Technische Universität Dresden

Alexander Grund
11th EasyBuild User Meeting, @ Guimarães, Portugal

EasyBuild @ ZIH – TU Dresden
Take (it) EESSI

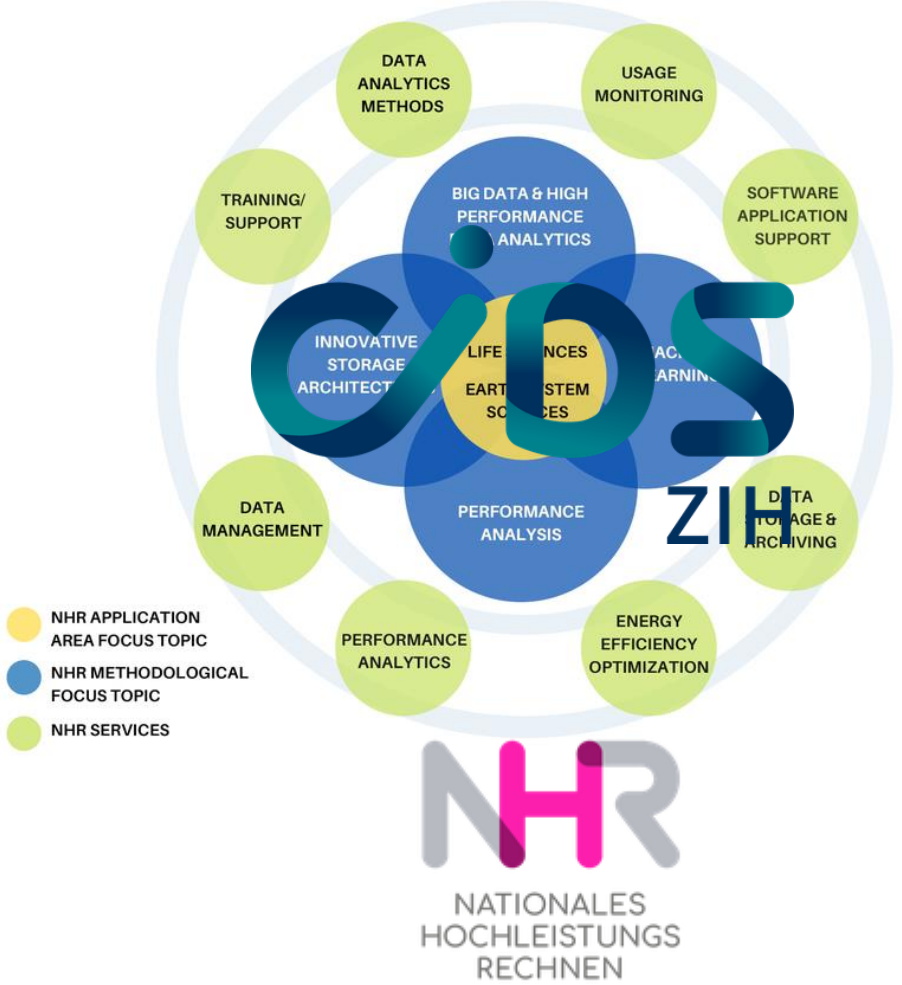
22nd April 2026

TU (University of Technology) Dresden

Center for Information Services and High Performance Computing (ZIH)

- IT services for TUD
- Research & Development
- Connected to ScaDS.AI
  - Big Data, Data Science, Artificial Intelligence
- 1 of 9 NHR centers, Tier 2
  - Researches of German Universities
  - Life sciences
  - Earth system sciences
  - Trainings

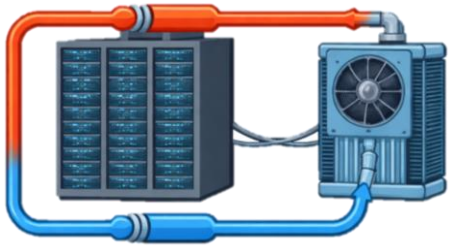
ScaDS.AI
DRESDEN LEIPZIG



NATIONALES HOCHLEISTUNGS RECHNEN

5 (+2) HPC Clusters

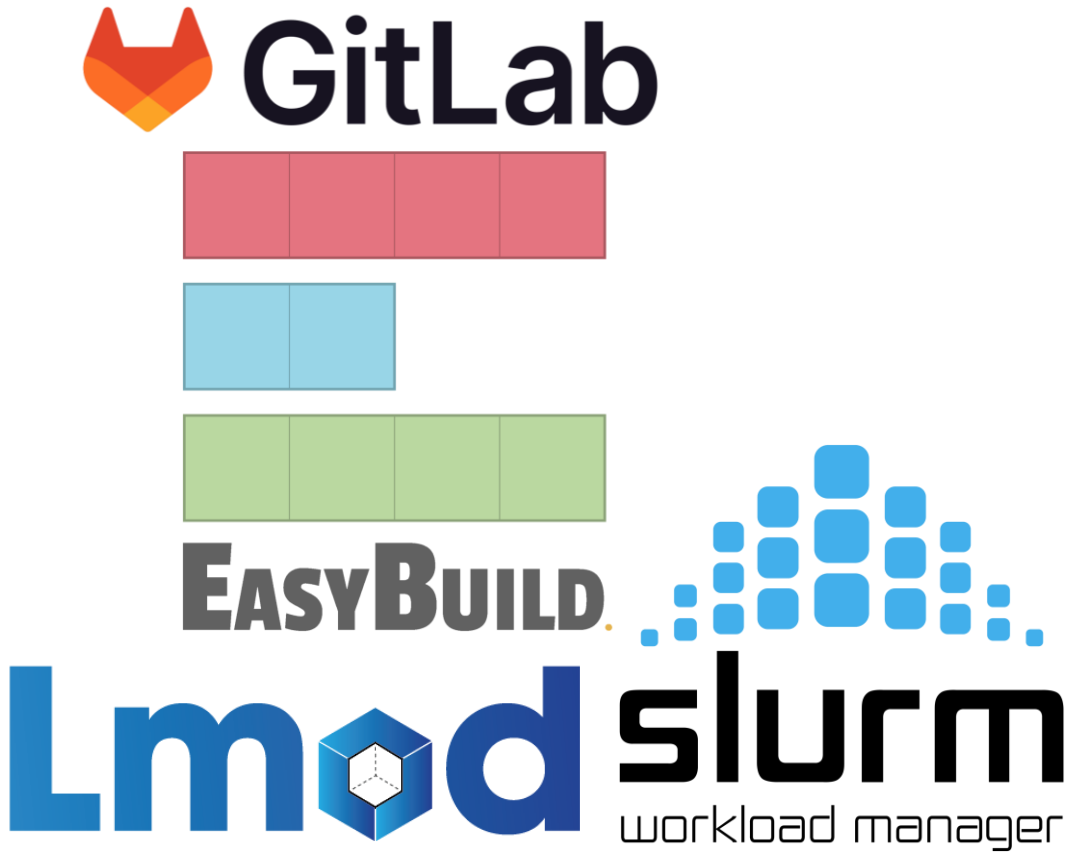
Direct Hot Water Cooling



- Barnard TOP500 #419 #152
  - 720 nodes →74880 cores
    Intel Sapphire Rapids
    512 GB RAM
- Romeo
  - 192 nodes →24064 cores
    AMD EPYC 7702
    512 GB RAM
- Julia
  - **1 Node,** 896 cores
    Intel Cascade Lake 8276
    48 TB RAM
    370 TB NVMe memory
- AlphaCentauri TOP500 #493 #108
  - 37 nodes → 1776 cores
    AMD EPYC 7352
    296 NVIDIA A100 (tot.)
    1 TB RAM
    3.5 TB NVMe
- Capella TOP500 #73 #6
  - 156 nodes → 9984 cores
    AMD EPYC 9334
    624 NVIDIA H100 (tot.)
    768 GB RAM
    814 GB NVMe
- CARA @ DLR
  - 770 nodes → 98560 cores
    AMD EPYC 9555
  - 10 nodes → 1280 cores
    AMD EPYC 9555
    40 NVIDIA RTX4000 (tot.)
  - 20 nodes → 640 cores
    Intel Emerald Rapids
    80 NVIDIA H100 (tot.)
- Future: Deneb
  - 46 nodes → 1656 cores
    2 NVIDIA GB200 → 184 GPUs (tot.)
    372 GB RAM

Software Stack

- Slurm
  - Dedicated login nodes per cluster
  - „Interactive“ partition on login nodes
    → MIG mode for GPU slices
- EasyBuild
  - Hierarchical module structure
  - “release” meta module (23.04, 23.10, 24.04, …)
    - Yearly since 2025
  - Optimized per cluster → *$LMOD_SYSTEM_NAME*
- GitLab
  - Automated builds and reports



Going EESSI

First steps

EESSI

EUROPEAN ENVIRONMENT FOR SCIENTIFIC SOFTWARE INSTALLATIONS

TUD

EESSI Choice

Current issues

1. Preparing new software stack releases very time consuming
   - Gather „standard“ list of mandatory modules
   - Installing many modules present in previous releases
   - ➔ Duplicated effort
2. Users request additional software
   - ➔ Never done
3. Keeping reasonable amount of “old” software

EESSI Choice

EESSI as base

1. Additional modules built on top
2. Customization through targeted rebuilds, possibly *module-only*
    1. Hierarchical module tree
    2. PYTHONPATH → EBPYTHONPREFIXES
    3. LD_LIBRARY path to avoid rpath-wrappers for users
3. Periodic rebuilds to ingest new modules

EESSI Approach

Rebuild of **ALL** modules with *--module-only*

1. Load EESSI module → CPU & GPU paths
2. Install EasyBuild module to new path
3. Reset *$MODULEPATH*
   Configure EB
4. Collect installed EasyConfigs
   → *easybuild/reprod* folder
5. Install collected EasyConfigs
   → Skips already installed modules

(Not so) EESSI Approach

Rebuild of **ALL** modules with *--module-only*

1. Load EESSI module → CPU & GPU paths
2. Install EasyBuild module to new path
3. Reset *$MODULEPATH*
   Configure EB
4. Collect installed EasyConfigs
   → *easybuild/reprod* folder
5. Install collected EasyConfigs
   → Skips already installed modules

1. Which EasyBuild version?

   Different EasyBuild versions used in EESSI
   - Framework & EasyBlocks changed
   - Different config options

   ➔ Latest from EESSI

2. Which EasyConfigs?

   Changed/removed between versions

   ➔ Use only names
   ➔ Copy missing ones

3. Failing sanity checks

   Missing files, e.g. shared ⇔ static builds

   ➔ Skip

(Not so) EESSI Approach

PYTHONPATH → EBPYTHONPREFIXES

1. Install Python dependencies
2. Rebuild Python
3. Install other modules

Open issues:

1. Filtered dependencies

➔Extract settings from EESSI-extend

2. RPATH to compat layer / *--sysroot*

   --filter-env LD_LIBRARY_PATH

3. Integration with system libraries
   - OpenMPI: Mellanox KNEM, Slurm, (PMIx)
   - Intel MPI: PMI 2
4. Combining hooks

Ideas wanted