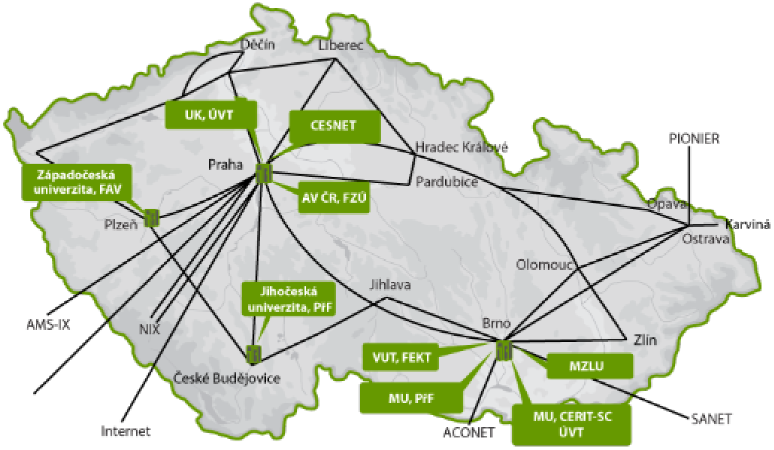
Using L&B to Monitor Torque Jobs across a National Grid

Voců, M., Šustr, Z., Kouřil, D., Dvořák, F., Ruda, M., Tóth, Š., Sitera, J., Filipovič, J., Poul, M., Matyska, L.
CESNET, Czech Republic

National Grid

- geographically distributed computing centres
- clusters with both thin and fat nodes, currently ~3500 cores



National Grid

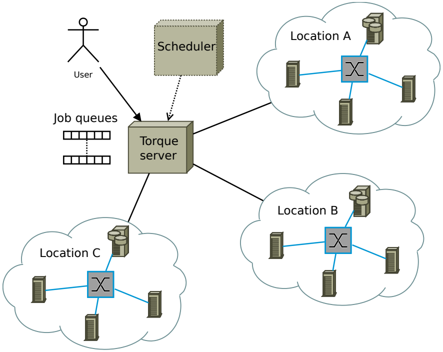
Looks like grid, but we have:

- single user management system (Perun)
  - developed in–house
  - based on federated identities (eduID.cz)
  - supports multiple VOs
- common authentication system (Kerberos)
  - one user realm per VO
- single job management system (Torque)
  - modified to support virtualization
- shared filesystems (NFS4)
  - main storage co–located with big clusters
  - all data visible everywhere (but no location transparency)

National Grid - Job Management

Current state:

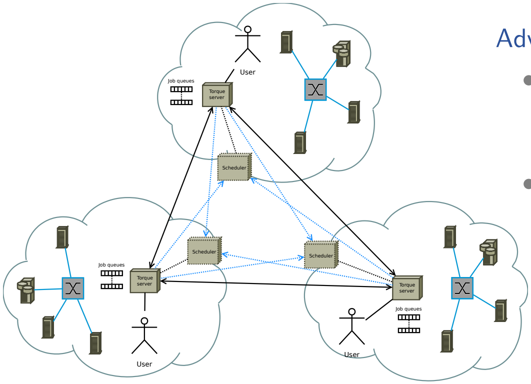
- moved from using PBSPro to open–source Torque
- one big cluster with central Torque server/scheduler
- single point of failure
- may not handle the planned cluster extensions (in the order of magnitude in #cores)
- needs tuning for network latencies



National Grid - Job Management

Next phase (now in testing):

- clusters at main sites, Torque server/scheduler deployed at each cluster
- schedulers cooperate in P2P structure, migrating jobs between servers



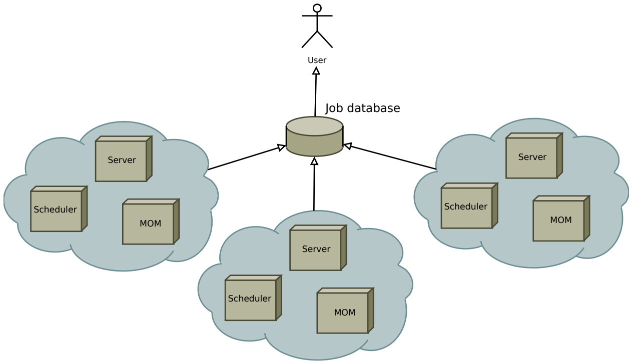
Advantages:

- local cluster always accessible even when the servers are partitioned
- scales well

National Grid - Job Management

We need to keep track of the jobs flowing through the infrastructure:

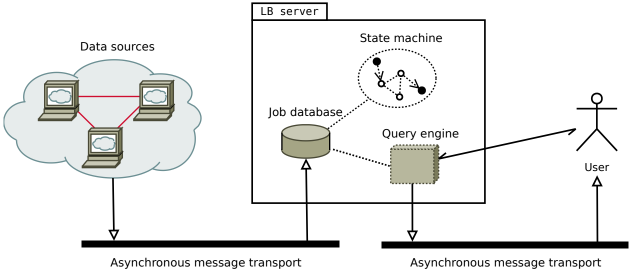
- single job database for all Torque servers, central point of contact for users
- non-intrusive, should not influence normal job processing
- should provide the same information Torque provides, more expressive queries



L&B overview

L&B is used to keep track of jobs in grid, in production for Datagrid (and successors) since 2003:

- fast, asynchronous and secure message transport layer
- relational database for job data
- pluggable state-machines for various job types (currently WMS, Condor, CREAM, PBS/Torque, file transfers supported)
- notification mechanims for informing users about job status changes
- query engine supporting both notifications and direct user queries



Motivation for L&B use

- separation of concerns
  - job data are handled and archived separately from Torque
- reliability
  - data collected and delivered from Torque components independently
  - eliminate single point of failure by using redundant L&B servers (see future work)
- flexibility
  - support for various Torque deployment/configuration schemes (central, distributed P2P, distributed hierarchical, . . .)
  - user job annotation (user tags)
  - custom notification of job state using legacy L&B client or standard messaging infrastructure (eg. STOMP)
- advanced use cases
  - post-termination job data processing, data mining
  - data archivation
  - keeping track of complex job/subjob structures

Implementation – overview

- use Kerberos instead of GSI as an authentication mechanism
- instrument Torque components (server, scheduler, MOM) to log data to L&B:
  - job description – identification, required resources
  - job location – detailed information when job is transferred between Torque components
  - job status – Torque job state/substate, PID of running processes, used resources
- enhance Torque data structures with L&B specific information (server and job attributes)
- add Torque-specific data messages (events), update state machine
- provide `qstat`-compatible replacement using L&B database

Implementation – Torque

L&B specific attributes in Torque:

- L&B server address – configurable server attribute
- L&B jobid – job attribute, assigned when job is submitted by user
- L&B sequence number – job attribute, enables correct ordering of messages at L&B server

Instrumentation of Torque components (server, scheduler, MOM):

- messages are sent (at least) whenever job state/substate changes
- MOM logs job resource usage periodically
- message delivery is unobtrusive, done by external process

Implementation – L&B

PBS/Torque jobs already supported:

- PBS job state–machine
- events collected by PBS log parsing

Access to Torque source code allows to instrument Torque to get more information:

- define message formats for data from Torque
  - new data types – resource list (list of *name* = *value* pairs)
  - new message types:
    - message format for each Torque state change
    - code is generated from format description
- enhance Torque specific state machine
  - maps Torque states/substates into L&B job states
  - Torque internal state/substate visible in L&B job status data (job attribute)
  - resources (requested and used) also stored in L&B job attribute

Implementation – Kerberos

Replace GSI with Kerberos:

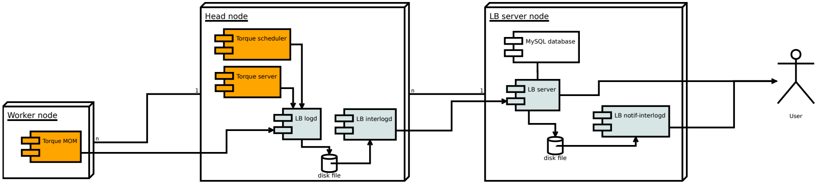
- L&B uses its own abstraction of GSSAPI
- all neccessary modifications contained within single abstraction library (needed to drop VOMS support, though)
- link with Kerberos GSSAPI implementation instead of GSI GSSAPI

Consequences:

- clients (from the protocol point of view) need external nanny to keep Kerberos tickets valid
- identity identifier (string in L&B) in different format (`michal@META` instead of `/CN=Michal Vocu/O=...`)
- need identity translation eg. when displaying job status in browser with user certificate

Deployment

- Torque – compiled from (modified) sources
- L&B – installable packages (RPM, .deb)



Conclusions and future work

Conclusions:

- all necessary software modifications were straightforward and well contained
- data collected asynchronously from all Torque components
  - users have more information than from the Torque server(s) alone
- modified Torque is about to be deployed in testing environment

Future work:

- distributed L&B to enhance reliability
- mapping of Kerberos and X.509 identities to support access both from CLI and browsers