No. 187
May 2014

Bits & Bytes

RECHENZENTRUM GARCHING

Garching Computing Center of the Max Planck Society and the Institute for Plasma Physics*
Boltzmannstraße 2, D-85748 Garching bei München

Status of the Max Planck Supercomputer Hydra

Hermann Lederer

In autumn 2013, the main part of the Max Planck supercomputer Hydra went into operation at the RZG. Compute nodes with Intel IvyBridge processors (Xeon E5-2680v2 with 2 sockets per node, 10 cores per socket with 2.8 GHz clock rate) are connected via Infiniband FDR14 from Mellanox. 3500 IvyBridge based compute nodes were integrated into a single Infiniband system with a common software stack together with the SandyBridge-based part (Xeon E5-2670 with 8 cores per socket and 2.6 GHz) from 2012. In order to support codes that are already able to benefit from GPU technology, 338 nodes were equipped with 676 Nvidia Kepler K20x cards, a system powerful enough to allow for large production runs with simulation codes like, e.g., GROMACS. To support code developments for the Intel MIC architecture, 12 nodes were equipped with Intel Xeon Phi coprocessor cards. The different architecture types of Hydra were listed as three separate entries in the November 2013 Top500 list: The phase-I SandyBridge-based part with 200 TeraFlop/s peak performance as rank 226, the GPU equipped part with 1 PetaFlop/s peak performance as rank 49, and the remaining homogeneous IvyBridge based system with 1.5 PetaFlop/s peak performance as rank 31. The GPU-based system was ranked Top 8 in the Green Top500 list and Top 1 in Germany. In early 2014, the system was further expanded by the Max Planck Institute for Astrophysics and by a consortium of other Max Planck Institutes.

While the standard main memory per node is 64 GB, there are 200 nodes with 128 GB of memory. The I/O subsystem with 5 PB of disk capacity supports three GPFS file systems which can also be accessed from dedicated institute clusters hosted at the RZG for mid-range runs and pre- and post-processing purposes. Also direct HPSS access to the 100 PB mass storage system is realized via the so-called GPFS HPSS Interface. High energy efficiency is achieved through direct ground water cooling of the whole system.

Hydra Environment for GPU and MIC Applications

Markus Rampp

The accelerator/coprocessor partition of Hydra

The new Max Planck supercomputer Hydra (see article in this issue) comprises an 'accelerator' partition with 338 nodes, each equipped with two Nvidia K20x 'Kepler' GPUs (676 GPUs in total), as well as 24 nodes, each equipped with two Intel Xeon Phi 5110p coprocessors based on Intel's new many integrated cores (MIC) architecture (24 Xeon Phis in total). With the accelerators or coprocessors, respectively, a nominal performance of roughly 2000 GigaFlop/s (double precision) is added to each node, on top of the 450 GigaFlop/s provided by the two Xeon E5-2680v2 'IvyBridge' CPUs. The cards are connected to the 'host' CPU via the PCIe(2.0) bus, resulting in a bandwidth of about 6 GB/s for transferring data between the host and card memory. Sustained performance boosts of efficiently ported applications which are due to the additional accelerators are typically in the order of a factor of 2 to 3.

On Hydra, all tools and applications can easily be used with the familiar environment modules system (see the GPU or MIC sections in the output of the command `module available`). Additional libraries and applications can be investigated and installed on request. The RZG applications team supports users with porting, benchmarking and operating of codes for the GPU and MIC architecture and provides general consulting on programming models and tools.

*Tel.: +49(89) 3299-01, e-mail: benutzerberatung@rzg.mpg.de, URL: http://www.rzg.mpg.de/
Editor: H.Lederer & M.Rampp, Rechenzentrum Garching am IPP, Boltzmannstr. 2, 85748 Garching

GPU (Nvidia K20x 'Kepler')

The Nvidia K20x 'Kepler' GPU hardware is characterized by 14 streaming multiprocessors with 192 CUDA cores each, and provides 6 GB of RAM with a bandwidth of about 250 GB/s.

The programming environment for GPU applications on Hydra comprises the CUDA development tools and runtime system, including the CUDA-C compiler `nvcc`, CUDA's numerical libraries like CUBLAS and CUFFT, as well as profiling tools (`nvprof`, `nvvp`). The current CUDA/5.5 environment will soon be upgraded to the recently released version 6. CUDA-FORTRAN and the directive-based OpenACC programming model are supported by the PGI compiler family (currently in version 14.4 with OpenACC/2.0 API support). The MAGMA library provides LAPACK functionality for heterogeneous CPU/GPU nodes (single node with single or multiple GPUs). The Allinea debugger `ddt` allows graphical debugging.

Comprehensive documentation for developing and running CUDA applications can be found at http://docs.nvidia.com/cuda/. Additional training material, including courses on OpenACC and description of experiences from application-porting projects which were presented by Nvidia and a number of Max-Planck scientists in the context of two workshops hosted by the RZG in 2013 and 2014 can be found at the workshop pages (follow hyperlinks to 2013, 2014).

A number of production-ready GPU-enabled applications are provided on Hydra, currently the classical molecular dynamics codes GROMACS, NAMD, LAMMPS. In order to run an application on GPU-accelerated nodes, the line

```
# @ requirements = (Feature=='gpu')
```

has to be added to the LoadLeveler submit script, together with loading the proper run-time module `cuda` at the beginning of the body of the script. The job will then run on the requested number of nodes, each accelerated by 2 Nvidia K20x 'Kepler' GPUs. A complete template script can be found on the Hydra documentation pages.

MIC (Intel Xeon Phi 5110p)

The Intel Xeon Phi 5110p coprocessor hardware is characterized by 60 in-order cores (1 GHz) with 4 hardware threads and 512-bit wide SIMD vector units each, and provides 8 GB of RAM with a STREAM bandwidth of about 170 GB/s.

The programming environment for MIC applications is based on the familiar Intel tool chain with C,C++, and FORTRAN compilers, the Math Kernel Library (MKL) and profiling tools like the Intel trace analyzer and collector. On Hydra, the so-called offload mode (conceptually similar to the GPU programing model) with explicit (by using Intel's *Language Extensions for Offload* set of compiler directives) or implicit ('automatic' offloading by MKL) offloading is supported, as well as the so-called native mode, where the code is compiled only for, and executed on the MIC coprocessor. The latter mode is currently limited to a single node and is considered most useful for prototyping and assessment of algorithms and code.

Comprehensive documentation and training material for developing Xeon Phi applications can be found at http://software.intel.com/en-us/mic-developer.

It is recommended to employ latest version of the Intel compilers (14.0) and the corresponding MKL (11.1) by using the following commands:

```
module switch intel intel/14.0
module switch mkl mkl/11.1
```

In order to run an application on MIC-accelerated nodes, the line

```
# @ requirements = (Feature=='mic')
```

has to be added to the LoadLeveler submit script, together with loading the proper run-time module `xeon-phi-rt` at the beginning of the body of the script. The job will then run on the requested number of nodes, each accelerated by 2 Intel Xeon Phi 5110p coprocessors. A complete template script can be found on the Hydra documentation pages.

GIT Hosting Service

Lorenz Huedepohl, Christof Hanke

As a new service, the RZG provides hosting of git repositories. Git is a distributed revision control system that has become very popular in the last few years, and it is by many considered as the 'de-facto' standard for managing distributed source code development.

Upon request, a group (typically a Max Planck department or working group, represented by a registered RZG user) can get a new, so called 'instance' at the RZG server. Within this instance the representative ('instance administrator') can create and manage one or more git repositories. The instance administrator can invite new users and grant them permissions to view or edit certain or all of

the contained repositories. This, in particular allows them to invite also external collaborators, without the need to apply for an RZG account or for intervention by RZG personnel - instance administrators simply add new users and grant privileges with the help of the web-interface. External users receive an initial e-mail with generated login credentials (valid only for the git service). Alternatively, existing RZG Kerberos accounts can also be configured as the mode of authentication with the git web-frontend. Once authenticated at the web-frontend the new user registers one or more ssh-keys which allows to conveniently interact with the actual repository on the command line (git clone, push, pull, ...).

Technical background information: The RZG employs the open-source git hosting software gitolite for this service which allows to define fine-grained permissions down to the level of git branches, if desired. Instance administrators can also configure scripts to run at a number of pre-defined events. This can be used, for example, to send out notification e-mails for newly published commits to interested users or for a development mailing list, to trigger testsuites, etc.

The new service is immediately available at https://gitta.rzg.mpg.de.

Contact: `git-service@rzg.mpg.de`

Long-Term Archiving

Andreas Schott

As explained in the previous issue of Bits and Bytes (186), HPSS has become the new standard technology for long-term tape archiving at the RZG. To allow all users access to this long-term archiving storage area, several special systems have been set up for the internal access to the `/ghi/r` tree.

`archive.rzg.mpg.de`

This machine provides all users with access to the general migrating file system, which is mounted as `/ghi/r` on the Hydra system and on the visualization cluster. Every user can access this system via ssh and ftp. The home-directory on this machine is here:

```
/ghi/r/<userid-initial>/<userid>
```

Access via ssh is possible without password and also without key, if a proper ticket is available for the usage of the GSSAPI. It is highly recommended not to put too small files onto that filesystem, since there will be a quota on the number of files. And in the case of a necessary tape retrieval each file may require a tape mount, which will take then a very long time to retrieve. Thus, for smaller files the `tar` command to archive directories should be employed. For example, for a directory `dir` to be archived as `dir.tar` the following command can be used:

```
tar cf - dir | ssh archive.rzg.mpg.de \
'cat - > dir.tar'
```

This command will not generate any intermediate file. If you want to archive files from /afs, note that there is no access to /afs available on the machine, so you have to use a command similar to that above.

`afs2ghi.rzg.mpg.de`

This system is available to users of specific projects, where the data is to be moved from /afs to /ghi.

More documentation will appear on the RZG-website in the 'Data Storage' section of the 'Services' soon.