Introduction to HPC-Data Analytics at ZIH

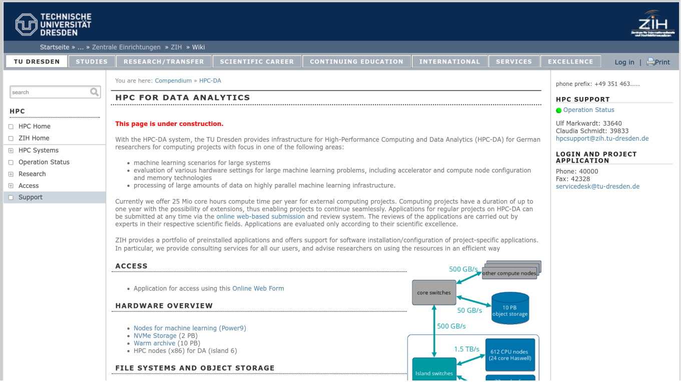
19 September 2019

Dr. Christoph Lehmann
+49 351 - 463 42489
christoph.lehmann@tu-dresden.de

ZIH
Center for Information Services &
High Performance Computing

HPC-DA wiki (maybe) has the answer

Please check our HPC-DA wiki at
`https://doc.zih.tu-dresden.de/hpc-wiki/bin/view/Compendium/HPCDA`



TECHNISCHE UNIVERSITÄT DRESDEN

Startseite » ... » Zentrale Einrichtungen » ZIH » Wiki

TU DRESDEN | STUDIES | RESEARCH/TRANSFER | SCIENTIFIC CAREER | CONTINUING EDUCATION | INTERNATIONAL | SERVICES | EXCELLENCE | Log in | Print

search

HPC

- □ HPC Home
- □ ZIH Home
- ⊞ HPC Systems
- □ Operation Status
- ⊞ Research
- ⊞ Access
- □ Support

You are here: Compendium » HPC-DA

HPC FOR DATA ANALYTICS

This page is under construction.

With the HPC-DA system, the TU Dresden provides infrastructure for High-Performance Computing and Data Analytics (HPC-DA) for German researchers for computing projects with focus in one of the following areas:

- machine learning scenarios for large systems
- evaluation of various hardware settings for large machine learning problems, including accelerator and compute node configuration and memory technologies
- processing of large amounts of data on highly parallel machine learning infrastructure.

Currently we offer 25 Mio core hours compute time per year for external computing projects. Computing projects have a duration of up to one year with the possibility of extensions, thus enabling projects to continue seamlessly. Applications for regular projects on HPC-DA can be submitted at any time via the online web-based submission and review system. The reviews of the applications are carried out by experts in their respective scientific fields. Applications are evaluated only according to their scientific excellence.

ZIH provides a portfolio of preinstalled applications and offers support for software installation/configuration of project-specific applications. In particular, we provide consulting services for all our users, and advise researchers on using the resources in an efficient way

ACCESS

- Application for access using this Online Web Form

HARDWARE OVERVIEW

- Nodes for machine learning (Power9)
- NVMe Storage (2 PB)
- Warm archive (10 PB)
- HPC nodes (x86) for DA (island 6)

FILE SYSTEMS AND OBJECT STORAGE

phone prefix: +49 351 463.....

HPC SUPPORT

● Operation Status

Ulf Markwardt: 33640
Claudia Schmidt: 39833
hpcsupport@zih.tu-dresden.de

LOGIN AND PROJECT APPLICATION

Phone: 40000
Fax: 42328
servicedesk@tu-dresden.de

Hint: These docs are still under construction.

Agenda

1. Overview Data Analytics

2. How to Use Data Analytics Tools on HPC?

3. Recap and Support

What is Data Analytics (DA)?

- There is no standard definition of DA.
- Currently, DA can incorporate areas as Big Data, machine learning, statistics, artificial intelligence etc.
- Two main categories of tasks (not excluding):
  - data intensive (e. g. processing high-frequency sensor data)
  - compute intensive (e. g. Monte Carlo simulations, training of neural networks)
- Combinations of both categories (e. g. train complex neural networks based on large datasets) typically need fast communicating data storage and compute resources – Taurus has it available!

Resources of Interest for DA on Taurus

Heterogenous compute resources (for illustrative purposes)

- Normal compute nodes
  - 270 nodes Intel Sandy Bridge (2 x 8 cores, AVX), 2,4,8 GB/core
  - 1456 nodes Intel Haswell, (2 x 12 cores), 64,128,256 GB/node
  - 32 nodes Intel Broadwell, (2 x 14 cores), 64 GB/node
- Large SMP nodes
  - 2 nodes with 1 TB RAM, Intel Sandy Bridge (4 x 8 cores)
  - 5 nodes with 2 TB RAM, Intel Haswell (4 x 14 cores)
- Accelerator and manycore nodes
  - 44 nodes with 2 x NVidia K20x, Intel Sandy Bridge (2 x 8 cores)
  - 64 nodes with 2 x NVidia K80, Intel Haswell (2 x 12 cores)
  - 22 nodes with 6 x NVidia V100-SXM2, IBM Power9 (2 x 22 cores)

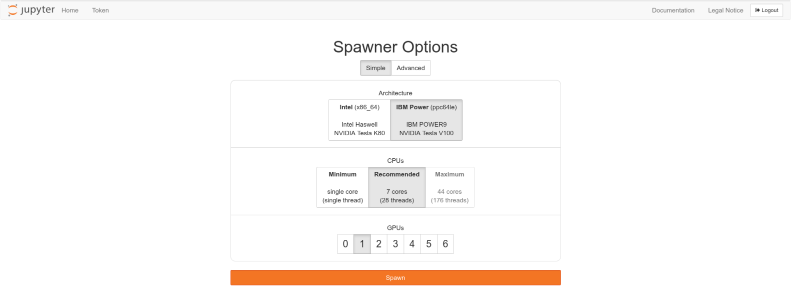
Please note: The GPU-containing nodes (i. e. partitions gpu1, gpu2, ml) are not the general answer for every DA task. Think carefully which resources are really needed for some task.

Data Analytics on Taurus

- Central points for a DA workflow:
  - Typically, the whole process of DA is handcrafted and a trial-and-error chain.
  - Many tools for DA are based on interpreter languages and allow for an interactive processing.
- Basic problem 1: interactive working style is not the standard case on an HPC cluster
- Basic problem 2: DA needs strong hardware resources already for the development of a workflow as otherwise data interaction is not possible at all. E. g. how to handle dozens of GB on your local machine?
- Typical tools for DA workflow: Python/JupyterNotebook and R/RStudio and Spark

Interactive Data Analytics

- Start at: `https://taurus.hrsk.tu-dresden.de/jupyter`



Spawner Options

Architecture

Intel (x86_64)	**IBM Power (ppc64le)**
Intel Haswell NVIDIA Tesla K80	IBM POWER9 NVIDIA Tesla V100

CPUs

Minimum	**Recommended**	Maximum
single core (single thread)	7 cores (28 threads)	44 cores (176 threads)

GPUs

0 [✓ 1] 2 3 4 5 6

- default settings in simple mode:
  - Slurm time limit: `--time=08:00:00`
  - Default choice: Jupyterlab (extended version of JupyterNotebook)
- Docs can be found at: `https://doc.zih.../Compendium/JupyterHub`

Interactive Data Analytics with Python

- the available Python3 kernel already contains
  - TensorFlow
  - Pytorch
- list all available packages within the Python3 kernel: `!conda list` (put into an arbitrary cell within your JupyterNotebook)
- creation of own Python kernels is possible, see the docs at: https://doc.zih.../JupyterHub#...using_own_environment
- for using JupyterLab:
  - first important steps, shortcuts: https://blog.ja-ke.tech/2019/01/20/jupyterlab-shortcuts.html
  - more info at the docs: https://jupyterlab.readthedocs.io/en/stable/

Data Analytics with R

- run RStudio, R console or Rscript directly in interactive mode

```
$ module load modenv/scs5
The following have been reloaded with a version change:
1) modenv/classic => modenv/scs5

$ srun --time=01:00:00 --nodes=1 -c 5 --partition=haswell --pty --x11 bash
srun: job 14292822 queued and waiting for resources
srun: job 14292822 has been allocated resources

taurusi6223 ~ $ module load R/3.4.4-foss-2018a-X11-20180131
Module R/3.4.4-foss-2018a-X11-20180131 and 55 dependencies loaded.
```

Now, we have a bash running on the allocated resources, that allow for parallelization over 5 cores (`-c 5`) on one node (`--node=1`).
Starting RStudio:

```
taurusi6223 ~ $ module load rstudio/1.1.456
Module rstudio/1.1.456 loaded.

taurusi6223 ~ $ rstudio
```

or run an R console calling:

```
taurusi6223 ~ $ R
```

or some R-script directly, calling:

```
taurusi6223 ~ $ Rscript /path/to/script/your_script.R param1 param2
```

Data Analytics with R

- running R on Taurus (cont'd)
  - run R kernel on Jupyterhub (interactive)
  - Rscript command via sbatch (production runs of a workflow)
- How to install packages in R?
  - By default, user-installed packages are stored in the folder `/$HOME/R/` within a subfolder depending on the architecture (on Taurus: x86 vs. PowerPC).
    1. Ask for resources on the respective architecture, e. g.

       ```
       $ srun --time=01:00:00 --nodes=1 -c 2 --partition=haswell --pty bash
       ```

    2. Start an R console

       ```
       taurusi4114 ~ $ module load R

       Module R/3.4.4-intel-2018a-X11-20180131 and 56 dependencies loaded.
       taurusi4114 ~ $ R

       R version 3.4.4 (2018-03-15) -- "Someone to Lean On"
       Copyright (C) 2018 The R Foundation for Statistical Computing
       Platform: x86_64-pc-linux-gnu (64-bit)
       ```

    3. run R-command: `install.packages("package_name")`
  - Alternatively, RStudio can be used as well (currently not available on ml partition).

Big Data Processing with Spark

- Initialize a Spark cluster:

```
# Asking for resources, here: 8 nodes with 60GB memory for each node
$ srun --partition=haswell --time=04:00:00 --nodes=8 -c 24 --mem=60G --pty bash
srun: job 14321467 queued and waiting for resources
srun: job 14321467 has been allocated resources

# Load module, here: Spark
taurusi6181 ~ $ module load Spark
Module Spark/2.4.4-Hadoop-2.7-Java-1.8.0_161-OpenJDK-Python-3.6.6-fosscuda-2018b and 25 deper

# Configure Spark
taurusi6181 ~ $ source framework-configure.sh spark $SPARK_HOME/conf
Warning: Permanently added 'taurusi6181,10.1.139.61' (ECDSA) to the list of known hosts.
Warning: Permanently added 'taurusi6182,10.1.139.62' (ECDSA) to the list of known hosts.

# Start cluster
taurusi6181 ~ $ start-all.sh
starting org.apache.spark.deploy.master.Master, logging to /home/cluster-conf-14321467/spark/
taurusi6181: starting org.apache.spark.deploy.worker.Worker, logging to /home/cluster-conf-14
```

- The initialized cluster can be used interactively, e. g. with the command `spark-submit` see the docs at `https://spark.apache.org...#launching-spark-applications`
- Coming soon in the HPC compendium: docs about using Spark.

How to Check Used Resources?

- Check resources that are allocated to your job, e. g. using `scontrol show job JOB_ID` (get JOB_ID with `squeue -u $USER`).
  Hint: Check out the power of `scontrol` at
  https://slurm.schedmd.com/scontrol.html
- How to check whether the asked resources are really used by some script/program?
  - Detailed view: PerfTools (see the docs at https://doc.zih...Compendium/PerfTools)
  - For development purposes in DA, we are interested typically in a first rough overview:
    1. CPU-utilization: connect to some allocated node of interest via e. g. `ssh taurusi1234`, then run `htop`
    2. GPU-utilization: connect to some allocated node of interest via e. g. `ssh taurusml21`, then run `watch nvidia-smi`

How to Become Faster for Data Analytics?

- Typically, parallelizing code and/or using GPUs is a task on its own and will take some time for development!
- Check carefully what strategies are provided for parallelization by the used tools. Become familiar with threads, processes, cores, CPUs, nodes etc.
- Different tools, different approaches:
  - Python: check `https://wiki.python.org/moin/ParallelProcessing`
  - R: check `https://cran.r-project...HighPerformanceComputing.html` esp. a good-to-read intro `https://nceas.github.ip...parallel-computing-in-r.html`
- consider easy-to-use possibilities of Slurm to run independent tasks in parallel (array jobs): e. g. `#SBATCH --array 0-9`
  see the docs at `https://doc.zih...Compendium/Slurm#Array_Jobs`

Recapitulation and General Hints

- Think carefully which resources are really needed for
  - development,
  - testing,
  - production.
- Please do not use the ml partition if you do not need GPUs!
- Close your interactive session(s) if resources are not needed anymore.
- Consider different architectures on Taurus: x86 vs. PowerPC.
- Build/install packages/libraries/kernels on the right architecture!
- Don't get confused by similar sounding terms:
  - name: ml (machine learning partition)
  - command: `ml` (short for `module load`)
  - PowerPC is the name of an architecture (on the Power9 nodes).
  - Power9 is the name of compute nodes by IBM that are optimized for AI.
- If you are running into deep trouble with unavailable packages and/or complex dependencies the use of containers might be of interest. Check the docs at `https://doc.zih.../Compendium/Container` .

Support and Consulting for Data Analytics

Technical support: hpcsupport@zih.tu-dresden.de

Advanced consulting for applications and complex workflows:
Scalable Data Services and Solutions – Dresden-Leipzig

https://www.scads.de/services or services@scads.de

ScaDS consulting for data analytics:

- data analysis tools (parallel R/Python, RStudio, Jupyter, etc.)
- Big Data Frameworks (Apache Hadoop, Spark, Flink, etc.)
- software for Deep Learning (TensorFlow, Keras, etc.)
- survey of performance optimization of the mentioned software
- For development purposes complete workflows can be built up in a virtual machine (VM).