Ръководство за използване на суперкомпютър Avitohol

Глава 1. Конфигурация на хардуер и софтуер на системата Авитохол

1. Въведение

През юни 2015 г. новият български суперкомпютърен клъстер беше разположен в ИИКТ-БАН и започна тестови изпитания. Клъстерът се състои от 150 сървъра HP Cluster Platform SL250S GEN8.
Всеки изчислителен възел има два Intel Xeon процесора E5-2650 v2 @ 2.6 GHz (с по 8 ядра всеки), 64GB RAM и два Intel Xeon Phi 7120P копроцесора (с по 61 ядра всяка). По този начин, в момента има общо 20700 ядра.

Освен това, има 4 сървъра за входно-изходни операции, осигуряващи връзка по оптични кабели към 96 TB пространство за съхранение на данни (24 диска с по 4 TB).
Името на системата е *Авитохол*.

Постигнатите при тестовите изпитания резултати показват следните възможности: теоретична пикова производителност от 412.32 TFlop/s, общо 9600 GB памет и постигната реално производителност от 264,2 TFlop/s.

Тя е класирана на 389-то място в последната листа от ноември 2015 в класацията Топ 500 (http://www.top500.org), като първоначално зае 332 място в листата от юли 2015 г.

Фигура 1: Система Авитохол

2. Системна архитектура и конфигурация

2.1. Системна конфигурация

Текущата конфигурация на системата включва:

- 150 сървъра HP ProLiant Gen8 SL250S

- един от тях е запазен за вход и подаване на задачи, тестване и разработка на приложения
- 4 входно-изходни възела HP ProLiant DL380p Gen8 с по два Intel Xeon процесора E5-2650 v2, 64GB RAM, с достъп до дисковете по Fibre Channel
- два сървъра за управление HP ProLiant DL380p Gen8 с по два процесора Intel Xeon E5-2650 v2, 64GB RAM
- 96 TB на онлайн дисково пространство, свързано по Fibre Channel с входно-изходните възли.
- Напълно неблокираща 56Gbps FDR InfiniBand мрежа, свързваща всички по-гореописани възли, с латентност близо 1 микросекунда
- Процесори на изчислителните възли: двойка Intel Xeon 8-ядрен E5-2650 v2 @ 2.6 GHz
- Копроцесори на изчислителните възли: два Intel Xeon Phi 7120P копроцесора, 16 GB RAM и 61 ядра на всеки един
- Основна памет на изчислителните възли: 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.

Операционна система

Операционна система на Авитохол е Red Hat Enterprise Linux версия 6.7.

На копроцесорите работи софтуерът MPSS версия 3.6-1, който прави всеки един копроцесор видим като отделен сървър.

2.2. Файлови системи

Системата Авитохол е сравнително нова. Ето защо в момента единственият файловата система, която е достъпна за четене и запис за всички потребители е `/home` файловата система, която е от тип Lustre. Групи от потребители могат да поискат да имат споделена директория, така че те могат да споделят данни помежду си.

Някои софтуерни продукти, които се предоставят на разположение на всички потребители, са достъпенни в директорията `/opt` и обикновено са разположени с използване на модули. Тази файлова система се споделя чрез NFS и е само за четене за потребителите.

В момента на файловата система Lustre се предоставя само чрез два OST. Това на практика означава, че един голям файл може да се предоставя от два сървъра паралелно, ако този файл е създаден по подходящ начин.

За да се предоставят файловете от дадена директория по този начин може да се направи следното:

```
lfs setstripe <filename|dirname> --count 2
```

Това се препоръчва за големи файлове, с които ще се извършват много входно-изходни операции.

Като цяло файловата система Lustre е оптимизирана за обработка на големи файлове. Това означава, че създаването на голямо количество малки файлове може да бъде относително бавен процес. В такъв случай може да се използва /dev/shm файловата система на всеки възел, която е разположена в паметта. Тази файлова система обаче не е споделена между възлите и няма да запази данните след рестарт! За повечето потребители не е необходимо използването на /dev/shm.

Допълнителна информация

За повече информация за файловата система Lustre, справка с документацията за Lustre. Вижте, например, https://wiki.hpdd.intel.com/display/PUB/HPDD+Wiki+Front+Page

Допълнителни детайли

Приложения с високи изисквания за входно-изходни операции могат да поискат специално дисково пространство, даже отделна файлова система.

3. Достъп до системата

3.1. Отдалечен достъп

Входният възел за достъп до Авитохол е **gw.avitohol.acad.bg**. Пръстовите отпечатъци на ключовете в момента са:

1024 CD: 97: 16: 41: 2в: CD: 3d: 59: c4: f0: d2: 67: CE: 1в: 5e: 62 /etc/ssh/ssh_host_dsa_key.pub (DSA)

2048 B9: e3: 6e: f5: A2: EA: 8e: 95: 0A: 96: 5б: 55: 03: 0C: BF: E9 /etc/ssh/ssh_host_rsa_key.pub (RSA)

От съображения за сигурност се препоръчва на потребителите да използват ключове с най-малко 2048 бита.

За достъп се използва криптирана връзка по протокол SSH.
Входният възел има практически идентична хардуерна и софтуерна конфигурация с възлите за изпълнение.
Само този възел може да се използва за пускане на задачи (с командите на Torque/PBS).
След влизане в клъстера, потребителят може да се логва с SSH на изчислителните възли, на които той или тя има пусната задача (може да се провери с команда `qstat -na`) или на ускорителите на Xeon Phi, които са свързани с тези възли.

3.2. Използване на изчислителните ресурси на Авитохол

Входният възел **gw.avitohol.acad.bg** е предназначен предимно за редактиране на изготвяне и стартиране (пускане) на интензивни изчисления програми (задачи – `jobs`). Може да се извършват някои кратки тестове на паралелни програми. В случай на съмнение, потребителите са насърчавани да пуснат задача, която да заеме един цял изчислителен възел и да го използват за отстраняване на грешки и развитие на програми, като използват интерактивна задача.
Изпълнителните възли са именувани **sl001-sl150**.

3.3. Интерактивни пускания за отстраняване на грешки

Ако трябва да се отстраняват грешки в програмен код може да се използва входния възел **gw.avitohol.acad.bg**. Този възел има два процесора и два Xeon Phi копроцесора по същия начин, както възлите за изпълнение.

Необходимо е да се следи използването на ресурсите и в случай, че системата се претоварва (както може да се види от програмата **top**), следва да се спрат процесите и да се продължи на отделен изчислителен възел, където потребителят ще има ексклузивен достъп. Потребители, които причиняват видими проблеми на системата - системен срив или зависване на копроцесорите Xeon Phi, трябва да се опитат да разрешат проблемите с техните кодове. Ако проблемите продължават, те се насърчават да се свържат със системните администратори.
Имейлът за поддръжка е:

avitohol-support@parallel.bas.bg.

Допълнителни детайли

Има и други начини за достъп до системите за съхранение на данни от системата, които не се отнасят до повечето потребители. Ако Вашите нужди за съхранение надхвърлят един терабайт, моля опишете ги напълно

във формуляра за достъп. Достъп до различни интерфейси могат да бъдат предоставени на базата на оценка на тези нужди.

Копроцесорите Xeon Phi са видими като **sl001-mic0** до **sl150-mic0** и **sl001- mic1** до **sl150- mic1**. От потребителите се очаква да ги използват само след получаване изключителен достъп до съответният изчислителен възел. Забранява се ползването на произволни Xeon Phi копроцесори, без пускане на съответни изчислителни задачи.

4. Програмиране и потребителска среда

4.1. Система за изпълнение на задачите

Входният възел **gw.avitohol.acad.bg** на системата Авитохол е предназначен предимно за редактиране и съставяне на паралелни програми. Интерактивно използване на **mpirun / mpiexec** е позволено на входния възел и неговите копроцесори, но следва да се избягва. Препоръчително е да се премине към използване на изчислелни възли, заявени чрез системата за изпълнение на задачите, ако такъв тип паралелно изпълнение отнема повече от няколко минути, или ако се използва много памет.

Кратки тестови задачи (по-кратки от 15 мин.) с 2, 4 или 8 ядра ще стартират с предимство. Въпреки това, използването на частични възли (т.е. по-малко от 16 ядра от един възел) не се препоръчва. Използването на множество частични възли, например, като се поискат 10 възела с по 4 ядра, не се препоръчва. Дори ако се използват само 4 от ядрата, следва да се заявят всичките 16, тъй като има риск друга задача да заеме много повече от очаквания брой ядра.

По подразбиране, ограничението за време за работа на задача е настроено на 24 часа. Ако Вашите задачи не могат да работят независимо една от друга, моля използвайте стъпки за изпълнение или се свържете с администраторите на системата чрез Helpdesk на имейл адреса

avitohol-support@parallel.bas.bg.

Процесорите на Intel поддържат "Hyperthreading / Multithreading (SMT)" режим, който може да увеличи производителността на едно приложение с до 20%. Въпреки това, ние оставяме на потребителите да решават дали да използват Hyperthreading или не. В момента изчислителните възли се декларират като имащи 16 процесорни ядра, въпреки че операционната система GNU/Linux вижда 32 (логически) ядра.
Препоръчва се на потребителите да заявяват кратни на пълни възли, например, чрез добавяне

```
#PBS -l nodes=100:ppn=16
```

в скрипта чрез PBS можете да се изискат 100 изчислителни възела за ексклузивен достъп.
Тъй като опцията за HyperThreading е включена в конфигурацията на възлите, възможно е да се използват до 32 логически процесора на сървър, т.е. да се увеличи броят на използваните MPI процеси или OpenMP нишки до 32 вместо 16 за всеки възел.
Въпреки това, най-добре е първо да се измери ефективността с използване на реалистични входни данни като база за сравнение, за да се види дали HyperThreading всъщност е полезен. И в двата случая, със и без HyperThreading, искането за възли към PBS ще бъде същото, но командния ред ще се промени.
Например, може да се сравни ефективността на една и съща програма testprogram със и без HyperThreading на 100 възела за измерване на времето за изпълнение:

time mpirun -f hostfile -np 1600 -ppn 16 ./testprogram

спрямо

time mpirun -f hostfile -np 3200 -ppn 32 ./testprogram

Файлът с име `hostfile` може да съдържа списък на всички възли в задачата. Това може да бъде постигнато по този

cat $PBS_NODEFILE| sort |uniq > hostfile

Ако се комбинират OpenMP и MPI, следва да се използва команда от типа на

time mpirun -f hostfile –genv OMP_NUM_THREADS 16 -np 100 -ppn 1 ./testprogram

където се задава променливата на средата за всички процеси OMP_NUM_THREADS да бъде равна на 16.
Ако искаме да тестваме с 32, трябва съответно да стартираме

time mpirun -f hostfile –genv OMP_NUM_THREADS 32 -np 100 -ppn 1 ./testprogram

Понякога може да бъде по-добре да се стартират по 2 MPI процеса на възел, всеки с по 8 или 16 нишки, т.е. да се сравнят

time mpirun -f hostfile –genv OMP_NUM_THREADS 8 -np 200 -ppn 2 ./testprogram

и

time mpirun -f hostfile –genv OMP_NUM_THREADS 16 -np 200 -ppn 2 ./testprogram

Моля, имайте предвид, че с 32 MPI процеса на възел всеки процес получава само половината от паметта по подразбиране. В клъстера Авитохол има 150 изчислителни възли с по 64 GB памет. Следователно може да се използват по 2 GB на MPI процес при използване на HyperThreading или 4 GB на процес без използване на HyperThreading.

Най-често се използва паралелната среда за Intel MPI. Можете да използвате и други библиотеки за MPI. Например, openmpi също е на разположение на цялата система.

Към всеки изчислителен възел има асоциирани два Xeon Phi копроцесора. Например, на възел **sl039** съответстват копроцесори, достъпни чрез SSH или `mpiexec`, с имена **sl039-mic0** и **sl039- mic1**.

Всеки копроцесор работи със своя собствена операционна система, което означава, че е видим като отделна машина.

За всеки изчислителен възел съществуват алтернативни имена, които съответстват на алтернативни мрежови интерфейси, свързани с InfiniBand картите. Например, **ibsl039** съответства на IP-over-InfiniBand мрежов интерфейс ib0 на сървър **sl039**.

Същият тип IP-over-InfiniBand интерфейси са достъпни на копроцесорите, например **ibsl039-mic0** и **ibsl039-mic1**.

Въпреки, че тези интерфейси предлагат по-висока честотна лента от стандартните Ethernet интерфейси, те не са необходими за нормална употреба, защото чрез MPI се използвате InfiniBand директно, а не IP-over-InfiniBand.

Файловата система Lustre използва директно InfiniBand и тъй като файловете са споделени между възлите и то е дори на разположение на копроцесори, необходимост за движение на данните между възлите чрез интерфейса IP-над-InfiniBand може да е индикация за неоптимално използване на системата

По InfiniBand се пренасят и данните при MPI задачи, включително когато се използват копроцесорите.

След като даден потребител е получил изключителен достъп до възел за изпълнение, той или тя може да използва съответните Xeon Phi копроцесори. Вътре в скрипта на задачата може да се прочете съдържанието на променлива на обвивката PBS_NODEFILE, която посочва файл, съдържащ имената на съответните възли. Достъпът до копроцесорите Xeon Phi не се управлява отделно.

Потребителите нямат право да се логват с SSH към възли, където нямат работещи задачи. За да се определи кои възли са били разпределени за дадена задача, потребителят може да изпълни

```
qstat -na <jobnumber>
```

Потребителите не се очаква да използват Xeon Phi копроцесори, без първо да получат достъп до съответния основен възел.

Допълнителна информация

За повече информация относно използването на системата за управление на задачите Torque/Moab моля вижте документацията от интернет страницата на доставчика на софтуер: **www.adaptive computing.com**.

4.2. Компилатори

Компилаторите на Intel за Fortan (`ifort`) и C / C ++ (`icc, icpc`) се предполага да се използват най-много поради това, че са оптимизирани за Xeon Phi.

За да компилирате и свързвате MPI програми, може да използвате командите `mpiifort, mpiicc` или `mpiicpc`, съответно.

Колекцията от GNU компилатори (`gcc, g++, gfortran`) също е на разположение. За съжаление, версията по подразбиране, която идва с операционната система, е 4.4.7, която е сравнително стара. За по-новите версии може да се провери чрез системата за модули. За да компилирате и свържете MPI кодове, използващи GNU компилатори, но все пак използвайки Intel MPI библиотека, използвайте командите `mpicc, mpic++/mpicxx, mpif77` или `mpif90`, съответно.

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

За да заредите средата за развитие за Xeon Phi може да изпълните

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

Тогава инструментите за развитие стават достъпни с команди като:

```
k1om-mpss-linux-gcc

k1om-mpss-linux-g++

k1om-mpss-linux-nm
```

Изпълнете командата **module avail**, за да се направите преглед на всички налични компилатори и версии на разположение на Авитохол.

Обвивка с модули (Modules environment)

- За повече информация относно общото използване на модули моля вижте ръководството на PRACE Generic x86 Best Practice [http://www.prace-ri.eu/IMG/pdf/Best-Practice-Guide-Generic-x86.pdf].
- За повече информация относно използването на MPSS справка с документацията от Intel [https://software.intel.com/en-us/articles/intel-xeon-phi-coprocessor-developers-quick-start-guide]

4.3. Паралелно програмиране

На Авитохол има два основни метода за паралелно изпълнение.

MPI стандартът осигурява максимално ниво на преносимост, като използва разпределена памет и може да се използва с голям брой възли.

OpenMP е стандартизиран набор от директиви на компилатора за машини със споделена памет. На Авитохол с използване само на OpenMP може да се работи само на един възел (основен възел или копроцесор).

Възможно е да се смесват MPI и OpenMP паралелизъм в един и същи код, за да се постигне максимална производителност в голям мащаб.
Поради наличието на Xeon Phi ускорители, MPI и OpenMP също са на разположение на копроцесорите, и всички начини на използване на копроцесорите - `native`, `symmetric` и `offload`, са на разположение.
Всеки Xeon Phi копроцесор може да изпълнява програми с OpenMP в директен режим. Броят на физическите ядра на копроцесор е 61. Препоръчително е да запазваме едно ядро за използването от операционната система и да се използват най-много 60 ядра. Въпреки това, броят на логически ядра на копроцесор е 4 пъти броя на физически ядра, т.е. 244. Опитът показва, че използването на брой нишки, по-висок от броя на физическите ядра може да бъде полезно. Потребителите могат да се опитат да открият какви настройки на броя на нишките за OpenMP чрез променливите OMP_NUM_THREADS (в директен режим) или MIC_OMP_NUM_THREADS (в режим на offload) е най-добър. В много случаи 120 изглежда като добър компромис.
За програмите, които се изпълняват на основния възел на системата, броят на нишките може да се настрои да бъде равен на броя на физическите CPU-ядра (16) или броя на логическите ядра (32), ако се използва само OpenMP. Ако хибридна MPI / OpenMP програма се изпълнява, трябва да се вземе предвид как много процеси MPI се изпълняват на един възел. Например, ако две MPI процеси се изпълняват на всеки възел, тогава OMP_NUM_THREADS трябва да бъде настроен на 8 да използвате всички физически ядра или 16, за да използвате всички логически ядра.

4.4. Паралелни MPI приложения

Компилаторите на Intel за Fortran / C / C ++ са налични по подразбиране на Авитохол.
Командите са съответно `mpiifort`, `mpiicc` и `mpiicpc`. Тези команди фактическа са wrappers за извикване на съответните компилатори и включват информация за свързване на правилните библиотеки за MPI.
Паралелните програми може да се стартират с `mpiexec`. Припомняме, че променливата на обвивката може да се предава чрез опции. Например, за да зададете променлива MKL_MIC_ENABLE да бъде 1, като по този начин се дава възможност за автоматично използване на offload за всички MPI процеси, може да добавите

-genv MKL_MIC_ENABLE 1

към командния ред на mpiexec.

Система за управление на задачите (Batch System)

MPI програмите трябва да се пускат като работни изчислителни задачи чрез системата за управление на задачите. Моля, вижте раздел 4.1, "Система за изпълнение на задачите" за допълнителна информация.

Променливи на обвивката (Environment variables)

Някои променливи на обвивката могат да дават информация за софтуер, който не идва директно от дистрибуцията на операционната система. Примери за това са някои инструменти като `ant/maven`, които са предоставени.

4.5. Multithreaded (OpenMP или хибридни MPI/OpenMP) приложения

За да компилирате и свържете OpenMP приложения използвайте опция `-qopenmp` (която заменя предишната опция `-openmp`, който сега е `deprecated`) на компилатора Intel, и `-fopenmp` за GCC компилаторите.

В някои случаи е необходимо да се контролира размерът на стека по време на изпълнение, например когато приложението излиза с грешка "`segmentation fault`". На Авитохол при използване на компилаторите на Intel размерът на стека е зададен чрез променливата KMP_STACKSIZE на околната среда. Стойността по подразбиране е 4 мегабайта (4MB). Например, за да направите заявка за размера на стека от 128 мегабайта на Авитохол, определете KMP_STACKSIZE = 128m.
Следва да се има предвид, че на копроцесора практическата граница за брой нишки е 244 (61 хардуерни нишки по 4 заради HyperThreading). По този начин с помощта на голяма стойност на KMP_STACKSIZE може да се получи срив на задачата.

Забележка

Приложенията понякога може да използват `swap` памет. По принцип такова поведение при изпълнение е нежелателно и следва да се избягва. Предполага се, че то е в резултат на прилагане на погрешна конфигурация или грешка във входните данни. От потребителите се очаква да

спират такива задачи и да се опитват да поправят грешките. Ако те смятат, че това е нормално поведение, следва да обсъдят това със системните администратори.

За информация за съставяне на приложения, които използват `pthreads` моля вижте документацията на Intel [http://software.intel.com/en-us/articles/intel-c-composer-xe-documentation/#lin].

4.6. Използване на MKL математическата библиотека на Авитохол

Математическата библиотека на Intel (MKL) е налична на Авитохол. MKL предоставя високо оптимизирани реализации на

- LAPACK/BLAS,
- директни и итеративни солвъри,
- FFT,
- ScaLAPACK
- генератори за случайни числа и много други

Части от библиотеката са приспособени за многонишково изпълнение или за използване на разпределена памет.
Чрез задаване на променливата `MKL_MIC_ENABLE` на средата да бъде 1, например, чрез
`MKL_MIC_ENABLE = 1`
може да се даде възможност за автоматично прехвърляне на натоварването на някои библиотечни функции за копроцесорите на Xeon Phi, които са налични. За използване на тази опция с MPI, виж по-горе.

Забележка

Обширна информация за характеристиките и използването на MKL се осигурява от официалната документация на Intel MKL [http://software.intel.com/en-us/articles/intel-math-kernel-library-documentation/].

За автоматичния offload може да се прочете в:

https://software.intel.com/sites/default/files/11MIC42_How_to_Use_MKL_Automatic_Offload_0.pdf

Свързване на програми с MKL

По подразбиране, модулът за MKL е зареден. Модулът настройва променливите на обвивката MKL_HOME и MKLROOT до инсталационната директория на MKL. Тези променливи могат да бъдат използвани в `Makefile` и скриптове.
Чрез Intel MKL Link Line Advisor често е полезно да се получи информация за това как да се свържат програми с MKL. Например, за да се свържат статично с многонишковата версия на MKL на Авитохол (Linux, Intel64):

```
-Wl,--start-group
$(MKLROOT)/lib/intel64/libmkl_intel_lp64.a
$(MKLROOT)/lib/intel64/libmkl_intel_thread.a
$(MKLROOT)/lib/intel64/libmkl_core.a
-Wl,--end-group -lpthread -lm –qopenmp
```

(на един ред в Makefile). Ако съставяне командата в bash обвивката, следва да се използва $ {MKLROOT} вместо $ (MKLROOT).

4.7. Изчислителни библиотеки

FFTW

Библиотеките fftw са инсталирани и конфигурирани. Версията на FFTW 3, която идва с операционна система е 3.2.3 и се предлага в стандартните места за разработка на софтуер.

PETSc

Комплект от структури от данни и рутинни процедури за мащабируеми (с използване на MPI) програми за решаването на научни приложения, моделирани от частни диференциални уравнения. Предлага се като модул.

SLEPc

Библиотека за решаване на мащабни задачи с разредени матрици и собствени стойности на паралелни компютри. Предлага се като модул.

GSL

GNU научната библиотека (GSL) е библиотека за C и C ++ програмисти (FORTRAN-addon интерфейсът FGSL е също инсталиран). GSL предлага широка гама от математически функци и процедури за генератори на случайни числа, специални функции и метод на най-малките квадрати и много други. Предлага се като част от стандартната дистрибуция.

Python

Версията на Python от дистрибуцията на операционната система е от серията 2.6. Последната настояща версия от 2.7 серия е достъпна като модул. Модулът за tensorflow се предлага като част от инсталацията 2.7.

4.8. Входно-изходни библиотеки

NetCDF

NetCDF е набор от софтуерни библиотеки и протоколи за обработка на данни в машинно-независими формати, които подкрепят създаването, достъпа и обмена на големи масиви от научни данни. Предлага се като модул (версия 4).

4.9. Разнообразни библиотеки

1. Boost: библиотеките boost за C ++ осигуряват много класове за различни приложения. Версията на boost, предоставени от операционната система, е доста стара. По-нова версия е достъпна като модул.
2. TBB: Intel Threading Building Blocks. TBB дава възможност на C ++ програмистите да използват техники за паралелно изпълнение в кода. Предлага се като част от Intel Compiler Suite.
3. IIP: Intel Integrated Primitives. В IPP се съдържат силно оптимизирани примитивни операции, които могат да се използват за цифрово филтриране, аудио и обработка на изображения.
4. PAPI е библиотека за за четене на броячи за събития, за оценка на производителността по преносим начин.

4.10. Друг софтуер

Други популярни пакети са на разположение за цялата система. Например - `R, cmake, gnuplot`.

Модули на средата

- Моля, използвайте **module avail** на Авитохол, за да получите цялостен списък на системни модули.
- В повечето случаи е налична само една препоръчителна версия.

5. Средства за разработка

За достъп до софтуера, описани по-долу, моля използвайте командите **module avail**, **module load**.

5.1. Отстраняване на грешки (debugging)

- Опции на компилатора: Компилаторите обикновено имат някои опции, удобни при процеса на отстраняване на грешки. Моля, консултирайте се с документацията на компилатора за подробности. Обикновено опцията –g трябва да се използва.
- GDB - GNU дебъгер.
- Intel inspector - дава възможност за отстраняване на грешки на многонишкови приложения.
- След като грешките са отстранени, може да се използва командата **strip** върху програмата, за да се отстрани излишна информация.

5.2. Средства за профилиране и анализ на производителността

- Intel VTune ampliefier е мощен инструмент за анализиране на изпълнение на код.
- gprof - GNU profiler.
- Intel Trace Analyzer and Collector е инструмент за профилиране на MPI комуникациите.
- Scalasca дава възможност за анализ на MPI / OpenMP / хибридни кодове.
- Оптимизирани изпълними файлове могат да бъдат получени като първо се състави профил (опция –prof-gen за Intel, --fprofile-generate на GCC), изпълни се тестова задача и след това се използва генерираният профил с опции, като
    - -prof_use -prof_dir ./profdir за Intel
    - "-fprofile-use - за gcc, като понякога е необходимо да се добавят -fprofile-correction или -flto-division = none.

Допълнителна информация

За повече информация, свързана с раздел 5, моля вижте уебсайта на Авитохол:

• http://www.hpc.acad.bg/systems/avitohol/index.php/system-1

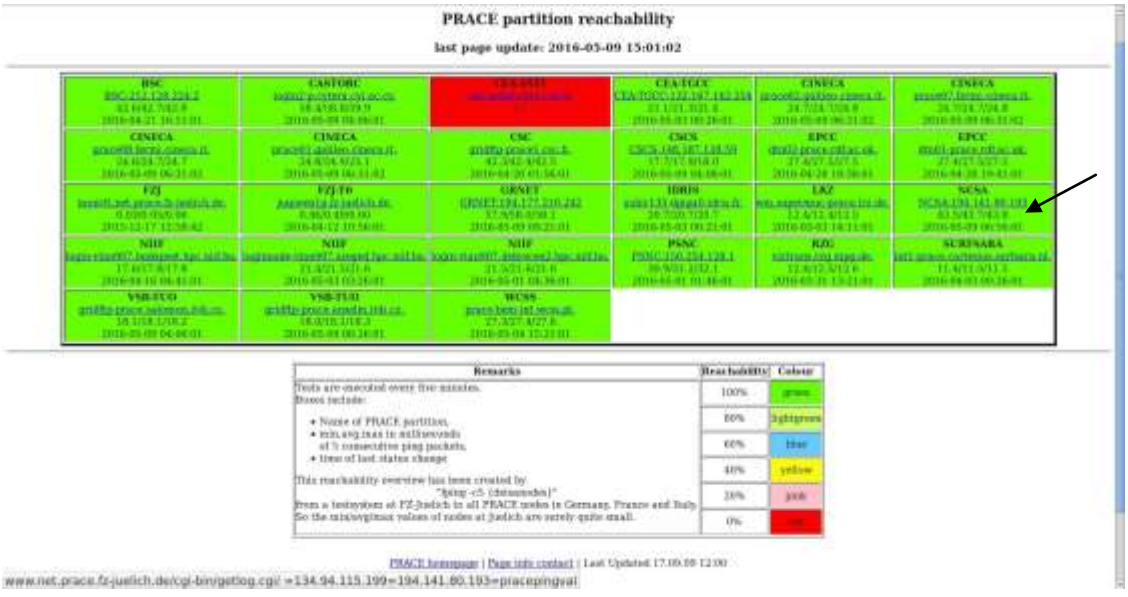
Глава 2. Услуги и мидълуър за свързване към PRACE като Tier-1

Основните задачи на пакета WP6.1: Operation and coordination of the comprehensive common PRACE operational services (Опериране и координация на цялостните общи оперативни услуги на PRACE) включват:

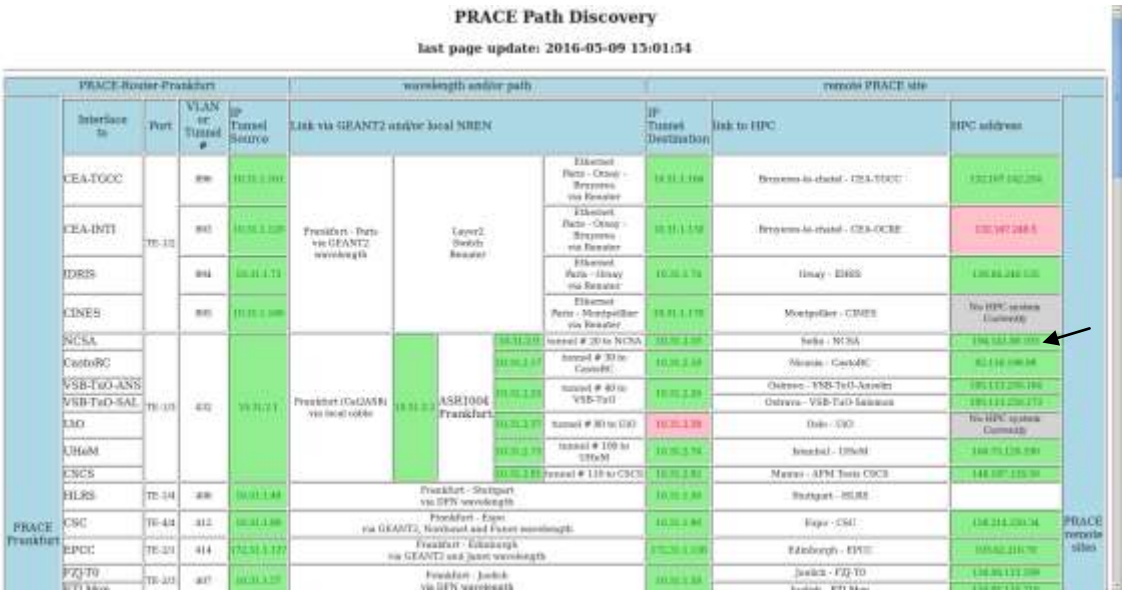
- Разполагане и поддръжка за услугите за данни (GridFTP)
- Услуги за управление на ресурсите (напр. UNICORE и GLOBUS GRAM)
- Хармонизирани процедури за получаване на автентикация и оторизация (контрол на достъпа)
- Поддръжка за инфраструктурата с публични и частни ключове (PKI), администриране на потребителите, поддръжка за GSI-SSH
- Поддръжка за потребители и приложения
- Мониторинг на операциите
- Участие в регулярните дежурства HoD и OoD, създаване и проследяване на trouble tickets.

Резюме на извършената работа през периода по задача 6.1:

Извършено е тестване на разположената в рамките на предишните проекти криптирана мрежова връзка относно възможностите за прехвърлянето й към новата система Авитохол. Изпълнени са необходимите физически връзки и е извършена преконфигурация, така че вече има достъп от новата Tier-1 система. Мониторинг-тестовете, които се извършват по проекта, вече показват положителен резултат. Това се вижда на следващите две фигури (със стрелка е отбелязано NCSA).



Фигура: Екран от системата за наблюдение на мрежовата достъпност на PRACE



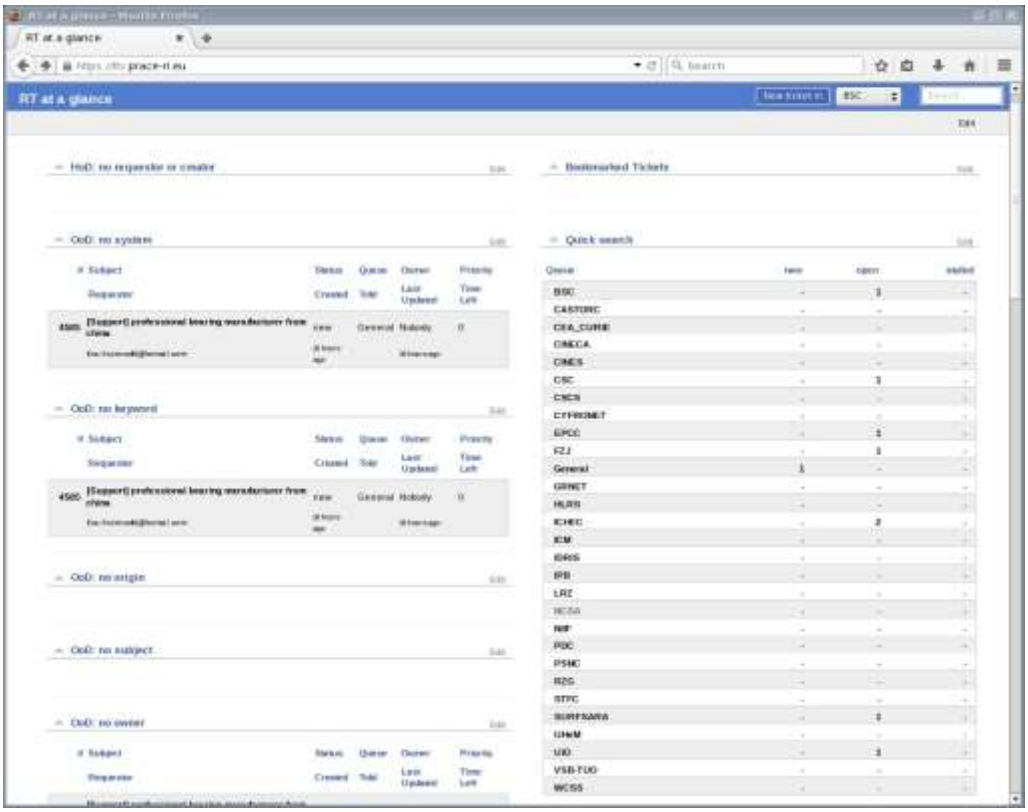
Фигура: Екран от системата за наблюдение на мрежовите пътища на PRACE

Извършено е планиране на дисковото пространство, необходимите услуги и софтуерна поддръжка за осигуряване на достъп по протоколите GridFTP, UNICORE, GLOBUS – GRAM и GSI-SSH.

Разположени са сървърите, които да поддържат тези услуги, осигурен е достъп до тях по криптираната мрежа на PRACE. Осъществен е достъп от тези сървъри до общата файлова система Lustre и системата за обработка на задачите Moab (показано на фигурата).

Фигура: Екран от системата за управление на заданията

Българският екип се включи пълноценно в регулярните дежурства HoD (Helpdesk on Duty) и OoD (Operator on Duty), по време на които се наблюдава състоянието на цялата инфраструктура на PRACE, създават се и се проследяват trouble tickets и се координира поддръжката на потребителите. На по-долната фигура се вижда уеб-интерфейсът за работа по време на дежурствата.



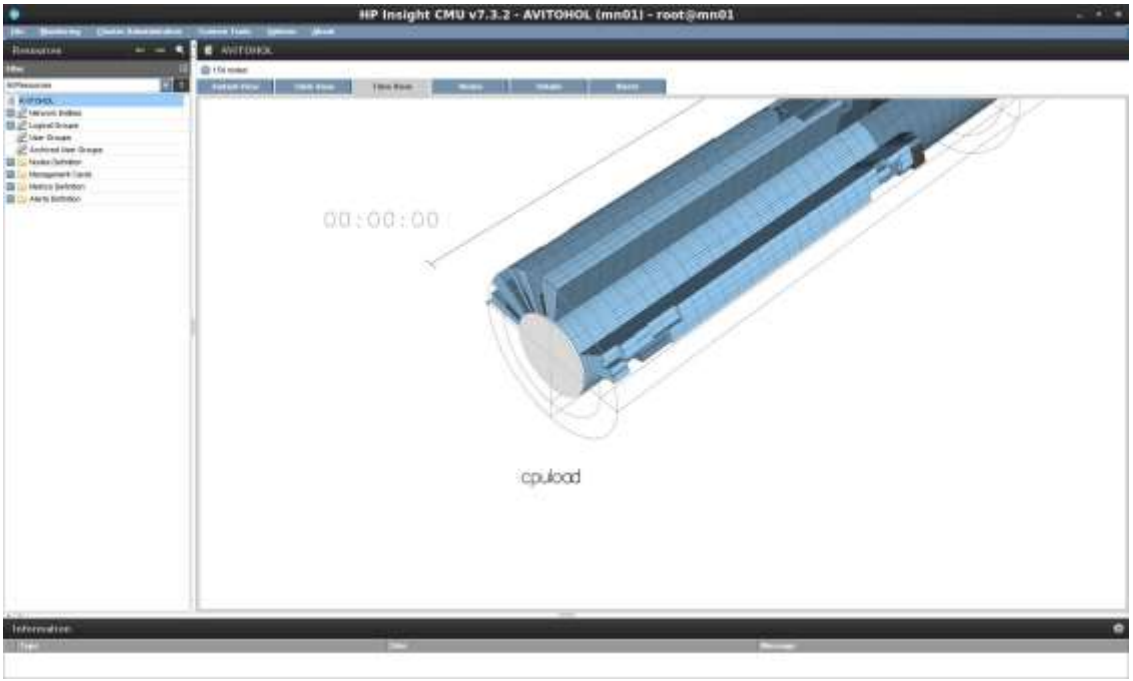
Фигура: Екран от системата за наблюдение на създаване и проследяване на trouble tickets на PRACE

Хармонизирани са политиките и процедурите за достъп и за допустима употреба (Acceptable use) съобразно с политиките на проекта PRACE.

Създадени са потребителски акаунти, осъществена е поддръжка на потребителите и е инсталиран софтуер, необходим за изпълнението на задачите им, осигурено е издаването и обновяване на X509 сертификати.

Извършени са регулярни обновявания на базисния и приложен софтуер и наблюдение за сигурността.

По-надолу са показани различни уеб-страници, в които администраторите могат да следят натоварването на системата.



Фигура: Екран от системата за наблюдение и контрол CMU

Фигура: Екран от системата за наблюдение на натоварването

6. Заключение

Системата Авитохол е сравнително нова, но вече е оборудвана с богат набор от инструменти за разработка, библиотеки и софтуер. Потребители, които изискват допълнителни инструменти или софтуер да бъде инсталиран или обновен на съществуващ софтуер трябва да се свържат с екипа за поддръжка на **avitohol-support@parallel.bas.bg**.