



Jul 12th, 3:50 PM - 4:10 PM

Scaling Watershed Models: Modern Approaches to Science Computation with MapReduce, Parallelization, and Cloud Optimization

Jonathan Flaishans

Oak Ridge Institute for Science and Education, flaishans.jonathan@epa.gov

Meridith Fry

U.S. Environmental Protection Agency Office of Pesticide Programs, fry.meridith@epa.gov

Trip Hook

U.S. Environmental Protection Agency Office of Pesticide Programs, hook.james@epa.gov

Nelson Thurman

U.S. Environmental Protection Agency Office of Pesticide Programs, thurman.nelson@epa.gov

Jim Carleton

U.S. Environmental Protection Agency Office of Pesticide Programs, carleton.jim@epa.gov

Follow this and additional works at: <https://scholarsarchive.byu.edu/iemssconference>

 Part of the [Civil Engineering Commons](#), [Data Storage Systems Commons](#), [Environmental Engineering Commons](#), [Hydraulic Engineering Commons](#), and the [Other Civil and Environmental Engineering Commons](#)

Flaishans, Jonathan; Fry, Meridith; Hook, Trip; Thurman, Nelson; Carleton, Jim; Thawley, Shelly; Wolfe, Kurt; Young, Dirk; and Purucker, Tom, "Scaling Watershed Models: Modern Approaches to Science Computation with MapReduce, Parallelization, and Cloud Optimization" (2016). *International Congress on Environmental Modelling and Software*. 33.
<https://scholarsarchive.byu.edu/iemssconference/2016/Stream-A/33>

This Event 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.

Presenter/Author Information

Jonathan Flaishans, Meridith Fry, Trip Hook, Nelson Thurman, Jim Carleton, Shelly Thawley, Kurt Wolfe, Dirk Young, and Tom Purucker

Scaling Watershed Models: Modern Approaches to Science Computation with MapReduce, Parallelization, and Cloud Optimization

Jonathan Flaishans^a, Meridith Fry^b, Trip Hook^b, Nelson Thurman^b, Jim Carleton^b, Shelly Thawley^b, Kurt Wolfe^c, Dirk Young^b, Tom Purucker^c

^aOak Ridge Institute for Science and Education (Flaishans.Jonathan@epa.gov)

^bU.S. Environmental Protection Agency Office of Pesticide Programs (Fry.Meridith@epa.gov,
Hook.James@epa.gov, Thurman.Nelson@epa.gov, Carleton.Jim@epa.gov,
Thawley.Michelle@epa.gov, Young.Dirk@epa.gov)

^cU.S. Environmental Protection Agency Office of Research and Development (Wolfe.Kurt@epa.gov,
Purucker.Tom@epa.gov)

Abstract: Environmental models are products of the computer architecture and software tools available at the time of development. Scientifically sound algorithms may persist in their original state even as system architectures and software development approaches evolve and progress. Dating back to the 1980s, the EPA has developed algorithms to estimate the flux of pesticides from treated fields to neighboring water bodies. Recent development of the EPA's Spatial Aquatic Model (SAM) has provided an opportunity to redevelop, optimize and modernize this code used for regulatory decisions. Use of profiling has indicated a number of efficiencies that could be gained by updating the code to address execution time, memory utilization, CPU utilization and disk I/O issues. Porting the code to Python in order to access modern scientific computing and database libraries has allowed for a number of improvements and new use cases for SAM. These improvements include improved scalability, cloud infrastructure deployment, simpler cross-language communication and use of NoSQL databases. These improvements allow for SAM to be run as a service regardless of intensive input data, processing, and large output data requirements. Concurrent treatment of individual watersheds as embarrassingly parallel processes increases efficiency and scalability, while implementing MapReduce methods speeds up post-processing of the model outputs while accounting for network watershed structures. Converting to Python also has allowed the development process to leverage modern software testing frameworks and continuous integration design techniques. We discuss the experience of modernizing this code base with a goal of communicating useful design patterns for other science models.

Keywords: Watershed modeling; Cloud computing; parallel processing; modernization