UNICORE 8 - development directions

Bernd Schuller
September 21, 2018
Outline

- Where is UNICORE right now
  - Strengths, weaknesses, ...

- UNICORE 8 plans

- UNICORE 8 roadmap

  - Note: will be mostly focusing on core features!
Strengths

- Allows integration of HPC into federated environments
  - Federated authentication, local authorization, site keeps full control

- Can simplify HPC use for non-experts (if done correctly :-) )
  - Application concept, abstract resource model

- Feature-rich and flexible

- Portability, independence of the OS and batch system

- Proven extensibility
Weaknesses – complexity

- Several servers on different physical servers, requiring matching entries in config files
- Many config files (IDB, xnjs.xml, wsrflite.xml). Many different formats. Not very intuitive.
- Internal details in admin-targeted config files (xnjs.xml, wsrflite.xml)
- X.509 PKI / trusted CA management required
- Manual adaptation to local BSS (queues, nodes, …)
Weaknesses – HPC back end

- Resource model
  - Needs an update – current heterogeneous machines cannot be modeled properly
  - Weak support for „expert“ users who require specific BSS or cluster features

- Applications
  - Common cases require too many tricks
    - *e.g. a serial preparation step before the parallel main simulation run as in the Neuron app in HBP*
  - Expert help required to setup applications
Weaknesses – deployment and packaging

- Full system has (too) many moving parts
  - Overkill for single-cluster-webportal style
  - Difficult for typical admins (who always have little time)

- Lack simple setup for simple cases
  - Nanomatch style – everything on one cluster
  - Hook into existing user management (passwords / ssh-Keys) without requiring Unity
UNICORE 7 was released January 2014

... high time for UNICORE 8
UNICORE 8 – basic guidelines

- Remove complexity

- Strengthen core use cases

- REST API is the primary API
Remove complexity

- Remove stuff: CIP (already gone), BES, Gridbean service, virtual TSS, iRods, Hadoop execution, „default SMS“, … what else?
- Disable stuff by default: data triggered processing

- Simplify config files
  - xnjs.xml: remove internal details, simple property format
  - wsrflite.xml: remove internal details, simple property format
  - High-level module / feature definitions instead of internal details
  - IDB: introduce simpler format (JSON)

- More later?
  - Access control policies – nobody ever changes them, right?
  - Standard policies could be hard-coded, with the XACML as an option?
Core use cases – federated access to HPC

- Model current clusters
- Improve abstract batch submission
- Support low-level batch system interaction
  - Sometimes UNICORE is the only way to access a system
  - We do not want to limit what the user can do
IDB / resource model

- Support heterogeneous clusters
  - Central concepts: partitions (as in Slurm)
  - Define runtime
  - Use nodes OR total CPUs
  - Less is more: avoid passing resources like memory or CPUs per node

- Remove execution environments – redundant concept, rarely used as intended. Can be replaced by applications if necessary.
Jobs and Applications

- Extend job features
  - Job modes
    - *Interactive / login node* (now has to be done via `UC_PREFER:_INTERACTIVE_EXECUTION`)
    - *Batch* (like now)
    - „*Raw“ Batch“ - essentially only „sbatch userscript“
  
- Applications
  - Default resource specs (default nodes, node constraints, default partition…)
  - Serial prepare / postprocess (user extensible)
  - Prolog / epilog (user extensible)
REST API

- Make it the primary API
  - SOAP/XML will go eventually (in UNICORE 9)

- REST delegation (User → A → B) using JWT (JSON Web Tokens)
  - User authenticates to server A
  - When server A makes REST call to B,
    - JWT token asserting identity of the user’s
    - Signed with A’s secret (RSA private key)
    - B validates with RSA public key (→ registry)
  - Can be chained A → B → C…
  - Unity not required for delegation to work
Some new stuff

- Batch system features
  - Get job info (allocated nodes, scheduling info)
  - Support tunneling scenarios (for visualisation or steering)
  - Auto-configure UNICORE/X IDB by querying the BSS?

- AuthN supporting simpler scenarios
  - Support username+ssh key and username+PAM password
  - SSH pubkey can be retrieved automatically via TSI and/or uftpd
Some new stuff – 2

- Notification support
  - Get updates about job status changes (queued/started/finished/failed)
  - REST endpoint in job description
- Notification endpoints
  - Endpoint available on the portal and other relevant servers
  - User „Home“ endpoint for collecting notifications. User clients can retrieve batch status updates instead of polling every job
UNICORE 8 roadmap

- 7.x goes into „maintenance mode“
  - → only bugfixes, or very important feature requests

- XNJS 2.0
  - New component and configuration management
  - New resource model, JSON IDB

- USE (UNICORE Services Environment) 4.0
  - Simpler configuration (less internal details...)
  - Base classes for building SOAP/REST aware clients
  - REST delegation using JWT
UNICORE 8 roadmap

- UNICORE/X 2
  - Remove stuff, cleanup
  - New component / config management XNJS 2 and USE 4

- Clients
  - Dual SOAP / REST
    - *Use REST whenever available*
  - High-level properties / APIs independent of XML or JSON
  - Low-level interfaces (SOAP/REST) available
  - Smart(er) authentication
UNICORE 8 – workflow system outlook

- New component / config management XNJS 2 and USE 4
- Provide the option for a „merged“ workflow system
  - brokering / job running
- Use notification endpoints
- Remove Tracer
- Update location manager – allow general use as a file catalogue?

→ This will require a lot of work and some heavy lifting … e.g. updating the servorch / brokering parts
Questions?