Optimizing the coupling in parallel air quality model systems

M. Lieber and R. Wolke

Optimizing the coupling in parallel air quality model systems

Matthias Lieber[⋆], Ralf Wolke[⋆⋆]

Leibniz Institute for Tropospheric Research, Permoserstraße 15, 04318 Leipzig, Germany

Abstract

Today, parallel computers facilitate complex simulations of physical and chemical processes. To obtain more accurate results and to include multiple aspects of environmental processes, model codes of different scientific areas are coupled. An often used coupling strategy is to run the individual codes concurrently on disjoint sets of processors, as this keeps the codes mostly independent. However, it is important to improve the workload balance between the codes to achieve a high efficiency on parallel computers. In this paper, the parallel air quality model system LM-MUSCAT is presented. It consists of the chemistry-transport model MUSCAT and the meteorological model LM. Since an adaptive time step control is applied in MUSCAT the overall load fluctuates during runtime, especially at applications with highly dynamical behavior of the simulated processes. This causes load imbalances between both models and, consequently, an inefficient usage of the parallel computer. Therefore, an alternative coupling method is investigated. In this approach, all processors calculate alternately both models, whereby the load is distributed equally. Performance tests show that this "sequential" approach is well suited to increase the efficiency of coupled systems that have workload fluctuations in one or more models. In general, load variations can occur in models which use adaptive grid techniques or an adaptive step size control. Systems using such techniques can take benefit from the described coupling approach.

Key words: Parallel computing, Air quality model, Model coupling, M x N problem, Model data exchange, Sequential coupling, Parallel efficiency

[⋆] Present address: Center for Information Services and High Performance Computing, Dresden University of Technology, Germany.
E-mail address: Matthias.Lieber@tu-dresden.de (M. Lieber).
[⋆⋆]Corresponding author. Tel.: +49-341-235-2860.
E-mail address: wolke@tropos.de (R. Wolke).

1 Introduction

Nowadays, model systems consisting of two or more simulation models help scientists to investigate more and more aspects of complex systems. Interactions between the simulated processes of each single model can be considered to obtain results closer to reality. This trend is particularly noticeable in environmental sciences. The most prominent examples are global climate models, which typically consist of models for atmosphere, ocean, sea-ice, and land surface (Jacob et al., 2001; Collins et al., 2006; Jungclaus et al., 2006), thus achieving a high degree of complexity. Typical applications on the regional scale are climate/lake model systems (Leon et al., 2007), atmosphere/groundwater models (Chow et el., 2006), wildfire simulation models (Coen, 2005), and air quality models (Grell et al., 2005; Zhang et al., 2006; Cheng et el., 2007; San Jose et al., 2007). A frequently used technique is offline coupling, which means that the output data from one model is used to drive a second model. But this is not always sufficient, since interactions in both directions might be of interest, for instance the heat flux in ocean/atmosphere simulations. In such cases, online coupling is required, i. e. the models run simultaneously and exchange data periodically. More details about online coupling are discussed by Frickenhaus et al. (2001) and Jacob et al. (2005). In most cases, independently developed model codes are coupled. Therefore, the first step of implementation is to make the codes work together, i. e. synchronize them to each other, exchange data fields, and perform necessary transformations of the data. The more such systems are used for operational applications, the more it becomes important not only to couple model codes but to optimize the method of coupling to achieve the best performance on parallel computers.

The mesoscale chemistry-transport model MUSCAT (MUltiScale Chemistry Aerosol Transport) has been developed for investigations of pollutant dynamics in the atmosphere like sulfur dioxide emissions from power plants and the evolution of aerosol particles (Wolke et al., 2002, 2004). It is coupled with the meteorological code LM (Lokal-Modell), which is the operational regional forecast model of the German Weather Service (Steppeler et al., 2003). The coupler provides MUSCAT with meteorological fields like temperature, humidity, and density from LM. Moreover, a feedback is implemented whereby the aerosol particle distribution calculated by MUSCAT influences the aerosol optical thickness and, hence, the radiation budget in LM. In the original coupling scheme, both codes run parallel on their own predefined set of processors and have their own separate step size control. The analysis of the used coupling scheme in Sec. 3 shows that the adaptive step size control implemented in MUSCAT leads to a variable workload and consequently to load imbalances between the models. Therefore, in Sec. 4 an improved coupling scheme is investigated to optimize the parallel efficiency. The proposed approach is applicable to other model systems, which have load imbalances between their models as well. As a component of the new coupling scheme the self-contained library MDE (*Multiblock Data Exchange*) is introduced. It enables an efficient exchange of coupling fields between models that use different decompositions of the same three-dimensional basic grid. Concluding performance comparisons of the new implemented coupling scheme with the current one show that a higher parallel efficiency is achieved for some typical applications.

2 The air quality model system LM-MUSCAT

The LM is a non-hydrostatic limited-area meteorological model. It has been designed for both the operational numerical weather prediction and various scientific applications at the meso-β and meso-γ scale. The LM is based on the primitive thermo-hydrodynamic equations describing compressible flow in a moist atmosphere. The model equations are formulated in rotated geographical coordinates using a generalized terrain following height coordinate. A variety of physical processes (e. g. radiation, turbulence, clouds, and precipitation) are taken into account. For a more detailed description we refer to Steppeler et al. (2003) and the scientific documentation available at the COSMO website (2005).

The chemistry-transport code MUSCAT includes advection, turbulent diffusion, deposition, emission, and chemical reactions of gas phase species as well as aerosol dynamical processes of particles. These processes are described by three-dimensional mass balance equations:

$$\frac{\partial y}{\partial t} + \frac{\partial}{\partial x_1}(u_1 y) + \frac{\partial}{\partial x_2}(u_2 y) + \frac{\partial}{\partial x_3}(u_3 y) = \frac{\partial}{\partial x_3}(\rho K_z \frac{\partial y/\rho}{\partial x_3}) + Q + R(y) \qquad (1)$$

The vector y contains the predicted species concentrations and the variables of the aerosol particle size distribution. The term $R(y)$ represents the gas phase chemistry and the aerosol dynamical processes. Q stands for other time-dependent source and sink terms, like emissions, dry and wet deposition. The wind field (u_1, u_2, u_3) and the vertical diffusion coefficient K_z are computed simultaneously by the LM. The solution of systems of nonlinear ordinary differential equations, resulting from atmospheric chemistry transport problems, is numerically very expensive. Such systems are well-suited for parallelization by domain decomposition techniques.

The model system LM-MUSCAT is applied for the operational forecast of pollutants in regional areas and also for detailed studies of tropospheric processes. Gas phase processes, especially the formation of photooxidants as well as the transport and the transformation of particulate matter, can be simulated. The chemical reaction mechanisms are given in ASCII data files. All information required for the computation of the chemical term and the corresponding Jacobian is generated from this input file. Therefore, changes in the chemical mechanism can be performed in a simple and comprehensive way. Several gas phase mechanisms, e. g. RACM of Stockwell et al. (1997) with 73 species and over 200 reactions, are used in 3D case studies. Time resolved anthropogenic emissions are treated in the model as point, area and line sources. The different time evolution of several emitting groups is taken into account for the emission intensity. Biogenic emissions are parameterized in terms of land use type, temperature, and radiation. Dry and wet deposition processes are also included. For simulation of particulate matter, the size distribution and the aerosol dynamical processes (condensation, coagulation, sedimentation, and deposition) are described using a modal technique. The mass fractions of all particles within one mode are assumed to be identical. Particle size distribution changes owing to various mechanisms, which are divided into external processes like particle transport by convection and diffusion, deposition, and sedimentation as well as internal processes like condensation and coagulation. A more detailed description of MUSCAT is given by Wolke et al. (2002, 2004).

2.1 Grid decomposition

The meteorological model LM uses a rotated spherical grid with a hybrid vertical coordinate. To distribute the horizontal grid over all processors, it is decomposed into rectangular partitions with an as equal as possible number of grid cells. The MUSCAT grid is based on the LM grid, but is subdivided into so-called *blocks*, which can have different horizontal resolutions. This multiblock technique is used to reduce computational costs in less interesting boundary regions and to focus on certain regions of interest, like power plants and urban regions, with a finer resolution. For example, when plumes are injected into coarse grid cells, they are diluted immediately with the cell contents and the details of the near field chemistry are lost. The multiblock approach enables also a more efficient cache utilization of modern high performance computers since better data locality can be achieved by adjusting the block size. The partitions in MUSCAT are created by assigning blocks to processors. This assignment is determined by means of the grid-partitioning library ParMETIS (Karypis et al., 2003). It minimizes the length of partition border lines ("edge cut"), while balancing the number of grid cells of each processor. The more blocks are used for decomposition the finer the number of grid cells can be balanced. Fig. 1 shows an example of the MUSCAT multiblock structure with grid cells of different resolutions (a) and a partitioning of the grid (b).

2.2 Time integration in MUSCAT

For the time integration of the spatially discretized mass-balance equations, an implicit-explicit method is applied (Wolke and Knoth, 2000). Explicit second-order Runge-Kutta methods are used for the integration of the horizontal advection and an implicit method is applied for the remaining processes. The explicit time step is the same in all blocks and is chosen as a multiple of the constant LM time step under consideration of the CFL criterion to ensure the stability of the method. The processes within a column (vertical advection, turbulent diffusion, deposition, chemistry) are integrated with the implicit second-order BDF method (Backward Differentiation Formula). The nonlinear corrector iteration is performed by a Newton method in which the sparse linear systems are solved by linear Gauss-Seidel iterations. Alternatively, a direct sparse solver is implemented for the solution of linear equations (Wolke and Knoth, 2000). Both approaches utilize the special sparse block structure of the system. Therefore, the application of linear algebra libraries is not beneficial here. Due to the implemented error control, the length of the implicit time steps varies for different blocks. Shorter time scales of atmospheric processes require smaller time steps to maintain accuracy. For instance, large point emissions or local precipitation lead to smaller implicit time steps and, thus, to a higher workload of the corresponding block. This may cause load imbalances between the processors at runtime. To avoid this, a dynamic load balancing is implemented in MUSCAT, which periodically redistributes the blocks again by means of ParMETIS (Wolke et al., 2004). By using this technique, the load is well balanced for most of the applications.

2.3 Online coupling

In the old coupling scheme, both model codes run concurrently each on their own disjoint set of processors. In the following, this strategy is called *concurrent coupling*. The number of processors for meteorology and chemistry-transport (*processor ratio*) has to be defined at model startup. The codes are synchronized only for data exchange between LM and MUSCAT. This takes place each explicit time step (*couple time step*). Since this time step is chosen as a fraction of the CFL number, its length varies over the prediction time. Fig. 2 shows the coupling scheme. The bars on the time lines correspond to the time steps, which have different lengths in LM and in each MUSCAT block. The coupling scheme provides time-interpolated meteorological fields – except for wind fields, which were time-averaged to preserve mass balance. Therefore, LM has to calculate one couple time step in advance. This causes the feedback to reach LM "too late", which is neglected in most of the previous applications. The data exchange takes place as follows: Since the LM solves a compressible version of the model equations, with the pressure as prognostic variable, mass conservation is not ensured. This can produce “artificial” sources or drains of some species in the chemistry-transport model. Therefore, an additional step is necessary in which the wind fields are modified such that a discretized continuity equation is satisfied. The main task of this adjustment is the solution of an elliptic equation by a preconditioned conjugate gradient method. This is also done in parallel on the LM processors. The meteorological fields are sent from the LM processors to MUSCAT utilizing the Message Passing Interface (MPI). For each of the data fields and each overlap of LM and MUSCAT partitions an MPI message is exchanged. The MUSCAT processors transform the received data into the multiblock grid. Due to the different possible resolutions, this is done by averaging or interpolating. For some applications, a feedback from chemistry-transport to meteorology is implemented. For instance, the simulated aerosol properties are directly used for the radiation calculations in LM instead of climatological input values. This can significantly change the energy budget and, therefore, the atmospheric dynamics in the simulations (Heinold et al., 2007). The feedback data exchange takes place directly after the data transfer from LM to MUSCAT. In this case, the transformation into the LM grid is done by the MUSCAT processors before the transfer.

3 Performance issues of online coupled models

The concurrent coupling scheme is the method of coupling stand-alone parallel models with the least coding effort. Both models can still be started as separate programs and keep a maximum of independence. Note that most of the available coupling environments like MpCCI (2005) or OASIS (Valcke et el., 2004) are based on this approach. However, performance problems can occur in some model systems. These will be discussed generally at first. By means of a selected scenario it is investigated how far performance problems emerge in the coupling of LM-MUSCAT.

3.1 Load balance

The most important criterion for optimal performance of concurrently coupled model systems is – apart from the optimal performance of the individual codes – the load balance between the models. The number of processors for each model code in the parallel coupled system is crucial for the load balance (Eltgroth et al., 1997; Drake et al., 2005). If the load ratio of the codes differs from the chosen processor ratio, some processors will arrive earlier at synchronization points than others. This leads to processor idle time and, in consequence, to undesirable loss of performance. Load imbalances between the models can occur due to a variation of the workload of one or more models as well. Highly dynamical processes simulated by models with adaptive step size control or adaptive grid techniques (Berger and Colella, 1988; Steensland, 2001) can lead to such fluctuations. Other reasons are variable sizes of regions, at which additional calculations are required. For instance, Wilhelmsson (2002) shows that the ice coverage in ocean models highly influences the runtime. Several causes for workload variations in global climate models are discussed by Michalakes (1991). Idle times can also arise in systems, where complete models are activated after a required start-up period only. Even if the workload of the models is constant over the whole simulation time, problems can arise with the appropriate partitioning of the processors. It depends on the specific set up of the model run, the used computer system, and the total number of utilized processors (due to different scalability properties of the coupled models). This makes the estimation of the processor ratio in advance a hard choice for scientists.

3.2 Data exchange

Another important aspect of an efficient coupler is the method of data field exchange between the models. This is often stated as the "M×N"problem (Jacob et al., 2005), which denotes the problem of transferring data distributed on M processors to N processors with different data decomposition and different data structures. The task of the coupler is to find the data needed by processors of one model in the processors of the other model, transfer this data to the requesting processors, and transform the data into their data structures, which may have a different numerical grid or different resolution. Common implementations use either an intermediate coupler process between the models, which knows about the different decompositions and data structures (Jacob et al., 2005), or direct data transfer between the model processors (Larson et al., 2005). The first case is the more flexible, especially when more than two model codes are coupled. The models need an interface to the coupler process only and do not need to care about other models involved. On the other hand, the direct transfer is more efficient since data are directly sent from source to destination, which avoids the overhead of an intermediate process.

3.3 Coupler performance of LM-MUSCAT

To assess the load balance of LM-MUSCAT, the CPU time per couple time step of both models is analyzed for several scenarios. As an example, the results are presented for one selected scenario ("Europe", see Sec. 5). Since LM and MUSCAT are well load balanced within their own processors, it is reasonable to determine only the CPU time of each model, not of each processor. Fig. 3 shows a plot of the CPU time during the first 24 hours of prediction time. As can be seen, the CPU time of MUSCAT has intensive fluctuations, which reflect one course of a day. The ratio between minimum and maximum is approximately 1:4. The first peak at about 4 hours is caused by sunrise, which speeds up atmospheric chemistry. The increase of computational costs over the daytime results from a higher vertical diffusion in the atmospheric boundary layer. On the other side, the meteorological model LM has less workload variations. The peaks are caused by a time consuming module, which hourly updates the radiation budget. The two levels at about $7\,s$ and $5\,s$ arise from changes in the length of the couple time step, which is determined as a multiple of the constant LM time step under consideration of the CFL criterion.

We can summarize the following problems of the implemented coupler in LM-MUSCAT, which lead to load imbalances and complicate an efficient usage of parallel computers:

- Usually, the load ratio of LM and MUSCAT is unknown for new applications of the model system. Hence, the optimal processor ratio can not be determined a priori. It has to be found empirically.
- Due to the applied step size control and the dynamics of the underlying processes, the overall load in MUSCAT varies over the prediction time. Matching the load ratio of the models to the (constant) processor ratio is impossible.
- To initialize the meteorological conditions, only the LM is run without MUSCAT for a predefined period of time. During this startup phase, the MUSCAT processors idle.

Depending on the application, the workload balance between the codes can show a different behavior. For instance, fewer variations are noted for scenarios with a reduced number of chemical reactions. In this case, matching the CPU load of LM and MUSCAT is roughly possible. The described performance problems are typical for model systems coupled by the concurrent approach. Therefore, an improved coupling scheme is developed and implemented in LM-MUSCAT (Lieber, 2005).

4 Concept of an optimized coupler

4.1 The sequential coupling scheme

The implemented coupling scheme is based on the idea of a *sequential* scheduling of the model codes (Bettge et el., 2001). In contrast to the concurrent coupling, in the sequential approach each model runs on all available processors. Each processor is assigned to perform one partition of the coupled codes alternately. Since the workload of each model code is distributed equally over all processors, imbalances between the model codes are compensated. Benefit of sequential coupling can be always expected on architectures where the single codes are well balanced. Both discussed coupling schemes are illustrated in Fig. 4. In the sequential coupling cycle (Fig. 4 a) firstly a couple time step of model A is calculated on all processors and then data are exchanged from A to B. In the next stage model B is run on all processors and feedback data exchange is carried out. If the models run concurrently (Fig. 4 b), before start the used processors are divided into one group for model A and one for model B. Both groups need to be synchronized for coupling, which may cause processor idle time.

An essential advantage of the sequential scheme is that the a priori partitioning of the processors is not necessary. Another benefit is the possibility of reducing the MPI communication when exchanging coupling data fields. Intersections between LM and MUSCAT data on the same processor can be copied directly without any inter-processor communication. Depending on the way of implementation, this method may also reduce the size of MPI communication buffers and, thus, saving memory. In the ideal case, the same data decomposition is used in both models. For instance, Jacob et al. (2001) use this approach in the global climate model FOAM to reduce communication costs. Of course, this is only applicable if the model grid structures support it. Note that the overall number of partitions does not rise with the processor number only, but also with the number of models in the coupled system. A loss of parallel efficiency is expected, if one of the model codes scales poorly. For instance, consider both schemes for a coupled system of two models running on 256 CPUs. With the concurrent scheme, assuming a load ratio of 1:1, the models run on disjoint sets of 128 CPUs only. But when using the sequential scheme, both models utilize all 256 CPUs, which requires a better scalability of the models. Usually, atmospheric models achieve less parallel efficiency when running on more CPUs (Skålin, 1997; Michalakes et al., 2004), so that the sequential scheme can reduce the efficiency of the whole coupled system. Consequently, better performance results can be obtained only in cases where the benefits of the sequential approach compensate this disadvantage.

4.2 Multiblock Data Exchange

To provide a general interface for the exchange of coupling fields between simulation models using rectangular grids, the library Multiblock Data Exchange (MDE) has been developed. It is written in Fortran 90 and uses MPI for communication. MDE hides the programmer tasks of finding overlapping partitions, inter-process data exchange, and data exchange by direct copy within a process ("M×N" data exchange). The problem of finding overlapping partitions is illustrated in Fig. 5. The processors need to know which subset of the own partition has to be transferred to which other processor. To obtain this information, the processors need to exchange the location of their partitions and determine the overlap between their own and each other partition. The data of the overlapping regions are then transferred directly between the processors.

The general concept of MDE is the exchange of floating point arrays, defined on a global three-dimensional grid, between parallel processes. Every process requests and offers data of subsets of the global grid. These subsets are defined by a list of cuboids (blocks), which enables the definition of non-rectangular partitions as required in MUSCAT. The two main steps of data exchange by means of MDE are:

(1) For each block, a derived type with the position in the global grid, a data field identifier (an integer number to distinguish multiple data fields), the direction of communication (send or receive), and the pointer to the array containing the data has to be filled. The processes pass an array of this type containing their local block definition to the routine `mde_set_blocks`. Within MDE each process sends its own block description to all other processes and creates a list of intersections of local send-blocks with the receive-blocks of other processes having the same field identifier and vice versa. Also intersections of local send-blocks and local receive-blocks are determined. This routine can be considered as the core of MDE, as it defines the communication structure.

(2) The actual data exchange is performed by `mde_exchange`. The basic principle is shown in Fig. 6. The data in the list of intersections are put into one contiguous buffer for each receiver (step 1). Then the buffers are transferred via MPI's immediate send routine (step 2). Finally, the received data are copied from buffers to the data fields of the model (step 3). The buffering ensures, that for n processes a maximum number of $n(n-1)$ messages are exchanged. This method reduces message passing overhead and communication costs, especially on distributed memory architectures. Intersections of local blocks are copied without inter-processor communication. Once the communication structure is set up, multiple calls of `mde_exchange` are possible. Only changes in the grid decomposi-

tion require further calls of mde_set_blocks.

The communication structure created by the collective call mde_set_blocks is stored in a so-called *MDE communicator* object. By creating multiple of such communicators, several data exchange relations between different groups of processes can be defined. This allows the coupling of more than two models or the temporal separation of data exchange from model A to model B and the feedback from model B to model A. Benefits of using MDE are a strict separation between the data (program) and the algorithm of transmission (MDE), the complete hiding of the "M×N" data exchange from the programmer, and fast communication by exchanging as few messages as possible.

MDE has been developed for coupling of LM-MUSCAT. Nevertheless, it is flexible enough to be used in other parallel programs that use data decomposition techniques. In comparison to MCT (Larson et al., 2005), MDE does not perform interpolation between different grids and is restricted to exchange floating point numbers only. MDE defines no own data structures for coupling fields like MCT's AttributeVector. Instead, MDE assumes the use of arrays with at most 3 dimensions, which can be directly passed to the library. Due to low abstraction, MDE is clearer to use but offers less flexibility than MCT. The subroutine mde_set_blocks can be compared to MCT's Router initialization routine ("handshaking"). MCT's data transfer between disjoint sets of processes (Send and Recv) and data transfer within a group of processes (Rearrange) are unified by the mde_exchange subroutine.

5 Implementation in LM-MUSCAT

The sequential coupling scheme is implemented as an option to the concurrent scheduling. In the sequential approach, all processors first calculate the meteorology over one coupling interval. Then the meteorological coupling data are exchanged and all processors continue with the calculation of chemistry-transport over the same interval. Required arrays for feedback are sent from MUSCAT to LM, before the next coupling step is performed. To increase communication speed and reduce message buffer usage, the same domain decomposition in both models can be applied. But this option is only available, if LM and MUSCAT use the same grid resolution. In this case, each processor has the same subset of the grid in LM and MUSCAT, so that no inter-processor communication takes place when exchanging coupling data. This transfer is performed by the library MDE for sequential and concurrent coupling. It detects overlapping partitions itself so that no extra configuration is necessary to enable the "intra-process" data exchange. The subroutine mde_set_blocks needs to be called once at startup and after every repartitioning of MUSCAT.

The performance of LM-MUSCAT is investigated on an IBM p690 cluster utilizing up to 4 nodes consisting of 32 processors each. The parallel efficiency of the new sequential coupling scheme is compared with the concurrent coupling. For this comparison two scenarios with very different characteristics have been chosen for testing purposes:

- The “Europe” scenario has been utilized to supply boundary values for a scenario in a nested region. The model region comprises central Europe. Since a multitude of chemical reactions are considered and a refined grid is used, the main workload is located in MUSCAT. As shown in Sec. 3.3, the load fluctuations in MUSCAT are very strong.
- The “Samum” scenario is used for investigations of the influence of Saharan dust particles on the radiation budget (Heinold et al., 2007). Only the emission, transport, and deposition of dust particles without aerosol dynamical processes are considered in the chemistry-transport model. A uniform grid of 150 × 150 horizontal cells is used in LM and MUSCAT. In contrast to the “Europe” scenario, the main computational load is located in the meteorological model and only small workload variations in MUSCAT can be observed.

As presumed, a comparison of simulation results of the two coupling schemes shows only marginal differences mainly caused by the conjugate gradient iterations for the adjustment of wind fields. The implemented pre-conditioner depends on the domain decomposition of the LM grid and, therefore, from the number of LM processors. Note that for the “Europe” scenario, all runs with the same LM processor number produce exactly the same results. Small additional differences appearing in the “Samum” runs originate from the fact, that a chronological offset of feedback occurs in concurrent mode only, but not in sequential mode (see Sec. 2.3).

The parallel efficiency for both coupling schemes is compared for the “Europe” (Fig. 7) and “Samum” scenario (Fig. 8). In summary, the sequential coupling scheme is the more efficient one. The lower efficiency of the concurrent approach is due to load imbalances caused by temporal load variations in MUSCAT. However, the concurrent scheme is better for the “Samum” scenario when using larger processor numbers. This may be caused by the weaker temporal load variations of MUSCAT so that the main advantage of the sequential approach is less effective in this case. Instead, the issues of scalability discussed in Sec. 4.1 lead to a higher efficiency of the concurrent coupling. Moreover, a similar behavior of the schemes is observed for both very different applications. The insufficient performance at low processor numbers is typical for the concurrent scheme. Here, it is not possible to adjust the processor ratio accurately to the average load ratio of the models. Note that for the presented performance measurements with concurrent coupling several runs with the same overall processor number have been performed to find the optimal ratio of LM and MUSCAT processors. Therefore, the shown results of the concurrent coupling can not be expected to be reached for real applications.

The workload percentage of the most time-consuming LM-MUSCAT components for both discussed scenarios is shown in Fig. 9. The differences can be seen clearly: For the “Europe” scenario most time is spent in MUSCAT, whereas the meteorology and the adjustment of wind fields are the dominant components for the “Samum” scenario. This is clear, as “Samum” includes no chemistry simulation. Fig. 9 also shows that the coupler scales well at both scenarios. The workload fraction of the coupler is about 1% and 4% for “Europe” and “Samum” scenario, respectively. The difference is due to more coupling data of the “Samum” scenario. One can also see from the figure that LM and MUSCAT have a nearly similar scaling. However, the writing of MUSCAT output files scales poorly. A sequential method is used, which takes the more time, the more processors are involved. Therefore, an implementation based on MPI-2 I/O promises better performance. This is currently under development.

6 Conclusion and software availability

The sequential coupling scheme is an appropriate method to increase the performance of model systems with high workload variation in one or more of the single models. The portable library MDE supports the efficient implementation of this scheme. Overlapping partitions of different models within one processor are detected automatically by MDE whereby their coupling data are copied locally, which reduces inter-processor communication. Through the implementation of the sequential coupling scheme in the air quality model system LM-MUSCAT, promising performance improvements are achieved. At scenarios with “LM only” startup phase, no idle MUSCAT processors consume CPU time. Scientists benefit from the simplified model startup, since the processor numbers do not have to be defined a priori.

Further developments in LM-MUSCAT will include microphysical and multiphase chemical cloud processes, which are usually much more heterogeneous in time and space. Therefore, dynamic data structures and new strategies for load balancing of the cloud model are required for an efficient implementation. The model MUSCAT and the library MDE were developed at the Institute for Tropospheric Research, Leipzig. Both codes are written in Fortran 90 utilizing MPI-1 for parallelization. For code accessibility the corresponding author should be contacted.

Acknowledgements

The work was supported by the NIC Jülich. Furthermore we thank Michael Jung, HTW Dresden, and the German Weather Service for their cooperation.

References

Berger, M.J., Colella, P., 1988. *Local adaptive mesh refinement for shock hydrodynamics.* J. Comput. Phys. 82, 64–84.

Bettge, T., Craig, A., James, R., Wayland, V., Strand, G., 2001. *The DOE Parallel Climate Model (PCM): The Computational Highway and Backroads.* In: Alexandrov, V.N., Dongarra, J.J., Juliano, B.A., Renner R.S., Tan, C.J.K. (Eds.), *ICCS 2001.* Springer, 149–158.

Cheng, S., Chen, D., Li, J., Wang, H., Guo, X, 2007. *The assessment of emission-source contributions to air quality by using a coupled MM5-ARPS-CMAQ modeling system: A case study in the Beijing metropolitan region, China.* Environ. Modell. Softw., In Press.

Coen, J.L., 2005. *Simulation of the Big Elk Fire using coupled atmosphere-fire modeling.* Int. J. Wildland Fire 14, 49–59.

Chow, F.K., Kollet, S.J., Maxwell, R.M., Duan, Q., 2006. *Effects of soil moisture heterogeneity on boundary layer flow with coupled groundwater, land-surface, and mesoscale atmospheric modeling.* 17th Symposium on Boundary Layers and Turbulence, San Diego, CA.

Collins, W.D., Bitz, C.M., Blackmon, M.L., Bonan, G.B., Bretherton, C.S., Carton, J.A., Chang, P., Doney, S.C., Hack, J.J., Henderson, T.B., Kiehl, J.T., Large, W.G., McKenna, D.S., Santer, B.D., Smith, R.D., 2006. *The Community Climate System Model Version 3 (CCSM3).* J. Climate 19, 2122–2143.

COSMO (Consortium for Small-Scale Modelling), 2005. *Documentation of the LM Package, 2nd version.* `http://www.cosmo-model.org/public/documentation.htm`.

Drake, J.B., Jones, P.W., Carr, G.R., 2005. *Overview of the Software Design of the CCSM.* Int. J. High Perf. Comput. Appl. 19, 177–186.

Eltgroth, P.G., Bolstad, J.H., Duffy, P.B., Mirin, A.A., Wang, H., Wehner, M.F., 1997. *Coupled Ocean/Atmosphere Modeling on High-Performance Computing Systems.* In: *PPSC 1997.* SIAM.

Frickenhaus, S., Redler, R., Post, P., 2001. *Parallel coupling of regional atmosphere and ocean models.* In: Zwieflhofer, W., Kreitz, N. (Eds.), *Developments in Teracomputing*, World Scientific Publishing, 201–213.

Grell, G.A., Peckham, S.E., Schmitz, R., McKeen, S.A., Frost, G., Skamarock, W.C., Eder, B., 2005. *Fully coupled "online" chemistry within the WRF model.* Atmos. Environ. 39, 6957–6975.

Heinold, B., Helmert, J., Hellmuth, O., Wolke, R., Ansmann, A., Marticorena, B., Laurent, B., Tegen, I., 2007. *Regional Modeling of Saharan Dust Events using LM-MUSCAT: Model Description and Case Studies.* J. Geophys. Res. 112, D11204.

Jacob, R., Schafer, C., Foster, I., Tobis, M., Anderson, J., 2001. *Computational Design and Performance of the Fast Ocean Atmosphere Model, Version One.* In: Alexandrov, V.N., Dongarra, J.J., Juliano, B.A., Renner R.S., Tan, C.J.K. (Eds.), *ICCS 2001.* Springer, 175–184.

Jacob, R., Larson, J., Ong, E., 2005. *$M x N$ Communication and Parallel Interpolation in Community Climate System Model Version 3 Using the Model Coupling Toolkit.* Int. J. High Perf. Comput. Appl. 19, 293–307.

Jungclaus, J.H., Keenlyside, N., Botzet, M., Haak, H., Luo, J.J., Latif, M., Marotzke, J., Mikolajewicz, U., Roeckner, E., 2006. *Ocean Circulation and Tropical Variability in the Coupled Model ECHAM5/MPI-OM.* J. Climate 19, 3952–3972.

Karypis, G., Schloegel, K., Kumar, V., 2003. *ParMETIS: Parallel graph partitioning and sparse matrix ordering library (Version 3.1).* Technical Report, University of Minnesota.

Larson, J., Jacob, R., Ong, E., 2005. *The Model Coupling Toolkit: A New Fortran90 Toolkit for Building Multiphysics Parallel Coupled Models.* Int. J. High Perf. Comput. Appl. 19, 277–292.

Leon, L.F., Lam, D.C.L., Schertzer, W.M., Swayne, D.A., Imberger, J., 2007. *Towards coupling a 3D hydrodynamic lake model with the Canadian Regional Climate Model: Simulation on Great Slave Lake.* Environ. Modell. Softw. 22, 787–796.

Lieber, M., 2005. *Die Optimierung der Kopplung von Simulationsmodellen mit unterschiedlichen Gitterstrukturen auf Parallelrechnern.* Diplomarbeit, Hochschule für Technik und Wirtschaft Dresden.

Michalakes, J.G., 1991. *Analysis of Workload and Load Balancing Issues in the NCAR Community Climate Model.* Technical Report ANL/MCS-TM-144, Argonne National Laboratory.

Michalakes, J., Dudhia, J., Gill, D., Henderson, T., Klemp, J., Skamarock, W., Wang, W., 2004. *The Weather Research and Forecast Model: Software architecture and performance.* In: *11th ECMWF Workshop on HPC in Meteorology.* 156–168.

MpCCI Technical Reference (MpCCI 3.0). 2005. Fraunhofer Institute for Algorithms and Scientific Computing SCAI, Sankt Augustin.

San Jose, R., Perez, J.L., Gonzalez, R.M., 2007. *An operational real-time air quality modelling system for industrial plants.* Environ. Modell. Softw. 22, 297–307.

Skålin, R., 1997. *Scalability of Parallel Gridpoint Limited-Area Atmospheric Models. Part II: Semi-Implicit Time-Integration Schemes.* J. Atmos. Ocean Tech. 14, 442–455.

Steensland, J., 2001. *Dynamic Structured Grid Hierarchy Partitioners Using Inverse Space-Filling Curves.* Technical Report 2001-002, Department of

Information Technology, Uppsala University.

Steppeler, J., Doms, G., Schättler, U., Bitzer, H.W., Gassmann, A., Damrath, U., Gregoric, G., 2003. *Meso-gamma scale forecasts using the nonhydrostatic model LM.* Meteorol. Atmos. Phys. 82, 75–96.

Stockwell, W.R., Kirchner, F., Kuhn, M., Seefeld, S., 1997. *A new mechanism for regional atmospheric chemistry modeling.* J. Geophys. Res. D22, 102, 25847–25879.

Valcke, S., Caubel, A., Vogelsang, R., Declat, D., 2004. *OASIS3 User Guide (oasis3_prism_2-4).* PRISM-Report No 2, 5th Ed. CERFACS, Toulouse.

Wilhelmsson, T., 2002. *Parallelization of the HIROMB Ocean Model.* Licentiate Thesis, Royal Institute of Technology, Stockholm.

Wolke, R., Knoth, O., 2000. *Implicit-explicit Runge-Kutta methods applied to atmospheric chemistry-transport modelling.* Environ. Modell. Softw. 15, 711–719.

Wolke, R., Knoth, O., Renner, E., Schröder, W., Weickert, J., 2002. *Modelling of atmospheric chemistry-transport processes.* In: Rollnik, H., Wolf, D. (Eds.), *NIC Symposium 2001.* NIC Jülich, 453–462.

Wolke, R., Knoth, O., Hellmuth, O., Schröder, W., Renner, E., 2004. *The Parallel Model System LM-MUSCAT for Chemistry-Transport Simulations: Coupling Scheme, Parallelization and Applications.* In: Joubert, G.R., Nagel, W.E., Peters, F.J., Walter, W.V. (Eds.), *ParCo 2003.* Elsevier, 363–370.

Zhang, Y., Liu, P., Pun, B., Seigneur, C., 2006. *A comprehensive performance evaluation of MM5-CMAQ for the Summer 1999 Southern Oxidants Study episode–Part I: Evaluation protocols, databases, and meteorological predictions.* Atmos. Environ. 40, 4825–4838.

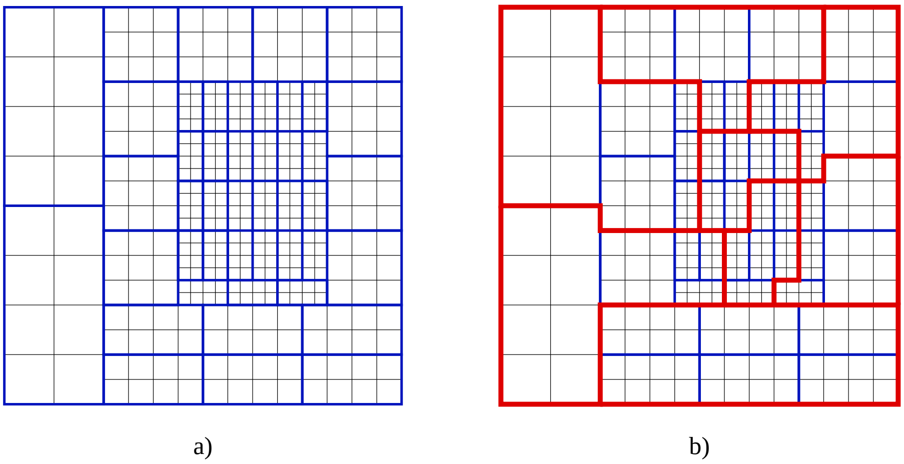

Figure 1. Horizontal MUSCAT grid: a) block structure, b) partitioning.

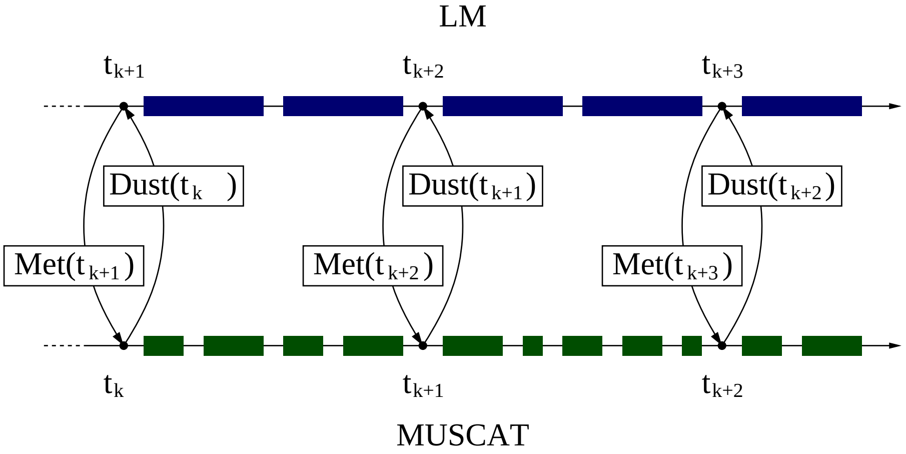

Figure 2. LM-MUSCAT coupling scheme. Bars on the time lines represent time steps of constant length (LM) and varying length (MUSCAT).

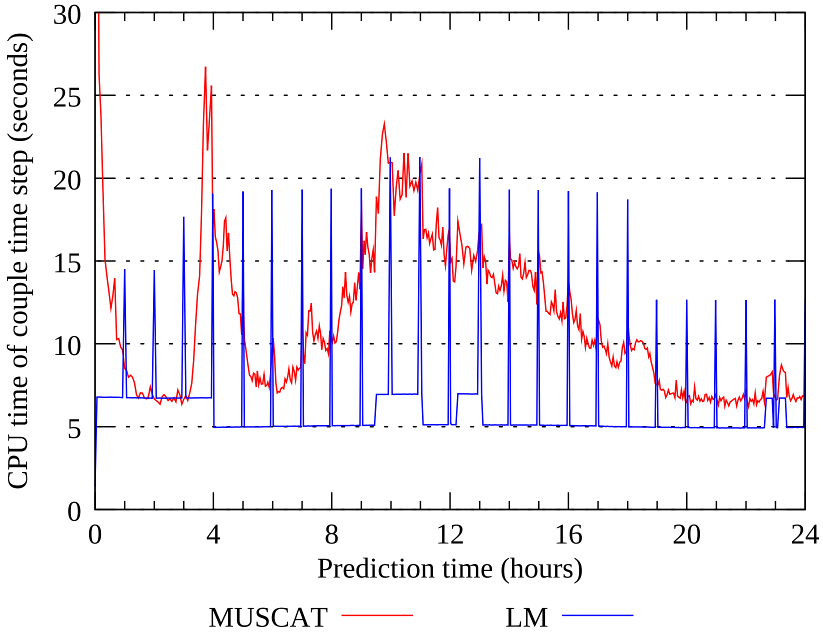

Figure 3. Analysis of the CPU time of LM and MUSCAT.

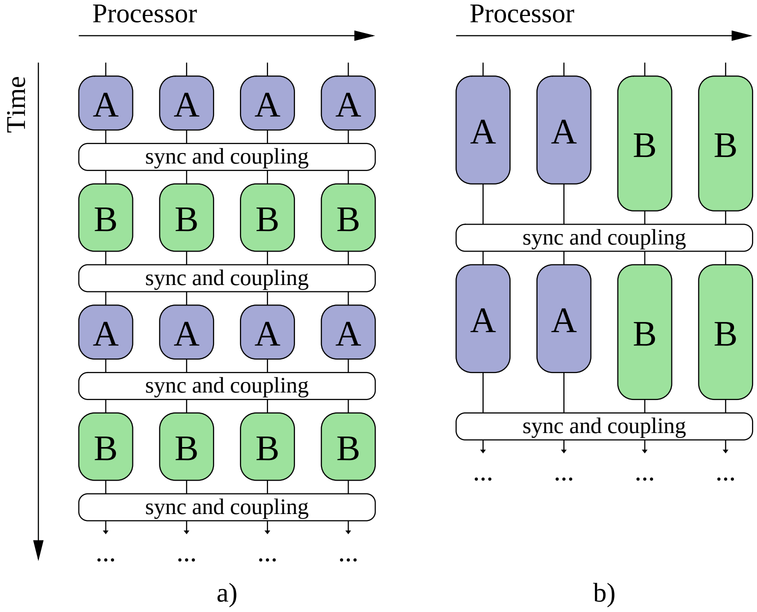

Figure 4. Coupling schemes for model systems: a) sequential, b) concurrent. The letters A and B represent one model code each.

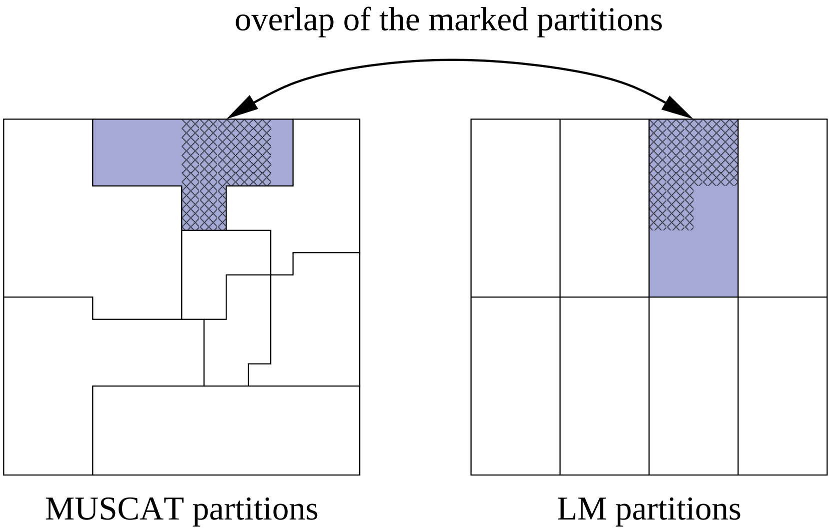

Figure 5. Overlapping partitions illustrated for one MUSCAT partition/LM partition pair.

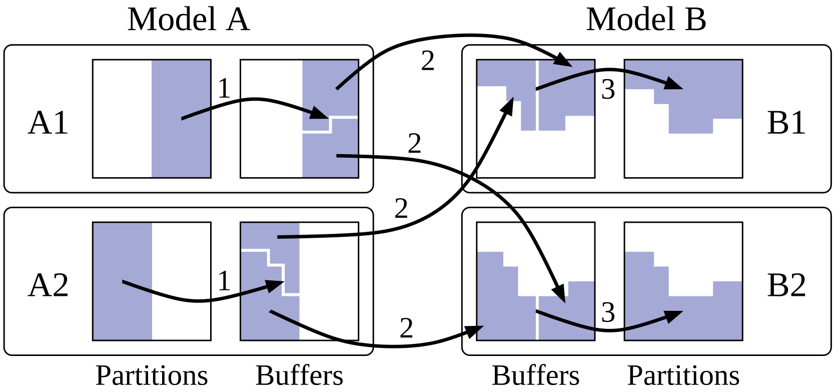

Figure 6. MDE data transfer principle. Example for two models A and B running concurrently on two processors each.

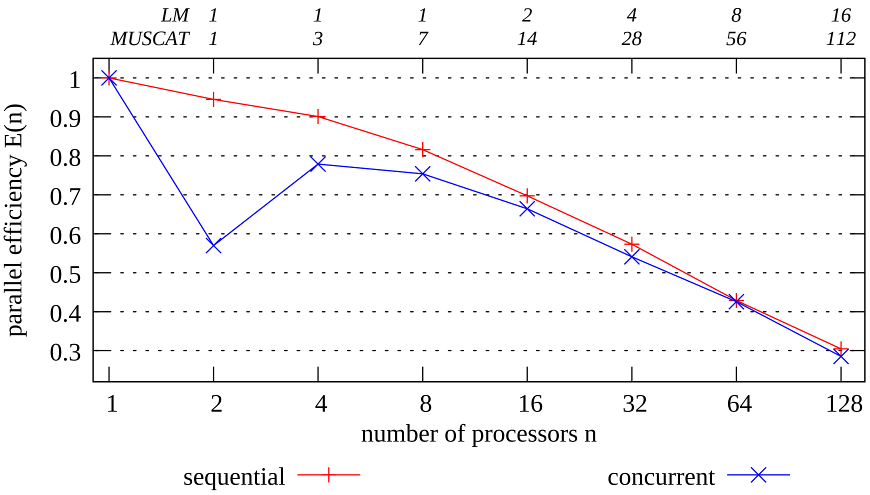

Figure 7. Parallel efficiency of LM-MUSCAT with concurrent and sequential scheduling. "Europe" scenario. The table on top indicates the processor partitioning used for concurrent scheduling.

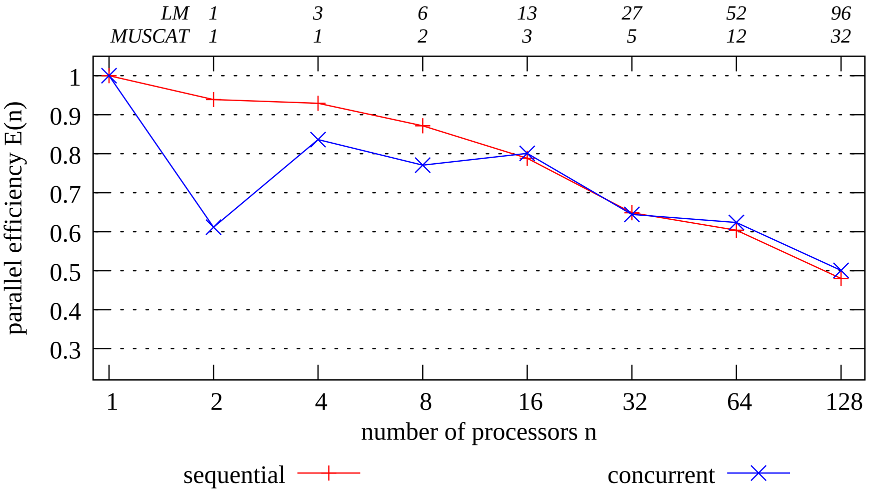

Figure 8. Parallel efficiency of LM-MUSCAT with concurrent and sequential scheduling. "Samum" scenario. The table on top indicates the processor partitioning used for concurrent scheduling.

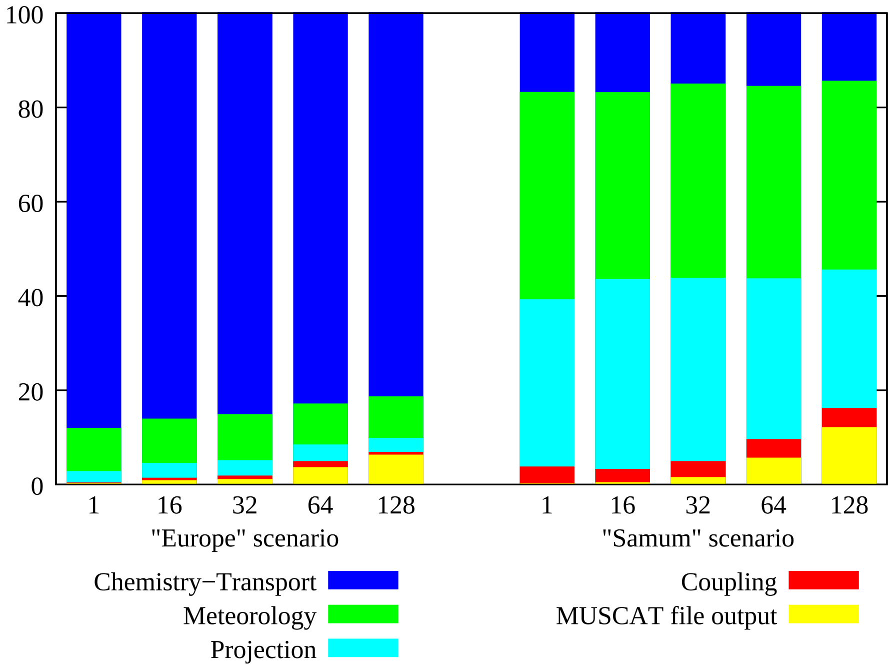

Figure 9. Workload percentage of LM-MUSCAT components for different processor numbers using the sequential coupling scheme.