Памятка пользователю вычислительного кластера Кемеровского государственного университета

Работа на вычислительном кластере Кемеровского государственного университета осуществляется по следующему алгоритму (с незначительными вариациями):

1. Запустить ssh-клиент (например, программу «Putty»).
2. Установить соединение с хостом hpc-cluster.kemsu.ru (82.179.12.210)
3. Ввести login name и пароль.
4. При первом сеансе работы на кластере и далее через каждый месяц следует сменить пароль посредством команды "**passwd**".
5. Большинство действий удобнее выполнять из файлового менеджера Midnight Commander, вызов которого осуществляется командой "**mc**". Для выхода из Midnight Commander следует нажать клавишу *F10*. Для того, чтобы временно выйти в командную строку и обратно (свернуть окно mc и восстановить его) следует нажать комбинацию клавиш *Ctrl+O*.
6. Для копирования файлов на головной узел кластера с Windows-компьютера можно воспользоваться программой «WinSCP». При запуске программы в поле "Host name" необходимо вписать master.kemsu.ru и нажать на кпопку "login". В появившемся окне в поле "username" ввести свой login, а в окне, которое появится после данного – пароль в поле Password. Работа в данной программе производится аналогично работе с файловыми менеджерами (Far, Total Commander и т.д.).
7. Для создания новых файлов можно применять команду "**touch**".

   Пример: [user@hpc-cluster temp]$ **touch** newfile
8. Для редактирования созданного файла следует выделить его в Midnight Commander и нажать клавишу *F4*. В открывшемся окне редактирования можно ввести текст и сохранить при помощи клавиши *F2*. Для возврата в файловый менеджер следует нажать клавишу *F10*.
9. На кластере установлены 3 реализации стандарта MPI (OpenMPI 1.6.2, MPICH2 1.4.1p1 и IntelMPI 4.0.1.007). При этом IntelMPI существует в двух вариантах: для 32-битной архитектуры и для 64-битной. Для выбора одной из них необходимо воспользоваться механизмом Environment Modules. Для отображения установленных реализаций требуется ввести команду:

   [user@hpc-cluster temp]$ **module avail**

   Для того, чтобы установить одну из них для использования по умолчанию в ходе текущего сеанса работы, необходимо выполнить команду:

   [user@hpc-cluster temp]$ **module load** *имя_реализации*

   Если всегда требуется только одна MPI-реализация, то можно прописать команду «module load …» в файл «.bash_profile», располагающийся в домашнем каталоге.

   Увидеть установленную по умолчанию реализацию можно при помощи команды «**module list**». Если необходимо сменить текущую используемую реализацию на другую, перед тем, как загрузить новый модуль командой «**module load** …», сначала рекомендуется выгрузить используемый при помощи команды:

[user@hpc-cluster temp]$ **module unload** *имя_реализации*

Пример:

[user@hpc-cluster temp]$ **module avail**

```
---------------------------- /usr/share/Modules/modulefiles --------------
dot                          mpi/intelmpi_32bit          null
module-git                   mpi/intelmpi_64bit(default) rocks-openmpi_ib
module-info                  mpi/mpich2                  use.own
modules                      mpi/rocks-openmpi
```

[user@hpc-cluster temp]$ **module list**

```
No Modulefiles Currently Loaded.
```

[user@hpc-cluster temp]$ **module load** mpi/intelmpi_64bit

[user@hpc-cluster temp]$ **module list**

```
Currently Loaded Modulefiles:
  1) mpi/intelmpi_64bit(default)
```

После этого команды типа “mpicc”, “mpicxx”, “mpirun” будут вызываться из директории IntelMPI для 64-битных приложений:

[user@hpc-cluster temp]$ **which mpicc**

```
/opt/intel/impi/4.0.1.007/intel64/bin/mpicc
```

10. Компиляция параллельных MPI-программ на языке C++ осуществляется при помощи команды “**mpicxx**” (для “чистого” C - mpicc), на языке Fortran – при помощи команд “**mpif77**” или “**mpif90**”. В случае, если выбрана реализация intelmpi-4.0.1.007, то рекомендуется использовать команды “**mpiicc**” и “**mpiifort**” соответственно – в этом случае компиляция будет производиться компиляторами Intel, а не GNU, что зачастую приводит к увеличению производительности программы.

    Примеры: [user@master temp]$**mpiicc program.cpp -o program.out**

    [user@master temp]$**mpiifort program.f -o program.out**

    [user@master temp]$**mpif90 program.f -o program.out**

11. Компиляция OpenMP-программ на языке C/C++ производится при помощи команды “**g++**” с ключом –fopenmp. Для OpenMP-программ на Fortran следует использовать компилятор “**gfortran**” с ключом –fopenmp.

    Примеры: [user@master temp]$**g++ -fopenmp program.cpp –o program.out**

    [user@master temp]$**gfortran -fopenmp program.f –o program.out**

    Также можно использовать компиляторы Intel, установленные на кластере. Для этого служат команды **icc** или **icpc** (для программ на C/C++) и **ifort** (для программ на Fortran). Оба компилятора можно вызывать с ключом –openmp для компиляции OpenMP-программ.

    Примеры: [user@master temp]$**icc -openmp program.cpp –o program.out**

[user@master temp]$**ifort -openmp program.f –o program.out**

12. Для запуска задания, выполняющего на кластере созданный исполняемый файл MPI-программы, следует создать файл задания следующего содержания:

```
#PBS -l walltime=00:30:00,nodes=4:ppn=8

#PBS -q rail1

#PBS -N job_name

#PBS -o /home/[user]/out

#PBS -e /home/[user]/err

#!/bin/sh

module load mpi/mpich2 #даже если некоторый модуль был загружен из командной строки в текущем сеансе, при формировании задания это следует задать еще и здесь, поскольку задание выполняется не на головном узле

mpirun –n 32 /home/[user]/program.out   #будет выбран /opt/mpich2/gnu/bin/mpirun
```

где:

- *walltime* – предполагаемое время выполнения,

- *nodes* - число вычислительных узлов,

- *ppn* – число процессорных ядер (более точно – аппаратных потоков) на каждом узле,

- *#PBS -q rail1* - имя очереди,

- *#PBS -N job_name* - имя задания,

- *#PBS -o /home/[user]/out* – указание пути для записи файла потока вывода,

- *#PBS -o /home/[user]/err* – указание пути для записи файла потока ошибок,

- *#!/bin/sh* – командный интерпретатор,

- *mpiexec, mpirun* – команда запуска параллельного приложения,

- *n* – число ветвей параллельного приложения,

- *program.out* – исполняемый файл параллельной программы.

Внимание!

Замечание 1. Очереди заданий

На кластере существуют 3 очереди:

- **rail0** - 2 узла HP BL260c G5 (каждый узел - 8 процессорных ядер, 8 Гб оперативной памяти, управляющая сеть - Ethernet, вычислительная сеть - InfiniBand);

- **rail1** - 5 узлов HP BL280c G6 (каждый узел - 8 процессорных ядер, 24 Гб оперативной памяти, управляющая сеть - Ethernet, вычислительная сеть - InfiniBand);

- **rail2** – 5 узлов Dell PowerEdge M630 (каждый узел – 32 процессорных ядра по 2 аппаратных потока каждое, 192 Гб оперативной памяти, сеть – 10 Gigabit Ethernet).

Замечание 2. Выбор коммуникационной среды

Команда **mpirun** реализации Intel MPI обладает ключами для выбора коммуникационной среды (Ethernet, Infiniband), над которой будет запущена параллельная программа. Один из способов явного задания среды – установка переменной окружения **I_MPI_FABRICS** через ключ **"env"**. Данная переменная имеет следующий синтаксис:

I_MPI_FABRICS=<*intra-node fabric*>:<*internodes-fabric*>

где:

intra-node fabric – среда для коммуникаций между процессами, запущенными на одном узле кластера. Может принимать значения *shm, tcp, tmi* или *ofa* (Таблица 1).

inter-node fabric - среда для коммуникаций между процессами, запущенными на разных узлах кластера. Может принимать значения *tcp, tmi* или *ofa*.

Таблица 1. Некоторые значения переменной I_MPI_FABRICS

Мета-символ	**Коммуникационная среда**
shm	Общая память
tcp	Среда, управляемая стеком TCP/IP (в нашем случае - Ethernet)
ofa	Среда, управляемая стеком OFED (в нашем случае - Infiniband)

Примеры:

1. Запуск восьми процессов таким образом, что процессы на одном узле будут обмениваться данными через общую память, а на разных – через Infiniband: **mpirun –n 8 –env I_MPI_FABRICS shm:ofa /home/[user]/program.out**
2. Запуск восьми процессов таким образом, что процессы на одном узле будут обмениваться данными через общую память, а на разных – через Ethernet: **mpirun –n 8 –env I_MPI_FABRICS shm:tcp /home/[user]/program.out**

Файл задания для OpenMP-программы различается только тем, что после указания командного интерпретатора вместо строки с вызовом **"mpirun…"** содержит следующие строки:

```
export OMP_NUM_THREADS=8

./program.out
```

где:

- OMP_NUM_THREADS – количество потоков приложения (их число не должно превышать числа аппаратных потоков на вычислительном узле).

13. Для постановки задачи в очередь на запуск следует использовать команду “**qsub**”.

    Пример: **qsub my_job**

    где my_job – имя файла задания (см пункт 12).

14. Для отслеживания состояния задачи в очереди можно использовать команды “**qstat**” и “**showq**”.

15. По окончании работы программы файлы с перенаправленными потоками вывода и ошибок будут находиться в директории /home/user (если были использованы параметры, указанные в пункте 12) и иметь имена out и err соответственно.

16. Для завершения сеанса работы с кластером необходимо ввести команду “**exit**”.

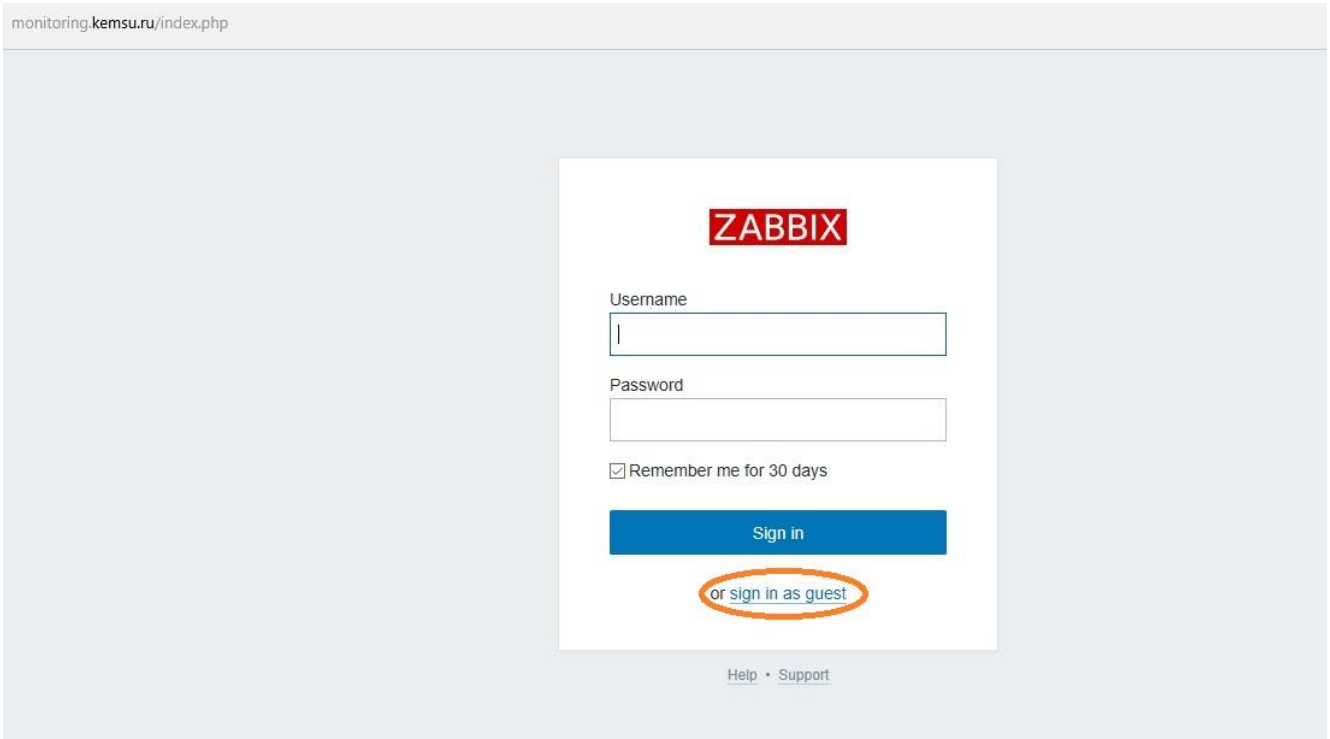
Мониторинг загрузки вычислительных ресурсов

Высокопроизводительные вычислительные ресурсы центра коллективного пользования научным оборудованием КемГУ подключены к системе мониторинга Zabbix. Благодаря этому пользователи в момент выполнения своей задачи на каких-либо узлах могут отслеживать текущий уровень загрузки процессора, памяти, сетевых интерфейсов и др.

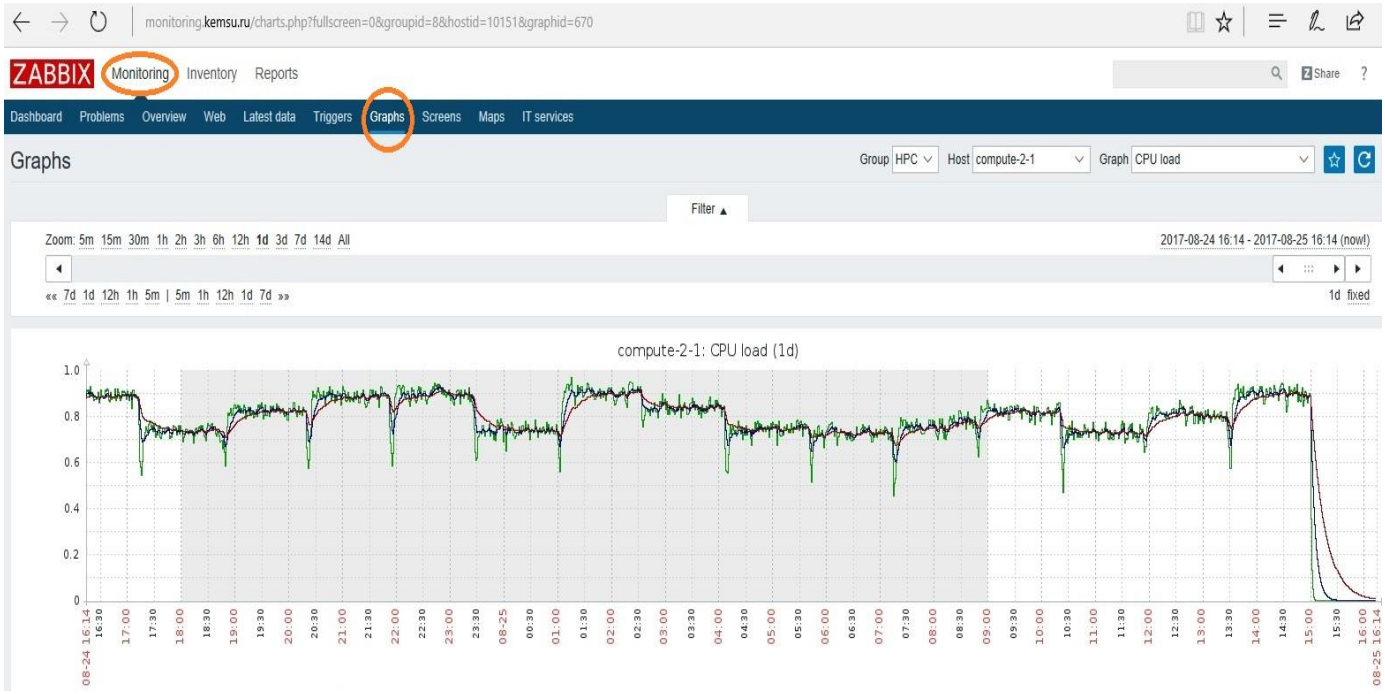
Для этого в броузере необходимо:

1. открыть URL monitoring.kemsu.ru

2. перейти по ссылке «sign in as guest»



3. в меню «Monitoring» выбрать пункт «Graphs». В раскрывающемся списке «Group» выбрать «HPC», а в списках «Host» и «Graph» - соответственно интересующий Вас сервер и график. Многие графики можно масштабировать при помощи ползунка, расположенного над графиком.


monitoring.kemsu.ru/charts.php?fullscreen=0&groupid=8&hostid=10151&graphid=670
ZABBIX
Monitoring
Inventory
Reports
Share
Dashboard
Problems
Overview
Web
Latest data
Triggers
Graphs
Screens
Maps
IT services
Graphs
Group HPC
Host compute-2-1
Graph CPU load
Filter ▲
Zoom: 5m 15m 30m 1h 2h 3h 6h 12h 1d 3d 7d 14d All
2017-08-24 16:14 - 2017-08-25 16:14 (now!)
«« 7d 1d 12h 1h 5m | 5m 1h 12h 1d 7d »»
1d fixed
compute-2-1: CPU load (1d)
1.0
0.8
0.6
0.4
0.2
0
08-24 16:14
16:30
17:00
17:30
18:00
18:30
19:00
19:30
20:00
20:30
21:00
21:30
22:00
22:30
23:00
23:30
08-25
00:30
01:00
01:30
02:00
02:30
03:00
03:30
04:00
04:30
05:00
05:30
06:00
06:30
07:00
07:30
08:00
08:30
09:00
09:30
10:00
10:30
11:00
11:30
12:00
12:30
13:00
13:30
14:00
14:30
15:00
15:30
16:00
08-25 16:14