25th International Congress on Modelling and Simulation, Darwin, NT, Australia, 9 to 14 July 2023
mssanz.org.au/modsim2023

Leveraging high-performance computing facilities for the SWAT water quality catchment model

Sandy Elliott [a], Linh Hoang [a], Thanh Duc Dang [a], Alexander Pletzer [b,c] and Chris Scott [b,c]

[a] National Institute of Water and Atmospheric Research, Hamilton, New Zealand
[b] National eScience Infrastructure, Wellington, New Zealand
[c] National Institute of Water and Atmospheric Research, Wellington, New Zealand
Email: sandy.elliott@niwa.co.nz

Abstract: SWAT (Soil and Water Assessment Tool) is a popular model for simulating water quality in catchments. The model is of intermediate-complexity, with daily time-resolution, a quasi-distributed spatial representation using hydrological response units (HRUs), and a well-mixed stream segment for each subcatchment. Despite these simplifications, long-term simulations with many HRUs can take an hour or so to run, making parameter conditioning ('calibration') or optimisation-based mitigation system selection using thousands of runs impractical on personal computers. We are applying SWAT for the Hauraki catchment in New Zealand, which entails considerable complexity and thus provides motivation for exploring methods to speed up model runs by utilising the High-Performance Computing (HPC) facilities of New Zealand's National eScience infrastructure (NeSI). NeSI includes a Cray cluster called Mahukia with 7552 AMD Milan processors spread across 64 nodes.

A review of the literature on parallel processing of SWAT revealed several approaches. One approach splits the catchment up, runs each part separately and then combines and routes the flow and contaminants. While this achieves speedup, there is considerable overhead and complexity with this split-apply-combine approach, and the speedup reduces when doing multiple model runs. Recognising the heavy and sequential I/O requirements of SWAT, some authors modified the I/O to make use of nosql databases, but these modifications have not been made generally available. We found that speedup of single runs through diagnostics and addressing bottlenecks and using alternative compliers did not result in many gains. We therefore adopted an embarrassingly parallel approach whereby individual model runs were allocated to separate cores. Each ran a parameter sample from a sample set. Preparing model input files with new parameters involves intricate modifications to the many SWAT files. Rather than developing our own routines to do this, we utilised existing software, SWATPlusR and R-SWAT, which run multiple SWAT instances through R-based parallelisation. We found that SWATPlusR did not scale well on Mahuika, with maximum speedup of 4 times and saturation at about 16 cores. We therefore developed an approach called SWAT-Parallel (https://github.com/pletzer/swatParallel), which initiates, runs multiple SWATPlusR workers through slurm job array scheduling and applies a final step to merge the results. The implementation of SWAT-Parallel allows for parallel scalability beyond the number of cores on a node (up to 128). Any number of jobs can now be submitted to a queue and they can run as soon as resources become available. In practice, a speedup of 100 was observed using 256 cores for GLUE-based parameter conditioning for the Hauraki catchment compared to a monolithic approach involving a single large job. SWATPlusR does not have calibration methods, and it just runs a set of parameters and combines results. Hence, developing more sophisticated calibration approaches would entail additional coding. Therefore, we are currently investigating R-SWAT, which includes several calibration methods. R-SWAT also has moderate scaling of 16x with 128 cores (a full node). Alternative MPI parallelisation options, shifting to a Capability platform (Māui), and profiling/optimisation are being investigated.

To run models on HPC, SWAT modellers who typically work on Windows machines need to transition to a Linux environment and familiarise themselves with scheduling software. This is eased to some degree by the emphasis by NeSI on training modellers and providing utilities such as a Jupyter web interface and standard Python, R and compiler libraries. Our experience is that this approach makes running SWAT models on HPC much less daunting. Overall, we have demonstrated that SWAT can be scaled up well for a cluster-based HPC with minimal code using multiple SWATPlusR workers, but there is still work to overcome limitations of SWAT's I/O overheads and to implement calibration methods apart from GLUE.

Keywords: *High-performance computing, water quality, catchment models, parallel processing, scalability*