
Leibniz Supercomputing Centre

of the Bavarian Academy of Sciences and Humanities

# Leibniz Supercomputing Centre

How We Learned to Stop Worrying and Love* Spack | 11/13/2018 | J. Albert-von der Gönna

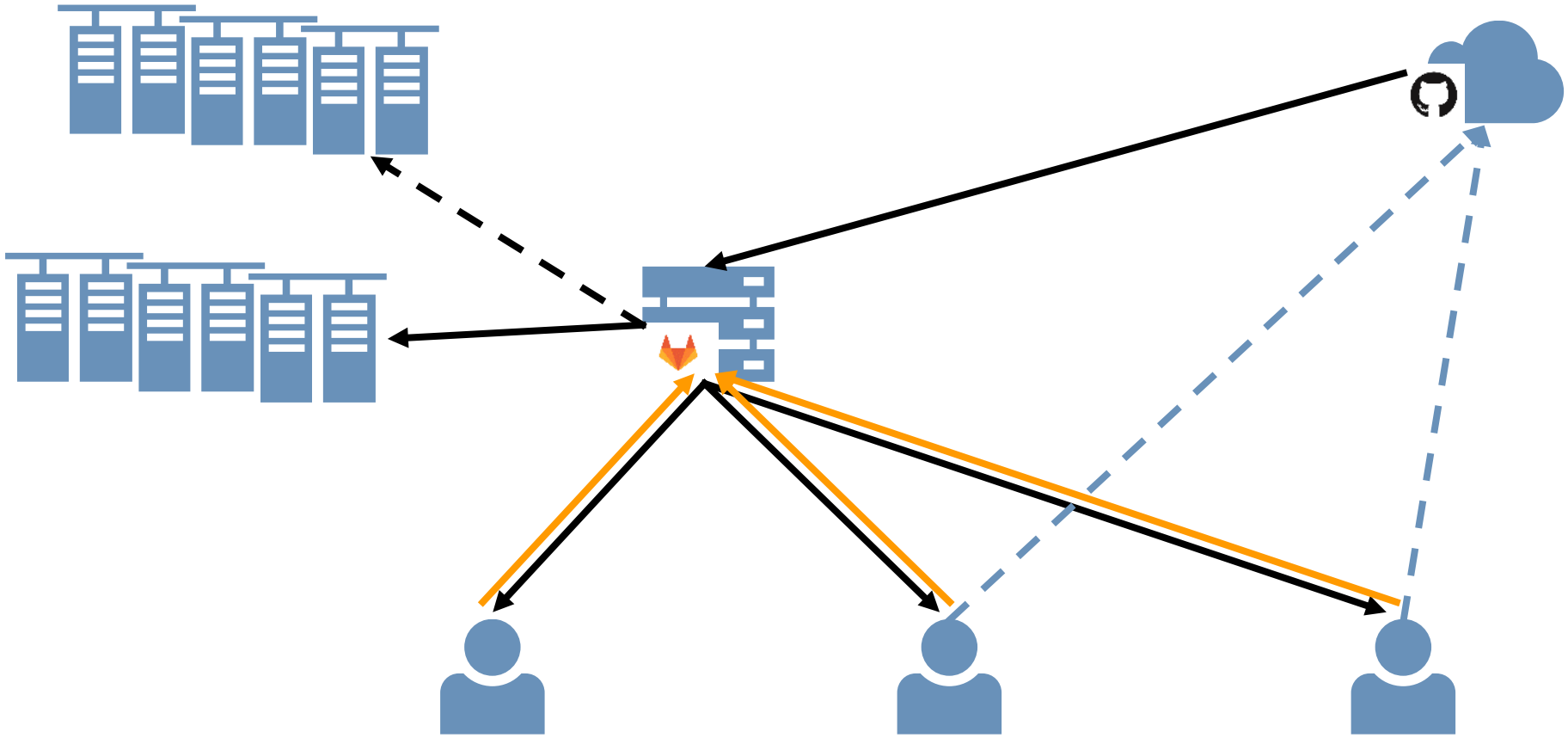
# The LRZ Road to Spack

- About a year ago…
  - joined LRZ, tasked to manage R installations on various systems (amongst other things)
  - came across Spack & began asking around whether it was known/used: a colleague went to a tutorial, another one had started to look into it
- In spring, the three of us got to work…

  … we explored Spack's general functionality, API, and multiuser workflow capabilities

  … and set up infrastructure to work towards
    - consistent (yet flexible) software installations (i.e. Spack)
    - maintained in a collaborative way (i.e. Git)

# State of the LRZ Spack



# State of the LRZ Spack

- 15 internal contributors
- 300+ commits
- 150+ “recipes” (i.e. specs) form the basis for our stack

```
SPEC_lib=library@version%compiler+variants
^dependency@version%compiler+variants^…

SPEC_app=application@version%compiler+variants
^$SPEC_lib
```

- Bash script to automate build process (alternative: make-based approach to allow for parallelization)
- 200+ modules

In production on (most of) the 500+ node Linux Cluster

- CoolMUC-2 (Haswell; AVX2)
- CoolMUC-3 (KNL; MIC-AVX512)

Ready to be put in production on the 6480 node SuperMUC-NG (Skylake; AVX512)

# Conclusion and Outlook

- **Challenges/Lessons Learned**
  - Heterogeneous, perpetually changing cluster structure
  - Make sure everybody is in the same boat (weekly progress meetings)
  - Document and teach the tools

- **Discussion**
  - Quality of built-in packages varies. Is there a strategy to establish certain standards/control?
  - What's the status of dependency re-use improvements (i.e. concretization)?

- **Wishlist**
  - Spack chains (and the Stacks vision)
  - Reverse-dependency resolution
  - Versioning, regular releases, changelogs

*To love is
to find pleasure
in the happiness of others.
– G. W. Leibniz