Minerva User Survey Comments and Response

2019-01-31

System Issues

- Slow/overloaded login nodes:
  - We implemented two data transfer nodes (data2 and data4) for people to run heavy I/O operations, for example, scp, ftp, aws upload, TSM, etc.
  - We restricted users from running jobs on the head nodes. Long process will be killed and user will be contacted.
- Minerva has been down and/or unstable for many periods. At times, jobs got killed, user kicked off the login nodes, file system is not accessible.
  - We are aware of a couple of issues that have caused system issues. We resolved most of the issues and/or took actions to reduce the system failure.
    - Allocation manager (GOLD) failure causing jobs get killed.
      - We optimized GOLD database, increased the process limit, constructed a new expressalloc queue was added to avoid GOLD authentication and reconfigured other queues to be able to requeue the failed jobs.
    - A file corruption in orga causing login nodes reboot while accessing that folder.
      - We removed this corrupted file with the help of IBM.
    - LSF not responsive while some user submit millions of short jobs.
      - We put restrictions on a couple of users and contact users when we observe harmful behavior.
    - Stale file handler of GPFS file system when a node's memory is fully used.
    - When the memory of the login nodes is overused, it triggers the Out of Memory (OOM) killer to kill random process. System admins have to reboot the node to restore its functionality.
      - We restricted memory usage on the login nodes for all users, throttled aws upload bandwidth for its memory usage.
- There are too many maintenance disruptions.
  - In 2018, we had a couple of major tasks: to upgrade the GPFS code level, to integrate the newly purchased storage ESS to orga, and to upgrade OS. Some of the operations, such as network reroute, configuration changes in storage and OS, will affect the production system and can only be performed in a maintenance period.

    - In 2019, our major task will be put the new compute partition Chimera in production. We will need to schedule necessary maintenance accordingly, but we will keep in mind to minimize the number of PMs and send out announcements 2 weeks ahead.

Queues

- The interactive queue needs to get higher priority and faster to access. Interactive nodes with outside internet access would be nice to have.
    - We are looking into ways to reduce the waiting time to get into the interactive nodes through the job scheduler.
    - The interactive nodes in the new compute partition Chimera will have external internet access.
    - We will implement an interactive GPU node in Chimera to make the debug of GPU code easier.
- Minerva is slow. Wait times of jobs are often too long.
    - In the end of 2018, we purchased a new compute partition which contains a total of 14,304 cores. Once it is in production (estimated in 2019 Q1) the waiting time of jobs should be greatly reduced.

Job scheduler

- DRMMA-compatible job submission system (http://www.drmaa.org)
    - /hpc/lsf/9.1/linux2.6-glibc2.3-x86_64/lib/libdrmaa.so is the location of the drmaa library
    - Corrected a bug in python so that “import drmaa” now works without error message.
- A job scheduler that can handle multi-node spark would be good (I believe later versions of LSF, but not the current one, can)
    - Implementation planned when LSF 10 is installed.
- It would be great if the killed job information can be sent to the user.
    - Mail to user function is turned off globally in LSF since some of the users has millions of job arrays. However, you can manually check the state of your job and set up the email alert by adding following line in your job script:
      echo output > mail -s “job complete” your-email
- Create a checker that makes sure that a job is "runable." For example, if you've requested resources that do not exist (like 1TB of total memory), kill the job with an error notes why.
    - We already have some check on mandatory parameters when submit a LSF job, such as walltime(-W), project account (-P), and queue (-q). The

LSF job will be killed immediately if these parameters are missing. For compute resources, you will have to refer to our specs of the compute nodes (https://hpc.mssm.edu/resources/hardware). We will also add some general guidelines for the choices of compute resources on our website and tutorial sessions.

Storage

- Need more storage space! scratch run out of space easily.
    - Storage space (/sc/orga) is currently sufficient for all users. The total disk space is 9.1PB with 3.7PB (40%) free.
    - We are in the process of proposing to Scientific Computing Committee to adjust the fee for grants. The price will most likely drop since we have adequate storage space.
    - We will implement per-user space limit on scratch.
- File system problems lagging was a significant issue last year.
    - During 2018 we had to perform some critical updates on the file system software (GPFS) and we had to integrate the new IBM ESS flash servers into the existing storage. These activities had some impact (no access to the file system, NFS Stale Handle from the users side) also outside the maintenance windows due to the complexity of our cluster (different types of nodes, connections and network configurations), but we alway tried to resolve the issues as soon as possible.

TSM

- Retrieving from TSM is slow.
    - We currently have a total of 16.7PB data on TSM which sits on more than 12,000 tapes. Unfortunately, the current TSM library is not big enough to accommodate all these tapes. When a retrieve command is evoked, the operators have to fetch the tape load manually into the library. The responding time dependents but should be less than 1.5h. Once the tape is loaded, the data transfer starts at a constant rate of 150M/s.

Software/packages

- There are no modules on data2 or data4 so I can't use python3 to manage my download scripts.
    - Since data2/data4 nodes are dedicated to data transfer, we didn’t explicitly link the module path to the major packages on these two nodes. But you can set it up by

$ module use /hpc/packages/minerva-common/modulefiles".
$ module load python/3.6.2

We will announce and document this on our website.

- Support for pipeline frameworks like Snakemake. Support for conda. Support for Jupyter Notebooks.
    - Snakemake is available via python packages. To access, use
      $ module load python/3.6.2
      $ module load py_packages/3.6
      $ which snakemake
      /hpc/packages/minerva-common/python/3.6.2/bin/snakemake
    - Conda and Jupyter Notebooks are available in Minerva. Please access through module system:
      $ module load anaconda2(3)
      $ module load jupyter
- Support for shinyapps.
    - Thanks for the suggestion. Unfortunately, we don't have a proper setup tailored for running these servers stably on Minerva at this time. Minerva is mostly just for batch processing. You may need to contact the IT team if you need to run the shiny server program. They may set up a VM on their server to handle it.
- Support for RStudio Server.
    - We don't have an RStudio server directly for you to use. But, we have RStudio client, which can be launched by the command line and it will open a window if you enabled X-Forwarding. Wonder whether it is enough if you just use RStudio via X-Forwarding over SSH.
- Support for docker, singularity, kubernetes user-level environments/containers.
    - Singularity is available in Minerva, currently on data2 and compute nodes in test-centos7 queue. It will be available on all nodes in Chimera compute partition when it was in production.
- Capability for GUI desktop, Make available remote desktop software so people can use GUI software interactively. Using ssh -X is too slow.
    - We are considering setting up a NX like service that is dedicated to GUI softwares in the Chimera partition.

Tickets

- The HPC staff is amazing, incredibly responsive and helpful. The training sessions and information from the slides are very helpful. However, the response time to package and software request could be reduced.

    - We will try our best to reduce the time taken for packages/software install. However, it really depends on the packages requested. Usually, we finish most of the package/software requests in hours. For the packages that is not well-written, or well-suited for running on cluster/Linux system, it will need some extra time to tweak around. Also some packages have a list of dependencies to address, which needs extra time.

Tutorial/Training

- A tutorial would be amazing.
    - We offered three tutorials last year, Introduction to Minerva and Advanced LSF. We will continue to schedule tutorials this year.
    - Slides for the tutorial are available online https://hpc.mssm.edu/docs/mug.
    - If your research group would like to schedule an addition tutorial or tutorials focused on special HPC topics, you can reach out to our computational scientists (hpchelp@hpc.mssm.edu).

Documentation

- Better website documentation is needed severely.
    - Documentation is clearly insufficient. We have started a project to revamp and update of the documentation on Minerva's website.
- Information that would be helpful include:
- LSF quick start guide and LSF documentation include flags that do and do not work on Minerva.
    - We will update the LSF documentation accordingly.
- An explanation to the module system, instructions on how to install R packages, python modules and other modules.
    - Most of the packages are available in our module system, and the instruction is available at https://hpc.mssm.edu/about/modules. If you would like to build your own packages, please refer to the software's website. We will also update the software-related documentation accordingly regarding to how to build your own module..
- Better web documentation for daily things. I for example still don't know how to load git repositories into my workspace. Give more instructions on how to run deep learning packages.
    - For most packages and software, user will have to refer to the user guide on the corresponding website. We will try to put in a link for those

packages and states its special environment settings or configuration flags in Minerva.

- Submitting jobs to the interactive node and to the GPU nodes.
- Could use improved clarification on which nodes are available and how to decide between them.
    - This information as well as other most recent updates are available in the Weekly Newsletter (subscribe to hpcusers@mssm.edu). We will update the documentation in a timely manner to reflect the changes in our system.

Cloud vs Minerva

- From a *brief* check seems like the costs are comparable if not higher than AWS. If costs are higher than cloud services it would be logical to switch to cloud.
    - We did an On-premise vs Cloud analysis for Minerva workload and you can refer to https://hpc.mssm.edu/files/CloudAnalysis.pdf for detailed comparison. In summary, it would cost $2.8 million per year for Minerva's actual compute core current usage alone (57.5 million core hours). The storage in cloud is 6-8 times more costly than Minerva, let alone other service costs such as network traffic to download data. It would take ~1 year to migrate all the data and rebuild packages on Cloud.

Other Comments

- The environment can't support the big data science that we want to do It is unfortunate that we spent $$ million for data generation and we have to find other ways to analyze data.
    - We will need more information on this item. Please contact the folks at Scientific Computing for assistance in getting your analyses to run.