EESSI hackathon - show and tell

Jan 2022

https://github.com/EESSI/hackathons/tree/main/2022-01

Agenda

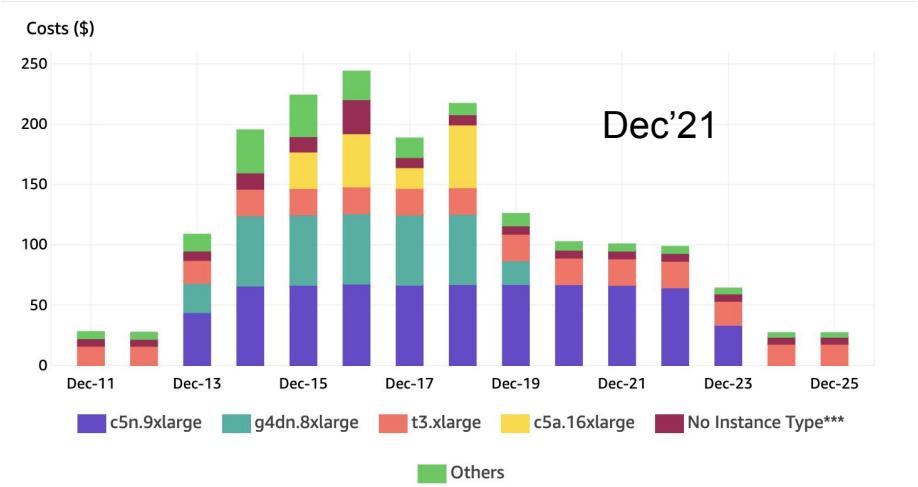
- General feedback
- Spent credits in AWS/Azure
- Task [02]: Installing software on top of EESSI
- Task [03]: Workflow to propose additions to EESSI software stack
- Task [05]: GPU support
- Task [06]: EESSI test suite
- Task [07]: Monitoring
- Task [16]: Export a version of the EESSI stack to a tarball and/or container image

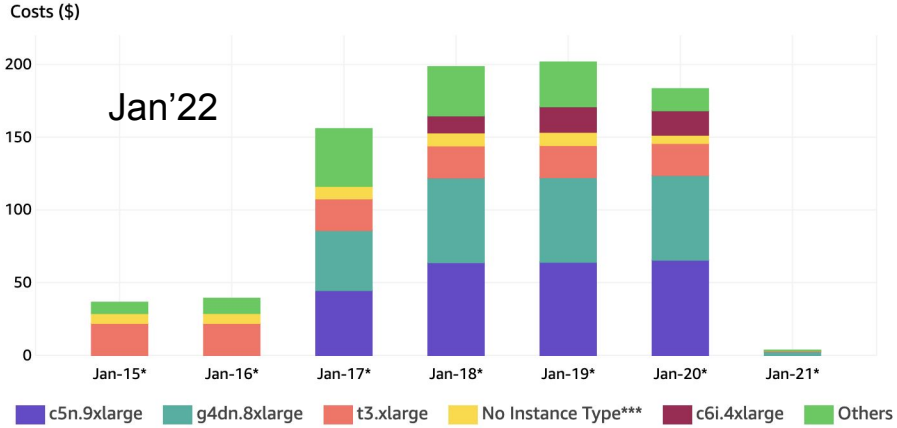
General feedback

- What went well? What didn’t?
- What could be changed/improved for the next hackathon(s)?
- Organisation: HackMD notes, Zoom calls, Slack, GitHub hackathons repo, …
- Infrastructure: virtual clusters using resources in FENIX, AWS, Azure
- Allocating time for hackathon

Spent credits in AWS for hackathons

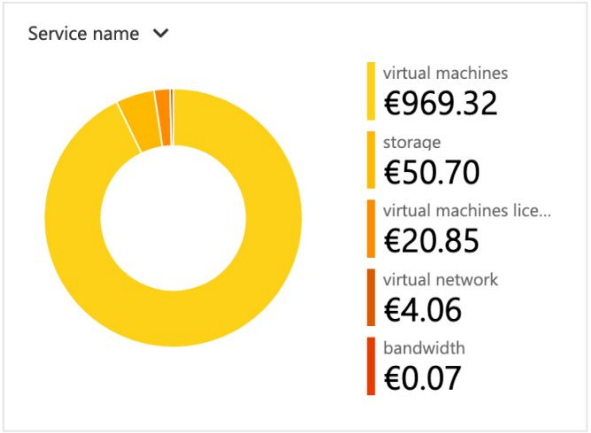
- (partial, credits consumed on/after Fri 21 Jan 2022 are incomplete)
- ~$750 worth of sponsored AWS credits spent on Jan'22 hackathon (estimated)
  - Magic Castle: ~$60/day on GPU node (g4dn), ~$65/day on EFA nodes (c5n)
  - CitC cluster: < $50/day in total (only small, short-lived instances)

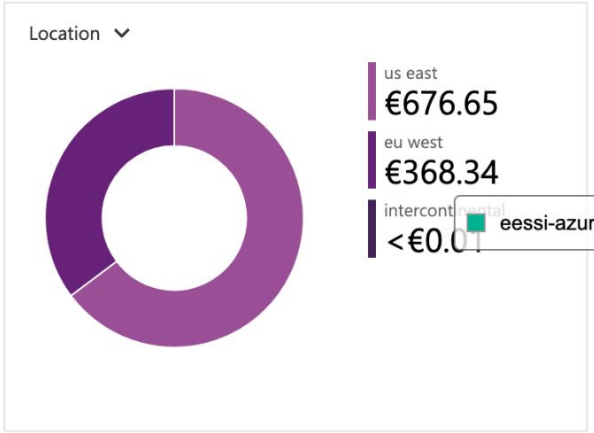


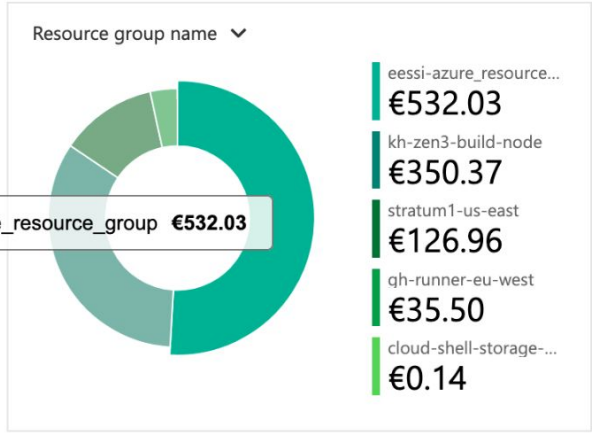


Spent credits in Azure for Jan’22 hackathon

- (partial, credits consumed on/after Fri 21 Jan 2022 are incomplete)
- ~€550 worth of sponsored Azure credits spent on Jan’22 hackathon (estimated)
  - Magic Castle cluster with HC44rs VMs (incl. InfiniBand network)
- ~€350 worth of sponsored Azure credits spent build node for AMD Zen3 (Milan)







Task [02]:
Installing software on top of EESSI

https://hackmd.io/sLBLV7RDQdmyYfh1rYHGSQ

[02] Installing software on top of EESSI

- TODO list for clean support for including RPATH wrappers in compiler installation: https://github.com/easybuilders/easybuild-framework/issues/3918
- Working on more generic implementation of `prepare_rpath_wrappers`

  in `toolchain.py` in EasyBuild framework
  - Add parameters for `new_wrapper_dir`, `single_subdir` and `disable_wrapper_log`
  - Defaults should replicate previous behavior
- Add `create_rpath_wrappers` function to `toolchain/utilities.py` in EasyBuild framework
  - Setup toolchain with given name & version
  - Then call `prepare_rpath_wrappers` and put wrappers in target dir

(Martin)

- **Just providing wrappers for `ld` seems to be enough!**

[02] Installing software on top of EESSI

- Manually building software on top of EESSI
  - Showcase package is LAMMPS 21Sep2021 (Molecular Dynamics):
    - CMake based with library dependencies from EESSI stack:

      OpenMPI, libfftw, ffmpeg, Eigen (CUDA support optional)
  - Main caveat is RPATH linking to avoid using host libraries (only compat layer)
    - Requires development module from EESSI (foss/2020a)
    - Correctly set up build configuration files in LAMMPS

(Frank)

[02] Installing software on top of EESSI

- Problem: Execution fails with "`error while loading shared libraries`"
  - The affected library was `libfftw3.so`
  - This is due to `CMakeLists.txt` not correctly specifying RPATH consistently.
- Can be fixed by including RPATH wrapper scripts for compilers in EESSI
  - Built by Kenneth in `/mnt/shared` for `x86_64` on CitC cluster

    using enhanced GCC easyblock by Martin (see easyblocks PR #2638)
- After building, LAMMPS now runs! Scripts added to hackathon repo
- 16 core, 4 node job, 1000 water molecule simulation

```
LAMMPS (29 Sep 2021 - Update 2)
OMP_NUM_THREADS environment is not set. Defaulting to 1 thread. (src/comm.cpp:98)
  using 1 OpenMP thread(s) per MPI task
  2 by 2 by 4 MPI processor grid
```

(Frank)

Task [03]:
Workflow to propose additions to EESSI software stack

https://hackmd.io/6V91CHRWRtuutANPaZRVPw

[03] Workflow to add software to EESSI

- All code for the Github App itself in in a separate repo:
  - https://github.com/EESSI/eessi-bot-software-layer
- Very little progress during Jan'22 hackathon on this :(
- Mainly script to autonomously build/install software in EESSI software layer (see PR #163 + PR #164)

(Kenneth)

[03] Workflow to add software to EESSI

- GitHub App still needs quite a lot of work:
  - Implement CI tests in https://github.com/EESSI/eessi-bot-software-layer
  - Use the EESSI build container and settings (paths, Easybuild config)
  - Monitor build/test jobs + handle failures
  - Reply results back to PR (comment on success vs fail, logs via gist, …)
  - Pick up logs and tarballs
  - More architectures
  - Event dependencies
  - Handle many more events

(Kenneth)

Task [05]:
GPU support

https://hackmd.io/47FAwaeWRi66tdiqjy2Zvg

[05] GPU support

- Hackathon impressions
  - Magic Castle environment worked pretty well
    - Would need to also test on other OSes/setups
  - Tweaking a working setup proved easier
- Achievements
  - GPU support is working, initial script for new pilot version `2021.12`: https://github.com/EESSI/hackathons/tree/05_gpu/2021-12/05_gpu
  - GROMACS with GPU support has been installed and seems to work
- Issues
  - Final script will need a lot of tests (checks for drivers, space, location,...)
    - Started to implement checks here
  - Need to be able to unpack `.deb` and `.rpm` with tools only from compat layer
    - Ok for deb (`ar` + `tar`), need `rpm.eclass` for RPMs

(Alan)

[05] GPU support

- **DEMO!**
- What still has to be done
  - Create EasyBuild hook to add Lmod tag to CUDA and CUDA-enabled modules
  - Create Lmod hook to hide tagged modules unless some condition is met
    - Existence of `$EBROOTCUDA/bin`?
    - Allow forcing show via environment variable (would require global CUDA installation)
    - This will need to independently handle different EESSI versions
  - Need a symlink in CVMFS
    - Software installation path of CUDA with `versions` -> `host_injections`
    - Module should fail to load if `$EBROOTCUDA/bin` does not exist

(Alan)

Task [06]:
EESSI test suite

https://hackmd.io/wx2hjHiWQnmkERSVR2-a2A

Almost no activity for this task during Jan’22 hackathon in the end :(

Task [07]: Monitoring

https://hackmd.io/YWDG2GO5R3Sm3wS1SpvYrg

Very little progress for this task during Jan’22 hackathon in the end :(

Task [16]:
Export a version of the EESSI stack to a tarball and/or container image

https://hackmd.io/2YpzQGgUSDyTvW3ILulzwA

[16] Exporting EESSI to a tarball/container

- Node local I/O performance and capacity sufficient
- Singularity `--fakeroot` still requires root configuration
  (see https://sylabs.io/guides/3.5/user-guide/fakeroot.html)
- Learned that Singularity is not willing to copy broken symlinks into a container
- Using EESSI init scripts is problematic in noninteractive container startup
- Current script implementation just prints out commands for user
  to run within container to init the environment
- We now have ability to pick up generic modules instead of optimized ones
- Overall pretty happy with “minimum viable product” resulting from these two hackathons
- Script available in hackathon repo

(Jure)

[XX] Adding Azure support to CitC

- PR's for Terraform (PR #68) and Ansible (PR #118) submitted
- Infiniband RDMA working on
  - HC44rs (Skylake) with EDR
  - HB120rs_v2 (Zen2) with HDR
- Infiniband support implemented using vm's and availability zone per shape
  - Advantage: relative easy to implement and working now
  - Downside: currently serial autoscaling, so getting 16 nodes takes a while
- Seeing proper scaling using WRF3 using above node shapes

Sidenote: testing EESSI integration with Az-hop

(Hugo)