visibility of this document: everybody who has this link can edit

short-link to this document: <http://bit.ly/hpc-reimagined>

Reimagined HPC

In this workshop, we aim for a vision of HPC which is more accessible for everyone. This document should be our final output which outlines a path forward: analyzes different topics and proposes solutions.

# Template for group work (but it’s OK to freestyle instead):

How will we know that it’s done and successful?

* *What is the end state. How does it help and who does it help?*

How to do it?

* *What are the steps?*

Which obstacles?

* *Which problems do you anticipate?*

Who needs to be involved?

* *(key players: both tech and admin)*

# Example: filesystems

### Papercuts/vision

### Solution A

How will we know that it’s done and successful?

* Filesystems are organized in a logical manner consistent across all systems
* Simplified filesystem documentation: storage can be explained independent of what system it is on
* Preferably, similar allocation policies

How to do it?

* Standard mount points: /mnt/$filesystem\_name on all systems
* Research filesystems available on

Which obstacles?

* If research filesystems are available on daily systems, there is greater risk of security breaches. kerberos can somewhat minimize this risk but brings
* If storage is not in the same domain (uids, gids match, shared auth), it can not be transparently cross-mounted.
* If users do not know the source of a filesystem, they may not know who to ask for administration tasks.

Who needs to be involved?

* Storage admins of *each* type of storage
  + Maintainers of all shell servers and research infrastructures
* Network and security people: open the relevant ports and
* Unknown: whoever creates storage policies and pressures others to adopt them.

### Solution B

...

# Accounts/access

### Papercuts/vision

* Better authentication, authorization, delegation
* Easy on-demand access to HPC for new users
* Being able to login everywhere with my ORCID or my university identity
* “Free tier” in HPC (easy access sandbox)
* HPC in your pocket! (HPC on a smartphone?)
* HPC required a different account to manage
* Lack of “community accounts” on HPC sites
* Better terminal MOTDs
* JupyterHub as an interface
* Easy and fast access not always the case
* Cloud ←→ HPC : some workloads (data science) work fine on cloud. Or only portions of workloads need HPC

### Solution A

* Better authentication, authorization, delegation
* Ideally, any user within the EU/EAA should be able to identify themselves using a legally recognised electronic identity (EIDAS)

The problem is that this (EIDAS) is not in place today.

* Easy on-demand access to HPC for new users
  + Define “Easy”
* Being able to login everywhere with my ORCID or my university identity
  + This is connected to the first issue. Also, in many instances Edugain solves parts of this issue.
* “Free tier” in HPC (easy access sandbox)
  + Would require a solid E-identification schema as described above.
* HPC in your pocket! (HPC on a smartphone?)
  + Given that more or less standardised interfaces are used this is already solved (at least for a limited set of applications) Also, if web based submission forms are used the platforms where these are displayed is irrelevant.
* HPC required a different account to manage
  + Local decisions, is possible to solve but depends on what they decide to trust
* Lack of “community accounts” on HPC sites
  + And for good reason.
  + Organize users into groups, group-based resources by default. If default storage space is group shared, ...
* Better terminal MOTDs
  + This is a problem best handled by the local administration. The administrators of the system must be responsible for what is in the message and that it is up to date.
* JupyterHub as an interface
  + Implemented at some HPC-centers already (to some extent at least)
* Easy and fast access not always the case
  + This is inherent in the system, as we need to know who is using our systems. There is also limited amounts of support staff.
* Cloud ←→ HPC : some workloads (data science) work fine on cloud. Or only portions of workloads need HPC

### Solution B

...

# Storage/filesystems

### Papercuts/vision

* Filesystems different everywhere
* Global-filesystem: accessing data without copying huge amounts of data
* Better understanding of filesystems: filesystems hard to understand
* Data owned by individual users - sharing not easy, leads to reinventing and starting from scratch
* Web-based access to filesystems
* Better illustrations in docs, especially filesystems
* Remote mounts of filesystems
* Standard FS paths and names - at least within organization

### Solution A

How will we know that it’s done and successful?

* *What is the end state. How does it help and who does it help?*

How to do it?

* *What are the steps?*

Which obstacles?

* *Which problems do you anticipate?*

Who needs to be involved?

* *(key players: both tech and admin)*

### Solution B

“Global filesystem” or seamless access of your data everywhere

How will we know that it’s done and successful? The Vision is:

* *When everyone has their own global cloud personal storage, accessible everywhere (almost) at the speed of a local SSD disc.*
* *When everyone has access to global public data (e.g. large scientific databases) everywhere (almost) at the speed of a local SSD disc.*

Which obstacles?

* *Large files / large amounts of data (small data is “easily” moved around).*
* *Orchestration/locality. Locality of data is important. Making sure data is where it needs to be when it needs to be.*
* *Security. Personal data should only be accessible by the owner.*
* *Hardware.*
* *Need many storage servers all around the world. Mirroring of data?*
* *Network. Bandwidth and latency.*
* *Designing a new filesystem*

*- Scalability. Multiple meta-data servers.*

*- Distributed over multiple sites?*

*- Migrating data around with the user?*

How to do it?

Who needs to be involved?

* *Everyone! Politicians, developers, technical staff, etc.*

### Suggested standard filesystem paths

(what is most commonly used? what existing standard?)

# Documentation

### Papercuts/vision

* Documentation is a headache. It is a limit to what you can document
* Users will \*NOT\* read documentation
* No need of documentation because everything is so easy (as easy as reading email)
* Look-and-feel differs between resources and nomenclature differ sometimes. This is confusing for users
* Better illustrations in docs , especially filesystems
* Searchability - easy to find what you are looking for in documentation.

### Solution A

How will we know that it’s done and successful?

* *What is the end state. How does it help and who does it help?*
* *Documentation is not one way: users should be able to contribute and “correct” it*
  + *To verify: At least XXX users contributed to the documentation*
* *Add a glossary explaining all the jargon*
* “Google like” search engine over all compute centra
  + To verify: make a survey to define a list of “typical” questions and demonstrate that you can find information easily

How to do it?

* *What are the steps?*
  + *Create a github organization NordicHPC (I think you have it…) - yes:* [*https://github.com/nordichpc/*](https://github.com/nordichpc/)
  + *Move/Add documentation for each Nordic country with a list of available resources (map of resources where we can click to get more information)*
  + *Make a list of experts in the Nordics (RT queue is nice but too anonymous and users feel they are just numbers)*
  + *Connect users with each other using for instance slack or zulip (someone running GROMACS in Sweden may be able to help and give advice to another user in Finland or Norway; better than “local” HPC support)*

Which obstacles?

* *Which problems do you anticipate?*

*Get another documentation that is not read by anyone.*

*Difficult to coordinate within the various HPC centers*

Who needs to be involved?

* *(key players: both tech and admin)*
* *HPC centres, cloud providers, storage providers and community representatives (and users)*

SNIC has some standard variables which are expected to be available everywhere

### Solution B

Which obstacles?

* Documentation can get to unwieldy
* Users can be too lazy to read documentation

How to do it?

* *Force users to read documentation before they get access to software / computing power ← And how to check it? Do they need to answer to a questionnaire and get acces to HPC only if they are right?!!! Yes :)*
* *Pay-per-support ticket ← That is called google or AWS services.*
* *Documentation should be easily searchable, seamlessly tying related subjects together (e.g. via links, both going forwards and back again)+1*

# Outreach/training

### Papercuts/vision

* Data stewardship is needed
* Be strategic, not tactical. if goal is to enable science, begin there. How to make HPC easier to use
* Focus on long tail of science more (real long tail, not long tail of HPC!)
* Poor preliminary study of user's needs - lot of work with little feedback/results
* Visibility / awareness of resources available

### Solution A

* Train the trainer program in the Nordics to make sure everyone can get information on where to access HPC and how to make data FAIR
* Identify key communities and key “users” to disseminate information and training
* Stop talking about HPC but talk about Enabling Science with both computing and storage
* Have a more co-design approach i.e. users need to be aware of future trends (using simple words) so they can be prepared.
* HPC Carpentry or “ComputingCarpentry” so users better understand what to use and for which purpose
* NordicHPC website with a map containing list of HPC centres and resources, data steward, information on how to make your data FAIR i.e. one single access for getting information (with link to training, etc.).

How will we know that it’s done and successful?

* + XXX% of new users from new communities
* *What is the end state. How does it help and who does it help?*

How to do it?

* *What are the steps?*

Which obstacles?

* *Which problems do you anticipate?*

Who needs to be involved?

* *(key players: both tech and admin)*

### Solution B

...

# Queuing/computing

### Papercuts/vision

* Lightweight computing more common
  + Needs more clarification. We observe the opposite (because data size grows, and laptops, phones are more capable for computing)
* Ability to extend my job once automatically for free
  + Not always feasible, e.g. privileged command to run on queueing system master. Responsive support organization usually works better.
  + Superdome setup: User can define the max wallclock time. Can set this really high, but then becomes difficult to schedule the job.
* The task cycle is split into subtasks which are out of our control
  + User needs to know how certain libraries work. (Training?)
  + ‘Point and click’ portals can limit users’ activities (but limit the scope of use cases too)
* Why is my job not prioritized. Can you change the queuing system for me?
  + NO
* Testing/debugging runs for only 1-2 min, why do we have to use the queueing system?
  + 1-2 minute jobs can just as easily run on user workstation/laptop - no need/reason to involve HPC in that.
  + If special hw is required (e.g. GPU), an interactive queueing session is recommended.
  + A “solution” can be having a test-queue with low max wall time (5 min to 1 hour). This will usually not be full, or at least wait times should not be to long.
* As user cannot extend own jobs (duplicate of “Ability to extend…” above)
* Efficient resource usage (compute, storage) is an important task.
  + Need to workout a policy for the queue system.
  + As we discussed in the morning, it is currently not possible to have fine-grain control on GPU usage.
  + Tools for estimating the O(N) scaling/complexity of software and data input.
  + Tools for estimating how “parallel” the software and data input is, i.e. what is the optimal number of nodes/cores/resources to throw at a given problem.
  + Dynamic (re-)estimation of job runtimes while the job is running, such that e.g. a job scheduler can dynamically update how jobs are scheduled
* threads, cores, processes… abstraction is needed
* Not enough diversity in hiring sysadmins
  + Need to define the level of support the sysadmin group wants to deliver.
  + More employment is needed if the level of support requires.

### Solution A

How will we know that it’s done and successful?

* *What is the end state. How does it help and who does it help?*
* *Extending a job could be done via web or command line automatically, but should not be for free… otherwise what should be preventing me doing it on a regular base*

How to do it?

* *What are the steps?*

Which obstacles?

* *Which problems do you anticipate?*

Who needs to be involved?

* *(key players: both tech and admin)*

### More use of interactive partitions

How will we know that it’s done and successful?

* Light users / new users start with interactive partitions

How to do it?

* Standard interactive partition name: *interactive*
* Some interface recommended, e.g. sinteractive
* CSC's taito-shell service gives interactive job with just ssh: <https://research.csc.fi/en/taito-interactive-batch-jobs>

Which obstacles?

* Inefficient use of resources - downtime. Limit to small jobs then.

Who needs to be involved?

* cluster admins
* Allocation groups

### Solution B

...

# Organizational

### Papercuts/vision

* Buy big resources first, usability later
* Need to agree on a common framework for communication (organizational issue)
* Look-and-feel differs between resources and nomenclature differ sometimes. This is confusing for users
* Need good guideline for data sharing

### Solution - Buy big resources first, usability later

How will we know that it’s done and successful?

* *A usable system from the start. Usability not an afterthought, that isn’t ready directly.*

How to do it?

* *User survey and needs assessment study.*
* *Add more user codes in benchmark?*
* *Technical architecture based on findings in first step*
* *HPC System procured based on second step*
* *“Usability requirement in procurement”*
* *Userdriven resources*

Which obstacles?

* *Getting input from users*
* *Technology for the sake of technology.*
* *User needs doesn’t come first. Technology is more important than user needs.*
* *Lack of synchronisation between local and national organisations*
* *Lack of coordination between local, regional, national and nordic organizations*

Who needs to be involved?

* *Strategy must be anchored in organisation*
* *Reference groups of key users take part in defining the requirements for the resource.*
* *Coordinate user needs on a national level.*

### More standardization

How will we know that it’s done and successful?

* Each cluster doesn't get its own new documentation
* Home directories similar paths
* Standard mount points
* (more) portable submission scripts

How to do it?

* (related: OpenHPC)
* Follow FHS more
* At least have organizations maintain standards between their own upgrades
* NeIC can exert some pressure or provide funding as motivation to move towards standards

Which obstacles?

* Everyone wants to be similar to what they used to do, not the "new standard"
* Which standard to pick? Who decides? Picking anything will annoy most people

Who needs to be involved?

* *(key players: both tech and admin)*

### Solution A

How will we know that it’s done and successful?

* *What is the end state. How does it help and who does it help?*

How to do it?

* *What are the steps?*

Which obstacles?

* *Which problems do you anticipate?*

Who needs to be involved?

* *(key players: both tech and admin)*

### Solution B

...

# Software installation

* “Normal” installation instructions don’t work. -> no sudo, apt-get, ...
* User level software installation (without the intervention of Admin)

### Papercuts/vision

Reinstallation/Chroot installation with sudo rights for user. Lives according to “Walltime”. Docker/Singularity solves this?

Have domain knowledge for installing software (sometimes, software is installed but not usable)

### Solution A

How will we know that it’s done and successful?

* *What is the end state. How does it help and who does it help?*

How to do it?

* *What are the steps?*

Which obstacles?

* *Which problems do you anticipate?*

Who needs to be involved?

* *(key players: both tech and admin)*

### Solution B

...

# ...

# Categories of problems to discuss

* Accounts
* Firewalls and access limitations
* Storage
* Software installations
* Compilers
* Non-compiled software distribution, e.g. Python, R, ...
* Linux issues
* Difficulty of sharing
* Management and organizational silos
* Too many sources of documentation and information: local, national, international.
* Staff time pressures
* Follow-up meeting planning