

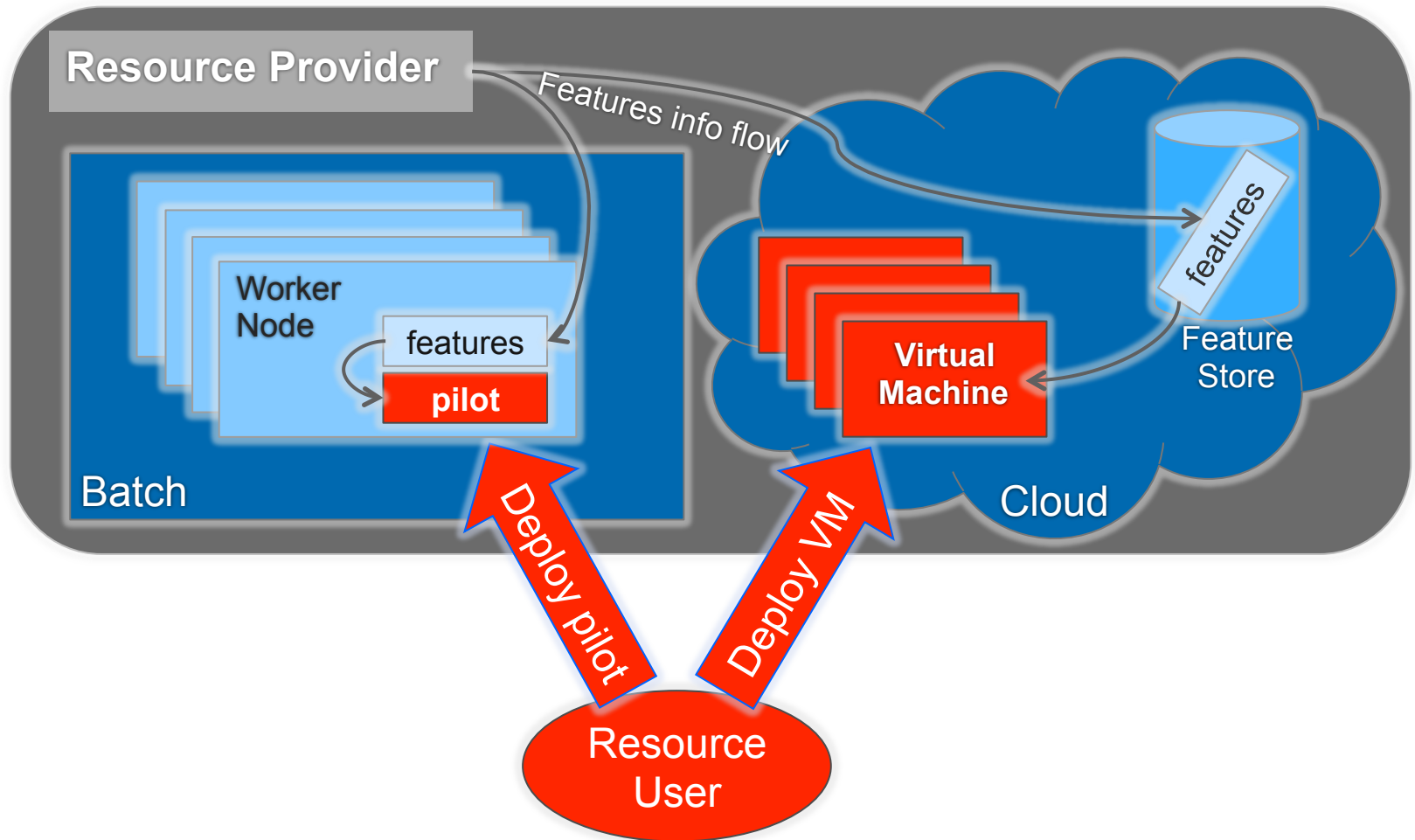
Job / Machine Features TF Status

Stefan Roiser / IT-SDC
GDB / 10 Dec '14

Content

- Recap of Machine / Job Features
- Batch Infrastructure Status
- Virtualized Infrastructure Status
- Next steps

Machine / Job Features Overview



What information contain ...

Machine Features:

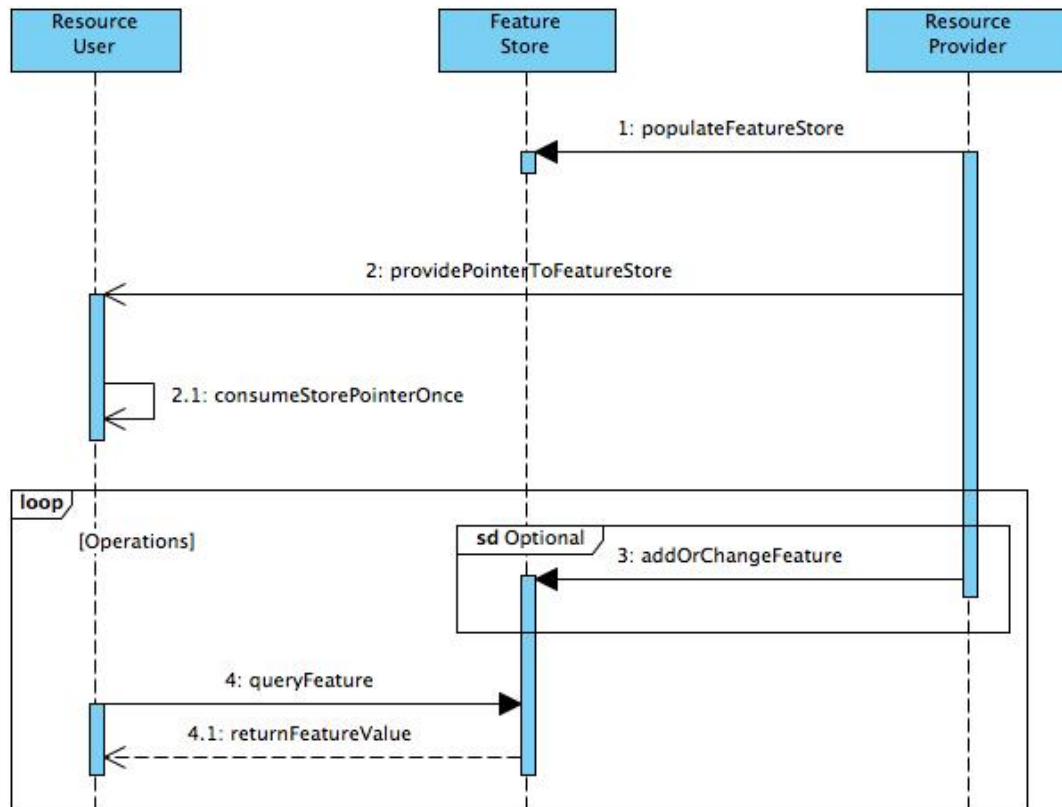
- WN power
- Shutdown time
- Total # of jobslots
- # of physical cores
- # of logical cores

Job Features:

- CPU time limits
- Wall time limits
- Scratch space limit
- Job start time
- Memory limit
- Allocated # of cores

... and possibly many more that we agree to have as the WLCG community ...

How to propagate and use features



1. Provider populates the feature store
2. Provider gives pointer to store
 1. User consumes this pointer **ONCE**
3. Optionally provider changes / adds features in store
4. User consumes features at its pace

Options during intialisation

Where the user can find the feature store pointer?

Environment variables

- \$JOBFEATURES
- \$MACHINEFEATURES

or

Metadata service

- <http://169.254.169.254/latest/jobfeatures>
- <http://169.254.169.254/latest/machinefeatures>

What are possible values of the feature store pointer?

Value starts with a slash “/”

eg. “/var/machinefeatures”

-> features served via the local file system

or

Value starts with “http(s)://”

eg. “http://mjf/wn42/jobfeat”

-> features served via a web service

Status Batch System

- Implementations for almost all batch system types available
 - LSF, PBS/Torque, SGE, HTCondor,
 - Slurm close to be finished
- Note: These are “reference” implementations which may need some customization at your local batch system
- Plus another alternative ...

Virtualized Infrastructure Status

- First implementation proposed on top of couchdb
 - Produced differences between the virtualized/ http(s) and bare-metal/file based implementation
 - ... which initiated the introduction of the mjf.py client to abstract those
- Alternative implementation possible, based e.g. on apache which mimics the batch system implementation 1 to 1
 - Several advantages ...

Nice goodies of the apache system

- With e.g. python the location (file/http) does not matter, e.g. one can query the features always with
 - No abstraction needed, therefore remove the client

```
>>> urllib.urlopen('http://mjf-server/mf/phys_cores').read()  
'8'  
>>> urllib.urlopen('/var/mf/phys_cores').read()  
'8'
```

- Authentication (e.g. x509) easier than with couchdb
- Easily horizontally scalable
- Could also be used for batch systems

Status & ToDo

- File system based services
 - At least two sites for each implementation done
 - Get more sites to deploy those implementations
 - Reference implementations in the repository
- Apache based service
 - Running at one UK site for its pool of VMs
 - ... and on one experiment VOBox for its Openstack VMs
 - Reference configuration available in the repository
- Client
 - Not needed anymore, therefore abandoned
- Bi-directional use case
 - Initial request from CMS not clear if still needed
 - No other experiments expressed interest so far
 - -> shall we pursue?

Conclusion

- Architecture, workflow, implementations and configurations are defined and available
- Let's go for deployment on the remaining sites
 - Propose to finish this TF before Easter next year
 - Would have lasted 1 ½ years then ...

Many thanks to

- Jan Just Keijser / Torque/PBS
- Manfred Alef / SGE
- Marian Zvada / HTCondor
- Ulrich Schwickerath / LSF
- Ulf Tigerstedt / SLURM
- Andrew McNab / apache based
- Maarten Litmaath / ALICE
- Miguel Martinez Pedreira / ALICE
- Dirk Hufnagel / CMS

Additional Info

- **Twiki:**

<https://twiki.cern.ch/twiki/bin/view/LCG/MachineJobFeatures>

- **Repository:**

<https://github.com/roiser/JobMachineFeatures>

- **Mailing List:**

wlcg-ops-coord-tf-machinejobfeatures@cern.ch

- **Mailing List Archive:**

<https://groups.cern.ch/group/wlcg-ops-coord-tf-machinejobfeatures/default.aspx>

