HTCondor at GRIF

GRIF Tech. group: *A.Bailly-Reyre, C.Cavet, C.Diarra, S.Ferry, A.Garcia, M.Jouvin, J.P.Meyer, M.Mellin, V.Mendoza, G.Philippon, A.Ramparison,* ***A.Sartirana****, F.Schaer, E.Vamvakopoulos.*

Introduction

- This just a **brief report on the HTCondor experience at GRIF**
  - quick description of the setup of the two main pools
    - **HTCondor + ARC-CE at GRIF_IRFU;**
    - **HTCondor + CREAM-CE + HTCondorCE at GRIF_LLR/GRIF_IJCLab;**
  - some **admins feedback:** maintainability, performance, problems, etc.;
- GRIF installations are **pretty "simple"**
  - just **HTC-GRID clusters** with rather standard policies and fairshare rules;
  - …but we have **few things** which may be **worth mentioning**
    - a **distributed pool** shared by GRIF_LLR/GRIF_IJClab;
    - HTCondor + CE with **all CE flavors:** CREAM, ARC, HTCondorCE;
- we have now **5 years** of experience running HTCondor
  - a good time to make a **first evaluation** of our experience w.r.t the initial expectations and motivations.

Grille pour la Recherche en l'IdF.

5 labs in the **Paris** region

- were 6: LAL/IPN merged in 01/2020;
- **single** fed. and **distr.** WLCG **T2**;
- 18k cores, ~9PB disk;

GRIF has **3** GRID **clusters**

IRFU:

- HTCondor + ARC-CE;
- 7.4k cores;

LLR/IJCLab:

- HTCondor + 2 CREAM-CE + 2 HTCondorCE;
- 7.2k cores;

LPNHE:

- Torque/Maui + CREAM. 3.6k cores;
- Migrating to HTCondor + HTCondorCE now;
- Will join the LLR/IJCLab cluster.

Motivations

- In 2014, starting the migration to HTCondor was strongly motivated by **torque/maui decommissioning**
  - **maui** was **no longer maintained/developed**
    - potential security issues;
    - potential scaling issues as sites were growing bigger;
  - **multicore** jobs **support** required by LHC Vos
    - not straightforward to implement it in torque/maui;
  - no hierarchical fairshare;
  - HTC common choice for grid sites. Very positive feedback
- **CREAM-CE EOL** was looming (plus **doubts on** CREAM/HTCondor **integration**)
  - **IRFU** decided to move to **ARC-CE;**
  - **LLR and IJCLab** decided to make a **1st step with** HTCondor + **CREAM**
    - eventually the 2nd step waited until **2020** with the **deployment of 2 HTCondorCE.**

Timeline

Q4/2014 Irfu — *HTCondor + ARC in prod (new cluster)*

04/2015 LLR — *HTCondor + CREAM in prod (decom. Torque/maui)*

06/2015 IJCLab Irène Joliot-Curie — *HTCondor + CREAM in prod (new cluster)*

07/2015 IJCLab Irène Joliot-Curie — *decommissioned Torque/Maui + CREAM*

05/2017 LLR IJCLab Irène Joliot-Curie — *merged to distributed pool*

02/2019 Irfu — *decommissioned Torque/Maui + CREAM*

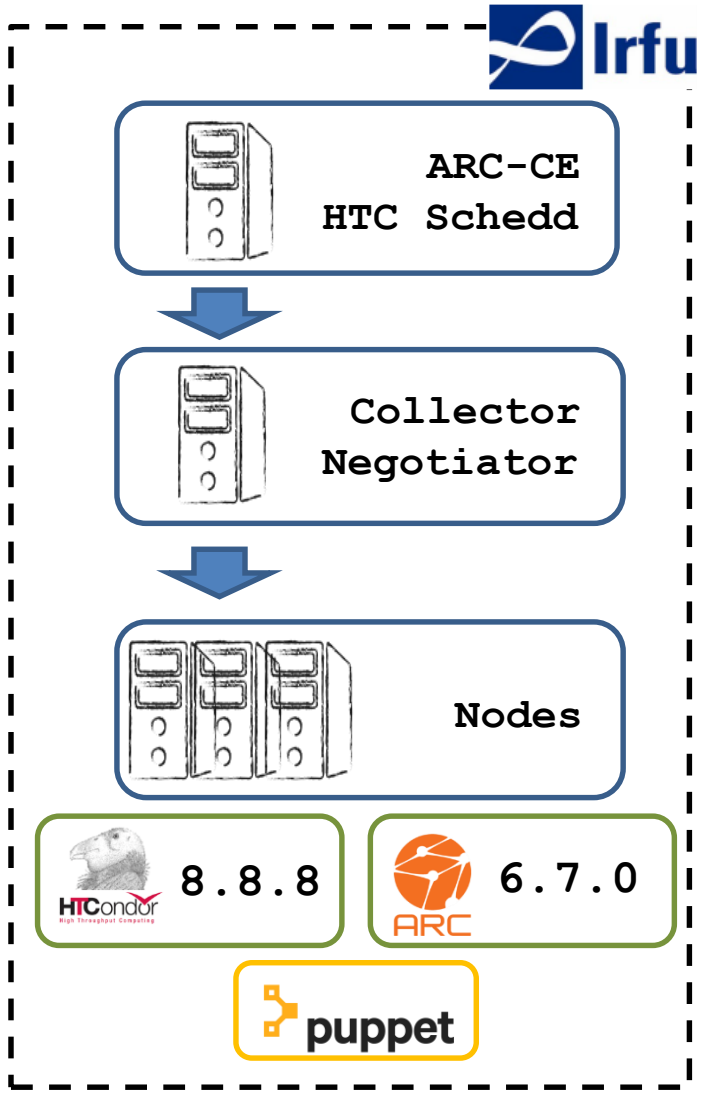
05/2020 LLR IJCLab Irène Joliot-Curie — *1st HTCondorCE in prod (LLR)*

09/2020 LLR IJCLab Irène Joliot-Curie — *2nd HTCondorCE in prod (IJCLab)*

10/2020 LPNHE Paris — *HTC + HTCondorCE in prod*

HTCondor @ IRFU

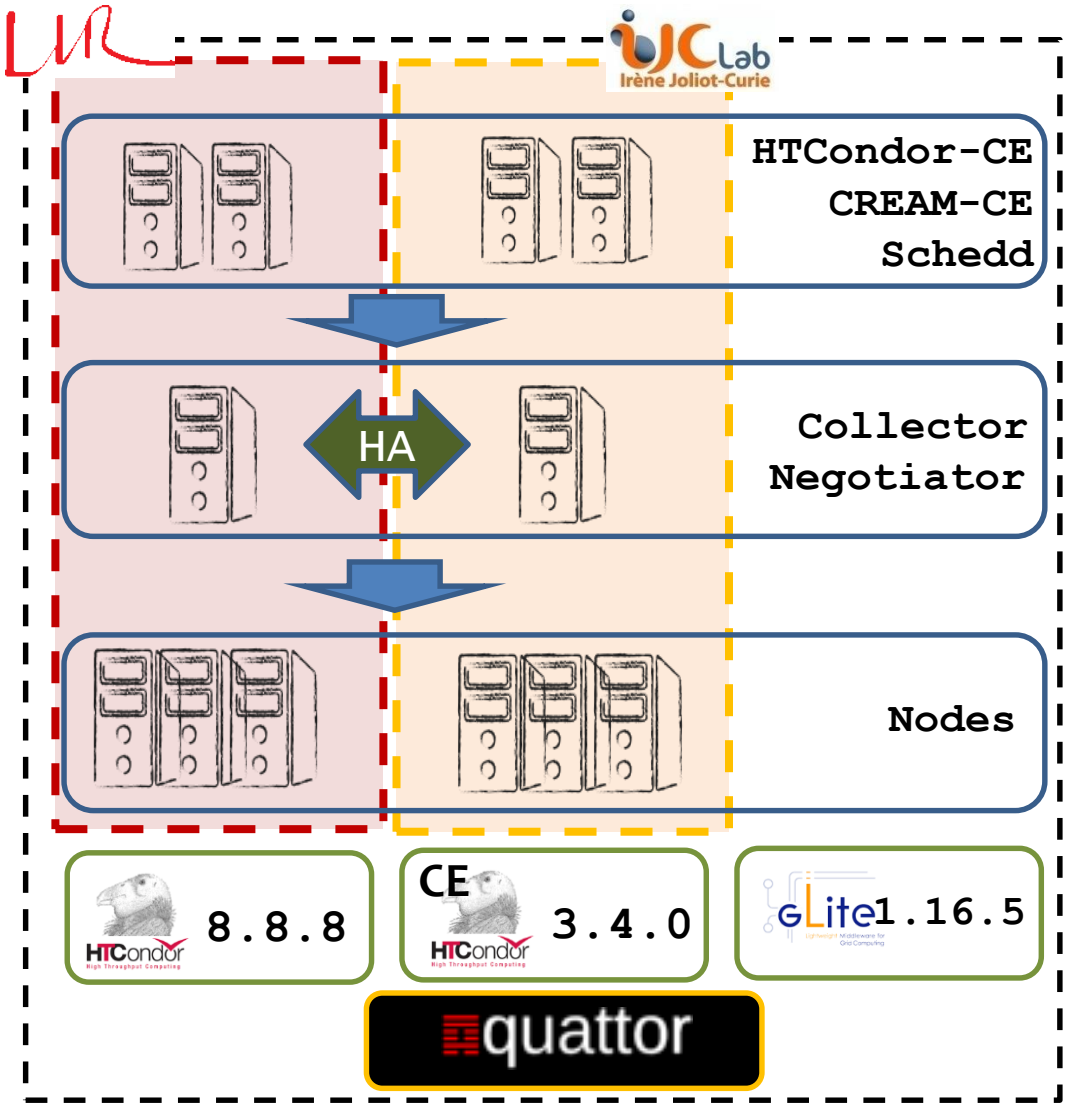
- **ARC** provides batch **plugins** for condor
  - perform actions: submit, cancel, etc;
  - translate **grid requirements into HTC requirements;**
  - we **modified** the submit **plugin** in order to
    - **map VO/FQAN** and requirements **to accounting groups;**
    - format vo names;
    - **correct** grid **requirements** (e.g. required RAM);
- **limits** and **policies** enforced via HTCondor
  - **cgroups** integration: job req. -> cgroup limits;
  - **SYSTEM_PERIODIC HOLD/REMOVE** control 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.**



[*] Details in back slides

HTCondor @ LLR/IJClab…

- **Distributed** pool
  - HTcondor uses only 1 port (“**shared_port**”) which makes **WAN clusters very easy** to build;
- **High Avail.** for Negotiator/Collector
  - straightforward with “had” condor service;
- 1 **CREAM** + 1 **HTCondorCE** for **each sub-site**
  - each CE gives access to all resources;
  - eventually CREAM will be decommissioned (Q4 2020);
- all the above would have very hard to build with, e.g., Torque/Maui.

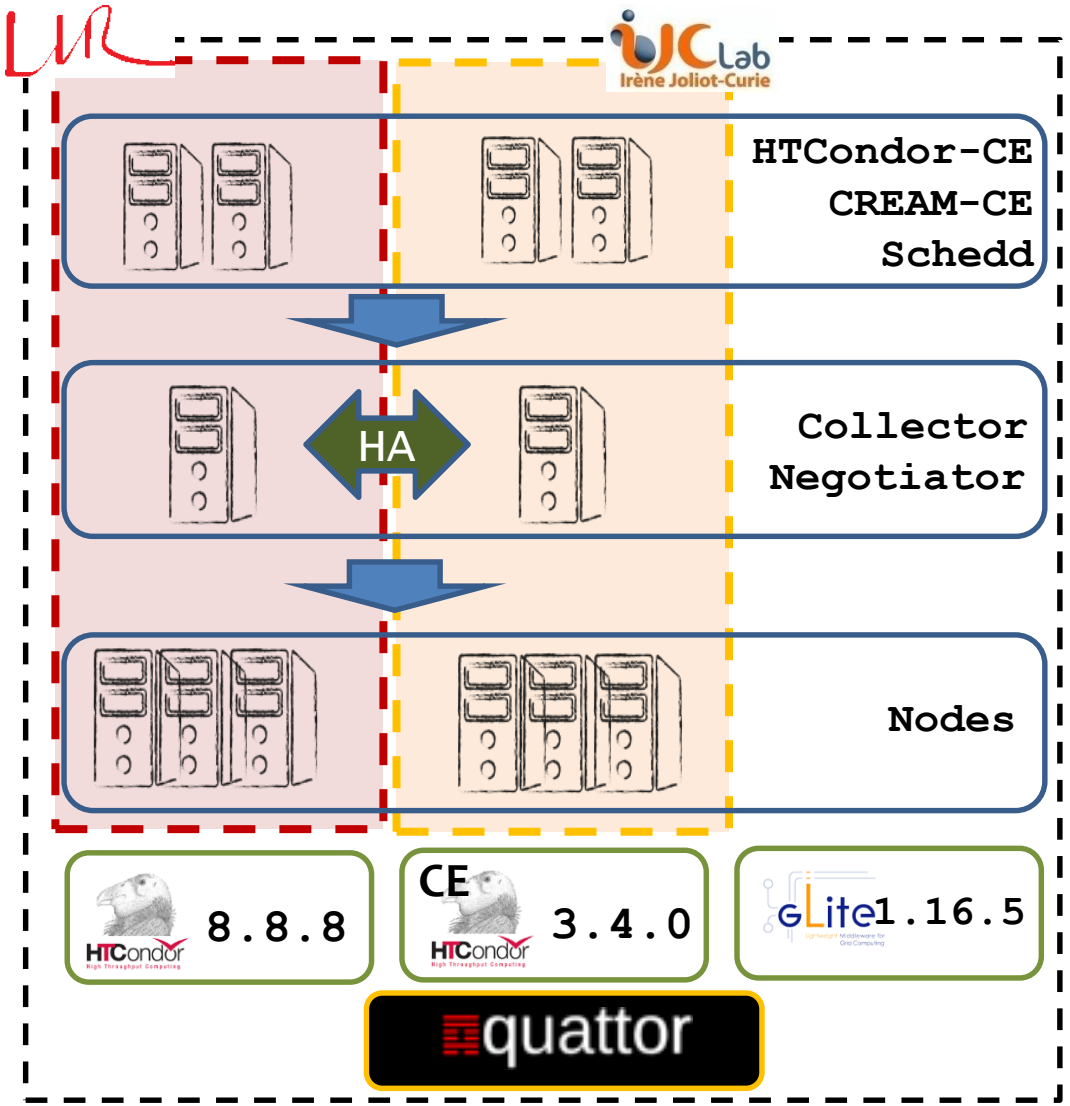


…HTCondor @ LLR/IJClab…

CE/Batch integration

- get GRID job **DN/VO/FQAN/Queue**…
- …into ClassAds and **map** to HTC job params
  - the **AccountingGroup**
  - an expr for **ConcurrencyLimits**
  - **WNTag:** restrict jobs to tagged node
  - **PolicyGroup:** set of policies (e.g. WCT)
- in CREAM: /usr/libexec/**condor_local_submit_attributes.sh**
- in HTCondorCe via a **Job Router hook**
- matching "(VO,FQAN,DN,Queue)" against a **set of regexp-based** substitution rules
  - 4 sets: **group, limit, policy, tag**
  - this allows for a **very flexible mapping**
  - for coherence **same set** for the whole cluster
- no cgroups (can be easily done if needed).

[*] Details in back slides



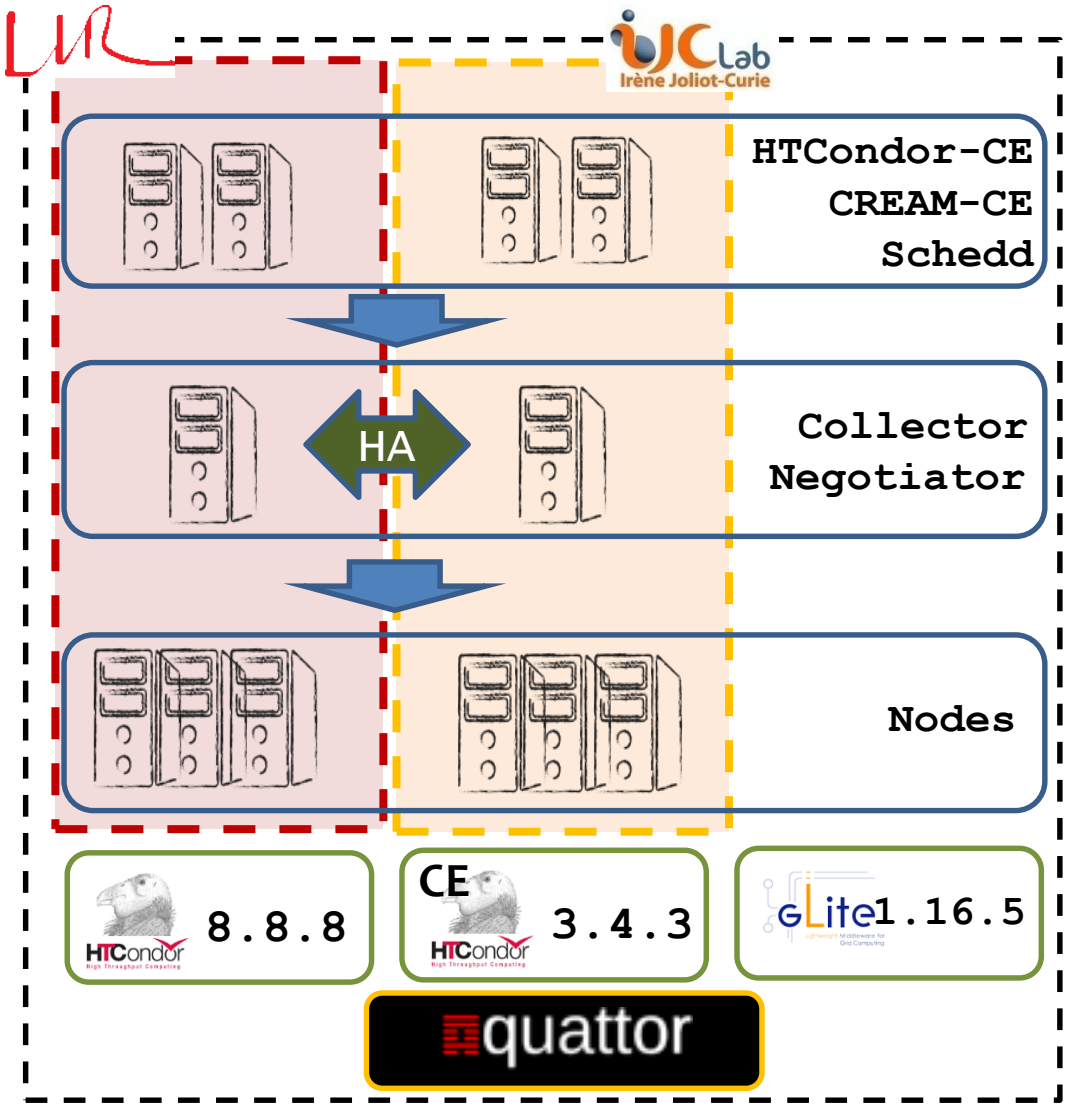
…HTCondor @ LLR/IJClab

BDII (on the CE/schedd)

- CREAM: custom(ized) scripts:
  - /usr/libexec/info-dynamic-condor
  - /usr/libexec/lrmsinfo-condor
- HTCondorCE:
  - htcondor-ce-provider (made small PR)
  - htcondor-ce-provider-glue1, home made to publish glue1 params needed by the FR apel collector

APEL: (on the CE/schedd)

- CREAM: custom script
  - reformatting logs then call apelparser with "type=HTCondor"
- HTCondorCE: htcondor-ce-apel pkg
  - just modified condor_ce_apel.sh as we do not publish directly to central apel but via the FR collector.



Feedback…

- After **5 years, very positive feedback on HTCondor** as a batch system
  - only few months using HTCondorCE but so far everything seems fine;
- setting up a working cluster is rather **simple**
  - most of standard things work **out of the box;**
  - except some very peculiar stuff (i.e. CREAM), of course;
  - some aspects of **HTCondorCE not** that **much documented** (e.g. LCMAP);
  - should be careful with default parameters (e.g. history logs rotation);
- ops/config **logic is** pretty **different** from, e.g. Torque/maui
  - takes some time to get used to it (classads, matching, etc.);
  - the instantaneous status is often incoherent
    - integration over few minutes gives a better view;
  - very powerful framework
    - this also means complexity which is mostly unused in our case (but is there).

...Feedback

- Running in production is **extremely stable**
  - **scaling** is impressive: almost no changes in config from functionality testbed to full production scale;
  - rarely saw the batch system stuck even under big load (at least at our scale);
  - most of the issues come from CREAM and ARC CE. Or from misconfiguration;
  - **ARC CE:** all jobs stdout/stderr (sched) + ARC IO. **Load issues** (need SSDs);
  - @IRFU **mcore/score** jobs idle/running **unbalance** issue (mcore stay idle)
    - no matter what defrag settings we tried to use and despite our recent attempt to favor mcore jobs with GROUP_SORT_EXPRESSION;
- but a couple of annoying flaws…
  - binary change or reconf. cause **demon restart** which **triggers nodes draining**
    - this is particularly annoying when upgrading;
  - **upgrades** have had important **interface changes** (command outputs, classads, configs, etc.) a bit more often than one would have liked
    - latest example: apel cron in HTCondorCE: from sched (3.4.0) to sys (3.4.3).

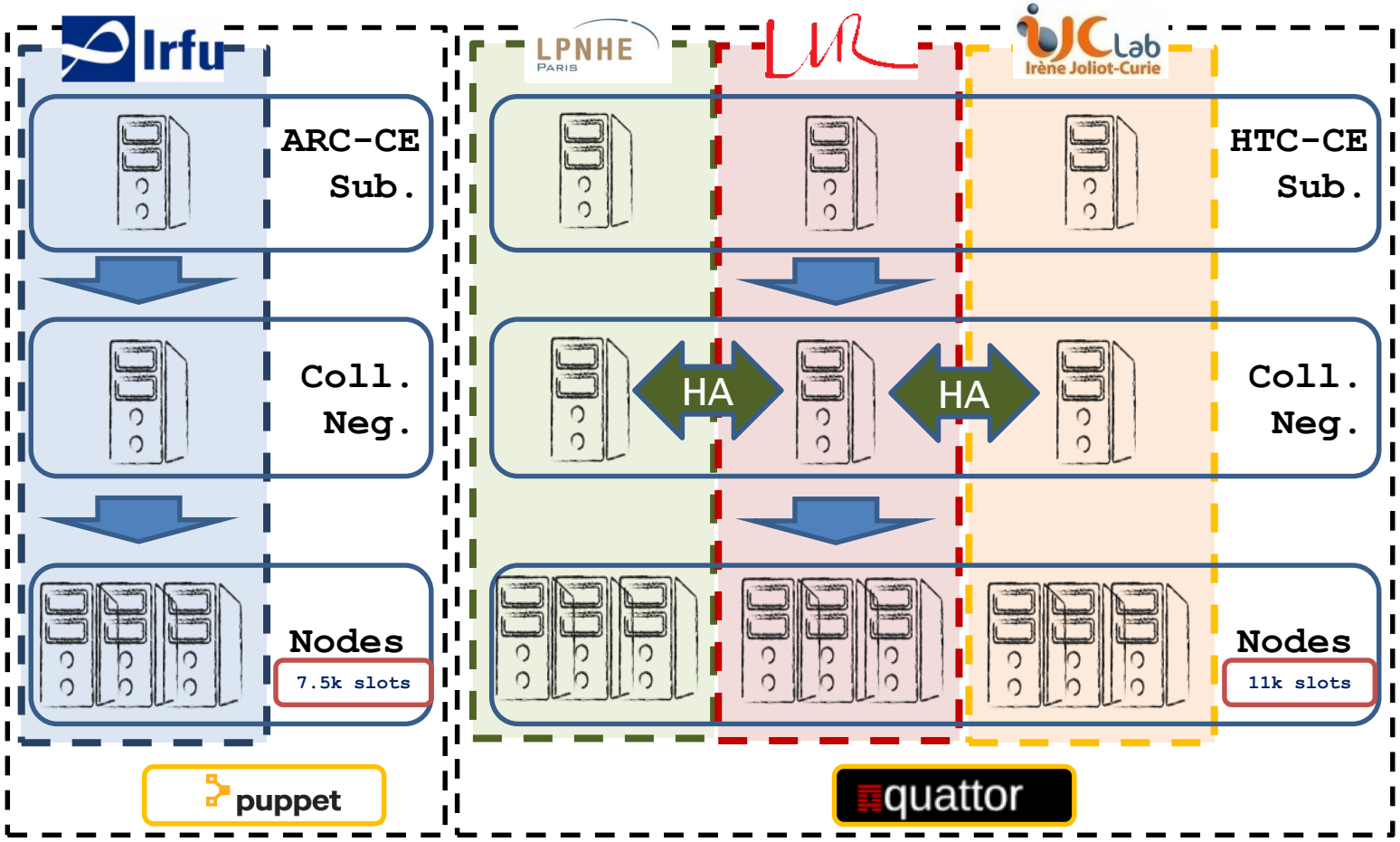
The Future

Near future…

- **LPNHE** now moving **to HTC-CE**
  - migration ongoing;
  - will **join the LLR/IJClab** pool in the next months;
- **decommission CREAM-CE**
  - all LHC already on HTC-CE at LLR/IJCLab. Need to move smaller VOs;
  - LPNHE will decommission CREAM with the migration;

…longer term

- 2 big clusters may **merge**
  - tech diff.: not that simple
- more sophisticated setup
  - e.g. **containerized WNs**…
  - IJCLab plans to add a container universe to allow local users (non grid) to submit jobs on its OpenStack cloud (4.5 kcores).



Summary

- GRIF has been running **HTCondor since 2014**
  - ARC + HTCondor pool at IRFU;
  - CREAM + HTCondorCE + HTCondor at LLR/IJC
    - LPNHE will soon join;
- running **HTCondorCE since 5/2020**
  - so far everything seems fine;
  - the deployment was quite straightforward
    - we **struggled a bit to find documentation;**
- our **experience** so far has been **very positive**
  - stable and well performing and powerful tool;
  - **allowed** us to move toward the longstanding goal of merging the subsites cluster into a single **distributed pool;**
  - some headaches with upgrades.

Backup Slides

HTCondor @ IRFU

```
FQAN=$(voms-proxy-info -file $x509 --fqan|tr '\n' ','|head -1);
VO_NAME=$(voms-proxy-info -file $x509 --vo|sed -e 's/[^a-zA-Z0-9]/_/g');
# accounting group is the VO name for LHC VOs, prefixed by nonlhc for others
shopt -sq nocasematch
if [[ "${VO_NAME}" =~ ^(atlas|cms|alice|lhcb)$ ]] ; then
    accounting_group="group_${VO_NAME}"
    if [[ "x$FQAN" =~ "/cms/local/Role=pilot" ]]; then
      accounting_group="${accounting_group}.t3" ;
    fi
else
  accounting_group="group_nonlhc.${VO_NAME}"
fi

#add a "multicore" subgroup suffix if using ... multicore
if [ ! -z $joboption_count ] && [ $joboption_count -gt 1 ] ; then
  accounting_group="${accounting_group}.mcore"
fi
echo "accounting_group = ${accounting_group}" >> $LRMS_JOB_DESCRIPT
shopt -uq nocasematch
```

```
# update 2017-03-07 : make sure jobs request a sensible amount of VMEM :
memory_req = $(validate_memory_req memory_req)
memory_kbytes=$[ memory_kbytes * joboption_count ]
memory_req=$[ memory_req * joboption_count ]
```

Modified submit-condor-job

- ***Accounting groups***
- ***Memory requirements***

```
# hold jobs using absurd amounts of disk (30+ GB) or using more memory than requested.
SYSTEM_PERIODIC_HOLD = \
   (JobStatus == 1 || JobStatus == 2) && ((DiskUsage > $(MAX_DISK_KB) || ResidentSetSize > JobMemoryLimit * 2 ))

## Put job on hold when outsandboxfile size too big.MAX_TRANSFER_OUTPUT_MB = 50
# Report why went on hold.
SYSTEM_PERIODIC_HOLD_REASON = strcat("Job in status ", JobStatus, " put on hold by SYSTEM_PERIODIC_HOLD due to ", \
  ifThenElse(isUndefined(DiskUsage) || DiskUsage < MAX_DISK_KB, strcat("memory usage ", ResidentSetSize, " > ", JobMemoryLimit * 2), \
  ifThenElse(LastHoldReasonCode == 33, "Output file size > MAX_TRANSFER_OUTPUT_MB", \
  strcat("disk usage ", DiskUsage, " > ", MAX_DISK_KB))), ".")
```

Policies

- ***SYSTEM_PERIODIC_HOLD***
- ***SYSTEM_PERIODIC_REMOVE***

```
SYSTEM_PERIODIC_REMOVE = $(REMOVE_MAX_WALLTIME) || $(REMOVE_HELD_RUNCOUNT) || $(REMOVE_HELD_TOOLONG) ||
$(REMOVE_MAX_IDLE) || $(REMOVE_MAXRUN_CMS_SAM)

SYSTEM_PERIODIC_REMOVE_REASON = \
  strcat("Job removed by SYSTEM_PERIODIC_REMOVE due to ", \
  ifThenElse((JobStatus == 5 && LastHoldReasonCode == 33), "Output file size > MAX_TRANSFER_OUTPUT_MB", \
  ifThenElse( $(REMOVE_HELD_TOOLONG)   , "being in hold state for TIME_MAX_HOLD_STATUS hours", \
  ifThenElse( $(REMOVE_MAX_WALLTIME)   , "runtime longer than allowed", \
  ifThenElse( $(REMOVE_HELD_RUNCOUNT) , "too many restarts", \
  ifThenElse( $(REMOVE_MAX_IDLE)       , "JobStatus remaining idle too long", \
  ifThenElse( $(REMOVE_MAXRUN_CMS_SAM) , "CMS SAM test ran for too long", \
  "input files missing (default)"\    ))))),\  ".")
```

HTCondor @ LLR/IJClab

```
#SCHEDD_DEBUG = D_FULLDEBUG
#JOB_ROUTER_DEBUG = D_FULLDEBUG
JOB_ROUTER_HOOK_KEYWORD = MyHook
MyHook_HOOK_TRANSLATE_JOB = /etc/condor-ce/hook.py
JOB_ROUTER_SCHEDD2_POOL = llrcondor.in2p3.fr:9618 grid09.lal.in2p3.fr:9618
```

HTCondorCE integration

- *Job router config. Defines a hook*
- *hook code relevant for mapping*
- *mapping rules*

```
….
MATCH_FMT = '(%(x509UserProxyVOName_Fmt)s,‘
MATCH_FMT += '%(x509UserProxyFirstFQAN)s,‘
MATCH_FMT += '%(x509userproxysubject)s,‘
MATCH_FMT += 'condorce)‘
MATCH = MATCH_FMT % CLASSADS
MATCH = MATCH.replace('"', '')

LIMIT = mapping(MAPPER, 'limit', MATCH)
ACCGRP = mapping(MAPPER, 'group', MATCH)
OWNER = CLASSADS['Owner'].replace('"', '')
if LIMIT != 'NONE':
    CLASSADS['ConcurrencyLimits'] = "\"%s\"" % LIMIT
CLASSADS['AcctGroup'] = "\"%s\"" % ACCGRP
CLASSADS['AccountingGroup'] = "\"%s.%s\"" % (ACCGRP, OWNER)
CLASSADS['AcctGroupUser'] = CLASSADS['Owner']
CLASSADS['WNTag'] = "\"%s\"" % mapping(MAPPER, 'tag', MATCH)
CLASSADS['PolicyGroup'] = "\"%s\"" % mapping(MAPPER, 'policy', MATCH)
…
```

```
<group-mapping>
 <group match="^\(atlas,[^,]+,[^,]+,multicore\)$" result="group_atlas.multicore“ />
 <group match="^\(atlas,[^,]+,[^,]+,analysis\)$" result="group_atlas.analysis"/>
 <group match="^\(atlas,[^,]+,[^,]+,condorce\)$" result="group_atlas.multicore"/>
 <group match="^\(([^,]+),\S+admin\S+,[^,]+,[^,]+\)$" result="group_$1.admin"/>
 <group match="^\(([^,]+),\S+prod\S+,[^,]+,[^,]+\)$" result="group_$1.prod"/>
 <group match="^\(([^,]+),\S+pilot\S+,[^,]+,[^,]+\)$" result="group_$1.pilot"/>
 <group match="^\(([^,]+)" result="group_$1.default"/>
 <policy match="^\(([^,]+),\S+admin\S+,[^,]+,[^,]+\)$" result="$1.admin"/>
 <policy match="^\(([^,]+),[^,]+,[^,]+,([^,]+)\)$" result="$1.$2"/>
 <tag match="^\(vo_llr_in2p3_fr,[^,]+,[^,]+,[^,]+\)$" result="llr"/>
 <tag match="^\([^,]+,[^,]+,[^,]+,[^,]+\)$" result="ALL_BOTH"/>
 <tag match=".*" result="ALL"/><limit match=".*" result="NONE"/>
</group-mapping>
```

HTCondor @ LLR/IJClab

CreamCE integration

- condor_local_submit_attributes.sh
- mapping rules

```
…
#Getting informations about the user identity
FQAN=$(voms-proxy-info --fqan|head -1);
SUBJECT=$(voms-proxy-info --acsubject);
VO_NAME=$(voms-proxy-info --vo);
VO_NAME_FORMATTED=$(echo $VO_NAME|tr '.-' '_');
IdentityString='('$VO_NAME_FORMATTED','$FQAN','$SUBJECT','$QUEUE')‘

#Map the user identity to an accounting
groupAcctGroup=$(/usr/libexec/matching_regexps "$IdentityString"
/etc/condor/groups_mapping.xml group 2>/dev/null)
PolicyGroup=$(/usr/libexec/matching_regexps "$IdentityString"
/etc/condor/groups_mapping.xml policy 2>/dev/null)
WNTag=$(/usr/libexec/matching_regexps "$IdentityString"
/etc/condor/groups_mapping.xml tag 2>/dev/null)
Concurrenclimits=$(/usr/libexec/matching_regexps "$IdentityString"
/etc/condor/groups_mapping.xml limit 2>/dev/null)
echo 'accounting_group='$AcctGroup
echo ‘accounting_group_user='$(whoami)
if [[ "xConcurrenclimits" != "xNONE" ]];
then
    echo 'concurrency_limits_expr='$Concurrenclimits
fi
echo '+CreamQueue="'$QUEUE'"‘
echo '+PolicyGroup="'$PolicyGroup'"‘
echo '+WNTag="'$WNTag'"‘
…
```

```
<group-mapping>
 <group match="^\(atlas,[^,]+,[^,]+,multicore\)$" result="group_atlas.multicore“ />
 <group match="^\(atlas,[^,]+,[^,]+,analysis\)$" result="group_atlas.analysis"/>
 <group match="^\(atlas,[^,]+,[^,]+,condorce\)$" result="group_atlas.multicore"/>
 <group match="^\(([^,]+),\S+admin\S+,[^,]+,[^,]+\)$" result="group_$1.admin"/>
 <group match="^\(([^,]+),\S+prod\S+,[^,]+,[^,]+\)$" result="group_$1.prod"/>
 <group match="^\(([^,]+),\S+pilot\S+,[^,]+,[^,]+\)$" result="group_$1.pilot"/>
 <group match="^\(([^,]+)" result="group_$1.default"/>
 <policy match="^\(([^,]+),\S+admin\S+,[^,]+,[^,]+\)$" result="$1.admin"/>
 <policy match="^\(([^,]+),[^,]+,[^,]+,([^,]+)\)$" result="$1.$2"/>
 <tag match="^\(vo_llr_in2p3_fr,[^,]+,[^,]+,[^,]+\)$" result="llr"/>
 <tag match="^\([^,]+,[^,]+,[^,]+,[^,]+\)$" result="ALL_BOTH"/>
 <tag match=".*" result="ALL"/><limit match=".*" result="NONE"/>
</group-mapping>
```

HTCondor @ LLR/IJClab

```
…
START_TAG = ((WNTag == "ALL7")||(WNTag == "ALL_BOTH")||(WNTag ==
"WRAP")||(WNTag == "llr")|| false)

…

START = $(START_DRAIN) && $(START_OFFLINE) && $(START_TAG) &&
$(START_CUSTOM)
…
```

Policies

- *WN config: WNTags*
- *Schedd config. Using PolicyGroup and SYSTEM_PERIODIC_REMOVE to enforce limits*
- *Submit requirements*

```
MAXMEM = 2000
MAXWALLTIME = IfThenElse( PolicyGroup == "cms.admin",20,\
  IfThenElse( PolicyGroup == "vo_grif_fr.gridq",1,\
  4320))

SYSTEM_PERIODIC_REMOVE = (JobStatus == 5 && time() - EnteredCurrentStatus > 3600*48)\
  || ((JobStatus == 2)&&(($(MAXWALLTIME)>0)&&((time() - EnteredCurrentStatus) >
(60*$(MAXWALLTIME)))))\
```

```
SUBMIT_REQUIREMENT_NAMES = slots
SUBMIT_REQUIREMENT_slots = (RequestCpus == 1)||(RequestCpus == 8)
SUBMIT_REQUIREMENT_slots_REASON = "Only 1core and 8cores jobs are accepted."
```

Defrag config:

```
…
DEFRAG_DRAINING_MACHINES_PER_HOUR = 100
DEFRAG_INTERVAL = 300
DEFRAG_MAX_CONCURRENT_DRAINING = 100
DEFRAG_MAX_WHOLE_MACHINES = 300
DEFRAG_SCHEDULE = graceful

## Allow some defrag configuration to be settable
DEFRAG.SETTABLE_ATTRS_ADMINISTRATOR = DEFRAG_MAX_CONCURRENT_DRAINING,DEFRAG_DRAINING_MACHINES_PER_HOUR,DEFRAG_MAX_WHOLE_MACHINES
ENABLE_RUNTIME_CONFIG = TRUE

# If a machine have more than 8 CPUs free or if a machine is a 8 or less CPUs machine
# We put a negative value to be not desirable
# Else, we try to find the machine who is closest to be free

DEFRAG_RANK = ifThenElse(Cpus >= 8, -10, (TotalCpus - Cpus)/(8.0 - Cpus))

# Definition of a "whole" machine:
# - anything with 8 free cores
# - empty machines
# - must be configured to actually start new jobs (otherwise machines which are deliberately being drained will be included)
DEFRAG_WHOLE_MACHINE_EXPR = ((Cpus == TotalCpus) || ((Cpus >= 8)&&(DynamicSlot=!=true))) && (Offline=!=True)

# Decide which machines to drain
# - must be Partitionable
# - must be online
# - must have more than 8 cores
DEFRAG_REQUIREMENTS = PartitionableSlot && Offline=!=True && TotalCpus>8
…
```

HTCondor @ LLR/IJClab

```
# LCMAPS policy definition# Auto-generated by Quattor ncm-lcmaps. DO NOT EDIT.
## default lookup path for modules
path = /usr/lib64/lcmaps

vomspoolaccount = "lcmaps_voms_poolaccount.mod“
                      "-gridmapfile /etc/lcmaps/gridmapfile“
                      "-gridmapdir /etc/grid-security/gridmapdir“
                      "-override_inconsistency“
                      "--add-primary-gid-from-mapped-account “

good = "lcmaps_dummy_good.mod“
bad = "lcmaps_dummy_bad.mod“


vomslocalgroup = "lcmaps_voms_localgroup.mod“
                     "-groupmapfile /etc/lcmaps/groupmapfile“
                     "-mapmin 0“
                     "--map-to-secondary-groups“

vomslocalaccount = "lcmaps_voms_localaccount.mod“
                       "-gridmapfile /etc/lcmaps/gridmapfile“

# Policies:authorize_only:
vomslocalgroup -> vomslocalaccount
vomslocalaccount -> good | vomspoolaccount
vomspoolaccount -> good|bad
```

HTCondorCE auth.

- ***Lcmaps config***
- ***Condorce mapfile***

```
GSI "^/O=GRID-FR/C=FR/O=CNRS/OU=LLR/CN=llrcondorce.in2p3.fr$" llrcondorce.in2p3.fr@daemon.htcondor.org
GSI "^/O=GRID-FR/C=FR/O=CNRS/OU=LLR/CN=llrt3condor.in2p3.fr$" llrt3condor.in2p3.fr@daemon.htcondor.org
GSI (.*) GSS_ASSIST_GRIDMAP
GSI "(/CN=[-.A-Za-z0-9/= ]+)" \1@unamapped.htcondor.org
CLAIMTOBE .* anonymous@claimtobe
FS (.*) \1
```