HEP-CCE Complex Workflows

Shantenu JHA, Vincent R. PASCUZZI, Mikhail TITOV, Matteo TURILLI

Brookhaven National Laboratory

Brookhaven National Laboratory

Motivation

- **Harmonize use of general/experiment-agnostic components** across DOE-funded HEP experiments (i.e., re-introduce communication interfaces).
  - Each experiment maintains *in-house/proprietary* workflow middleware solutions: tailored for specific use-case/infrastructure; cannot afford the risks in adopting alternative tools.
  - Difficult to foster ("sell", persuade) and integrate (technical aspects) new software into existing experiment software infrastructures. *N.B.* This is not limited to distributed computing!
- **Design new approaches for extreme-scale data processing** in conjunction with computing environment.
  - Distributed computing software for/by HEP is constantly evolving, but overhauls in infrastructure are constraints.
- Complex workflow effort originally targeting more workflow-rich cosmic experiments:
  - "Bursty" readout, and real-time data processing, transmission and storage.

Effort

- **Expand HTC capabilities with HPC** for particular use cases:
  - Manage workload and resource heterogeneity (spatial and temporal).
  - Support multi-node MPI tasks.
  - Enable software transition to a new platform.
- Focus on **integrated solutions** to support modularity and interoperability properties.
  - Collaboration with HEPCloud (https://computing.fnal.gov/hep-cloud/)
    - Serve roughly 10 HEP experiments, including some CMS workloads.
    - Workload offloading to both HPC and cloud resources.
    - Current system limitation: MPI tasks not supported.
  - RADICAL-Pilot integration into HEPCloud as an additional backend for workload execution on HPC resources to enable MPI at scale.
    - RADICAL-Pilot as a Service to bridge between different architectures and/or environments.

Collaboration

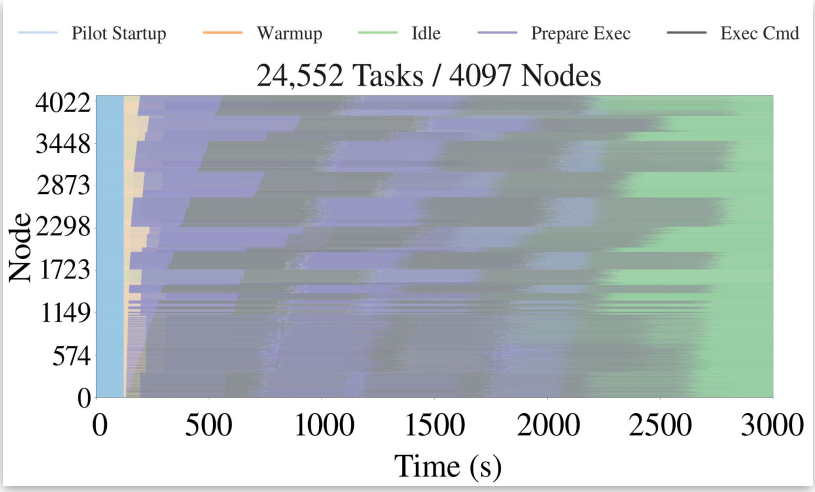
- Discussions initiated with HEP experiment distributed computing leads:
  - Understand use-cases, and current and future workload execution requirements.
- Existing experiments (*e.g.,* ATLAS, CMS):
  - Codes (primarily C++, Python) originated 10 years ago and not MPI- or GPU-based (*i.e.*, not “HPC codes”).
    - Continuous improvements made over the years (*e.g.*, MP, MT, adopting modern C++ and Python).
    - BUT: Refactor/isolate O(10^5)-O(10^6) SLoC for offloading is extremely costly.
    - Also concern from physics point-of-view: *implement*, *validate*, *verify* (rinse and repeat), *reproducibility.*
  - Each experiment maintains *in-house/proprietary solutions* for workflow management and execution.
  - *Consensus*: *Prefer to see demonstrated utility of new tools before considering adoption.*
- New/planned experiments (*e.g.*, DUNE):
  - Initial software development aiming to utilize HPC resources (multi-core, multi-node, (multi-)GPU).
  - Acknowledge advantages adopting pilot enabling scale-up/scale-out.
    - Much more inclined to adopt given longer time frames, few constraints.

Scientific highlights

- Early runs of adapted HEP frameworks on HPC production resources Theta & ThetaGPU (ALCF):
  - FastCaloSim GPU (https://github.com/cgleggett/FCS-GPU): adapted version of the ATLAS Fast Calorimeter Simulation framework.
  - HEPscore (https://gitlab.cern.ch/hep-benchmarks/hep-score): a benchmark based on containerized HEP workloads.
- Construction of workflows test models to evaluate management tools on various testbeds:
  - HPC testbeds: Arcticus (ALCF), Polaris (ALCF), Aurora (ALCF), Crusher (OLCF), Frontier (OLCF) and Perlmutter (NERSC).

Scientific highlights | cont.

- RADICAL-Cybertools (RCT) - high performance middleware for workflow/workload management
  - *Adaptability* to new platforms and updated environments, *Scalability*
  - Example of an approach for resource partitioning: using multiple Distributed Virtual Machines (multi-DVM) with PMIx/PRRTE, enabling use of up to 4K nodes on Summit for a heterogeneous workload
- Streamlining of RADICAL runtime system to facilitate integration with HEP party middleware, with approbation using hep-benchmarks (containerized HEP workloads, https://gitlab.cern.ch/hep-benchmarks)
- Proposed a design* for integration of RADICAL-Pilot into HEPCloud as part of the RCT component isolation effort (tentative title: *Pilot as a Service*)



RADICAL-Pilot application run using multi-DVM mode on Summit@OLCF (16 DVMs, each DVM managed tasks execution on up to 256 nodes)

(*) Leveraging past experience on integration with PanDA WMS to support ATLAS workflows

Summary

- **Exploration of HEP workloads** (and corresponding frameworks) and their test runs on HPC testbeds **is ongoing**.
- Capabilities of the [proposed] **execution management tools** are aligned with requested requirements and **further development is ongoing**.
- Proposed a design for **integration of execution management tools (RADICAL-Pilot)** as a backend for HEPCloud, a portal to an ecosystem of heterogeneous commercial and academic computing resources.