iEMSs International Congress on Environmental Modelling and Software

9th International Congress on Environmental Modelling and Software
Fort Collins, Colorado, USA, Mazdak Arabi, Olaf David, Jack Carlson, Daniel P. Ames (Eds.)
https://scholarsarchive.byu.edu/iemssconference/2018/

Going Serverless: Evaluating the Potential of Serverless Computing for Environmental Modelling Application Hosting

Baojia Zhang[1], Wes Lloyd[1], Olaf David[2], George Leavesley[2]
[1]Institute of Technology, University of Washington, Tacoma, Washington; [2]Object Modelling System Laboratory, One Water Solutions Institute, Department of Civil and Environmental Engineering, Colorado State University, Fort Collins, Colorado (wlloyd@uw.edu)

Abstract: Recently serverless computing platforms have emerged that provide automatic web service hosting in the cloud. These platforms are promoted for their ability to host "micro" services to end users while seamlessly integrating key features including 24/7 high availability, fault tolerance, and automatic scaling of resources to meet user demand. Serverless Computing environments abstract the majority of infrastructure management tasks including VM/container creation, and load balancer configuration. A key benefit of serverless computing is FREE access to cloud computing resources. Many platforms provide free access for up to 1,000,000 service requests/month with 1 GB of memory for 400,000 seconds/month. Additionally, serverless platforms support programming languages common for modeler developers including Java, Python, C#, and Javascript.

We present results from a proof-of-concept deployment of a Java based implementation of a 2008 version of the Precipitation-Runoff Modelling System (PRMS). PRMS is a deterministic, distributed-parameter model developed to evaluate the impact of various combinations of precipitation, climate, and land use on stream flow and general basin hydrology (Leavesley et al., 1983). The Java based version of PRMS, implemented using the Object Modelling System (OMS) 3.0 component-based modelling framework, was deployed to the Amazon AWS Lambda serverless computing platform. This version of PRMS consists of approximately ~11,000 lines of code and easily fits within the 256 MB maximum code size constraint of AWS Lambda. We report our performance observations based on varying the memory reservation size on AWS Lambda for PRMS deployments. Additionally, we assess implications of infrastructure initialization (COLD vs. WARM performance) on performance and assess the platform's ability to dynamically scale infrastructure to support many concurrent user modelling requests. We compare our performance vs. deploying the application to Docker application containers hosted by Amazon EC2 virtual machine (VM) instances. We contrast average model execution time, service throughput (requests/minute), as well as the cloud hosting costs of PRMS using these deployment alternatives.

Keywords: Serverless Computing, Function-as-a-Service, microservices.