Container Orchestration for Scientific Workflows

Wolfgang Gerlach[1,2]
[1]University of Chicago
Chicago, USA
e-mail: wgerlach@mcs.anl.gov

Wei Tang[2], Andreas Wilke[2],
Dan Olson[2,1] and Folker Meyer[2,1]
[2]Argonne National Laboratory
Argonne, USA

Recently, Linux container technology has been gaining attention as it promises to transform the way software is developed and deployed. The portability and ease of deployment makes Linux containers an ideal technology to be used in scientific workflow platforms. AWE/Shock is a scalable data analysis platform designed to execute data intensive scientific workflows. Recently we introduced *Skyport*, an extension to AWE/Shock, that uses Docker container technology to orchestrate and automate the deployment of individual workflow tasks onto the worker machines. The installation of software in independent execution environments for each task reduces complexity and offers an elegant solution to installation problems such as library version conflicts. The systematic use of isolated execution environments for workflow tasks also offers a convenient and simple mechanism to reproduce scientific results.

I. AWE/SHOCK PLATFORM

The core component of our AWE/Shock [1] platform is Shock, a data management system designed as an object storage system, conceptually similar to the cloud storage system Amazon S3. The advantages of using such an object storage system are scalability, no dependency on shared filesystems, and implicit support for deployments spanning multiple clouds. The Shock API can be used to store, query, and retrieve data and metadata (e.g., scientific metadata).

The AWE server is a resource manager and job scheduler. The server takes a workflow description document as input and creates smaller work units that can be checked out by AWE workers. The workflows are modeled as directed acyclic graphs to describe the data dependencies between individual tasks within the workflows. AWE workers check work units out from the server and download all required input files and databases from Shock. After processing workloads, the AWE workers upload the results to Shock.

Skyport [2] is an extension to the AWE/Shock platform. Software needed to execute tasks is installed in Docker images that are stored in Shock. When an AWE worker checks out a new work unit, it downloads the Docker image (unless its already cached locally) in addition to the input files, and spawns a container from this image which then executes the task. When the task has finished, the container is deleted and output files are uploaded to Shock.

The integration of Docker into the AWE/Shock platform increases flexibility and can improve overall resource usage. Each AWE worker can check out any task and is only limited by the hardware requirements of individual work units. The use of isolated environments greatly simplifies installation of software for each task. Storing the tasks as Docker images in Shock makes it possible to rerun workflows without the need of complicated or unreliable software installation procedures, and thus ensures a very high level of scientific reproducibility. The AWE/Shock data analysis system is open source and has been written in Go using the REST architectural style. All components support Simple Auth and OAuth.

II. FUTURE DIRECTIONS

A. Multi-cloud optimization

Because all data connections to Shock are based on HTTP requests, connections from clients (i.e., AWE workers) to the Shock server are usually non-problematic and can be established between different administrative domains, which allows AWE to operate in multi-cloud mode (also known as "hybrid-cloud"). However, since the connection between clouds poses an I/O bottleneck and may induce data transfer costs on commercial cloud providers, it is critical to improve I/O efficiency by use of a Shock cache on remote sites running AWE workers. To support the development of such a cache for remote sites running AWE workers, we plan to evaluate and compare performance gains in different scenarios, in particular involving I/O-dominated workflows.

B. Linux containers on bare-metal

Skyport exploits Docker containers to achieve software isolation. For hardware isolation (i.e., to split up a multi-core machine into smaller units), AWE workers have to run on virtual machines. Compared to processes running on a bare metal operating system, processes running in Docker containers are significantly less impacted by performance loss than processes running in virtual machines [3]. Linux containers can also be used to isolate hardware and thus have the potential to replace virtual machines in certain situations. We plan to evaluate and compare the overall performance of Skyport by running AWE workers with Docker on bare metal (e.g. using a cloud testbed like NSFCloud) and on virtual machines. We expect that different workflows and

workflow tasks have different runtime characteristics and we want evaluate the potential performance improvement by avoiding overhead induced by virtual machines. For this we can use the performance (CPU and memory) monitoring API of AWE. We also plan on using other various systems performance tools (i.e., collectl, sar, systemtap), not only within the virtual machines and containers, but also on the hypervisors themselves, as this will provide a unique window into the performance dynamics.

ACKNOWLEDGMENT

This work was supported in part by the NIH award U01HG006537 "OSDF: Support infrastructure for NextGen sequence storage, analysis, and management", and U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research DE-AC02-06CH11357 as part of Resource Aware Intelligent Network Services (RAINS) and as part the Office of Science, Office of Biological and Environmental Research Systems Biology Knowledgebase (KBase).

REFERENCES

[1] W. Tang, J. Wilkening, N. Desai, W. Gerlach, A. Wilke, and F. Meyer. "A scalable data analysis platform for metagenomics," 2013 IEEE International Conference on Big Data, pp. 21–26. IEEE, 2013.

[2] W. Gerlach, W. Tang, KP. Keegan, T. Harrison, A. Wilke, J. Bischof, M. D'Souza, S. Devoid, D. Murphy-Olson, NL. Desai and F. Meyer. "Skyport – Container-Based Execution Environment Management for Multi-Cloud Scientific Workflows," Proceedings of the 5th International Workshop on Data Intensive Computing in the Clouds, 2014.

[3] W. Felter, A. Ferreira, R. Rajamony, and J. Rubio. "An updated performance comparison of virtual machines and linux containers," Technology, 28:32, 2014.