Current and Future Converged Cloud-HPC Workflows at LLNL

Daniel J. Milroy*, Stephen Herbein*, Dong H. Ahn*
*Lawrence Livermore National Laboratory, Livermore, CA, USA
{milroy1, herbein1, ahn1}@llnl.gov

***Abstract*—Current and emerging scientific workflows at the Lawrence Livermore National Laboratory (LLNL) require the integration of cloud technologies with traditional HPC to make discoveries. In this talk, we present prominent workflow examples, trends in these converged workflows, and gaps that they face at one of the world's largest computing centers. Based on application examples, we will describe successful workflow patterns that make use of loose convergence between HPC clusters and on-premises container orchestration clusters. While the converged approach is making significant strides, we still find critical gaps such as lack of integration with Resource and Job Management Software, keeping it from realizing its full potential. We will discuss how LLNL is co-designing our critical software infrastructure with workflow teams, the computing facility, and industry partners. Finally, we will highlight some of the key techniques we use to address outstanding challenges in resource expression and scheduling in a converged environment.**

***Index Terms*—HPC-cloud convergence, RJMS, Generalized Multi-Level Scheduling**

I. INTRODUCTION

The Lawrence Livermore National Laboratory (LLNL) has been a global leader in HPC since its inception. Workflows that integrate HPC and cloud technologies are becoming common at LLNL and in computational science in general. We perform a survey of current, converged workflow trends at LLNL to understand how computational scientists make use of emerging capabilities.

II. BACKGROUND

As complex, multi-stage and pipelined workflows become more common, computational scientists increasingly rely on cloud technologies together with HPC to perform workflow steps. These so-called converged workflows will soon become more prevalent as cloud technologies advance to accommodate higher performance workloads.

LLNL has a formidable array of traditional HPC resources and is witnessing the trend toward convergent workflows. Accordingly, the Laboratory is investing in on-premises container orchestration frameworks to facilitate science that depends on both cloud and HPC technologies. To better understand current trends in computational science at LLNL, we perform a survey of current and emerging converged workflows.

III. A SURVEY OF CONVERGED WORKFLOWS AT LLNL

There are several well-documented application frameworks that are a product of LLNL and collaborators. Perhaps the most urgent are those that participate in the global efforts to discover treatments for COVID-19. The ATOM Modeling Pipeline [1] is one such workflow, and its complexity translates to a composite architecture that employs multiple HPC and cloud technologies. The workflow is an example of so-called "loose coupling" convergence, meaning that modules or segments of the workflow are executed within one environment or the other.

Through a survey of applications using a newly-available on-premises container orchestration framework, we observe that LLNL applications currently exhibit loose convergence. Furthermore, we determine that many workflows use orchestrated databases and visualization which communicate data with HPC resources. Figure 1 provides a diagrammatic depiction of cloud dependencies for several LLNL workflows. AMPL is noteworthy in its use of many cloud technologies. From similarities in workflow patterns, we conclude that loose coupling will be a common convergence pattern for the present and near future.

IV. RESOURCE AND JOB MANAGEMENT SOFTWARE ROLE IN CONVERGENCE

Current-generation Resource and Job Management Software (RJMS) such as Slurm [8], LSF [9], and PBS Pro [10] all support the popular convergence method of cloud bursting. However, they require that cloud resources be defined ahead of time, and are limited on how to relinquish resources. The limitations are due to the systems' use of rigid, simplistic models to express resources.

The complexity of converged workflows and their resource requirements (which may change in time) presents a significant challenge for current Resource and Job Management Software (RJMS). Flux [11] is an RJMS created at LLNL to address limitations of current RJMS and facilitate exascale computing. Unlike many current RJMS, Flux does not use a simplistic representation of resources such as bitmaps representing the scheduling state of compute nodes, which gives it more flexibility to express complex resources. Such freedom of resource expression is necessary to capture the complexity of converged workflow resources. We observe that many converged workflows at LLNL make use of Flux. Enabling cloud bursting and more general general elasticity in Flux is in

---

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 (LLNL-ABS-813921).

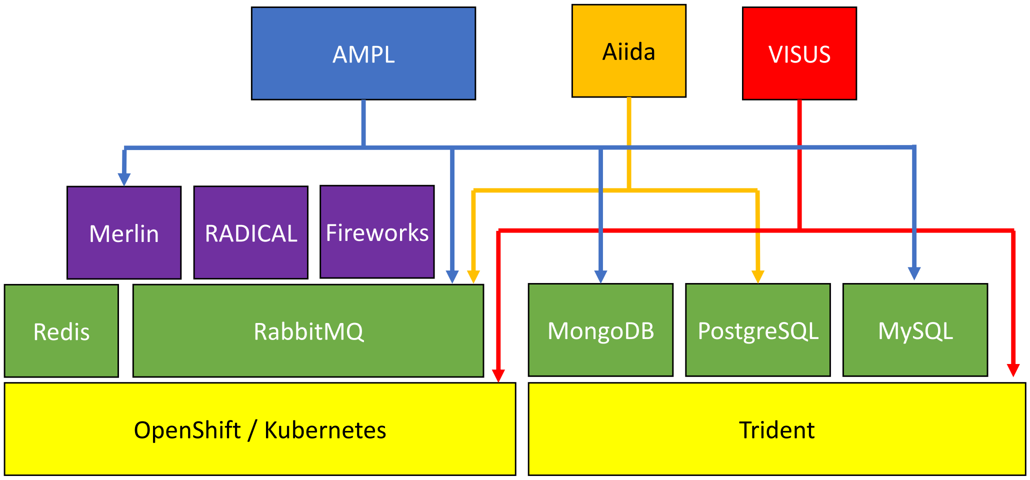

Fig. 1. LLNL's on-premises cloud software stack. OpenShift orchestrates the deployment of both RabbitMQ and Redis; on top of which the Merlin [2], RADICAL [3], and Fireworks [4] workflow systems run. Trident [5] orchestrates storage and persistent databases like MongoDB and PostreSQL. The AMPL [1], AiiDA [6], and ViSUS [7] cloud software use a combination of the services hosted on OpenShift and Trident.

the experimental phase, but will allow for much more dynamic bursting.

V. IMPLICATIONS FOR THE FUTURE OF CONVERGENCE

We identify three common challenges for converged workflows. Autoscaling, where an application (orchestrated or otherwise) can grow its allocation to consume more resources, requires an RJMS capable of resource elasticity. A plug-and-play, component-based ecosystem facilitates decoupling of the workflow specification from the computing environment. Finally, security issues can be complex and difficult to identify in an orchestrated environment.

Autoscaling is represented by two convergence patterns. While the loose coupling pattern is common, it is unlikely to exhibit as high performance as a pattern like "cloud-in-HPC". In such a pattern, HPC resources host orchestrated portions of a complex workflow together with stages that use traditional HPC. Cloud-in-HPC addresses the constraint of hosting a dedicated orchestration cluster by allowing the orchestrator to run on a much larger HPC system. This approach could minimize communication and data transfer time as stages can be co-located on the same physical hosts, or at least within the same cluster.

LLNL is also pursuing the reverse pattern, i.e. "HPC-in-cloud", as it will allow current workloads to run on a wider variety of systems. AMPL is one such workflow, and its developers are interested in allowing it to run efficiently contained entirely within the cloud. HPC-in-cloud requires that the RJMS job allocation grow and shrink depending on the provisioned cloud resources.

A recent Exascale Computing Project known as ExaWorks was launched to create a plug-and-play workflow ecosystem at LLNL. ExaWorks is tasked with the co-design of vertical and horizontal APIs to interface between workflows, workflow representations, task orchestration, resource abstraction, and RJMS. It will also harden the ExaWorks toolbox implementation for deployment on exascale systems.

Finally, security is a key consideration for orchestration. ViSUS allows for chaining of orchestrated servers to provide seamless access to federated archival data. The data may reside on local, parallel filesystems or a public cloud object store. Authentication mechanisms may be different for each source, and have credentials that violate policies where the orchestration system resides. Furthermore, containerized applications may use stacks with severe CVEs. Understanding how to balance the potential convenience of orchestration with security concerns is crucial for effective convergence.

REFERENCES

[1] A. J. Minnich, K. McLoughlin, M. Tse, J. Deng, A. Weber, N. Murad, B. D. Madej, B. Ramsundar, T. Rush, S. Calad-Thomson, J. Brase, and J. E. Allen, "AMPL: A Data-Driven Modeling Pipeline for Drug Discovery," *Journal of Chemical Information and Modeling*, vol. 60, no. 4, pp. 1955–1968, 2020.
[2] Lawrence Livermore National Laboratory, "Merlin," https://github.com/LLNL/merlin, 2019.
[3] M. Turilli, V. Balasubramanian, A. Merzky, I. Paraskevakos, and S. Jha, "Middleware building blocks for workflow systems," *Computing in Science Engineering*, vol. 21, no. 4, pp. 62–75, 2019.
[4] A. Jain, S. P. Ong, W. Chen, B. Medasani, X. Qu, M. Kocher, M. Brafman, G. Petretto, G.-M. Rignanese, G. Hautier, D. Gunter, and K. A. Persson, "Fireworks: a dynamic workflow system designed for high-throughput applications," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 17, pp. 5037–5059, 2015, cPE-14-0307.R2. [Online]. Available: http://dx.doi.org/10.1002/cpe.3505
[5] NetApp, "Trident," https://github.com/netapp/trident, 2020.
[6] G. Pizzi, A. Cepellotti, R. Sabatini, N. Marzari, and B. Kozinsky, "Aiida: automated interactive infrastructure and database for computational science," *Computational Materials Science*, vol. 111, pp. 218 – 230, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0927025615005820
[7] ViSUS.org, "Openvisus," https://github.com/sci-visus/OpenVisus, 2010.
[8] A. B. Yoo, M. A. Jette, and M. Grondona, "SLURM: Simple Linux Utility for Resource Management," in *Job Scheduling Strategies for Parallel Processing*, D. Feitelson, L. Rudolph, and U. Schwiegelshohn, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, p. 44–60.

[9] "IBM Spectrum LSF resource connector overview," https://www.ibm.com/support/knowledgecenter/SSWRJV_10.1.0/, 2020, retrieved May 23, 2020.

[10] *PBS Professional 18.2 Administrator's Guide*, https://www.altair.com/pdfs/pbsworks/PBSAdminGuide18.2.3.pdf, Altair Engineering Inc., December 2018.

[11] D. H. Ahn, N. Bass, A. Chu, J. Garlick, M. Grondona, S. Herbein, J. Koning, T. Patki, T. R. W. Scogland, B. Springmeyer, and M. Taufer, "Flux: Overcoming Scheduling Challenges for Exascale Workflows," in *2018 IEEE/ACM Workflows in Support of Large-Scale Science (WORKS)*, 2018, pp. 10–19.