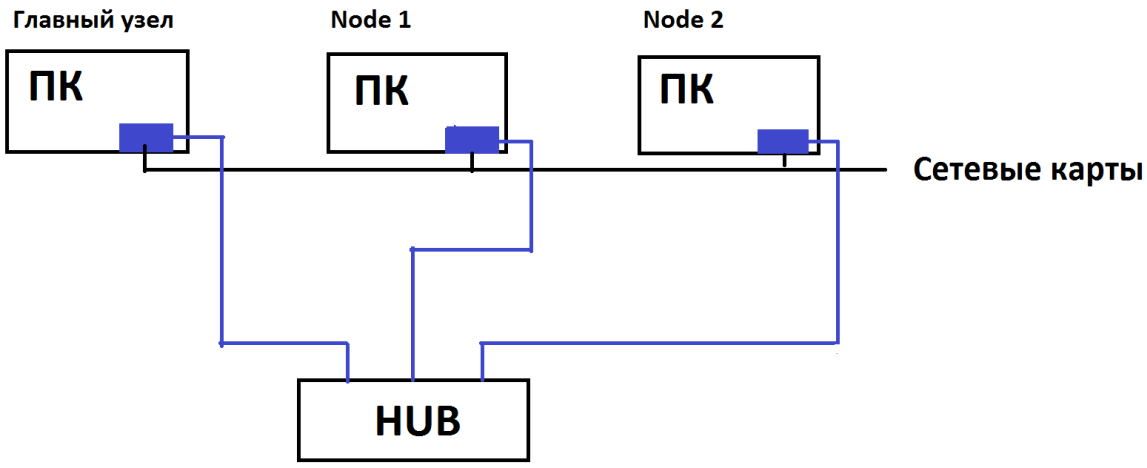


Рисунок 3 - изолированный сегмент локальной сети .

Такую реализацию кластера можно использовать для демонстрации преимуществ использования суперкомпьютера перед обычными персональными компьютерами, а также для некоторых целей преподавателя или иного лица, нуждающегося в применении кластера.

Второй вариант предполагает использование головного узла кластера, содержащего в себе две сетевые карты. Одна карта для соединения с вычислительными узлами, вторая для выхода во внешнюю сеть. При такой конфигурации сети DHPC-сервер, расположенный в системе PelicanHPC будет работать с IP-адресами отдельно для внутренней сети (Рис.4).

Рисунок

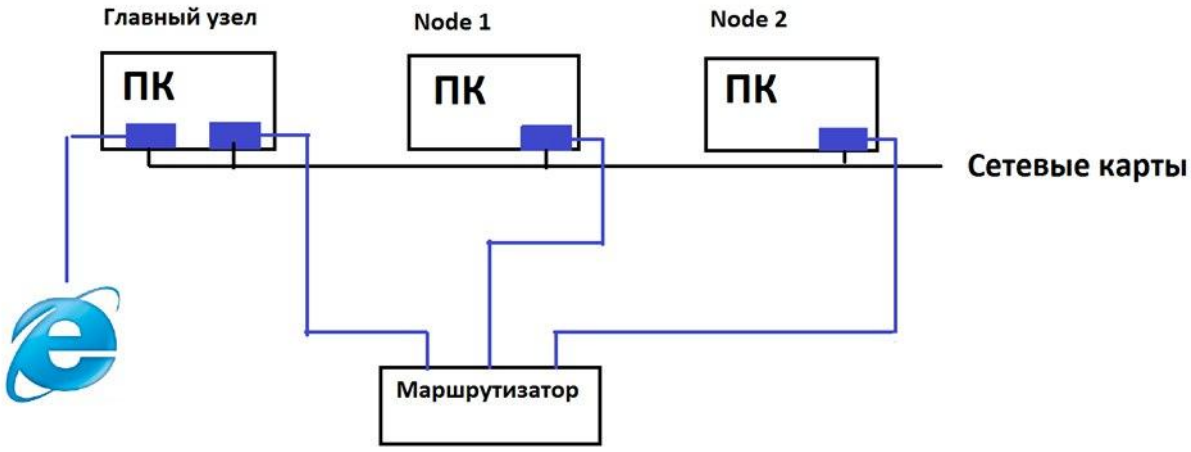


Рисунок 4 - Подключение главного узла к внешней сети

Такое соединение вычислительных узлов по локальной сети обеспечит возможность эффективной его работы и последующую загрузку программного обеспечения на главный узел, а также загрузку операционной

системы вычислительных узлов по сети, избежав инсталляции на винчестер компьютера.

Как видно из схемы (Рис. 4), на головном узле установлено две сетевые карты. Это позволит реализовать удаленное подключение к кластеру по внешней сети пользователей, которые могут находиться в другой аудитории и будут иметь возможность запускать оттуда свои задачи на кластере (Рис. 5).

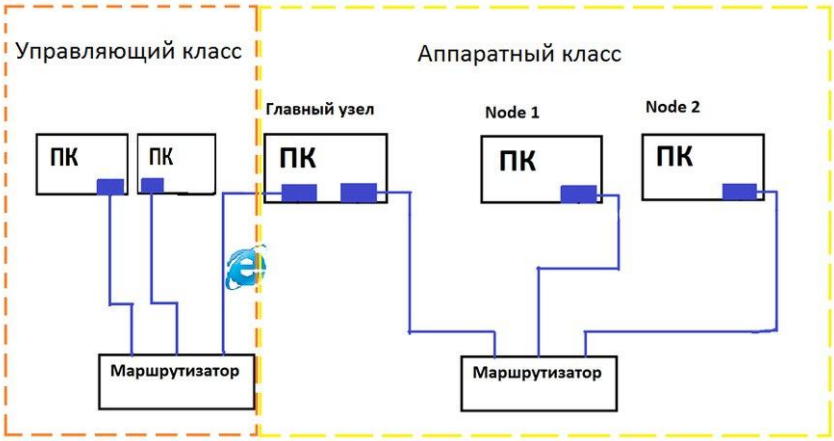


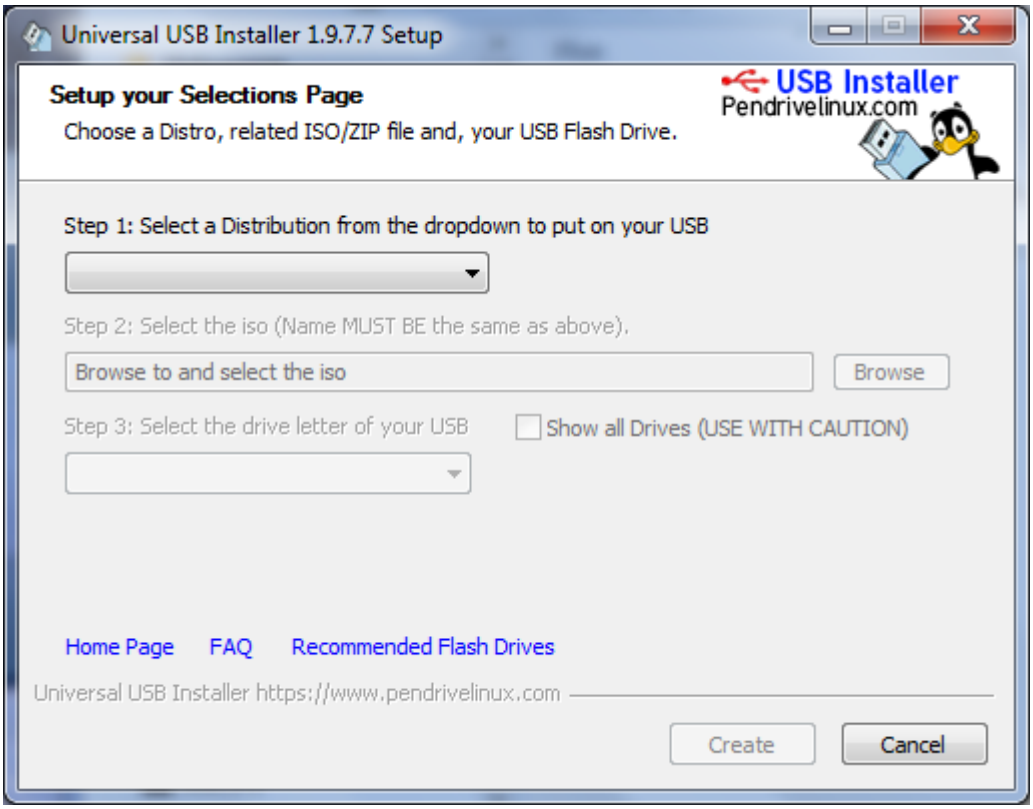
Рисунок 5 - удаленное подключение к кластеру пользователей, которые могут находиться в другой аудитории.

2.4. Инструкция по развертыванию кластера посредством «PelicanHPC».

Для работы кластера необходимо, чтобы компьютеры, которые будут использоваться в качестве вычислительных узлов, были объединены общей локальной сетью. Для этого в компьютерной аудитории должны присутствовать: маршрутизатор, стандартная локальная сеть (протокол Ethernet), дополнительная сетевая карта для головного узла кластера, подключение к которой обеспечит возможность соединение кластера с внешней сетью. Если использование одной и той же компьютерной аудитории для развертывания временного кластера планируется постоянное, то в один из компьютеров класса следует вмонтировать дополнительную сетевую карту. Если же развертывание кластера будет происходить каждый раз в разных аудиториях, то целесообразней выделить отдельный компьютер

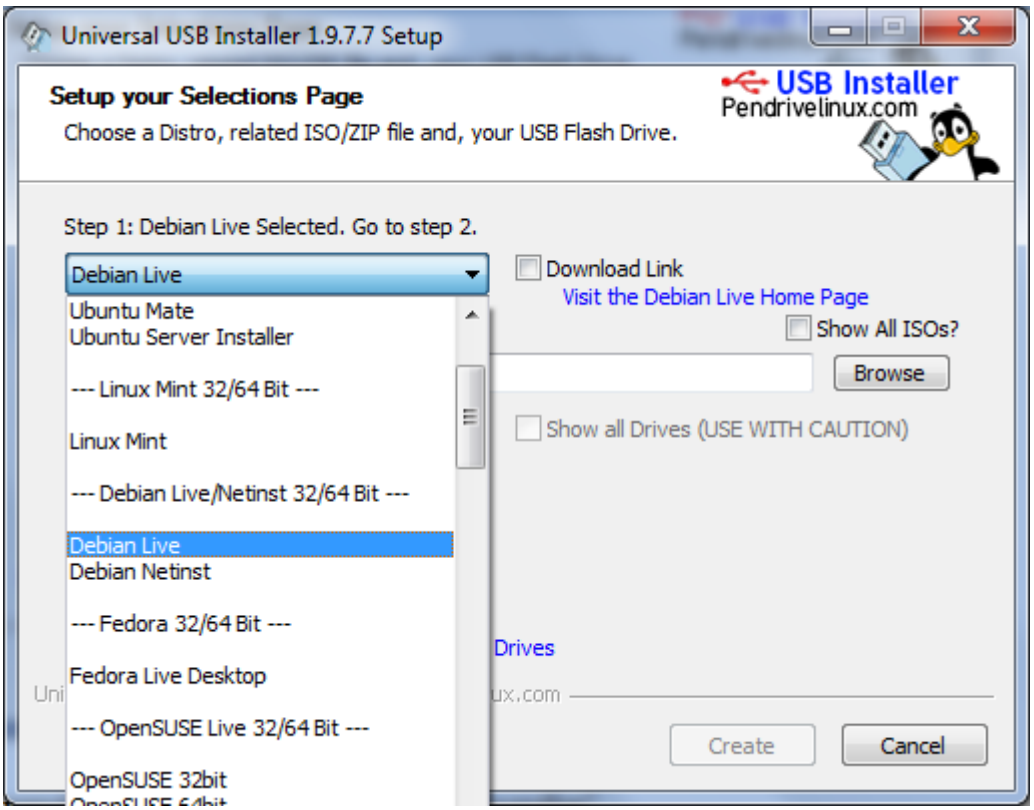
с двумя сетевыми картами, который будет переноситься в нужный класс. Так как этот компьютер будет являться головным узлом кластера, к нему не предъявляются требования по производительности, Все расчеты и решения задач будут происходить непосредственно на вычислительных узлах.

Для развертывания кластера, потребуется USB накопитель с дистрибутивом PelicanHPC, загрузить который можно с официального сайта PelicanHPC. Чтобы создать загрузочную флешку с дистрибутивом PelicanHPC, необходимо с официального сайта PelicanHPC скачать iso образ системы. Затем воспользоваться утилитой "universal-usb-installer" с помощью которой создается загрузочная флешка.



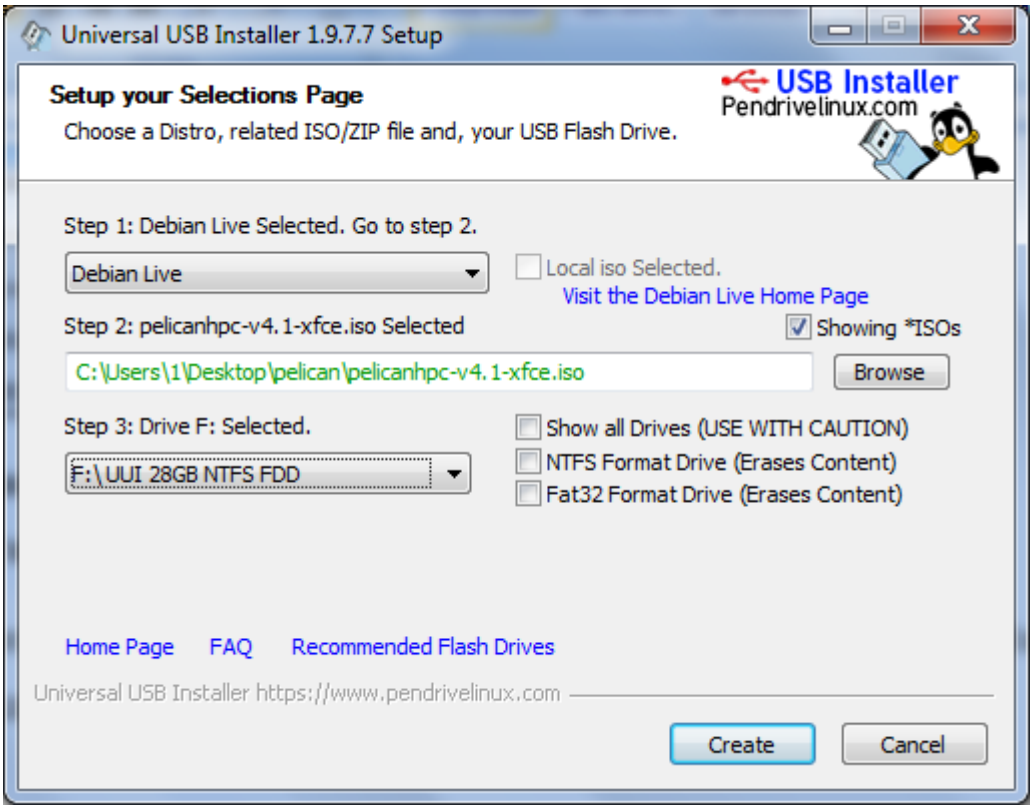
Главный экран утилиты "universal-usb-installer"

Из списка "Step 1" выбираем дистрибутив, на основе которого будет создаваться загрузочная флешка. Так как PelicanHPC основан на системе Debian Live, выбираем его.

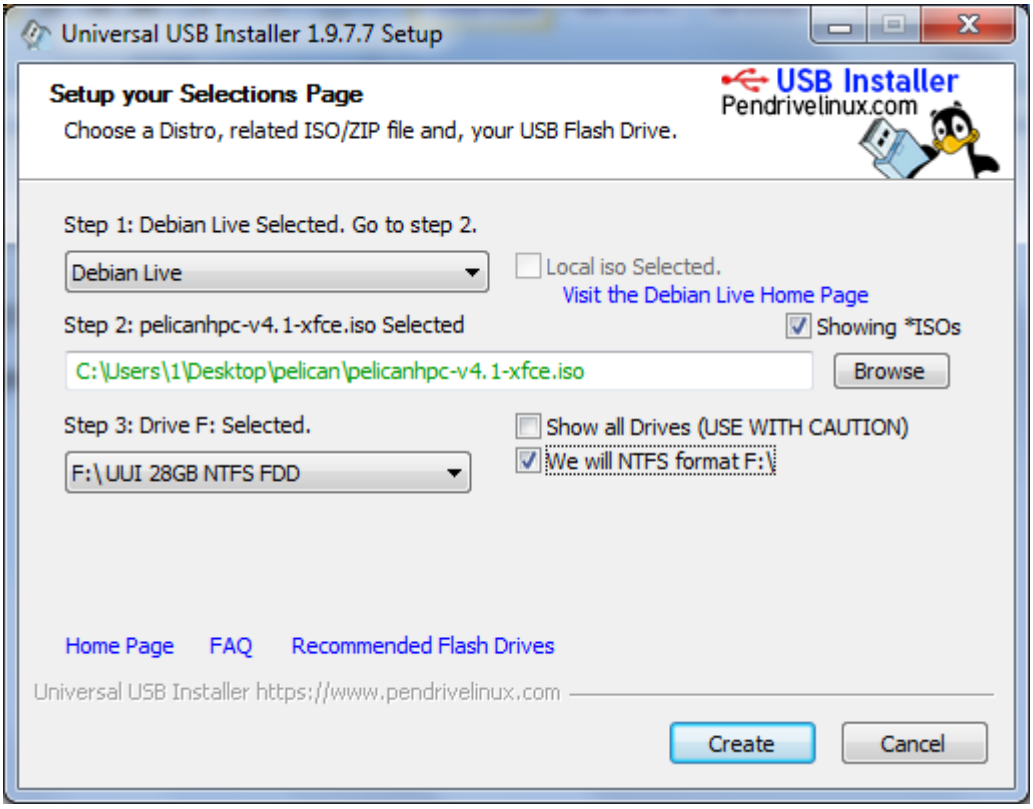


Затем ставим галочку возле "Show All ISOs?" и нажимаем на кнопку "Browse" выбираем скачаный iso образ "pelicanhpc-v4.1-xfce".

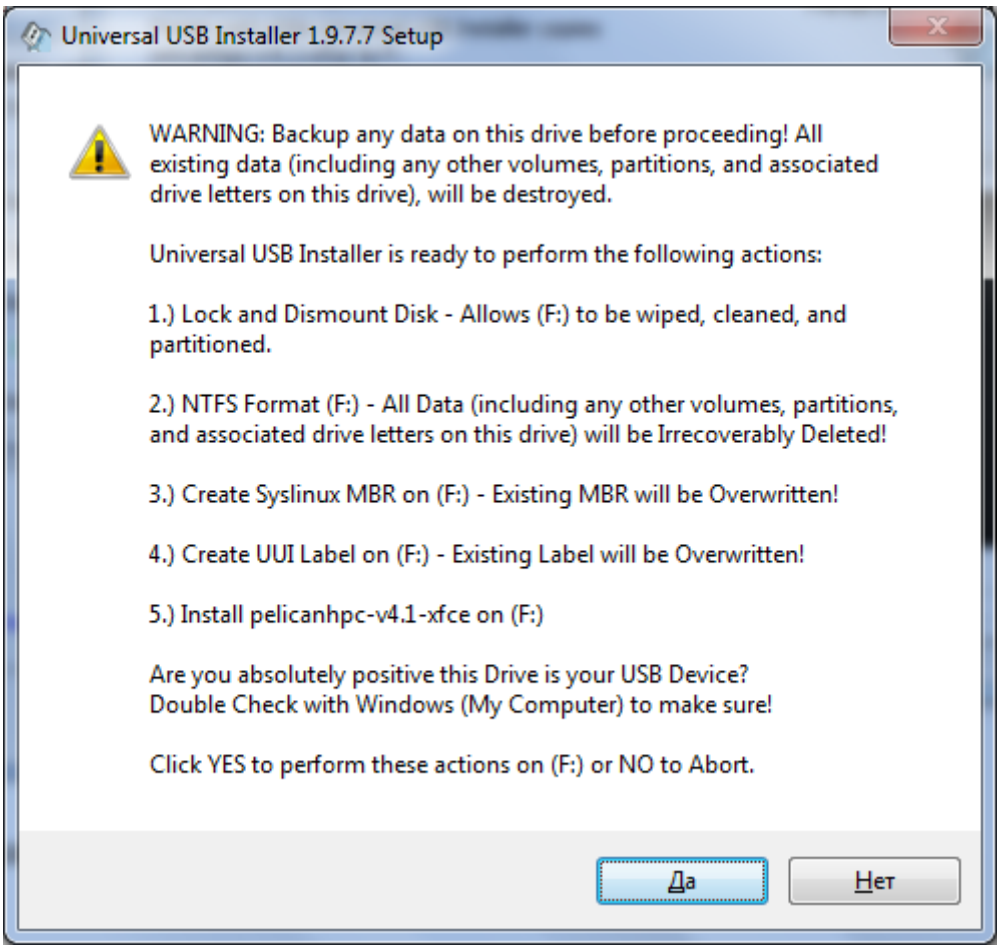
После этого во вкладке "Step 3" выбираем USB-карту на основе которой и будет создана загрузочная флешка.



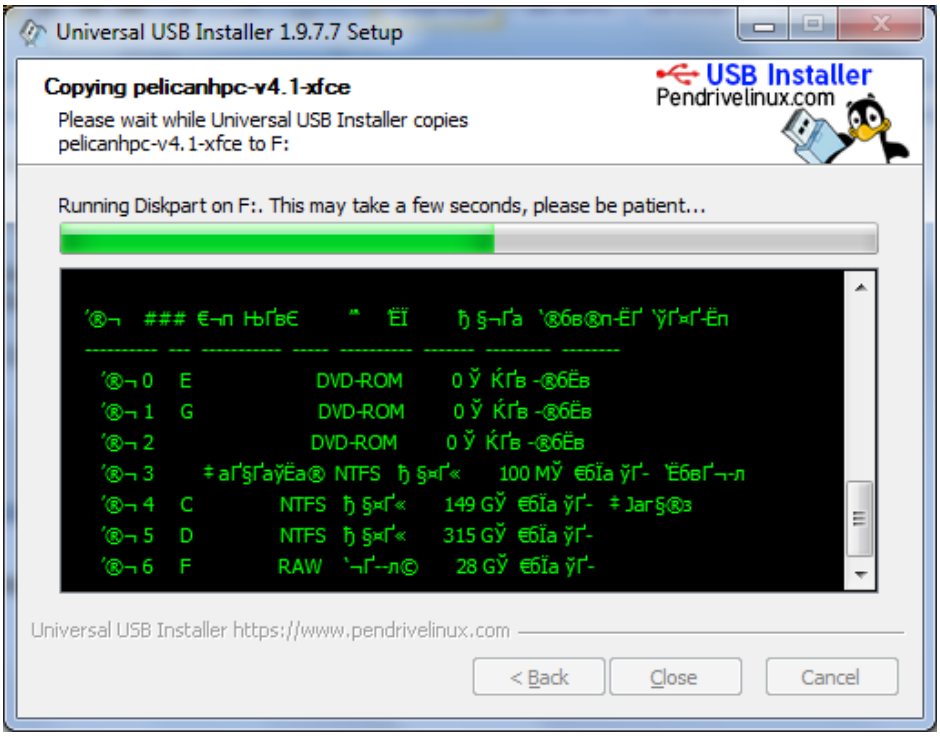
Справа нужно поставить галочку на "NTFS Format Drive", для форматирования USB-карты в формат "NTFS".

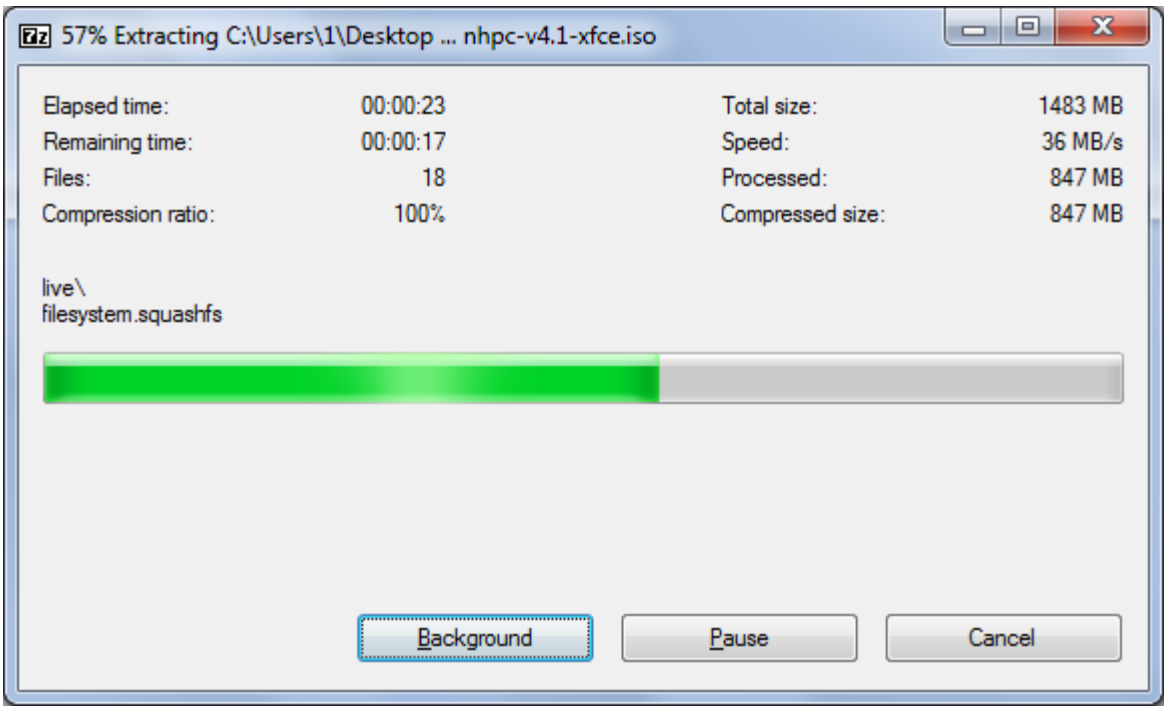


И нажимаем кнопку "Create".



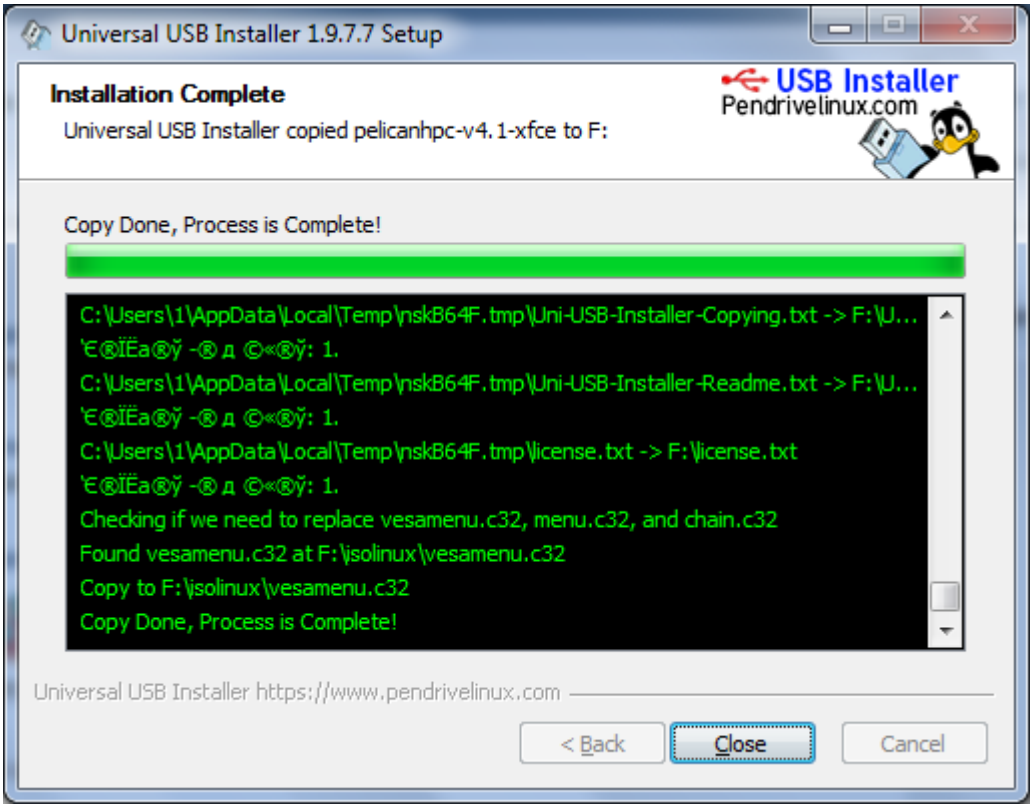
Нажимаем "Да". Начинается процесс создания загрузочной USB-карты.





Процесс длится несколько минут.

После завершения появится вот такое окно:



Нажимаем "Close". Все, загрузочная флешка готова.

С этого внешнего накопителя загружается операционная система кластера (без установки ее на винчестер) на компьютере, непосредственно за

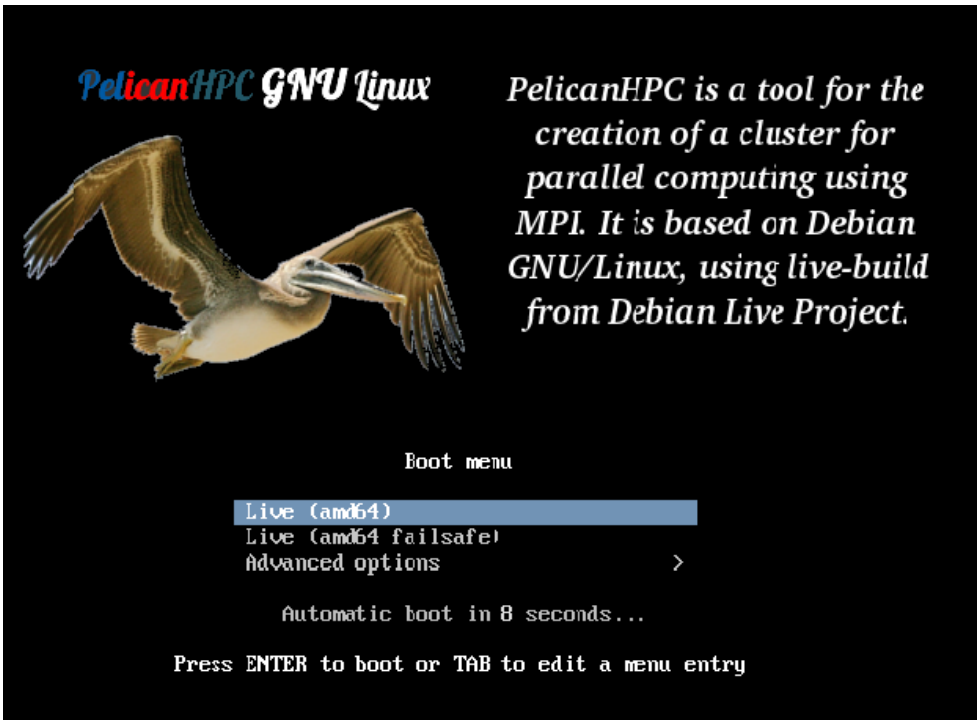
которым будет работать пользователь (или подключаться к нему через внешнюю сеть), компилируя и запуская параллельные программы.

1. Необходимо выделить компьютеры, на которых будет разворачиваться кластер, установив настройки BIOS на главном узле для загрузки ОС с USB накопителя, на вычислительных - загрузку по сети.

В разных версиях BIOS это делается по-разному. В приоритете загрузки (вкладка BOOT) должна быть выбрана сетевая загрузка (Network). Для загрузки ОС головного узла выбирается внешний накопитель (USB FLASH).

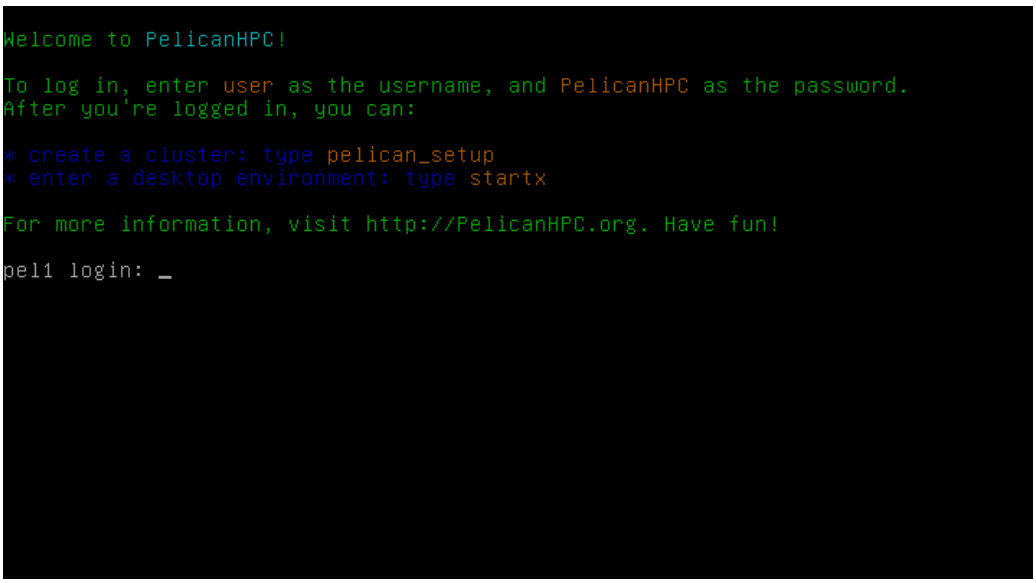
Также некоторые версии BIOS требуют предоставления доступа к сетевым разъемам (вкладка Advanced).

2. После того как компьютеры были настроены, подключаем к главному узлу USB накопитель и запускаем машину. На мониторе высветится меню системы.

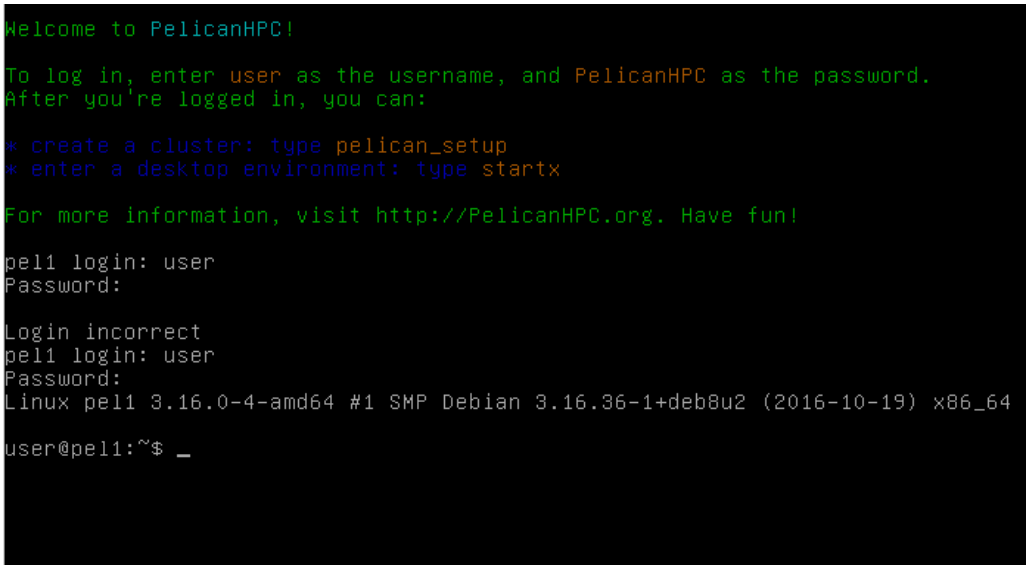


В меню выбираем первый пункт и система начинает загрузку.

Затем появится экран приветствия



Для входа в систему нужно использовать логин "user" и пароль “PelicanHPC”.



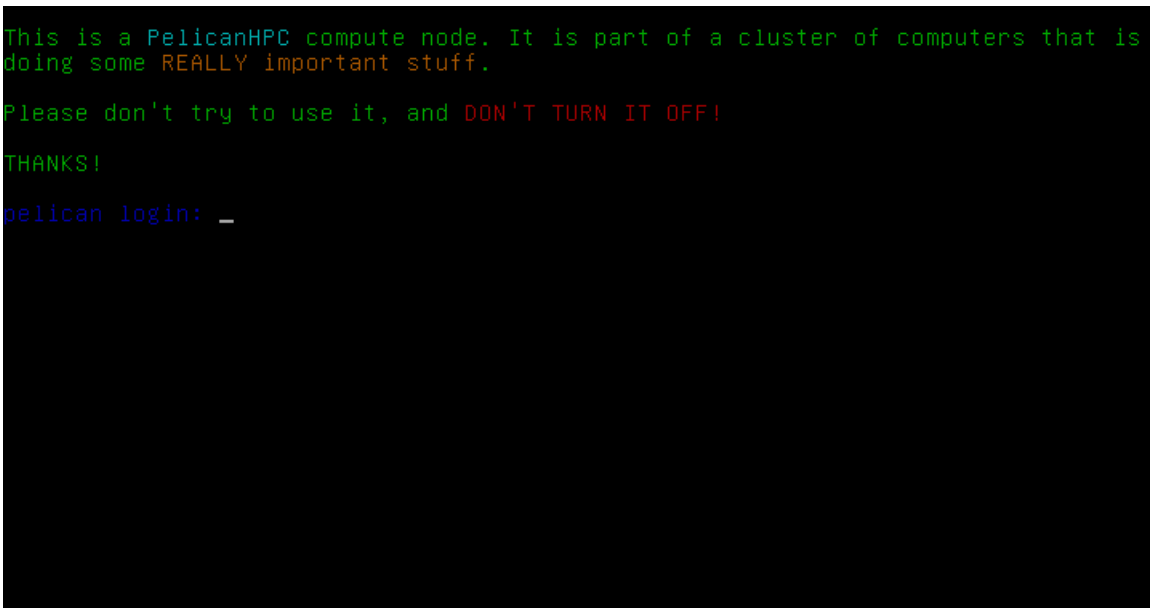
3. Когда выполнен вход в систему консоли кластера, необходимо подключить к кластеру все вычислительные узлы. Для этого нужно запустить команду конфигурации кластера pelican_setup.

Система спрашивает, будем ли мы конфигурировать сетевую загрузку вычислительных узлов.

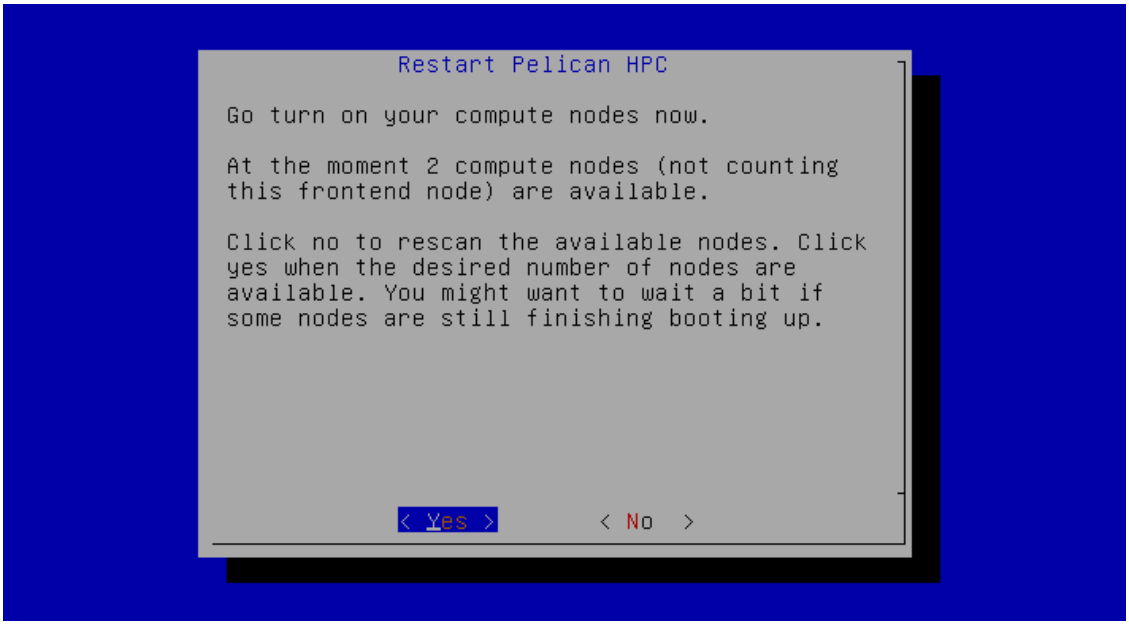
Выбираем "Yes".

Сконфигурировав сервер сетевой загрузки, программа предложит выполнить загрузку всех стальных узлов кластера:

В этот момент нужно включить все остальные компьютеры кластера Вмешательства в процесс загрузки вычислительных узлов кластера не требуется. Надо просто дождаться, когда они все закончат процедуру загрузки, о чем будет свидетельствовать следующая информация на экранах этих компьютеров.

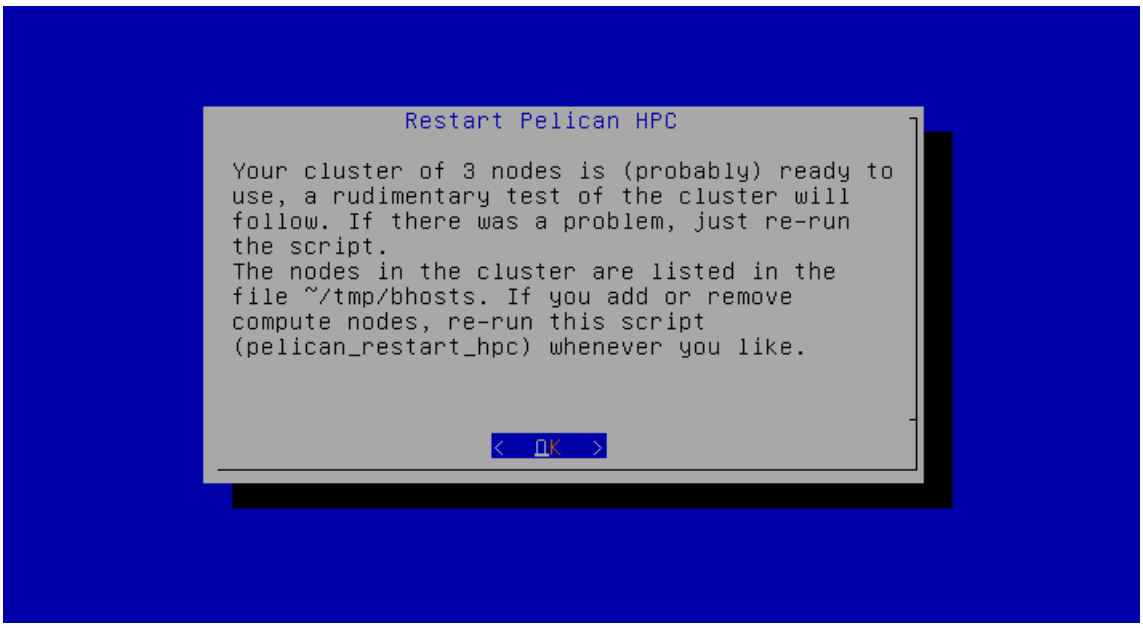


На экране центрального узла программа сообщает, сколько было найдено вычислительных узлов (в данном случае один) кроме узла, который является консолью кластера. Если все нормально - нажимаем "Yes".

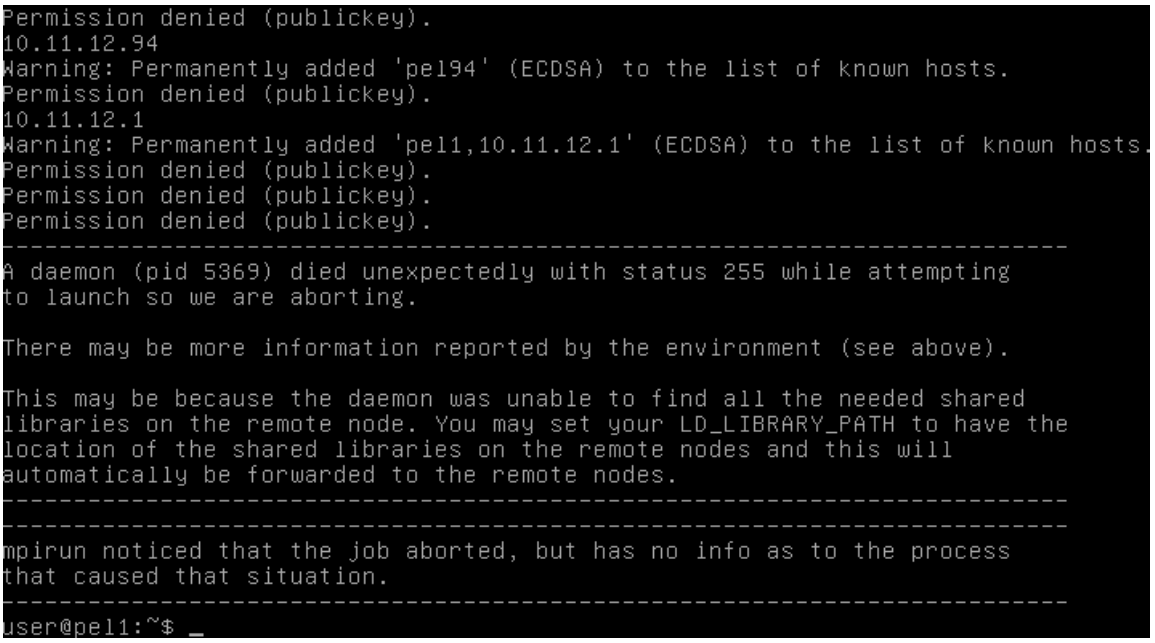


Если вычислительные узлы загрузились, но на этом экране в счетчике их нет, то нажимаем ""NO". После этого должны отобразиться все узлы, включенные в сеть.

После этого программа конфигурации кластера сообщает, что все настройки выполнены и кластер готов к эксплуатации:



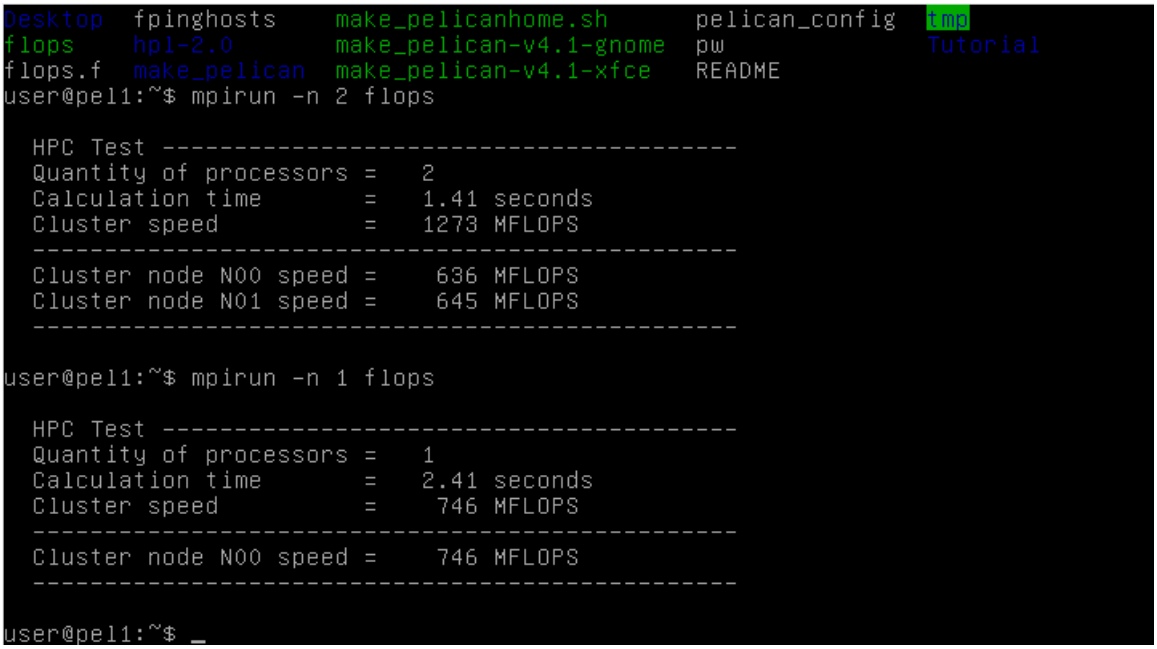
Нажимаем «ОК».



Теперь кластер готов к использованию.

Проверить его работу можно на тестовой программе вычисления числа flops.f, которая имеется в файловой системе ОС.

Откомпилировать программу в параллельной среде MPI можно с помощью команды mpirun –n 2 flops (где n 2= количество вычислительных узлов).



Как видно из результатов работы программы, скорость вычисления на двух узлах примерно в два раза больше, чем на одном. То есть кластер делает именно то, что ожидалось.

Результатом выполнения программы является полученное время выполнения и производительность кластера, выраженная в MFLOPS (количество операций с плавающей точкой, которое выполняет ЭВМ за одну секунду). Как видно из результатов работы программы, скорость вычисления на двух узлах примерно в два раза больше, чем на одном. То есть кластер делает именно то, что ожидалось.

В случае необходимости удаленного управления кластером, например, при проведении занятий с группой учащихся, использовать нужно вариант конфигурации сети с двумя сетевыми картами.

Удаленное подключение к кластеру изначально не предусмотрено системой PelicanHPC. Для того, чтобы его настроить, потребуется внести некоторые правки в систему:

Проинсталлировать Midnight commander – файловый менеджер. Он потребуется для удобства обращения с файлами. Сделать это можно командой sudo apt-get install mc. Команда вызывает процедуру установки. Следуя подсказкам на экране, менеджер устанавливается довольно легко.

Проверкой правильности его инсталляции в Pelican служит запуск Midnight commander командой mc [20].

Аналогичным образом устанавливается сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой – команда sudo apt-get install ssh.

Для удаленного доступа к кластеру, нужно внести правки в конфигурационный файл протокола ssh [19].

Набираем команду sudo mc – запускается Midnight commander в режиме суперпользователя (режим необходим для редактирования и сохранения системных файлов). Следуя по пути /etc/ssh, находим файл sshd_config. Нажимаем «F4» (редактирование файла). Пользователю будет предложено на выбор три разных редактора файлов, выбираем третий, так он более удобен в обращении. В этом конфигурационном файле нужно найти следующую строку: PasswordAutentification no (идентификация по паролю отключена). Вместо «no» прописываем «yes» и сохраняем файл клавишей «F2», тем самым обеспечив включение парольного доступа к вычислительному кластеру. Чтобы изменения вступили в силу, необходимо перезапустить сервис ssh командой: sudo service ssh restart [13]. С этого момента возможен удаленный доступ к вычислительному кластеру через внешнюю сеть по протоколу ssh, например, при помощи программы PuTTY [15].

2.5. Особенности проведения занятий по суперкомпьютерным технологиям в учебном классе

Для проведения полноценных занятий по параллельному программированию, и соревнований или олимпиад по параллельному программированию временный вычислительный кластер должен быть развернут до начала занятия в компьютерном классе. Этот класс будет использоваться только как аппаратная часть вычислительного кластера. Непосредственно сами занятия проводятся в другой аудитории, подключение к кластеру осуществляется удаленно по внешней сети.

Примерное олимпиадное задание может иметь следующий вид: "Вам дан код последовательной программы. Необходимо ее распараллелить так, чтобы время выполнения оказалось минимальным".

Программа, выполняющая роль задания загружается преподавателем на запущенный кластер. Для этого можно использовать утилиту WinSCP, которая обладает возможностью переноса файлов из одной операционной системы в другую. Эта же утилита может быть использована обучающимися при работе с кластером для переноса/сохранения своих данных в файловую систему компьютера, за которым они работают. Для замера времени выполнения программы, все учащиеся должны использовать функцию, например, «clock()», она показывает время работы программы. После выполнения программы преподаватель фиксирует результаты каждого учащегося, сравнивает все результаты, анализирует и делает выводы о том, кто справился лучше.

2.6. Недостатки PelicanHPC и пути их преодоления организационными методами

Официальная версия PelicanHPC обладает рядом недостатков, которые вызывают неудобства при полноценном использовании кластера. Самый главный недостаток на наш взгляд, это отсутствие полноценного менеджера задач в системе PelicanHPC , который вызывает ряд проблем, возникающих при его использовании на учебных занятиях. При выполнении нескольких программ, запущенных несколькими пользователями одновременно, отследить время выполнения конкретной программы оказывается проблематично. Связано это с тем, что при запуске нескольких задач одновременно, время их выполнения квантуется. Выполнение задач происходит маленькими фрагментами поочередно, что вызывает наложение времени выполнения от разных программ в один счетчик.

Обойти этот недостаток возможно, правильно организовав процесс проведения занятия, преподаватель должен следить чтобы учащиеся запускали программы в порядке очереди.

Также важным недостатком является невозможность сохранение всех настроек системы. Поскольку пеликан это Live-система, после ее перезагрузки все данные и установленные службы будут возвращены в исходное состояние. При этом частичным решением может послужить утилита WinSCP [16]. Она позволит перенести файлы из системы пеликан в операционную систему компьютера, на котором осуществляется управление кластером.

2.7. Методика использования вычислительного кластера на основе PelicanHPC в учебном процессе

Подготовка к занятию.

До начала занятия нужно сконфигурировать сеть, выделить компьютеры, которые будут использоваться в качестве вычислительных узлов. Головным узлом будет выступать компьютер, содержащий две сетевые карты. Так как этот компьютер будет являться головным узлом кластера, к нему не предъявляются требования по производительности, решения задач будут происходить непосредственно на вычислительных узлах.

Затем произвести первоначальную настройку кластера, настроив удаленный доступ по протоколу ssh и проинсталлируя Midnight commander.

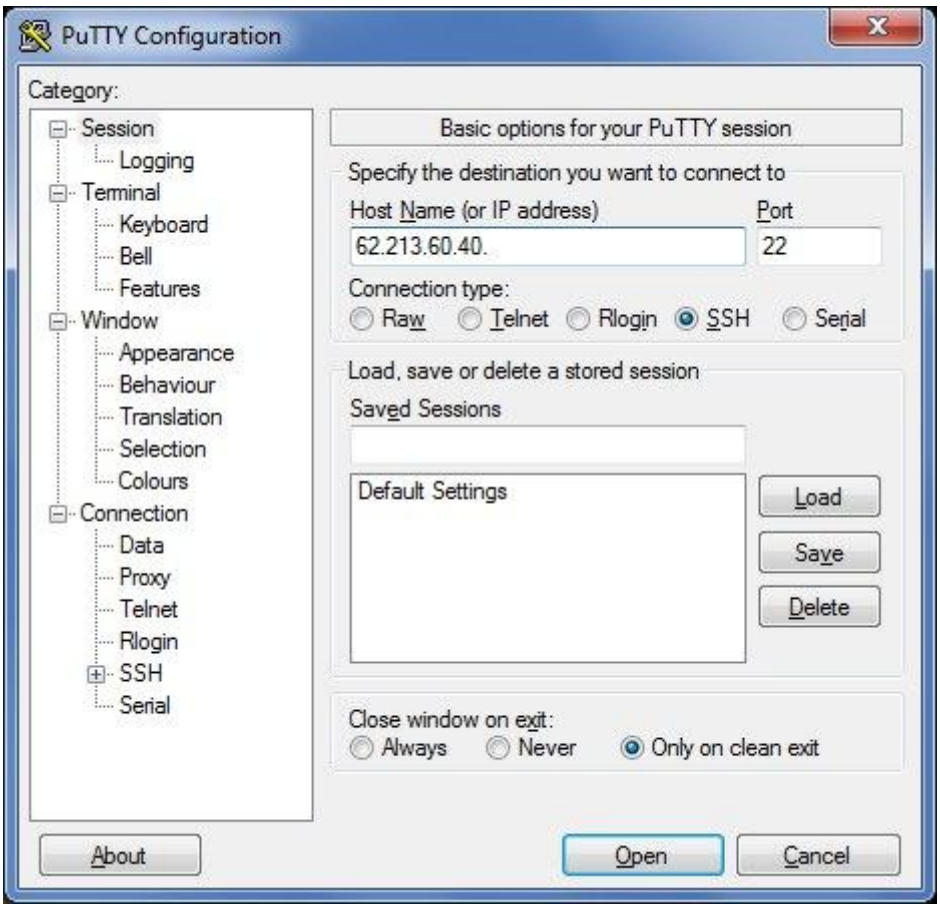
Используя утилиту WinSCP, поместить в файловую систему кластера задание для учащихся. Так как все пользователи, которые подключатся удаленно к кластеру заходят под одним и тем же логином и паролем, все они будут видеть файл с заданием, положенный в конкретный каталог.

Написать на доске логин и пароль для доступа к учетной записи кластера, а так же IP-адрес для удаленного подключения.

Проведение занятия. Организационный момент.

В начале занятия преподаватель демонстрирует обучающимся подключение к кластеру при помощи приложения PuTTY.

В программе указываем IP-адрес кластера, который вы можете видеть на экране



Объяснить задание, показать на экране путь, где располагается файл с заданием. Провести краткий инструктаж, касающийся времени, отведенному на выполнение задания. Акцентировать внимание учащихся на том, чтобы после выполнения программ учащиеся запускали их в порядке очереди для точного определения результатов [15].

По мере выполнения обучающимися заданий и запуска ими программ преподаватель фиксирует у себя результаты каждого ученика. После того, как все выполнили задание делает вывод о проделанной ими работе и выделяет тех учащихся, которые справились с заданием лучше, то есть у кого параллельная программа выполнялась на кластере быстрее всего.

Подобные занятия могут послужить неплохим вариантом для обучающихся, интересующихся суперкомпьютерными технологиями и параллельным программированием, поскольку дают непосредственное понимание и демонстрацию работы вычислительного кластера.

Глава 3. Модификация пакета PelicanHPC для учебных целей

3.1. Вычислительные кластеры основанные на PelicanHPC

На официальной странице PelicanHPC приведены подобные дистрибутивы, которые могут быть более подходящими для целей в некой конкретной области. Некоторые из них созданы путём модификации дистрибутива PelicanHPC[25].

Например, для исследования молекулярных взаимодействий между белками, которое имеет большое значение для фармацевтической промышленности, был создан дистрибутив MOLA. Кластер на основе MOLA позволяет производить вычисления такого уровня, используя доступные ресурсы персональных компьютеров. Когда кластер MOLA больше не требуется, компьютеры могут быть перезапущены в их первоначальное состояние. Оригинальность MOLA в том, что любой независимый от платформы компьютер, может быть добавлен к кластеру, не используется жесткий диск компьютера. Такая система подойдет для доступного проведения расчетов в сфере медицины[26].

BirgHPC является дистрибутивом для биоинформатики , построенной с использованием адаптированной версии make_pelican. Он добавляет MPICH2, а также набор инструментов для биоинформатики, занимающейся теоретическими вопросами хранения и передачи информации в биологических системах. Основными разделами биоинформатики являются компьютерная геномика, решающая проблемы расшифровки генетических "текстов", хранящихся в последовательностях нуклеотидов ДНК (РНК), и метабономика, исследующая организацию метаболизма клетки и его управления со стороны генома [10], [27].

Cluster by Night еще один live-CD подход к созданию кластера для работы MPI. Он работает внутри существующей сети - не как в случае с PelicanHPC. Это представляется очень хорошим решением, когда невозможно настроить частную сеть.

Так же существует ряд других дистрибутивов основанных на дистрибутиве PelicanHPC, или подобных ему.

Сам проект PelicanHPC предоставляет средства для его модификации пользователями под их конкретные нужды, что позволяет создать свою версию дистрибутива развёртывания кластера в локальной сети.

3.2. Недостатки пакета PelicanHPC и возможные пути их преодоления

Система Пеликан отлично подходит для развертывания учебного кластера на базе компьютерного класса, однако она имеет некоторые недостатки, например, отсутствие полноценного менеджера задач, невозможность сохранения настроек системы стандартными средствами, отсутствие многопользовательского режима. Все недостатки можно устранить, произведя модификацию стандартного варианта пакета PelicanHPC.

Отсутствие менеджера задач ограничивает вычислительный кластер в области его применения и делает кластер не удобным в использовании для учебных целей. Избавиться от этого недостатка можно, инсталлируя менеджер задач в систему PelicanHPC. Это позволит пользователю запускать задачи и быть уверенным, что выполнению конкретной задачи на кластере не помешают выполнения задач других пользователей. Также такой менеджер позволит грамотно распределять ресурсы вычислительного кластера. Если для выполнения конкретной задачи требуется, к примеру, 2 вычислительных узла, то менеджер выделит их из системы, не задействуя другие узлы, которые могут быть использованы для расчета других задач, запущенных параллельно с этой.

По умолчанию в PelicanHPC все данные находятся в /home/user на виртуальном диске, расположенным в оперативной памяти и после выключения все будет потеряно. Необходимо сохранять свою работу в период между сессиями, если необходимость в дальнейшем использовать ее. Есть много вариантов, таких как монтирование жесткого диска,

использование USB-устройства, и т.д. Если настроено соединение с интернетом, то возможно отправить работу себе самому.

Если используется PelicanHPC для серьезной работы, очень удобно смонтировать внешний накопитель для использования в качестве /home директории, так что ваша работа будет сохранена в период между сессиями без принятия каких-либо специальных мер. При загрузке интерфейс узла, у вас есть возможность выбора накопителя для использования. При использовании раздела с другими данными на нем, необходимо убедиться, что сделана его резервная копия, прежде чем использовать его с PelicanHPC. Существует также возможность автоматического монтирования тома. Это лучшее решение для пользователей, которые хотят использовать PelicanHPC на долгосрочной основе.

3.3. Возможность создания модифицированных версий пакета PelicanHPC

PelicanHPC сделан на основе Debian GNU/Linux при помощи Debian Live системы. Создание дистрибутива PelicanHPC производится путем запуска одного скрипта make_pelican-4.1-xfce[28]. Запуск необходимо производить в операционной системе Debian GNU/Linux, и иметь подключение к интернету. Выполнение скрипта запускает процедуру загрузки необходимых файлов. Результатом выполнения скрипта является iso образ пакета PelicanHPC. Для получения своей версии продукта, необходимо модифицировать этот скрипт и/или вспомогательные файлы, которые этот скрипт использует. Например, если необходимо добавить пакеты, которые нужны для работы в модифицированной версии PelicanHPC, то надо вносить изменения в файл pelicanhpc.list.xfce Эти пакеты загрузятся из репозитария Debian при создании версии PelicanHPC, и будут проинсталлированы как в образ live-CD, который запускается на головном узле, так и в образы операционной системы, загружаемые по сети на вычислительные узлы.

3.4. Выбор и установка менеджера задач

Вычислительный кластер на основе компьютерного класса, для полноценного использования в учебном процессе, требует наличия менеджера задач. Чтобы выбрать такой менеджер, который бы удовлетворял поставленным задачам для использования кластера в учебных целях, рассмотрим основные функции, которые исполняют менеджеры задач: выделение ресурсов задаче, старт и наблюдение за процессами на вычислительных узлах, ведение очереди.

Одним из менеджеров, решающих подобные задачи является TORQUE (англ. Terascale Open-Source Resource and QUEue Manager) — менеджер распределенных ресурсов для вычислительных кластеров из машин под управлением Linux и других Unix-подобных операционных систем. Для менеджера существует более 1200 патчей и расширений, написанных крупнейшими организациями и лабораториями, среди которых US DOE, USC, PNLL и др., это позволяет достичь высокой степени масштабируемости и отказоустойчивости менеджера как системы.

Основная функция TORQUE — распределение задач среди доступных вычислительных ресурсов. TORQUE содержит собственный планировщик заданий, определяющий момент запуска задач.

Аналогом TORQUE являются система Cleo, а также другие версии Portable Batch System. Существует также сторонний планировщик заданий Maui, который обладает значительно большей функциональностью по сравнению со стандартным, и, поэтому, часто используется совместно с TORQUE. Оригинальная версия TORQUE предназначена только для Unix-подобных систем, однако, существует порт для Cygwin, позволяющий использовать TORQUE под Windows [22].

Другой менеджер задач, который можно использовать, SLURM – это высокомасштабируемый отказоустойчивый менеджер кластеров и планировщик заданий. SLURM поддерживает очередь ожидающих заданий и управляет общей загрузкой ресурсов в процессе выполнения работы. Также SLURM управляет доступными вычислительными узлами (как функция

потребности в ресурсах). В дополнение к мониторингу параллельных заданий вплоть до их завершения SLURM распределяет нагрузку по выделенным узлам.

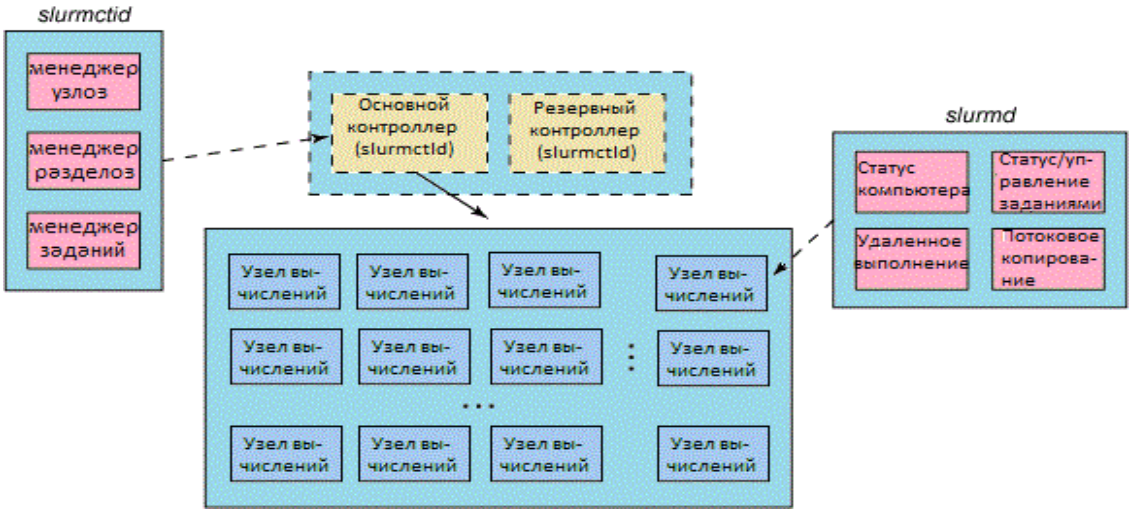
В общем виде SLURM - это надежный, переносимый, масштабируемый на большое количество узлов, отказоустойчивый и, что самое важное, открытый менеджер кластеров (ориентированный больше на необходимые функции, чем на обеспечение дополнительных возможностей). В настоящее время SLURM является лидером среди менеджеров ресурсов и используется на многих самых мощных и используемых суперкомпьютерах.

Менеджер SLURM используется приблизительно на 40% систем из Top500 списка самых производительных суперкомпьютеров мира и обладает неплохими показателями для менеджера очередей, поэтому использование SLURM на временном вычислительном кластере послужит хорошей реализацией управления ресурсами кластера [23].

3.4.1. Менеджер задач SLURM и установка его в PelicanHPC

В SLURM реализована типичная архитектура управления кластером. Верхний уровень управления – это резервированная пара контроллеров кластера. Эти контроллеры управляют вычислительным кластером и содержат демон управления под названием slurmctld. Демон slurmctld следит за вычислительными ресурсами, но, что более важно, он занимается распределением этих ресурсов между разными задачами.

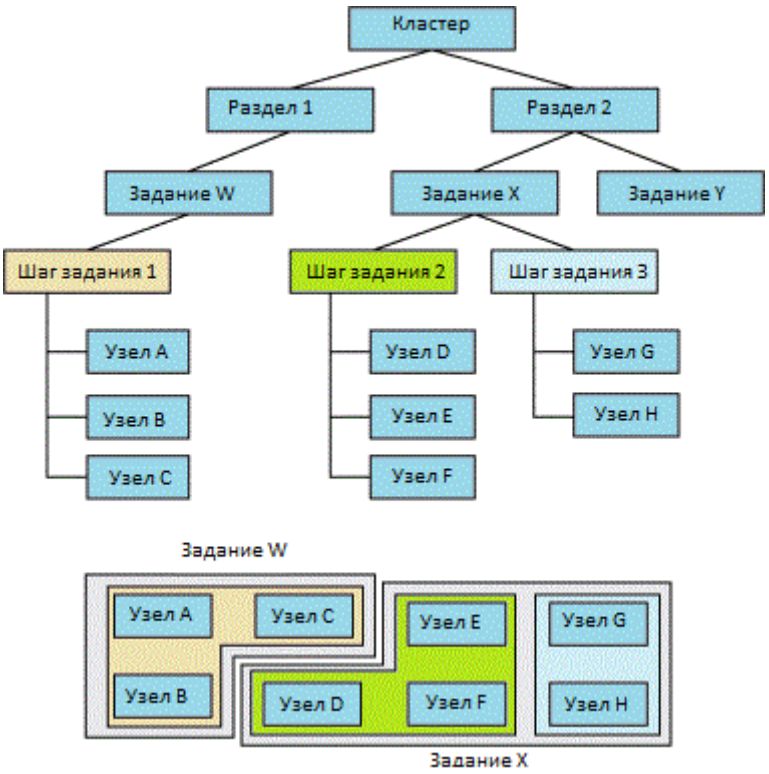
Каждый вычислительный узел содержит демон под названием slurmd. Демон slurmd управляет узлом, в котором он запущен, в том числе занимается мониторингом выполняющихся на узле заданий, получением заданий от контроллера и их распределением по ядрам внутри узла. Кроме того, slurmd останавливает выполнение заданий по запросу контроллера.



Высокоуровневое представление архитектуры SLURM

Набор узлов можно объединить в логическую группу, называемую разделом и включающую в себя очередь входящих заданий. Для разделов можно задавать ограничения, например, указывать пользователей, которые могут их использовать, размер задания или предельный срок обработки. Другая особенность раздела заключается в том, что пользователю на определенный период времени выделяется часть узлов для выполнения работы, которая называется заданием. Каждое задание содержит один или несколько шагов, представляющих собой наборы задач, выполняющихся на выделенных узлах.

Эта иерархия, более подробно показывающая разделение ресурсов на разделы в SLURM, изображена на рисунке.



Процесс инсталляции SLURM зависит от конкретного окружения Linux, но может сводиться к простой работе с менеджером пакетов. SLURM доступен в виде готового пакета, который просто установить и настроить. В дистрибутиве Debian имя пакета - slurm-llnl. Перед инсталляцией необходимо обновить систему, воспользовавшись командами aptitude update и aptitude safe-upgrade. Для инсталляции можно использовать пакет Advanced Packaging Tool (APT): apt-get install slurm-llnl.

После установки пакета SLURM его необходимо настроить. В первую очередь необходимо создать файл конфигурации: /etc/slurm-llnl/slurm.conf Для автоматизации создания этого файла, при установке пакета, устанавливается конфигуратор: /usr/share/doc/slurmctld/slurm-wim-configurator.html, используя который можно получить текст для файла /etc/slurm-llnl/slurm.conf

Для случая простого конфигурирования SLURM, можно воспользоваться в основном значениями предлагаемыми по умолчанию, изменив только некоторые из них:

1. Правильно указать имя головной машины, а также имена вычислительных узлов.

2. В разделе Auythentication and Security в пункте Select one value for AuthType нужно выбрать первый вариант: None: No auntification. Во втором пункте Select one value for CryptoType нужно выбрать вторую строку: OpenSSL: OpenSSL. Так же в этом разделе нужно указать расположение набора ключей сертификации, например, /usr/local/etc/slurm.key и /usr/local/etc/slurm.cert, которые надо обязательно указать при создании этих ключей.

Нажав кнопку Submit внизу страницы конфигуратора, получаем текст для файла /etc/slurm-llnl/slurm.conf. Далее создаём этот файл и копируем в него этот текст.

К сожалению, конфигуратор для SLURM версии 14.3, который находится в репозитарии Debian версии 8.6.0 и 8.8.0, содержит ошибки. Пакет SLURM при установке создаёт ряд подкаталогов с именем slurm-llnl, а конфигуратор записывая маршруты к необходимым файлам использует имя slurm. Ввиду этого, необходимо вручную в файле /etc/slurm-llnl/slurm.conf, все имена подкаталогов "slurm" поменять на "slurm-llnl".

Далее необходимо создать набор ключей сертификации командами openssl:

- openssl genrsa -out /usr/local/etc/slurm.key 1024 (Команда генерации ключа)
- openssl rsa -in /usr/local/etc/slurm.key -pubout -out /usr/local/etc/slurm.cert (команда для сертификации)

Чтобы изменения вступили в силу, нужно перезагрузить службы SLURM командами /etc/init.d/slurmd.d restart и /etc/init.d/slurmctld restart

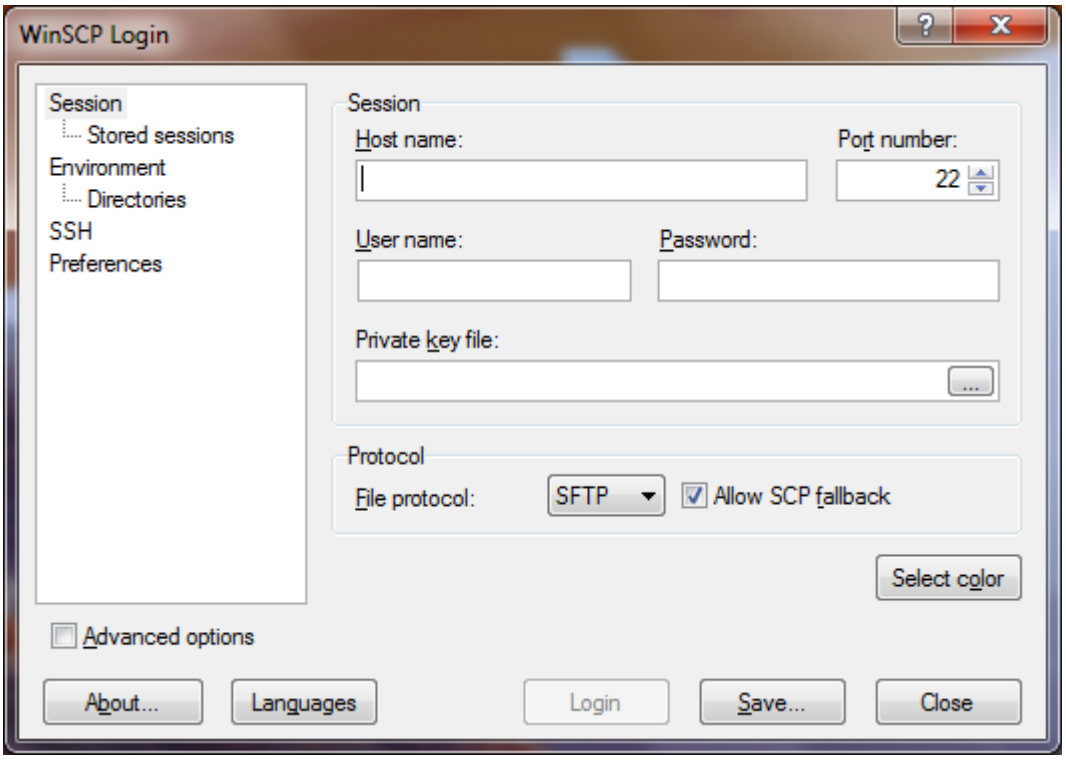
Чтобы убедиться в том, что SLURM запущен, можно использовать стандартную команду sinfo, которая возвращает информацию об узлах и разделах SLURM.

Описанный выше процесс установки SLURM необходимо проделать в той версии дистрибутива Debian, в которой будет запускаться скрипт для создания модифицированного Пеликана. Целью этой установки является получение правильно настроенного конфигурационного файла slurm.conf для дальнейшего использования его в модифицированном Пеликане [23].

3.5. Пути сохранения настроек системы и данных учащихся

Если вы используете PelicanHPC для серьёзной работы, очень удобно смонтировать внешний накопитель для использования в качестве /home директории, так что ваша работа будет сохранена в период между сессиями без принятия каких-либо специальных мер. Существует также возможность автоматического монтирования тома, который имеет специальное название. Это лучшее решение для пользователей, которые хотят использовать PelicanHPC на долгосрочной основе.

При работе с временным вычислительным кластером возникает потребность в переносе и сохранении данных пользователей. Поскольку Pelican является Live-CD системой, работа с файлами происходит в оперативной памяти, данные будут удаляться при каждом перезапуске кластера. Решением может выступить приложение WinCSP.



Главный экран программы WinSCP

WinSCP — свободный графический клиент протоколов SFTP и SCP, предназначенный для Windows. Распространяется по лицензии GNU GPL. Обеспечивает защищённое копирование файлов между компьютером и серверами, поддерживающими эти протоколы. Позволяет копировать и переносить данные из одной операционной системы с другую.

После того, как подключение к нужной машине осуществилось, необходимо ввести логин и пароль системы. В приложении можно работать с каталогами и файлами в графическом режиме. Графическая оболочка и функциональность приложения WinSCP представляет собой некоторый аналог всем известного менеджера FAR. Переносить файлы и данные таким способом довольно просто [16].

3.6. Создание модифицированного пакета - PelicanHPC-EDU

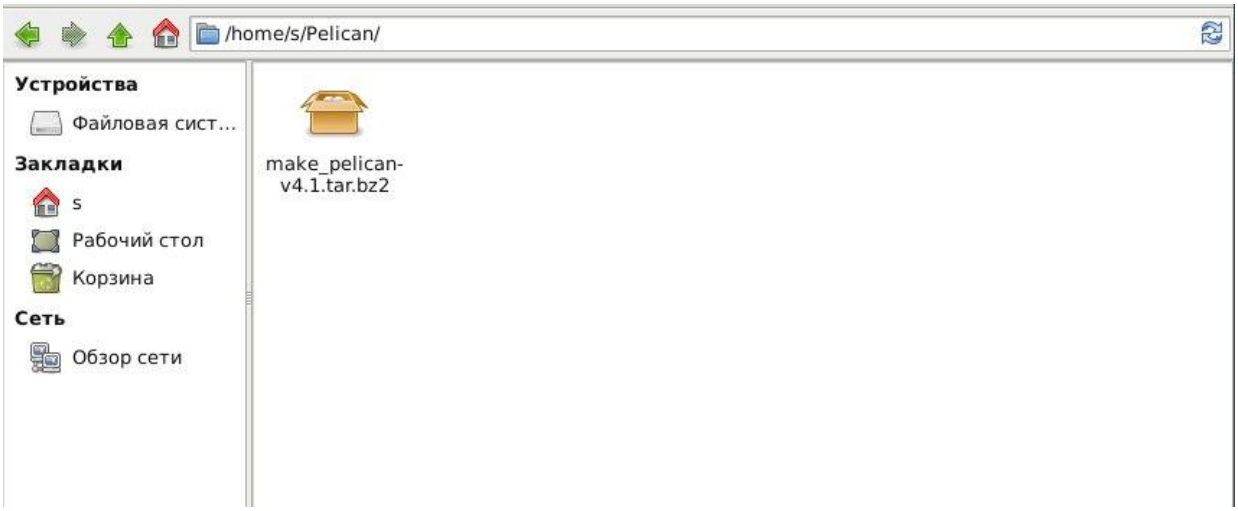
Все манипуляции и модификации, производимые над временным вычислительным кластером необходимо проделывать каждый раз после перезапуска кластера.

Создание собственной версии системы развертывания кластера на основе PelicanHPC позволить решить эту проблему, поскольку все необходимые пакеты и службы будут установлены и загружаться вместе с включением кластера.

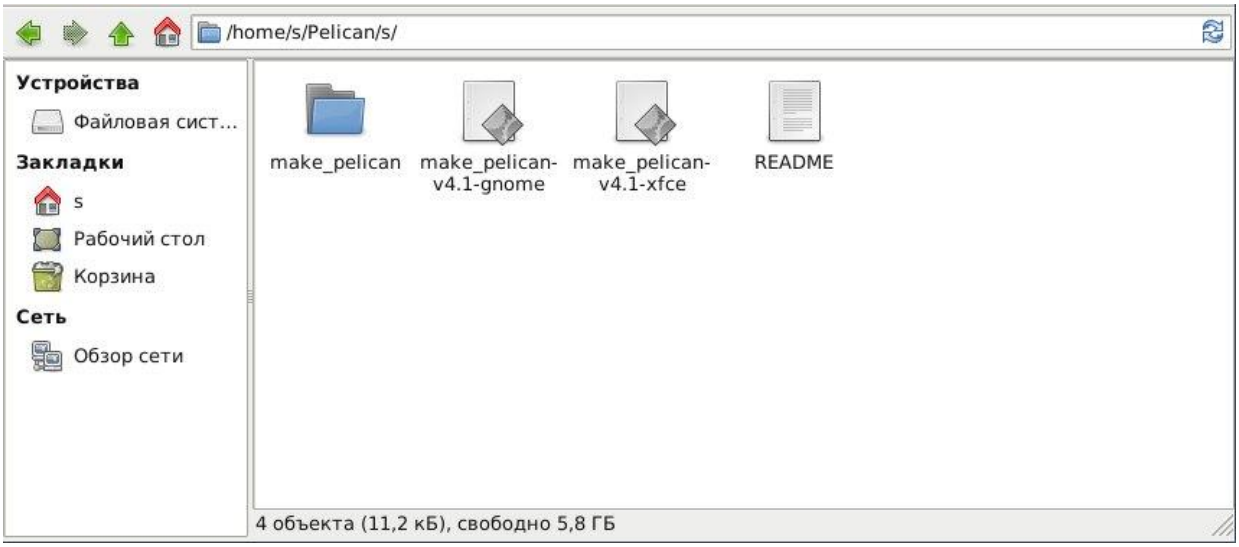
PelicanHPC 4.1 создается на базе Debian версии 8.6.0 Jessie запуском скрипта make_pelican-v4.1xfce [17]. Запуск скрипта можно также производить в операционной системе Linux Debian 8.8.0 Jessie. Все действия можно проводить с помощью виртуальной машины.

Добавляя необходимые изменения в скрипт, мы получим собственный образ системы развертывания на базе PelicanHPC [30].

Скачать скрипт можно с официального сайта PelicanHPC в разделе загрузки.



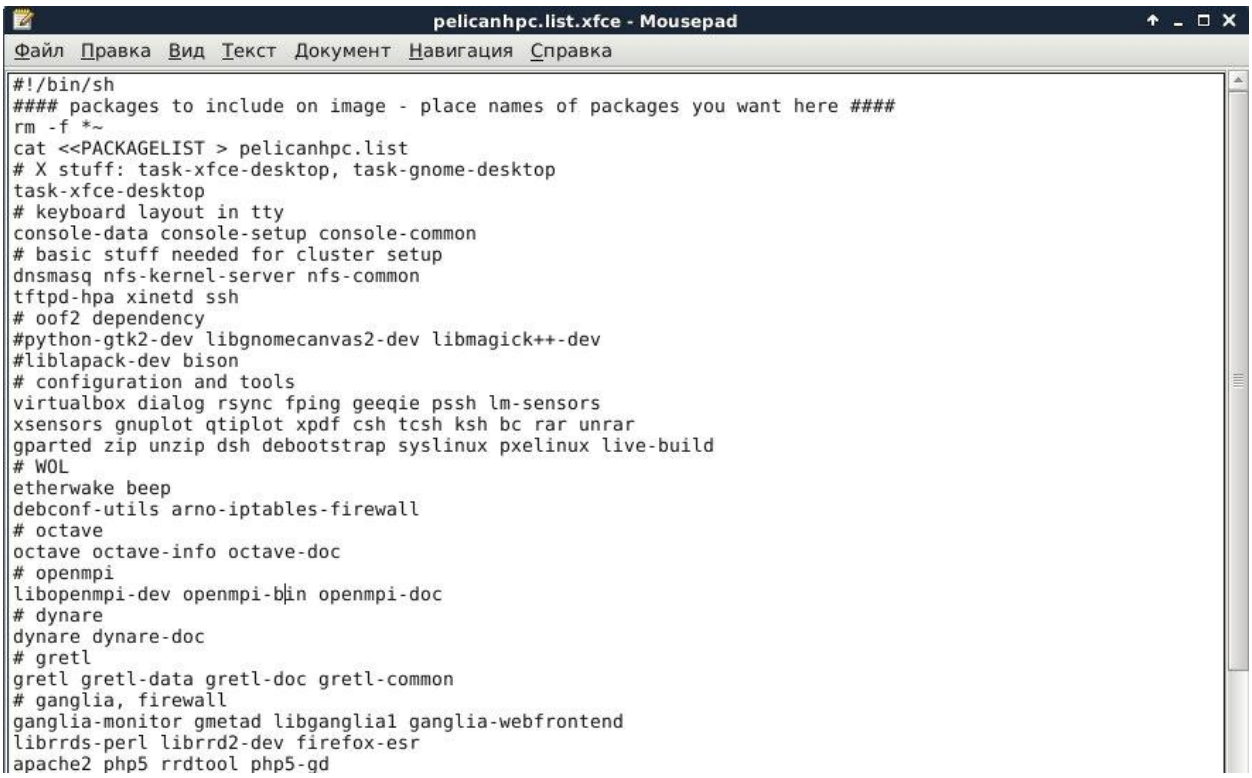
Так выглядит загруженный с официального сайта архив с нужным скриптом.



Извлеченный из архива набор файлов в отдельную папку

Для того чтобы добавить в модифицированную версию необходимый софт, нужно прописать его в скрипте, при создании пакета он скачается автоматически.

В папке make_pelican лежит файл pelicanhpc.list.xfce, который нужно открыть редактором Mousepad.



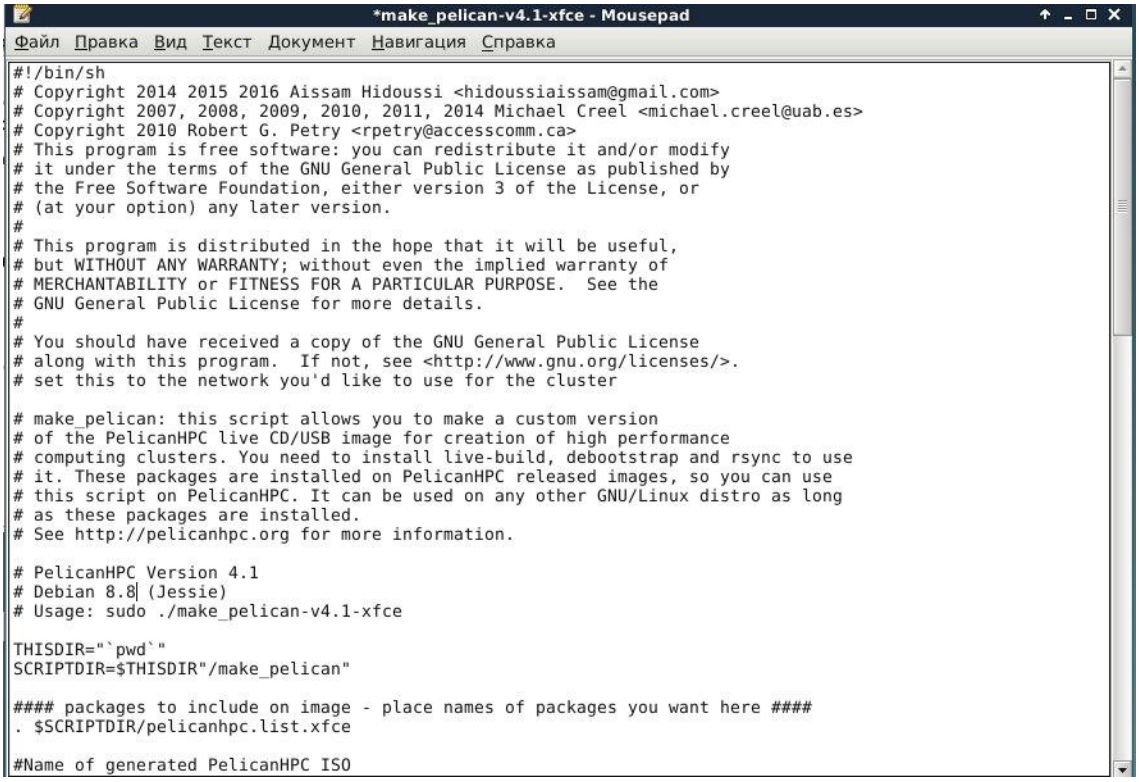
В эту часть добавляются имена необходимых программ, которые мы хотим видеть в новом образе:

```
# octave
octave octave-info octave-doc
# openmpi
libopenmpi-dev openmpi-bin openmpi-doc
# dynare
dynare dynare-doc
# gretl
gretl gretl-data gretl-doc gretl-common
# ganglia, firewall
ganglia-monitor gmetad libganglia1 ganglia-webfrontend
librrds-perl librrd2-dev firefox-esr
apache2 php5 rrdtool php5-gd
# Python
python-scipy python-matplotlib python-numpy
ipython python-mpmath python-gmpy python-mpi4py
```

Для удобства обращения, а также для использования кластера в учебных целях обязательно нужны файловый менеджер Midnight commander и менеджер задач SLURM, поэтому дописываем коды для установки этих пакетов: mc, slurm-llnl. Отредактированный файл скрипа сохраняется.

```
# octave
octave octave-info octave-doc
# slurm
slurm-llnl
# mc
mc
# openmpi
libopenmpi-dev openmpi-bin openmpi-doc
# dynare
dynare dynare-doc
# gretl
gretl gretl-data gretl-doc gretl-common
# ganglia, firewall
ganglia-monitor gmetad libganglia1 ganglia-webfrontend
librrds-perl librrd2-dev firefox-esr
apache2 php5 rrdtool php5-gd
# Python
python-scipy python-matplotlib python-numpy
ipython python-mpmath python-gmpy python-mpi4py
```

Чтобы изменить имя полученного в итоге образа редактируем файл make_pelican-v4.1-xfce.



Далее находим следующие строки в тексте скрипта:

```
#Name of generated PelicanHPC ISO
IMAGENAME="pelicanhpc-v4.1-xfce.iso"
```

Изменяем на имя, которое получит новый модифицированный пакет - PelicanHPC-EDU. Сохраняем и закрываем файл.

Перед запуском необходимо обновить систему, воспользовавшись командами aptitude update, aptitude safe-upgrade.

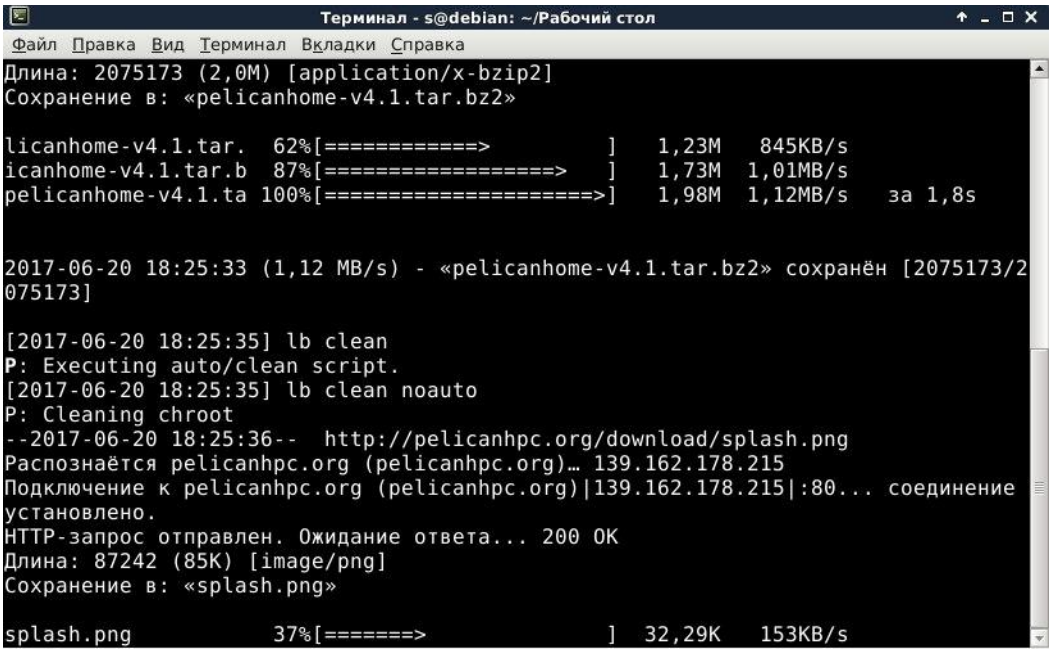
Теперь скрипт нужно запустить. В терминале находим папку с содержимым скрипта. Для этого можно воспользоваться командами: cd имя каталога - перейти на каталог вверх с заданным именем, cd . . – перейти на один каталог назад. Команда ls отображает содержимое каталога, в котором находимся:

```
root@debian:/home/s/Pelican/PelicanHPC# ls
make_pelican  make_pelican-v4.1-gnome  make_pelican-v4.1-xfce  README
root@debian:/home/s/Pelican/PelicanHPC#
```

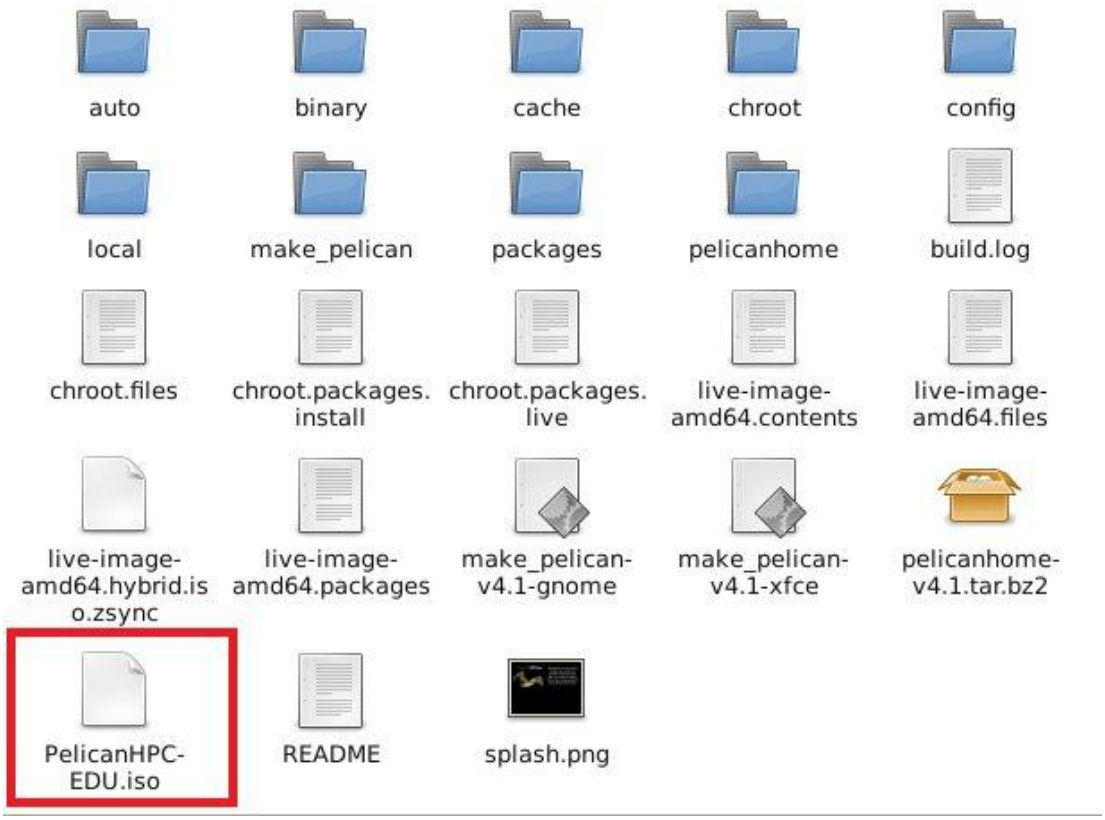
и запускаем скрипт командой ./make_pelican-v4.1-xfce

```
make_pelican  make_pelican-v4.1-gnome  make_pelican-v4.1-xfce  README
root@debian:/home/s/Pelican/PelicanHPC# ./make_pelican-v4.1-xfce
```

Начинается процесс выполнения скрипта, заключающийся в скачивании необходимых файлов из сети интернет и в последующей их установке.



Время выполнения зависит он производительности компьютера, скорости подключенной сети и может занять около часа. После завершения работы, получаем нужный образ диска



Полученный образ монтируем на USB-накопитель в качестве загрузочного диска PelicanHPC-EDU и разворачиваем кластер на основе этого диска. Как только кластер запущен, необходимо установить на всех узлах кластера конфигурационный файл SLURM, ключи openssl сертификатов, и перезапустить демоны SLURM.

Чтобы миновать процедуру ручной настройки SLURM на запущенном с live-USB кластере, был написан скрипт, который копирует конфигурационный файл SLURM на все узлы, создаёт ключи openssl сертификатов и копирует их в соответствующие поддиректории на все узлы кластера, и перезапускает на них демоны SLURM. Для этого предварительно копируются файл slurm.conf, созданный заранее как было описано выше, и необходимый скрипт из внешней сети, посредством утилиты WinSCP в /home/user/. Далее совершается вход на головной узел кластера при помощи программы PuTTY, и запускается данный скрипт, который для компьютерного класса из шести машин будет иметь следующий вид:

```
#!/bin/bash
```

```
cp /home/user/slurm.conf /etc/slurm-llnl/slurm.conf
openssl genrsa -out /usr/local/etc/slurm.key 1024
sudo openssl rsa -in /usr/local/etc/slurm.key -pubout -out /usr/local/etc/slurm.cert
ssh user@pel1 "sudo service slurmctld restart"
ssh user@pel1 "sudo service slurmd restart"

scp /usr/local/etc/slurm.key user@pel2:/usr/local/etc/slurm.key
scp /usr/local/etc/slurm.cert user@pel2:/usr/local/etc/slurm.cert
scp /etc/ssh/slurm.conf user@pel2:/etc/ssh/slurm.conf
ssh user@pel2 "sudo service slurmd restart"

scp /usr/local/etc/slurm.key user@pel3:/usr/local/etc/slurm.key
scp /usr/local/etc/slurm.cert user@pel3:/usr/local/etc/slurm.cert
scp /etc/ssh/slurm.conf user@pel3:/etc/ssh/slurm.conf
ssh user@pel3 "sudo service slurmd restart"

scp /usr/local/etc/slurm.key user@pel4:/usr/local/etc/slurm.key
scp /usr/local/etc/slurm.cert user@pel4:/usr/local/etc/slurm.cert
scp /etc/ssh/slurm.conf user@pel4:/etc/ssh/slurm.conf
ssh user@pel4 "sudo service slurmd restart"

scp /usr/local/etc/slurm.key user@pel5:/usr/local/etc/slurm.key
scp /usr/local/etc/slurm.cert user@pel5:/usr/local/etc/slurm.cert
scp /etc/ssh/slurm.conf user@pel5:/etc/ssh/slurm.conf
ssh user@pel5 "sudo service slurmd restart"

scp /usr/local/etc/slurm.key user@pel6:/usr/local/etc/slurm.key
scp /usr/local/etc/slurm.cert user@pel6:/usr/local/etc/slurm.cert
scp /etc/ssh/slurm.conf user@pel6:/etc/ssh/slurm.conf
ssh user@pel6 "sudo service slurmd restart"
```

Здесь pel1, pel2, и т.д. - имена узлов, назначаемые системой Пеликан.

3.7.Особенности проведения занятий на вычислительном кластере на основе PelicanHPC-EDU

Для проведения полноценных занятий, соревнований или олимпиад по параллельному программированию временный вычислительный кластер на основе PelicanHPC-EDU должен быть развернут до начала занятия в компьютерном классе.

Для этого преподавателю или организаторам олимпиады нужно заранее прийти в компьютерную аудиторию, сконфигурировать локальную сеть машин и загрузить кластер. После того, как головной узел будет загружен, все вычислительные узлы выполнили загрузку ОС по сети, нужно скопировать установочный скрипт и конфигурационный файл SLURM в файловую систему кластера в поддиректорию /home/user/. По протоколу SSH

из приложения PuTTY запустить установочный скрипт, который произведет на кластере настройку менеджера задач SLURM.

Этот класс будет использоваться только как аппаратная часть. Непосредственно сами занятия проводятся в другой аудитории, подключение к кластеру осуществляется удаленно по внешней сети.

Для замера времени выполнения программы, все учащиеся должны использовать функцию, например, «clock()», она показывает время работы программы.

Благодаря полноценно функционирующему менеджеру задач SLURM, PelicanHPC-EDU выполняет задачи пользователей строго в порядке очереди, распределяет ресурсы кластера по вычислительным узлам для наиболее эффективной работы задач.

В PelicanHPC-EDU установлен файловый менеджер Midnight Commander, что позволяет преподавателю и обучающимся легко обращаться к каталогам и файлам при работе с заданиями и запуском программ.

После выполнения программы преподаватель фиксирует результаты каждого учащегося, сравнивает все результаты, анализирует и делает выводы о том, кто справился лучше.

Заключение

Осмысление результатов проведенного нами исследования привело к ряду содержательных выводов.

В ходе проделанной исследовательской работы были проанализированы некоторые системы автоматического развёртывания вычислительных кластеров, в их числе такие пакеты как Rocks и PelicanHPC. Нами выбрана система PelicanHPC, предпочтение данной системы обусловлено наибольшей степенью соответствия для учебных целей. Освоив и обработав различный теоретический материал, были изучены базовые характеристики PelicanHPC, а также возможности и особенности этой системы.

Далее, стоит отметить, что система PelicanHPC была проанализирована и протестирована на практике, вследствие чего были разработаны методические рекомендации по установке системы и проведению занятий на временном вычислительном кластере.

Именно эти обстоятельства являются выявлением нами недостатков системы PelicanHPC, основными из которых являются:

1. Отсутствие полноценного менеджера задач;
2. Невозможность сохранения настроек и данных пользователей.

Рассмотрев и проанализировав перечисленные выше недостатки, мы обнаружили, что важным является факт их преодоления организационными методами. Однако такое решение не в полной мере раскрывает потенциал использования временного вычислительного кластера.

С учетом вышесказанного, для использования данной системы в учебном классе мы обозначили в своей работе пути модификации пакета PelicanHPC, ключевыми из которых являются установка полноценного менеджера задач SLURM, а также создание нового пакета PelicanHPC-EDU. Кроме того, в работу включены особенности проведения занятий на вычислительном кластере на основе PelicanHPC-EDU.

В заключение, произведя в процессе данного исследования модификацию пакета PelicanHPC, был создан модифицированный пакета PelicanHPC-EDU, в конечном результате который обладает следующими главными характеристиками:

1. Легко используется в компьютерной аудитории, без инсталляции на жесткий диск, что позволяет использовать класс в рамках стандартной образовательной программы;
2. Имеет набор программного обеспечения, необходимого для проведения учебных занятий на вычислительном кластере.

Библиографический список

1. Антонов, А.С. Параллельное программирование с использованием технологии MPI: Учебное пособие [Текст] / А.С. Антонов. – М.: МГУ, 2004. – 71 с.

2. Борисенко, А.А. Локальная сеть, просто как дважды два [ил.] / А.А. Борисенко. – М.: Эксмо, 2007. – 160 с.

3. Гергель, В.П., Стронгин, Р.Г. Основы параллельных вычислений для многопроцессорных вычислительных систем [Текст] / В.П. Гергель, Р.Г. Стронгин. – Нижний Новгород: ННГУ им. Н.И. Лобачевского, 2000. – 176 с.

4. Лацис, А.О. Как построить и использовать суперкомпьютер [Текст] / А.О. Лацис. – М.: Бестселлер, 2003. – 274 с.

5. Лабутин, Д.Ю. Система удаленного доступа к вычислительному кластеру (менеджер доступа): Высокопроизводительные параллельные [Текст] / Д.Ю. Лабутин // Материалы третьего Международного научно-практического семинара / Нижегордск. гос. унив-ет. Нижний Новгород. – 2003. – С. 184 – 187.

6. Новопашин, А.П., Сидоров, И.А., Скоров, В.В., Костромин, Р.О. Вычислительный кластер «Академик В.М. Матросов» Руководство пользователя [Текст] / А.П. Новопашин, И.А. Сидоров, В.В. Скоров, Р.О. Костромин. – М., 2017, – 24 с.

7. Стахиов, А. А. Сетевое администрирование Linux [Текст] / А.А. Стахиов. – Спб.: БХВ-Петербург, 2004. – 480 с.

8. Сапожников, А.П., Сапожникова, Т.Ф. Реинженирингговая технология распределенных вычислений в локальной сети [Текст] / А.П. Сапожников, Т.Ф. Сапожникова // Труды международной конференции "Распределенные вычисления и Грид-технологии в науке и образовании" / ОИЯИ. Дубна, 2004. – С. 183 – 190.

9. Самофалов, В.В., Коновалов, А.В., Шарф, С.В. Динамизм или статичность: поиск компромисса [Текст] / В.В. Самофалов, А.В.

Коновалов, С.В. Шарф // Труды Всероссийской научной конференции "Высокопроизводительные вычисления и их приложения" / М., 2000. С. 165 – 167.

10. Садовничий, В.А., Савин, Г.И. Суперкомпьютерные технологии в науке, образовании и промышленности [Текст] / В.А. Садовничий, Г.И. Савин. – М.: Изд-во Московского университета, 2009. – 232 с.

11. Чекмарев, Ю.В. Локальные вычислительные сети [Текст] / Ю.В. Чекмарев. – М.: ДМК-Пресс, 2008. – 200 с.

12. Шпаковский, Г.И., Верхотуров, А.Е., Серикова, Н.В. Руководство по работе на вычислительном кластере [Текст] / Г.И. Шпаковский, А.Е. Верхотуров, Н.В. Серикова. – М.: БГУ, 2013 – 172 с.

13. Daniel, J. Barrett, Richard, E. SSH, The Secure Shell: The Definitive Guide, 2nd Edition [Текст] / J. Daniel Barrett. – O'Reilly Media, 2005. – 660 с.

14. Michael, Creel. PelicanHPC: A Linux Cluster Distribution for MPI-based Parallel Computing [Текст] / Creel Michael. – Barcelona, 2012. – 9 с.

15. PuTTY Feedback and Bug Reporting [Электронный ресурс] URL: http://www.putty.org/ (дата обращения: 27.04.2017).

16. WinSCP .NET Assembly and COM Library [Электронный ресурс] URL: https://winscp.net/eng/docs/library (дата обращения: 07.03.2017).

17. LINUX.ORG.RU - Русская информация об ОС Linux [Электронный ресурс] URL: www.linux.org.ru (дата обращения: 12.10.2016). 18. ROCKS Cluster [Электронный ресурс] URL: http://cluster.linux-ekb.info/rocks.php (дата обращения: 08.11.2016).

19. Краткое введение в SSH [Электронный ресурс] URL: http://bezopasnik.org/unix/dok/FreeBSD/dok/89.htm (дата обращения: 14.04.2017).

20. Midnight Commander – консольный файловый менеджер для Linux [Электронный ресурс] URL: http://rus-linux.net/MyLDP/consol/midnight-commander.html (дата обращения: 20.02.2017).

21. Наиболее распространенные коммуникационные технологии [Электронный ресурс] URL:http://parallel.ru/computers/interconnects.html (дата обращения: 27.05.2017).

22. Вычислительный Linux-кластер на базе torque своими руками [Электронный ресурс] URL: http://www.openkazan.info/linux-torque (дата обращения: 27.05.2017).

23. Оптимизация управления ресурсами суперкомпьютеров с помощью SLURM [Электронный ресурс] URL:https://www.ibm.com/developerwork-s/ru/library/l-slurm (дата обращения: 13.05.2017).

24. Официальный сайт PelicanHPC [Электронный ресурс] URL: http://pelicanhpc.org (дата обращения: 11.05.2017).

25. Дистрибутивы на основе PelicanHPC [Электронный ресурс] URL: http://www.pelicanhpc.org/similar_distros.html (дата обращения: 11.05.2017).

26. Journal of Cheminformatics [Электронный ресурс] URL: https://jcheminf.springeropen.com/articles/10.1186/1758-2946-2-10 (дата обращения: 01.06.2017).

27. About birgHPC [Электронный ресурс] URL: http://birg1.fbb.utm.my/birghpc/ (дата обращения: 01.06.2017).

28. What is KestrelHPC? [Электронный ресурс] URL: http://kestrelhpc.sourceforge.net (дата обращения: 02.06.2017).

29. Iker Castaños Chavarri Automatic Control Group at the E.U.I.T.I. of Bilbao [Электронный ресурс] URL: http://www.ehu.eus/AC/ABC.htm (дата обращения: 04.06.2017).

30. Customization of PelicanHPC [Электронный ресурс] URL: http://www.pelicanhpc.org/customization.html (дата обращения: 09.06.2017).