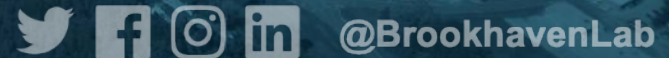




Parsl Integration in ATLAS

Tadashi Maeno (BNL)



12th Oct 2022
HEP-CCE All-Hands Meeting

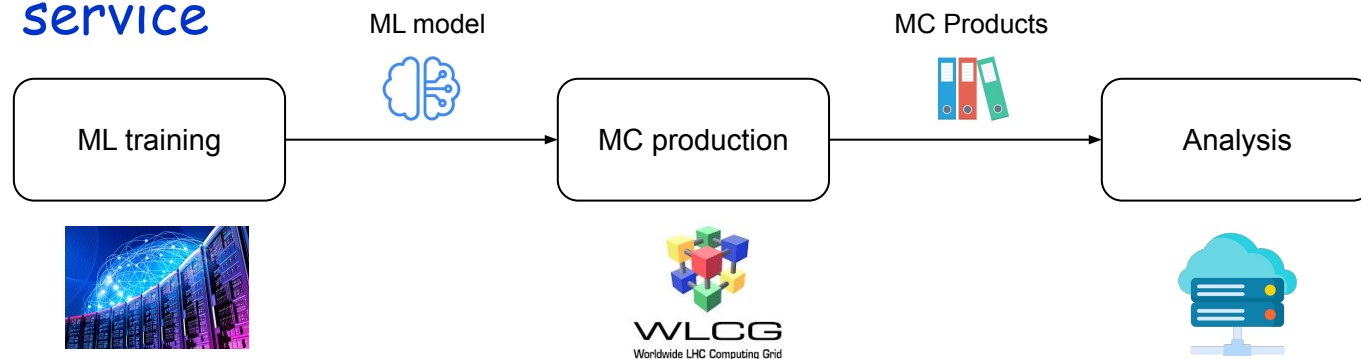
Needs for Distributed Heterogeneous Computing in ATLAS

- Distributed = Geographically distributed != Multi-node/process
- Most users love local resources, but have to go to remote providers when enough or suitable resources/services are locally unavailable
- A zoo of resource/service providers distributed worldwide with various benefits and constraints
 - The WLCG grid, commercial cloud resource/service providers, High-performance computing (HPC) and Leading Computing Facilities (LCFs), volunteer computing, Platform-as-a-Service (PaaS), Function-as-a-Service (FaaS), ...
- Complex and emerging workflows
 - Various resources/services even in a single workflow
 - To leverage an optimal provider for each part of the workflow



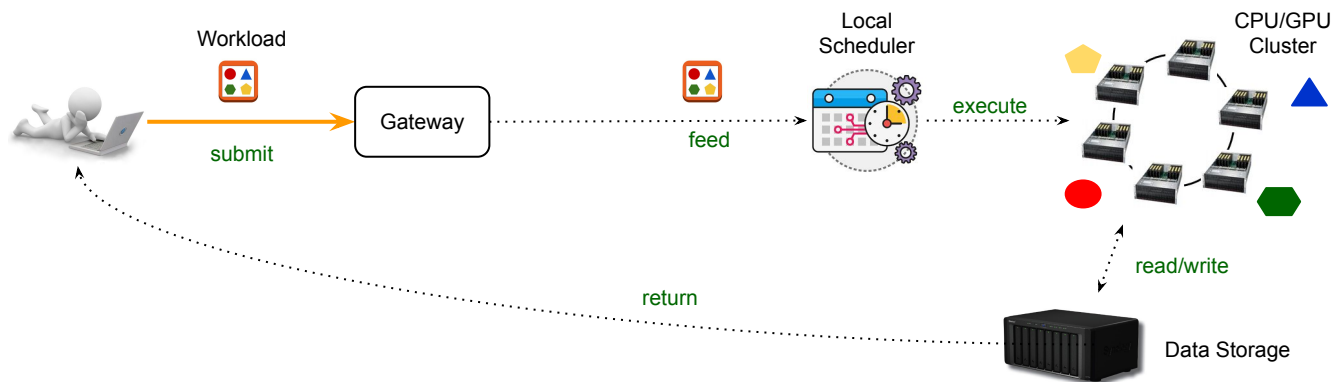
A Simple Usecase with Multiple Remote Providers

- A user has a workflow to perform physics analysis on Monte-Carlo (MC) samples produced with a Machine Learning (ML) model
 - Three tasks in the workflow
 - ML training, MC production, and Analysis
 - Each task could have different resource/service requirements
- The user happens to have
 - Allocation at a LCF where huge GPU resources are available
 - Approval from the experiment collaboration to use the grid
 - Credits for an analysis platform on a cloud service
- The user decided to run ML training at LCF, MC production on the grid, and Analysis on the cloud service
 - Three remote providers: LCF, the grid, and the cloud service



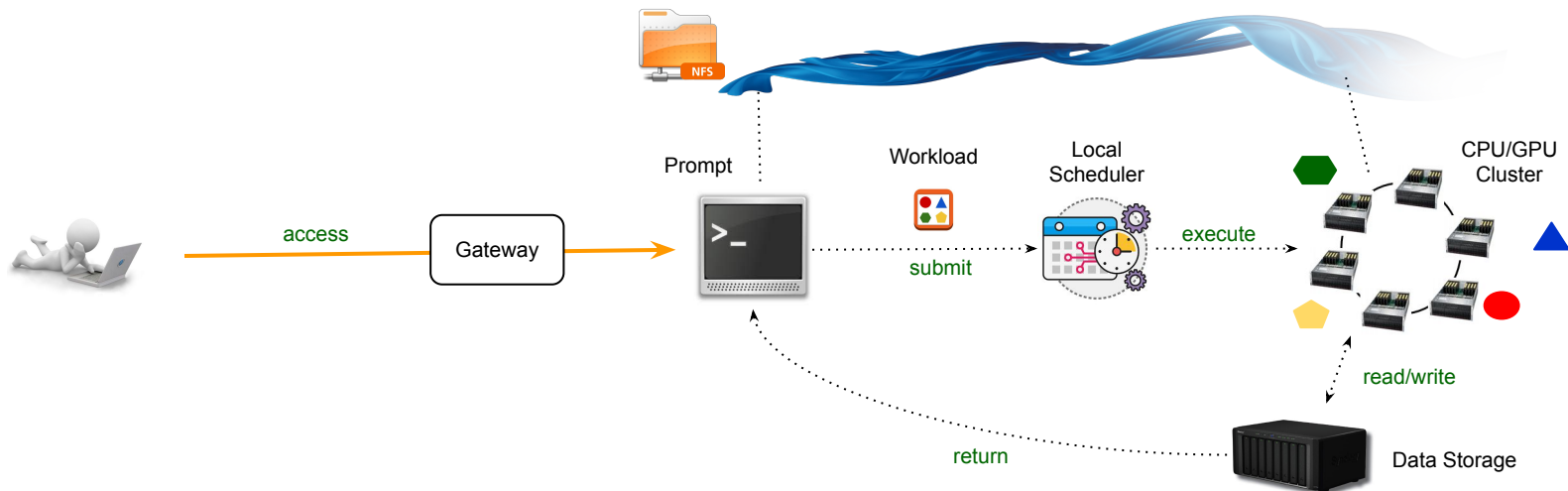
Integration of Remote Provider 1/3

- Traditional batch-like access to remote resources/services
 - The user describes a workload in a file
 - A remote gateway service authenticates the user to receive the file from the user
 - The gateway service feeds the file to a workload scheduler to process the workload on the computing resources behind
 - Outputs are delivered to the user somehow
- Gateway service: HTCondor Computing Element (CE), ARC CE, Kubernetes API service, ...



Integration of Remote Provider 2/3

- Traditional interactive access to remote resources/services
 - A remote gateway service authenticates the user to give an interactive prompt
 - The user submits a workload to a workload scheduler from the prompt
 - The user sees outputs in data storage from the prompt
- Gateway service: sshd, jupyter hub, AI platforms, ...

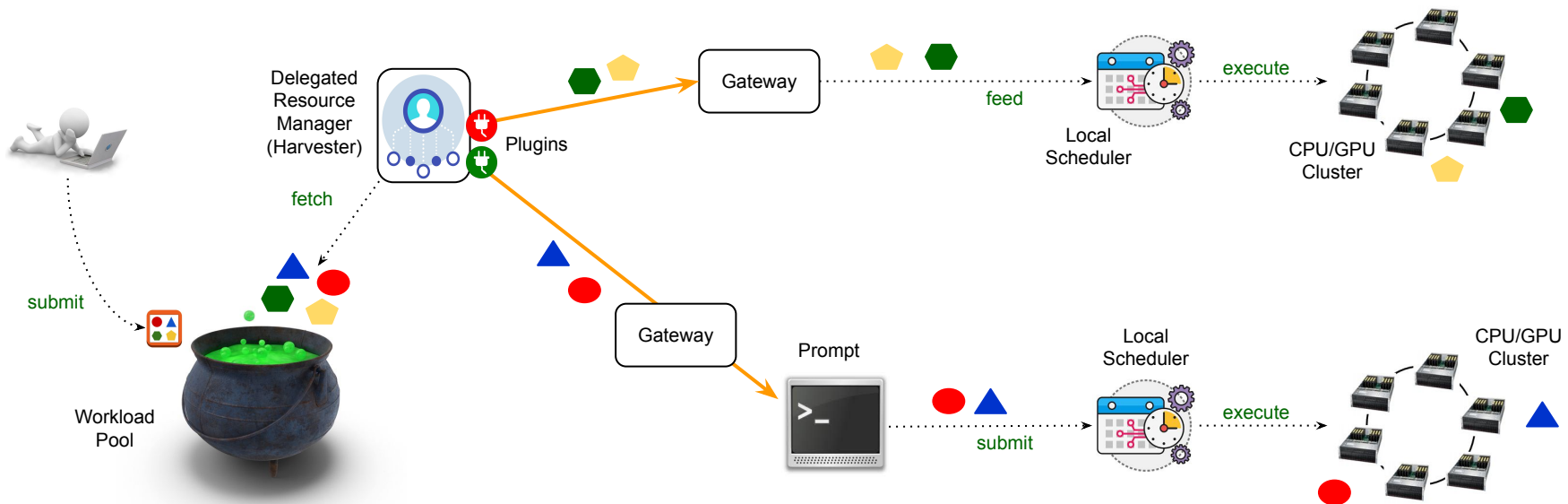


Integration of Remote Provider 3/3

- In ATLAS, central workload pool + delegated resource manager
 - The user submits workflows to the central workload pool where workflows are decomposed to smaller workload entities (tasks and jobs)
 - The delegated resource manager (Harvester) accesses to remote providers on behalf of users using common or user's credentials through plugins applicable for those providers
 - ~ One plugin for each provider

➤ Advantages

- Isolation between the user and remote providers
- Centrally managed fair-share and priorities among multiple users
- Workload routing based on fine-grained requirements and central knowledge of provider's characteristics and availabilities



Schedulers in Resource/Service Providers

- Traditional batch systems
 - HTCondor, Slurm, Pbs, Torque, ...
 - Many academic institutes including LCFs
- Kubernetes-based schedulers
 - Google Kubernetes Engine (GKE), Amazon Elastic Kubernetes Service (EKS), Azure Kubernetes Service (AKS), ...
 - De-facto standard available in many cloud services
 - The entire job or the entrypoint of the job to be containerized
 - Considerable cost difference between spot and on-demand instances
- Multi-node software
 - Dask, Horovod, Ray, ...
 - Application-level resource scheduling
- PaaS and FaaS
 - Google AI, Amazon ML service, REANA, funcX, ServiceX, ...
 - Platforms optimized for specific workloads
 - Very powerful for particular usecases
 - Not for all types of workloads, not straightforward to port existing workloads

Possible Integration of Parsl in ATLAS

- Parsl-based workload in a workflow
 - E.g. MC production → Parsl-based Analysis
- funcX as a gateway to allow users to run parsl-based workloads at Facility
- Harvester plugin to talk to funcX

