OPEN CARME
HPC MEETS MACHINE LEARNING

We combine existing open source solutions from HPC and the machine learning/data analytics community. With this in mind we build a framework that provides an easy interactive access to the compute power of HPC centers and gets the maximum out of your AI tool chain.

Carme’s key features

- Seamless integration into existing HPC setups
- User-friendly web-interface providing flexible and os-independent access from anywhere in the world
- Interactive jobs to develop directly on the cluster with your favorite deep learning tools
- Fully separated jobs with custom resources
- Intuitive abstraction of complex cluster topologies
- Distributed multi-node/multi-gpu jobs with direct access to GPI, GPI-Space, MPI, HP-DLF and Horovod
- User maintained and containerized environments using Singularity and Anaconda

Visit us at open-carme.org! For technical details, more information and a live demo come to our booth!
Carme Dashboard with:
status bar, system messages, cluster load, job status messages, job configuration with “Start button”, list of running jobs (with different entry points, job information and the “Stop button”).

Open Carme’s features (in more detail)

- Seamless integration with available HPC tools
  - Job scheduling via SLURM
  - Native LDAP support for user authentication
  - Integrate existing distributed file systems like BeeGFS

- Access via web-interface
  - OS independent (only web browser needed)
  - Full user information (running jobs, cluster usage, news/messages)
  - Start/Stop jobs within the web-interface

- Interactive jobs
  - Flexible access to GPUs
  - Access via web driven GUIs like Theia-IDE or JupyterLab
  - Job specific monitoring information in the web-interface (GPU/CPU utilization, memory usage, access to TensorBoard)

- Distributed multi-node and/or multi-gpu jobs
  - Easy and intuitive job scheduling
  - Directly use GPI, GPI-Space, MPI, HP-DLF and Horovod within the jobs

- Full control about accounting and resource management
  - Job scheduling according to user specific roles
  - Compute resources are user exclusive

- User maintained, containerized environments
  - Singularity containers
    (runs as normal user, GPU, Ethernet and Infiband support)
  - Anaconda Environments
    (easy updates, project/user specific environments)
  - Built-in matching between GPU driver and ML/DL tools