

The **HTC**Condor Software Suite

Compute Entry Point (CE)

Miron Livny

John P. Morgridge Professor of Computer Science
UW Center for High Throughput Computing
Morgridge Institute for Research



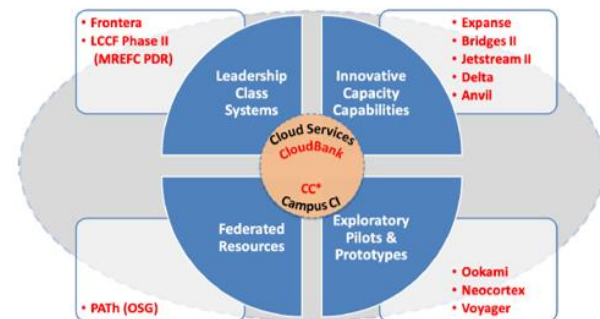
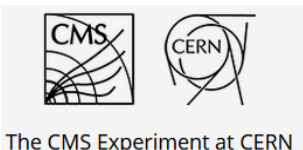
project - a Partnership launched by the NSF in 2020 between UW-Madison Center for High Throughput Computing (**CHTC**) and the **OSG** Consortium to advance **Throughput Computing**.



Software Suite (**HTCSS**) manages High Throughput workloads across all forms of research computing resources from laboratory and campus clusters to commercial clouds and national HPC facilities



OSG services enable organizations like science collaborations and campuses to build and operate (private) distributed high throughput computing environments (most of them HTCCondor Pools) across >200 (autonomous) clusters



Aligned with NSF CI Eco-system



Condor
High Throughput Computing

The Hunter of Idle Workstations (1985)



The Handler of HTC Workloads (2022)

(HT)Condor -> HTCSS

- Researcher places workloads to be handled at an Access Point (**AP**)
- Execution Point (**EP**) joins an HTCondor Pool
- **AP** can use the capacity of an **EP** to execute a job
- **AP** can delegate a job for execution by a remote batch system
- **AP + HTCondor Pool = HTCondor System**

CHTC is a Path Finder for Translational Computer Science – advancing the state of the art of distributed computing while actively providing services to Open Science Researchers.

CHTC staff is responsible for:

- The UW-Madison shared Research Computing capacity
- The Open Science Pool
- The PATh facility

The Open Science Pool (OSPool)

- An OSG service that provides a means for campus/organization to make (following autonomous policies) capacity available to the US open science community
- A HTCondor pool that serves as a fair-share source of High Throughput Capacity (**HTC**) to researchers
- Integrated with the OS Data Federation (OSDF)

OSPool last week numbers:

- **move** more than **34.0M** files
- **run** more then **2.3M** jobs
- **placed** by **84** users
- **from** **58** projects
- **at** more than **10** Access Points
- **on** more than **40K** cores
- **at** **49** sites
- **consume** more than **4.2M** core hours

Why do we need a CE?

To enable an external entity (Factory? AccessPoint? ...) to bring via a dynamically deployed Execution Point capacity managed by an autonomous batch system to a HTCondor pool.

- HTCondor pool defined by a Matchmaker
- HTCondor pool defined by an Access Point

What do we expect the CE to do?

- Transform a Capacity Acquisition Request into a GlideIn batch job
- Provide the input sandbox for the GlideIn job
- Handle the GlideIn job (submit, remove, monitor, account)
- Collect and report accounting information about the GlideIn job
- Move the output sandbox of the GlideIn job to the entity that issued the request
- (optional) monitor the EP launched by the GlideIn

Challenges

- Offer the CE as a (devOp) element of the HTCSS
- Reduce the effort required to deploy and configure a CE
- Reduce the effort required to maintain and trouble shoot a “hosted” CE
- Define a language/schema for Capacity Acquisition Requests (CapAR)

Supporting Bring Your Own Capacity (BYOC)

- Researcher can bring private capacity to the Access Point (AP)
- Establish affinity between workload (jobs) placed at the AP and the private capacity
- Namespace for private capacity
- Handling of private capacity
- How to formulate the GlideIn job?
- How to formulate the Capacity?

Thank You!



Follow us on Twitter!
<https://twitter.com/HTCondor>



PATH PARTNERSHIP to ADVANCE
**THROUGHPUT
COMPUTING**

This work is supported by [NSF](#) under Cooperative Agreement [OAC-2030508](#) as part of the [PATH Project](#). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.