Jun 27th, 9:00 AM - 10:20 AM

Building Containerized Environmental Models Using Continuous Integration with Jenkins and Kubernetes

Kyle Traff  
*Colorado State University, kyle.traff@colostate.edu*

Tyler Wible  
*Colorado State University - Fort Collins, tyler.wible@colostate.edu*

Lucas Yaege  
*Colorado State University - Fort Collins, lucasy@gmail.com*

Olaf David  
*Colorado State University - Fort Collins, olaf.david@colostate.edu*

Jack Carlson  
*Colorado State University - Fort Collins, jack.carlson@colostate.edu*

*See next page for additional authors*

Follow this and additional works at: [https://scholarsarchive.byu.edu/iemssconference](https://scholarsarchive.byu.edu/iemssconference)

[https://scholarsarchive.byu.edu/iemssconference/2018/Stream-A/2](https://scholarsarchive.byu.edu/iemssconference/2018/Stream-A/2)

This Oral Presentation (in session) is brought to you for free and open access by the Civil and Environmental Engineering at BYU ScholarsArchive. It has been accepted for inclusion in International Congress on Environmental Modelling and Software by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.
Building Containerized Environmental Models Using Continuous Integration with Jenkins and Kubernetes

Kyle Traff, Tyler Wible, Lucas Yaege, Olaf David, Jack Carlson, Mazdak Arabi
Colorado State University

Abstract: Environmental models typically consume vast amounts of computing resources. To effectively serve a growing community of physical, social and natural scientists, these models must be able to scale dynamically and horizontally to meet the demand. Models also require a vast array of software libraries, runtimes, compilers, and configurations specific to a particular application. Maintaining arrays of physical servers, each configured for one specific application, is expensive and inefficient to build and maintain. With the advent of software containers, model developers can isolate an application and all of its software dependencies from the physical server. Kubernetes, a container orchestration tool built by Google, has made it possible to dynamically deploy these containers seamlessly across a cluster of machines. We introduce key concepts and tools for building distributed modeling systems with containers using Kubernetes, managed with a continuous integration pipeline built in Jenkins. We then build and deploy a suite of comprehensive flow analysis (CFA) models as microservices. Finally, we test the service responsiveness, throughput, and average execution time of various containerized configurations of CFA models against deployment on virtual and bare-metal machines.

Keywords: Docker; Kubernetes; Jenkins; Continuous Integration; Distributed Systems