

# PelicanHPC Tutorial

Январь 2011

[Michael Creel](#)

Университет Autònoma в Барселоне

Перевод на русский язык

Февраль 2011

[Богатырев Иван](#)

Россия ДВФУ Владивосток

Вы можете проверить наличие более новой версии этого документа ( оригинал ) по адресу <http://pelicanhpc.org/Tutorial/PelicanTutorial.html>

## Содержание

1. Введение
2. Первоначальная настройка
3. Примеры программ
4. Сохранение работы
5. Использование make\_pelican сценария

## Введение

[PelicanHPC](#) является быстрым решением ( ~ 5 минут, когда вы знаете, что делаете ) для создания кластера «высокопроизводительных вычислений» ( HPC - high performance computing ) для параллельного программирования с использованием MPI. Это инструкция дает общее описание того, что может делать PelicanHPC, объясняет, как использовать релизы CD образов, чтобы создать HPC кластер, и дать некоторые основные пример использования.

## Описание PelicanHPC

PelicanHPC - это дистрибутив операционной системы GNU/Linux, который работает в режиме «Live CD» ( он также может быть установлен USB устройство, также может быть загружен с раздела жесткого диска или может быть установлен на виртуальную машину ).

Чтобы использовать загрузку с CD-диска или USB-диска, нужно записать ISO-образ на соответствующий носитель. Компьютер, с которого загружается PelicanHPC называют терминалом или интерфейсом кластера. Этот компьютер используется для управления кластером. После того как PelicanHPC будет загружен появится приглашение и можно запустить сценарий "pelican\_setup". Это сценарий настраивает терминал кластера как сетевой сервер. После чего становится возможным загрузка копий PelicanHPC на других компьютерах ( вычислительных узлах кластера ). PelicanHPC конфигурирует кластер из терминала и вычислительных узлов кластера, так что в основе параллельного программирования может быть использован MPI.

"Live CD" такие как PelicanHPC не используют жесткие диски узлов ( кроме файла или

раздела подкачки LINUX, если таковой существует ). Поэтому он никак не затронет установленную операционную систему. Когда PelicanHPC будет выключен, то все компьютеры будут в своем первоначальном состоянии и могут быть загружены снова с той операционной системой, которая была на них установлена. Опционально PelicanHPC может быть настроен для использования загрузки с жесткого диска, чтобы сохранить конфигурацию после перезагрузки. Он может быть настроен на загрузку без локально вмешательства пользователя, благодаря удаленному доступу по SSH. Также есть возможность загрузки узлов с использованием wake-on-LAN ( примечание переводчика: wake-on-LAN - удаленное включение компьютера ( в локальной сети ) при помощи отправки на него специального сигнала ). С помощью этих, более продвинутых возможностей, можно запустить постоянный PelicanHPC кластер в режиме тонких клиентов ( примечание: с минимально возможным набором устройств, обычно исключая монитор, мышь, клавиатуру или даже жесткий диск ).

PelicanHPC сделан на основе [Debian GNU / Linux](#) в качестве своей базы, при помощи [Debian Live](#) системы. Это делается путем запуска одного скрипта с помощью команды "SH-V \* make\_pelican". Специальную версию PelicanHPC, например, содержащую дополнительные пакеты, легко можно сделать, изменив make\_pelican сценарий . Make\_pelican script и необходимые пакеты предоставляются на PelicanHPC, так что вы можете создать собственные образы, используя предоставленные возможности. Вы также можете запустить make\_pelican на любом другом GNU / Linux дистрибутиве, если вы установите live-cd и несколько других пакетов.

## Особенности

- Узел терминала может представлять собой реальный компьютер и загружаются с помощью компакт-диска или USB-устройства, или виртуальную машину, которая загружается с помощью файла образа диска. При этом последний вариант, PelicanHPC может быть использован в то же время, когда работает хостовая ОС, которая может быть любой из распространенных операционных систем.
- Вычислительные узлы, как правило, состоят из реальных компьютеров, для достижения максимальной производительности, но они 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](#) . Также имеет [mpi4py](#).

## Требования и ограничения

- Вычислительные узлы должны быть загружен по сети. Это вариант предлагают все современные сетевые карты поставляющиеся с материнскими платами, но должна быть включена опция в BIOS Setup. Включите ее и задайте ему более высокий приоритет, чем загрузка с жесткого диска или других источников. Если у вас есть

сетевая карта, не поддерживающая сетевой загрузки, можно обойти это, используя [ROM-O-Matic](#) . Другая вещь, чтобы быть в курсе, то что терминал PelicanHPC работает как DHCP сервер. Вы не должны использовать его в локальной сети с уже существующим DHCP-сервером, чтобы избежать возможные конфликты. Это может стать причиной проблем с сетевым администраторам. Плюс, ваши вычислительные узлы не загрузятся должным образом.

- 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.). make\_pelican может быть использован, чтобы сделать 32-битную версию, если это необходимо.
- На веб-сайте PelicanHPC перечислены некоторые другие подобные дистрибутивы, которые могут быть более подходящим для определенных целей.

## Лицензирование и ответственность

PelicanHPC является образом компакт-диска в результате выполнения скрипта (см. ниже). Сценарий имеет лицензию GPL v3. В результате образ компакт-диска содержит программное обеспечение от дистрибутива [Debian](#) GNU / Linux, а также несколько других источников, которые подлежат лицензии, выбранной авторами этого программного обеспечения.

Этот релиз CD PelicanHPC образа распространяется в надежде, что он будет полезен, но БЕЗ КАКИХ-ЛИБО ГАРАНТИЙ, даже без подразумеваемых гарантий КОММЕРЧЕСКОЙ ЦЕННОСТИ или ПРИГОДНОСТИ ДЛЯ ОПРЕДЕЛЕННОЙ ЦЕЛИ.

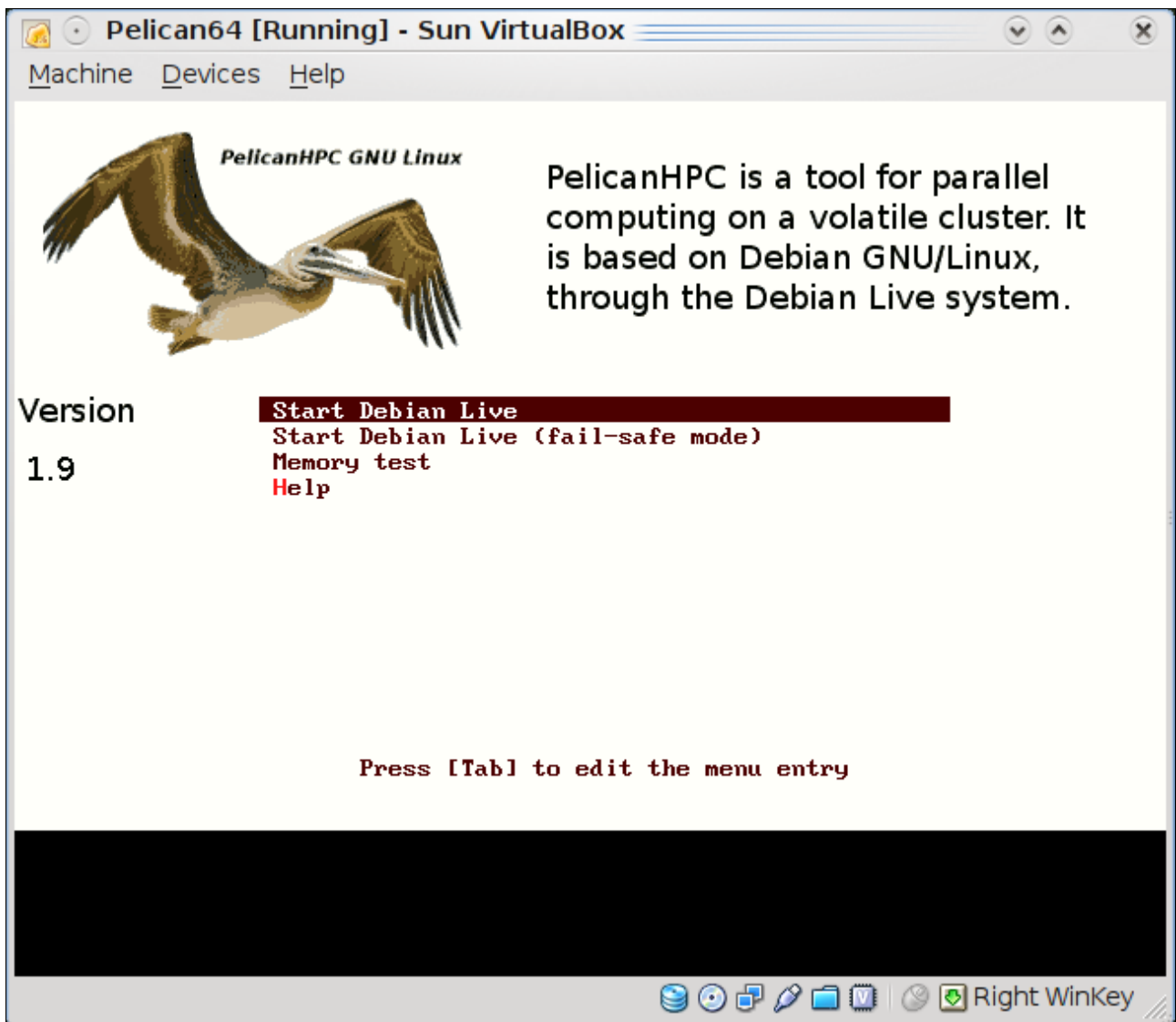
## Первоначальная настройка

Две основные команды для администрирования кластера: "pelican\_setup", чтобы настроить терминал как сервер, NFS export/home и т.д., и "pelican\_restart\_hpc", которая используется для добавления / удаления узлов после начальной установки. В остальной части этого учебника объясняется, как это работает.

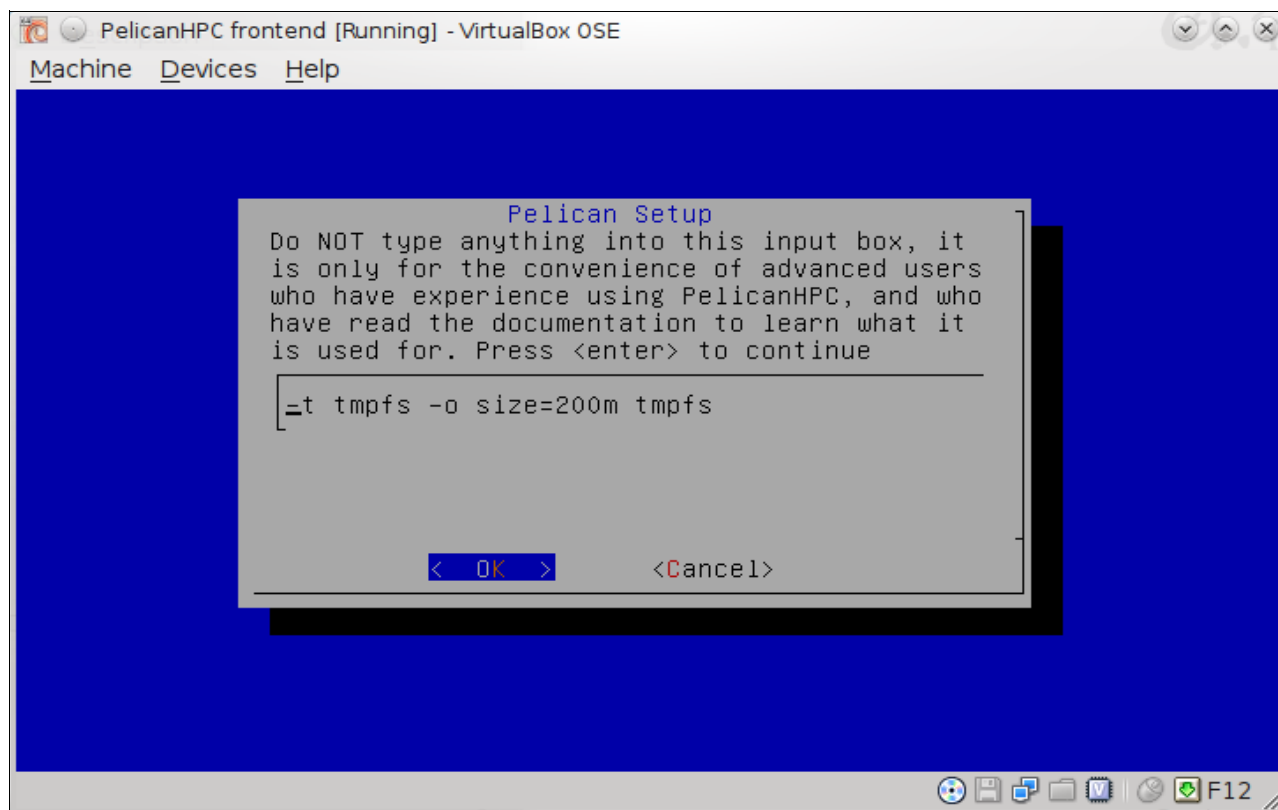
Интерфейс и все вычислительные узлы должны быть объединены в сеть. ВАЖНО: терминал кластера будет выступать в качестве сервера DHCP, так что не забудьте выделить независимую сеть, используемую для кластера отдельно от общей сети, чтобы избежать конфликтов с другими серверами DHCP. Если вы начнете раздавать IP адреса для компьютеров ваших коллег им это может не понравится. Если терминал-узел имеет несколько сетевых интерфейсов, вы можете использовать один для подключения к кластеру, а другой для подключения к Интернету.

Вставьте компакт-диск в компьютер ( или другое устройство, с которого будет производится загрузка, например USB-носитель ), который будет выполнять роль терминала кластера, и включите его. Убедитесь, что BIOS Setup позволяет загрузиться с компакт-диска. При загрузке, вы увидите нечто вроде следующего. Здесь, если вы нажмете <Tab>, у вас есть возможность выбрать варианты для установки клавиатуры, или специальные "Cheatcodes", чтобы сделать загрузочный компакт для проблемного оборудования. Например, я могу получить испанскую раскладку для клавиатуры, нажав <Tab> а затем добавить keyb=es" в конфигурации по умолчанию. Вы можете либо изучить варианты, выделив Справка и нажав <Enter>, или изменить параметры, нажав <Tab>, или просто нажмите <Ввод> для загрузки с

параметрами по умолчанию.



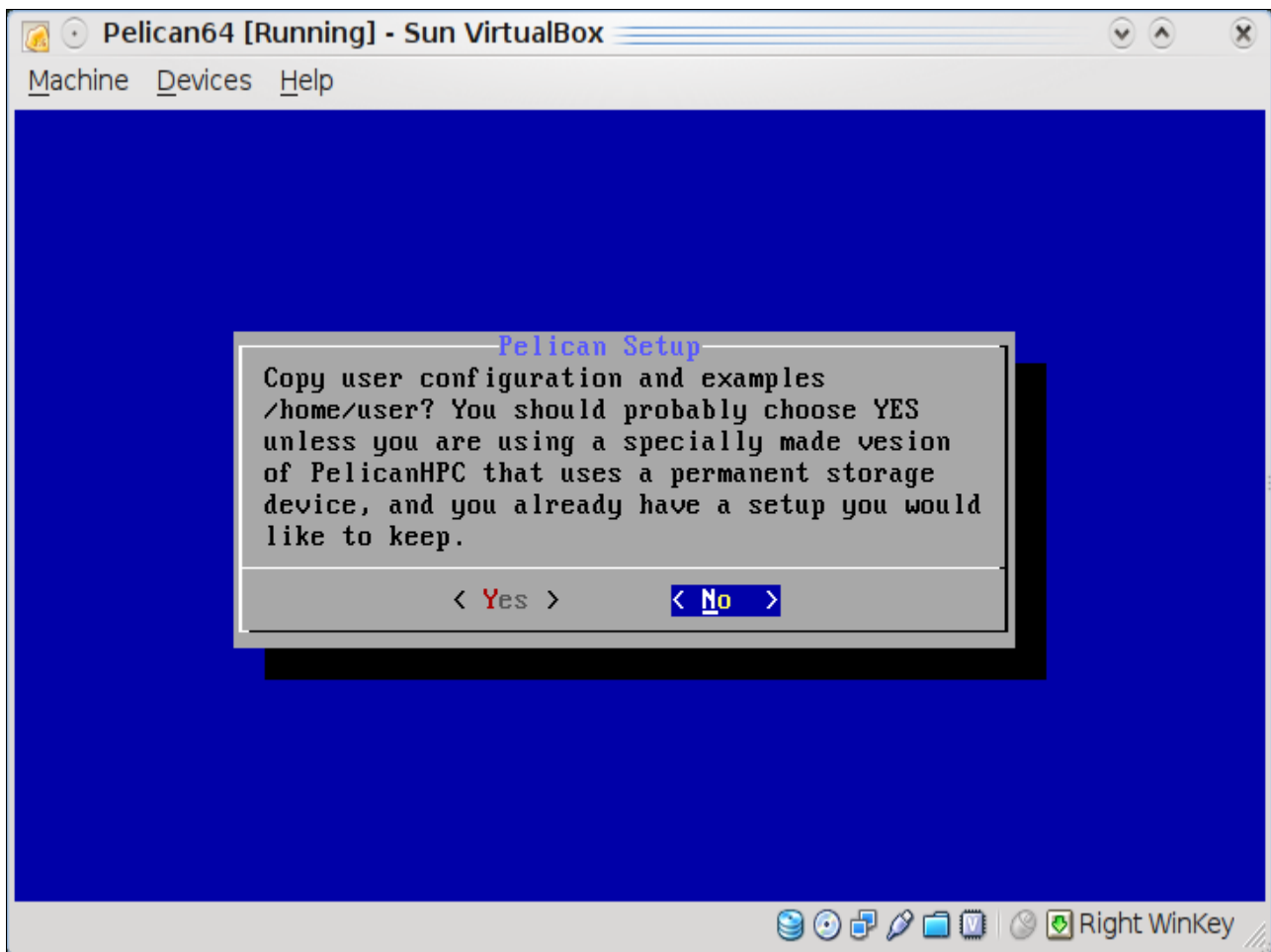
После загрузки в конечном итоге вы увидите:



На этом экране вы можете выбрать: использовать постоянное запоминающее устройство для /home ( домашнего каталога ) пользователя PelicanHPC. По умолчанию, если вы просто нажимаете <Enter>, жесткие диски не используются, и PelicanHPC не будет изменять какие-либо данные на компьютерах, используемых в кластере. ( Информация будет храниться в оперативной памяти ) Это безопасно и просто ( рекомендуется начинающим пользователям для ознакомления с работой системы ), но это имеет тот недостаток, что любая работа, у вас исчезает при завершении работы кластера. Для использования постоянного носителя информации, вы можете ввести имя устройства (раздела жесткого диска, USB-диск и т.д.), который отформатирован в ext2 или ext3 раздел, который будет смонтирован в /home. Например, можно заменить с "ram1" на "sda2" или "hdb5" (без кавычек). Если вы сделаете это, тогда каталог "user" будет создан в корневом каталоге указанного устройства, и будет использоваться в качестве домашнего каталога кластера ( /home ) пользователя (имя пользователя "user"). Если вы выключите узлы кластера, каталог не будет удален, и он может быть повторно использован при перезагрузке PelicanHPC. Если у вас есть какие-либо сомнения по этому поводу, просто нажмите <Enter>. Для экспериментов, вам не нужна эта функция. Эта функция представлена для удобства для продвинутых пользователей. Невозможно проверить эту функцию на всех возможных конфигурациях оборудования, так что **НЕТ НИКАКИХ ГАРАНТИЙ**, что она не станет причиной удаления данных с вашего жесткого диска ( или иного другого устройства ). Рекомендуется выполнить резервное копирование данных, прежде чем пытаться что-нибудь сделать.

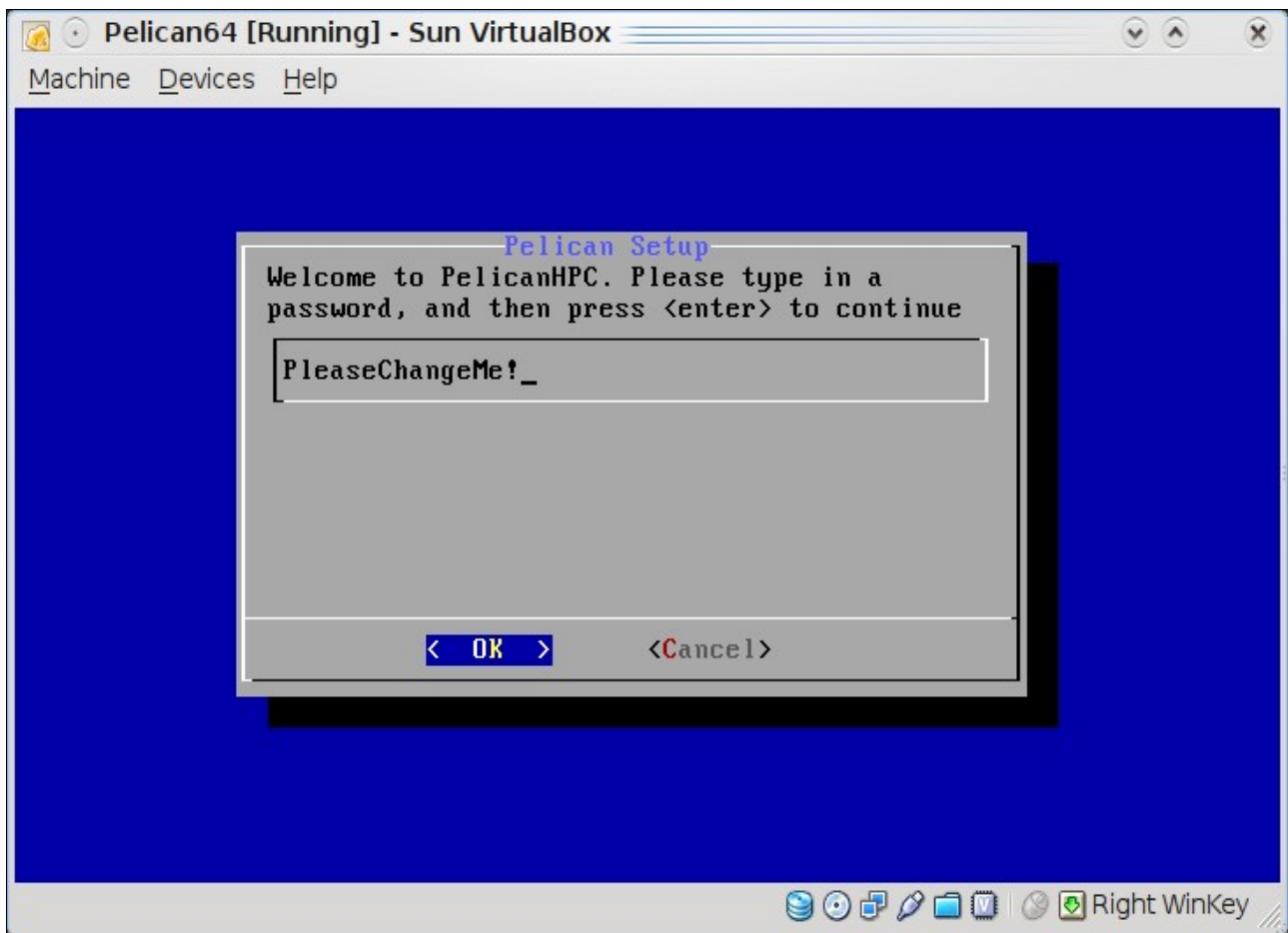
**ВНИМАНИЕ:** есть еще один способ использования постоянного хранения, что является довольно гибким решением. Это описано в файле /home/user/pelican\_config, которые вы можете увидеть, если загрузитесь с параметрами по умолчанию. Если это ваш первый опыт работы с PelicanHPC, я рекомендую делать загрузки по умолчанию, изучить pelican\_config, а затем выбрать вариант для постоянного хранения, то что вы найдете наиболее подходящим для себя.

Далее, вы увидите



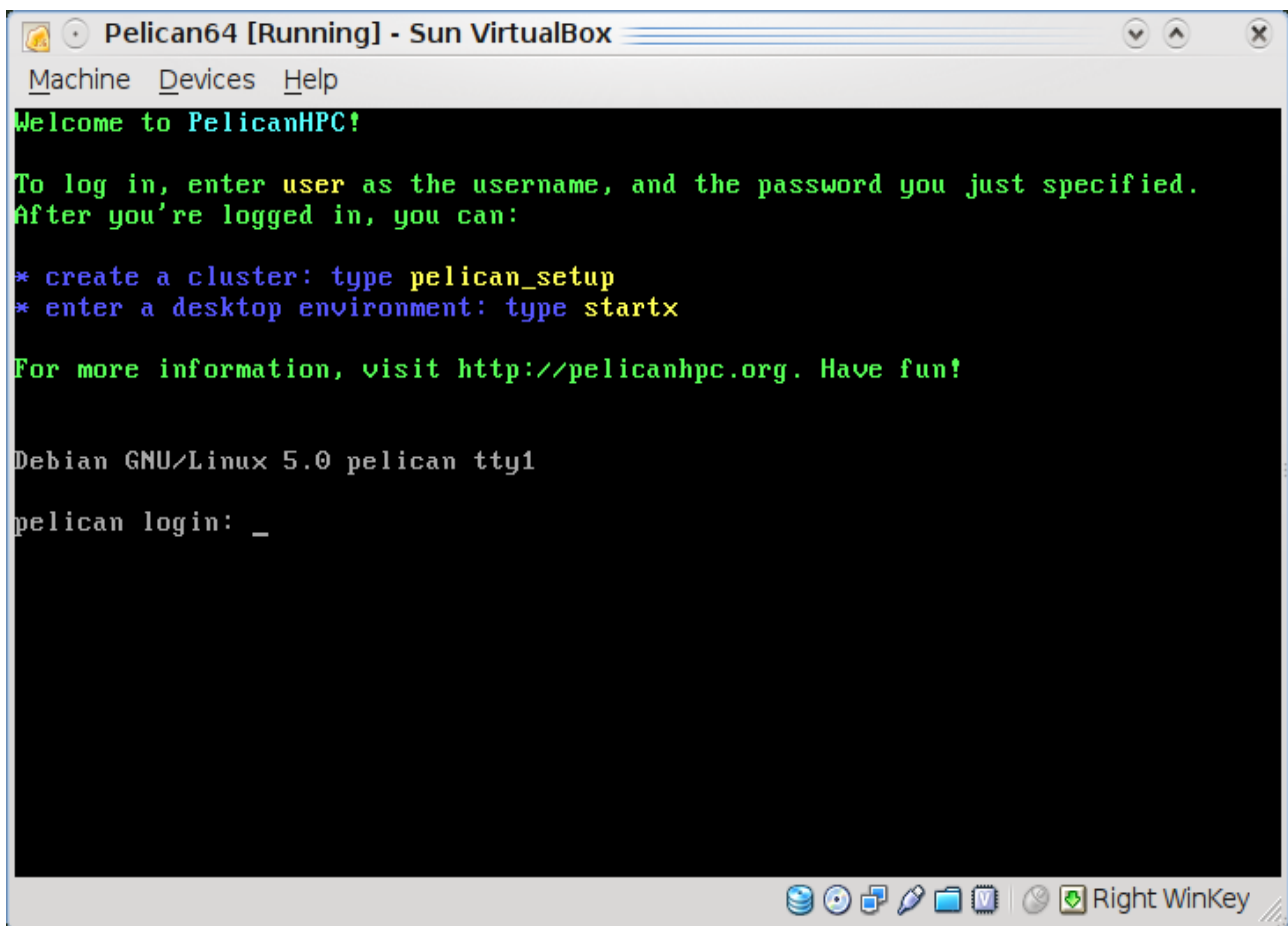
Здесь предлагается скопировать конфигурацию и примеры в /home/user . Если вы используете ram1 ( оперативную ) память ( предыдущий экран, параметр по умолчанию ), тогда выберете «да». Если вы используете устройство с постоянной памятью, тогда нужно скопировать эти файлы только один раз.

Далее, вам будет предложено изменить пароль по умолчанию:



Вы должны нажать Backspace для удаления пароля по умолчанию, а затем ввести новый пароль. Это будет пароль задан для пользователя "user" на терминале кластера и на всех вычислительных узлах, тоже.

Наконец, все загружается и появляется приглашение Логин:



```
Pelican64 [Running] - Sun VirtualBox
Machine  Devices  Help
Welcome to PelicanHPC!

To log in, enter user as the username, and the password you just specified.
After you're logged in, you can:

* create a cluster: type pelican_setup
* enter a desktop environment: type startx

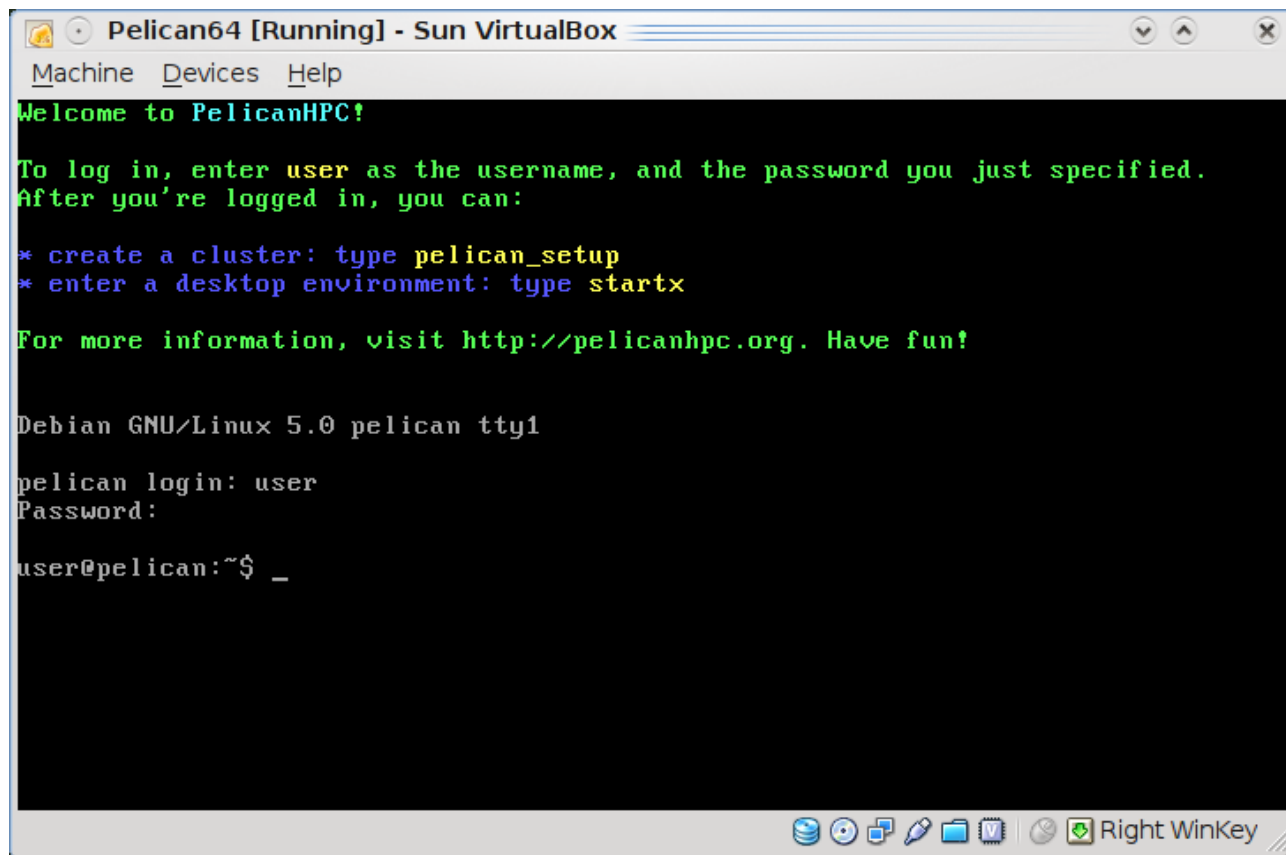
For more information, visit http://pelicanhpc.org. Have fun!

Debian GNU/Linux 5.0 pelican tty1
pelican login: _
```



Введите «user» и пароль, заданный вами ранее.

После это вы будете авторизованы



```
Pelican64 [Running] - Sun VirtualBox
Machine  Devices  Help
Welcome to PelicanHPC!

To log in, enter user as the username, and the password you just specified.
After you're logged in, you can:

* create a cluster: type pelican_setup
* enter a desktop environment: type startx

For more information, visit http://pelicanhpc.org. Have fun!

Debian GNU/Linux 5.0 pelican tty1

pelican login: user
Password:

user@pelican:~$ _
```

Обратите внимание, что вы можете загрузит Xfce графический интерфейс, набрав "startx". По умолчанию, PelicanHPC использует консольный режим, только чтобы избежать возможных проблем с необычным графическим оборудованием.

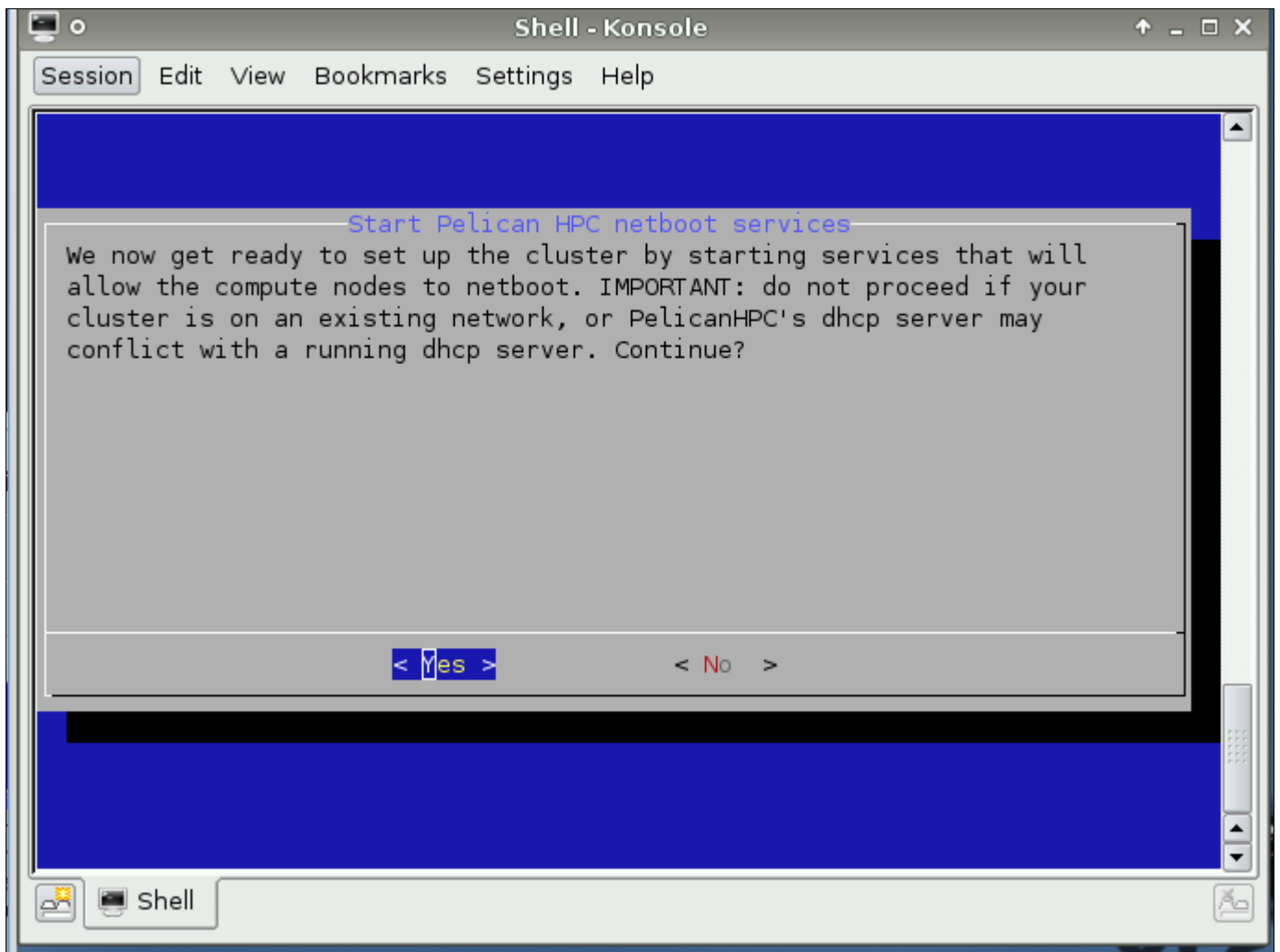
Чтобы создать кластер наберите "pelican\_setup". Вы можете сделать это из консоли, как в данной инструкции, так и в окружении Xfce путем открытия терминала.

Далее, мы увидим следующий выбор, предполагая, что у вас есть более чем 1 сетевой интерфейс:

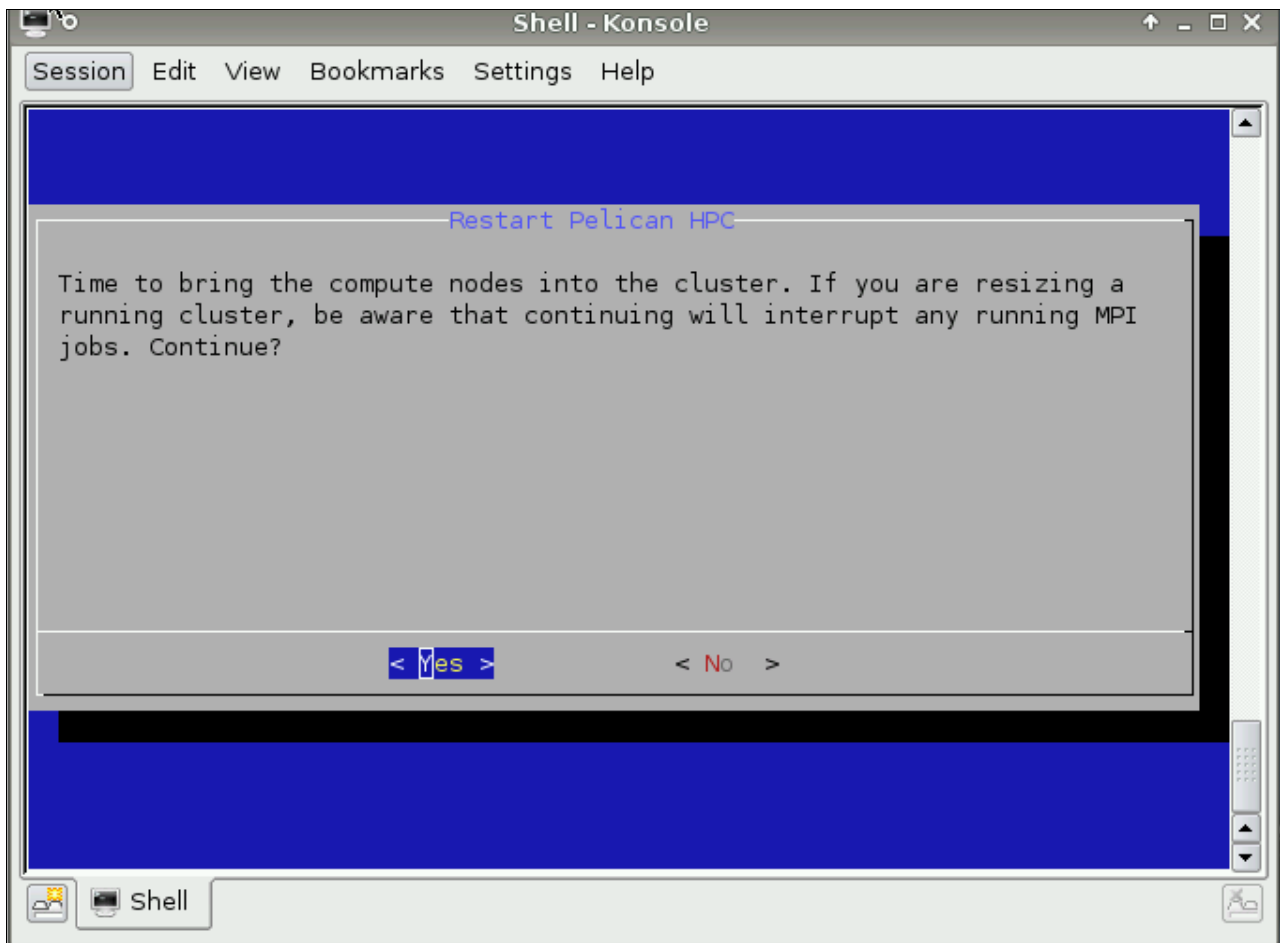


После выбора сетевого устройства, службы будут запущены. Пожалуйста, прочтите предупреждение на следующем скриншоте. Настройка сервера DHCP PelicanHPC может создать проблемы с администраторами сети, если вы делаете это на существующей сети с динамическим распределением арендой ip адресов . Вы должны убедиться, что сетевое устройство используемое для кластера изолировано от всех сетей, за исключением кластера!

Когда вы увидите следующий экран, выберите "да".



Потом вы должны увидеть:



Нажмите клавишу Enter и перейдите к загрузке на вычислительных узлах.

Когда вычислительный узел начнет загрузку по сети( перед этим нужно включить загрузку по сети в Bios Setup и установить самый высокий приоритет для загрузки ) увидите это:

```
Copyright (C) 1997-2000 Intel Corporation

CLIENT MAC ADDR: 00 0C 29 82 67 83  GUID: 564D2BDA-39FC-BD39-149F-957809826783
CLIENT IP: 10.11.12.3  MASK: 255.255.255.0  DHCP IP: 10.11.12.1

PXELINUX 3.61 Debian-2008-02-05  Copyright (C) 1994-2008 H. Peter Anvin
UNDI data segment at:  00099BF0
UNDI data segment size: 4D60
UNDI code segment at:  0009E950
UNDI code segment size: 0BBC
PXE entry point found (we hope) at 9E95:0106
My IP address seems to be 0A0B0C03 10.11.12.3
ip=10.11.12.3:10.11.12.1:0.0.0.0:255.255.255.0
TFTP prefix:
Trying to load: pxelinux.cfg/564d2bda-39fc-bd39-149f-957809826783
Trying to load: pxelinux.cfg/01-00-0c-29-82-67-83
Trying to load: pxelinux.cfg/0A0B0C03
Trying to load: pxelinux.cfg/0A0B0C00
Trying to load: pxelinux.cfg/0A0B0C
Trying to load: pxelinux.cfg/0A0B0
Trying to load: pxelinux.cfg/0A0B
Trying to load: pxelinux.cfg/0A0
Trying to load: pxelinux.cfg/0A
Trying to load: pxelinux.cfg/0
```

Когда вычислительный узел закончит загрузку, вы увидите это ( если у узлов кластера есть монитор )

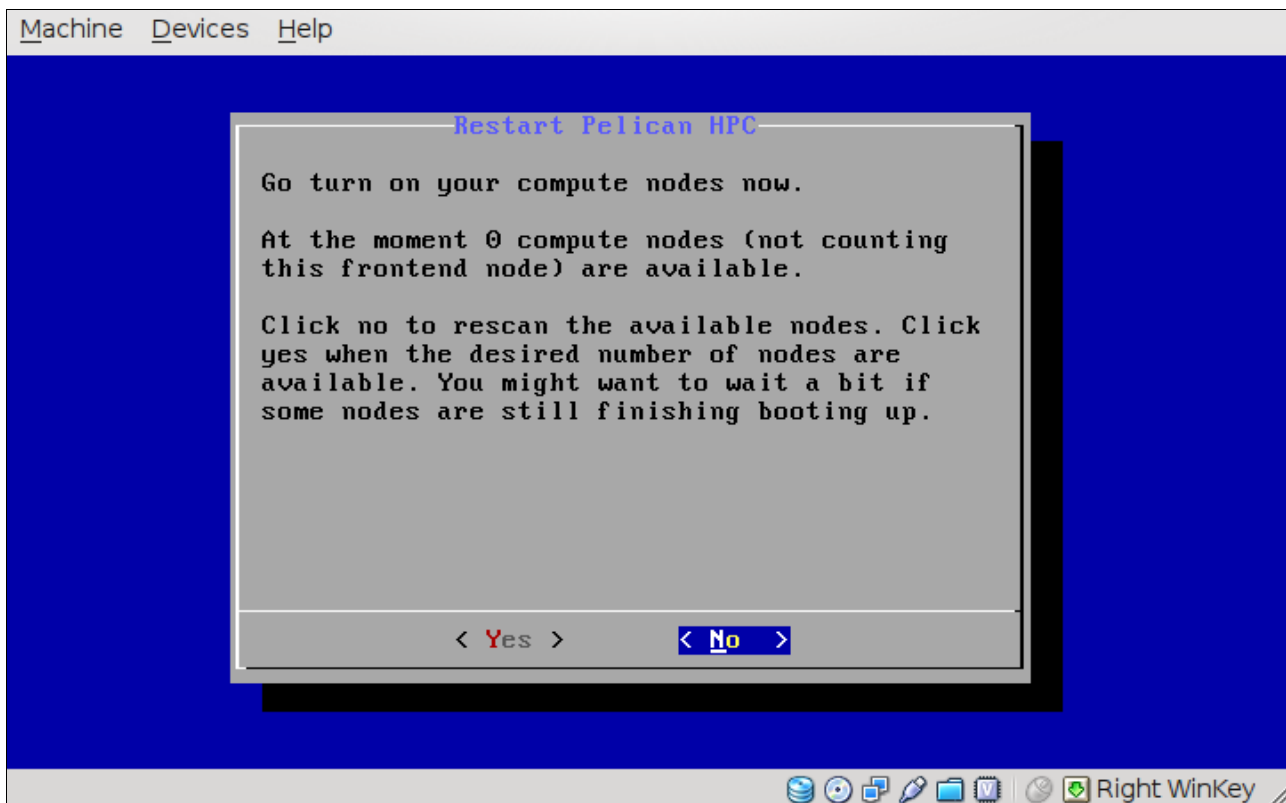
```
This is a PelicanHPC compute node. It is part of a cluster of computers that is
doing some REALLY important stuff.

Please don't try to use it, and DON'T TURN IT OFF!

THANKS!

Debian GNU/Linux lenny/sid debian tty1
debian login: _
```

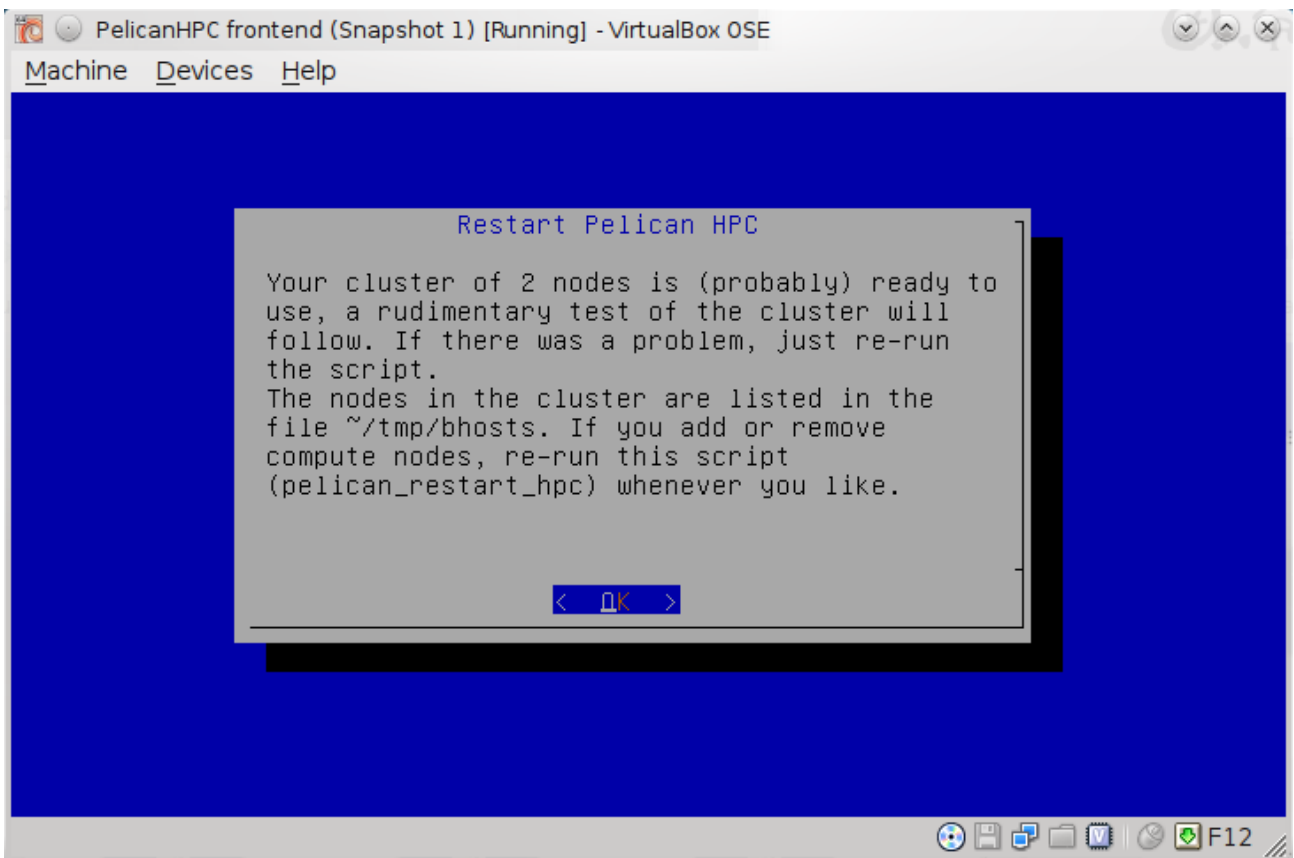
Вернитесь к терминалу кластера и вы увидите это ( количество найденных узлов ), нажмите «да», чтобы продолжить, нет, чтобы просканировать сеть снова.



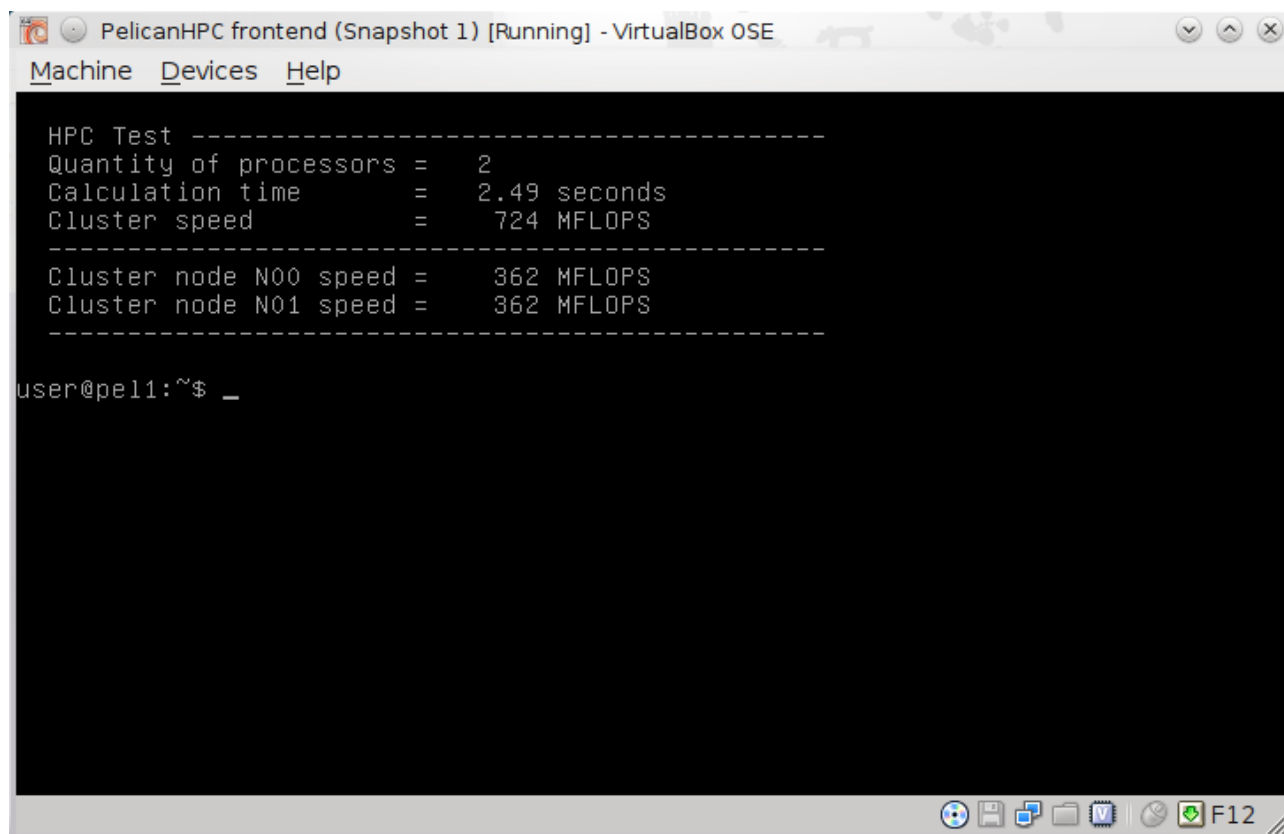
Когда хотя бы один узел закружится вы увидите число узлов в сети:



Нажимайте "нет", пока все ваши вычислительные узлы не загрузятся или не будут найдены. После нажатия кнопки Да, вы увидите нечто вроде следующего, в зависимости от количества узлов:



Наконец, запустится быстрый тест производительности кластера. Вы должны увидеть что-то вроде следующего:

A screenshot of a terminal window titled "PelicanHPC frontend (Snapshot 1) [Running] - VirtualBox OSE". The terminal displays the output of an HPC test. The output is as follows:

```
HPC Test -----
Quantity of processors = 2
Calculation time      = 2.49 seconds
Cluster speed         = 724 MFLOPS
-----
Cluster node N00 speed = 362 MFLOPS
Cluster node N01 speed = 362 MFLOPS
-----
user@pel1:~$ _
```

The terminal window has a menu bar with "Machine", "Devices", and "Help". At the bottom, there is a taskbar with various icons and the text "F12".

Хорошо, что кластер будет готов к использованию. Некоторые другие советы:

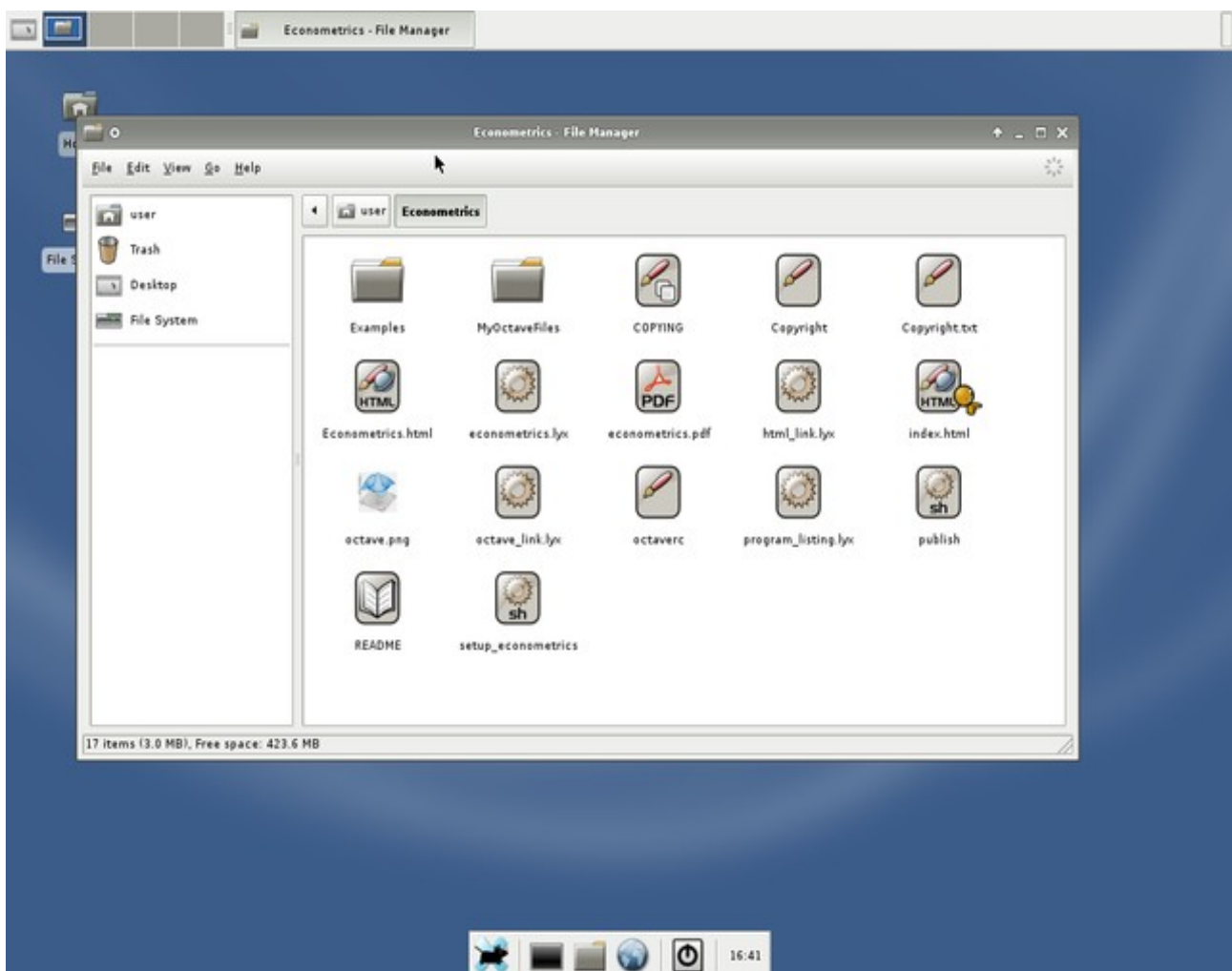
- Вы можете добавить программного обеспечения для узла, используя интерфейс "APT-get install <имя-пакета>", предполагая, что терминал кластера имеет вторую сетевую карту и что у вас настроен доступ в Интернет (или вы имеете .deb пакеты, расположенные локально на ваших постоянных носителях). Чтобы добавить программу так, чтобы она была доступна для всех узлов, она должна быть установлена где-то в /home/user директории.
- По умолчанию настройки MPI находятся в файле /home/user/tmp/bhosts. В нем находятся имена адресов узлов кластера и др. параметры. Если хосты имеют разные производительности количество ядер, и т.д., вы должны изменить этот файл. Если интерфейс узла является виртуальным, а вычислительные узлы являются реальными, вероятно, следует удалить интерфейс узла из расчетов.
- Доступна для использования ksysguard, и небольшое количество усилий превратит его в хорошее средство мониторинга кластера. См. [этот пост](#) для получения общей информации о том, как это сделать.
- Если вам нужны другие пакеты, то вы можете сделать свою собственную версию довольно легко, с помощью make\_pelican сценария, который доступен на главной странице PelicanHPC. Как это сделать будет рассмотрено далее.
- Вы можете изменить размер кластера (добавить или удалить вычислительные узлы), если хотите, запустив "pelican\_restart\_hpc".

**ВАЖНО:** В каталоге /home/user есть файл pelican\_config. Этот файл содержит варианты для дополнительных параметров, которые позволяют задействовать такие функции, как использование постоянного хранения, загрузки без вмешательства пользователя, изменение

сети кластера, Wake-On-LAN и т.д. Обычным пользователям нет необходимости исследовать его, но люди, которые хотят кластер на постоянной основе, должны изучить файл. Он хорошо прокомментирован.

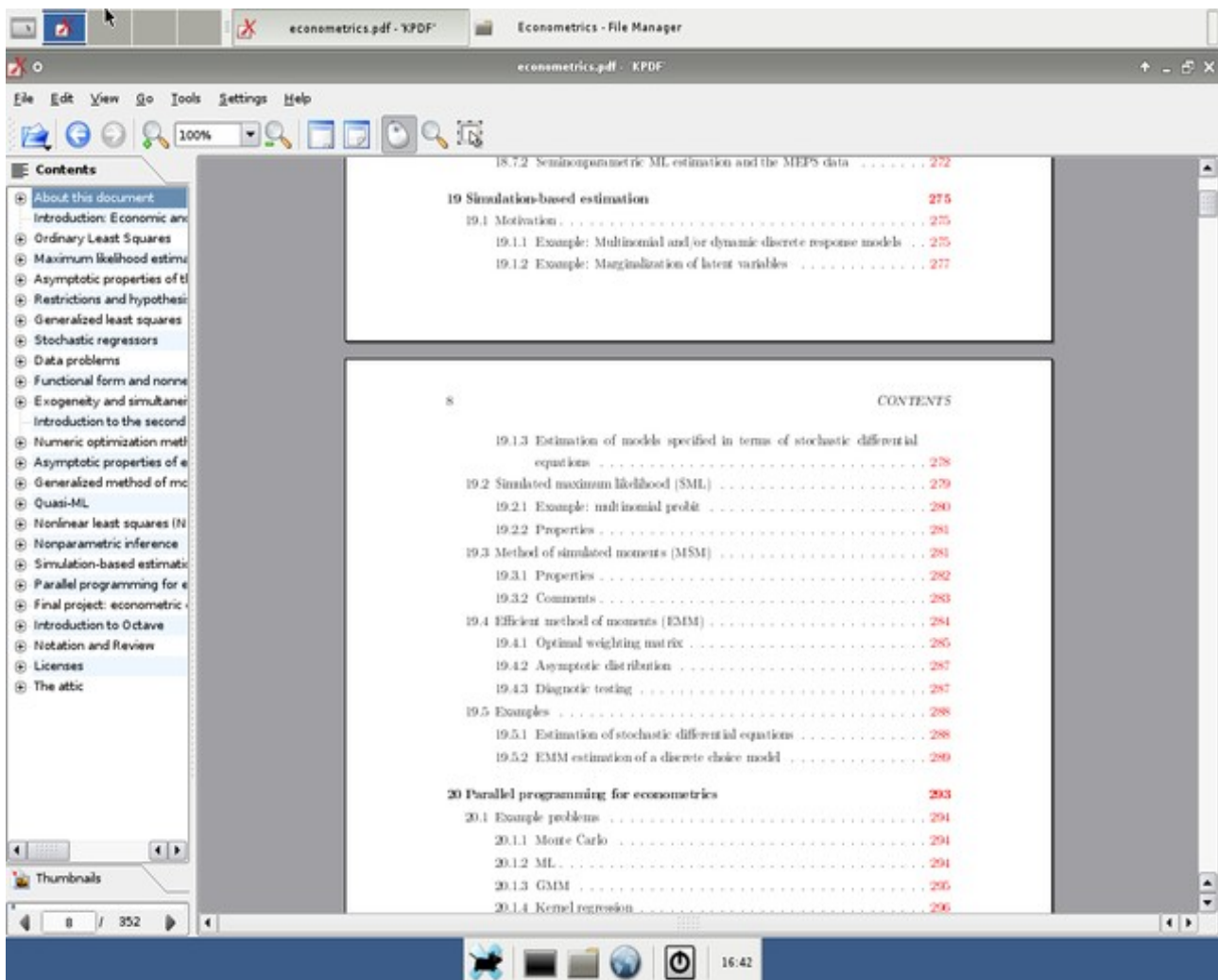
## Примеры программ

PelicanHPC имеет [Linpack HPL](#) тест и некоторые примеры из обширной области эконометрики, которые используют [GNU Octave](#). Эконометрические исследования, в которых применяется статистические методы в экономических моделях. Программное обеспечение находится в каталоге Econometrics :

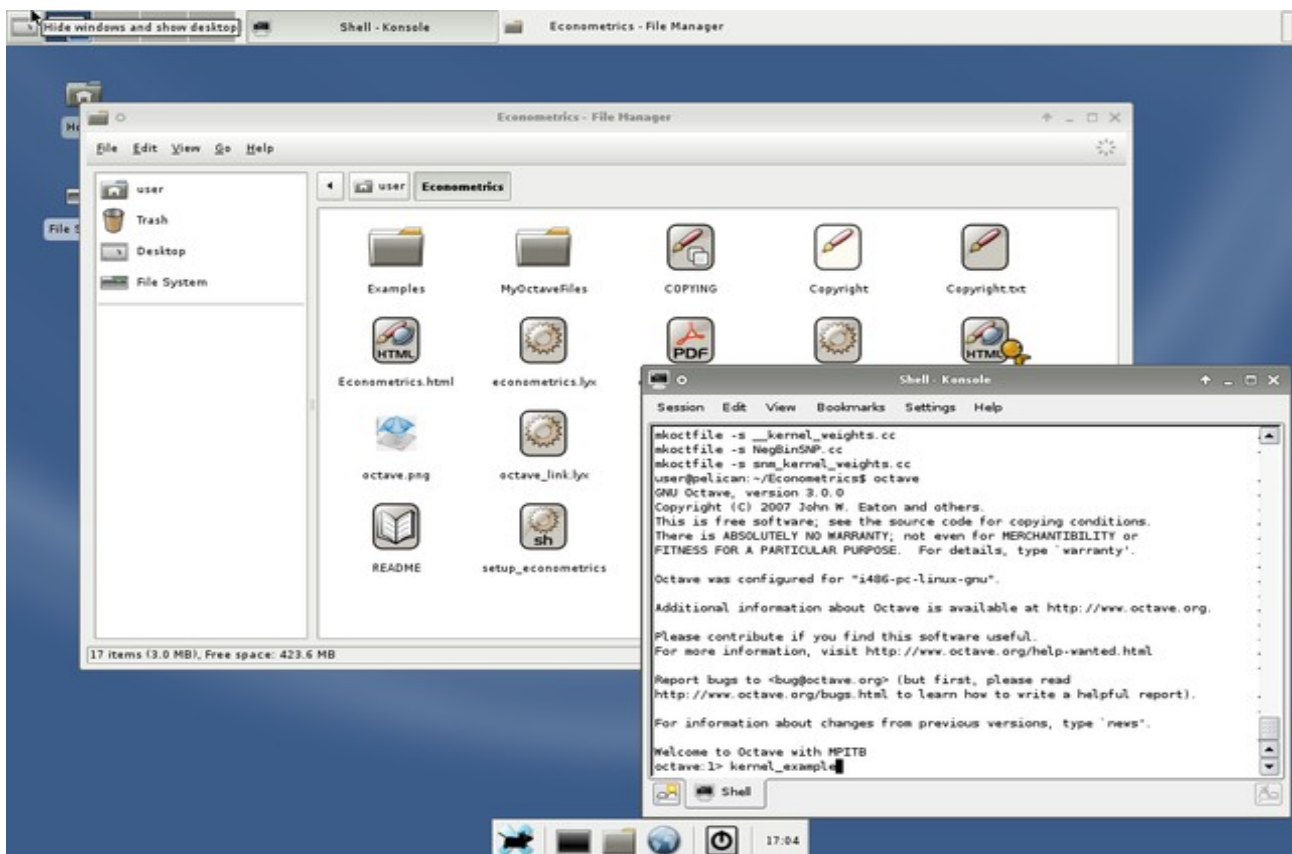




Здесь есть документ "econometrics.pdf", содержащий много информации, включая параллельные вычисления:



Откройте терминал, наберите «octave» и затем «kernel\_example» :



```
user@pelican: ~/Desktop
Report bugs to <bug@octave.org> (but first, please read
http://www.octave.org/bugs.html to learn how to write a helpful report).
For information about changes from previous versions, type `news'.
octave:1> kernel_example
time for kernel regression example using 1000 data points: 0.740000
octave:2>
```

File System



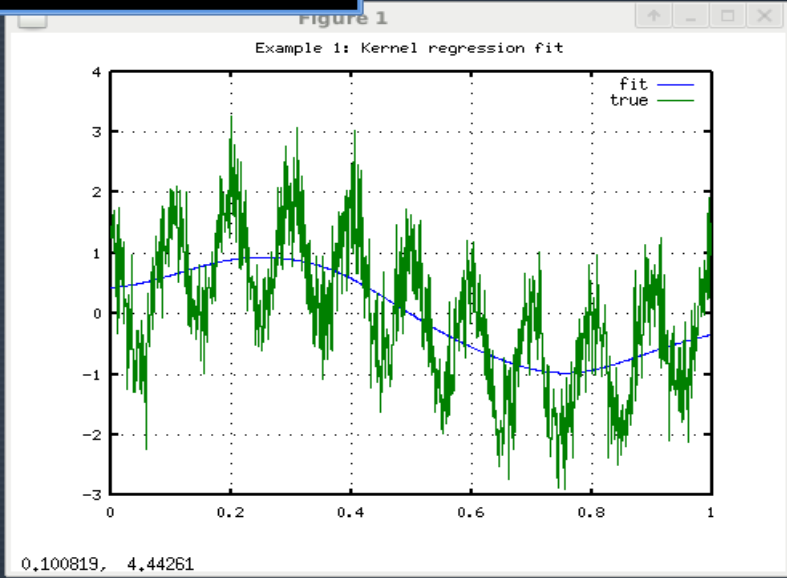
Trash



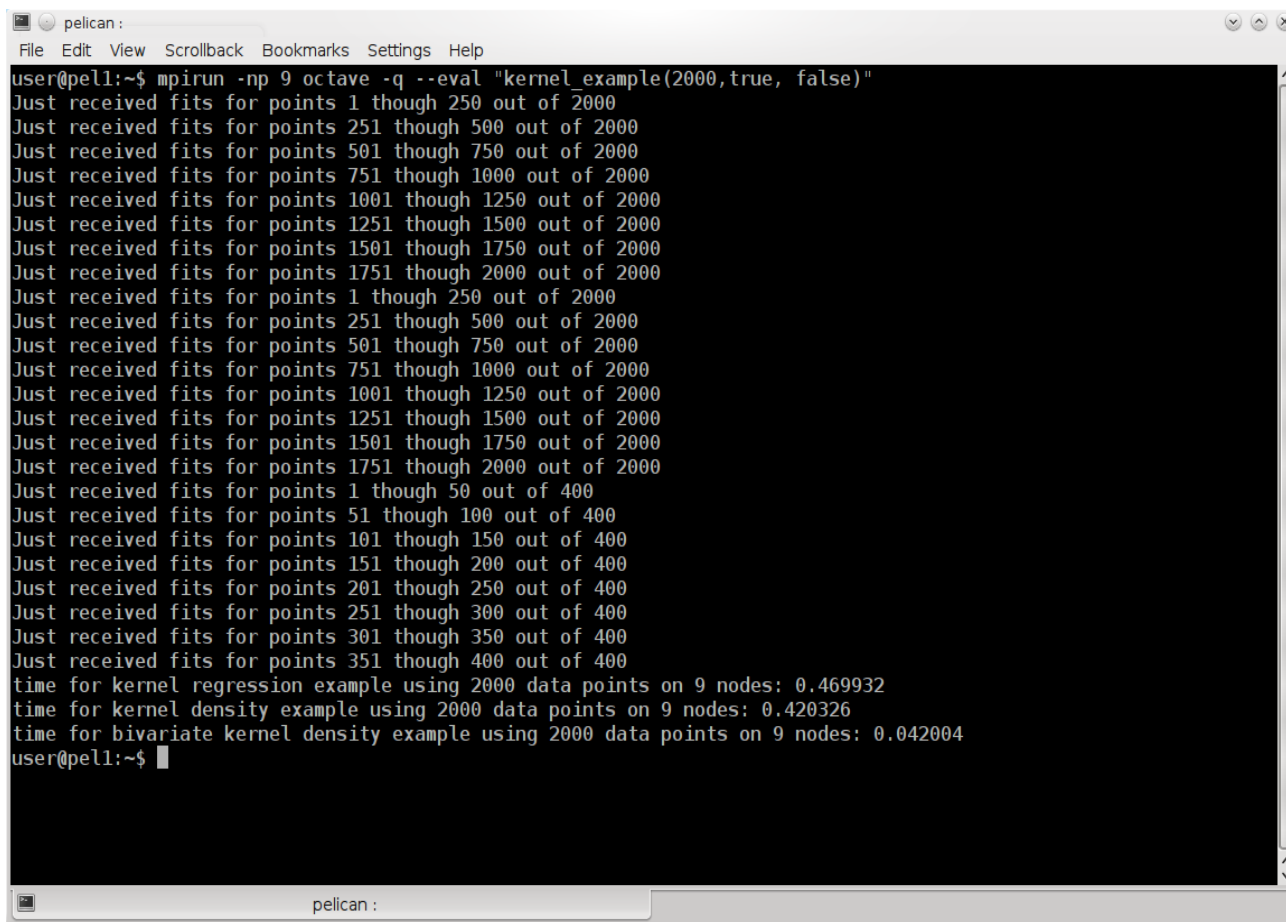
PelicanHPC



PelicanTuto...



Это последний скриншот - вывод `kernel_example.m`, если она выполняется последовательно, на одном ядре. Чтобы увидеть, как запустить его в параллельный, см. следующий снимок. **ПРИМЕЧАНИЕ** процедуры не вычисляются на ядре с рангом 0 (оно используется для сбора результатов), поэтому вы должны указать по крайней мере 2 MPI ядра.

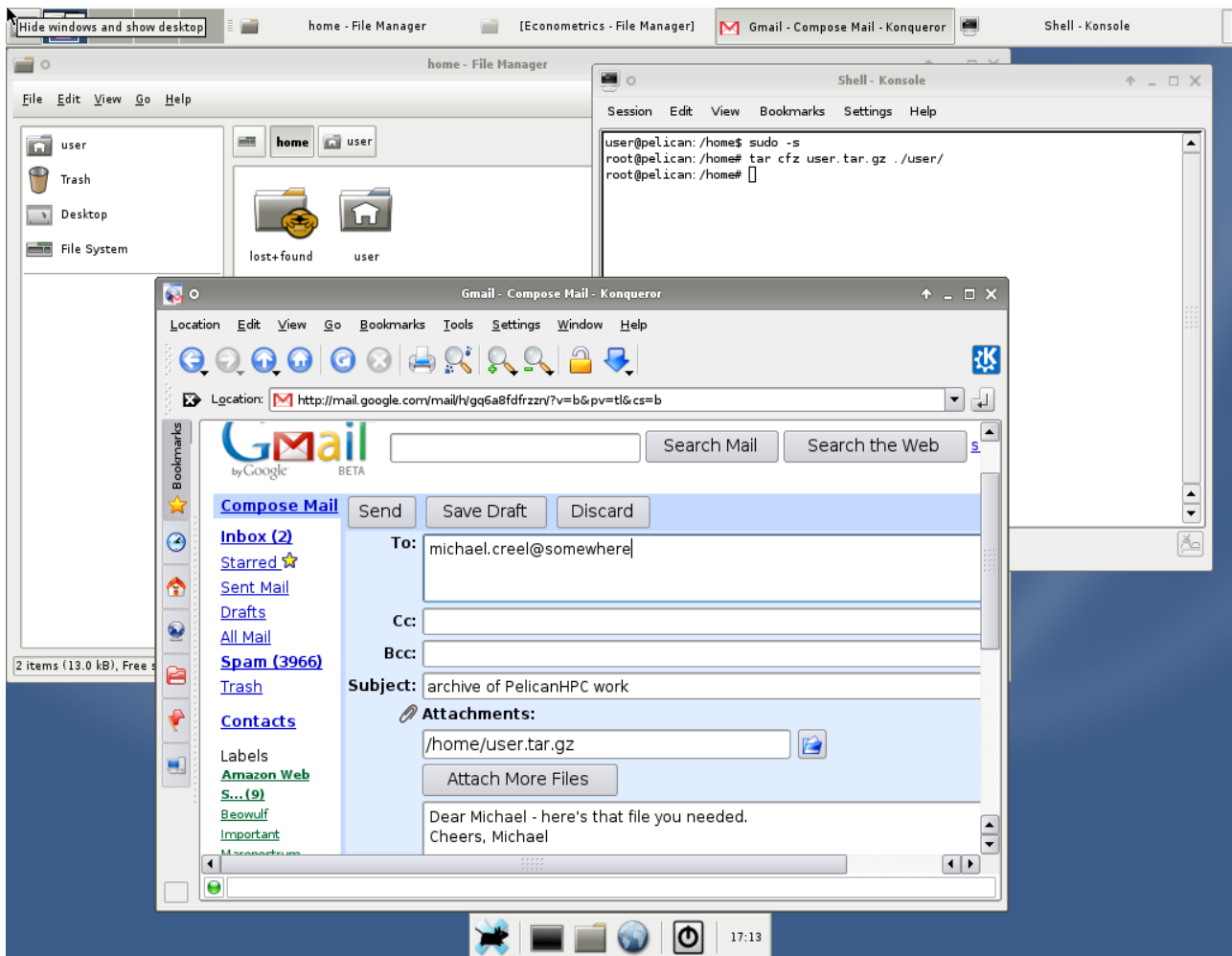


```
user@pell:~$ mpirun -np 9 octave -q --eval "kernel_example(2000,true, false)"
Just received fits for points 1 though 250 out of 2000
Just received fits for points 251 though 500 out of 2000
Just received fits for points 501 though 750 out of 2000
Just received fits for points 751 though 1000 out of 2000
Just received fits for points 1001 though 1250 out of 2000
Just received fits for points 1251 though 1500 out of 2000
Just received fits for points 1501 though 1750 out of 2000
Just received fits for points 1751 though 2000 out of 2000
Just received fits for points 1 though 250 out of 2000
Just received fits for points 251 though 500 out of 2000
Just received fits for points 501 though 750 out of 2000
Just received fits for points 751 though 1000 out of 2000
Just received fits for points 1001 though 1250 out of 2000
Just received fits for points 1251 though 1500 out of 2000
Just received fits for points 1501 though 1750 out of 2000
Just received fits for points 1751 though 2000 out of 2000
Just received fits for points 1 though 50 out of 400
Just received fits for points 51 though 100 out of 400
Just received fits for points 101 though 150 out of 400
Just received fits for points 151 though 200 out of 400
Just received fits for points 201 though 250 out of 400
Just received fits for points 251 though 300 out of 400
Just received fits for points 301 though 350 out of 400
Just received fits for points 351 though 400 out of 400
time for kernel regression example using 2000 data points on 9 nodes: 0.469932
time for kernel density example using 2000 data points on 9 nodes: 0.420326
time for bivariate kernel density example using 2000 data points on 9 nodes: 0.042004
user@pell:~$
```

Другие вещи для экспериментов: `"bfgsmin_example"`, `"mle_example"`, `"gmm_example"`, `"mc_example"` и несколько других... Чтобы найти где находится исходный код, наберите `"help mc_example"` например, в то время когда находитесь в Octave. Затем откройте для редактирования соответствующий файл, чтобы узнать больше о том, что он делает. Или, в то время когда находитесь в Octave, наберите `edit «bfgsmin_example»` (или измените на что угодно) и имя файла, содержащееся в пути Octave, который открывается в редакторе Vim.

## Сохранение работы

По умолчанию в PelicanHPC все данные находятся в `/home/user` на виртуальном диске, расположенным в оперативной памяти и после выключения все будет потеряно. Вы должны сохранить свою работу в период между сессиями, если вы хотите в дальнейшем использовать ее. Есть много вариантов, таких как монтирование жесткого диска, использование USB-устройства, и т.д. Если у вас настроено соединение с интернетом, вы можете отправить работу себе самому, как это показано на следующем снимке:



Если вы используете 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. Это функция для продвинутых пользователей, и я настоятельно рекомендуем вам выделить раздел жесткого диска для использования с PelicanHPC. При использовании раздела с другими данными на нем, вы должны убедиться, что вы сделали его резервную копию, прежде чем использовать его с PelicanHPC! Поддерживаются только ext2 и ext3 форматы. Эта функция была проверена с помощью очень ограниченного набора аппаратных средств, так что используйте его с осторожностью. Существует также возможность автоматического монтирования тома, который имеет специальное название. См. pelican\_config в /home/user Это лучшее решение для пользователей, которые хотят использовать PelicanHPC на долгосрочной основе.

## Использование make\_pelican сценария

ISO образ дистрибутива обеспечивает первоначальную настройку системы кластеров, а также некоторые пакеты, которые я использую в своей исследовательской и преподавательской деятельности. Есть несколько примеров, взятых из моей работы, которые могут представлять интерес для тех, обучения основам МРІ, или людей, заинтересованных в эконометрике. Однако, многие пользователи сочтут, что пеликан не содержит пакеты, в которых они нуждаются. Если настроить pelican\_config правильно, можно обеспечить всем узлам кластера доступ в Интернет через подключение интерфейса узла, поэтому пакеты

могут быть просто добавлены с помощью "АРТ-GET". Тем не менее, некоторые пользователи предпочитают иметь специальную версию образов . PelicanНПС создается путем запуска одного сценария "make\_pelican", который доступен на [странице загрузки](#) . Если у вас есть предпосылки для запуска сценария, очень легко сделать специальную версию Пеликан. Минимальные требования для создания своей версии Pelican\_НПС следующие:

- Установленная версия GNU / Linux. Это может быть минимальная установка в изолированной среде. Вы даже можете использовать виртуальную машину под Windows, если вы пользователь Windows.
- Live-cd версия пакета. Подходящие версии доступны в Debian Unstable: получить их можно на <http://packages.debian.org/sid/live-build> . Они доступны в качестве .deb пакета, а также как исходные код для использования с другими дистрибутивами. Кроме того, необходимы пакеты debootstrap, Wget и Rsync.
- Изучение make\_pelican скрипта, который содержит несколько пояснений и комментариев. Добавьте нужные пакеты в раздел списка пакетов.
- Вам необходимо запустить скрипт make\_pelican в качестве суперпользователя ( root ). Высокоскоростное подключение к Интернет может быть полезным, поскольку многие пакеты должны быть загружены по сети. Кроме того, эта возможность позволяет создавать образы на быстрых, как мы надеемся многоядерных компьютерах. Время сборки для конфигурации по умолчанию на двухъядерном ноутбуке, с большим количеством оперативной памяти занимает меньше чем полчаса.
- Когда вы закончите получится файл "binary-hybrid.iso" в каталоге ../<architecture>/frontend, где .. / является местом расположения make\_pelican сценария, и <architecture> либо i386 или amd64 , в зависимости от того, какие вы не закомментировали в файле сценария.
- Существует [пособие для Debian Live](#) . Пожалуйста, посмотрите его, прежде чем пытаться использовать make\_pelican. Дополнительная информация на [домашней странице Debian Live](#) . Эта информация является основной документацией, так как make\_pelican это просто скрипт, который обеспечивает конкретную конфигурацию системы Debian Live для сборки live-cd образа системы. Также имейте в виду, что файлы "man live-build", "man lb\_config" и "man lb\_build" могут дать вам дополнительную информацию.