

Table of Contents

Lightweight sites to end of 2016.....	1
Introduction.....	1
Lightweight Sites Profile.....	1
May 2016 GDB session summary.....	1
Boundaries.....	1
Storage.....	2
T2 vs. T3 sites.....	2
T2 simplifications.....	2
Selected items from Mikalai's talk.....	2
CE and batch systems.....	3
Accounting.....	3
Cloud systems.....	3
Selected items Andrew's talk.....	3
Authorization systems.....	4
Information systems.....	4
Configuration.....	4
Simplified deployment.....	4
Selected items from US-ATLAS talk: Ubiquitous Edge Platform.....	4
Selected items from US-CMS talk: Tier-3 in a box.....	5
Better documentation.....	5
Monitoring.....	5
Experiment approaches and views.....	5
Selected items from Alessandra's talk.....	5
Summary by Aug 2016.....	5
Followup in Ops Coordination.....	6

Lightweight sites to end of 2016

This is an archived page, reflecting the status at the end of 2016.

Introduction

One of the goals of WLCG Operations Coordination activities is to help simplify what the majority of the sites, i.e. the smaller ones, need to do to be able to contribute resources in a useful manner, i.e. with *large benefits compared to efforts invested*.

Classic grid sites may profit from *simpler mechanisms* to deploy and manage services. Moreover, we may be able to get rid of some service types in the end.

New sites may rather want to go into one of the *cloud directions* that we will collect and document.

There may be different options also *depending on the experiment(s)* that the site supports.

There is no one-size-fits-all solution. We will rather have a *matrix of possible approaches*, allowing any site to check which ones could work in its situation, and then pick the best.

Lightweight Sites Profile

The Lightweight Sites Profile collects interfaces we need to retain or replicate.

May 2016 GDB session summary

agenda [↗](#)

Boundaries

- The session was about small sites
- T0 and T1 are not directly targeted
 - ◆ But may profit from common simplifications
- Build on earlier discussions and ideas
 - ◆ Optimisation of operational costs [↗](#), presented in Okinawa
 - ◆ Resource provisioning [↗](#), ditto
 - ◆ Several sessions in the WLCG workshop in Lisbon [↗](#)
- Storage and data access are mostly handled elsewhere
 - ◆ Various demonstrator projects e.g. presented in the April MB [↗](#)
 - ◆ Federations, caches, Ceph,
- Here we are more concerned with services needed to enable computing at a site
 - ◆ CE
 - ◆ Batch system
 - ◆ Cloud setups
 - ◆ AuthZ system
 - ◆ Info system
 - ◆ Accounting
 - ◆ CVMFS Squid
 - ◆ Monitoring
- Storage service deployment may also profit from generic simplifications pursued here

Storage

- ATLAS have 75% of their storage provided by just 30 sites
 - ◆ the remaining 25% are provided by a long tail of small sites
 - ◆ storage is difficult to manage and requires significant support effort
 - ◆ small sites typically cannot afford the desired level of local support
- Big vs. small sites
 - ◆ "Real" storage vs. caches?
- Caches
 - ◆ Inconsistencies are "normal" and should be resolved automatically
 - ◆ Internal per site vs. externally visible?
 - ◆ ARC CE has a cache by design
- Object stores?
 - ◆ *Ceph* is on the rise
 - ◆ More common, less grid-specific
 - ◇ More defensible to spend effort on

T2 vs. T3 sites

- T3: a site that did not sign the WLCG MoU
- T3 sites typically are dedicated to a single experiment
 - ◆ Can take advantage of shortcuts
 - ◆ Can be pure AliEn / DIRAC / ... sites
 - ◆ E.g. AliEn integration with OpenStack (Bergen Univ. Coll.)
 - ◆ ...
- T2 sites have rules that apply
 - ◆ Accounting into EGI / OSG / WLCG repository
 - ◆ Availability / Reliability targets
 - ◆ EGI: presence in the info system, at least for Ops VO
 - ◆ Security regulations
 - ◇ Mandatory OS and MW updates and upgrades
 - ◇ Isolation & traceability
 - ◇ Security tests and challenges

T2 simplifications

- **Reduce** the catalog of required services, where possible
- **Replace** classic, complex services with new, simpler portfolio
 - ◆ And more **common** technologies, less grid-specific
- **Simplify** deployment, maintenance and operation of services
- Some sites could offer a partial portfolio, e.g. just "worker nodes"
 - ◆ Need to get their contributions properly recognized

Selected items from Mikalai's talk

talk [↗](#)

- Deploy via easy installers
 - ◆ But: need to make such tools, support many options, ...
 - ◆ Grid services: make use of VM/CT/... images

- ◆ Cloud services: e.g. use OpenStack Fuel
- Small sites may provide just "worker nodes"
 - ◆ Drive them directly from "WLCG"
 - ◆ Or integrate them first per NGI/region/...
 - ◇ Compatibility?
- Outsource complex management to remote experts

CE and batch systems

- CE flavors
 - ◆ ARC
 - ◆ CREAM
 - ◇ Said to be the most complex
 - ◇ Maybe needed for other VOs supported by the site
 - ◆ HTCondor
 - ◇ On the rise
- Batch systems
 - ◆ More variety, possibly with complex configurations
 - ◆ May not be easy to change at a site (other customers)
 - ◆ The majority of sites still use PBS/Torque
 - ◇ Still the default in EGI
 - ◆ HTCondor on the rise
 - ◇ Default in OSG
- Reductions in the phase space would help

Accounting

- Can it be simplified for classic grid sites?
 - ◆ A site's APEL host assembles records from CE and batch system
 - ◆ APEL would benefit from fewer CE and batch system flavors to support
- ARC CE publishes directly into central APEL service
- HTCondor CE?
- Transition toward cloud systems ought to help

Cloud systems

- Tap directly into local cloud deployments at sites
 - ◆ But OpenStack etc. may not be so easy for them either
- Paradigm shift: batch slots are replaced with VM instances
 - ◆ Need to ensure proper accounting
- Experiments can handle this today, but:
 - ◆ would like to see less variety, fewer interfaces to deal with
 - ◆ do not want to become sysadmins of the acquired resources
- Other supported VOs may be unable to use such resources
 - ◆ Sites may still need to run their classic setups instead
 - ◇ Or even in parallel

Selected items Andrew's talk

talk [↗](#)

- Instantiate ready-made VMs at sites

Selected items from Mikalai's talk

- ◆ Ready to take on work for a supported experiment
- Vac: autonomous hypervisors
 - ◆ Easy installer: Vac-in-a-Box
- Vcycle: instantiate via OpenStack API etc.
 - ◆ Can be done remotely
- VMs shut themselves down when there is no work for their VO
 - ◆ Allow fair shares
- In production in UK for ATLAS, LHCb, GridPP
 - ◆ CMS in progress

Authorization systems

- Argus now is supported under the aegis of Indigo-DataCloud
 - ◆ Version 1.7 is close to official release
 - ◇ Important improvements
 - ◇ CERN has upgraded already
- GUMS is a cornerstone of OSG sites since many years

Information systems

- Information System TF [TF](#) explores simplifications for WLCG
- BDII services may no longer be needed for WLCG
 - ◆ Still required for other VOs and EGI ops monitoring
 - ◆ Aiming for less work to support WLCG VOs

Configuration

- Slow but steady move towards Puppet
 - ◆ YAAM still being used for many services at many sites
 - ◆ Some prefer Ansible, CFEngine, Chef, Quattor, Salt, ...
 - ◇ But they presumably know what they are doing!
- Small sites also need help there
- Shared modules?
 - ◆ Not a lot of reuse evidence yet?
- DPM comes with self-contained mini Puppet
 - ◆ An idea for other services?

Simplified deployment

- Can we provide VM images that require little local configuration?
 - ◆ The easiest test case may well be the Squid for CVMFS
 - ◆ Complex services would benefit more
 - ◇ But is the idea realistic for them?
- Would containers be better?
 - ◆ Could even work at sites without a (compatible) cloud infrastructure
- Ready-made (HW +) SW solutions?
 - ◆ Deployed in a DMZ like perfSONAR
 - ◆ Remotely operated by experts

Selected items from US-ATLAS talk: Ubiquitous Edge Platform

[talk](#)

- Cyber-infrastructure "substrate" within trusted zones
- Remotely deploy services on top
- Science DMZ edge container platform HW spec
- Some security and network questions not yet fully resolved

Selected items from US-CMS talk: Tier-3 in a box

talk [↗](#)

- Host to submit to local and remote resources
- Plus CVMFS
- Plus Xrootd cache & server
 - ◆ Both connected to global federation
- In use at 5 Univ. of California

Better documentation

- Online video tutorials?
- Or rather improve the textual documentation?
 - ◆ Screenshots in some places might be sufficient
- Cost-benefit analysis?

Monitoring

- Sites ought to have local fabric monitoring
 - ◆ Nagios, Ganglia,
- SAM test failures ought to raise local alerts
 - ◆ And be understandable, well-documented
 - ◆ Supported for Nagios and used by some sites
- Experiment-specific monitoring could support alert subscriptions
 - ◆ RSS, Atom, e-mail
 - ◆ MonALISA does that

Experiment approaches and views

- Natural for dedicated sites
- Try to identify / encourage commonalities?

Selected items from Alessandra's talk

talk [↗](#)

- Experiments want less variety to deal with
 - ◆ But still be able to run "anywhere"
- Some MW choices are better fits to the needs of a particular experiment
- Developments in experiments may help
 - ◆ Example: ATLAS Event Service allows reduced QoS requirements on sites

Summary by Aug 2016

- **Reduce** the set of services required per site
 - ◆ Some services might no longer be needed at all
 - ◆ Small sites could provide a partial portfolio

◇ And still get proper recognition

- **Replace** complex grid-specific services with simpler and/or common technologies
 - ◆ Local cloud setups can be used directly
 - ◆ Complex aspects can be handled by remote experts
 - ◆ Site admins may not need to acquire grid expertise
- **Simplify** deployment of whatever remains
 - ◆ Easy installers for grid / cloud services
 - ◆ Science DMZ: only the HW must be managed locally
- Experiment computing model **evolution** can help
 - ◆ Expect lower QoS from most of the sites
- We aim for a matrix of **recipes**
 - ◆ Some may prevail, while others fade away

Followup in Ops Coordination

In Oct and Nov 2016 a **questionnaire** was filled out by a representative set of EGI sites.

The results were presented[↗] in the Dec Ops Coordination meeting[↗].

The largest benefits are expected to come from **shared repositories** for:

- OpenStack images
- Docker containers
- Puppet modules
 - ◆ One already exists: <https://github.com/HEP-puppet>[↗]

This topic: LCG > LightweightSites2016

Topic revision: r1 - 2018-05-09 - AndrewMcNab



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback