

Report on Cream Migration Workshop

J. Flix

GDB @ EGI Conference 2019

EGI Conference 2019, 6-8 May 2019, Amsterdam

Workshop sessions

13:00

Introduction and setting the scene. CREAM-CE decommissioning.	
<i>VK1,2 SURFsara, WCW Congress Centre</i>	13:30 - 13:35
Plans for EGI migration readiness	<i>Alessandro PAOLINI et al.</i>
<i>VK1,2 SURFsara, WCW Congress Centre</i>	13:35 - 13:45
ARC Introduction	<i>Balazs KONYA</i>
<i>VK1,2 SURFsara, WCW Congress Centre</i>	13:45 - 14:05
ARC Site Experience	<i>Catalin CONDURACHE</i>
<i>VK1,2 SURFsara, WCW Congress Centre</i>	14:05 - 14:15
HTCondor and HTCondor-CE Introduction (focussing on HEP)	<i>Brian BOCKELMAN et al.</i>
<i>VK1,2 SURFsara, WCW Congress Centre</i>	14:15 - 14:30
APEL accounting with HTCondor	<i>Stephen JONES</i>
<i>VK1,2 SURFsara, WCW Congress Centre</i>	14:30 - 14:40
HTCondor-CE Site Experience (Pic)	<i>Jose FLIX</i>
<i>VK1,2 SURFsara, WCW Congress Centre</i>	14:40 - 14:50
HTCondor-CE Site Experience (Frascati (INFN))	<i>Stefano DALPRA</i>
<i>VK1,2 SURFsara, WCW Congress Centre</i>	14:50 - 15:00

14:00

16:00

No CE solutions. Introduction.	<i>Dr. Maarten LITMAATH</i>
<i>VK1,2 SURFsara, WCW Congress Centre</i>	15:30 - 15:40
No CE. VAC & VCycle	<i>Andrew MCNAB</i>
<i>VK1,2 SURFsara, WCW Congress Centre</i>	15:40 - 15:55
No CE. DODAS.	<i>Daniele SPIGA</i>
<i>VK1,2 SURFsara, WCW Congress Centre</i>	15:55 - 16:10
Pros and cons of various CE alternatives	<i>Stephen JONES</i>
<i>VK1,2 SURFsara, WCW Congress Centre</i>	16:10 - 16:20
SIMPLE framework (Easy deployment)	<i>Mr. Mayank SHARMA</i>
<i>VK1,2 SURFsara, WCW Congress Centre</i>	16:20 - 16:35
Panel discussion	
<i>VK1,2 SURFsara, WCW Congress Centre</i>	16:35 - 16:45
Dirac and No CE	<i>Sorina POP</i>
<i>VK1,2 SURFsara, WCW Congress Centre</i>	16:45 - 17:00

15:00

17:00

2 sessions - 13 talks
1 panel discussion
45 people attended

Workshop sessions

13:00

Introduction and setting the scene. CREAM-CE decommissioning.	
VK1,2 SURFsara, WCW Congress Centre	13:30 - 13:35
Plans for EGI migration readiness	Alessandro PAOLINI et al. 
VK1,2 SURFsara, WCW Congress Centre	13:35 - 13:45
ARC Introduction	Balazs KONYA
VK1,2 SURFsara, WCW Congress Centre	13:45 - 14:05
ARC Site Experience	Catalin CONDURACHE
VK1,2 SURFsara, WCW Congress Centre	14:05 - 14:15
HTCondor and HTCondor-CE Introduction (focussing on HEP)	Brian BOCKELMAN et al.
VK1,2 SURFsara, WCW Congress Centre	14:15 - 14:30
APEL accounting with HTCondor	Stephen JONES 
VK1,2 SURFsara, WCW Congress Centre	14:30 - 14:40
HTCondor-CE Site Experience (Pic)	Jose FLIX
VK1,2 SURFsara, WCW Congress Centre	14:40 - 14:50
HTCondor-CE Site Experience (Frascati (INFN))	Stefano DALPRA
VK1,2 SURFsara, WCW Congress Centre	14:50 - 15:00

14:00

16:00

No CE solutions. Introduction.	Dr. Maarten LITMAATH
VK1,2 SURFsara, WCW Congress Centre	15:30 - 15:40
No CE. VAC & Vcycle	Andrew MCNAB
VK1,2 SURFsara, WCW Congress Centre	15:40 - 15:55
No CE. DODAS.	Daniele SPIGA
VK1,2 SURFsara, WCW Congress Centre	15:55 - 16:10
Pros and cons of various CE alternatives	Stephen JONES 
VK1,2 SURFsara, WCW Congress Centre	16:10 - 16:20
SIMPLE framework (Easy deployment)	Mr. Mayank SHARMA
VK1,2 SURFsara, WCW Congress Centre	16:20 - 16:35
Panel discussion	
VK1,2 SURFsara, WCW Congress Centre	16:35 - 16:45
Dirac and No CE	Sorina POP 
VK1,2 SURFsara, WCW Congress Centre	16:45 - 17:00

15:00

17:00

2 sessions - ~~13~~ 12 talks
 1 ~~panel~~ open discussion
 45 people attended



The integration procedure

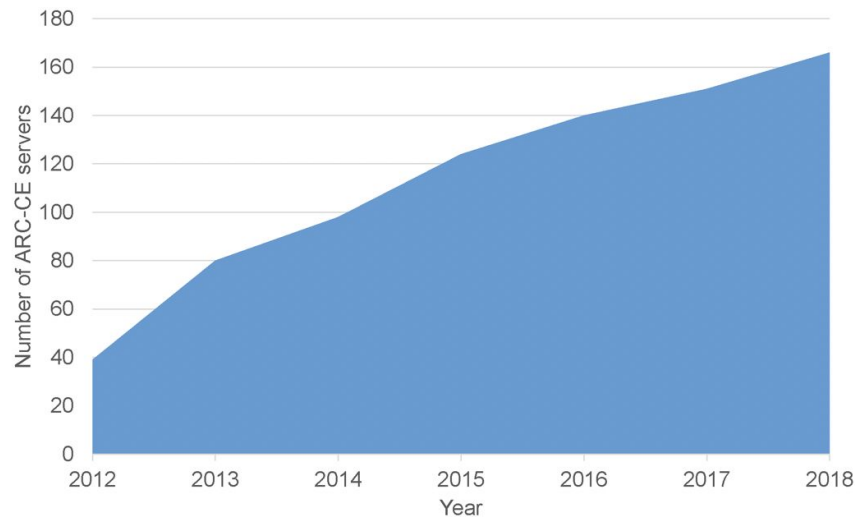
<https://wiki.egi.eu/wiki/PROC19>

- Integration steps for the new technology:
 - Underpinning Agreement
 - Configuration management
 - Information System
 - Monitoring
 - Operations (ROD) Dashboard
 - Support
 - Accounting
 - UMD
 - VM image Marketplace
 - Documentation
 - Security

- HTCondor-CE service present in the GOCDB
- GGUS Support Unit created
- Security being worked out with EGI CSIRT
- IS info-provider available for testing
- APEL accounting for HTCondor BS ready (previous work by PIC, improved and integrated by Steve Jones) - Liverpool tested it, other sites testing it
- HTCondor-CE packaged without dependencies on OSG software
- Nagios probes for direct submission to HTCondor-CE
- Documentation being written



ARC-CE instances in GOCDB



compare to appx 370 CREAM-CE instances



Integration with the (WLCG) world

- ARC interacts with:
 - Storage Elements: Dcache, StoRM, DPM, ...
 - Security services: ARGUS, VOMS
 - Accounting servers: APEL, SGAS
 - Info and monitoring services: Top-BDII, GOCDDB,...
- ARC-CE is fully integrated into WLCG and EGI operations
 - Registered service in GOCDDB, accounting reports sent to APEL by ARC's JURA module, GLUE2 Info
 - Part of UMD releases, User support via GGUS
- Widely used by ATLAS sites, also by LHCb, CMS, ALICE and smaller VOs that are supported by respective WLCG sites



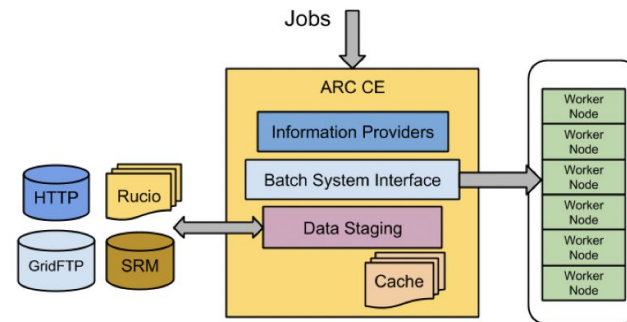
WHY ARC?

Powerfull data staging with CACHE

- The DTR subsystem of ARC CE performs the critical role of transferring input and output data for jobs, [arex/data-staging]
 - Generally copying data between a shared file system and Grid storage
 - transfershares, speedcontrol, transferretries, etc..

tech description:
https://wiki.nordugrid.org/wiki/Data_Staging
- The CACHE module of an ARC CE may keep a cache of input data on the shared file system, [arex/cache], [arex/cache/cleaner]
 - Jobs requiring already cached files do not need to re-download them
 - Cache is self-managing using LRU and/or a file lifetime based cleanup
 - Multiple cachedirs, cache draining

tech description: Section 6.4 sysadmin guide



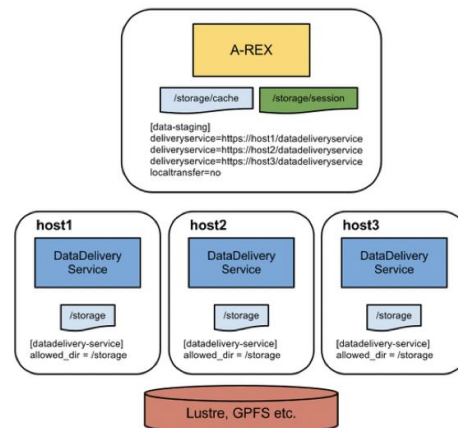
Variety of protocols supported:
ACIX (ARC Cache Index), File, GridFTP, HTTP(S), LDAP, Rucio (ATLAS data management system), SRM, S3 Xrootd.... LFC, dcap, rfiio, ... (legacy)



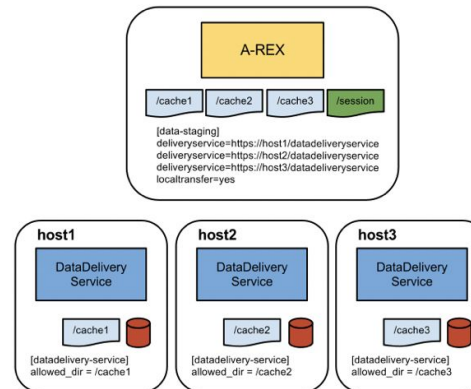
Datadelivery-service: scaling up data staging

- Data transfer capability can be scaled up by adding extra data staging hosts, [datadelivery-service]
- The master CE hosts delegates data transfer to the other hosts
tech description: https://wiki.nordugrid.org/wiki/Data_Staging/Multi-host

Multiple hosts with one large shared FS

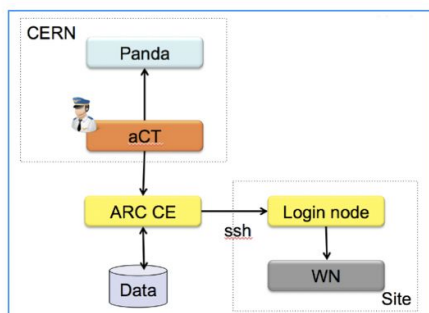


Multiple hosts each with own cache





WHY ARC? HPC friendly...., nevertheless fully EGI integrated



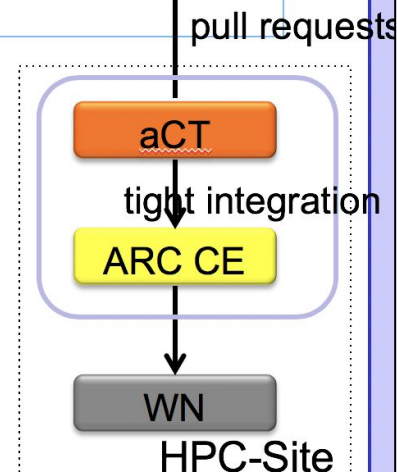
HPC sites are very restrictive allowing only ssh communication with outside world.

ARC jobs can still run on these sites e.g. with ARC-CE ssh-ing to site's login-node (remote lrms feature of ARC)

Some HPC sites are even more restrictive and don't want any unusual service with open network interfaces

Solution: pull jobs/data to the HPC site: using aCT@site

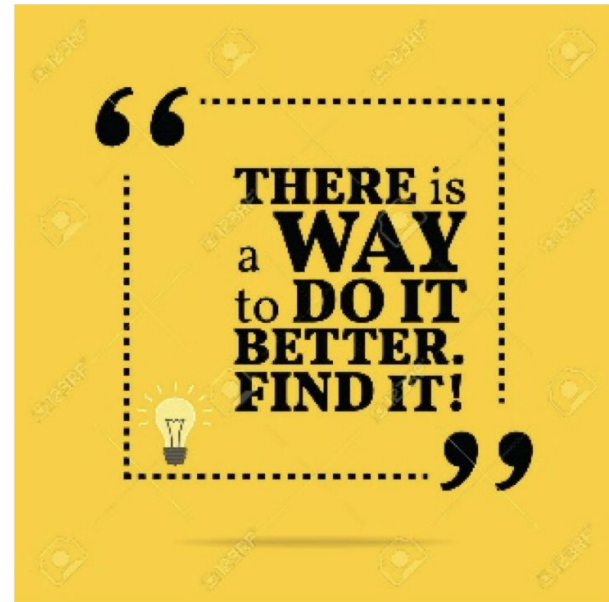
- a tightly integrated aCT and a networkless ARC CE on the HPC the site





Sometimes ARC is not the best option

- Communities not familiar with the “x509 grid security” world
- Native Condor sites



...then ARC took over...

- Ran ARC and CREAM in parallel during 2013, then CREAM retired
- Along the years RAL (Andrew Lahiff) contributed to the ARC code
- Patches relating to HTCondor tested first at RAL then committed to the NorduGrid repository and included in future releases
 - scan-condor-job tool, APEL accounting plugins
 - Patch for v4.1.0 - bug fix for memory requirements for multi-core jobs
 - Fixed bug in ARC condor submission script for multicore jobs – March 2015
- Overall happy with ARC + HTCondor


ARC at RAL - Current Status

- 4 (lately 5) ARC-CEs SL6, VMs on WMWare
- Recent upgrade from 5.0.5 to 5.4.3
- No longer in-house patches
- Helped by Florido Paganelli (many thanks!)
- Had to disable GLUE2 publishing - everyone happy
- Had to move from 'db3' to 'sqlite' – delegation issues
- Warned about potential problems with virtual I/O on 'controldir' (/var/spool/arc/jobstatus) - so far so good
- Each ARC @RAL up to 4-5k (peaks of 7k) running jobs

CE at RAL – Future Plans

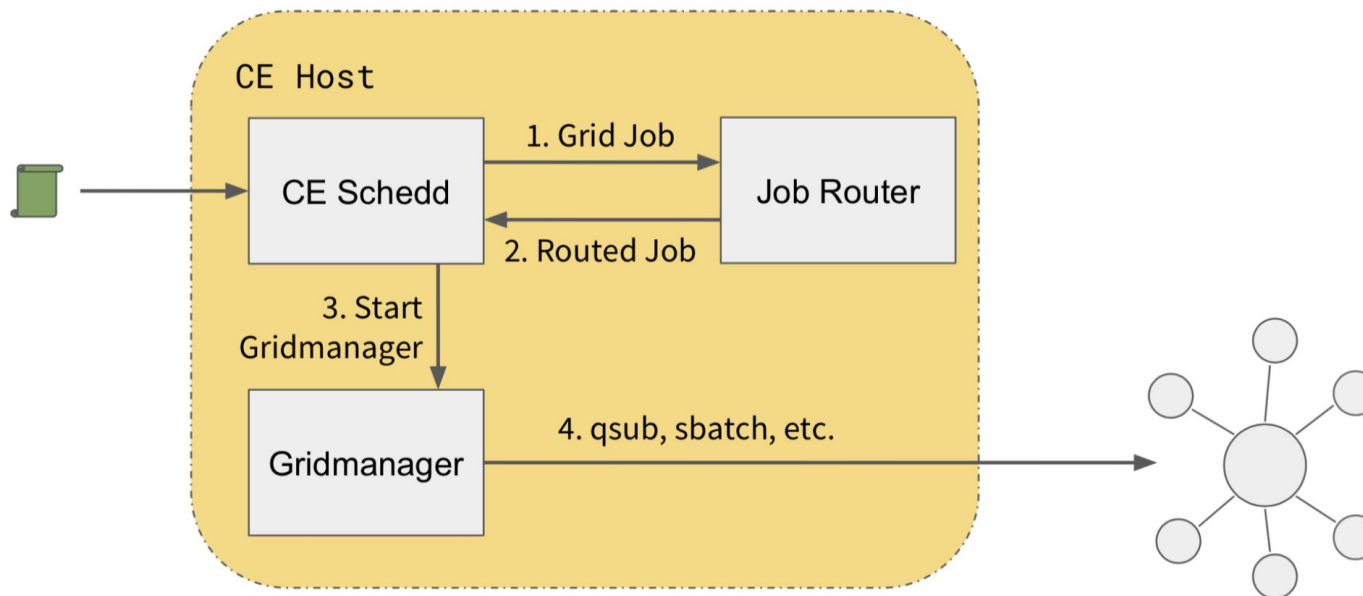
- Possible to have ARC6 in production this year, but...
- ...very likely to end up with a HTCondor environment
 - By end 2019, or Q1 2020
 - More natural configuration
 - APEL accounting on HTC-CE to be sorted out this year

HTCondor-CE MORGRIDGE INSTITUTE FOR RESEARCH

- If you look hard enough at the previous slide, you might realize that HTCondor itself fulfills all these needs! 
- HTCondor offers a remote API, has extensive auth{z,n} features, and can transform / submit jobs to an underlying batch system (another HTCondor, SLURM, PBS, etc).
- Effectively, we took a normal HTCondor submit host, enabled GSI configuration, enabled the built-in “job router” for transformations, and used the blahp to integrate with site batch systems.
 - Yes, that’s the same blahp used in CREAM!

CE!

Internals



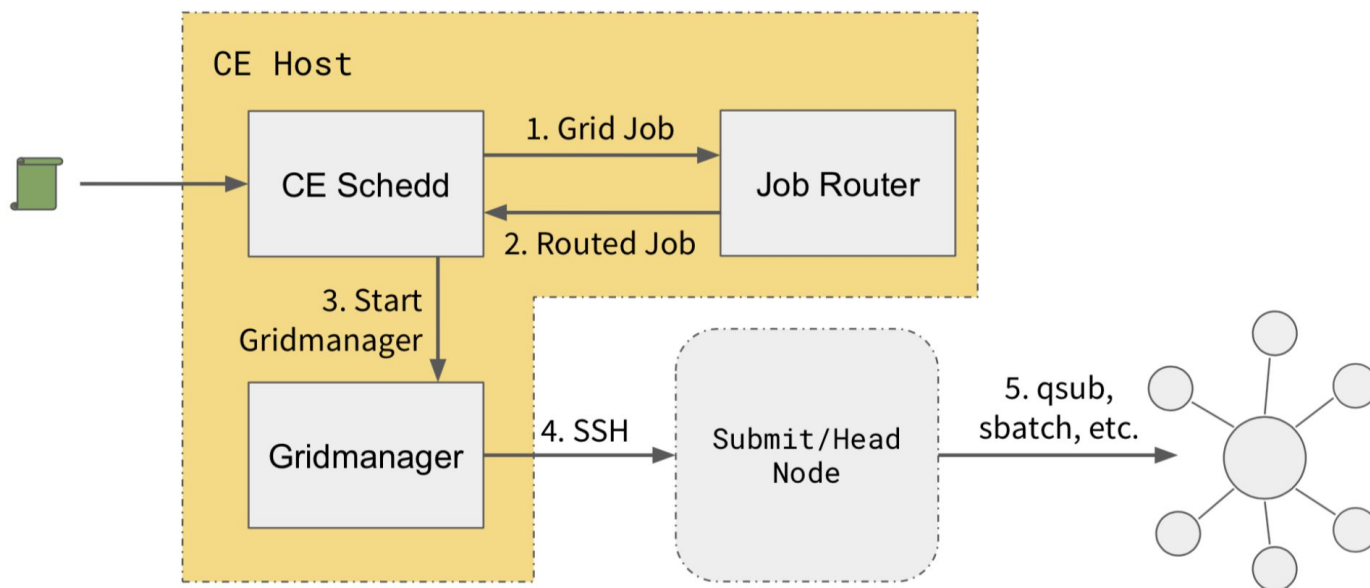
pstree output:

```
├─condor_master├─condor_collector│├─condor_schedd├─condor_gridmanager├─condor_shared_port└─condor_job_router└─blahp
```


Internals



MORGRIDGE
INSTITUTE FOR RESEARCH



Integrating the HTCondor-CE



MORGRIDGE
INSTITUTE FOR RESEARCH

- The HTCondor-CE started within the OSG but almost immediately became broader than that, especially with OSG's collaboration with CERN.
 - **Additional backends:** SLURM support was a noticeable gap in the original version in 2013!
 - **BDII integration:** HTCondor-CE came about around the time OSG retired the BDII; original implementation.
 - **Authorization plugin:** OSG uses LCMAPS exclusively; CERN used Argus. Both use the same plugin interface, but there's a lot of configuration details to figure out.
 - **Accounting:** Integrates cleanly with the OSG accounting system but the interfaces for APEL are very different.
- Work still needed on the last item - contributions welcome!

Future Work - Integration



- As the breadth of resources used by HEP widens, there's increased interest in using HPC resources and non-traditional resources:
 - There are a few places (e.g., Argonne National Lab) that have done a HTCondor-CE in front of their HPC center. Not clear how many will go this route.
 - At PIC, work is ongoing to provide more direct integration of payload jobs for Barcelona Supercomputing Center — would allow for pure HTCondor infrastructures to integrate directly without requiring a network service.
 - DODAS has been working on CE-less pilots - presentation later this session. I believe this work could be extended and widely adopted!

Building a Community



- What's important going forward is starting to build a larger community around the HTCondor-CE.
 - OSG is happy to help all of our partners, but doesn't have a clear mandate for some tasks (e.g., APEL integration is tough!).
 - Similarly, the HTCondor team doesn't have the effort to do the integration with every infrastructure out there.
- But we can help serve as a common watering hole where everyone can gather.
- As a community, we've done a poor job of coalescing around accounting, for example. I'm aware of about 4-5 sites that each did their own APEL accounting implementation.
 - Let's get this built-in. Thanks to Stephen who is starting to hammer this home!
- What other things can our team do to make the community feel welcome?

Want More?

**This has been a short overview -
We are planning a more in-depth session at the 2019
European HTCondor Week in Ispra, Italy.**



<https://indico.cern.ch/e/htcondor2019>

Save the Date! 24-27 September

Background

- **Liverpool T2 uses HTCondor-CE (since January.)**
- **It worked well, but it had no APEL parser to provide accounting.**
- **We tried HTCondor-CE PIC changes to APEL client parser (single input file). Functionally perfect. But not completely compatible.**
- **PIC changes conflicted with CREAM-CE + HTCondor, since it overrides existing HTCondor parser with a new version, and a new file format.**

Goals

- We extended APEL parsers to support HTCondor-CE in a way that is compatible with everything else, i.e. no config change required for non-HTCondor-CE sites who update APEL client.
- We directly support HTCondor-CE + HTCondor batch system, but architecture “supports” other backends, i.e. option to extend further for PBS, LSF, SGE, SLURM (and perhaps others in due course.)
- To do this, we preserved (to the largest extent) the existing data flow/file format conventions, giving minimal code changes (BTW: APEL parser is already well designed in this respect.)
- It is to be released with UMD as a standard way to link APEL with HTCondor-CE.

Remaining (vaguely related) issue

- This is a new general requirement for all sites using APEL client.
- Until now, APEL client obtains CE benchmark reference via BDII (Glue 1) and puts it in the accounting records to be sent.
- Problems: HTCondor-CE only gives Glue 2; and in any case the BDII is “soon” going away, it is said. Although the dates for this are not set.
- **Solution: ~ 20 line code change to allow admin to hard configure the scaling benchmark for CE in the APEL client. No query to BDII. Change still awaiting acceptance test/release.**
- Note: To “get the show on the road” a workaround is used for the time being - a one-off “static data” SQL insert done by sysadmin/build system. See documentation.

Ongoing work

- **Additional packaging work is ongoing with HTCondor-CE team to implement a couple of “nice-to-haves”.**
 - Make a RPM of the necessary interface scripts, within the HTCondor-CE release.
 - The existing HTCondor-CE puppet module may also be enhanced to install / configure the RPMs.
- **But system is already functionally complete and ready to use without this work. See links above to relevant set up documents.**

End user documentation

https://twiki.cern.ch/twiki/bin/view/LCG/HtCondorCeAccounting#Technical_setup

Scaling factor scheme

https://twiki.cern.ch/twiki/bin/view/LCG/HtCondorCeAccounting#Implement_scaling_factor

Tests

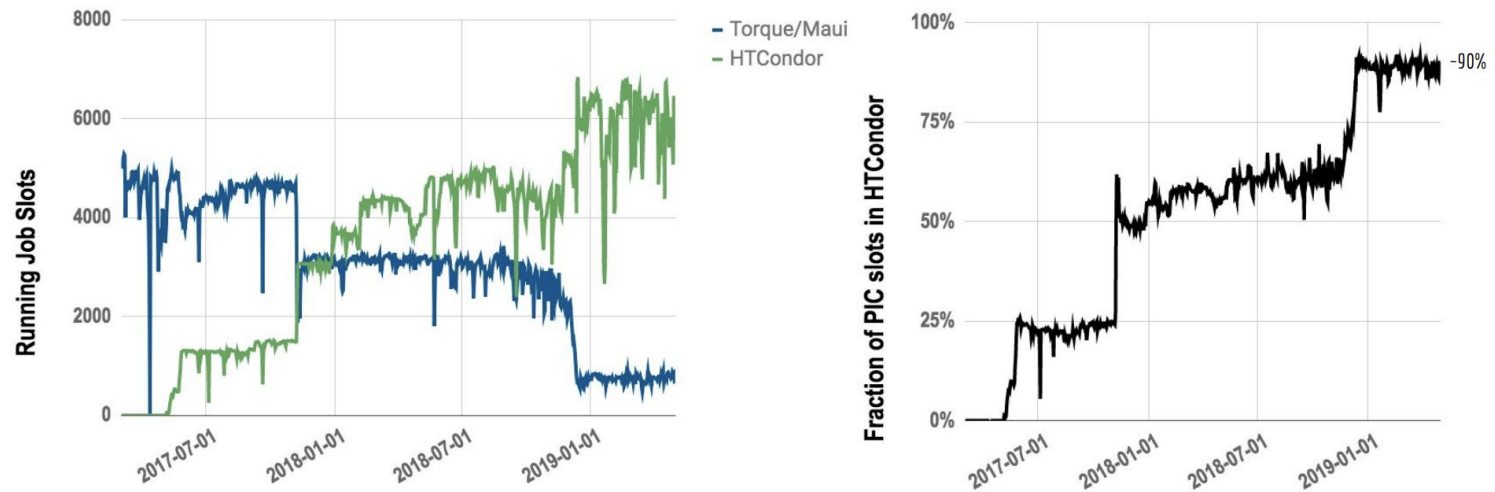
https://twiki.cern.ch/twiki/bin/view/LCG/HtCondorCeAccounting#Tests_on_a_HTCondor_CE

Design notes

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

HTCondor at PIC

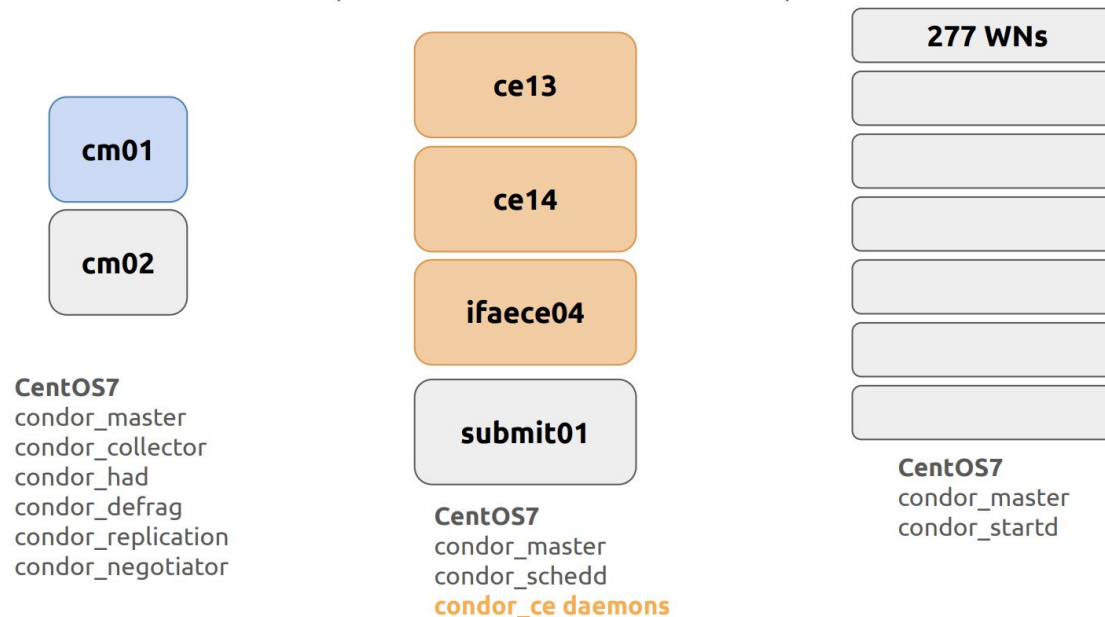
- At April 2019, the HTCondor pool at PIC consists in a pool of around 7300 CPU-cores and ~100 kHS06; Shrinking partition of ~11 kHS06 in Torque/Maui for small VO's... soon to be integrated into HTCondor + [Local submissions](#)





HTCondor at PIC

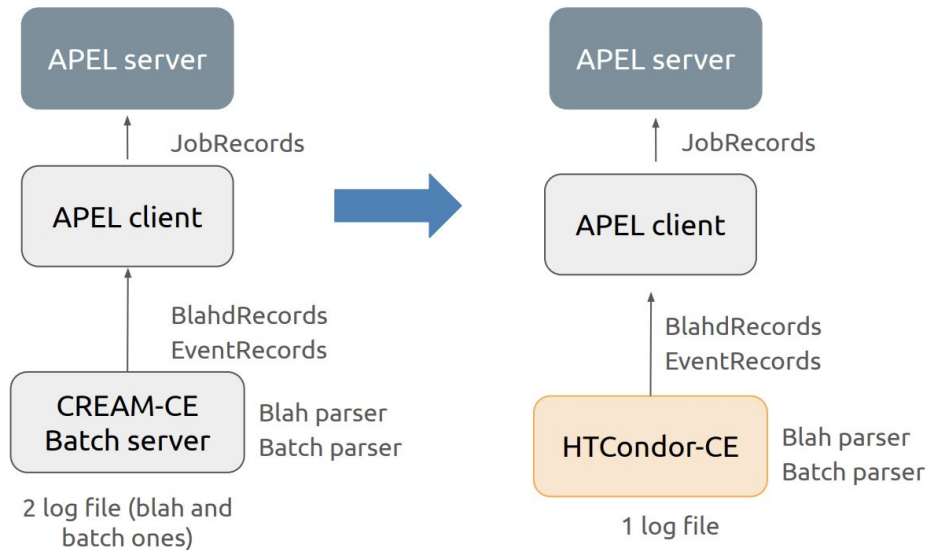
- At April 2019, the HTCondor pool at PIC consists in a pool of around 7300 CPU-cores and ~100kHS06
- Current version: Stable 8.8.1 (migrating to 8.8.2 soon)
- We have 2 Central Managers (CMs - HA), 277 Worker Nodes, 1 local schedd and 3 HTCondor-CEs (let's talk about them later)



HTCondor-CE at PIC

APEL integration

- It was the first stopper to put HTCondor-CEs in production
- We create our own APEL integration based in CREAM-CE/Torque scheme



HTCondor to exploit HPC resources



HTCondor for BSC



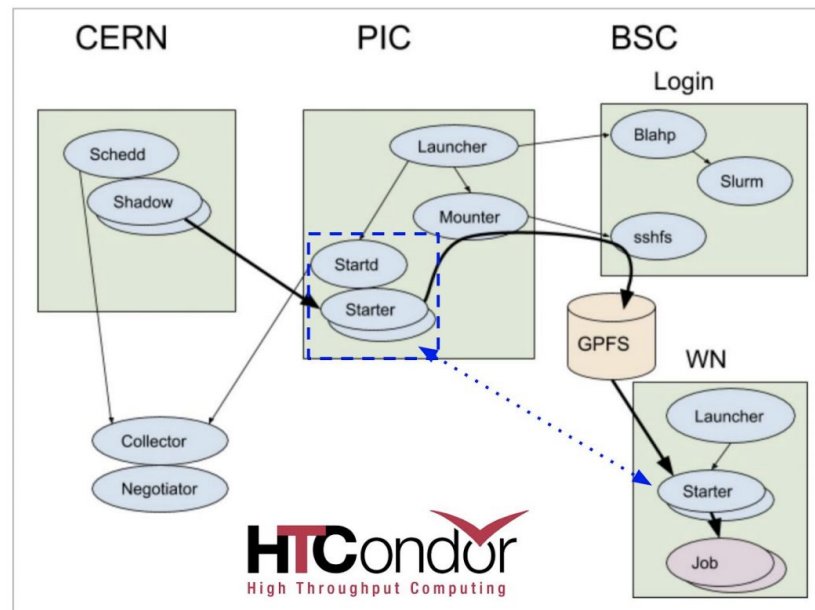
Main idea: a **bridge node** at PIC which allows access to **starter** processes running in the BSC nodes, mirroring **starter** at PIC

Input sandbox, status, etc, **passed as .tar files** via gpfs from bridge to WN

The **startd** process remains at PIC, where it can be accessed to **negotiate** by CMS/PIC Central Managers

From a functional perspective, **the node at BSC has joined the HTCondor pool at PIC**

See more details in the backup slides



PIC experience so far...

- HTCondor (and HTCondor-CE in extension) is a free software with a dynamic team working continuously to always improve it
- User support and extensive documentation available
- Once you get a stable configuration, HTCondor-CEs works fine without any remarkable issues
- Easy to play with HTCondor-CE as works with the same concepts as HTCondor
- HTCondor and HTCondor-CE flexibility allow us to test and implement new features: PIC-CIEMAT federation, connect to cloud resources (HelixNebula, AWS), interfacing to HPCs (collaboration PIC with HTCondor developers)
- Some issues and bugs, but HTCondor developers very proactive and helpful
 - Dual-stack issue with High Availability
 - condor_annex bug reported (to connect resources with AWS)
 -
- Lack of documentation supporting HTCondor-CE+EGI resources



Migrating from CREAM-CE/LSF to HTCondor-CE/HTCondor First tests including use of GPUs

Preprod cluster (HTC-CE 3.2.1, HTC 8.8.2)

- 3×HTC-CE on top of 1×CM/Collector, 15×WN, 16 slot each
- One more WN, with 2×K-40 GPUs (ongoing tests from VIRGO, ATLAS)
- Current latest stable versions (improved GPU support and monitoring)

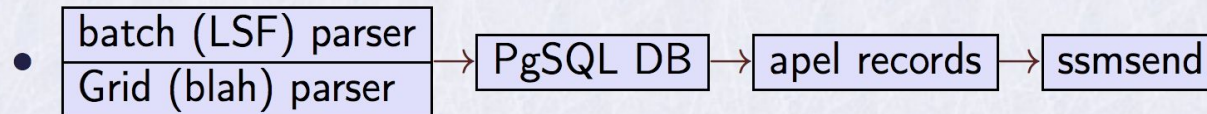
We plan to begin production activity starting from next days, in this order:

1. LHC VOs
2. Grid VOs using a WMS (i.e. Dirac)
3. Other VOs
4. Local submitters

Accounting

We are using our own custom accounting system for six years now, so we have been considering about adapting it.

Accounting with LSF



- We collect a few more data for internal use: job exit status, WN name (this is then mapped to HS06 of the node), requested resources, ...
- If we can collect the same data from HTC-CE, we can re-use the other components.

Conclusions

- HTC-CE works best with pilot jobs
- Currently, a few “gaps” in the documentation for non OSG people
- Can be seen as a “thin layer” on top of HTCondor:
- most of the desired behaviours are to be obtained by configuring HTCondor services, at CE side and/or batch side
- JobRouting is a very important mechanism to deal with when managing a working HTC-CE. Need some practice to get more confident with its configuration.

Experience in Liverpool when using/testing ARC-CE, HTCondor-CE and VAC

Which to use?

- If your batch system is supported by only one of ARC or HTCondor, end of discussion...
- Assuming your batch system is supported by both ARC and HTCondor, then it's game on ...
- If your batch system is not HTCondor, then you would have to wait for APEL accounting in HTCondor-CE, since only HTCondor-CE/HTCondor has fully tested APEL support at present. If this is your position, then ARC might be the best choice if you want to start soon, since it already has APEL accounting (via JURA) for all its supported batch systems.

16

Which to use?

- If you get this far, your batch system is HTCondor, because HTCondor-CE presently has APEL support only for that (or you can write your own APEL code, or afford to wait for someone else to do so.)
- So you'll have to explore some of the finer details to make the choice.
- If low-load and low use of file-space is your priority, then HTCondor-CE might be the better choice.
- If you'd like to use the ARC caching system to pre-place files, or some other ARC feature, then ARC is your choice.
- If you like a system where the same code is used for both the CE and the batch system (same dev, same processes), then HTCondor-CE is your choice.
- If you absolutely must have GLUE1, then only ARC definitely supports that.

17

EOSC-hub **DODAS in a nutshell**



DODAS: Dynamic On-Demand Analysis Service

A **Thematic Service** within EOSC-hub EU project service portfolio

- A open source deployment manager
 - Allows on-demand **creation and configuration of container based clusters for data processing with almost zero effort**
 - A cluster can be a standalone set of resources, a **WLCG Tier*-like** an extension of an existing center and more
 - BigData Analytics, Batch System as a Service, Distributed processing framework for ML
- Support for **hybrid clouds deployment**
- **High level of automation and self-healing**
- Oriented to the **ZeroOps model**
- Supports **communities-tailored (user-tailored) applications and software for data processing**
- Flexible **Authentication and Authorization** model
- Based on “industry standards” to minimise code development and maintenance



DODAS: main motivations



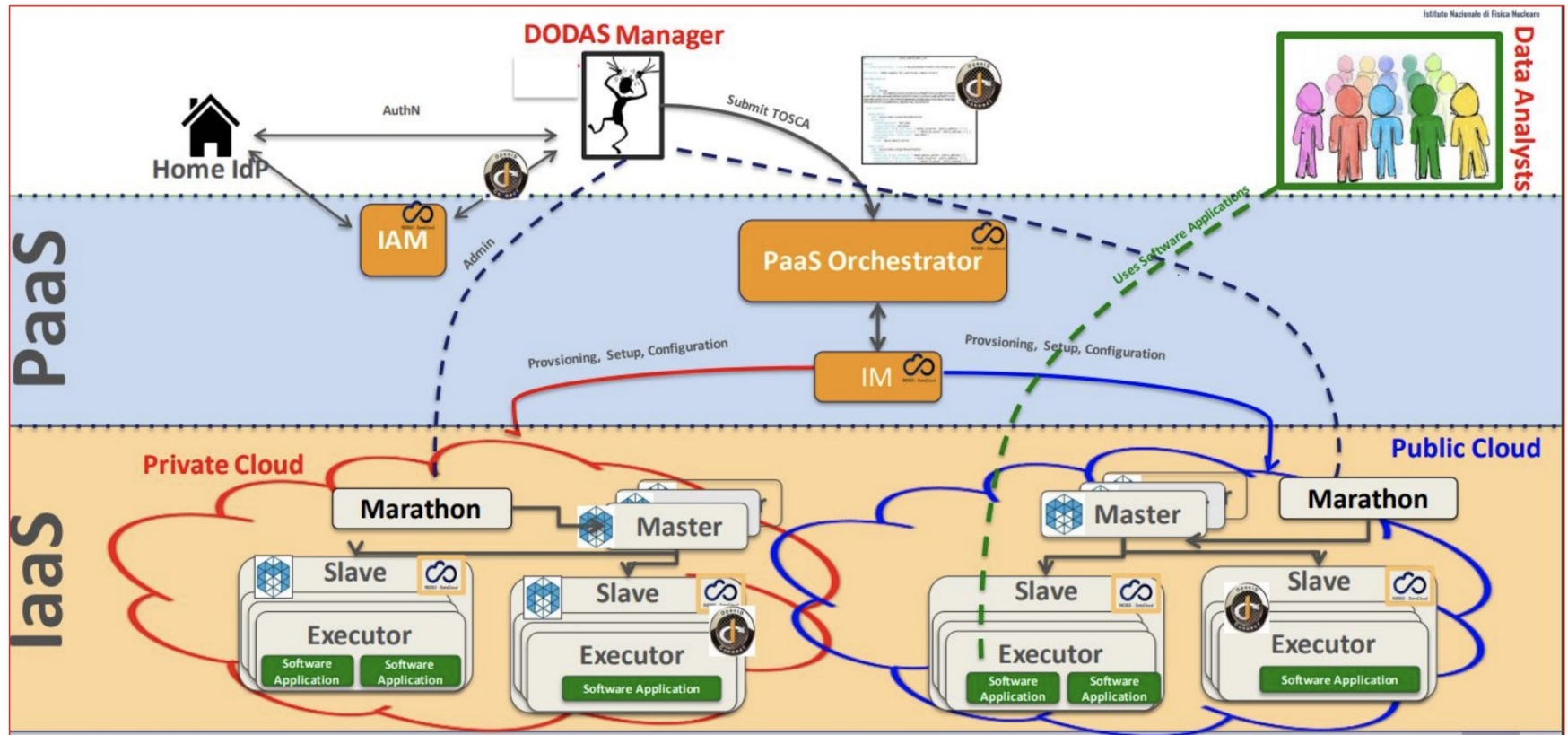
- To provide a “ZeroOps solution” to exploit **opportunistic computing**
 - Optimising the costs/benefits to exploit resources not necessarily/permanently dedicated to a specific experiment
- To allow **sites with limited (or without) effort for experiment specific support** to provide computing to the collaboration
 - Tier3-like, campus facilities and general purposes farming
- To ease site overflow and/or **elastic site extension**
 - To absorb peaks of requests at Tiers1/2 by using external providers such as public clouds
 - To accommodate workflows with special requirements
- Moreover DODAS can be a technology enabler to build:
 - Regional/national level computing infrastructure
 - Prototype of quasi interactive analysis facility to exploit different model for physics analysis
 - integrated into the data and workflow management of the experiments.

No CE solution: DODAS

Daniele Spiga



Architectural overview





Implementing Vacuum with DODAS



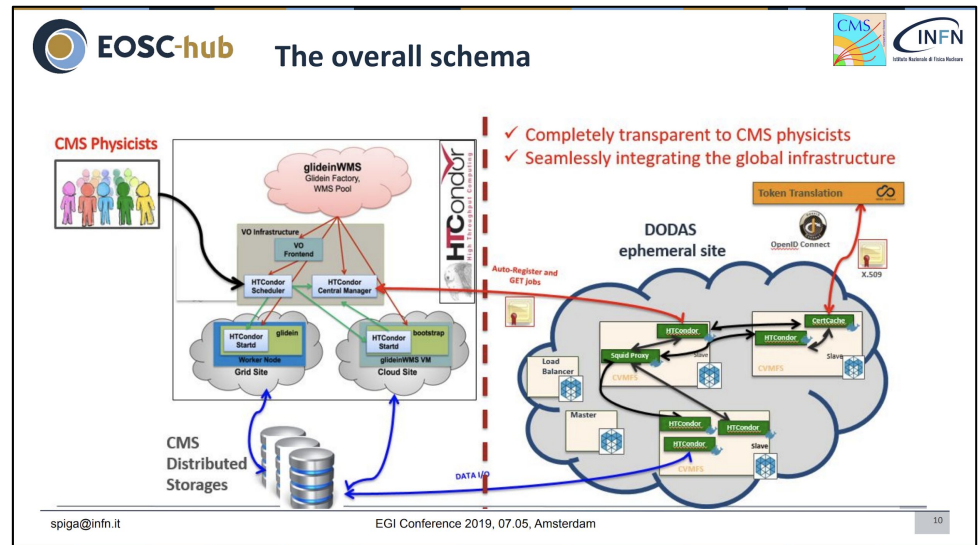
- DODAS relies on vacuum approach to provide WLCG-like resources and it is responsible to automate:
 - **Bare-hosts (e.g. VMs) instantiation** based on user requirements, defined at TOSCA level
 - **Virtual hardware can be scaled** up/down (elasticity)
 - **Services and software configurations** at host levels (e.g. CVMFS, docker engine etc)
 - **Container orchestrator deployment** (e.g. K8s, Mesos/Marathon), and this is in form of dockers
 - **Deployment and execution of services/microservices** (e.g. Worker Nodes, squid proxy, x509 cache) over orchestrators
 - **this is how worker nodes are “spontaneously produced” and scaled up/down**
 - In addition DODAS provides a JWT based ecosystem for **authentication and authorization: INDIGO-IAM**
- Current incarnation is based on HTCondor as a mean to manage (aka overlay) distributed worker nodes (startd/glideins)
 - Does not deploy Computing Element (CE) but it could be added (example of modularity)
 - **DODAS Vacuum system is integrated in the CMS Computing infrastructure aka HTCondor Global pool (see next slides)**



But not only HTCondor...



- DODAS is fully integrated into the CMS computing model to create lightweight ephemeral WLCG-Tier on demand
 - Generated sites automatically deploy and manage
 - HTCondor services
 - Squids
 - ProxyCaches
 - CVMFS
 - ...





Not only compute... data ingestion



- The very strategy to the automation has been applied to a Xcache Docker container has been setup to allow an easy deployment

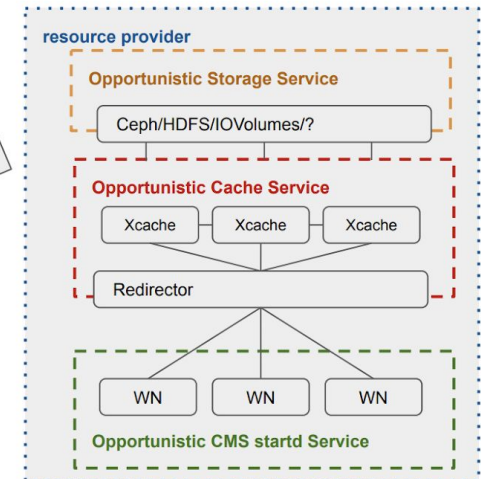
- passing a complete xcache config file
- or setting caching parameter as arguments/env
- healthcheck call implemented

A Docker Compose configuration file is available to **orchestrate the deployment of a local test instance**. The stack contains a test remote server, a cache instance and cache redirector ([preliminary docs here](#))

- A variety of recipe for **orchestration tools** have been created:
 - **docker swarm, k8s and marathon services** (redirector+caches)
 - config and scale services with compose-like recipes

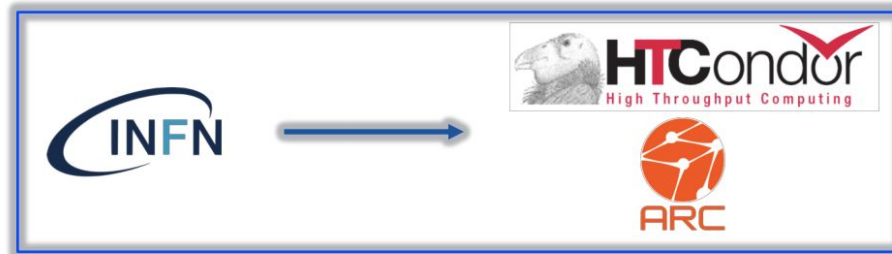
- **Work in progress**

- Preparing to test the new HTCondor token support, together with INDIGO-IAM authZ service



Use case

- A first natural use case for the framework is migration from CREAM-CE.



- Simplify **switching to HtCondorCE/HTCondor batch** powered site

SIMPLE Framework

- Package sensible default configurations for grid services into **Docker containers**.
- Enable **hassle-free deployment** of these containers **across the site** using popular technologies under the hood:
 - container orchestration tools (**Docker Swarm/ Kubernetes**)
 - configuration management tools (**Puppet/Ansible**)



SIMPLE Framework: Example

Config Master(CM)

simple-cm



Install puppetserver, puppet

Lightweight Component(LC)

simple-lc01

simple-lc02

simple-lc03

simple-lc04



Install puppet and complete certificate signing process by the puppet master.

Then, install `simple_grid_puppet_module` on all nodes. For instance,

```
[root@simple-cm ~]# puppet module install maany-simple_grid
```

SIMPLE Framework: Example

- Write a **site-level-configuration.yaml** File:

declare variables

```
1 global_variables:
2   - &simple_cm_ip_address 188.184.91.176
3   - &simple_cm_fqdn simple-cm.cern.ch
4   - &simple_lc01_ip_address 188.184.88.69
5   - &simple_lc01_fqdn simple-lc01.cern.ch
6   - &simple_lc02_ip_address 188.184.83.48
7   - &simple_lc02_fqdn simple-lc02.cern.ch
8   - &simple_lc03_ip_address 188.185.76.205
9   - &simple_lc03_fqdn simple-lc03.cern.ch
10  - &simple_lc04_ip_address 188.184.86.181
11  - &simple_lc04_fqdn simple-lc04.cern.ch
12
```

SIMPLE Framework: Example

Describe the grid
services should be
deployed at the site

```
32  lightweight_components:  
33  - name: simple-htcondor-ce  
34    type: compute_element  
35    repository_url: "https://github.com/maany/simple" }  
36    repository_revision: "master" }  
37    execution_id: 0  
38    deploy:  
39      - node: *simple_lc01_fqdn  
40        container_count: 1  
41    config:  
42      - reserve_swap: 0  
43    supplemental_config:  
44      - {condor_knob}:{value}  
45
```

Simple's
repository for
HTCondorCE

SIMPLE Framework: Example

- Summing up:
 - Install puppet and simple grid puppet module on all nodes.
 - Write a **site-level-config-file.yaml**.
 - Execute the framework.

```
[root@simple-cm ~]# puppet agent -t
```

SIMPLE Framework: Example

- Summing up:
 - Install puppet and simple grid puppet module on all nodes.
 - Write a **site-level-config-file.yaml**.
 - Execute the framework.

```
[root@simple-cm ~]# puppet agent -t
```

et voilà

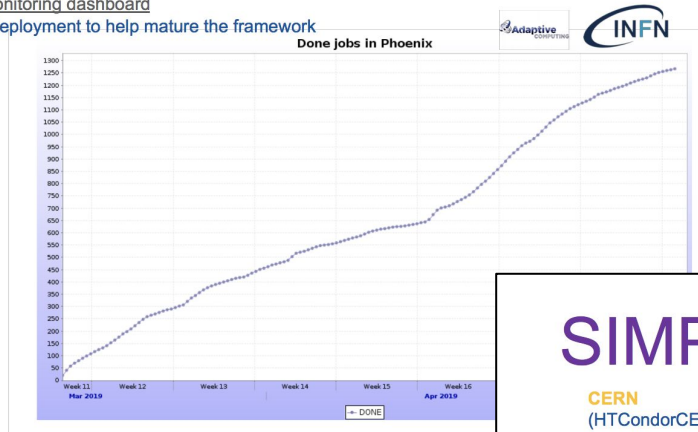
SIMPLE Framework: Deployments

Centro Brasileiro de Pesquisas Físicas (CBPF, Tier-2 in Brazil)

Cremon-CE, PBS batch system and workers

[Monalisa monitoring dashboard](#)

*small test deployment to help mature the framework



7/5/19

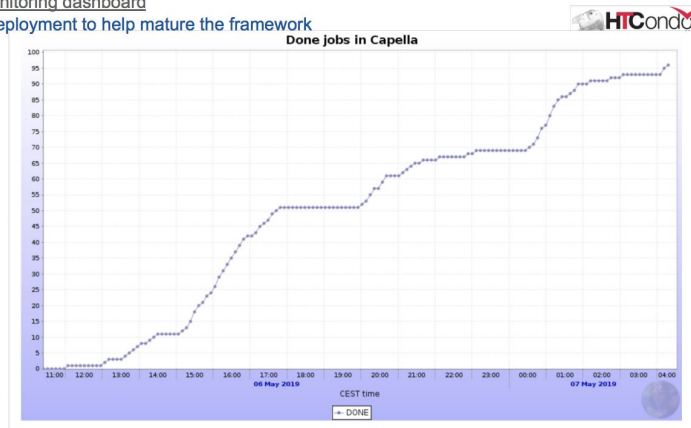
SIMPLE Framework: Deployments

CERN

(HTCondorCE, HTCondor batch system and workers)

[Monalisa monitoring dashboard](#)

*small test deployment to help mature the framework

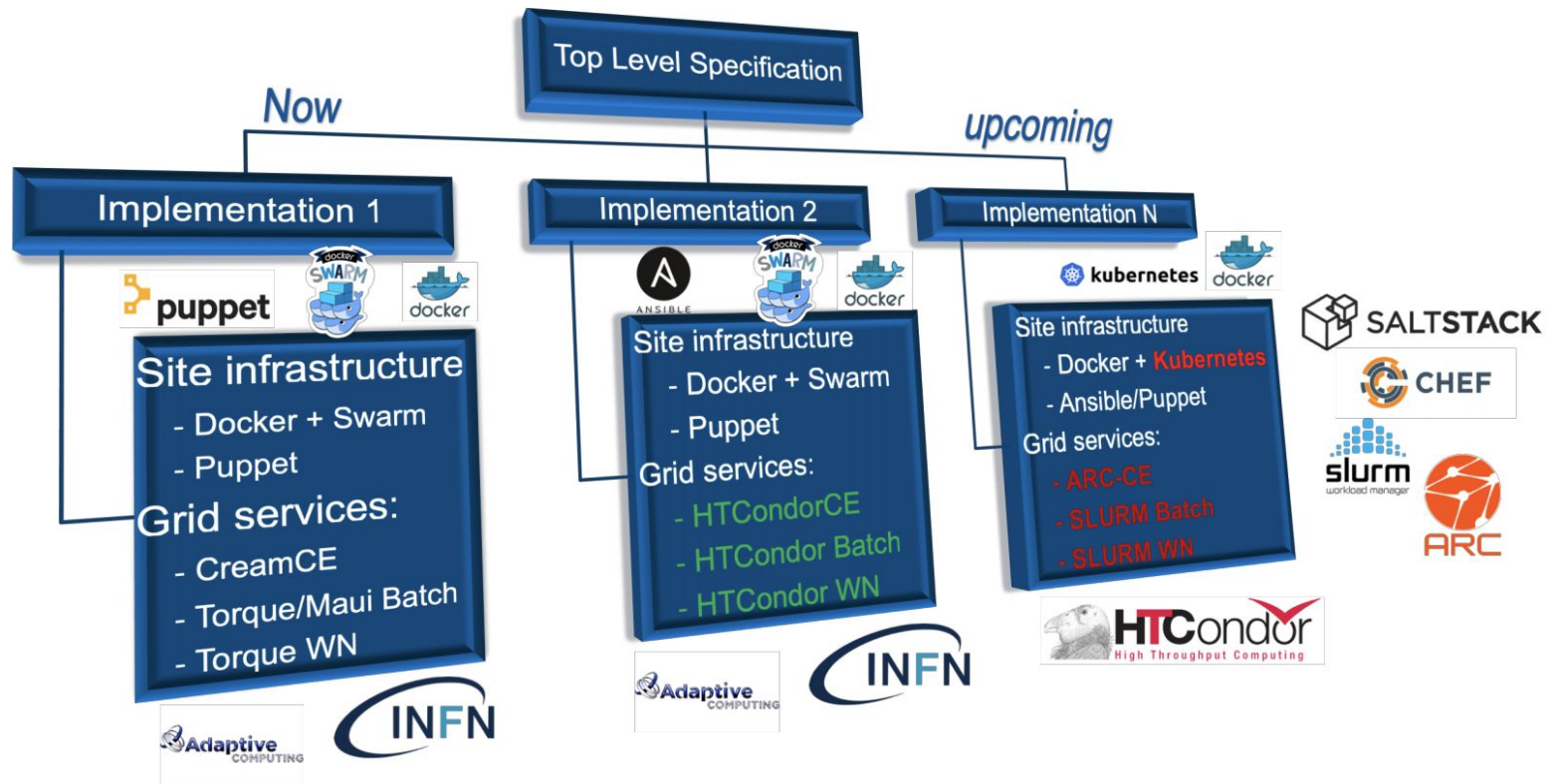


7/5/19

EGI Conference 2019

6

SIMPLE – Project Structure



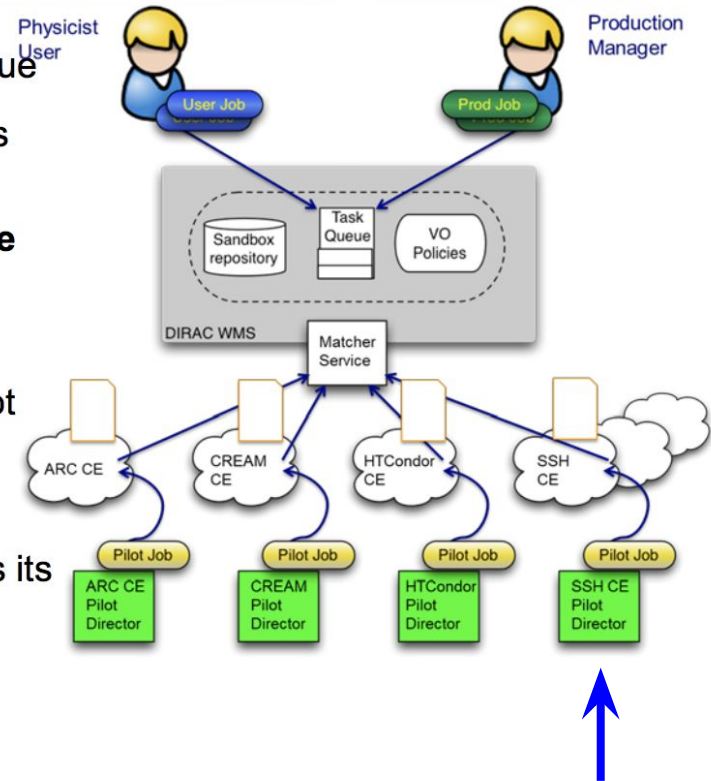
SIMPLE Framework

- The SIMPLE HT-Condor repositories should be ready for use in production in next few weeks. (Accounting/ BDII/Default configurations)
- Join the mailing list to get notified:
 - E-Groups : <http://cern.ch/go/Hz7S>
 - Google Group: <http://cern.ch/go/l9wZ>



Job scheduling

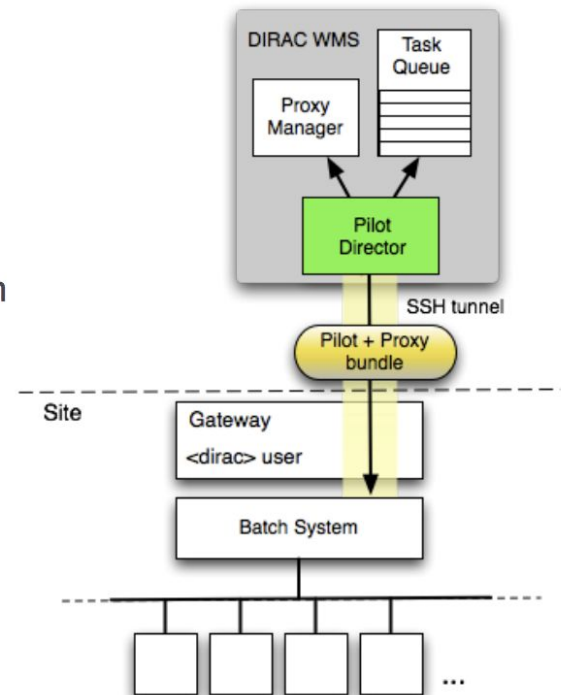
- **Users submit jobs**, which are stacked in the DIRAC Task Queue
- DIRAC Pilot Directors submit **Pilot jobs** to computing resources
- After the start, Pilots check the execution environment and requests a job from the **Matcher service** providing the **resource description**
 - OS, capacity, disk space, software, etc
- The **Matcher service** selects the appropriate user job for the pilot
 - Matching based on (i) the resources description and (ii) job requirements
- The user job description is delivered to the pilot, which prepares its execution environment and executes the user application
- At the end, the pilot uploads the results and output data





Standalone computing clusters

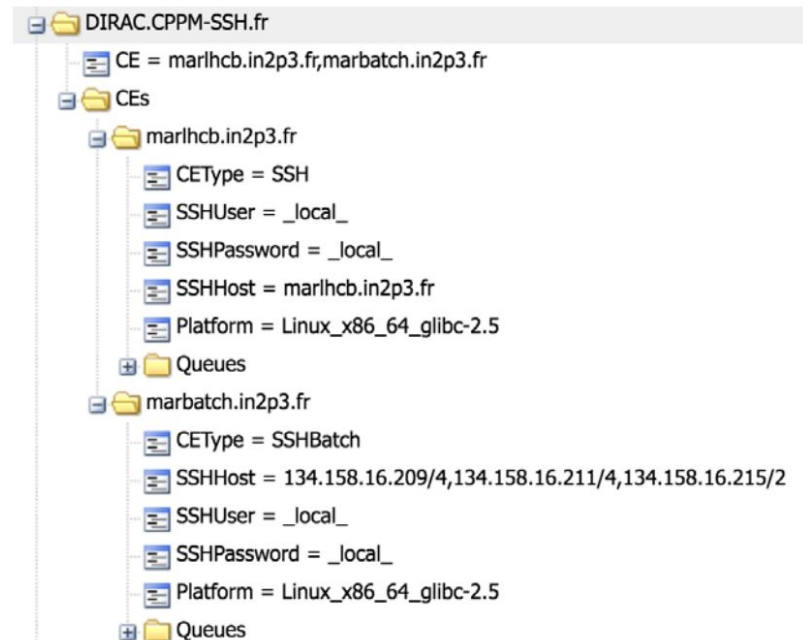
- **Off-site Pilot Director**
 - Site must only define a dedicated local user account
 - The payload submission through an SSH tunnel
- **The site can be**
 - Single computer or several computers without any batch system
 - Computing cluster with a batch system
- **Pilots are sent as an executable self-extracting archive with the pilot proxy bundled in**
- **The user payload is executed with the owner credentials**
 - No security compromises with respect to external services





SSH CE examples

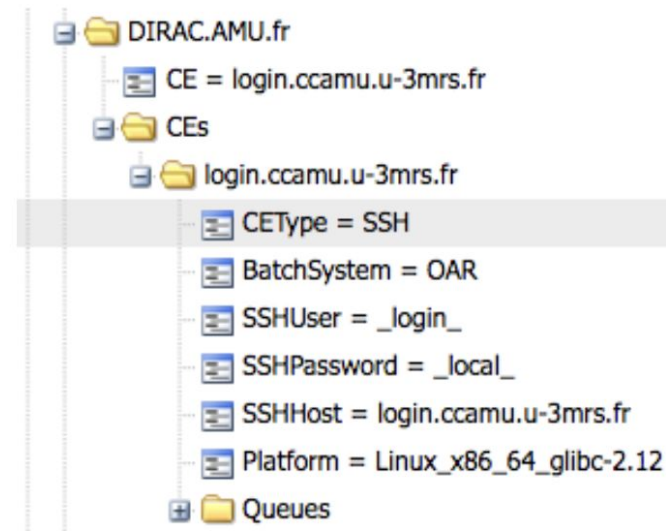
- SSH CE simplest case
 - One host with one job slot
- SSHBatch CE
 - Several hosts form a CE
 - Same SSH login details
 - Number of job slots per host can be specified





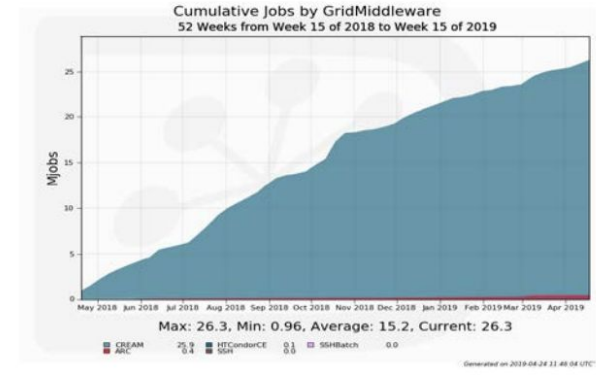
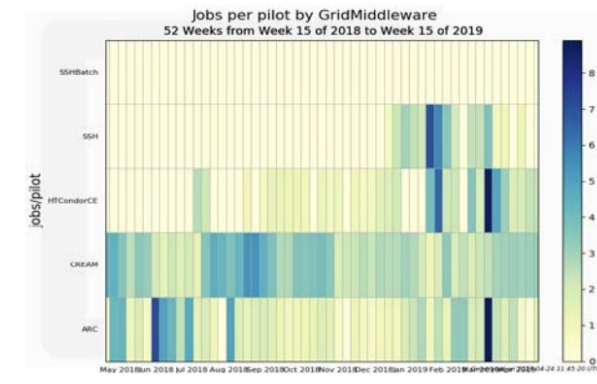
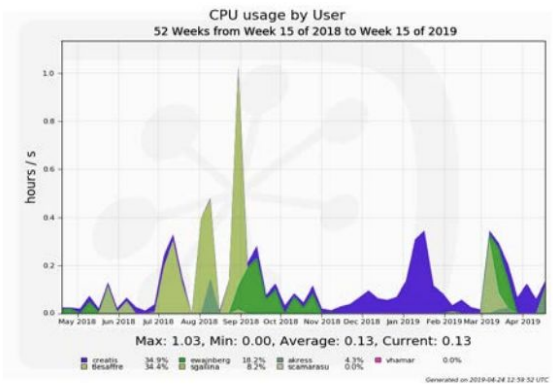
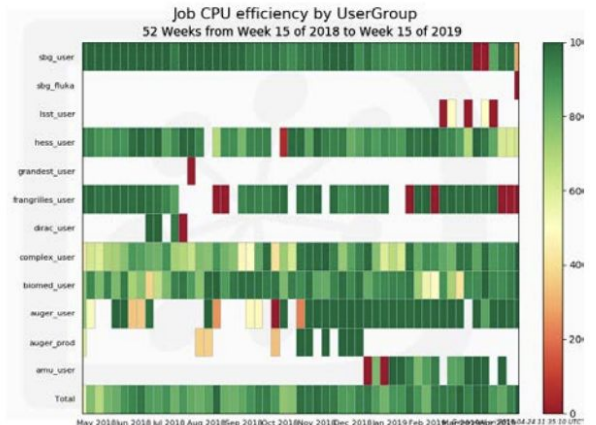
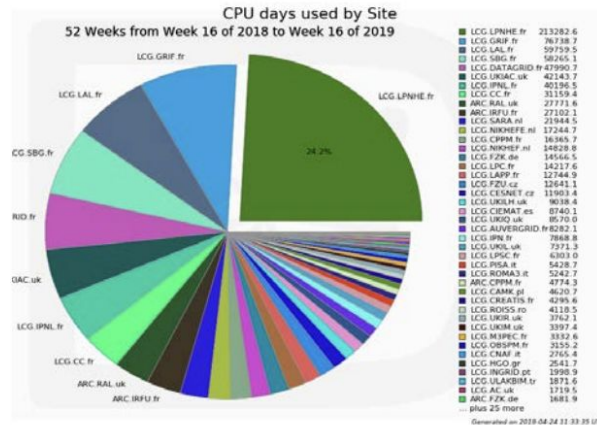
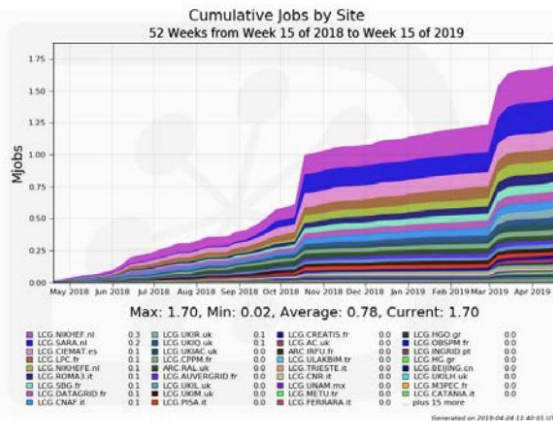
SSH CE with a batch system

- SSH login to the cluster interactive host
 - Copy several tools, e.g. BatchSystem plugin at the first time
- Submit pilots to the local cluster using a relevant BatchSystem plugin
 - Condor, GE, LSF, Torque (HTC)
 - SLURM, OAR (HPC)
- Site admins only need to allow ssh connexion
- Transparent for DIRAC end users





Accounting



PANEL → Open Discussion

Q: Fairshare in DODAS? How to handle this?

A: DODAS can easily be interconnected to a Batch System, so the pilots are created by the BS, preserving fairshare, etc....

Q: How do we go forward, since we have 18 months time for the transition?

A: Working Group should be created to handle the transition and help the sites during the transition. Communication channels need to be created and lines of action will be broadcasted

Q: EGI perspective on sites with no CE at all?

A: Security via ssh? It might be a showstopper for some sites, but of course this might be an option. It might happen that this is adopted by some sites - follow-up

Q: Does any VO in EGI expressed any concern for CREAM-CE migration?

A: All of the VOs should (already) know that the migration is happening, which are the options available, etc... We need documentation and spreading more info

Q: Which are the standards as of now? This might help sites/VOs...

A: Uniform solutions (dream) did never happened in the Grid. More common components are being used elsewhere nowadays. We will need to deal with the diversity. Even, standards have also timelines and death-dates... This is IT!

Q: Which is the plan for VOs that does not have developers? Support?

A: Tools and documentation will be available. There is a DIRAC catch up instance for these VOs, which is a generic solution. Diverse knowledge is difficult to manage, it might need funding agencies to be (more) committed

Note: Globus end-of-life was managed through a dedicated list... maybe a new list can be created to share experiences, documents, etc... for the CREAM-CE migration

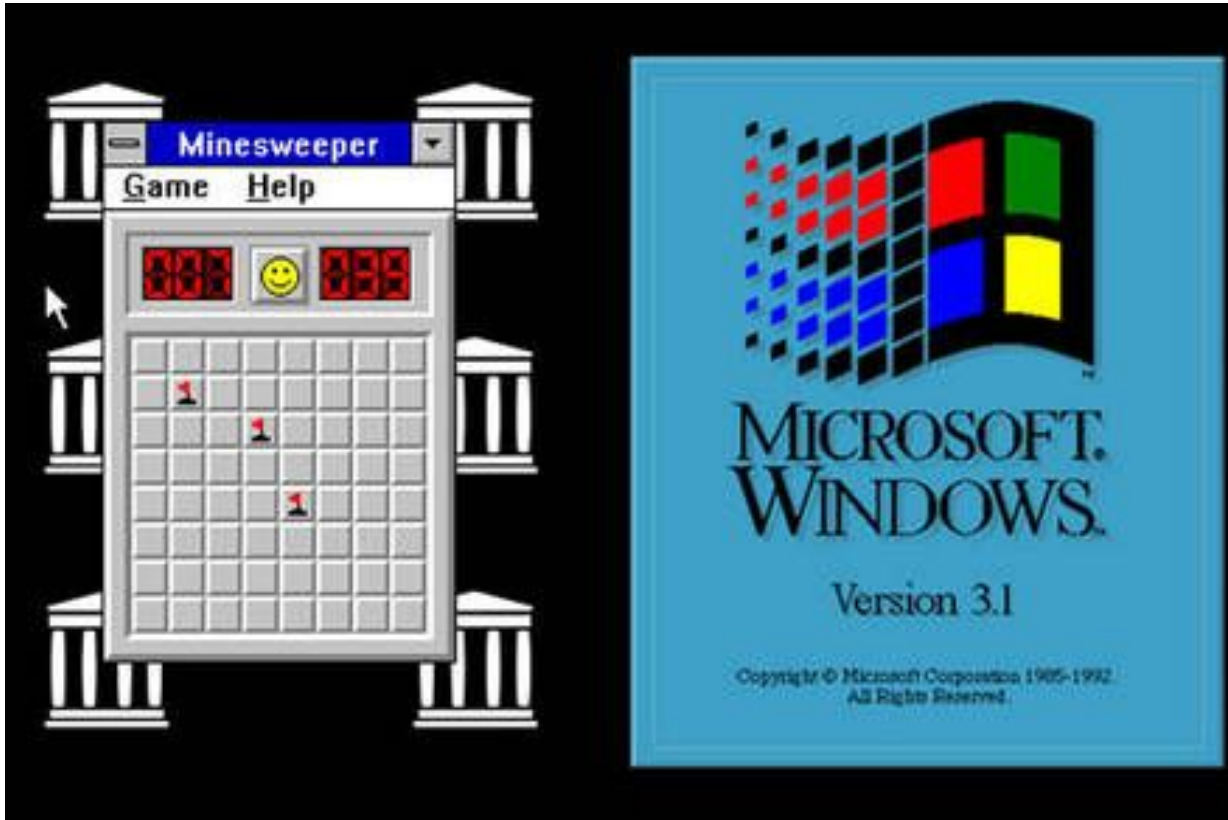
Conclusions/Outlook

- CREAM-CE decommissioning on-going... 18 months ahead
- Good discussions on available solutions and experiences:
 - ARC-CE, HTCondor-CE
 - No-CE solutions: DIRAC SSH CE, DODAS, VAC
 - Tools to ease deployments: SIMPLE, SLATE
- HTCondor-CE is the natural choice for sites using or migrating their BS to HTCondor
- Which solution to adopt depends on your VOs requirements, or the type of site to deploy or the resources you want to exploit... It's not just the 'entry point of jobs', a CE might not be needed at all in some cases:
 - Do you want a CE that handles data Caches?
 - Do you want to exploit opportunistic resources easily?
 - Do you want to launch pilots that connect to central VO WMs?
 - Do you need to exploit and integrate HPC or Clouds into your site?
 - Do you want to offer a lightweight site to a particular VO?
 - Do you want to reduce deployment efforts, it might be possible

Conclusions/Outlook

- It was a workshop → **Actions:**
 - (more) documentation and procedures should be made available
 - Collaborative effort to bring all of the necessary elements that can help VOs and sites, for the different options
 - Tightly coupled coordination effort from EGI and WLCG Ops
 - Creation of a Working Group to manage the transition
 - Communication channel to spread information → mailing list
 - Expecting a diversity of solutions to be adopted elsewhere
 - Foreseen follow-up discussions/check-points during the period

Conclusions/Outlook



Questions?