HTCondor deployment at GRIF

Vamvakopoulos Emmanouil
On behalf of Technical Committee at GRIF

A.Bailly-Reyre, C.Cavet, S.Ferry, A.Garcia, M.Jouvin,, M.Mellin, V.Mendoza, G.Philippon,

A.Ramparison, E.Vamvakopoulos

HTCONDOR Workshop 20-22 Sep 2023

IJCLAB

ORSAY-PARIS

GRIF is a distributed site made of four (4) different subsites, in different locations of the Paris region.

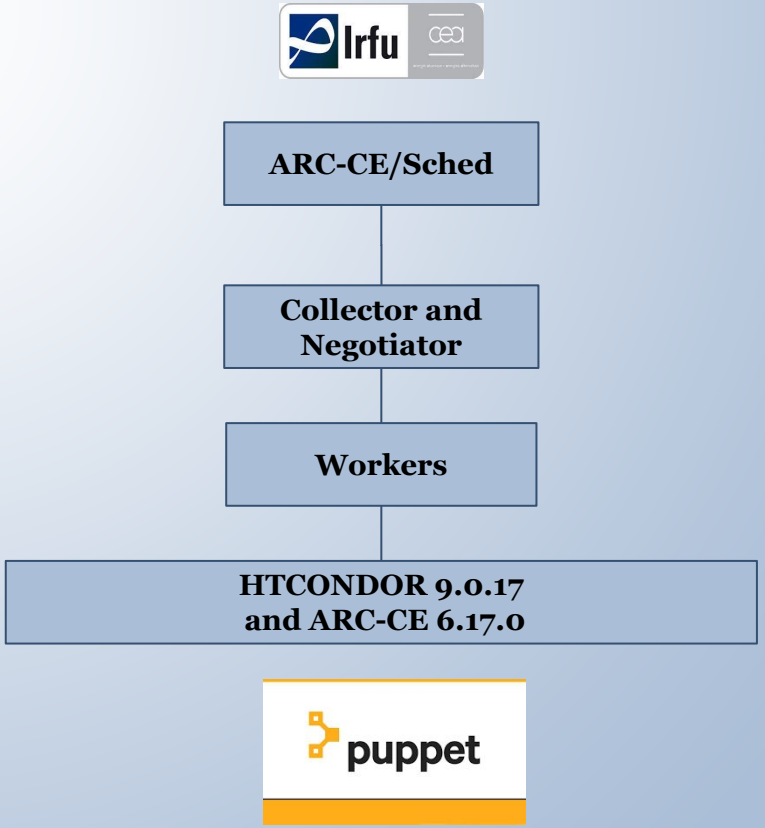
- **IRFU, LLR** and **IJCLAB** are interconnected with 100Gb link, LPHNE is interconnect with 20Gb link.
- The worst network latency between the subsites is within 2-4 msec
- Two independent condor pool at IRFU and LPHNE and one common at LLR and IJCLAB
- Total Compute Capacity 15000 cores (logical)
- Supports the four (4) WLCG VOs: **Alice, Atlas, CMS and Lhcb** and several EGI and OSG VOs
- Hardware configuration is mainly 2-way compute servers with 10Gbit nics and multicore CPU and 2GB physical memory per core and ~20GB of scratch space per core.
- Quite heterogeneous hardware layout (e.g. number of core per CPU) between the sites and servers' generations

Historical milestones of GRIF's deployment and future plans

- **Early phase 2014-2020**
  - Deploy HTcondor on GRIF sites as a replacement of Torque/Maui
  - A common pool behind two cream-ce at LLR/IJCLAB
  - A independent pool behind a cream-ce at LPHNE
  - A independent pool behind a ARC-CE at IRFU CEA
- **Middle phase to 2020-2023 -** Introduction of HTCondor-CE gateways
  - Introduction of WLCG tokens - update to htcondor version 9.x
  - Decommission of CREAM-CEs
- **Current Phase - 2023 and later -** Update to htcondor version 10.x
  - Abardon GSI authentication method and switch to pure token
  - IJCLab plans to add a container universe to allow local users (non grid) to submit jobs on its OpenStack cloud
  - Future intention to incorporate LPNHE worker on common pool

HTCondor/ARC-CE at IRFU/CEA

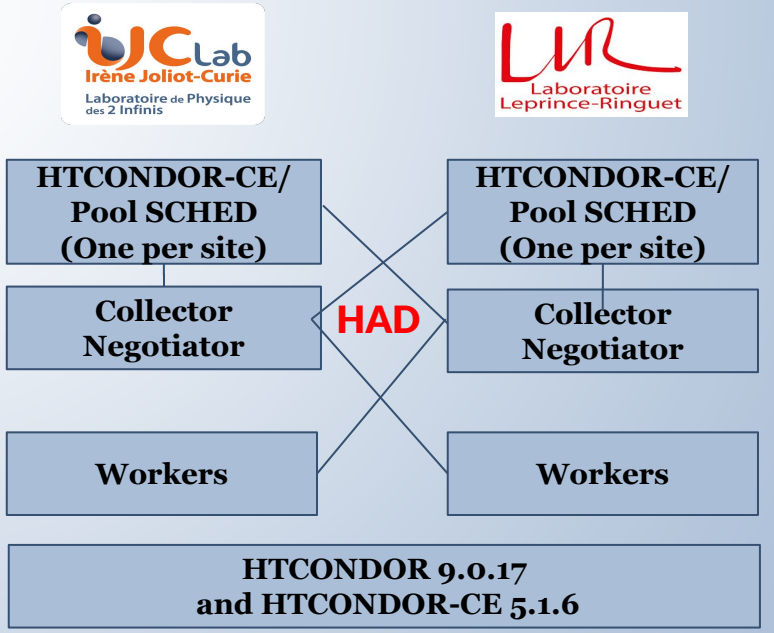
- ARC provides batch plugins for condor
  - Job submission and job control,
  - translate grid requirements into HTC requirements;
- We modified the submit plugin in order tomap VO/FQAN and requirements to accounting groups; format vo names;correct grid requirements (e.g. required RAM)
- Limits and policies enforced via HTCondor
- cgroups integration: job req. -> cgroup limits;SYSTEM_PERIODIC HOLD/REMOVE
  - controls disk usage, WCT, queue time etc.
- BDII provider integrated in ARC
- APEL: we use the std tool apel-parsers slightly modified. FR sites have “local” publication.



Distributed common pool LLR/IJCLAB

- Distributed pool
- HTcondor uses only 1 port (“shared_port”) which makes WAN clusters very easy to build
- High Availability for Negotiator and Collector straightforward with “had” condor service
- one (1) HTCondor-CE for each sub-site

Each HTCondor-CE gives access to all compute resources



quattor

New mapping syntax
(rule order matters)

10-gsi.conf

- *GSI ^\/O=GRID-FR\/C=FR\/O=CNRS\/OU=LAL\/CN=grid02.lal.in2p3.fr$/ grid02.lal.in2p3.fr@daemon.htcondor.org*

10-sci.conf

- *SCITOKENS /^https:\/\/atlas-auth\.web\.cern\.ch\/,**5c5d2a4d-9177-3efa-912f-1b4e5c9fb660**$/ **atls***

50-gsi-callout.conf

- *GSI /(.*)/ GSS_ASSIST_GRIDMAP*

/usr/share/condor-ce/mapfiles.d/50-common-default.conf

- *SSL /\/CN=([-.A-Za-z0-9\/=]+)/ \1@unmapped.htcondor.org*
- *CLAIMTOBE /.*/ anonymous@claimtobe*
- *FS /^(root|condor)$/ \1@daemon.htcondor.org*
- *FS /(.*)/ \1*

Jobrouter and hook transformation

- AccountingGroup (up to four level), expr for concurrencyLimit, WNTag: restrict jobs to tagged node PolicyGroup: set of policies (e.g. WCT) are setup via via job router hook features (job transformation) base on external script hook.py (and the predefined matching rules)
- The accounting group (and other attributes) can be set against against a collection of regular exp-based substitution rules for x509UserProxyVOName, x509UserProxyFirstFQAN, X509userproxysubject, queue
- When we use wlcg bear token and not use x509 proxy certificate the above attributes do not exist the classadd of the job and the hook fail (e.g. for some jobs).
- We need to modify the hook.py and add the manipulation of AuthTokenGroups, AuthTokenId AuthTokenIssuer, AuthTokenScopes, AuthTokenSubject in order to define the correct accounting group

Hook.py modification

```
if 'x509UserProxyVOName' in CLASSADS:
 CLASSADS['x509UserProxyVOName_Fmt'] = CLASSADS['x509UserProxyVOName'].replace('.', '_').replace('-', '_')
 CLASSADS['x509userproxysubject_Fmt'] = CLASSADS['x509userproxysubject'].replace(',', ' ')

elif 'AuthTokenIssuer' in CLASSADS:
 CLASSADS['x509userproxysubject'] = CLASSADS['x509userproxysubject_Fmt'] = CLASSADS['AuthTokenSubject']
 CLASSADS['x509UserProxyVOName_Fmt'] = get_auth_group(CLASSADS['AuthTokenIssuer'])
 CLASSADS['x509UserProxyVOName'] = get_auth_group(CLASSADS['AuthTokenIssuer'])
 CLASSADS['x509UserProxyFirstFQAN']  = '"/'+get_auth_group(CLASSADS['AuthTokenIssuer'])+"/Role=null"+'"'
 #print CLASSADS['x509userproxysubject'] ,CLASSADS['x509UserProxyVOName_Fmt'],CLASSADS['x509UserProxyFirstFQAN']
else:
 sys.exit(-1)
```

- Quick solution to mitigate the issue
- We need to check which attributes are need by **apel/accounting** and make the same re-population on token case (need to discuss with WLCG and or Condor developers).
- As we did not touch the old part of the code, will not be any issue with the other VO
- Later we could review the assignment on accounting group based on tokens
- Need to check if same issue exist on ARC-CE (on ARC-ce the submit script is just a bash scripts modification and exception could be more eazy)

Token attributes and htcondor classadd

AuthTokenId = "xxxx-xxxx-xxxx-xxxx"

AuthTokenIssuer = "https://atlas-auth.web.cern.ch/"

AuthTokenScopes = "compute.cancel,compute.create,compute.modify,compute.read"

AuthTokenSubject = "xxxx-xxxx-xxxx-xxxx"

SciTokensFile = "/cephfs/atlpan/harvester/tokens/ce/prod/xyzxyzyzxyzxyz

- Those Attribute are not enough to setup a token → group mapping
- AuthTokenScopes need some extra parsing in order to extract roles
- For the moment we mapping manually the AuthTokenIssuer to VoName

BDII (on condor-ce/sched) and APEL

HTCondorCE:

- htcondor-ce-provider (made small Pull-Request)
- htcondor-ce-provider-glue1, home made to publish glue1 params needed by the FR apel collector
- Just modified condor_ce_apel.sh as we do not publish directly to central apel but via the FR collector.

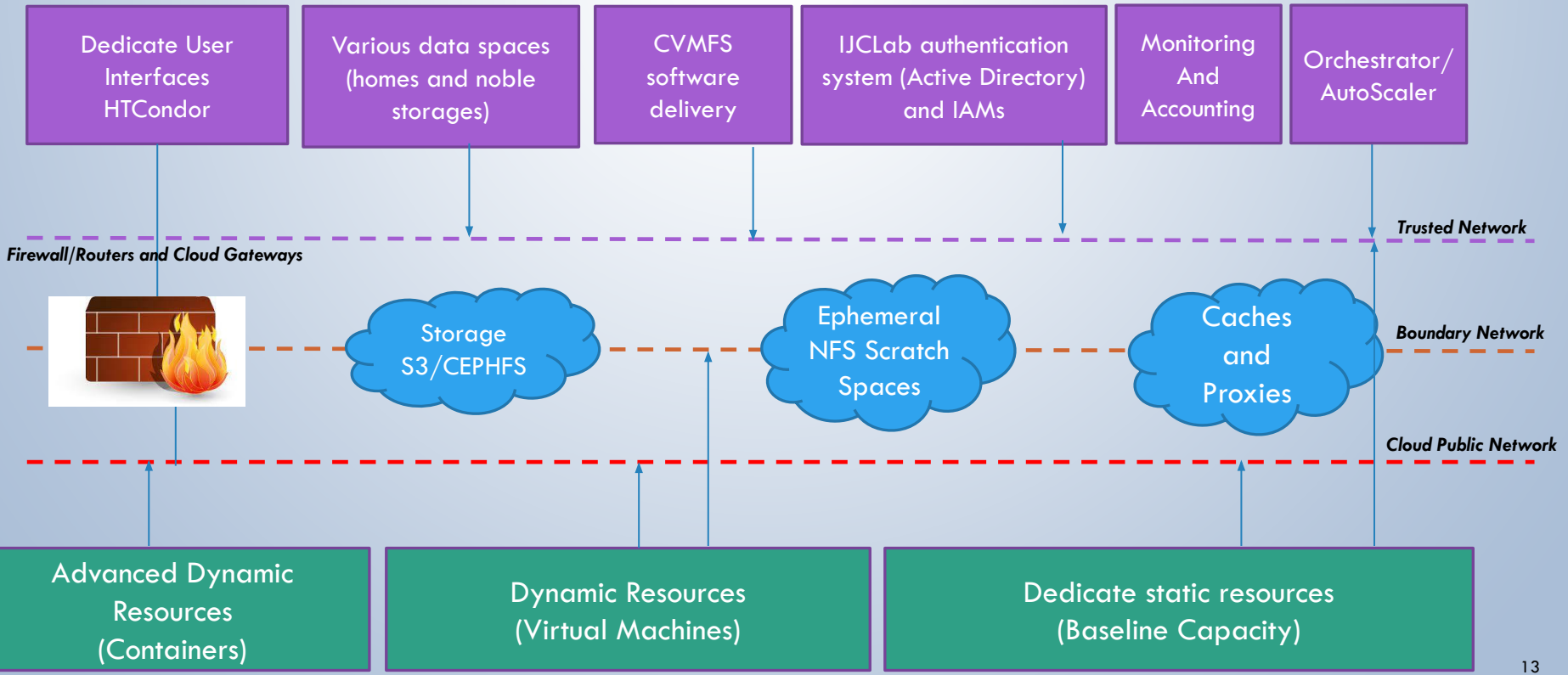
Conclution/Comments

- Quite stable operation on Grid instance
- Smoothly update to **HTCONDOR 9.0.17 and HTCONDOR-CE 5.1.6**
- **Need a revision of** AMD EPYC 256 HT usage (e.g. 256 jobs slot per machine is large enough)
- Prepare the next update to version 10.x.x.

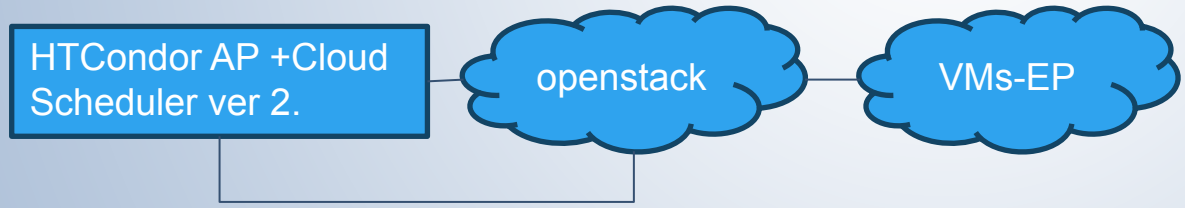
Local User and HTCondor Deployment

- Various use-case of batch submission (e.g. htcondor) at IJCLAB
  - **Local pool for development and analysis jobs of LHC and other experiments**
    - Intention to consolidate all experiment interactives server and/or VMs under a unique condor pool
  - **Provides htcondor pool for embarrassing parallel jobs**
    - Very specific VM topology, usage of openmp and openmpi
  - **Ephemeral htcondor pool for tutorials and student laboratories**
  - **Isolate htcondor pool for particular experience activity**

HIGHLIGTH CONTEXT DIAGRAM OF HTCONDOR VIRTUAL-DATA POOL



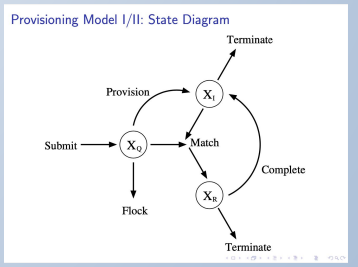
Dynamic provision of virtual machines (workers)



- https://github.com/hep-gc/cloudscheduler

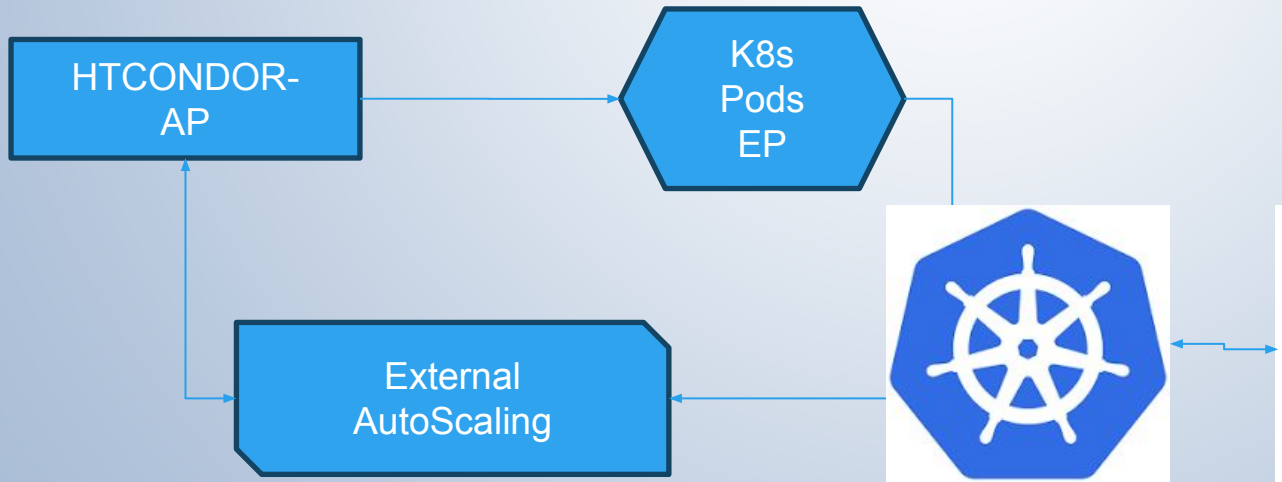
- https://github.com/MatterMiners/tardis

Dynamical Provisioning of Cloud Computing Resources for Batch Processing
Marty Kandes, HEPiX Fall 2016



Autoscaling is a quite difficult process problem due to large number of involved "states" and rates

Advanced Dynamic Resources (Containers)



- Deploy K8s cluster based on standard tools
- Management of all k8s resources
- Base of a user self-service of k8s cluster

v1.21.14/Fedora Coreos32

Auto-scaling HTCondor pools using Kubernetes compute resources
Igor Sfiligoi, Thomas DeFanti, Frank Würthwein
PEARC '22: Practice and Experience in Advanced Research Computing (2022) 57 1-4

Cloud containers integration

- **Different solution for different use case**
  - **Local pool for development and analysis jobs of LHC and other experiments**
    - Static pool and a dynamic VM pool
  - **Provides htcondor pool for embarrassing parallel jobs**
    - Very special static pool
  - **Ephemeral htcondor pool for tutorials and student laboratories**
    - A full personal htcondor integration into K8s (AP+EP)
  - **Isolate htcondor pool for particular experience activity**
    - Dynamic K8s EP integration

Many thanks for yours attention

Questions and Comments ?

BACKUP SLIDES

MULTI-SERVER TOPOLOGY