Azerbaijan Journal of High Performance Computing, Vol 6, Issue 2, 2023, pp. 203-208
https://doi.org/10.32010/26166127.2023.6.2.203.208

Difference Between OpenHPC and HTCondor Cluster Systems: In-depth Analysis

Elviz Ismayilov

Azerbaijan State Oil and Industry University, Baku, Azerbaijan, elviz.ismailov@asoiu.edu.az

*Correspondence: Elviz Ismayilov, Azerbaijan State Oil and Industry University, Baku, Azerbaijan, elviz.ismailov@asoiu.edu.az

Abstract

The rapidly developing field of high-performance computing (HPC) requires efficient and scalable solutions to manage extensive computing loads. Although they use different approaches and architectures, OpenHPC and HTCondor are well-known platforms that meet these needs. This article thoroughly analyzes OpenHPC and HTCondor to identify their fundamental differences and work paradigms. OpenHPC is a comprehensive modular structure designed to facilitate the deployment, management, and maintenance of HPC clusters, offering a rich set of pre-integrated HPC software components. Conversely, HTCondor specializes in efficiently planning and managing resource-intensive tasks, using a unique partner selection system for dynamic resource allocation based on job requirements and resource availability. By examining aspects such as system architecture, resource management efficiency, scalability, flexibility, and the user ecosystem, this analysis sheds light on the strengths and weaknesses of each structure. The research aims to provide stakeholders in high-performance computing with the knowledge necessary to make informed decisions regarding the selection and implementation of high-performance computing management systems, ultimately aimed at optimizing the use of computing resources and optimizing research and development workflows.

Keyword: OpenHPC, HTCondor, HPC Clusters, High Performance Computing.

1. Introduction

OpenHPC is a community collaboration that started around 2015-2016 with the intention of developing a comprehensive and flexible open-source environment for high-performance computing (HPC) (Schulz, K. W., et al., 2016). Its foundation is based on providing system administrators and users of HPC clusters with an integrated and hardware-independent set of software components and tools needed to configure and manage HPC clusters. The initiative aims to make the deployment and management of HPC systems more accessible and efficient, promoting collaboration and exchange of best practices in the community.

The OpenHPC project was officially launched in November 2015 during the Supercomputing Conference (SC15). It was introduced as an open-source platform for

high-performance computing environments with initial support from various industrial, academic, and research organizations. The founding members included hardware manufacturers such as Intel and IBM and educational institutions, demonstrating broad support for the project and intentional cooperation between various sectors.

The main goal of OpenHPC is to provide a stable and flexible set of HPC software components independent of a specific hardware architecture. This includes tools for system administration, resource management, I/O services, development tools, and various scientific libraries. Thus, OpenHPC aims to:

• Reduce the complexity of building and managing the software stack of the HPC system.

• Promote collaboration and exchange of best practices and tools in the high-performance computing community.

• Provide the basis for creating a reproducible HPC software stack.

• Encourage the use of open standards and the compatibility of HPC systems.

Since its inception, OpenHPC has expanded to include a wide range of software components, from provisioning tools and resource managers to compilers, libraries, and development tools (Simmons, C., Schulz, K., & Simmel, D., 2020, July). He has significantly contributed to simplifying the deployment and management of HPC clusters, providing a valuable resource for experienced system administrators and newcomers to HPC. The impact of OpenHPC goes beyond simplification and efficiency improvement; it also promotes the adoption of high-performance computing technologies in various industries and research areas, reducing the entry barrier.

HTCondor is a specialized workload management system for computationally intensive tasks, also known as a batch scheduling system. It is designed to maximize computing resources by distributing high-performance tasks across available computing resources. This makes HTCondor particularly well-suited for environments that must manage large computing jobs or require detailed scheduling policies to allocate resources efficiently. The development and evolution of HTCondor highlight its importance in distributed computing and its contribution to scientific research and solving complex computational problems (Orejuela, V., Ramirez, Á. S., Toro, A. F., Gonzalez, A. F., & Briñez, D., 2018).

HTCondor development began in the late 1980s at the University of Wisconsin-Madison. It was initially conceived as a " Condor " project by Myron Livny, a professor at the university's Computer Science department. The project aimed to use unused CPU cycles of network workstations to perform resource-intensive tasks. This concept was innovative for its time and sought to create a "cycle absorber" that could efficiently use idle computing resources on the network.

• High-performance Computing: HTCondor was explicitly designed to support high-performance computing (HTC). Unlike high-performance computing (HPC), which focuses on completing a single task as quickly as possible, HTC focuses on efficiently completing many loosely coupled tasks over long periods.

• The candidate selection system. One of the main functions of HTCondor is a candidate

selection system that combines submitted assignments with the most appropriate computing resources based on a set of requirements specified by both the assignment and the resource owner.

• Checkpoints: HTCondor supports checkpoints for specific jobs, allowing you to pause and resume calculations. This feature is crucial for using energy-intensive computing resources, such as desktop workstations, which may not always be available.

DAGMan: Directed Acyclic Graph Manager (DAGMan) is a workflow management tool for HTCondor that allows users to define dependencies between jobs, providing orchestration of complex workflows (Orejuela, V., Ramirez, Á. S., Toro, A. F., Gonzalez, A. F., & Briñez, D., 2018).

HTCondor was developed by both the user community and the core development team at the University of Wisconsin-Madison. The project strongly emphasizes the principles of open-source code, allowing it to be freely used, modified, and distributed. This approach has created a large and active community of users and participants from academia, government, and industry.

HTCondor's influence is widespread in various fields requiring large-scale computational efforts, including high-energy physics, climate modeling, and bioinformatics. The flexibility and efficiency of different computing resources have made it an essential tool for researchers and organizations worldwide.

2. Related Work

OpenHPC and HTCondor are two different software products designed to work with high-performance computing systems (HPC), but they serve other purposes and are used in different contexts.

OpenHPC is an extensive software suite designed to simplify the deployment and management of HPC systems. It includes many tools and libraries for configuring the HPC environment, including cluster management, working with file systems, task planning, and monitoring. OpenHPC offers a modular architecture that allows users to select the most suitable components for their specific requirements.

HTCondor, on the other hand, is a distributed computing system optimized for managing high-performance computing tasks on many distributed resources. It allows you to organize computing resources into a pool, automatically spreading tasks between available nodes to maximize resource usage. HTCondor is especially suitable for tasks that require a large amount of computing and can effectively manage functions of varying complexity, including job queues, priorities, and dependencies between tasks.

• Purpose: OpenHPC provides tools and libraries to simplify the deployment and management of HPC systems, whereas HTCondor focuses on distributing and managing computing tasks in large distributed networks.

• Modularity: OpenHPC offers a modular structure that allows you to customize the HPC environment according to specific needs, while HTCondor provides efficient task allocation and management mechanisms.

• Application: HTCondor is ideal for projects requiring complex calculations on many independent nodes, while OpenHPC is focused on facilitating the deployment and management of the HPC infrastructure.

These tools can be used together within a single HPC system, where OpenHPC provides the basis for infrastructure deployment and management. HTCondor optimizes the distribution and execution of computing tasks.

Of course, the question arises here when it is right to choose which cluster. Here, you need to select each one according to its problems. OpenHPC offers a wide range of tools and libraries for deploying, managing, and supporting your HPC systems if you need a comprehensive platform for managing your HPC infrastructure. This is a good choice if you are building or upgrading an HPC cluster and need extensive tools for resource management, network, data storage, and security.

OpenHPC allows you to select specific components best suited to your system and requirements.

But For Distributed Computing Tasks: HTCondor is optimized for managing distributed computing tasks. If your project requires complex calculations that can be distributed across multiple nodes, HTCondor will offer an effective solution for task allocation, resource management, and job queues.

• To maximize resource usage: HTCondor is specifically designed to efficiently use available computing resources, automatically allocating tasks based on node availability and set priorities.

Sharing: In some cases, using OpenHPC for infrastructure management and HTCondor to optimize the distribution of computing tasks may be the best solution. This allows you to combine the advantages of both tools, providing a highly efficient and flexible HPC environment.

Architectural differences between OpenHPC and HTCondor: The architectural differences between OpenHPC and HTCondor reflect their different goals and approaches to managing high-performance computing (HPC) resources. While OpenHPC provides an integrated software stack for building and managing HPC clusters, HTCondor focuses on distributed computing and job optimization. Let's look at the key architectural differences:

• Integrated Software Stack: OpenHPC is a set of tools and libraries designed to simplify the deployment and management of HPC clusters. This includes task schedulers, monitoring tools, development libraries, and more, all of which can be configured to work in a single system.

• Modularity: The OpenHPC architecture is modular, allowing users to select and customize components according to their specific requirements. This makes it easier to scale and adapt the cluster to particular tasks.

• Standardization: OpenHPC strives to standardize installation and configuration processes to facilitate compatibility and interaction between the various components of the HPC system.

• Specialized Job Management System: HTCondor is explicitly designed to distribute

and manage computing jobs efficiently in large-scale distributed systems. It uses a unique queue and priority mechanism to optimize task execution.

• Dynamic Resource allocation: HTCondor can dynamically allocate jobs based on resource availability, job priorities, and usage policies. This ensures flexible and efficient use of computing resources.

• High scalability and flexibility: The HTCondor architecture is designed to work in scalable and heterogeneous computing environments, allowing it to manage jobs in various system configurations efficiently.

• Purpose and Focus: OpenHPC is designed to simplify the deployment and management of HPC clusters, while HTCondor focuses on optimizing the execution of specific computing tasks in distributed environments (Gavrilovska, A. et al., 2007, March).

• Component versus Specialization: OpenHPC offers a wide range of integrated components for various aspects of HPC cluster management, whereas HTCondor is a specialized system for job management (Diwan, S. M., 1999; Bockelman, B., Livny, M., Lin, B., & Prelz, F., 2021).

• Modularity and Standardization versus Flexibility and Scalability: OpenHPC facilitates the creation of standardized and modular HPC clusters, whereas HTCondor offers flexibility and scalability to optimize computational tasks in a wide range of computing environments.

Conslusion

The choice between OpenHPC and HTCondor depends on the specifics of your tasks, performance requirements, and available resources. It is essential to carefully evaluate both tools in the context of your goals and preferences.

OpenHPC is an integrated open-source software suite designed to simplify deploying and managing high-performance computing (HPC) clusters. This project provides users of the HPC community with a comprehensive solution that includes all the necessary tools for creating and maintaining HPC systems (Yang, Y., 2007).

OpenHPC facilitates the creation and management of HPC clusters by providing tools and resources that help researchers, engineers, and developers maximize the productivity and efficiency of their computing tasks.

HTCondor (formerly known as Condor) is a specialized job management system for computing clusters and distributed computing environments. It is designed to distribute and optimize computational tasks among available resources efficiently.

HTCondor is a powerful tool for scientific research and engineering projects that require complex calculations and efficient resource management in distributed computing environments.

The choice between OpenHPC and HTCondor (or a combination of them) depends on the project's specific requirements, including the computing scale, the types of tasks to be performed, and the available infrastructure (Hollowell, C., Barnett, J., Caramarcu, C., Strecker-Kellogg, W., Wong, A., & Zaytsev, A., 2017, October).

• The best in versatility and infrastructure management: OpenHPC.

• The best in distributed computing and efficiency: HTCondor.

Reference

Schulz, K. W., et al. (2016, November). Cluster computing with OpenHPC. In Inaugural HPC Systems Professionals Workshop, HPCSYSPROS16 (pp. 1-6).

Simmons, C., Schulz, K., & Simmel, D. (2020, July). Customizing OpenHPC: integrating additional software and provisioning new services, including open on-demand. In Proceedings of the Conference on Practice and Experience in Advanced Research Computing (p. 1).

Orejuela, V., Ramirez, Á. S., Toro, A. F., Gonzalez, A. F., & Briñez, D. (2018). Application for computational cluster performance tests configured in HTCONDOR. In MATEC Web of Conferences (Vol. 210, p. 04029). EDP Sciences.

Kalayci, S., Dasgupta, G., Fong, L., Ezenwoye, O., & Sadjadi, S. M. (2010, December). Distributed and Adaptive Execution of Condor DAGMan Workflows. In SEKE (pp. 587-590).

Hollowell, C., Barnett, J., Caramarcu, C., Strecker-Kellogg, W., Wong, A., & Zaytsev, A. (2017, October). Mixing HTC and HPC workloads with HTCondor and slurm. In Journal of Physics: Conference Series (Vol. 898, No. 8, p. 082014). IOP Publishing.

Gavrilovska, A., et al. (2007, March). High-performance hypervisor architectures: Virtualization in hpc systems. In Workshop on system-level virtualization for HPC (HPCVirt).

Diwan, S. M. (1999). Open HPC++: An open programming environment for high-performance distributed applications. Indiana University.

Bockelman, B., Livny, M., Lin, B., & Prelz, F. (2021). Principles, technologies, and time: The translational journey of the HTCondor-CE. Journal of Computational Science, 52, 101213.

Yang, Y. (2007). *A fault tolerance protocol for stateless parallel processing.* Temple University. [Ph.D thesis, AAI3293270].

Submitted 06.10.2023
Accepted 24.11.2023