Using Stochastic Climate Replicates In a Whole-Of-Catchment Catchment Model

Bridgart, R.J. [1,2], **A.C. Freebairn**[1,2], **and J-M. Perraud**[1,2]

[1]Commonwealth Scientific and Industrial Research Organisation, Land and Water
[2]eWater Cooperative Research Centre
E-Mail: robert.bridgart@csiro.au

Keywords: *catchment modelling, stochastic, climate variability*

EXTENDED ABSTRACT

There is a growing need for capabilities centred on modelling the effects of climate variability. WaterCAST (Water and Constituent Accounting Simulation Tool) is a software application with the ability to aid catchment managers by informing them of the probable effects caused by climate variability. This ability was realised through the incorporation of a stochastic rainfall generation model into a catchment modelling framework. The stochastic rainfall generation model comes from the Stochastic Climate Library (SCL) and the catchment modelling framework is E2. The system works by generating a series of statistically similar replicates from the catchment models' input data sets and running the model multiple times with the replicates. Once completed, the catchment models' outputs are compiled and presented in the form of a statistical summary. From this summary, important insight can be gained into system variability that can help to quantify uncertainty. Although heavily based on the E2 modelling framework, WaterCAST required the development of custom user interface components and careful design to incorporate as seamlessly as possible into existing E2 subsystems. Performance was critical to the usability of the system. Due to the production of large quantities of data whilst running WaterCAST with stochastic replicate inputs, mechanisms had to be developed to cope in such a way as to minimise computer memory consumption and ensure acceptable runtimes. The running of the catchment model multiple times, each run being fully independent of the others, lends itself to performance optimisation using parallel computing. The Invisible Modelling Environment (TIME), upon which E2 is built, already contains distributed computing capabilities. Those capabilities have the potential to be expanded to allow for the distribution of WaterCAST runs.

1. INTRODUCTION

Modelling systems for environmental decision support are traditionally used to explore the impacts of alternate scenarios of management actions. The growing public awareness of climate change projections and impacts is leading to a need for enhanced modelling capabilities. These capabilities include the evaluation of those projected impacts of changes in the climate drivers; a better representation of their variability; and the ability to estimate and quantify the uncertainty of those projections. These enhanced capabilities are more than a scaling up of current ones offered by environmental software decision support systems. They require a definite shift towards considering model results as an ensemble of data realisations.

WaterCAST (Water and Constituent Accounting Simulation Tool), being developed by the eWater Cooperative Research Centre, is a software application based on the E2 whole-of-catchment modelling framework (Argent *et al* 2005). When complete it will possess a number of capabilities designed to aid catchment managers in making informed decisions regarding the management of their catchments. WaterCAST will be able to run scenarios that simulate land use changes, changes in land management practices, climate variability and climate change. The first step toward completing WaterCAST was to add the capability to generate and use stochastic replicates to perform a statistical analysis of a system's response to a range of probable conditions. Using data stochastically generated, with similar overall statistical properties to observed data, opens a window on the possible range of system behaviour, with a better representation of variability, long-term behaviours and probabilities of extreme circumstances.

In order to add this capability to WaterCAST, new components needed to be developed and existing ones incorporated into the E2 modelling framework. A stochastic rainfall generation model from the Stochastic Climate Library (Srikanthan *et al* 2005) was utilised to generate the replicates. A component for the generic multi-running of a system developed in The Invisible Modelling Environment (Rahman *et al* 2003) was used to underpin the Monte-Carlo aspect of the WaterCAST product. To analyse the results of a simulation, a statistics suite was created to provide an overview of a river system's responses. The capabilities of WaterCAST are an extension to, not a replacement of, the existing E2 catchment modelling application. This means that all of E2's existing scenario definition capability and highly customisable environment are all accessible to the WaterCAST user. The original E2 system has three stages: scenario configuration, model execution and results presentation. All three stages have their associated user interface components and workflow defined in order to facilitate a normal single run of the system. One of the main goals of the WaterCAST product was to add the new multi-run capabilities to each of the three stages in such a way as to reuse interface components and maintain established workflow procedures. This makes WaterCAST extremely intuitive to use for existing E2 users and allows for the seamless use of existing E2 scenarios.

2. EXISTING TOOLS

2.1. E2 modelling framework

The E2 modelling framework is a software tool developed by the Cooperative Research Centre for Catchment Hydrology. It was designed as a flexible and easily adaptable software framework capable of achieving high levels of customisation without the need for redesign or restructure. E2 uses a robust conceptualisation of catchment processes, based around the notion of a catchment, made up of sub-catchments, wherein component models of different natural processes are positioned and linked to form a whole-of-catchment model. (Argent *et al* 2005, p. 1).

This conceptualisation is effectively expressed in the E2 core engine. The core engine is decoupled from the user interface and other support components making it easy to use programmatically. Its design and inherent flexibility made it possible for WaterCAST to use the data subsystem required for the stochastic scenario model runs. The development of this interface to the E2 engine was central to the success of the WaterCAST product.

2.2. The Stochastic Climate Library

SCL is a library of stochastic models for generating climate data. It includes models at the sub-daily, daily, monthly and annual timescales. Some models produce rainfall data at a single site, others produce climate data (rainfall, evaporation and maximum temperature) at a single site. One of the latest models to be included in SCL, the daily multi-site rainfall model, was selected as the best candidate for use in WaterCAST. It preserves the cross-correlation between the rainfall sites, which is important for use in a catchment modelling system composed of sub-catchments with different climate input time series. This site correlation leads to the generation of better quality stochastic replicates. SCL was originally designed as a standalone desktop application, though some steps were taken to pre-empt the use of the modelling engine in other applications. These pre-emptive measures meant that SCL required only minor modification to allow programmatic access to its stochastic generation models. The design allows current and future models to be exposed in a consistent manner. This is important because of the ongoing development of stochastic models running at different timescales and generating different climatic variables. The incorporation of new models into WaterCAST would provide modellers with the ability to assess a system's response to changes to various climatic inputs.

3. STOCHASTIC RAINFALL INPUT CAPABILITIES IN WATERCAST

To produce a product capable of running E2 river system scenarios using stochastic data input, a flexible and robust design was required to combine new and existing software components. The product needed to be able to support the specification of the input rainfall data to be used as the historic basis from which to generate the stochastic replicates. It would also be required to generate these replicates and run the E2 engine multiple times, storing the simulation results for statistical analysis. Addressing these needs proved difficult as their implementation raised serious performance issues. Due to the large quantities of data that would be consumed and produced when running WaterCAST, striking a balance between speed and efficient memory use was imperative. In order to achieve an optimal balance, WaterCAST was designed from the ground up with performance in mind.

3.1. Architecture

The WaterCAST application was built on the E2 framework. The E2 framework was built on TIME and TIME is built on Microsoft .NET. As WaterCAST is an extension of E2, it utilises all of E2's standard user interface components and adds several more components for its custom behaviour. A catchment model, comprising of input data, a river network and an assortment of configured rainfall runoff, constituent generation and filter models, is defined in WaterCAST as a river system scenario. After a river system scenario is configured, WaterCAST can utilise an SCL model to generate replicates and then execute the E2 engine with each replicate set, recording the desired outputs after each run. Upon completion, the outputs can be accessed through the standard user interface and statistics displayed (see Figure 1).

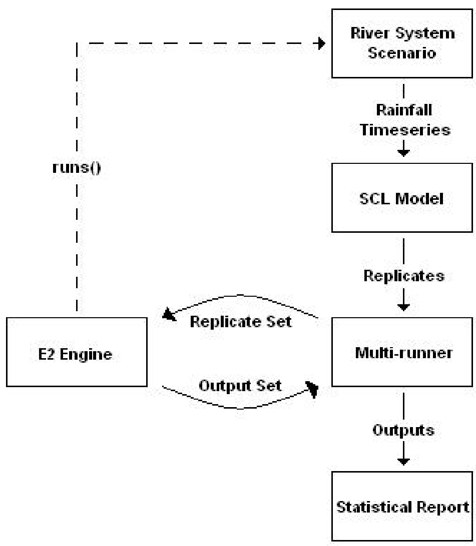

Figure1. WaterCAST workflow diagram.

3.2. Multi-run

The framework that supports the multi-run system is a generic framework developed in TIME. Conceptually it is a generalisation of previous work on estimating the effects of climate change projection on urban and regional water supply (Maheepala *et al* 2005). The multi-run framework was subsequently used in WaterCAST and serves to underpin the multi-running of the E2 engine (the core of WaterCAST). This multi-run system handles the acquisition of the input time series, the generation of replicates, the initialisation and running of the E2 engine and the storing of recorded output time series. Upon initialisation, the stochastic generation model is given the user specified input time series. Once all of the replicates have been generated from the specified time series they are then stored. The system then proceeds to retrieve one replicate set at a time and runs the simulation with that replicate set. After each run, the recorded outputs are stored. Having completed one run for each replicate set, all of the outputs are ready for statistical analysis. The design of the multi-run framework meant that it was relatively easy to implement in WaterCAST. The high-level design of the concrete multi-run system for using stochastic inputs is shown below in Figure 2.

1) Initialisation

Generate Stochastic Replicates

2) Execution

For n = 1 to num_replicates

a) Pre-execution

Load n‘ th replicate set

b) Execution

Run E2

c) Post-execution

Save outputs

3) Finalisation

Generate Statistics

Figure 2. Multi-run system design.

The initialisation stage generates the stochastic replicates and the finalisation stage compiles the output data sets for the generation of statistics. The execution cycle, i.e. one simulation run, is further broken into three smaller stages: pre-execution, execution, and post-execution. Pre-execution retrieves the next replicate set and resets the river system scenario. The execution stage runs the scenario and the post-execution stage ensures that the necessary output time series are saved. The inherent separation of the different stages within the multi-run system makes it relatively easy to change the behaviour of the different stages completely independent of each other. For example, one could easily change the initialisation stage to use a different stochastic generator without affecting the execution or finalisation stages.

3.3. User Interface Integration

The WaterCAST user interface operates in two basic modes, single-run and multi-run. When in single-run mode the system behaves in the standard E2 manner. A scenario is setup, run, and the results are viewed. Enabling multi-run mode causes subtle variations in the system designed to support multi-running operations. Operations such as the specification of input time series that are to be varied stochastically and defining the number of replicates to generate.

The same mechanism for selecting which results to view remains the same. In multi-run mode a statistical analysis is shown (See Figure 3) and alternatively when in single-run mode a single time series is shown.

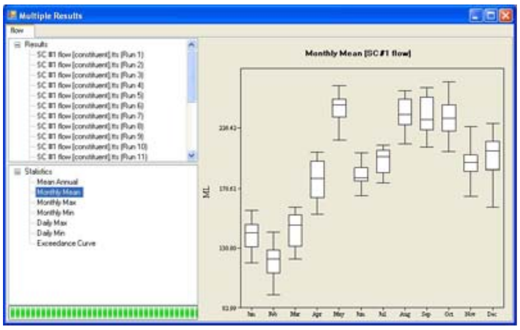

Figure3. Multiple results form. Showing a monthly mean box plot for recorded flow at the outlet of subcatchment #1.

This level of seamless integration was a high priority for the tool, as existing users would be readily able to utilise the new functionality and use their existing scenarios without the need for modification.

3.4. Data Handling

A typical E2 river system scenario may contain 400MB of spatial data, one hundred rainfall time series, five hundred other time series, and be set to record a further one hundred output time series. In this example, all time series are one hundred years long and are stored as double precision floating-point values. Running this scenario as a single run would produce the memory footprint shown in Figure 4. A multi-run of the scenario, stochastically varying all one hundred rainfall time series and generating fifty replicate sets would produce the footprint shown in Figure 5. This results in the generation of approximately 1.4GB of replicates and the production of a further 1.4GB of recorded time series.

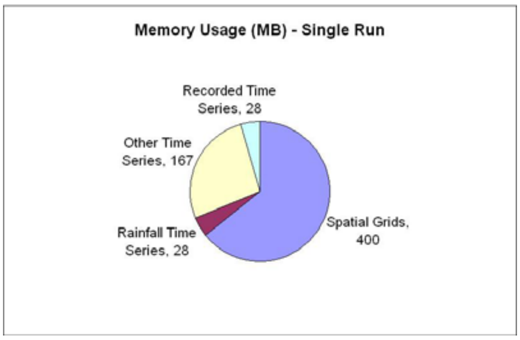

Figure 4. Typical memory usage for a single-run simulation.

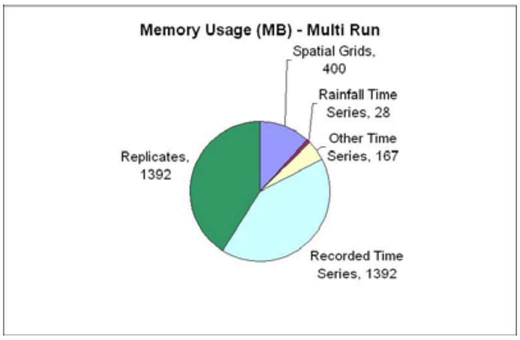

Figure 5. Typical memory usage for a multi-run simulation.

As can be seen, the replicate sets and the recorded outputs from the many E2 model runs easily dwarf the standard memory footprint of E2. Because of this, the application must be able to handle all of this data in a fast and efficient way. Two key strategies were implemented to handle the vast quantities of data produced by the system. The first centres on managing the stochastically generated input data and the outputs from the model runs. The generated replicates are stored on disk and only read in for the required model run. The outputs for a particular model run are saved to disk immediately upon model completion. This minimises the amount of data held in memory at any given time, ensuring that it will be no more than what one would expect during a normal single run of the catchment model. The second strategy attempts to reduce the load on the system caused by the calculation of the statistics on the outputs. WaterCAST utilises a "calculate on demand" principle in order to reduce computation times and to improve the user experience. The system also caches the statistics once they have been calculated to avoid unnecessary delay in recalculating results.

4. FUTURE

As the complexity of catchments being modelled increases, the performance of the modelling tools themselves becomes more important. Performance gains made through standard code optimisation, although potentially significant, are limited when compared to the potential gains made using parallel processing. The current trend in CPU technology is focused on the combination of multiple cores on a single chip. In order to utilise these cores, applications must be written to define tasks as parallel processes that can be run on the multiple cores. The running of the E2 engine in WaterCAST is a prime example of where multiple independent engine runs could be written to run in parallel. Conceptually, once designed for parallel execution, WaterCAST could also make use of a computational grid or cluster. Software packages like Alchemi (2007), that distribute work over a computational grid, have already been integrated into TIME. DIME (Davis *et al* 2005) is a component that allows TIME based applications to distribute work over an Alchemi Grid. This technology has the potential to be used in future versions of WaterCAST. Using distribution packages like Alchemi could dramatically reduce system runtimes and allow much more complex analysis to be performed in a given time period. The design of WaterCAST is such that it would be possible to utilise numerous other distribution middleware packages, such as Condor (2007) and Windows Compute Cluster Server (2007). The benefits of distributing the workload are obvious, either more runs can be completed within a given time period, or runtimes can simply be reduced.

5. CONCLUSION

WaterCAST is a software application that can assess the system behaviour of a catchment by running the system multiple times with stochastic input data. It was built on the E2 whole-of-catchment modelling framework. The incorporation of a stochastic rainfall data generator is a recent feature addition, the generation model itself having come from the SCL. Using stochastic input data, WaterCAST is able to highlight the effects of rainfall variability and allow assessments of the probability of extreme events to be completed. These capabilities were realised in WaterCAST through design stemming from attention to details, details such as performance and usability. The products inherent usability is primarily due to its fidelity to E2's original design. Extensive reuse of existing components and strict adherence to established workflow resulted in a high level of system integration. If not designed for memory efficiency and speed, WaterCAST would not be a useable desktop application. The enablement of a workload distribution system would serve to further increase the products usability through improved runtimes. The current generation of tools in WaterCAST for stochastic data generation and reporting are only the first step. The inclusion of new stochastic generation models operating on different timescales and replicating more climatic variables is a possible future direction for WaterCAST. Coupled with a more comprehensive data analysis package comprising of more statistical data and summaries, WaterCAST will be well positioned to address the future needs of catchment modellers.

6. REFERENCES

Alchemi (2007), Alchemi .NET Grid Computing Framework, http://www.alchemi.net/, Last Accessed August 2, 2007.

Argent, R.M., R.B. Grayson, G.M. Podger, J.M. Rahman, S. Seaton, J-M. Perraud. (2005), *E2 – A Flexible Framework for Catchment Modelling*. In Zerger, A. and Argent, R.M. (eds) MODSIM 2005 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand, December 2005, pp. 594-600, ISBN: 0-9758400-2-9.

Condor (2007), Condor: High Throughput Computing http://www.cs.wisc.edu/condor/, Last Accessed August 2, 2007.

Davis, G.P., R.J. Bridgart, T.R. Stephenson, and J.M. Rahman. (2005), *Adding grid computing capabilities to an existing modelling framework.* In Zerger, A. and Argent, R.M. (eds) MODSIM 2005 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand, December 2005, pp. 620-626, ISBN: 0-9758400-2-9.

Maheepala, S., B. Leighton, F. Mirza, M. Rahilly, and J. Rahman. (2005), *Hydro Planner – a linked modelling system for water quantity and quality simulation of total water cycle.* In Zerger, A. and Argent, R.M. (eds) MODSIM 2005 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand, December 2005, pp. 683-689, ISBN: 0-9758400-2-9.

Rahman, J., S. Seaton, J-M. Perraud, H. Hotham, D. Verrelli, and J. Coleman. (2003), *It's TIME for a New Environmental Modelling Framework.* In Post, D.A. (ed.) MODSIM 2003 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand, December 2003, pp. 1727-1732, ISBN: 1-74052-098.

Srikanthan, R., F.H.S. Chiew and A.J. Frost, (2005), Stochastic Climate Library User Guide, Catchment Modelling Toolkit, www.toolkit.net.au/scl/, Last Accessed August 2, 2007.

Windows Compute Cluster Server (2007), Microsoft Windows Compute Cluster Server 2003, http://www.microsoft.com/windowsserver2003/ccs/default.aspx, Last Accessed August 2, 2007.