Domain decomposition in shallow lake modelling for operational forecasting of flooding

Menno Genseberger[1], Edwin Spee[2], and Lykle Voort[3]

1 Introduction

The Netherlands is a highly urbanized area. In addition to flooding from the sea due to storm surges and high water discharges from rivers, flooding from major lakes is also a threat. Since 2011 there is a new system in operational use (24 hours per day, 7 days per week), for the prediction of flooding at Lake IJssel, Lake Marken, and the lakes bordering them. This system, RWsOS Meren (Genseberger et al. [2013]) enables a real-time dynamic forecasting of wind driven waves, water flow, wave runup, and overtopping at dikes.

At the moment the time horizon of forecasts with RWsOS Meren is two days ahead. To enlarge this time horizon, medium-range global weather forecasts from ECMWF (ECMWF) up to 15 days (two forecasts per day) and short-to-medium range forecasts of extreme and localised weather events from COSMO-LEPS (limited area ensemble prediction system) (COSMO) up to $5,5$ days (one forecast per day) will be used as input for RWsOS Meren. In RWsOS Meren, only the two shallow-water models of the lakes will be run with this input (and not the models for waves, wave runup, and overtopping). ECMWF and COSMO-LEPS use ensembles (51 and 16 ensemble members, respectively). Therefore, also the two shallow-water models will be run in ensemble mode. As a consequence, for these models 204 runs with a simulation period of 15 days and 32 runs with a simulation period of 5.5 days have to finish within a reasonable time on a daily basis. This asks for a balance between low computational times per ensemble member and the efficient use of the available hardware (and energy) resources. In this paper we investigate how to manage this on current hardware.

Deltares, P.O. Box 177, 2600 MH Delft and CWI, P.O. Box 94079, 1090 GB, Amsterdam, The Netherlands `Menno.Genseberger@deltares.nl` · Deltares, P.O. Box 177, 2600 MH Delft, The Netherlands `Edwin.Spee@deltares.nl` · SURFsara, P.O. Box 94613, 1090 GP Amsterdam, The Netherlands `lykle.voort@surfsara.nl`

Here, the essential ingredient is the domain decomposition technique in the shallow-water solver Simona (Simona [2012], Vollebregt et al. [2003], Borsboom et al. [2014]) that we apply. The implementation of this domain decomposition technique in Simona has the nice property that it enables (sub)structuring, distribution, and minimizing the exchange of data in a practical and efficient way. This is both on the high – modelling level (decomposition in physical subdomains with absorbing boundary conditions), intermediate – numerical level (parallel solver with minimized iteration count) and low – implementation level (data distribution with minimized data exchange between different memory blocks). A lower level inherits the gain in efficiency from a higher level. Therefore, most gain is on the high level and on the lower levels some fine-tuning remains. However, gain in efficiency on the high level will not always automatically be there and some effort is needed. This will be illustrated here for the practical example of the shallow lake models in RWsOS Meren.

The paper is organized as follows. First, the physical characteristics and the shallow-water models of the lakes are described in § 2. Then, in § 3 we apply domain decomposition in Simona for these models in two stages (automatic partitioning in § 3.1 and fine-tuning in § 3.2). For this purpose, we investigate the consequences for computational times and (parallel) efficiency by numerical experiments.

2 Shallow lake modelling

The operational system RWsOS Meren (Genseberger et al. [2013]) covers eight major lakes of the Netherlands: Lake IJssel (IJsselmeer in Dutch), Lake Marken (Markermeer), and six smaller lakes at the borders (with Dutch names Ketelmeer, Vossemeer, Zwarte Meer, IJmeer, Gooimeer, and Eemmeer), see Fig. 1. All lakes are quite shallow: depths are in the order of several meters whereas horizontal dimensions are in the order of kilometers. Ketelmeer, Vossemeer, and Zwarte Meer are in open connection with Lake IJssel. IJmeer, Gooimeer, and Eemmeer are in open connection with Lake Marken. Lake Marken is separated from Lake IJssel by a dike ("Houtribdijk") with two sluices. On the north, Lake IJssel is separated from the Wadden Sea by a dike ("Afsluitdijk") with two sluices. Most important driving force of the water system is wind. However in specific situations, for instance after heavy rainfall, river discharges are also important. Here, the largest contribution is from the river IJssel that enters Ketelmeer. Furthermore, river Overijsselse Vecht enters Zwarte Meer (via river Zwarte Water) and river Eem enters Eemmeer. The water level of Lake IJssel is kept to a fixed level by draining off superfluous water via the two sluices to the Wadden Sea. Lake Marken is also kept to a fixed water level, however discharges through the sluices are much smaller.

Fig. 1 Geographical domain with eight major lakes of the Netherlands.

For computing flow of water, based on medium-range global and short-to-medium range weather forecasts, the same two models will be used as in the current operational system of RWsOS Meren. One is the shallow-water model for Lake IJssel including the smaller lakes Ketelmeer, Vossemeer, and Zwarte Meer and parts of the rivers IJssel, Zwarte Water, and Overijsselse Vecht. The other is the shallow-water model for Lake Marken including the smaller lakes IJmeer, Gooimeer, and Eemmeer and the river Eem with its floodplain. For rivers IJssel and Overijsselse Vecht boundary conditions are imposed through discharges. Close to the sluices on the side of the Wadden Sea boundary conditions are imposed through water levels. Here, both discharges and water levels are a combination of observed values and predicted values (from neighbouring operational systems). Wind predictions (as computed externally) are downscaled to the required sizes for the models of the lakes.

For the numerical solution of the shallow-water models Simona (Simona [2012], Vollebregt et al. [2003], Borsboom et al. [2014]) is being used. Simona applies a so-called alternating direction implicit (ADI) method to integrate the shallow-water equations numerically in time, using an orthogonal staggered grid with horizontal curvilinear coordinates. For this application, the shallow-water models are depth averaged. The sizes of the horizontal computational grids are 486×1983 and 430×614 for the shallow-water models of Lake IJssel and Lake Marken, respectively. See Fig. 2 and Fig. 4 for the corresponding geographical lay-out. The grids are relatively fine in (the floodplain areas of) the rivers and coarse in the larger lakes. For the shallow-water model of Lake IJssel this can be observed by comparing the geographical lay-out with the memory lay-out in Fig. 2.

Calibration and validation of the models was carried out for periods with historical storms (including typical wind behavior, some also with high river discharges). For this, measured values of discharges, waterlevels, rainfall, and evaporation were used with some small corrections due to missing terms in the water balance for the physical system.

Here we take a simulation period of 32 hours for both shallow-water models. Note that computational times of Simona are almost not influenced by the physical conditions in a given simulation period (storm or mild wind conditions and/or high or low river discharges). To get the computational times for an ECMWF (COSMO-LEPS) ensemble member with a simulation period of 15 (5.5) days the computational time has to be multiplied with a factor 11.25 (4.125).

Domain decomposition will be used to have a good balance between computational times and (parallel) efficiency for running ensembles with the two shallow-water models.

3 Domain decomposition

The domain decomposition technique in the current versions of Simona is based on a nonoverlapping Schwarz method with optimized coupling at the subdomain interfaces (Borsboom et al. [2014]). This approach has shown to yield excellent parallel performance for practical flow problems from civil engineering. However, the two shallow-water models have a complicated geometry and a relatively small number of computational grid points. Because of this, obtaining a good balance is not straightforward: increasing the number of subdomains can lower computational times more but may result in less efficient use of the available hardware (and energy) resources.

As we can not investigate all possibilities, we proceed with a pragmatic approach. First, we analyse the parallel performance for two automatic partitioning methods as a function of the number of computational cores in § 3.1. Then, for a nearly optimal number of subdomains from § 3.1, we try to get efficient ensemble runs with the models on current hardware by fine-tuning in § 3.2.

3.1 Automatic partitioning

Here we analyse the parallel performance of both shallow-water models by a numerical experiment. For this we varied the number of subdomains from 1 to 16 for two automatic partitioning methods. Here, one subdomain is assigned to one computational core. Both methods are based on domain decomposition of the active computational grid points. One method makes a stripwise partitioning in one direction of the domain. The other method decomposes the domain based on orthogonal recursive bisection (ORB) (Berger and Bokhari [1987]).

The numerical experiment was performed on the H4+ linux-cluster at Deltares (nodes interconnected with Gigabit Ethernet, each node contains 1 Intel quad-core i7-2600 processor "Sandy Bridge" ([van der Steen, 2011, § 2.8.5.3], [van der Steen, 2012, § 2.8.4.1]), 3.4 GHz / core, hyperthreading off) with the 2011 version of Simona (compiled with Intel Fortran 11 compiler and OpenMPI for Linux 64 bits platform). For the distribution of the memory blocks (each block contains the unknowns in one subdomain) over the nodes two options were considered: round-robin (memory blocks are distributed alternated over the nodes) and compact (option tries to position each memory block close to blocks of neighbouring subdomains).

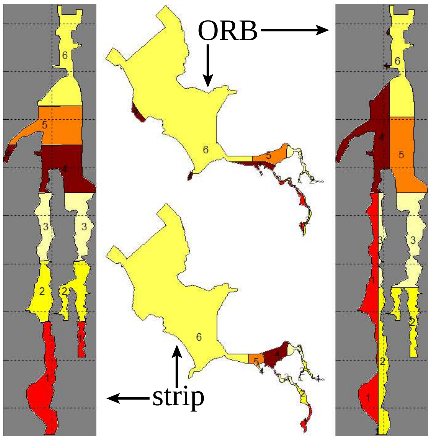

Fig. 2 Geographical and memory lay-out of computational grid of shallow-water model for Lake IJssel with automatic partitioning by domain decomposition. Middle bottom: geographical lay-out of domain decomposed in 6 subdomains (in different colours and numbered from 1 to 6) with stripwise partitioning, left: corresponding memory lay-out. Middle top: geographical lay-out of domain decomposed in 6 subdomains (in different colours and numbered from 1 to 6) with partitioning via orthogonal recursive bisection (ORB), right: corresponding memory lay-out.

Fig. 2 (Fig. 4) shows the corresponding geographical lay-out of the computational grid of the shallow-water model for Lake IJssel (Lake Marken) in case of 6 subdomains. The wall-clock time as a function of the number of computational cores for this model is shown on the left (right) in Fig. 3. Reported wall-clock times are averages of three measurements. For all cases the corresponding standard deviation is less than 3% of the average.

The speed up is not as ideal as linear (for that case lines will have a downward slope of 45° in the double logarithmic figures: doubling the number of computational cores will half the wall-clock time). But, in general, from Fig. 3 it can be observed that for both models the wall-clock time can be reduced substantially for decompositions in up to 6 subdomains. Based on this observation, we choose 6 as the nearly optimal number of subdomains for both models.

Furthermore, one of the automatic partitioning methods does not clearly seem to be more beneficial than the other (Fig. 3). This indicates the possibility to further optimize the decomposition by inspecting the configurations in 6 subdomains of both methods. That will be subject in § 3.2. Overall, the memory option compact improves the results of round-robin for more than 4 computational cores (i.e. the cases that more nodes are used). This is as expected: for option compact more neighbouring subdomains are positioned inside the same node, therefore there is less communication between nodes resulting in lower computational times.

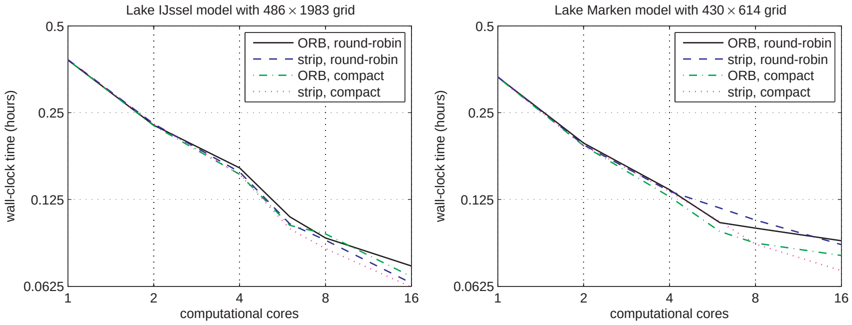

Fig. 3 Wall-clock time (in hours) for shallow-water model of Lake IJssel (left) and Lake Marken (right) as a function of the number of computational cores. Shown are results for two automatic partitioning methods: stripwise and ORB (orthogonal recursive bisection) and two options for memory distribution: round-robin and compact.

3.2 Fine-tuning

For the nearly optimal number of 6 subdomains for both models from § 3.1, we try to get efficient ensemble runs with the models on current hardware by fine-tuning.

We considered the following hardware at SURFsara:

- 2 socket L5640 node (2 Intel six-core Xeon L5640 processors "Westmere-EP" ([van der Steen, 2011, § 2.8.5.2]), 2.26 GHz / core) (Lisa),
- 2 socket 2650L node (2 Intel eight-core Xeon E5-2650L processors "Sandy Bridge" ([van der Steen, 2011, § 2.8.5.3], [van der Steen, 2012, § 2.8.4.1]), 1.8 GHz / core) (Lisa),

- 2 socket 2695 v2 node (2 Intel twelve-core Xeon E5-2695 v2 processors "Ivy Bridge" ([van der Steen, 2013, § 2.8.4]), 2.4 GHz / core) (Cartesius).

Note that, with 6 subdomains, multiple runs (2 runs for a 2 socket L5640 or 2650L node, 4 runs for a 2 socket 2695 v2 node) of the models fit in a single node. Instead of using more than one node for a single run to lower computational times more (like the numerical experiment in § 3.1), for efficiency we will consider here the use of a single node for multiple runs simultaneously. A 2013 version of Simona compiled with Intel Fortran 13 and OpenMPI for Linux 64 bits platform was used.

First, we try to further optimize the decomposition in 6 subdomains by inspecting the configurations of the two automatic partitioning methods from § 3.1. For that purpose we used the Visipart package of Simona. By comparing the geographical lay-out of subdomains for the shallow-water model of Lake Marken for the two automatic partitioning methods (left and middle picture) in Fig. 4 one can see that for the stripwise decomposition (left picture) there is a very long subdomain interface and a part of a subdomain is quite thin. This has a negative effect on the computational times. Relatively long subdomain interfaces require more data communication. Very thin subdomains with widths of less than a dozen grid cells affect the validity of the applied local optimized coupling in Simona. Therefore, we used the results of the other automatic partitioning method, by ORB (middle picture) as a basis for further optimization. The right picture of Fig. 4 illustrates the resulting geographical lay-out of subdomains for the shallow-water model of Lake Marken. In a similar way, the decomposition in 6 subdomains for the shallow-water model of Lake IJssel has been optimized. This strategy for further optimization is confirmed by the wall-clock times as shown in columns 2 (automatic stripwise partitioning), 3 (automatic partitioning with ORB), and 4 (fine-tuning of one of the automatic paritionings) of Table 1 (Lake IJssel) and Table 2 (Lake Marken). Here, the reported wall-clock times are averages of three measurements and the corresponding standard deviation is given after the ± symbol.

Then, with the further optimized decomposition we ran two models simultaneously on one single 2 socket L5640 and 2650L node. Corresponding wall-clock times are shown in column 5 of Table 1 (Lake IJssel) and Table 2 (Lake Marken). By comparing these times with column 4 (same decomposition but only one model run on the node) one can see there is some price to pay. We can relieve a part of this pain by binding one of the runs to 6 successive cores of socket 1 and the other run to 6 successive cores of socket 2 as shown in column 6 of both tables. Here data of each model stays inside one socket and no communication is needed between the sockets (this is somehow similar to the situation –with nodes instead of sockets– for memory option compact from § 3.1). On one single 2 socket 2695 v2 node we were not able to run multiple models without binding. For this type of node we observe from columns 4, 6, and 7 in the tables that they can be used efficiently for running 4 models simultaneously.

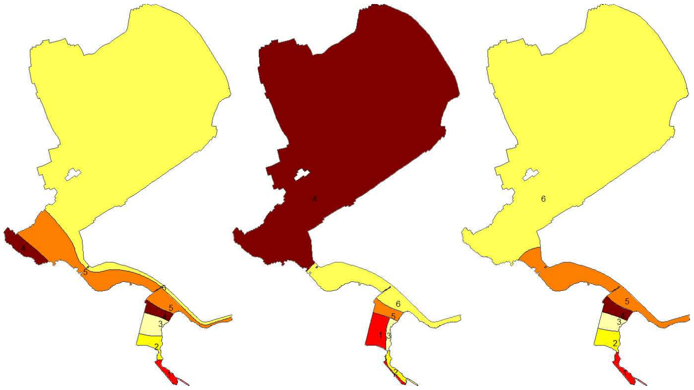

Fig. 4 Geographical lay-out of computational grid of shallow water-model for Lake Marken with partitioning by domain decomposition in 6 subdomains (in different colours and numbered from 1 to 6). Left: geographical lay-out of subdomains with automatic stripwise partitioning. Middle: geographical lay-out of subdomains with automatic partitioning via orthogonal recursive bisection (ORB). Right: geographical lay-out of subdomains with manual fine-tuning of the partitioning.

4 Conclusions

We studied how to run efficiently shallow-water models of an operational system for prediction of flooding at the borders of the major Dutch lakes. Aim is to combine the shallow-water models with short-to-medium weather ensemble forecasts to enlarge the time horizon. This asks for a balance between low computational times per ensemble member and the efficient use of the available resources on current hardware. Here, the essential ingredient is the domain decomposition technique in the applied shallow-water solver.

First, the parallel performance for two automatic partitioning methods of the shallow-water models was analyzed. Although the models have a complicated geometry and a relatively small number of computational grid points, the wall-clock time can be reduced substantially for decompositions in up to 6 subdomains. Then, for a nearly optimal partitioning, we tried to get efficient ensemble runs on current hardware by fine-tuning. The resulting optimized decompositions show relatively short internal interfaces between the subdomains (less communication needed) and subdomains that are not too thin (very thin ones affect the validity of the locally optimized domain decomposition coupling). Finally, multiple models can be run simultaneously in an efficient way on one 2 socket node of current hardware by binding the subdomains of each model to successive cores of one socket.

Acknowledgements We thank SURFsara (www.surfsara.nl) for their support in using the Lisa linux-cluster and the Dutch national supercomputer Cartesius. We acknowledge PRACE for awarding a part of this support.

Table 1 Wall clock time (in minutes) for shallow-water model of Lake IJssel on one two socket node for different decompositions in 6 subdomains and memory distributions.

decomp. type of	strip	ORB	fine-tuned	fine-tuned	fine-tuned	fine-tuned
# simultaneous runs on node	1	1	1	2	2	4
binding	no	no	no	no	yes	yes
2 socket L5640 node	8.07 ± 0.25	8.64 ± 0.16	7.83 ± 0.11	10.98 ± 0.24	9.79 ± 0.02	
2 socket 2650L node	6.98 ± 0.10	7.28 ± 0.06	6.41 ± 0.10	8.06 ± 0.12	7.44 ± 0.06	
2 socket 2695 v2 node	6.56 ± 0.00	7.08 ± 0.01	6.07 ± 0.01		6.15 ± 0.01	7.96 ± 0.01

Table 2 Wall clock time (in minutes) for shallow-water model of Lake Marken on one two socket node for different decompositions in 6 subdomains and memory distributions.

decomp. type	strip	ORB	fine-tuned	fine-tuned	fine-tuned	fine-tuned
# simultaneous runs on node	1	1	1	2	2	4
binding	no	no	no	no	yes	yes
2 socket L5640 node	8.02 ± 0.14	6.84 ± 0.19	6.75 ± 0.26	9.02 ± 0.29	8.26 ± 0.04	
2 socket 2650L node	7.19 ± 0.20	6.15 ± 0.04	5.85 ± 0.02	7.14 ± 0.32	6.62 ± 0.02	
2 socket 2695 v2 node	6.71 ± 0.01	5.70 ± 0.02	5.31 ± 0.00		5.35 ± 0.01	6.15 ± 0.02

References

M. J. Berger and S. H. Bokhari. A partitioning strategy for nonuniform problems on multiprocessors. *IEEE Transactions on Computers*, 36(5): 570–580, 1987.

M. Borsboom, M. Genseberger, B. van 't Hof, and E. Spee. Domain decomposition in shallow-water modelling for practical flow applications. In Erhel et al., editor, *Domain Decomposition Methods in Science and Engineering XXI*, pages 557–565. Springer, 2014.

COSMO. Consortium for small-scale modeling. [online]. `www.cosmo-model.org`.

ECMWF. European centre for medium-range weather forecasts. [online]. `www.ecmwf.int`.

M. Genseberger, A. Smale, and H. Hartholt. Real-time forecasting of flood levels, wind driven waves, wave runup, and overtopping at dikes around Dutch lakes. In F. Klijn and T. Schweckendiek, editors, *2nd European Conference on FLOODrisk Management, Comprehensive Flood Risk Management*, pages 1519–1525. Taylor & Francis Group, 2013.

Simona. *WAQUA/TRIWAQ - two- and three-dimensional shallow-water flow model.*, 2012. URL `http://apps.helpdeskwater.nl/downloads/extra/simona/release/doc/techdoc/waquapublic/sim1999-01.pdf`.

A. J. van der Steen. Overview of recent supercomputers. Technical report, 2011. Online at `http://www.euroben.nl/reports.php`. An earlier version of this report appeared in J. J. Dongarra and A. J. van der Steen. High-performance computing systems: Status and outlook. *Acta Numerica*, 21:379–474, 2012.

A. J. van der Steen. Overview of recent supercomputers. Technical report, 2012. See A. J. van der Steen, 2011.

A. J. van der Steen. Overview of recent supercomputers. Technical report, 2013. See A. J. van der Steen, 2011.

E. A. H. Vollebregt, M. R. T. Roest, and J. W. M. Lander. Large scale computing at rijkswaterstaat. *Parallel Computing*, 29:1–20, 2003.