VRE for regional Interdisciplinary communities in Southeast Europe and the Eastern Mediterranean

Достъп до високопроизводителния изчислителен клъстер Авитохол

Мария Дурчова
Димитър Димитров
Институт по информационни и комуникационни технологии – БАН

Съдържание

- Архитектура и конфигурация
- Достъп до системата Авитохол
- Потребителска среда
  - основни команди за управление на задания (**jobs**)
  - основни PBS променливи на средата
- Работа с модули на средата

Архитектура и конфигурация

- 150 сървъра HP ProLiant Gen8 SL250S – един от тях е запазен за вход и подаване на задачи, тестване и разработка на приложения;
- 4 сървъра HP ProLiant DL380p Gen8 с по 2 Intel Xeon E5-2650 v2, 64GB RAM, за достъп и управление на 96 TB дисково пространство по Fibre Channel
- 2 сървъра за управление HP ProLiant DL380p Gen8 с по 2 Intel Xeon E5-2650 v2, 64GB RAM
- Напълно неблокираща 56Gbps FDR InfiniBand свързаност между всички гореописани възли, с латентност близо 1 микросекунда

Архитектура и конфигурация

На изчислителните възли:

- 2 Intel Xeon 8-ядрен E5-2650 v2 @ 2.6 GHz
- 2 Intel Xeon Phi 7120P копроцесора, с по 16 GB RAM и 61 ядра всеки
- Основна памет - 64 GB (9600 GB общо)
- Памет на ускорителите: 16 GB (4.8 TB общо)
- Скорост на изпълнение на операции с двойна точност на Intel Xeon Phi 7120P – 1,25 TFlop/s.

Архитектура и конфигурация

- Операционна система - Red Hat Enterprise Linux, версия 6.7.
- Софтуерът MPSS версия 3.6-1 позволява всеки един копроцесор да е видим като отделен сървър.
- Файлова система за четене и запис е от тип Lustre - **/home**
- Файлова система за четене, споделена чрез NFS - **/opt**

  (тук са разположени софтуерни продукти, достъпни за всички потребители обикновено с използване на модули).

- Несподелена между възлите файлова система **/dev/shm**, разположена в паметта на всеки изчислителен възел.

  + Позволява бърз запис на голям брой малки по размер файлове.

  - Не запазва данните след рестарт.

Архитектура и конфигурация

- Проведените тестовете показват следните възможности:
  - теоретична пикова производителност - **412.32 TFlop/s**,
  - общо **9600 GB** памет
  - реално постигната производителност - **264,2 TFlop/s**.

- Ноември, 2015 - 388 място в класацията http://www.top500.org, като първоначално заема 332 място в листата от юни 2015 г.

- Актуална информация: http://www.hpc.acad.bg

- При възникнали и неотстраними от потребителя проблеми:
  avitohol-support@parallel.bas.bg

Достъп до системата Авитохол

Входният възел за достъп до Авитохол: **gw.avitohol.acad.bg**

- хардуерна и софтуерна конфигурация, идентична с тази на изчислителните възли
- само този възел може да се използва за пускане на задачи (с командите на Torque/PBS).
- само от тук потребителят може да се логва с SSH на изчислителните възли, на които вече е стартирана негова задача или на ускорителите на Xeon Phi, които са свързани с тези възли.

Изпълнителните възли са **sl001, ... sl150**

Ускорителите са **sl001-mic0, sl001-mic1, sl002-mic0, sl002-mic1, …**

Потребителска среда

Компилатори

- Компилатори на Intel: C / C ++ - **icc, icpc** Fortan – **ifort**, (оптимизирани са за Xeon Phi).
- Компилатори на Intel за MPI кодове: **mpiifort**, **mpiicc** ,**mpiicpc**.

- GNU компилатори: **gcc, g++, gfortran** – версия от ОС - 4.4.7 по подразбиране.
- GNU компилатори за компилиране и свързване на MPI кодове, но с Intel MPI библиотеки: **mpicc, mpic++/mpicxx, mpif77,mpif90** .

- Компилацията на програми за Xeon Phi обикновено се извършва на основните възли, т.е., като се използва крос-компилация.

Потребителска среда

- Команди за управление и изпълнение на задачите
  - **qstat -u <username>** - списък със стартираните от портебителя задания
  - **qstat -f <jobID>** - проверка дали заданието е получило необходимите му ресурси
  - **qstat -n <jobID>** - информация за възлите, върху които е стартирано заданието
  - **qstat -Q <queue>** - информация за характеристиките на конкретната опашка
  - **qsub <scriptname>** - стартиране на скрипта със заданието
  - **qdel <jobID>** - изтриване на конкретно задание

- Променливи на средата
  - **PBS_O_WORKDIR -** показва директорията, от която е изпълнено заданието
  - **PBS_JOBID -** идентификационен номер на старирано задание
  - **PBS_NODEFILE -** файл, съдържащ резервираните за работата ресурси
  - **PBS_NP –** максималния брой на процесите, които са резервирани в **PBS_NODEFILE**

Потребителска среда

- Зареждане на средата за използване на компилаторите на Intel:

source /opt/intel/compilers_and_libraries_2016.2.181/linux/bin/compilervars.sh intel64

Достъпват се **icc, icpc, ifort**

- Зареждане на средата за използване на паралелните версии

source /opt/intel/compilers_and_libraries_2016.2.181/linux/mpi/intel64/bin/mpivars.sh release_mt

!!! release_mt – има смисъл в случаите на употреба на "Hyperthreading / Multithreading (SMT)" режим

Достъпват се **mpiifort**, **mpiicc** ,**mpiicpc** **(**с или без Hyperthreading)

- Зареждане на средата за развитие за Xeon Phi:

source /opt/mpss/3.6/environment-setup-k1om-mpss-linux

Работа с модули на средата

- Модулите са средствата, с които лесно може да се достъпи вече инсталиран научен софтуер.
- Използването на различни компилатори, библиотеки и софтуерни пакети изисква от потребителя да си създаде специална среда (обвивка), подходяща за изпълнението на приложението.
- Модулът позволява динамично изменение на средата чрез използването на модулни файлове (modulefiles), съдържащи информация за конфигурирането на софтуера, както и инструкции, които променят определени променливи на средата (PATH, MANPATH и др.).
- В Авитохол потребителите имат възможност да прегледат, зареждат и отменят/премахват специфични променливи на средата за различни библиотеки и софтуерни пакети с една единствена команда.

Работа с модули на средата

module help - инструкции за начина на употреба на командата

module avail - показва всички налични модули на средата за вече инсталиран софтуер

module -t avail - показани по един модул на ред

module -l avail - показани по-обширна информация за файла на модула

module list - показва списък с вече заредените в средата модули

Работа с модули на средата

module whatis <modulename> - кратко описание на модула

module help <modulename> - някои помощни инструкции за конкретния модул, например специфични насоки как да се използва така заредения софтуер

module display/show <modulename> - показва промените, които модула ще направи, за да подготви средата, без да ги извършва

Зареждането в средата става с:

module load <modulename>

или

module add <modulename> -така се добавят промените от модула към променливите на средата

!!! Зареден по този начин, модулът е достъпен в среда САМО по време на работа на текущата сесия.

Работа с модули на средата

Премахване или отмяна на свързаните с модула променливи:

module unload <modulename> или

module remove <modulename>

!!! Премахнат или изтрит по този начин модул, който е бил зареден в средата по подразбиране, става неактивен само за текущата сесия - той ще се зареди при следващото влизане в системата.

За да премахнете всички предварително заредени по подразбиране софтуерни модули от текущата среда се изпълнява командата:

module purge

!!! Работи без никакво допълнително потвърждение от потребителя.

или

module clear - премахва всички заредени модули, но с необходимост от потребителско потвърждение с "[n]/y“.

Работа с модули на средата

- В една сесия не могат да се заредят модули с различни версии на един и същ софтуер. Едновременното зареждане на модули с версия X и Y води до съобщение за грешка - конфликт.
- Може да се избегне с премахване или смяна на конфликтния модул

  module switch <modulename1> <modulename2>

  Така, например променливата на средата **PATH** ще се промени като частта от **<modulename1>** се премахва и се доразширява с **/bin** директорията на софтуера от **<modulename2>**

Работа с модули на средата

Модули, зависещи от други модули (сложни):

- Някои сложни програмни модули зависят от библиотеки, които трябва да бъдат заредени в потребителската среда. За това съответните модули на софтуера трябва да се зареждат едновременно с модулите на библиотеките.
- По подразбиране те се опитват да заредят необходимите си модули и версии автоматично, но това не винаги е приложимо. (пример: за да работи софтуер е необходима една версия на библиотека, а в средата вече е заредена друга конфликтна версия).
- Ако преди зареждането на такъв сложен модул, вече е зареден някой от модулите от които зависи, то системата проверява за съвместимост на версиите - при конфликт излиза с грешка, при липса на конфликт -дозарежда останалата част от променливите от сложния модул.

Благодаря за вниманието

https://events.hpc.grnet.gr/event/20/evaluation