PARALLEL COUPLING OF REGIONAL ATMOSPHERE AND OCEAN MODELS

STEPHAN FRICKENHAUS
Alfred-Wegener-Institute for Polar and Marine Research,
Columbusstrasse, 27568 Bremerhaven, Germany
E-mail: sfrickenhaus@awi-bremerhaven.de

RENÉ REDLER AND PETER POST
Institute for Algorithms and Scientific Computing,
German National Research Center for Information Technology
Schloss Birlinghoven, D-53754 Sankt Augustin, Germany

In coupled models the performance of massively parallel model components strongly suffers from sequential coupling overhead. A coupling interface for parallel interpolation and parallel communication is urgently required to work out this performance dilemma. Performance measurements for a parallel coupling of parallel regional atmosphere and ocean models are presented for the CRAY-T3E-1200 using the coupling library MpCCI. The different rotated grids of the models MOM2 (ocean-seaice) and PARHAM (atmosphere) are configured for the arctic region. In particular, as underlying MPI-implementations CRAY-MPI and MetaMPI are compared in their performance for some relevant massive parallel configurations. It is demonstrated that an overhead of 10% for coupling, including interpolation and communication, can be achieved. Perspectives for a common coupling specification are given enabling the modeling community to easily exchange model components as well as coupling software, making model components reusable in other coupling projects and on next generation computing architectures. Future applications of parallel coupling software in parallel nesting and data assimilation are discussed.

1 Introduction

The climate modeling community produces a growing number of model components, e.g., for simulations of atmosphere, ocean and seaice. Currently, more and more model codes are parallelized for running on massively parallel computing hardware, driving numerical performance to an extreme, mostly with the help of domain decomposition and message passing techniques. From the undoubted need for investigation in coupled high performance models a new performance bottleneck appears from the necessary interpolation and communication of domain decomposed data between the model components[1]. In particular, scalability of coupled massively parallel models is strongly bound when using a sequential coupling scheme, i.e., gathering distributed data from processors computing the sending model component, interpolating, communicating and scattering data to processors computing the receiving

model component. The alternative to such an *external coupling* approach is the *internal coupling approach*: mixing the codes of model components to operate on the same spatial domains, for convenience with the same spatial resolution. Thus, internal coupling puts strong limits on the flexibility of the model components. In external coupling, the performance of the coupled model can be optimized by running model components in parallel, each on an optimal, load balancing number of processors. Furthermore, external coupling allows for an easy replacement of model components, at least, when a certain standard for coding the coupling is followed.

To overcome the bottleneck of sequential coupling, a set of parallel coupling routines is required, capable of parallel interpolation of data between partly overlapping domains and of managing all required communication in parallel, e.g., by a message passing technique.

As an implementation of such a functionality, the *mesh based parallel code coupling interface* MpCCI[3,4] is used in the following. MpCCI can be considered as a medium level application programming interface, hiding the details of message passing and interpolation in a library, while offering a small set of subroutines and an extensive flexibility by the use of input configuration files. It is advantageous for the integration into a certain class of model codes, to encapsulate calls to the library in a high level interface, the *model interface*, allowing, for example, for an easy declaration of regular domain decomposed grids and for a simple call to a coupling routine. The details of the interface developed for the presented models are not subject to this paper. Instead, performance measurements for the specified arctic regional model in different massively parallel configurations and an outline for further applications as well as for standardization of model interfaces are presented.

Concerning the programming effort of coupling it may be unpractical to mix model components, due to the fact that memory per processor is limited, used file unit numbers or naming of variables and/or common blocks may coincide.

Making model components compatible to work in a single executable (SPMD) by using a common I/O-library and a common memory allocation scheme may be achievable for model codes of low complexity. However, such a procedure must be repeated for every new model component and also for model code updates; furthermore the reusability of coupled model components is better without references to special I/O-managing libraries and naming conventions.

The approach to leave model component codes in separate binaries (MPMD, i.e., multiple Program Multiple Data) seems much more practical. However, on certain computing architectures this requires a metacomputing

library for message passing. For example, on a CRAY-T3E, using CRAY-MPI, two different executables cannot be launched in one MPI-context; it is also not possible with CRAY-MPI or CRAY-shmem to establish message passing communication between separately launched groups of MPI-processes, i.e., between application teams. This is worked around with metacomputing MPI-implementations, such as metaMPI or PACX[2]. Furthermore, a metacomputing MPI allows for coupling model components across different computing architectures, even in different locations, provided that a high bandwidth – low latency network connection is installed.

In the following presentation of performance measurements the potentials of metaMPI and CRAY-MPI are investigated in detail.

2 MpCCI and the Model Interface for Domain Decomposed Data

MpCCI is designed as a library. It enables to loosely couple different (massively) parallel or sequential simulation codes. This software layer realizes the exchange of data which includes neighborhood search and interpolation between the different arbitrary grids of any two codes that take part in the coupled problem. In parallel applications the coupling interfaces of each code can be distributed among several processors. In this case communication between pairs of processes is only realized where data exchange is necessary due to the neighborhood relations of the individual grid points.

In the codes themselves the communication to MpCCI is invoked by simple calls to the MpCCI library that syntactically follow the MPI nomenclature as closely as possible.

On a lower level, and hidden from the user, message passing between each pair of codes in a coupled problem is performed by the subroutine calls that follow precisely the MPI standard. Therefore the underlying communication library can be a native MPI implementation (usually an optimized communication libray tuned to the hardware by the vendor), MPICH or any other library that obeys the MPI standard like, e.g., metaMPI.

It must be noted, that for coupling of domain decomposed data by interpolation on the nodes, elements must be defined, spanning the processor boundaries of data domains. Otherwise, gridpoints of the receiving model lying between gridpoints of the domain boundaries of the sending model do not receive data. This also requires the introduction of ghostpoint data that must be updated before sending data. Such a functionality is easily implemented in the model interface to MpCCI.

Furthermore, due to the rather simple, but very precise conservative in-

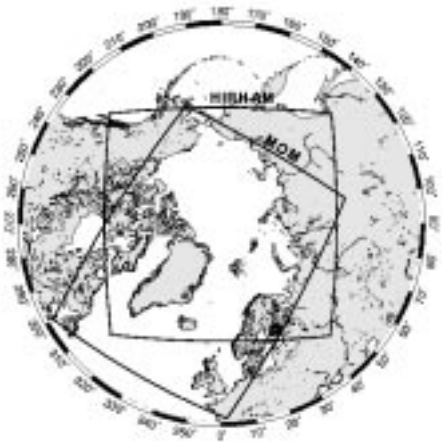

Figure 1. The different rotated grids of the arctic atmosphere model HIRHAM and the ocean-seaice model MOM.

terpolation of fluxes in MpCCI, the received fluxes show up artifical patterns with strong deviations from a smooth structure. These deviations must be smoothed out locally, either by calculation of local mean values, or by a more sophisticated local smoother, that may be based on an anisotropic diffusion operator. Such a smoother with local diffusion coefficients calculated from the interpolation error of a constant flux is currently under development. Alternatively, one might use the non-conservative interpolation also for the fluxes and rescale the received data such that global conservativity is restored.

3 Measuring parallel coupling performance

The $\frac{1}{4}^\circ$ arctic ocean-seaice model MOM2 has 243x171 horizontal gridpoints on 30 levels.

The $\frac{1}{2}^\circ$ atmosphere model HIRHAM works on 110x100 horizontal gridpoints on 19 levels. In figure 1 the rotated grids of the models are sketched over the arctic.

The atmosphere model communicates 6 scalar fluxes and 2 scalar fields to the ocean-seaice model, making a total of 0.08 MW (1 Megaword = 8 Megabyte) coupling data on the sender site, and 0.33 MW for the receiver after interpolation. In the reverse direction 4 scalar fields are sent, summing up to 0.2 MW coupling data in total for the sender and 0.06 MW for the

Table 1. Performance measurements of MpCCI for the configuration of the coupled model of the arctic; bandwidth data is given in Megawords per second [MW/s] (one word = one 8 byte double precision number), see text. OS: ocean send; OR: ocean receive; AR: atm. receive; AS: atm. send.

PEs		stdMPI [MGPD/s]				metaMPI-local [MGPD/s]			
ocn	atm	OS	OR	AR	AS	OS	OR	AR	AS
20	80	1.28	1.15	0.20	1.97	0.092	0.057	0.026	0.013
30	110	1.06	0.92	0.16	1.62	0.044	0.025	0.006	0.012
1	100	0.29	0.37	0.066	5.53	0.387	0.342	0.135	0.091

PEs		stdMPI/metaMPI			
ocn	atm	OS	OR	AR	AS
20	80	14	20	8	151
30	110	24	37	27	135
1	100	0.7	1.0	0.5	61

receiver. Here gridpoints from non-overlapping domains were included in the counting.

The lower block in table 1 shows the ratio of CRAY-MPI bandwidths over metaMPI bandwidths. Since the timed routines contain - besides the communication routines - MpCCI-implicit interpolation routines, the increase of bandwidth is not linearly dependent on the achievable increase in point-to-point bandwidth between processors of the two models when switching from metaMPI to CRAY-MPI. It is noteworthy at this point, that metaMPI has almost the same communication performance between the processors within the model components compared to CRAY-MPI, i.e., the performance of uncoupled models is unchanged.

It is seen that in the case of coupling a single MOM-process (holding the full size arrays of boundary data) with 100 HIRHAM processes, the use of metaMPI has noteworthy influence only on the HIRHAM-to-MOM send bandwidth (61 times the CRAY-MPI bandwidth). In the setups with parallel MOM-coupling (upper two rows) the reduction of the bandwidth for HIRHAM-to-MOM send is also dominant.

In figure 2 the timing results for a set of communication samples are displayed for 20 MOM processors coupled to 80 HIRHAM processors. The upper graph displays results from CRAY-MPI, the lower graph for metaMPI. The displayed samples are a sequence of 20 repeated patterns. The points in the patterns represent the timings of the individual processors.

It is observed in the upper graph that the receiving of data in MOM

(MOM-RCV) takes the longest times (up to 0.225 seconds), while the corresponding send operation from parallel HIRHAM (PH-SND) is much faster (needs up to 0.05 seconds). In contrast, the communication times for the reverse direction are more balanced. In the lower graph, displaying the results for metaMPI-usage, communication times appear more balanced. The times of up to 4 seconds are a factor 18 above the corresponding CRAY-MPI measurements. In this massive parallel setup coupling communication times would almost dominate the elapsed time of model runs, since the pure computing time for a model time interval between two coupling communication calls (typically one hour model time) is on the same order of magnitude (data not shown).

In figure 3 the timing results for communication are displayed for 30 MOM processors and 110 HIRHAM processors. Qualitatively the same behavior is seen as in figure 2. For the usage of CRAY-MPI (upper graph) comparable timings are measured. However, for metaMPI, the maximum times are 8 seconds for certain HIRHAM receive operations, which is a factor 2 longer than in figure 2, bottom graph. Clearly the ratio of communication times over computation times is even worse compared to the setup used for figure 2.

Figure 4 depicts the timing results for coupling communication between one MOM processor and 100 HIRHAM processors. It is seen in the upper graph, that the MOM receive operations dominate the coupling communication times (about 0.85 seconds at maximum). This characteristic is also found for metaMPI-usage (lower graph). Interestingly, in this setup, also the coupling times are nearly unchanged. Furthermore, the four displayed operations are performed partially in parallel. The net time of 1.33 seconds used for one coupling communication call is also found for MetaMPI (data not shown). This corresponds well to the bandwidth ratios given in the lower block in table 1.

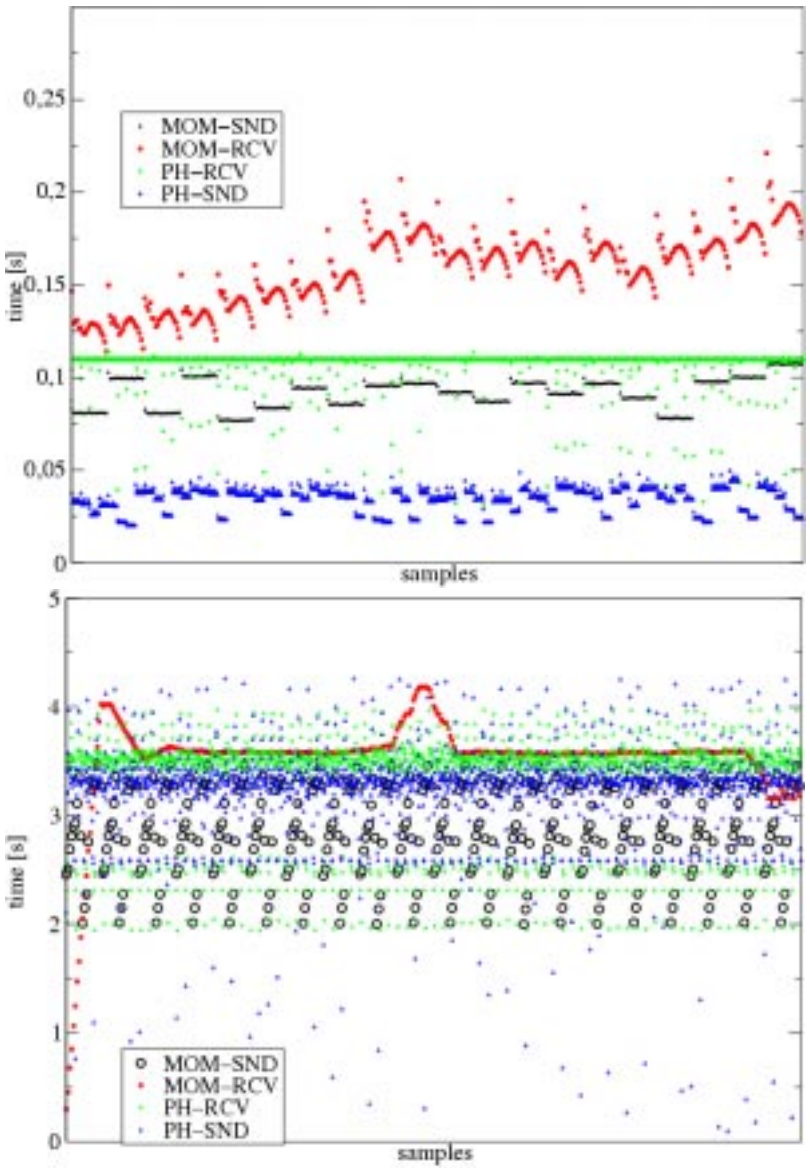

Figure 2. Timing data measured under CRAY-MPI (top graph) and metaMPI (bottom graph), coupling 20 MOM processes with 80 HIRHAM processes

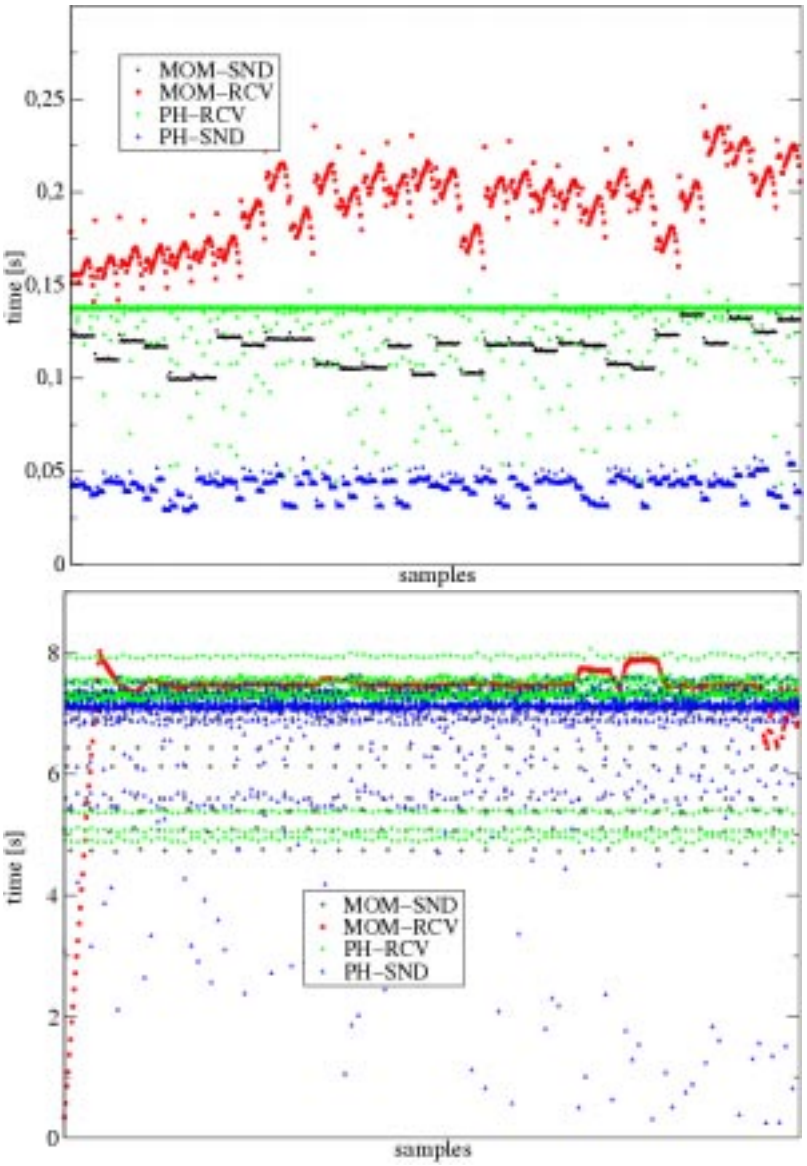

Figure 3. Timing data measured under CRAY-MPI (top graph) and metaMPI (bottom graph), coupling 30 MOM processes with 110 HIRHAM processes

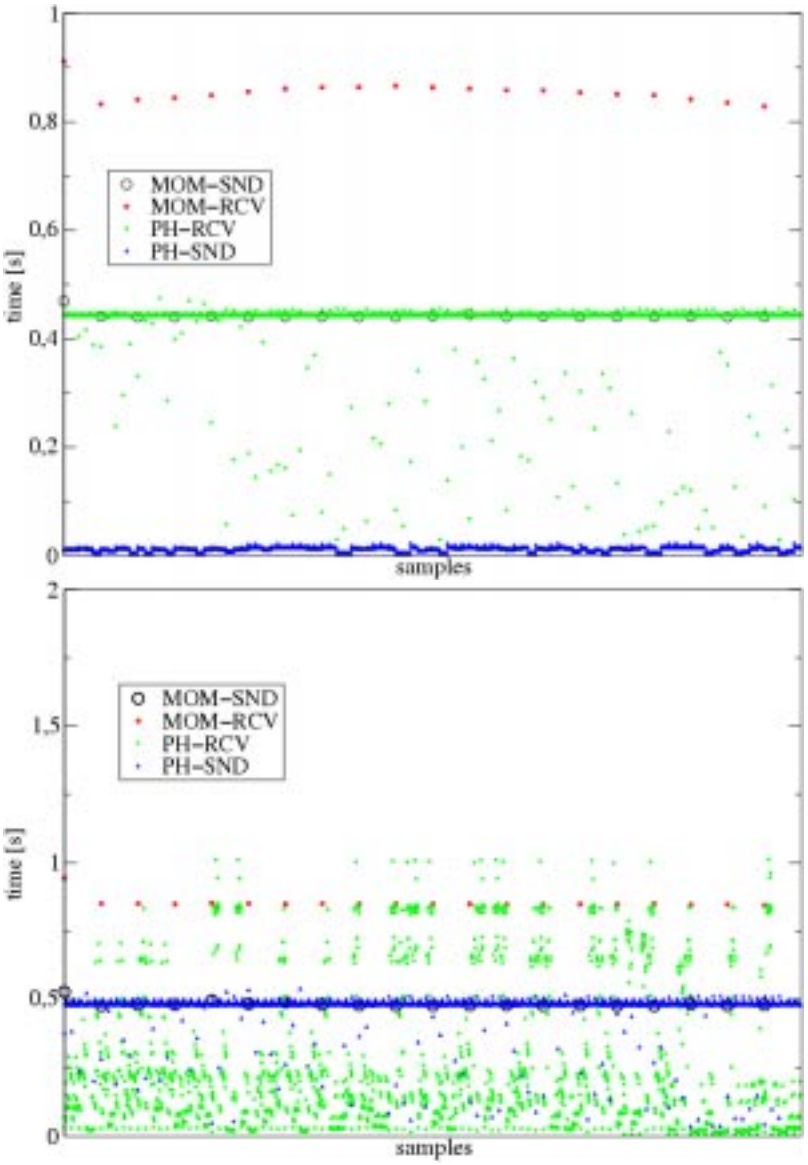

Figure 4. Timing data measured under CRAY-MPI (top) and metaMPI (bottom), coupling one MOM process, holding the full coupling boundary data arrays, with 110 HIRHAM processes

Using metaMPI has the obvious drawback that communication bandwidths between different processor groups, i.e., model components, are rather low, at least on the CRAY-T3E. This is due to the implementation of MetaMPI, using the low-bandwidth socket communication between application teams on the T3E.

The socket communication maximum bandwidth of about 20 MB/s and its sequential character makes parallel communication patterns between processor groups strongly inefficient for massively parallel coupling. Thus, the sequentialization inherent in coupling of 100 HIRHAM processors with only one MOM processor is expected to have a comparable influence on inter-model bandwidth for both, metaMPI as well as CRAY-MPI. This is proved by the results of measurements given in table 1 and depicted in figure 4. The overhead of coupling is, in this case, about 10 % percent of the total elapsed time used for the simulation of one hour model interval, with a coupling frequency of $1/hr$.

It is concluded that a high performance parallel coupling strongly requires parallel high-bandwidth communication between the model components. Within the MPMD model, on other computing architectures than CRAY's T3E, this parallel communication may be easily achieved without a metacomputing MPI. Within a SPMD approach of mixing the codes, which requires much more work for restructuring complex codes, a fivefold coupling performance increase can be achieved.

4 Outline of a Common Coupling Specification – CCS

Having in mind the reusability and exchangeability of model components, it is attractive to think about a certain standardization of coupling interfaces for classes of models. Here not only the coding standard must be taken into account by defining subroutines for coupling, but also a proper definition of the physical quantities that a model component must send and receive. As an example, a number of defined fields/fluxes must be specified to be exported by ocean models to the coupling interface that is common to all currently used models or can be at least easily implemented into them.

As the climate modeling community already operates with different types of couplers, e.g., the NCAR CSM flux coupler or the OASIS coupler, a common model interface to different coupling interfaces should be specified instead of selecting/developing a common coupling interface. This should allow for an easy exchange of the coupler as well as of the model components. However, it may become necessary to modify existing coupling interfaces to be compliant with the needs of the model interface.

In this sense a Common Coupling Specification is an intermediate specification of subroutines and data formats, independent from model as well as coupler implementation details.

For these considerations it is completely irrelevant, whether the coupling interface is a coupling library or a separate coupling process.

5 Further application areas for parallel coupling libraries

Although coupling interfaces have until now widely been used in two dimensional coupling across the sea-atmosphere interface, further applications are easily conceivable. Here two perspectives are drawn.

A common procedure in numerical weather prediction is to perform a sequence of simulations from the global scale down to regional scales. This implies an interpolation of boundary data as well as initial state data from large scale models to a hierarchy of nested models. The approach of storing this data on filesystems before being consumed by the more regional model simulations may in a parallel computation of the complete model hierarchy be replaced by the operation of a coupling interface. For reasons of reliability in an operational weather prediction environment nesting must be implemented on a flexible basis, allowing for both, I/O-based communication of data for sequential computation as well as message passing based communication for parallel computation.

One advantage of the parallel nesting approach may be, that model components can run below their scaling limit, i.e., on a moderate number of processors, as results for the nested models are used (consumed) almost at the same time as they are produced. Thus, the investment of computer resources used to minimize the elapsed time of a prediction from sequential nesting may be used for model enhancements, e.g., higher spatial resolution or more complex numerical approximations to physical processes.

In the framework of model nesting data assimilation is a central task, generating proper initial and boundary data for limited area models. Here one might think of coupling interfaces being used for the interpolation of simulated data onto a domain decomposed set of observations. This could allow for a load-balanced and scalable evaluation, for example in 4D-var, of the costfunction and its gradient. The mesh-based approach of MpCCI is very promising as it provides an efficient parallel interpolation on to a set of irregularly distributed points.

6 Conclusion

The performance dilemma of coupled parallel models can be overcome by the use of sophisticated parallel coupling interfaces. The profits of the external coupling approach in a MPMD model can fully be exploited only under an appropriate high performance implementation of the communication between processor groups. In general, a parallel communication facility is required for parallel coupling of massively parallel, domain decomposed models to be optimal.

The foreseeable needs for more flexible coupled modeling environments, based on a whole variety of model components used in a model hierarchy, should be met by a community wide initiative defining an interface standard that specifies the fields and fluxes to be exchanged between certain model classes. As well, a rather limited set of subroutines is to be defined as the model interface to different coupling interfaces. Such a Common Coupling Specification together with a parallel coupling interface may also serve as a basis for further fields of application, such as parallel nesting and data assimilation. Furthermore, the transition of modeling environments to next generation computing architectures is much better preconditioned by standardization. Here, the activities of the MPI forum[5] may serve as an example, having established successfully a specification for message passing and parallel I/O.

Acknowledgments

The performance measurements were part of the arctic model coupling project bvkp01 running at ZIB/Berlin using ZIB's T3E-1200. MetaMPI was kindly provided by Pallas GmbH, Brühl (Germany).

References

1. S. Valcke, International Workshop on Technical Aspects of Future Sea-Ice-Ocean-Atmosphere-Biosphere Coupling, CERFACS, Technical Report TR/CMG **00-79**, (2000).
2. Th. Eickermann, J. Henrichs, M. Resch, R. Stoy, and R. Völpel. Metacomputing in Gigabit Environments: Networks, Tools, and Applications, Parallel Computing **24**, 1847-1872 (1998).
3. R. Ahrem, M.G. Hackenberg, P. Post, R. Redler, and J. Roggenbuck. Specification of MpCCI Version 1.0, GMD-SCAI, Sankt Augustin (2000). see also http://www.mpcci.org

4. R. Ahrem, P. Post, and K. Wolf. A Communication Library to Couple Simulation Codes on Distributed Systems for Multi-Physics Computations. PARALLEL COMPUTING, Fundamentals and Applications, Proceedings of the International Conference ParCo99. E.H. D'Hollander, G.R. Joubert, F.J. Peters, and H. Sips (eds.), Imperial College Press, 2000.
5. see http://www-unix.mcs.anl.gov/mpi/index.html