Large-Scale High-Resolution Groundwater Modelling using Grid Computing

W.L. Berendrecht[1], A. Lourens[1], J.J.J.C. Snepvangers[1] and B. Minnema[1]

[1]TNO Built Environment and Geosciences – Geological Survey of the Netherlands, Utrecht, The Netherlands
Email: wilbert.berendrecht@tno.nl

Keywords: *grid computing, model decomposition, groundwater, high-resolution modelling, representer calibration*

EXTENDED ABSTRACT

Developments in numerical groundwater modelling have shown that models become more and more ambitious with increasing computer capacity. To meet the need for accurate instruments to support decision-making, both scale and resolution of models have grown enormously during last years.

Within the recent project 'Development of a Methodology for Interactive Planning for Water Management' (MIPWA) the challenge was taken to develop a high-resolution numerical groundwater model for the whole north of the Netherlands. This MODFLOW model encompasses an area of more than 24.000 km^2, has seven quasi-3D model layers and a resolution of 25 x 25 m^2. With this enormous groundwater model – which is unique in its size – 13 years of daily groundwater fluctuations had to be simulated.

Both running and calibrating such a large model require innovations in model building, model processing and data handling. Data-compression techniques were required to store all input and output data. To run the model, both upscaling and model-decomposition techniques were develoved. For a transient run (over 4500 time steps) on the highest resolution, the model was decomposed into 473 overlapping submodels. Transient boundary conditions of the submodels were taken from a lower-resolution model. Each submodel could be run individually, so the process was perfectly suited for parallel processing. Therefore, we developed a computational grid using the 200 computers available in our office. The moment employees logged off, their computer came available for the grid. Obviously the weekends appeared to be the most productive days!

The grid was also crucial for model calibration. We used the Representer method for calibrating model parameters in a stationary mode. The Representer method requires a forward run and an adjoint run each iteration to calculate the so-called representer of each observation. In total more than 8000 groundwater observation locations were available and hence more than 8000 runs had to be carried out. Each representer run was distributed over the grid using PVM (Parallel Virtual Machine).

Grid computing revealed itself as the only way to complete the whole project within reasonable time. Total CPU time of model calibration and running (ca. 50 runs during model-construction process) was estimated at more than 20 years. Using grid computing, the calculation time was reduced to several months.

In addition to model calibration and model running, grid computing is also helpful in data-assimilation applications. In a preliminary study, Ensemble Kalman Filtering techniques were applied for nowcasting and forecasting of groundwater fluctuations using assimilated groundwater. Model states were estimated by calculating 200 ensembles distributed over the grid. Subsequently, 10-day forecasts of groundwater levels were calculated by processing 50 ensembles of the Ensemble Prediction System (EPS) calculated by the European Centre for Medium-Range Weather Forecasts. As the intention is to produce forecasts on daily basis, a computational grid is necessary to run all ensembles within one day.

1. INTRODUCTION

In 2005 the 'Development of a Methodology for Interactive Planning for WAter management' (MIPWA) project has been started. In this project seventeen water management organisations in the north of the Netherlands (four provinces, three drinking water companies, six waterboards and three municipalities) developed – under leadership of TNO and together with research institute Alterra and two consultancy agencies – a high-resolution regional decision-making tool for groundwater management.

The MIPWA project area is shown in Figure 1. It is a varied agricultural and natural area with little urban development. Almost half of the area is near sea level and a spread of small channels controlled by weirs and pumps dominates the water system here. The other half of the area has a more natural sloping drainage system. Total groundwater withdrawal for drinking water and industrial needs is 380 million m^3 per year.

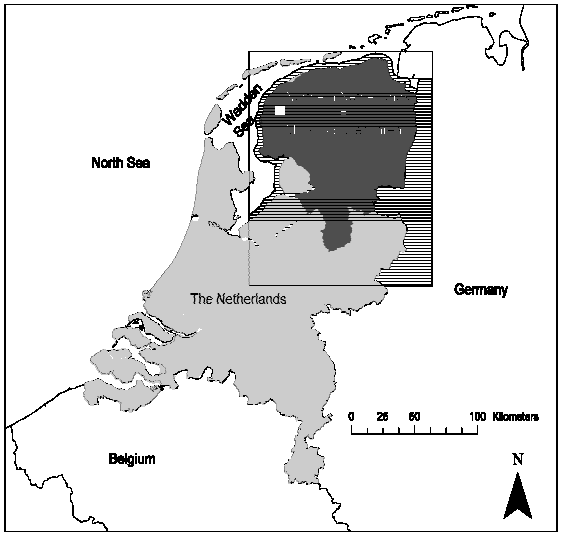

Figure 1. MIPWA project area (dark grey) and total model area (shaded)

The MIPWA model covers the area of interest plus a buffer area to decrease the impact of the model boundaries (145 km East-West and 167 km North-South). It is a MODFLOW model with 25 m grid cells. In total the model has more than 238.000.000 active model grid cells (Figure 1) over 7 quasi-3D model layers. The model time step is 1 day. The model has been run for a period of 13 years (1989-2001).

The large dimensions of the model required special computer facilities. We needed 2 TB data storage. For a model run 2 GB internal memory was needed. The main challenge, however, was reducing CPU time. As the model could not be run on a single computer, model decomposition technique needed to be applied. We choose to decompose the model in such way that the submodels could be run independently of each other and thus in a parallel mode.

This paper presents how we applied parallel computing within groundwater modelling research. Our goal is to demonstrate the strengths of grid computing in groundwater modelling applications, rather than to address detailed technical issues of computational grids.

First, we briefly discuss the computational grid we developed to facilitate parallel computing. Second, we present a method for model decomposition that makes it possible to distribute model runs over the grid easily. Furthermore, we describe the application of grid computing during model calibration. Finally, we end with some concluding remarks and describe future applications of grid computing in groundwater modelling.

2. DESCRIPTION OF COMPUTATIONAL GRID

To facilitate parallel running of the model, we built a computational grid (e.g. Foster and Kesselman, 1999; Snavely et al., 2003) using all computers available in the computer network. Approximately 200 computers were upgraded with 2 GB internal memory. In addition, a Linux operating system was installed on all computers. When one logs off from Microsoft Windows the system reboots itself after 5 minutes. When the computer restarts one can choose between Linux (making the computer available for the grid) and Microsoft Windows (normal login). If no choice is made within 5 minutes, the computer automatically starts in Linux mode and becomes available for the grid.

Practical advantages of grid computing over, for instance, a supercomputer are:

1. Grid computing is relatively cheap because it uses computers that are already available;
2. It is flexible. Computers can be added and removed easily;
3. A computational grid remains up-to-date. New computers that are added after several years generally have up-to-date specifications. So the grid is being upgraded "automatically".

Disadvantages of a computational grid are:

1. There is a danger of high network load. This may greatly influence the performance of the grid;
2. The size of the grid depends on the willingness of colleagues to log off every day.

3. MODEL DECOMPOSITION

3.1. Introduction

The MIPWA groundwater model as described in the first section has more than 238.000.000 grid cells. This model cannot be run on a single computer due to memory demand and CPU time. Hence the model needs to be decomposed into submodels. The main problem of using submodels, however, is that for transient calculations boundary conditions need to be transient as well. So for each of the 4748 time steps boundary conditions need to be determined a-priori. In this section we present a method for model decomposition using transient boundary conditions. This method consists of the following steps:

1. Run an upscaled model and store heads of each time step;
2. Downscale heads and store these on intervals only;
3. Define submodels using the heads of step 2 as boundary conditions;
4. Run overlapping submodels and store heads on intervals each time step;
5. Define new submodels using an offset so that heads of step 4 can be used as new boundary conditions and run new submodels.

3.2. Model upscaling

The first step is to upscale the model so that it can be run on a single computer. For the MIPWA model we had to go to a resolution 250x250 m^2. At this resolution the runtime was 48 hours on a 3.0 GHz machine (4748 time steps). For each time step calculated heads of each model layer were stored (ca. 41 GB data).

3.3. Downscaling of boundary conditions

High-resolution boundary conditions are now calculated by downscaling the stored 250x250 m^2 heads using an interpolation algorithm. This is illustrated in Figure 2. As 4748 time steps x 7 model layers = 33236 grids of high-resolution groundwater heads would need more than 5 TB disk storage, we only store high-resolution data at intervals of 100 grid cells, i.e. row number 1, 101, 201, …, 6680, and column number 1, 101, 201, …, 5800 (see Figure 2).

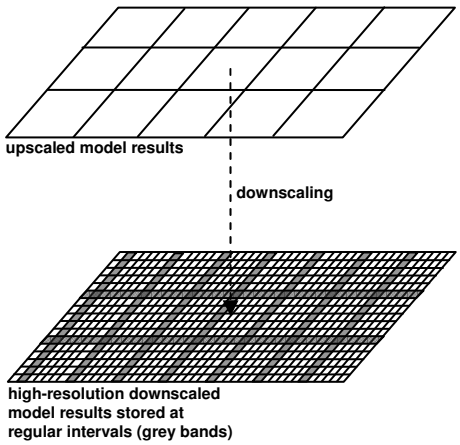

Figure 2. Procedure for calculating transient boundary conditions for submodels

3.4. Definition of submodels

As boundary conditions are stored on an interval of 100 cells, submodels must have a dimension of a multiple of 100. Now the question is: what are the optimal dimensions of a submodel? Computational time increases exponentially with increasing model dimensions. On the other hand, the influence of boundary conditions becomes relatively stronger with decreasing model dimensions. Nevertheless, the main constraint for choosing the model dimensions is a practical one. Model results are stored at the end of the run. When a job is removed from the computational grid before the run is finished, it has to be rerun completely. Since most computers in the grid are only available at night, a job must take no longer than 13-14 hours. This was realised with submodels of 300 x 300 grid cells.

As we used downscaled boundary conditions, we defined a buffer zone of 50 cells to damp the effect of errors in boundary conditions. Hence, the so-called area of interest of each submodel was 200 x 200 cells (see Figure 3). In this way we needed 473 overlapping submodels to cover the whole model area.

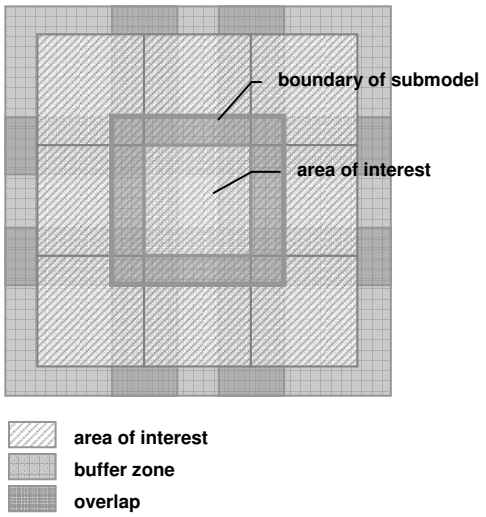

Figure 3. Illustration of areas of interest and overlapping buffer zones.

3.5. Storage of high-resolution boundary conditions

In groundwater modelling a buffer zone of 50 cells x 25 m = 1250 m is small. Boundary condition effects are generally not damped completely within this range. Therefore, for each submodel, calculated heads are stored each time step at an interval of 100 cells (see Figure 4). These heads replace the downscaled boundary conditions. When all submodels have been run, new submodels are defined with an offset of 100 cells (in both directions). The new submodels are then run with the newly calculated boundary conditions.

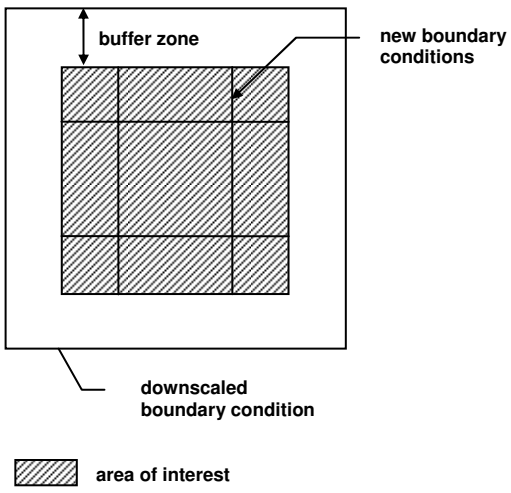

Figure 4. Definition of buffer zone and new boundary conditions for a submodel

Accurate model results were obtained by running 2-3 iterations as described above. Three iterations were needed only in areas with slow damping and near groundwater extraction wells.

Total CPU time of 473 model runs was more than 6600 hours. Two model runs cost 13200 hours. This is 1.5 years! Using the computational grid with, on average, 140 machines two model runs could be completed within 4 days. Hence, in this project the availability of a computational grid made a big difference to the feasibility of the project.

4. MODEL CALIBRATION

In various studies, parallel computing has been demonstrated to speed up the process of model calibration significantly (Eklund, 2004; Herrera et al., 1998; Vrugt et al., 2006). The MIPWA groundwater model presented in this paper was calibrated on observations of groundwater head using the Representer method (Valstar et al., 2004). This method is perfectly suited for parallelisation since many independent model runs has to be carried out. Running a Representer calibration run on the computational grid is therefore expected to reduce total runtime dramatically. This section briefly address the issues concerning the calibration of the MIPWA model using the computational grid.

We calibrated transmissivity of the aquifers and vertical conductances of the aquitards with a stationary model. The Representer method requires a forward run and an adjoint run per observation each iteration to calculate the so-called representer of the observation. More than 8000 groundwater observation locations were available. This means that 8000 forward and adjoint runs needed to be executed.

Obviously, running 8000 forward and adjoint runs with the high-resolution 25x25 m^2 model was infeasible. Therefore, we used an upscaled model. Scaling of the non-linear surface water system was of major importance. We applied the Cauchy correction methods proposed by Vermeulen et al. (2006) in an iterative manner. The Cauchy corrections were calculated a-priori as the difference between high-resolution heads and low-resolution (averaged) heads. These correction terms were then applied to the surface water and drainage system in the upscaled model. Observations were corrected as well.

Each representer run was distributed over the grid using Parallel Virtual Machine (PVM) software (Geist et al., 1994). PVM is very robust and stable

in a computational grid where machines are added and removed during the process. When a machine is removed from the grid, the job is reclaimed an sent to another machine.

Similar to a model run, a calibration run was only feasible due to the grid. Without a grid, calibration would have taken several years.

5. CONCLUSIONS

This paper described the application of grid computing in groundwater modelling research. During the MIPWA project we developed a large-scale high-resolution numerical groundwater model. Running and calibration of this model required the introduction of grid computing.

Model decomposition techniques were developed to enable parallel running on the grid. On average, the grid consisted of 140 computers with 3 GHz processors and 2 GB internal memory per computer. This enormous computer capacity made it possible to make several model and calibration runs within a couple of months instead of many years.

At this moment, the computer grid has already been applied successfully in three other groundwater modelling projects. Furthermore, the grid is also perfectly suited for data-assimilation purposes. Recently, a pilot project was carried out to set up a forecasting system for groundwater levels. In this project a numerical groundwater model has been embedded in an ensemble Kalman filter. Groundwater levels were estimated by calculating 200 ensembles distributed over the grid. Subsequently, 10-day forecasts of groundwater levels were calculated by processing 50 ensembles of the Ensemble Prediction System (calculated by the European Centre for Medium-Range Weather Forecasts ECMWF) through the groundwater model.

6. ACKNOWLEDGMENTS

We are very grateful to our computer services staff for their commitment to this project. Without the innovative ideas and support of Rogier van de Pol, John Veldhuis and Sylvester Sleeman of TNO, we would not have succeeded in building the computational grid.

7. REFERENCES

Eklund, S.E. (2004), A massively parallel architecture for distributed genetic algorithms. *Parallel Computing*, 30, 647–676.

Foster, I. and C. Kesselman, Eds. (1999), *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers Inc.

Geist, A., A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam (1994), *PVM Parallel Virtual Machine, User's Guide and Tutorial for Networked Parallel Computing*, MIT Press, 279 pp., Cambridge, Mass.

Herrera, F., Lozano, M., Moraga, C. (1998), Hybrid distributed real-coded genetic algorithms. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.P. (Eds.), *Parallel Problem Solving from Nature (V)*. pp. 879–888.

Snavely, A., G. Chun, H. Casanova, R.F. van der Wijngaart, and M.A. Frumkin (2003), Benchmarks for grid computing: a review of ongoing efforts and future directions. *SIGMETRICS Perform. Eval. Rev.* (30), 4, 27-32.

Valstar, J.R., D.B. McLaughlin, C.B.M. te Stroet and F.C. van Geer (2004), A representer-based inverse method for groundwater flow and transport applications, *Water Resour. Res.*, 40, W05116, doi:10.1029/2003WR002922.

Vermeulen, P.T.M., C.B.M. te Stroet, and A.W. Heemink (2006), Limitations to upscaling of groundwater flow models dominated by surface water interaction, *Water Resour. Res.*, 42, W10406, doi:10.1029/2005WR004620.

Vrugt, J.A., B. Ó Nualláin, B.A. Robinson, W. Bouten, S.C. Dekker, and P.M.A. Sloot (2006), Application of parallel computing to stochastic parameter estimation in environmental models, *Computers & Geosciences*, 32(8), 1139-1155.