Highly Scalable Dynamic Load Balancing in the Atmospheric Modeling System COSMO-SPECS+FD4

Matthias Lieber[1], Verena Grützun[2], Ralf Wolke[3], Matthias S. Müller[1], and Wolfgang E. Nagel[1]

[1] Center for Information Services and High Performance Computing, TU Dresden, 01062 Dresden, Germany
matthias.lieber@tu-dresden.de
[2] Max Planck Institute for Meteorology, Hamburg, Germany
[3] Leibniz Institute for Tropospheric Research, Leipzig, Germany

Abstract. To study the complex interactions between cloud processes and the atmosphere, several atmospheric models have been coupled with detailed spectral cloud microphysics schemes. These schemes are computationally expensive, which limits their practical application. Additionally, our performance analysis of the model system COSMO-SPECS (atmospheric model of the Consortium for Small-scale Modeling coupled with SPECtral bin cloud microphysicS) shows a significant load imbalance due to the cloud model. To overcome this issue and enable dynamic load balancing, we propose the separation of the cloud scheme from the static partitioning of the atmospheric model. Using the framework FD4 (Four-Dimensional Distributed Dynamic Data structures), we show that this approach successfully eliminates the load imbalance and improves the scalability of the model system. We present a scalability analysis of the dynamic load balancing and coupling for two different supercomputers. The observed overhead is 6% on 1600 cores of an SGI Altix 4700 and less than 7% on a BlueGene/P system at 64Ki cores.

Keywords: atmospheric modeling, spectral bin cloud microphysics, scalability, dynamic load balancing, model coupling.

1 Introduction and Related Work

Cloud processes still represent one of the major uncertainties in current weather forecast, air quality, and climate models [1,3,24]. This, however, contrasts to their high importance to the atmosphere. It is obvious that future high-resolution atmospheric models require a more detailed description of cloud processes in order to achieve more realistic predictions of, e.g., extreme weather events. Most of today's atmospheric models describe cloud microphysical processes with a *bulk* approach. The so-called one-moment bulk schemes represent the hydrometeor classes (e.g. cloud water, graupel, and snow) by their bulk mass only and assume a prescribed size distribution of the particles. Two-moment [21] and

K. Jónasson (Ed.): PARA 2010, Part I, LNCS 7133, pp. 131–141, 2012.

multi-moment [16] schemes extend the description of each class by additional prognostic variables, such as the hydrometeor number density. This allows a better parameterization of the size distribution function. However, several studies emphasize the importance of a size-resolving approach [5,13]. Such *spectral* microphysics models explicitly characterize the size distribution of the hydrometeors by applying a bin discretization. Spectral microphysics schemes have been introduced in the PSU/NCAR Mesoscale Model (MM5) [13], the Weather Research and Forecasting Model (WRF) [12], and the COSMO model (Consortium for Small-scale Modeling) [6]. One of the challenges for the application of spectral bin microphysics schemes in atmospheric models is their enormous computational complexity. Thus, they have been applied for process studies only, but not for operational applications and it is very unlikely that such schemes will be used for numerical weather prediction or climate studies in the near future. Nevertheless, they are an interesting method for research applications, such as studies on the aerosol-cloud interaction [18], air quality modeling [7] or as benchmark for bulk schemes [22]. Because of their huge computational costs, a high scalability on high-performance computing systems is essential to use such models for comprehensive studies. However, this is complicated by severe load imbalances induced by the spectral microphysics: Cloudy areas of the model domain generate a substantially higher workload than cloudless areas. Such irregular workload variations require dynamic load balancing techniques [25], which readjust the partitioning periodically during the run time to maintain an equal distribution of the computational work. Note, that only a few of the widely-used atmospheric models support dynamic load balancing: parallel versions of MM5 [15] (discontinued) and the Regional Atmospheric Modeling System [26] (experimentally).

We propose a dynamic load balancing scheme for detailed cloud models. The basic idea is to decouple the partitioning of the cloud model from the atmospheric model's partitioning. Instead of creating data structures for the hydrometeors within the atmospheric model, these data are managed by a highly scalable framework, which dynamically balances the workload over the parallel processes. For this task we have developed the framework FD4 (Four-Dimensional Distributed Dynamic Data structures [9,11]). To our knowledge, such dynamic techniques have not yet been used for detailed cloud models. Due to the separation, both models need to be (re)coupled and thus form a system comparable to climate models in the way the coupled atmosphere and ocean model communicate regularly with each other.

Several software frameworks and tools have been developed to provide services for the parallel implementation of complex simulation codes, such as distributed data management and dynamic load balancing [4,25], adaptive mesh refinement [2,27], and model coupling [8,19]. FD4 integrates dynamic data management, load balancing, and coupling into a single framework to operate on the same data structures, which allows more performance optimizations compared to the utilization of separate software for these tasks. However, specialized frameworks offer more functionality than FD4, like grid interpolation for coupling,

the selection of various partitioning methods, or adaptive mesh refinement. FD4 has been developed for the parallelization and coupling of detailed cloud models. For example, to account for the requirements of size-resolved cloud microphysics models, the framework is optimized for large numbers of values per grid cell. However, FD4 can also be used for other multiphase or multiphysics applications. FD4 is written in Fortran 95 and uses MPI-2 [14] for parallelization. It is available as open source software at `http://www.tu-dresden.de/zih/clouds`

The rest of the paper is organized as follows: In the next section we introduce the atmospheric modeling system COSMO-SPECS and explain why the detailed microphysics scheme causes load balance issues. In Sect. 3 we describe the dynamic load balancing approach applied in the recently developed COSMO-SPECS+FD4 and briefly introduce the framework FD4. Finally, in Sect. 4, we show performance results of a benchmark scenario on two different supercomputers comparing both versions of the modeling system.

2 The Atmospheric Modeling System COSMO-SPECS

The model system COSMO-SPECS [6] has been developed to study the interaction between aerosols, clouds, and precipitation with a high level of detail. It consists of the COSMO model (`http://www.cosmo-model.org`), a non-hydrostatic limited-area atmospheric model, and the spectral microphysics model SPECS (SPECtral bin cloud microphysicS [23]). From the implementation point of view, the cloud parameterization scheme of COSMO has been replaced by SPECS, which introduces 11 new variables to describe three types of hydrometeors (water droplets, frozen particles, and insoluble particles). These 11 variables are discretized into a predefined number of size classes (e.g. 66 for the case presented in Sect. 4), leading to a high amount of data that have to be allocated for each cell of the rectangular grid.

Since the cloud microphysical processes operate on much smaller time scales than the dynamical processes in COSMO, two different step sizes are applied for the time integration. The COSMO step size is about 10–100 s, whereas the step size for the microphysics is at most 1 s. This splitting amplifies the computing time proportion of SPECS and, consequently, the model system's run time is dominated by the microphysics computations. Additionally, the computing time of SPECS per grid cell varies strongly depending on the range of the present size distribution for the three hydrometeor types. Especially the existence of frozen particles, which triggers additional computations, leads up to a 10 times increase of the computational costs compared to clear sky. The relation between the concentration of cloud particles and the computing time is shown in Fig. 1. COSMO is MPI-parallelized using a static domain decomposition of the horizontal grid into regular rectangular partitions. Due to the mentioned variability of the computational costs of SPECS, severe load imbalances occur, which lead to a significant waste of resources and insufficient scalability.

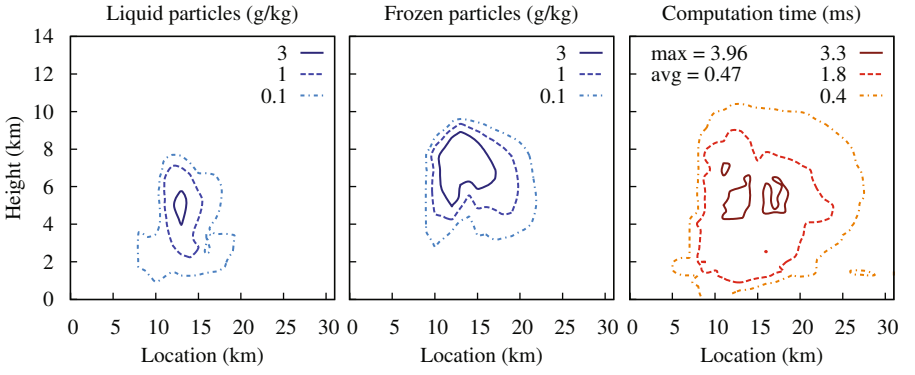
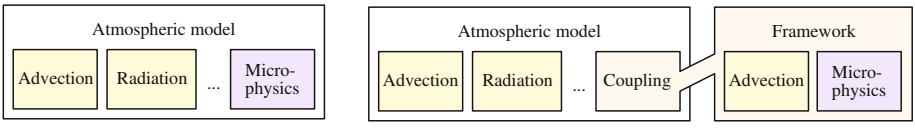


Fig. 1. Comparison of cloud particle mixing ratio and computing time of the spectral bin microphysics model SPECS for a vertical cross section through a simulated cumulus cloud. The plot on the right shows the computing time of one small time step of SPECS running on an SGI Altix 4700 (1.6 GHz Itanium 2 processor).



(a) Submodule (b) Separated

Fig. 2. Coupling concepts for cloud microphysics in atmospheric models: (a) Submodule based on data structures of the atmospheric model, (b) Separated data structures and decomposition using a data management framework

3 Load Balancing and Coupling Using FD4

In the original COSMO-SPECS implementation, the microphysics is incorporated as a submodule in the COSMO model, see Fig. 2(a). To enable the application of dynamic load balancing for the cloud model, we separated the hydrometeor data and related computations (microphysics and advection) from the COSMO model, see Fig. 2(b). These data are managed by the framework FD4 [9,11], which has been developed for the parallelization of multiphase cloud models. The program flow of one time step in COSMO-SPECS+FD4 is shown in Fig. 3. FD4 balances the microphysics computations and transfers coupling data between the different partitionings. The extensive hydrometeor data exist in the FD4 data structures only and are not exchanged with COSMO.

FD4 Data Structure. FD4 decomposes the regular grid in the three spatial dimensions into rectangular blocks, which consist of multiple grid cells. These blocks represent the smallest unit for load balancing. Consequently, their total number should be large enough to enable a fine-grained load balancing. FD4 allocates the data fields in the blocks according to a variable table that is specified

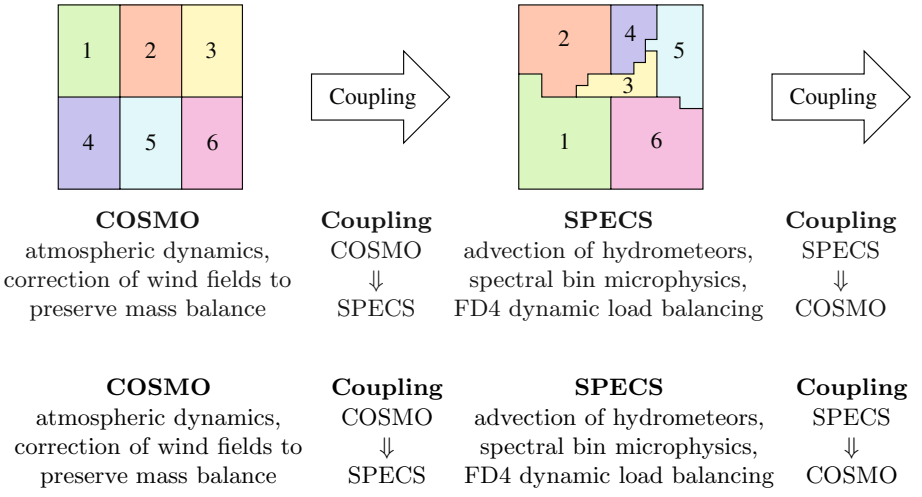


Fig. 3. Program flow of one time step in COSMO-SPECS+FD4 and exemplary illustration of the spatially different partitionings for COSMO and SPECS. The 6 parallel processes perform computations for COSMO and SPECS alternately. The partitions of each process for COSMO and SPECS are indicated by numbers 1–6. The SPECS computations are performed using a predefined number of smaller time steps per COSMO step.

by the user. An iterator is provided to traverse through the list of local blocks and access the data.

Dynamic Load Balancing. The blocks are distributed across the processes using space-filling curve (SFC) partitioning [25]. In general, SFCs provide a fast mapping from n-dimensional to one-dimensional space that preserves spatial locality. FD4 uses a Hilbert SFC [20] to reduce the three-dimensional partitioning problem to the contiguous partitioning of a one-dimensional array of block weights. For optimal load balance, the maximum load (*bottleneck value*) among all partitions has to be minimized. Several heuristics and exact algorithms exist for this problem [17]. FD4 uses a trivial parallel algorithm: Each process checks for a different bottleneck value whether a partitioning exists for it. Then, the minimum of the valid bottleneck values is identified and each process determines its own partitioning based on this value.

Performing dynamic load balancing involves costs for the calculation of a balanced partitioning and the redistribution of blocks. It is only beneficial (i.e. application run time is reduced), when the time-saving of a better balanced workload compensates these costs. This is addressed explicitly by FD4: The load balancing routine estimates the time required for load balancing and the time lost due to imbalance based on the elapsed steps and decides automatically whether load balancing is beneficial or not.

Coupling. FD4 facilitates to couple models based on FD4 to external models that have a different partitioning. It computes the overlaps of the external model's partitions with the FD4 block structure and transmits the data directly between the processes. Data can be exchanged in both directions between FD4 and the external models.

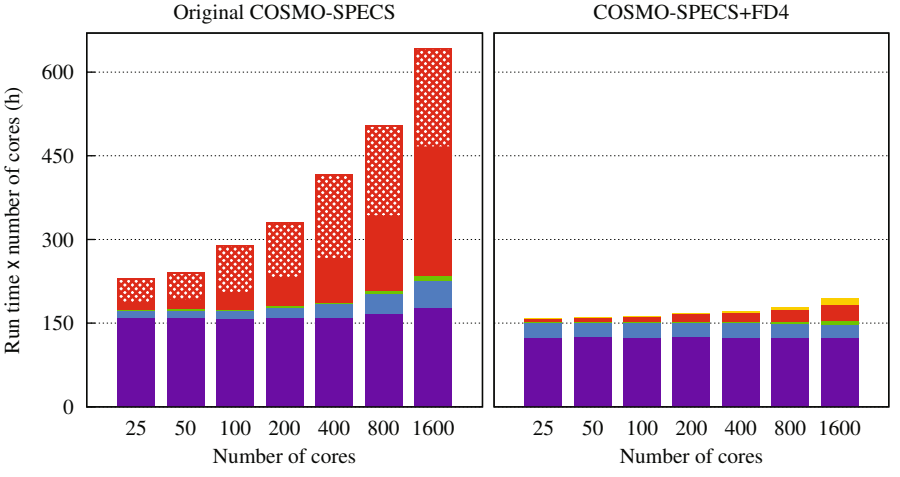
FD4 is based on the *sequential* scheduling [8] of the coupled models. Consequently, all processes perform computations for both COSMO and SPECS alternately. We expect this approach to perform better compared to the *concurrent* scheduling strategy, where the available cores are divided into fixed disjoint groups per coupled model. Since the total workload of SPECS varies strongly depending on the quantity and the type of clouds in the model domain, the latter approach would lead to load imbalances *between* the models [10].

4 Performance Results

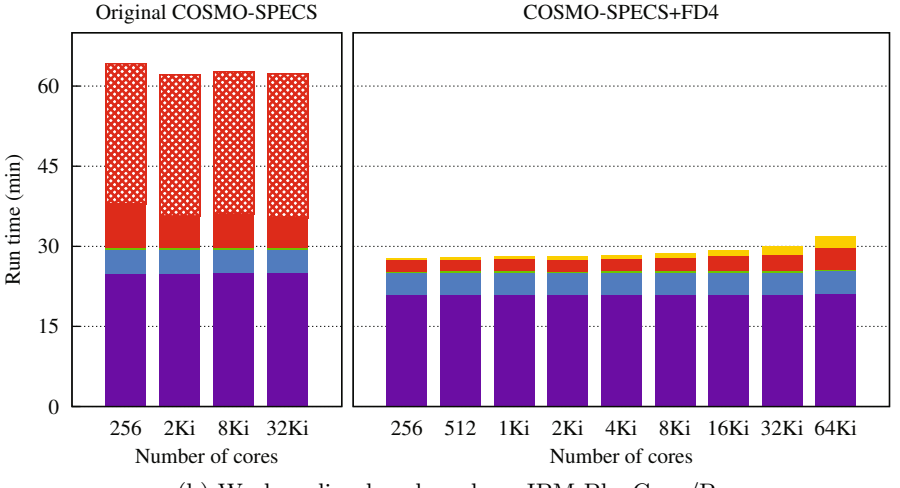
We compared the computational performance and scalability of the original COSMO-SPECS and its load balanced version COSMO-SPECS+FD4 using an artificial test scenario of a heat bubble over flat terrain [6]: A temperature perturbation, which is placed in the center of the horizontal grid, results in the growth of a precipitating mixed-phase cumulus cloud during the simulation period of 30 min. Additionally, we introduced a wind shear to the initial conditions. Figure 1 shows mixing ratios of liquid and frozen cloud particles after 30 min of simulation time. The resolution of the periodic horizontal grid was 1 km. The domain height of 18 km was discretized using 48 nonuniform height levels. The time step sizes were 10 s for COSMO and 0.5 s for SPECS, which results in 20 small microphysical steps per dynamical step. The original and the load balanced version yield identical simulation results except for small numerical deviations. All performance measurements are presented without model initialization time and output of simulation results.

4.1 Strong Scaling Benchmark on SGI Altix 4700

For this benchmark a fixed computational grid size of 80×80 cells with 48 height levels was used. The block size for the FD4 decomposition was 2×2×4, which results in a total number of 19 200 blocks. Figure 4(a) shows the performance results for 25 to 1600 cores on an SGI Altix 4700. Note, that the *overall* run time (wall clock time × number of cores) is shown, i.e. the total consumed CPU time. For a strong scaling benchmark, ideal scaling is achieved when the total consumed CPU time is constant with increasing number of cores. It is clear to see that the load balanced implementation scales much better. At 1600 cores, the original program took 24:10 min whereas the FD4 implementation required 7:22 min only, which is more than three times faster. The component breakdown of Fig. 4(a) reveals that the spectral microphysics consumes much more computation time than the COSMO model. However, with rising number of cores, the run time of the original COSMO-SPECS is increasingly dominated



(a) Strong scaling benchmark on SGI Altix 4700



(b) Weak scaling benchmark on IBM BlueGene/P

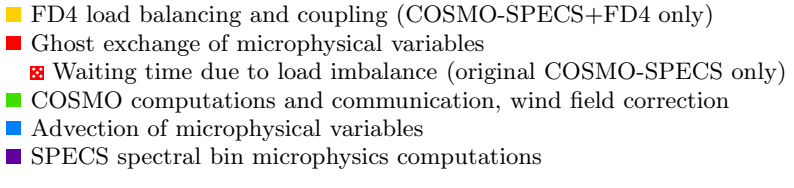


Fig. 4. Performance results and breakdown into components for the strong scaling benchmark (top) and the weak scaling benchmark (bottom). Flat horizontal lines indicate perfect scaling.

by MPI communication and waiting times due to load imbalance. The reason for the increasing MPI communication costs was found to be an inefficient message exchange scheme for the ghost cells of the microphysical variables using many small messages instead of few big ones. At 1600 cores less then 40% of the overall time is used for computations. The optimized COSMO-SPECS+FD4 has a much smaller communication overhead which is slightly increasing due to a decreasing average load balance and increasing costs for the actual message transfer. The run time percentage of FD4's data management is relatively low. However, it increases from 0.1% for dynamic load balancing and 0.1% for coupling at 25 cores to 2.7% and 3.3%, respectively, at 1600 cores.

4.2 Weak Scaling Benchmark on IBM BlueGene/P

The complexity of the dynamic load balancing and coupling algorithms applied in FD4 depends on the total number of blocks and the number of MPI processes. This poses the question, if COSMO-SPECS+FD4 can run on 10^4 cores efficiently. Therefore, we performed weak scaling benchmarks on an IBM BlueGene/P system. To scale up the problem size (and workload) exactly in the same proportion as the number of cores, we use a *replication scaling* approach [27]. At model initialization, the horizontal grid is *virtually* subdivided into tiles of 32×32 cells. Each tile is initialized with identical conditions for the heat bubble test scenario. The horizontal grid resolution and the number of height levels are kept constant at 1 km and 48 levels, respectively. We scaled our benchmark from a 32×32 grid containing one cloud at 256 cores up to a 512×512 grid containing 256 clouds at 64Ki cores. With an FD4 block size of 2×2×4 cells, the average number of blocks per process is constant at 12. Thus, FD4 had to balance 786 432 blocks dynamically on 64Ki cores in the largest run. Note, that neither COSMO-SPECS nor FD4 take advantage of the replication. Figure 4(b) shows the measured run times for the original COSMO-SPECS and the tuned COSMO-SPECS+FD4 divided into components. Since the workload per core is kept constant, perfect scaling is achieved when the program's run time does not increase with rising number of cores. Both versions scale almost perfectly, but the load balanced version is approximately twice as fast as the original one. The plot for COSMO-SPECS+FD4 indicates that the slight increase of run time is due to the load balancing and coupling of FD4 as well as growing costs for the ghost exchange. The FD4 workload is mainly growing because of the above mentioned complexity of the algorithms. At 64Ki cores, the percentage of FD4 is less then 7%, which shows that COSMO-SPECS+FD4 can efficiently utilize more than 10^4 cores.

4.3 Analysis of Load Balance

In Fig. 5 the measured load balance of both model versions is plotted against the time steps of the benchmark simulation on 8192 cores. Load balance is defined here as the average computing time among all processes divided by the maximum computing time among all processes. The ideal case is a load balance of one and

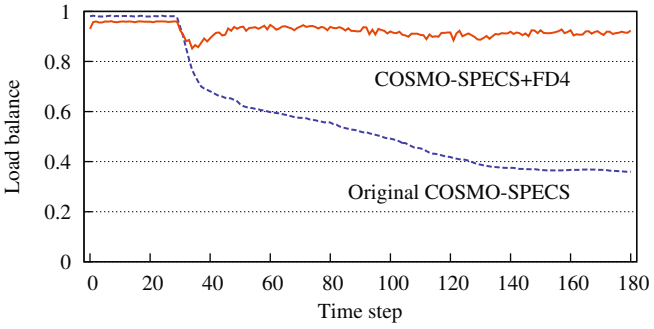


Fig. 5. Comparison of the load balance per COSMO time step between the original COSMO-SPECS and COSMO-SPECS+FD4 with dynamic load balancing. The measurement was performed for the weak-scaling benchmark case at IBM BlueGene/P on 8192 cores.

the worst case is the reciprocal of the number of processes. After 30 time steps, the load balance in the original COSMO-SPECS starts to drop notably, which indicates the beginning of the cloud growth. At the end of the simulation run, the balance is below 0.4. The load balance in COSMO-SPECS+FD4 drops down to 0.85 after 30 steps but stabilizes after 45 steps in the interval between 0.89 and 0.96 for the rest of the run. In the phase during steps 30–45, the load balancing approach to take the measured workload of the blocks as estimation for the next time step is not able to sufficiently keep pace with the high dynamics of workload variation. On average about 64% of the blocks have been migrated between the processes per COSMO time step during the COSMO-SPECS+FD4 run, which is very much. However, due to the costly microphysics computations, the relative communication overhead is very low. Furthermore, the communication pattern for the block migration is highly local: About 63% of the blocks were exchanged between direct neighbor MPI ranks in this run. Local communication patterns typically provide higher bandwidths than arbitrary patterns. The reason for this high locality is the SFC partitioning algorithm, which only shifts the process borders in the one-dimensional array of blocks.

5 Conclusion and Outlook

In this paper, we introduce a new way of coupling detailed cloud microphysics computations to atmospheric models, which allows dynamic load balancing. By using the framework FD4 to couple the mesoscale atmospheric model COSMO and the spectral bin microphysics model SPECS, a significant performance increase is achieved. Performance measurements on up to 64Ki cores show that the approach induces only little overhead for dynamic load balancing and coupling. While we expect the approach to be beneficial for other possibly less expensive spectral schemes, this most likely does not apply to two-moment or

multi-moment schemes due to their much smaller number of variables and considerably lower computational costs.

The high scalability of the new system is an important requirement for the feasibility of practical applications with spectral microphysics in atmospheric models. Additional improvements could render this possible in the very near future. As a next step we are aiming to reduce the computational costs of the microphysics by dynamically deciding for each grid cell whether the fast bulk parameterization scheme is sufficient (clear sky) or the spectral model is required. Another important aspect is the proper selection of the time integration step for the microphysics. The time scales of cloud processes are very heterogeneous in time and space, and thus, multirate time integration schemes provide a further approach of saving computational costs.

Acknowledgments. We thank the Deutscher Wetterdienst (German Weather Service) for providing the COSMO model. Furthermore, we thank Jülich Supercomputing Centre, Germany, for access to the JUGENE BlueGene/P supercomputer and Leibniz Supercomputing Centre, Germany, for access to their SGI Altix 4700 system. This work was funded by the German Research Foundation (DFG), grant No. NA 711/2-1.

References

1. Builtjes, P., Fowler, D., Feichter, J., Lewis, A., Monks, P., Borrell, P. (eds.): The Impact of Climate Change on Air Quality. The 4th ACCENT Barnsdale Expert Workshop (2008)
2. Burstedde, C., Burtscher, M., Ghattas, O., Stadler, G., Tu, T., Wilcox, L.C.: ALPS: A framework for parallel adaptive PDE solution. J. Phys. Conf. Ser. 180(1), 012009 (2009)
3. Chin, M., Kahn, R.A., Schwartz, S.E. (eds.): Atmospheric Aerosol Properties and Climate Impacts. U.S. Climate Change Science Program and the Subcommittee on Global Change Research (2009)
4. Devine, K., Boman, E., Heaphy, R., Hendrickson, B., Vaughan, C.: Zoltan Data Management Services for Parallel Dynamic Applications. Comput. Sci. Eng. 4(2), 90–97 (2002)
5. Fahey, K.M., Pandis, S.N.: Size-resolved aqueous-phase atmospheric chemistry in a three-dimensional chemical transport model. J. Geophys. Res. 108(D22), 4690 (2003)
6. Grützun, V., Knoth, O., Simmel, M.: Simulation of the influence of aerosol particle characteristics on clouds and precipitation with LM–SPECS: Model description and first results. Atmos. Res. 90, 233–242 (2008)
7. Jacobson, M.Z., Ginnebaugh, D.L.: Global-through-urban nested three-dimensional simulation of air pollution with a 13,600-reaction photochemical mechanism. J. Geophys. Res. 115, D14304 (2010)
8. Larson, J., Jacob, R., Ong, E.: The Model Coupling Toolkit: A New Fortran90 Toolkit for Building Multiphysics Parallel Coupled Models. Int. J. High Perf. Comput. Appl. 19, 277–292 (2005)
9. Lieber, M., Grützun, V., Wolke, R., Müller, M.S., Nagel, W.E.: FD4: A Framework for Highly Scalable Load Balancing and Coupling of Multiphase Models. In: AIP Conf. Proc., vol. 1281(1), pp. 1639–1642 (2010)

10. Lieber, M., Wolke, R.: Optimizing the coupling in parallel air quality model systems. Environ. Modell. Softw. 23(2), 235–243 (2008)
11. Lieber, M., Wolke, R., Grützun, V., Müller, M.S., Nagel, W.E.: A framework for detailed multiphase cloud modeling on HPC systems. In: Parallel Computing, vol. 19, pp. 281–288. IOS Press (2010)
12. Lynn, B., Khain, A., Rosenfeld, D., Woodley, W.L.: Effects of aerosols on precipitation from orographic clouds. J. Geophys. Res. 112, D10225 (2007)
13. Lynn, B.H., Khain, A.P., Dudhia, J., Rosenfeld, D., Pokrovsky, A., Seifert, A.: Spectral (Bin) Microphysics Coupled with a Mesoscale Model (MM5). Part I: Model Description and First Results. Mon. Weather Rev. 133, 44–58 (2005)
14. Message Passing Interface Forum: MPI-2: Extensions to the Message-Passing Interface (1997), `http://www.mpi-forum.org`
15. Michalakes, J.: MM90: A scalable parallel implementation of the Penn State/NCAR Mesoscale Model (MM5). Parallel Computing 23(14), 2173–2186 (1997)
16. Milbrandt, J.A., Yau, M.K.: A Multimoment Bulk Microphysics Parameterization. Part I: Analysis of the Role of the Spectral Shape Parameter. J. Atmos. Sci. 62(9), 3051–3064 (2005)
17. Pinar, A., Aykanat, C.: Fast optimal load balancing algorithms for 1D partitioning. J. Parallel Distrib. Comput. 64(8), 974–996 (2004)
18. Planche, C., Wobrock, W., Flossmann, A.I., Tridon, F., Van Baelen, J., Pointin, Y., Hagen, M.: The influence of aerosol particle number and hygroscopicity on the evolution of convective cloud systems and their precipitation: A numerical study based on the COPS observations on 12 August 2007. Atmos. Res. 98(1), 40–56 (2010)
19. Redler, R., Valcke, S., Ritzdorf, H.: OASIS4 - a coupling software for next generation earth system modelling. Geosci. Model Dev. 3(1), 87–104 (2010)
20. Sagan, H.: Space-filling curves. Springer, Heidelberg (1994)
21. Seifert, A., Beheng, K.D.: A two-moment cloud microphysics parameterization for mixed-phase clouds. Part 1: Model description. Meteorol. Atmos. Phys. 92, 45–66 (2006)
22. Seifert, A., Khain, A., Pokrovsky, A., Beheng, K.D.: A comparison of spectral bin and two-moment bulk mixed-phase cloud microphysics. Atmos. Res. 80, 46–66 (2006)
23. Simmel, M., Wurzler, S.: Condensation and activation in sectional cloud microphysical models. Atmos. Res. 80, 218–236 (2006)
24. Solomon, S., et al. (eds.): Climate Change 2007: The Physical Science Basis. Contribution of Working Group I to the Forth Assessment Report of the Intergovernmental Panel on Climate Change (IPCC). Cambridge University Press (2007)
25. Teresco, J.D., Devine, K.D., Flaherty, J.E.: Partitioning and Dynamic Load Balancing for the Numerical Solution of Partial Differential Equations. In: Numerical Solution of Partial Differential Equations on Parallel Computers, pp. 55–88. Springer, Heidelberg (2006)
26. Tremback, C.J., Walko, R.L.: The Regional Atmospheric Modeling System (RAMS): Development for parallel processing computer architectures. In: 3rd RAMS Users Workshop (1997)
27. Van Straalen, B., Shalf, J., Ligocki, T., Keen, N., Yang, W.S.: Scalability challenges for massively parallel AMR applications. In: IPDPS 2009, pp. 1–12. IEEE Computer Society (2009)