International Environmental Modelling and Software Society (iEMSs)
7th Intl. Congress on Env. Modelling and Software, San Diego, CA, USA,
Daniel P. Ames, Nigel W.T. Quinn and Andrea E. Rizzoli (Eds.)
http://www.iemss.org/society/index.php/iemss-2014-proceedings

Parallel Simulation of Environmental Phenomena

Ralf Denzer [1], Peter Fitch [2], Ioannis N. Athanasiadis [3], Daniel P. Ames [4]

[1] Environmental Informatics Group, Germany (ralf.denzer@enviromatics.org),
[2] CSIRO, Australia (peter.fitch@csiro.au), [3] Democritus University of Thrace, Greece
(ioannis@athanasiadis.info), [4] Brigham Young University, USA (dan.ames@byu.edu)

Abstract: This paper discusses the current situation with respect to the parallel simulation of environmental phenomena. It is not based on a complete and thorough investigation of the state-of-the-art in this domain and is more driven by observations, experiences and ideas of the four contributors who see the need to raise awareness towards a more systematic approach in the future. It is meant as a discussion paper to stimulate a more systematic approach towards parallel simulation in the environmental domain. While examples are mostly taken from hydrology, they are only used as illustration of a generic situation. The section on related work is mostly based on work published in Environmental Modelling & Software. There might be more literature to look at supporting or contradicting the statements made. The section on practical experiences is largely based on work carried out at CSIRO. The section on theoretical considerations is largely based on observations and ideas of the first author. Together we see the need for a more systematic approach based on sound principles derived from modern software engineering of distributed systems which needs to be grounded by the application domain.

***Keywords**:* Parallel simulation; concurrent processing; distribution patterns; catchment simulation; hydrological networks

1 MOTIVATION

Growing environmental data availability and growing needs of large scale environmental simulations lead to the requirement to run simulations faster, more responsive and more effective, in particular for ensemble simulations and for simulations with multiple models. There have been a number of efforts to parallelize simulation on computing clusters and distributed web-service-based environments, but it appears that these attempts have so far not lead to a systematic approach. At the same time cloud infrastructures and technologies emerging from the BigData domain offer new possibilities for environmental simulation, similar to main stream applications in the business domain.

The primary goal of this paper is to raise the awareness in the scientific community (the modelling as well as the computer science community) that a systematic approach is needed if parallel processing is to become reliable and affordable. This paper, together with the related iEMSs 2014 session is intended to foster the exchange of ideas, experiences, good practices, and considerations for a systematic approach which will lead towards recognised and re-usable distributed computing patterns. We anticipate the discussion resulting from this and related papers can ultimately contribute to a more complete position paper capturing the state of the art in the area of parallel processing of environmental phenomena.

2 RELATED WORK

During the past decade, production and use of single-processor hardware acceleration has reduced significantly, giving rise to multi-core computer architectures, while at the same time graphics processor unit (GPU) capabilities for matrix operations has exploded. This has had a tremendous impact on programming languages and software stacks that account for concurrent and parallel computation. As a result, environmental software is currently embracing such capabilities to improve performance.

Parallelization of environmental models mostly refers to three distinct methods:

1. **Run a model simultaneously on several machines**. This entails the parallel execution of alternative configurations/parameterizations, either on a local computer cluster or over the network on the grid or on the cloud, i.e. (Bryan, 2013; Pijanowski et al., 2014; Yalew, Griensven, Ray, Kokoszkiewicz, & Betrie, 2013).
2. **Components as services**. In this approach, a model is decomposed into autonomous computing entities that are offered over the web as services, i.e. (Goodall et al., 2013).
3. **Concurrent computations**. The model needs to be re-implemented so that it is optimal for multi-core, multi-threaded programming, as in (Zhang et al., 2013). A second method in this category is algorithm optimization for GPUs, when extensive matrix operations are involved, i.e as in (Vacondio, Palù, & Mignosa, 2014).

Below we present efforts published recently in the journal *Environmental Modelling & Software*. We focused on concurrent computing approaches, which are rising and growing fast.

US EPA's Storm-Water Management Model (SWWM) model has been the subject of several studies, mainly due to the fact that it is open source, and widely used. Wu et al (2013) re-implemented parts of SWMM using Message Passing Interface (MPI) and managed to reduce model execution time by 42%–70% (or speedup of 1.74 to 3.36) using up to five parallel processes. Similarly, Burger et al. (2014) used OpenMP (Open Multi-Processing)[1] to improve SWMM run-time, so that it can be executed on a multi-core machine. They reported speed gains of six to ten times on a twelve-core system.

In a simultaneous effort, Zhang et al. (2013) employed OpenMPI[2] (Open Message Passing Interface) for multi-objective calibration of SWAT. Reported gain ratios range between 45 to 109, depending on the model complexity. Liu et al. (2014) used OpenMP in a layered approach for Fully Sequential Dependent Hydrological Models (FSDHM): calculations on simulation units in the same layer are independent and can be conducted in parallel. Gains are bigger with large datasets than with small datasets, and the maximum speedup ratio exceeded twelve. Vacondio et al. (2014) present a GPU-enhanced implementation for a flood simulation system that is based on the CUDA framework. The reported improvement in speed is of two orders of magnitude with respect to single-core computers.

A special case is the work of Zhao et al. (2013) who employed code improvements for multi-threaded programming and executed their code on a heterogeneous, ad-hoc grid. An enormous number of APSIM simulations was performed in less than eleven days on CSIRO's Condor grid system, that made available some 5000 office computers during weekends and after office-hours. By using less than a third of this infrastructure, authors run simulations that would have taken over 30 years to execute on a single machine.

3 PRACTICAL EXAMPLES: CATCHMENT SIMULATION

In this section experiences with the simulation of catchment models at catchment at basin and catchment scale are presented.

[1] http://www.openmp.org
[2] http://www.open-mpi.org

3.1 Catchment modelling at basin scale

(Chiew et al., 2008) reports on a catchment modelling method that uses SIMHYD with Muskingum routing method to estimate current and future water availability in the Murray Darling Basin. This basin is located in South Eastern part of Australia comprising an area of around 1,000,000 km² and is one of Australia's most significant agricultural areas for the study the basin was partitioned into 5 km x 5 km grids across the entire basin which were then aggregated into 18 reporting regions. The total number of grid cells and model instances is c. $4x10^4$ leading to a significant computational challenge. The approach to calibration of the models was to calibrate at a catchment scale to an objective function, which minimised the differences between modelled and observed flow, and then to apply the parameter set to the individual models which are again run on a grid by grid basis. (Perraud, Vleeshouwer, Stenson, & Bridgart, 2009).

Three different parallelisation strategies were assessed. The first two are presented here as the third is a variant of the second. The first strategy parallelizes the calibration algorithm to enable whole of catchment or sub calibration on a grid cell by grid cell basis (Gregor & Lumsdaine, 2008). To achieve optimisation would require exchange of information between the parallel algorithms thus requiring a Message Passing Interface. This would enable an evolution of optimal parameter sets jointly over the gridded catchments. From a hydrological perspective this would be the most desirable option however it is also the most complex.

The second option is to run the model at grid scale and then have custom code to aggregate the outputs and evolve the calibration datasets. The upside of this approach is that it assumes no lateral flow between cells and eliminates the requirement for message passing, but may result in a sub-optimal parameter set used for the simulations.

Using a parallel approach in this study reduced the model calibration task from an estimated 40 days to around 2 days or around a 40 times speed up. It is clear that without the use of a parallel approach the study could not be completed in the time required for the projects.

3.2 Continental scale catchment modelling

One example of work has focused on using parallel approaches to reduce runtime for calibration. (Perrauda et al., 2013) report on the development of a continental scale landscape modelling system, which is used on a continental scale catchment with a data assimilation framework. The model used is AWRA-L, a continental scale biophysical model that simulates the water balance across the landscape.(Van Dijk et al., 2011) The landscape is divided into 277,770 grid cells combined with an Ensemble Kalman filter (EnKf) data assimilation scheme which uses 100 ensembles. The scale of this computation problem is self-evident and Perraud investigates both the use of multicore and of GPU to reduce computation time. Perraud reports that using running the system to simulate a 6 year period would require 10-15 days, that this could be reduced with software optimisation (using better software technologies) and parallelisation to 5-6 days, and using GPU to 18hours to 1 day. Without the benefit of parallelisation development of this system with lengthy runtimes would be extremely problematic.

3.3 Modelling Opportunities

Granell (Granell, Schade, & Ostländer, 2013) define a taxonomy or viewpoint of integrated modelling tools. These viewpoints are component based, scientific workflow systems, virtual research environments, service based and resource based modelling. Despite the different viewpoints, the purpose of these different approaches to integrated modelling remains the same, i.e. to easily create integrated modelling systems that are useful in simulating earth system phenomena. Many of the environmental challenges today, are interconnected interrelated systems based problems that necessarily require integrated environmental modelling approaches. The need therefore to continue developing systems frameworks and technologies for model integration in the environmental sciences is clear. These interconnected and interrelated systems more than ever can benefit from the application of parallelisation approaches.

As described above, the combination of models and observations using fundamental data assimilation approaches is expected to grow, particularly as new remote observations become available. This with the scale of the problem places much emphasis and importance on harnessing the opportunities that parallel approaches present.

Granell also concludes that perhaps the most fruitful area for integrated modelling is with scientific workflow systems using resource-based modelling on top of virtual research laboratories. A key challenge therefore is how the work can be partitioned successfully to make use of the resources available. Table 1 shows the assessment of the maturity of application of parallel technologies and their opportunity potential from the experiences.

Technology	Opportunity	Potential	Maturity
Multi-threading	Many existing environmental modelling codes have not kept pace with the advances in computer hardware and are not even multi-threaded. In Australia this is the case for our river and catchment modelling tools	med-low	Med-Low
Multi-core	As above	med-low	Med-Low
Cluster within enterprise - HPC	Many scientific and government institutions have access to cluster or HPC computation facilities within their environments. Many practitioners are using these facilities but the general uptake in environmental modelling and simulation is low due to the complexity of application. Development of guidelines and best practice approaches would be very beneficial.	high	Low
Cloud or GRID	There are few examples internationally of the GRID or cloud being used for hydrological modelling. The opportunity with this technology is tremendous, particularly as IT departments look to outsource complex and costly infrastructure. GRID world is becoming more mature, with application routine in the geosciences, and with the availability of toolkits such as GLOBUS and the environmental simulation could similarly benefit	high	Low

Table 1. Assessment of approaches.

3.4 Practical barriers

One of the clear problems in applying parallel approaches to environmental modelling and simulation is the lack of a clear set of best practice guidelines. Guidelines which help assess the class of parallel problem for example being “embarrassingly parallel” to one that is either Single Program or Multiple Data to one of Multiple Data and Multiple Program whilst providing guidance on the level of optimal granularity would be very beneficial. For example where the question is one of model parameterisation guidance on the structure of optimisation of the code for both development simplicity and performance would be helpful. Another aspect of the guidance required is that of navigating the complex ecosystem of tools and technologies to take advantage of parallelisation. (Granell et al., 2013) describe a review of integrated modelling tools, and a similar review of the tools through the lens of parallelisation opportunities would be beneficial. This lack of guidance for the practitioner is complicated by the multitude of tools and approaches as well as the complex ecosystem of technologies. Another barrier is the lack of experience with suitable software tools that are used to develop environmental modelling applications. In many areas of the environmental sciences adaptation to the rapid pace of change of technology has not occurred. Part of the problem is a lack of awareness of the methods and opportunities that parallel approaches present. The running of workshops and sessions at suitable conferences would go some way to assisting in this regard.

4 THEORETICAL CONSIDERATIONS

4.1 Assessment of the current situation

Of the three main approaches presented in section 2, only the second and third approaches are worth considering from the angle of distributed parallel systems. The first approach is a "poor man's parallelization" which just feeds a black-box input-output model with different parameters and dumps results to some storage. This is not a parallel distributed system.

Considering the two other approaches shows some of the problems associated with parallel simulation. Approach 2 is particularly volatile: we have all seen error messages saying "the process cannot be completed because service X is not available". This demonstrates the need of high availability and fault tolerance if applying a distributed simulation approach. Approach 3 is particularly consuming resources: it requires re-designing modelling code which can be very expensive. Both approaches are heavy when it comes to deploying the parallel code on some sort of infrastructure like Grid infrastructures, which are complex tools themselves.

We just have to face reality: designing, implementing, deploying, and operating parallel distributed applications in a highly available, reliable and secure fashion is a tough challenge. There are no best practices or theoretical foundations for parallel distributed systems development for the environmental domain. Specialized modelling tools and environments (e.g. OpenMI, Open Modelling Interface[3]) provide the basics for component to component communication, but do not really explain how to use them.

At the same time, the general technological theatre is rapidly changing and offers a wealth of new opportunities. Largely driven by the internet and business domains, new highly parallel and highly reliable software environments are becoming available. Cloud infrastructures offer the means for on-demand resource allocation which would at least theoretically enable anyone to perform large scale simulations without the need to own every piece of infrastructure themselves. Which opportunities this wealth of new technologies offer for the environmental domain is currently anybody's guess.

4.2 Vision

We share the opinion that the application of parallel approaches is currently **unsystematic and follows more of an ad-hoc approach than sound theoretical principles**. We see a need to change this situation. We propose to take a look at (large scale) environmental modelling problems from the viewpoint of **inherent distribution opportunities** and **potential** (problem-based) **software distribution patterns**, which can be **mapped to state-of-the-art infrastructures**. We see the need to build best practice solutions which can serve as blueprints for practitioners. These blueprints need to be thoroughly validated and tested towards: a) typical application requirements from the environmental domain and b) software quality requirements from the domain of distributed systems (availability, scalability, fault tolerance, manageability and so forth).

4.3 Example: implicit parallelism in hydrological networks

Coming back to the example of hydrological models we can demonstrate what we mean by this vision. A hydrological network is basically a directed graph, and can always be represented as a directed binary graph (see for instance the EEA ECRINS dataset[4]). Whether this graph contains cycles or not is not important for our current discussion.

As this a flow-based directed system, there is an inherent potential to parallelize the problem, by assigning graph nodes or clusters of graph nodes to processing nodes (to be precise: this is only the spatial parallelism of the problem; similarly the problem can also be parallelized on the time axis, as

[3] http://www.openmi.org/

[4] http://www.eea.europa.eu/data-and-maps/data/european-catchments-and-rivers-network

computing nodes can take over time steps or series of time steps from other “upstream” computing nodes).

In (Wang 2011) this principle is applied to nodes in a tightly-coupled local computing cluster. (Grübsch et. al 2001) are discussing the partitioning of typical hydrological networks into clusters. It remains unclear whether these approaches have been applied on problems of real world size.

A systematic approach as we would want it for the future would ideally contain at least the following elements:

1. Algorithms (as in Grübsch 2001) which would give good partitioning (or clustering) of nodes of a given hydrological network of realistic size, up to continental scale.
2. One (or several) software distribution patterns defining how the clusters would be distributed on top of one (or several) infrastructure patterns (which themselves represent real infrastructures).
3. An execution/scheduling plan which would allocate these clusters to computing nodes in the infrastructure.
4. A problem-adapted middleware or runtime environment supporting plugging-in of the actual modelling code, data sources and data sinks.
5. Benchmarks validating trade-offs between chosen parameters (hydrological cluster size, computing cluster size, choice of infrastructure pattern), based on chosen problem parameters (size of datasets, number of state variables, input and output variables etc.).

From the viewpoint of software architectures it would be interesting if such a blueprint for hydrological systems could be generalized towards other flow based systems like systems of drainage pipes.

Thinking this towards the end, we would like to see guidelines which would be able to answer questions like: a) “what approach should you take if you have a *flow-based system* of this overall size, that node size and such and such infrastructure choices”; b) “what approach should you take if you have a *grid-based* (for instance atmospheric) system …”; c) “what should you do if you need to *map* grid-based models on top of flow-based models of such and such variety”.

REFERENCES

Bryan, B. A. (2013). High-performance computing tools for the integrated assessment and modelling of social–ecological systems. Environmental Modelling & Software, 39, 295–303. doi:10.1016/j.envsoft.2012.02.006

Burger, G., Sitzenfrei, R., Kleidorfer, M., & Rauch, W. (2014). Parallel flow routing in SWMM 5. Environmental Modelling & Software, 53, 27–34. doi:10.1016/j.envsoft.2013.11.002

Castronova, T.A., Goodall, J.L., 2010. A generic approach for developing process-level hydrologic modeling components. Environ. Modell. Softw. 25, 819-825.

Chiew, F., Vaze, J., Viney, N., Jordan, P., Perraud, J.-M., Zang, L., et al. (2008). Rainfall-runoff modelling acrossthe Murray-Darling Basin, 1–70.

Goodall, J. L., Saint, K. D., Ercan, M. B., Briley, L. J., Murphy, S., You, H., … Rood, R. B. (2013). Coupling climate and hydrological models: Interoperability through Web Services. Environmental Modelling & Software, 46, 250–259. doi:10.1016/j.envsoft.2013.03.019

Granell, C., Schade, S., & Ostländer, N. (2013). Seeing the forest through the trees: A review of integrated environmental modelling tools. Computers, Environment and Urban Systems, 41(C), 136–150. doi:10.1016/j.compenvurbsys.2013.06.001

Gregor, D., & Lumsdaine, A. (2008). Design and implementation of a high-performance MPI for C# and the common language infrastructure. Presented at the Proceedings of the 13th ACM SIGPLAN

Grübsch M., David O. (2001). How to Divide a Catchment to Conquer Its Parallel Processing - An Efficient Algorithm for the Partitioning of Water Catchments, Mathematical and Computer Modelling 33, 2001, pp. 723-731)

Liu, J., Zhu, A.-X., Liu, Y., Zhu, T., & Qin, C.-Z. (2014). A layered approach to parallel computing for spatially distributed hydrological modeling. Environmental Modelling & Software, 51, 221–227. doi:10.1016/j.envsoft.2013.10.005

Perraud, J.-M., Vleeshouwer, J., Stenson, M., & Bridgart, R. J. (2009). Multi-threading and performance tuning a hydrologic model: a case study, 1059–1065.

Perrauda, J. M., Collins, D., Bowden, J. C., T Raupach, B., Manser, P. A., Stenson, M. P., Renzullo, L. J. (2013). A balancing act in heterogeneous computing – Developing the AWRA-Landscape data assimilation system. 20th International Congress on Modelling and Simulation, Adelaide, Australia, 1–6 December 2013 Www.Mssanz.org.Au/Modsim2013, 1–7.

Pijanowski, B. C., Tayyebi, A., Doucette, J., Pekin, B. K., Braun, D., & Plourde, J. (2014). A big data urban growth simulation at a national scale: Configuring the GIS and neural network based Land Transformation Model to run in a High Performance Computing (HPC) environment. Environmental Modelling & Software, 51, 250–268. doi:10.1016/j.envsoft.2013.09.015

Vacondio, R., Palù, A. D., & Mignosa, P. (2014). GPU-enhanced Finite Volume Shallow Water solver for fast flood simulations. Environmental Modelling & Software. doi:10.1016/j.envsoft.2014.02.003

Van Dijk, A., Bacon, D., Barratt, D., Crosbie, R., Daamen, C., Fitch, P., et al. (2011). Design and development of the Australian Water Resources Assessment system. Proceedings, Water Information Research and Development Alliance Science Symposium. Retrieved from https://publications.csiro.au/rpr/pub?list=BRO&pid=csiro:EP116648&sb=RECENT&n=8&rpp=10&page=272&tr=4282&dr=all&dc4.browseYear=2012

Wang H. et al (2011). A common parallel computing framework for modeling hydrological processes of river basins Parallel Computing 37 (2011) 302–315

Parallel Computing, ElsevierWu, Y., Li, T., Sun, L., & Chen, J. (2013). Parallelization of a hydrological model using the message passing interface. Environmental Modelling & Software, 43, 124–132. doi:10.1016/j.envsoft.2013.02.002

Yalew, S., Griensven, A. van, Ray, N., Kokoszkiewicz, L., & Betrie, G. D. (2013). Distributed computation of large scale SWAT models on the Grid . Environmental Modelling & Software, 41, 223–230. doi:10.1016/j.envsoft.2012.08.002

Zhang, X., Beeson, P., Link, R., Manowitz, D., Izaurralde, R. C., Sadeghi, A., … Arnold, J. G. (2013). Efficient multi-objective calibration of a computationally intensive hydrologic model with parallel computing software in Python. Environmental Modelling & Software, 46, 208–218. doi:10.1016/j.envsoft.2013.03.013

Zhao, G., Bryan, B. A., King, D., Luo, Z., Wang, E., Bende-Michl, U., Yu, Q. (2013). Large-scale, high-resolution agricultural systems modeling using a hybrid approach combining grid computing and parallel processing . Environmental Modelling & Software, 41, 231–238. doi:10.1016/j.envsoft.2012.08.007