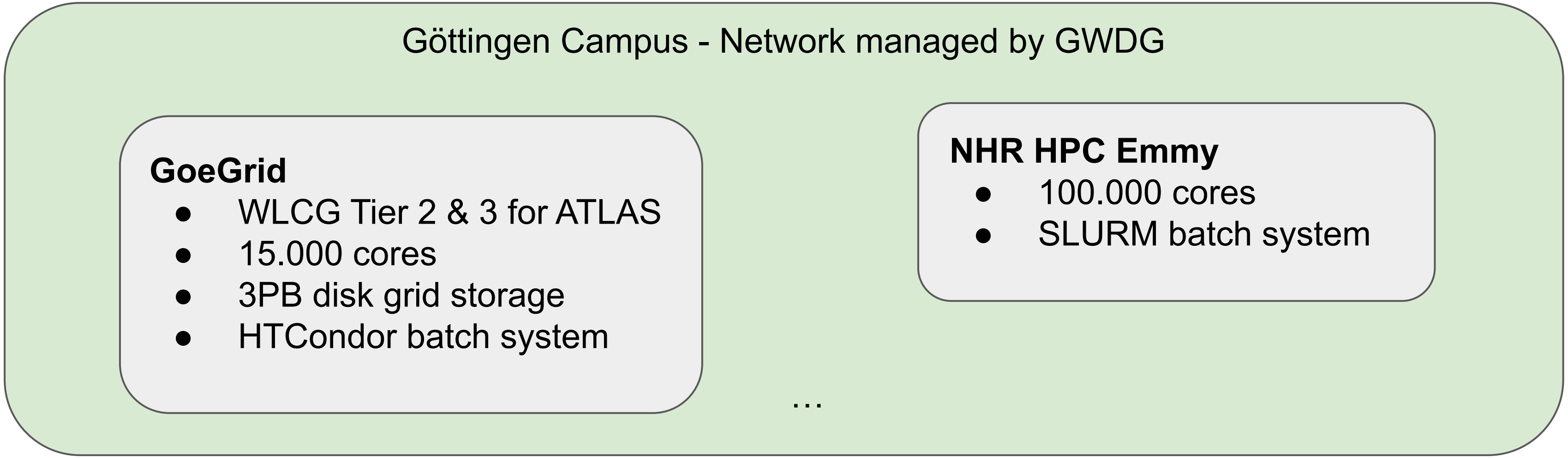
GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

GoeGrid & HPC Integration into GoeGrid (WLCG Tier2)

PUNCH4NFDI TA2 Workshop
30.09.22

Daniel Schindler, Sebastian Wozniewski

Overview



Göttingen Campus - Network managed by GWDG

GoeGrid

- WLCG Tier 2 & 3 for ATLAS
- 15.000 cores
- 3PB disk grid storage
- HTCondor batch system

NHR HPC Emmy

- 100.000 cores
- SLURM batch system

…

Can we make Emmy accessible for ATLAS jobs?

- Additional resources / opportunistic usage for free
- Long term: Only one (merged) cluster for the sake of shared and efficient resource usage
- In times of limited network between sites (20G link of GöNet): high connectivity to local Tier2 storage relevant; also avoid firewalls in between

Typical job requirements

- single-core jobs / single-node multi-core jobs (mostly 8-core, but also e.g. 64-core)
- ~ 2 GB memory per core
- ~ 4 GB local scratch space per core
- ~ 1 Mbit/s per core network usage (remote storage access)
- 12 hours walltime with large variance (due to pilot model)

In theory…

- Use COBALD/TARDIS:
  - Virtually extend GoeGrid batch system / separate virtual HTC batch system with containers turning HPC nodes into virtual nodes with own job scheduling
  - Flexible management of booked HPC resources (whole-node scheduling policy no problem)
  - Automated resource selection and booking
  - Containers allow for custom environment managed by people from the ATLAS/WLCG world
    - Multiple layers of containers already used by ATLAS to provide environment for pilot jobs and job payloads in order to manage it centrally. Grid sites only provide very basic infrastructure.

- Establish high-bandwidth connection to local grid storage

- Allow for outbound connections to outside grid services e.g. cvmfs

- Use present GoeGrid caches / additional caching on shared HPC file system

=> *HPC usage would be transparent to ATLAS*

Hurdles encountered so far

- Even if managed by same company, networks may be rather disjoint => fortunately new structure in progress (not just due to us)

- Restrictions imposed by HPC:

  - Network: Outbound connections from nodes not possible by default - using proxies problematic due to high traffic - allow connections to known IPs as a compromise?
    - Works for local grid storage, squid, batch system.
    - Feasible for other grid services? Automation other than IP list possible?

  - Software permissions
    - No FUSE - Prevents installing cvmfs as a user without a container layer
    - No unprivileged user namespaces - Prevents multiple layers of containers as needed for using CT with ATLAS jobs
      - User-specific temporary permission based on setuid-script?
      - Leave out network namespaces which is not needed but main reason for security concerns?
  - Walltime preferences
    - Drone lifetime should cover multiple jobs in sequence for efficient usage
    - HPC limits long-term jobs (most resources allow only 12h); ~2h for opportunistic usage

Other questions in context of PUNCH

- Does such a virtualized setup also work for multi-node jobs heavily exploiting MPI (astroparticle physics)?
- Can we enforce access restrictions to certain resources?
- …