EPJ Web of Conferences **245**, 07020 (2020) https://doi.org/10.1051/epjconf/202024507020
CHEP 2019

Dynamic integration of distributed, Cloud-based HPC and HTC resources using JSON Web Tokens and the INDIGO IAM Service

Danele Spiga[1,] [*], *Stefano Dal Pra*[2,**] *Davide Salomoni*[1], *Andrea Ceccanti*[1], *Roberto Alfieri*[3,4]

[1]INFN Sezione di Perugia, Via Alessandro Pascoli 23c, 06123 Perugia (ITALY)
[2]INFN-CNAF, Viale Carlo Berti Pichat, 6/2, 40127 Bologna (ITALY)
[3]Università di Parma, Parco Area delle Scienze 7/A, 4214 Parma (ITALY)
[4]INFN Gruppo collegato di Parma, Parco Area delle Scienze 7/A, 42124 Parma (ITALY)

Abstract. In the past couple of years, we have been actively developing the Dynamic On-Demand Analysis Service (DODAS) as an enabling technology to deploy container-based clusters over hybrid, private or public, Cloud infrastructures with almost zero effort. DODAS is particularly suitable for harvesting opportunistic computing resources; this is why several scientific communities already integrated their computing use cases into DODAS-instantiated clusters, automating the instantiation, management and federation of HTCondor batch systems. The increasing demand, availability and utilization of HPC resources by and for multidisciplinary user communities, often mandates the possibility to transparently integrate, manage and mix HTC and HPC resources. In this paper, we discuss our experience extending and using DODAS to connect HPC and HTC resources in the context of a distributed Italian regional infrastructure involving multiple sites and communities. In this use case, DODAS automatically generates HTCondor batch system on-demand. Moreover it dynamically and transparently federates sites that may also include HPC resources managed by SLURM; DODAS allows user workloads to make opportunistic and automated use of both HPC and HTC resources, thus effectively maximizing and optimizing resource utilization. We also report on our experience of using and federating HTCondor batch systems exploiting the JSON Web Token capabilities introduced in recent HTCondor versions, replacing the traditional X509 certificates in the whole chain of workload authorization. In this respect we also report on how we integrated HTCondor using OAuth with the INDIGO IAM service.

1 Introduction

Since a long time, INFN has been very active in the development of software and tools to simplify the access to Cloud-based as well as HTC and HPC resources, for the execution

* Corresponding author: spiga@infn.it
** Corresponding author: stefano.dalpra@cnaf.infn.it

of scientific data analysis and data simulation. The high level objective for these developments is to provide user-friendly tools and effective solutions for the utilization of heterogeneous (HPC or HTC) and hybrid resources, possibly opportunistically. An important driver in this process has been the open source data and computing platform developed by INDIGO-DataCloud project [1] to simplify the execution of applications on cloud. This represents the foundation for many of the activities and services discussed in this work. It is also with that project that several pilots, demonstrators and proof-of-concepts were started and then developed, with the direct involvement of scientific communities such as HEP, Astroparticles, Gravitational Wave etc.

The lessons we learned carrying on these activities, together with the increasing demand and utilization of HPC resources are the starting point for the presented work, and can be summarized as follows:

- On the demand side: several user communities would like to exploit an integrated, transparent use of HPC and HTC resources. I.e, they would like to run mixed HPC/HTC workloads through a common entry point or method for resource access.
- On the supply side: not being able to efficiently and effectively federate HPC and HTC resources often leads to under- or over-utilization of resources, which could be at least mitigated by an integrated approach.
- Any solution should be compliant with the computing models used by the communities, and should be generally applicable to multiple infrastructures (i.e. proven, open, dynamic, provider-independent).

Two concrete use cases recently demonstrated the general principles stated above: on the one hand, the increasing exploitation by INFN of HPC resources available at the Italian Supercomputing Center CINECA; and, on the other hand, the general integration of HPC and HTC resources provisioned in the context of the "Supercomputing Unified Platform – Emilia-Romagna" (SUPER), a project funded by the Emilia-Romagna region.

The needs of these projects mandate the possibility to transparently integrate, manage and mix HTC and HPC resources. A further essential requirement is that all the solutions developed in these contexts should be immediately applicable on every similar HPC and HTC integration scenarios.

In this contribution we describe the strategy we are adopting for an integrated technical solution coping with the objectives mentioned above. In Sec.2 we report on the requirement analysis, while in Sec.3 we describe the Emilia-Romagna testbed set up between the INFN-CNAF and Parma University sites. Sec. 4 shows the identified solution, detailing the motivations for the decision taken. Finally, the status and early integration results are summarized in Sec. 5. We conclude with the description of the next steps and future directions.

2 Requirements Analysis and use cases

As anticipated, one of the concrete use cases driving the presented activity is the integration of HPC and HTC resources funded and provisioned in the Emilia-Romagna region. The physical allocation of funded hardware is expected to be spread between various universities and research institutions such as INFN, ENEA and CMCC. The end users that will exploit these resources belong to different scientific domains. All this depicts the

high-level scenario where several scientific communities, with a range of diverse use cases, need to transparently exploit both local and distributed computing and storage resources for the execution of dynamic and complex workloads requiring HPC resources, HTC resources, or both. All the resources are meant to be shared following rules and policies agreed among the participants to the distributed infrastructure. Shared resources and mixing of the different resource types should be as transparent as possible for users. In other words, this set of heterogeneous and distributed resources should be perceived by users as a single entity, which internally handles the specified workloads, possibly implementing overflow mechanisms to maximize hardware usage.
The overflow mechanism should be automated whenever possible, but naturally workload routing must be possible based also on specific user requirements, for example for the access and usage of specialized hardware such as GPUs or low-latency interconnects. The rationale behind the overflow automation mechanisms is to enable opportunistic usage of idle resources belonging to the shared pool of resources.

Regarding resource provisioning, a specific requirement that we collected is to grant the cooperation with both Cloud-based compute and storage resources, together with classic batch systems made available over bare metal resources. The Cloud part is a key to cope with the on-demand and dynamic execution of complex workflows coming from the diverse set of scientific domains.

Another key aspect to be taken into account in this work is the authentication and the authorization system. In this respect, the need for a system capable of providing flexibility to accommodate various authentication mechanisms emerged very clearly. Another requirement is the need to support the integration with legacy authentication services, which brings to the need of credential translation capability. A final point is to simplify the user experience with regard to authentication, so an additional goal is the harmonization of multiple identities in a single account. Supporting flexible authorization policies, and possibly aggregating communities or users into groups or virtual organizations is also mandatory.

A final remark is about storage and data handling in general. While this is a major topic, we decided to defer any related evaluation to a second stage, also because there are many parallel activities in this context and we expect to benefit from their results in the near future. In particular, we expect to integrate in further developments of this work results coming from EU projects focused on data management, such as eXtreme-DataCloud [2] and ESCAPE [3], as well as best practices and results originating from WLCG-specific datalake work [4]. Therefore, in this paper we are going to focus mostly on compute-bound workloads, and all the planned validations activities are expected to be performed dealing with remote data access and/or with workloads having limited I/O requirements.

To summarize, the requirements for an integrated solution include the possibility to:

- dynamically instantiate or join batch systems on Cloud resources.
    - Both remote and local access need to be allowed, possibly with distinct priorities
- adopt a federated, group-aware system to handle Authentication and Authorization
- enable overflow mechanisms between batch systems (opportunistic model)
    - Overflow could be based on job requirements, and batch systems are supposed to be geographically distributed
- mix different resource types.
    - An HPC batch system must be transparently accessible to HTC workflows, possibly opportunistically.

3 The Emilia Romagna pilot testbed

The pilot testbed used in this work is based on the resources provisioned by two sites: University and INFN of Parma, and INFN-CNAF in Bologna. The Parma site provides two clusters, one offering HTC resources and the other HPC resources. All these resources are statically instantiated. The other site, INFN-CNAF in Bologna, provides HTC resources, offering them through Cloud APIs based on Openstack. More specifically:

- In Parma, the University and INFN HTC cluster is based on HTCondor 8.8.5; the HPC cluster is based on Slurm 17.11.5. The latter includes 2200 cores (800 Broadwell (BDW), 1100 Knights Landing (KNL), 300 Skylake (SKL)) and has also 16 NVIDIA GPUs (14 Tesla P100, 2 Tesla V100). All the mentioned nodes are interconnected with Intel OmniPath.
- In Bologna, INFN CNAF provides resources through the Cloud@CNAF infrastructure, which is based on Openstack and is part of the larger INFN computing infrastructure, which amounts to approximately 80,000 CPU cores, 60 PB of disk space and 90 PB of tape space

4 The strategy for an integrated solution

As explained above, the overall objective is to identify an integrated solution for a transparent and effective exploitation of different types of distributed resources. Based on the requirement analysis we did a mapping with open source solutions.

4.1 Batch systems

Due to disparate requirements and capabilities, we decided to keep using different batch systems to manage HTC and HPC resources, namely HTCondor and Slurm, respectively.

Since one of the objectives of the work is to allow the exploitation of unused processing resources available in HPC clusters, e.g. by single-node (multicore) or single-core HTC jobs for opportunistic usage, we decided to rely on the job router daemon capability natively built-in by HTCondor. This provides the ability to transform vanilla jobs to the "slurm" batch type, thus allowing HTCondor to interface with Slurm and therefore supporting a mixing of HPC and HTC resources. Another key feature of the job routing daemon is to allow both automatic transformation as well as custom policies. Last but not least, this grants a high level of flexibility since, for example, one could use any python based script for the implementation of a custom policy.

Introducing the other dimension, namely the federation of remotely deployed pools, the first at INFN-CNAF in Bologna and the second in Parma, one can easily realize that, once again, HTCondor is a technically suitable solution. Through the various configurations supported by HTCondor, in order to federate remote resources we identified flocking as the optimal one. This choice has been driven by several factors, first of all the simplicity, as we want to keep things as simple as possible, then because at the time of writing we expect a limited number of "trusted" condor_schedds to federate, and all of them with good networking connectivity. Of course we are aware that we might need to evaluate distinct configuration options once the full mesh of sites in the Emilia-Romagna region will be ready to be federated. We are also ready to take into account a mixing of different solutions, depending on the specific requirements and constraints we will face as soon as the topology will become more complex.

4.2 Access to Cloud based resources: DODAS

As discussed, INFN-CNAF provisions resources via a cloud interface. These resources are expected to be accessed on demand to create HTC clusters. In order to deal with this scenario we decided to rely on DODAS [5,6]. The latter is an open source deployer manager designed to dynamically exploit hybrid Cloud providers. DODAS is an example of a solution which originates from technical guidelines as well as several building blocks delivered by the INDIGO-DataCloud project. From a technical perspective, DODAS has the capability to create and configure analysis clusters on any cloud infrastructure with almost zero effort thanks to the adoption of the "infrastructure as code" paradigm, realized through suitable templates written in the TOSCA [7] language. DODAS is a suitable component for the integrated solution we want to create, given that it supports the dynamic creation of container-based HTC condor pools on Cloud resources, managed by a container orchestrator, such as Kubernetes. The declarative strategy adopted by DODAS, together with its automation capabilities, are key features for our scope. In particular, these DODAS features allow the generation of batch systems over Cloud resources, possibly auto-federating trusted pools.

4.3 AAI

The last element to introduce in this scenario is the authentication and authorization solution. DODAS leverages the INDIGO Identity and Access Management (IAM) service [8], that provides a standards-based solution for flexible authentication and authorization, identity management and administrator-vetted enrollment in support of distributed computing. IAM exposes authentication and authorization information to relying services using standard protocols (OAuth2 [9], OpenID Connect [10], JSON Web Tokens (JWT) [11]) and is currently being adopted by several scientific communities [3], in particular IAM has been selected to be the central component of the next generation, token-based WLCG AAI.

DODAS integrates nicely with IAM for user authentication, authorization and token translation needs. An example is the integration of IAM user management with HTCondor authentication and authorization mechanisms, where DODAS automates the provisioning of HTCondor accounts leveraging the IAM SCIM provisioning API [12]. In particular, DODAS periodically queries the IAM SCIM API to obtain information about registered users and groups and uses this information to create condor mapfiles required to enable GSI authentication. This integration is currently evolving towards a fully token based flow using the SciTokens [13] authentication method supported by the upcoming HTCondor versions, as detailed in Sec.5.

4.4 Connecting the dots

The final solution we identified for our target integration can be summarized as below:

- Rely on HTCondor to overlay distributed resources as well as support mixing HPC and HTC resources through a job routing daemon. The latter is used to enforce custom policies for job transformation, both to match static user requirements and

to dynamically (and opportunistically) exploit idle HPC cycles. HTcondor is also an optimal solution to implement the federation of distributed resource pools.

- Use DODAS as deployer manager to dynamically exploit hybrid Cloud providers. Having a high level of automation, it greatly simplifies the complex task of setting interdependent configurations between several different levels, by providing the ability to dynamically generate self-federating HTC pools.
- Rely on the INDIGO-IAM service for managing authentication and authorization, groups and roles, and to act as a token issuer for HTCondor, allowing advanced and tight integration of resources and services.

5 Current status and early results

At the time of writing, all the basic tests have been performed although the full integration is to be finalized. More in detail, we have implemented and tested the scenario which is shown in Fig. 1. We have successfully tested the submission to a HTCondor pool running at Parma configured to flock jobs to a second pool running at CNAF (Fig.1). This initial test was performed without any selective jobs, so this means that all the jobs can run on both pools depending just on the matching priority.

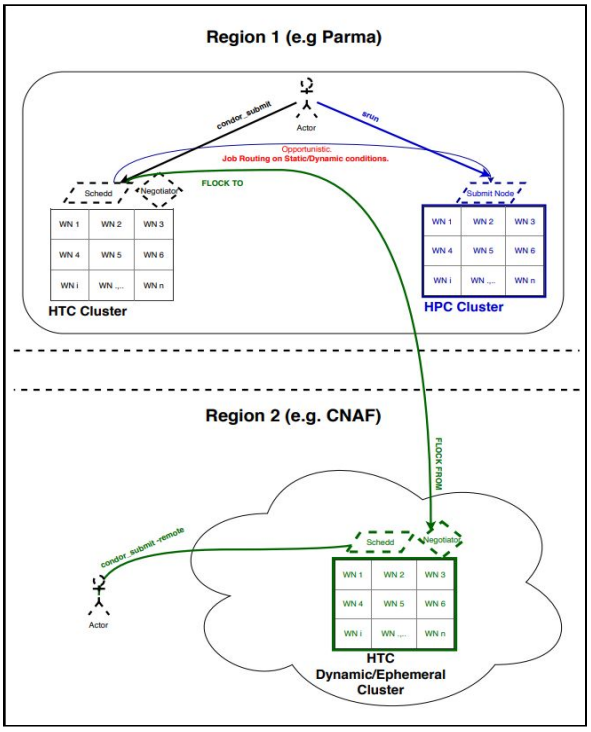

Fig. 1. The figure is a high level representation of the flow we are implementing in the testbed described in this paper. Particularly the green flow represents the specific test done so far.

The flocking from Parma to CNAF was also tested and the job submission has been verified working from both local and remote working stations. The current objective is to enable selective flocking configurations, in order to flock jobs based on user selection, as example through job requirements possibly specified by end users. The last test was done aiming at configuring the job routing to prove that we can achieve HPC-HTC resource mixing, as we originally planned. No technical showstoppers have been met so far in this respect.

Moreover, we have successfully deployed a docker based HTCondor pool managed by Kubernetes using DODAS on Cloud at CNAF (Fig.2.) testing both remote and local submission.

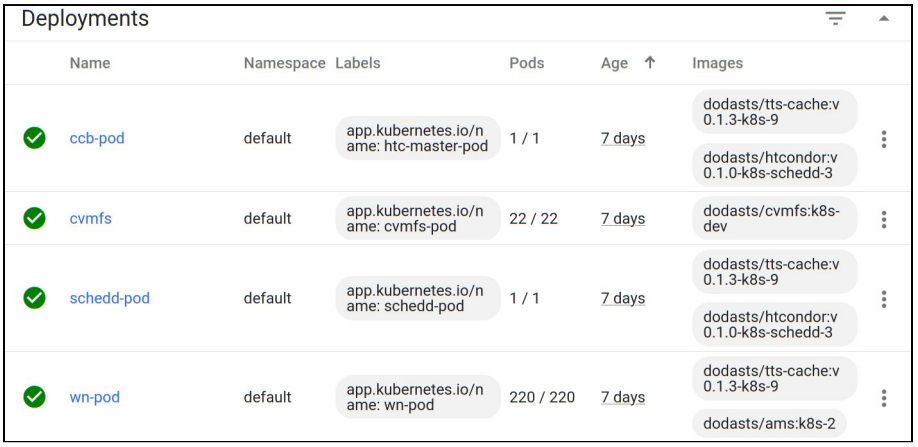

Fig. 2. The figure show a snapshot of the Kubernetes cluster running the HTCondor pool deployed at CNAF using DODAS

More in detail the DODAS generated cluster has been useful to deploy a specific condor version needed to allow the integration of INDIGO-IAM and HTCondor to implement the JWT authentication. The HTCondor version used for this integration test is condor-8.9.3-1.1.osgup.el7.x86_64 which supports the Scitokens based authentication mode. The Scitokens project proposes a profile for OAuth/JWT to enable capability-based authorization and for this reasons we used a dedicated instance of INDIGO-IAM (https://iam-escape.cloud.cnaf.infn.it/) providing support to the same JWT profile. Fig. 3 represents an excerpt of the condor log with the information about the authentication flow.

```
SciToken exchange server status: c: 4, s: 4
SSL authentication succeeded to https://iam-escape.cloud.cnaf.infn.it/,e9bf50ed-16ec-4a2c-8986-9ede3d9fba45
AUTHENTICATE: do_authenticate is 0.
AUTHENTICATE: auth_status == 256 (SCITOKENS)
Authentication was a Success.
ZKM: setting default map to scitokens@unmapped
ZKM: name to map is 'https://iam-escape.cloud.cnaf.infn.it/,e9bf50ed-16ec-4a2c-8986-9ede3d9fba45'
ZKM: pre-map: current user is 'scitokens'
ZKM: pre-map: current domain is 'unmapped'
ZKM: map file already loaded.
ZKM: attempting to map 'https://iam-escape.cloud.cnaf.infn.it/,e9bf50ed-16ec-4a2c-8986-9ede3d9fba45'
ZKM: 1: attempting to map 'https://iam-escape.cloud.cnaf.infn.it/,e9bf50ed-16ec-4a2c-8986-9ede3d9fba45'
ZKM: 2: mapret: 0 included_voms: 0 canonical_user: daniele@users.htcondor.org
ZKM: successful mapping to daniele@users.htcondor.org
ZKM: found user daniele@users.htcondor.org, splitting.
ZKM: post-map: current user is 'daniele'
ZKM: post-map: current domain is 'users.htcondor.org'
ZKM: post-map: current FQU is 'daniele@users.htcondor.org'
AUTHENTICATE: Exchanging keys with remote side.
In wrap.
AUTHENTICATE: Result of end of authenticate is 1.
DC_AUTHENTICATE: authentication of 103.80.251.66 complete.
DC_AUTHENTICATE: Success.
PERMISSION GRANTED to daniele@users.htcondor.org from host 103.80.251.66 for command 519 (QUERY_JOB_ADS_WITH_AUTH),
DC_AUTHENTICATE: sending session ad:
```

Fig. 3. The figure shows an excerpt of schedd daemon log where the Scitoken authentication method is used to authenticate a job submission with INDIGO-IAM JWT.

6 Summary and future directions

In this paper we described the successful activity done to identify the building blocks needed to implement a regional federation of resources. We described the technological

implementation strategy, based on three pillars: HTCondor, DODAS and INDIGO IAM. We successfully verified the identified solutions and we performed the integration test between INDIGO IAM token issuer with a recent version of HTCondor which provides support for JWT-based authentication and authorization. In the future, the plan is first to complete the integration scenarios depicted in Fig.1 and subsequently to measure performances and possible problems. From the architectural point of view, we envision the evaluation of additional solutions, such as the evolution toward a centrally supported HTCondor-CE, similarly to the OSG Hosted-CE model [14]. The latter is also functional to evaluate further strategies needed to extend the mixing models to cover additional, trans-regional or trans-national HPC and HTC resources. Finally, we plan to investigate an accounting model made possible when using the HTCondor-CE, as it can keep track of the usage runtime for the resources delivered upon requests managed by it.

References

1. Salomoni, D., Campos, I., Gaido, L. et al. INDIGO-DataCloud: a Platform to Facilitate Seamless Access to E-Infrastructures, J Grid Computing (2018) 16: 381. https://doi.org/10.1007/s10723-018-9453-3
2. D. Cesini et al, (2018). "The eXtreme-DataCloud project: data management services for the next generation distributed e-infrastructures." 1-4. 10.1109/ROLCG.2018.8572025.
3. S.Campana et al, (2019) "ESCAPE prototypes a Data Infrastructure for Open Science", proceedings of this conference
4. D.Bersano et al. HEP Software Foundation Community White Paper Working Group -- Data Organization, Management and Access (DOMA), arXiv:1812.00761 [physics.comp-ph]
5. D. Spiga et al. Sep.2019, Exploiting private and commercial clouds to generate on-demand CMS computing facilities with DODAS, https://doi.org/10.1051/epjconf/201921407027
6. D.Spiga et al (2019), "The DODAS Experience on the EGI Federated Cloud", proceedings of this conference
7. Palma, D., Rutkowski, M., Spatzier T.: TOSCA Simple Profile in YAML Version 1.1. Tech. rep., OASIS Standard. http://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.1/TOSCA-Simple-Profile-YAML-v1.1.html (2016) [Google Scholar]
8. Andrea Ceccanti, Enrico Vianello, & Marco Caberletti. (2018, May 18). INDIGO Identity and Access Management (IAM) (Version v1.4.0). Zenodo. http://doi.org/10.5281/zenodo.1874791
9. D. Hardt, The OAuth 2.0 Authorization Framework, RFC 6749, IETF Tools (2012), https://tools.ietf. org/rfc/rfc6749.txt
10. OpenID Foundation, The OpenID Connect identity layer (2018), https://openid.net/connect/
11. M.B. Jones, J. Bradley, N. Sakimura, The JSON Web Token RFC, RFC 7519, IETF Tools (2015), https: //tools.ietf.org/rfc/rfc7519.txt
12. The INDIGO IAM SCIM API. https://indigo-iam.github.io/docs/v/current/user-guide/api/scim-api.html

13. The SciTokens project, https://scitokens.org
14. https://opensciencegrid.org/docs/compute-element/hosted-ce/