Improving Water Resources Modeling Efficiency using Live Data Communication

Christopher Tubbs[1], Nels J. Frazier[2], Trey Flowers[1]

[1]*NOAA Office of Water Prediction, National Water Center, Tuscaloosa, AL*

[2]*Lynker Technologies*

Current workflows for modeling, writing to disk, evaluating outputs, and data dissemination generally follow a linear path which becomes increasingly inefficient at larger and larger scales. In this study we explore efficiency gains spanning from model execution all the way through product dissemination by developing methods to perform post-event analyses during run time. A typical modeling workflow involves gathering all input data, running the model across the entire domain, and finally writing all model output to disk. Afterwards, external programs and applications perform post-processing activities utilizing the generated model output. These applications often evaluate output, subset those outputs into regions of interest, transform the output into a number of different file and data formats, and shuttle the generated products off to locations that make it easier to access for external partners. These steps can easily take hours to complete for large domains.

We show that immediately publishing data generated by the model to a network for the consumption by external processes serves as an effective way to address these obstacles. Passing messages through channels facilitates the publication of application output followed by the subscription by external consumers, who are able to asynchronously detect and process anything passed through in real time. We use the capabilities offered by Redis, an open source in-memory data structure store, which provides easy to use asynchronous communication utilities. When an application publishes data to a channel, external processes immediately detect and consume them. To quantify the efficiency gained through processing data as it becomes available, we built two different workflows that create the same product. The first workflow was built to mimic the initial, linear process. We built the second workflow to distribute the work by publishing data from core processes, first a mock model, followed by evaluations performed by that modeling process. We show that publishing generated data at runtime, rather than upon completion, results in a significantly faster completion time.