No. 194
December 2016

Bits & Bytes

Max Planck Computing & Data Facility (MPCDF)*
Gießenbachstraße 2, D-85748 Garching bei München

HPC

Ingeborg Weidl, Markus Rampp, Renate Dohmen

SLURM as batch system

The new HPC extension cluster *draco* with ca. 800 Haswell nodes (Xeon E5-2698v3 with 2 x 16 cores) is operated with the Intel toolchain and SLURM as a batch system. The recommended MPI launcher is srun and not mpiexec or mpirun. In what follows we give some hints on how to operate common Intel tools like itac, mpi checking, etc. using srun and how to manage array jobs.

SLURM and Intel tools

The equivalent for 'mpiexec -trace' can be invoked with srun by preloading the ITAC library libVt.so like this:

```
$ module load itac
$ export LD_PRELOAD=$ITAC_HOME/intel64/slib/libVT.so
$ srun ...
```

The equivalent for 'mpiexec -check_mpi' is generated by preloading the library libVTmc.so:

```
$ module load itac
$ export LD_PRELOAD=$ITAC_HOME/intel64/slib/libVTmc.so
$ srun ...
```

SLURM and array jobs

SLURM (like Sun Grid Engine, but unlike *hydra's* LoadLeveler) supports so-called array jobs which is a convenient and efficient way to handle a collection of multiple *independent* jobs of the same type, for example to run parameter scans with a large number of input parameters. A SLURM batch job that is submitted with the command line option 'sbatch --array=0-31' (or a corresponding line in the header of the submit script) gets replicated to run 32 instances of the same script under a single job id. Within the batch script the value of the environment variable $SLURM_ARRAY_TASK_ID can be used to identify the instance. As an illustrative example, input files could be named input_0, input_1, ..., input_31 and a corresponding executable statement in the batch script could read like 'srun ./my_program --input-file=input_$SLURM_ARRAY_TASK_ID'.

The --array option argument can be a range of index values (like in the example above) and an optional step size (like --array=0-63:2), or a specific array of index values (like --array=1,3,9,42). For specifying the job's stdin, stdout, and stderr file names, %A will be replaced by the value of SLURM_ARRAY_JOB_ID (the single id of the job array) and %a will be replaced by the individual values of SLURM_ARRAY_TASK_ID (see above). Comprehensive documentation can be found in the SLURM documentation.

Note that this technique is *not* suited for handling jobs with dependencies. Please consult the MPCDF webpage -⟩ sample batch scripts for draco for solutions to handle those.

Interactive queues on hydra and draco

Both on *hydra* and *draco* means are provided to run jobs interactively for testing purposes.

Interactive jobs on hydra

To run a program interactively on *hydra* you have to login to one of the interactive nodes *hydra-i.rzg.mpg.de* (*hydra03* – *hydra08*). You have to create a so-called 'host list' file (default filename is host.list) that contains the line 'localhost' for each cpu you want to use. For exam-

*Tel.: +49(89) 3299-01, e-mail: benutzerberatung@mpcdf.mpg.de, URL: http://www.mpcdf.mpg.de/

Editor: Renate Dohmen

ple, if you want to test your code on 4 cpus, the host list file has to look like this:

```
hydra04:> cat host.list
localhost
localhost
localhost
localhost
```

Run your program:

```
poe ./myprog -procs 4
```

Please, limit the runtime of your interactive jobs to 2 hours at most. And please, take care that the machine does not become overloaded. Don't occupy more than 8 cpus and please do not allocate more than 30 GB of main memory. Neglecting these recommendations may cause a system crash or hangup!

If you need more cpus or even 1 or 2 full nodes for testing your code interactively, you can use the batch class (queue) 'inter_class' on *hydra*. Create a file (e. g. with filename LL_FILE) that contains the following LoadLeveler directives:

```
# @ job_type = parallel
# @ node = 1
# @ tasks_per_node = 16
# @ wall_clock_limit = 1200
# @ resources = ConsumableCpus(1)
# @ class = inter_class
# @ queue
```

Run your program using the above file 'LL_FILE':

```
poe ./myprog -rmfile LL_FILE
```

This gives you an interactive session on a SandyBridge compute node. Thus, this mode is also useful for debugging your code. The runtime in the 'inter_class' is 20 minutes at most.

Interactive jobs on draco

To run a program interactively on *draco* you can login to one of the interactive nodes *draco-i.mpcdf.mpg.de* (*draco03*, *draco04*). A parallel interactive job can simply be started with the command 'srun -n ⟨number of tasks⟩ ⟨program⟩', e. g. on 4 cpus:

```
srun -n 4 ./myprog
```

It will use the 'interactive' partition by default which runs on the nodes *draco03* or *draco04*. The runtime limit of the 'interactive' partition is 2 hours.

Galaxy instance for the MPG

Mykola Petrov, Markus Rampp

Galaxy (https://galaxyproject.org/) is an open-source, web-based platform for data-intensive biomedical research. Galaxy has become a popular platform in computational biomedical research for defining, executing, sharing, teaching and publishing workflows, such as large-scale genomic analysis or *RNAseq* pipelines. Researchers can use a public Galaxy service available at https://usegalaxy.org/ or install the software locally and operate their own 'instance' of Galaxy. Operating one's own Galaxy instance allows researchers, teams or organizations to provide a tailored set of pre-installed and pre-configured tools and workflows, and, in particular, to connect it to their own dedicated compute and storage resources.

In order to provide such a service for the Max Planck Society, the MPCDF, in collaboration with the Max Planck Institute for the Biology of Aging has deployed a Galaxy instance for the MPG. The service is available at https://www.bioinfo.mpg.de/galaxy and is open to all researchers of the Max Planck Society. Accounts can be requested by sending an informal E-mail message to bioinfo-help@mpcdf.mpg.de. A complete *RNAseq* pipeline has been implemented and tested, several other tools, e. g. the *MiModD* tool to identify mutations from Whole-Genome Sequencing data and the *CloudMap* tools for the analysis of mutant genome sequences are already installed. Additional tools and workflows can be provided on request.The MPCDF Galaxy instance is connected to a 38-node compute cluster with disk storage, the maximum run time for Galaxy jobs is 7 days. Resources can be prioritized or dedicated for running Galaxy jobs on request.

Python infrastructure

Klaus Reuter

Python in tandem with the SciPy stack has become a popular platform for scientific computing and data analysis. The default Python environment on MPCDF systems is now the Anaconda Python distribution (https://www.continuum.io/anaconda-overview). Currently, the versions 2 (Python 2.7) and 3 (Python 3.5) are available.

Anaconda is a Python distribution that is available for free under a BSD-type license. It contains numerous packages for scientific computing and data analysis which are updated on a regular basis and which play well with each other. Performance-critical packages such as NumPy and SciPy are linked against the Intel Math Kernel Library (MKL) by default. At the MPCDF, a default version of Anaconda is provided via the environment modules 'anaconda/2' and 'anaconda/3'. For newer versions, we append the distribution version string explicitly, e. g. 'anaconda/2_4.1.1'. Use the command 'module avail anaconda' to get an up-to-date list of the versions installed. Anaconda replaces the older environment modules 'python27/*' and 'python33/*' which will not be upgraded anymore. Using 'pip' or 'conda', users are able to install software (and automatically resolved dependencies) themselves into their home directories in case a package is not provided by the central anaconda installation.

Visualization

Klaus Reuter, Ingeborg Weidl

Visualization environment on draco

Interactive GPU-accelerated visualization services are now also offered on a subset of the HPC extension cluster *draco*. Setup and usage is as on *hydra*. Documentation is available at http://www.mpcdf.mpg.de/services/visualization/draco_vis.

Interactive visualization sessions are requested using a special batch queue (partition) on *draco*, similar as for compute jobs. The initial setup including the creation of a sample submit script needs to be performed once by the user via the 'setup_remotevis' script provided by the 'remotevis' environment module:

```
$ module load remotevis
$ setup_remotevis
```

Below, the commands to start an interactive visualization session on *draco* are given:

```
$ cd ~/remotevis
$ sbatch remotevis.cmd
```

As soon as a visualization session starts on a GPU node, the user will receive a notification E-mail with instructions.

Recent optimizations of ELPA eigensolvers

Andreas Marek, Hermann Lederer

ELPA is a library of eigensolvers for symmetric dense matrices which is developed under the leadership of the MPCDF. The library has recently been enhanced through a GPU version. In collaboration with Nvidia, both the one-step solver elpa1 and the two-step solver elpa2 have been optimized to now also support Nvidia Kepler GPUs (K20, K40, K80, P100) with Intel Xeon processors as host. The actual ELPA software is installed on all HPC systems and major clusters at the MPCDF. The new software development has been partially funded through the BMBF project ELPA-AEO. Further ELPA enhancements include the provisioning of a much faster 32-bit version for those programs for which single precision is sufficient. Detailed information is available at http://elpa.mpcdf.mpg.de.

csvkit - A command line suite of tools for converting and working with CSV data

John Alan Kennedy

Comma-seperated values (CSV) data is both simple and ubiquitous.While there are many tools to process CSV data there are also times when you just need a simple command line tool. For such cases csvkit is a perfect fit. csvkit provides a suite of command line tools that allow you to convert between CSV and other formats as well as query and manipulate CSV data sets. csvkit is available as a Python package, so **installation** is as easy as

```
pip install csvkit
```

This will install the whole suite of command line tools. There are tools for importing, manipulating, displaying CSV based data. A brief overview of the commands follows:

```
in2csv   - convert data to csv                  csvclean - fix errors in csv
csvlook  - pretty printing of csv               csvcut   - filter and truncate CSV files
csvgrep  - search CSV using grep-like syntax    csvsort  - sort CSV data
csvstack - stack rows from multiple CSV files   csvjoin  - merge CSV datasets
```

An **example workflow** best explains how these tools work. Here we refer to a dataset people.csv containing information about a group of people.

1) Get your CSV data (or convert data to CSV using in2csv) assuming people.csv contains:

```
id,first_name,last_name,email,gender,ip_address,age
```

2) Have a quick look at the data

```
csvstat people.csv
csvlook people.csv
```

3) Trim the dataset down (including re-ordering)

```
csvcut -c age,first_name,last_name people.csv > mini-people.csv
```

4) Sort your data

```
csvsort -c age mini-people.csv > sorted-mini-people.csv
```

5) Take a look at your results

```
csvlook sorted-mini-people.csv
```

Note: csvkit allows you to pipe the output from one command into another. This means you can chain the above tools together to form more complex/powerful mini-pipelines. For example the above workflow could be combined into a single one-liner:

```
csvcut -c age,first_name,last_name people.csv | csvsort -c age | csvlook
```

Bridging the gap between CSV and sql:

There are times when you may need a little more power than the command line and CSV offers. In these cases you can easily move data between CSV and SQL databases or even choose to run on-the-fly SQL queries against a CSV dataset. Two tools sql2csv and csvsql will help you do this.

```
sql2csv  - execute SQL on database and produce CSV output
csvsql   - generate table from CSV or run SQL query against CSV file.
```

An example of a simple query on a CSV dataset again performing the same task as the above workflow:

```
csvsql --query 'select age,first_name,last_name from people order by age;' people.csv | csvlook
```

Note: On-the-fly SQL queries build a database in memory, be careful when using large datasets.

We've seen how the command line tools from csvkit can be used to view and manipulate CSV files and also interact with SQL-based databases. Although we covered a lot there are still a few tools we didn't get a chance to see. The full tool suite is easy to install and use, each command is well documented on the csvkit website and can be called with a '-h' help option to gain more info. csvkit really is the Swiss army knife of CSV for the command line – it's simple to install, easy to understand and makes life easier. For further info see the official csvkit website: `https://csvkit.readthedocs.io/`.

Events

Raphael Ritz, Hermann Lederer

National IT-Summit

For the 10th time the German Federal Government organized the National IT-Summit, where about 1000 representatives from politics, economy, science and education meet to discuss challenges and opportunities of the digital age. This year, the event took place in Saarbrücken on November 16th and 17th, 2016 focusing on educational issues in particular.

Nationaler IT Gipfel
Saarbrücken 2016

In this context, the MPCDF is leading a working group on 'Intelligent and efficient use of open data in science, research and economy' that organized – together with the German Research Center for Artificial Intelligence – an event series (a preparatory workshop and a competition) leading up to a panel discussion at the summit hosted by the MPI for Informatics. Key questions addressed included the need of interoperable national and European research data infrastructures, educating the next generation of (data) scientists, as well as fostering the development of new services to the benefit of the German economy and society at large. Next steps will include summarizing the recommendations made and to suggest possible actions to the German research funding agencies (BMBF and DFG in particular).

International HPC Summer School 2017

The International HPC Summer School on Challenges in Computational Sciences is a series of events started in 2010 and takes place yearly. Supported by PRACE for Europe, XSEDE for the US, RIKEN AICS for Japan, and Compute Canada for Canada, event locations have been alternating between North America and Europe, with the 2016 event hosted in Slovenia. The eighth International HPC summer school is therefore going to take place in the United States, from June 25th to June 30th, 2017. The call for applications is expected to open soon and to last till February 2017. The expense-paid event will bring together 80 excellent students from many parts of the world, to participate for one week in an exciting program coupled with dedicated mentoring and networking. For details see `http://www.ihpcss.org`.