COMPUTATIONAL METHODS IN SCIENCE AND TECHNOLOGY **15(1)**, 9-19 (2009)

# Porting the OPATM-BFM Application to a Grid e-Infrastructure – Optimization of Communication and I/O Patterns

**Alexey Cheptsov [1], Kiril Dichev [1], Rainer Keller [1]**
**Paolo Lazzari [2] and Stefano Salon [2]**

[1] *High Performance Computing Center*
*University of Stuttgart (HLRS)*
*Nobelstrasse 19, 70569 Stuttgart, Germany*
*e-mail: {cheptsov/dichev/keller}@hlrs.de*

[2] *Dept. of Oceanography*
*Istituto Nazionale di Oceanografia e di Geofisica Sperimentale (OGS)*
*B.go Grotta Gigante 42/c*
*34010 Sgonico (TS), Italy*
*e-mail: {plazzari/ssalon}@inogs.it*

(Received: 18 December 2008; published online: 25 March 2009)

**Abstract:** OPATM-BFM is an off-line three-dimensional coupled eco-hydrodynamic simulation model used for biogeochemical and ecosystem-level predictions. This paper presents the first results of research activities devoted to the adaptation of the parallel OPATM-BFM application for an efficient usage in modern Grid-based e-Infrastructures. For the application performance on standard Grid architectures providing generic clusters of workstations, such results are important. We propose a message-passing analysis technique for communication-intensive parallel applications based on a preliminary application run analysis. This technique was successfully used for the OPATM-BFM application and allowed us to identify several optimization proposals for the current realization of the communication pattern. As the suggested improvements are quite generic, they can be potentially useful for other parallel scientific applications.

**Key words:** communication pattern, Grid, MPI, eco-hydrodynamic simulation, performance optimization

## I. INTRODUCTION

Environmental risks of natural or anthropogenic origin may be prevented or managed by the use of operational short-term forecasts. The Global Monitoring for Environment and Security [1] is a partnership of the European Commission and the European Space Agency, and aims to develop short-term forecasts of marine ecosystems to be used in the frame of environmental monitoring. To tackle this challenging goal, the MERSEA European Integrated Project [2] was launched in order to develop a sustainable pan-European pre-operational forecasting system. The objective of this system was to provide products, on a regional basis, to a variety of intermediate users. The OPATM-BFM coupled model [3] that has been developed at OGS is one of these products, and today represents a state-of-the-art numerical tool able to produce biogeochemical and ecosystem-level short-term predictions for the Mediterranean Sea. The next section of this paper is devoted to describing the basics of the model and its pre-operational application to the Mediterranean basin.

The efficient usage of high-performance resources (like an IBM SP5 machine of CINECA [4] currently used for the main production cycle) is a major point in reaching high application productivity. However, the model is expected to deliver additional products and information for different time scales as well as for climatic scenario analyses (a multi-decadal period of integration), this constituting the most considerable limitation of the application scalability. The facilities offered by scientific e-Infrastructures (e.g. DORII [5]) should provide scientific applications deployed in their framework with a full-range access to the distributed computational and storage Grid services [6, 7].

The experience of porting other scientific applications from different fields of science to the Grid [8] has encouraged us to examine the suitability of the Grid resources for improving the OPATM-BFM application performance. For this purpose the e-Infrastructure is planned to provide the application with all necessary monitoring and development tools as well. The anticipated effect from porting to the Grid is increasing of the application performance and scalability. This will allow an efficient usage of the OPATM-BFM application for more complicated use cases.

This paper describes the first research results devoted to porting issues of the OPATM-BFM application to the Grid. The application tests are performed on a cluster of workstations which is representative for a Grid component today. We show that application performance can dramatically decrease because of shortcomings due to the current realization.

Deep understanding of the message-passing communication pattern was therefore the obligatory step towards porting and efficient usage of the application on Grid resources. In sections III and IV of the paper we describe in detail the current implementation of the communication pattern. The analysis, categorization and optimization of the communication pattern is not a trivial task. We present a technique that allows us to proceed with analysis of communication-intensive parallel applications. Being successfully used for the OPATM-BFM application, the technique should also be valuable for the investigation of other parallel scientific applications.

Finally, this paper describes optimization proposals for the current realization of the OPATM-BFM message-passing communication pattern and evaluation results. The defined proposals could be valuable for the application runs on dedicated resources as well as Grid resources. The results might also be relevant for other scientific applications that implement a similar inter-process communication (e.g. the all-to-one communication) or for newly developed parallel applications.

## II. OVERVIEW OF THE OPATM-BFM COUPLED MODEL

OPATM-BFM is an off-line[1] three-dimensional MPI-parallel coupled eco-hydrodynamic model, and constitutes the core of a forecasting system embedded in a fully automatic procedure that produces maps of biogeochemical concentrations for the whole Mediterranean basin on a weekly basis.

The model solves the transport-reaction Eq. (1) for the generic biogeochemical concentration $c_i$ based on the advection-diffusion processes:

$$\frac{\partial c_i}{\partial t} + v \cdot \nabla c_i = w_i \frac{\partial c_i}{\partial z} + \\ + k_h \nabla_h c_i + \frac{\partial}{\partial z}\left[ k_z \frac{\partial c_i}{\partial z} \right] + R_{\text{bio}}\ (c_i, c_1 \ldots c_N, T, I \ldots) \tag{1}$$

where $v$ is the current velocity, $w_i$ is the sinking velocity, $k_h$ and $k_z$ – the eddy diffusivity constants, and $R_{\text{bio}}$ is the biogeochemical reactor that depends, in general, on the other concentrations and on temperature $T$, short-wave radiation $I$ and other physical variables. The complexity of the OPATM-BFM model consists in the high number of prognostic variables ($c_i$) to be integrated. Bacteria, oxygen, inorganic nutrients, phytoplankton, zooplankton, organic detritus are among the 51 variables produced by the model. The objective of developing a biogeochemical model that can be operationally applied to a basin such as the Mediterranean Sea requires an interdisciplinary effort. This work was achieved with the cooperation of different laboratories involved in the forecasting system within the framework of the Italian Group of Operational Oceanography [9]. The off-line coupling between physics and biogeochemistry was established between the operational forecasting system for the Mediterranean Sea managed by the National Institute of Geophysics and Volcanology (INGV) and the biogeochemical model BFM [10] embedded in the OPATM transport module. Computational resources and expertise were supplied by the Italian supercomputing center CINECA, while the Institute for Atmospheric Sciences and Climate (ISAC-CNR) provided satellite chlorophyll data, which were used to compare the model results.

The physical forcing fields supplied by INGV (current velocity, temperature, salinity, vertical eddy diffusivity, wind speed, short-wave radiation) are up-scaled to a lower horizontal spatial resolution, from INGV 1/16° to OPATM-BFM 1/8°, with an interpolating interface based on the cell-merging technique, which preserves the fluxes and, consequently, the divergence of the original data. The vertical resolution (72 levels) is left unchanged in order to maintain a good reproduction of the vertical processes known to be of great relevance for biogeochemical processes (mixing of the water column). This approach takes advantage of the benefits of state-of-the-art dynamic prognosis granted by INGV that includes extensive data assimilation and keeps the off-line dynamics files and the computational burden affordable, given the currently available resources.

[1] In the off-line approach the time integration of the transport-reaction biogeochemical equations is not synchronized with the Navier-Stokes solver: the circulation field is an external forcing and it has to be known before the integration of the equations.

The products of the simulations are the concentrations of key variables (macronutrients, chlorophyll, phytoplankton and bacterial productivity) and are routinely delivered from the OGS website [11] in order to track temporal and vertical dynamics of the basin biogeochemical properties.

This system represents a conceptual evolution when compared with previous basin-wide on-line coupled models developed for the Mediterranean Sea. Daily mean forcing fields used in the off-line coupling filter out the numerical noise introduced by integrations, consistently with the assumptions made in many experiments performed to estimate biological parameters. At the same time, every improvement in the physical model is immediately gained by the biogeochemical compartment.

OPATM-BFM has been implemented mainly in open sea regions and therefore can be safely used, when properly validated, for large-scale assessments such as:

- estimation of the carrying capacity of the Mediterranean basin (valuable information for ecosystem-based approaches to fisheries management);
- provision of habitat suitability indicators for risk assessment of a non-Mediterranean species invasion, such as algae;
- regional ecosystem responses to climate change;
- scenario analyses;
- design of observational research cruises and activities.

The parallel realization of the model developed at OGS is based on the message-passing communication pattern implemented by means of MPI [12].

The OPATM-BFM model was originally designed to provide initial and boundary conditions for coastal biogeochemical models in the Mediterranean Ocean Observing Network [13]. However, in light of the promising results achieved so far during the pre-operational phase (10-20 days of integration) [14], we expect that additional products and information can also be delivered at different time scales, as well as for climatic scenario analyses (multi-decadal period of integration).

## III. EXPERIMENT SETUP AND ANALYSIS TECHNIQUE

Investigation of the application was performed in cooperation of OGS with the High Performance Computing Center of the University of Stuttgart in the framework of the DORII project [5]. The project aims to deploy an e-Infrastructure for scientific communities focusing among other engineering areas on the environmental science community.

Having a long experience in the parallel computing, HLRS is actively involved in parallel software development, in particular providing support to application developers with the Open MPI [15] library interface implementation. Open MPI is an open-source implementation of the MPI-2 standard [16] developed by a consortium of research and industrial organizations of which HLRS is a member.

The application was analyzed on the cluster "Cacau" [17] of HLRS for a standard use case providing several types of input data that differ in complexity and duration depending on the corresponding simulation. Fig. 1 shows the analysis scheme of the application investigation.

The analysis of internal message-passing communication patterns is based on the profile collected during the application execution. The collection has been performed by means of instrumentation tools and libraries which are part of the instrumentation framework proposed for the DORII project, as shown in Fig. 1 (e.g. Vampir tools [18]). Based on the trace files, a communication profile can be analyzed by means of the provided visualization tools.

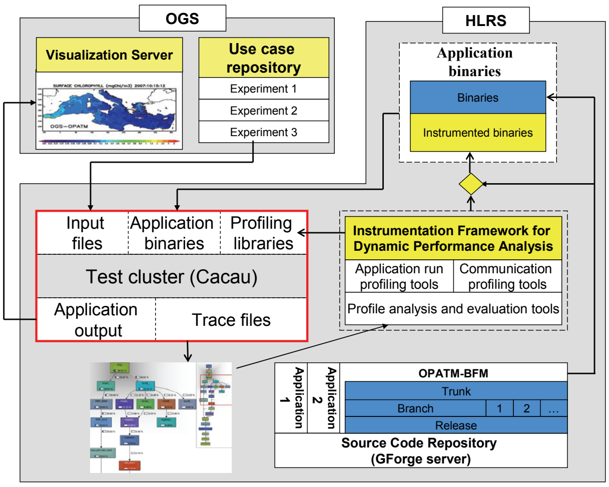


Fig. 1. The analysis scheme

Due to the long duration of the execution (several hours) for a standard OPATM-BFM use case (namely 816 steps of the main simulation – corresponds to 17 days of simulated ecosystem behavior), the size of recorded trace data increases accordingly. Hence the trace data being collected for the execution of the standard use case could not be processed by the available software for communication analysis and performance evaluation.

As a standard production run has many iterations and each iteration performs many communication routine calls, the trace file may get very large (up to tens of gigabytes). This is typical for an analysis of parallel scientific applications. However, the time integration of OPATM-BFM is performed in the main loop where the typical pattern of the MPI calls is iteration-independent. Hence, the iteratively repeated regions can be profiled for a limited number of iterations that are representative of the generic pattern communication of a longer run. The initialization and finalization of the simulation are profiled as usual. This can be done by means of event filtering in the defined regions of the execution or launching the application for special use cases with a limited number of iterations. The second approach is preferable because it allows to reduce the time needed for launching the application in the test mode.

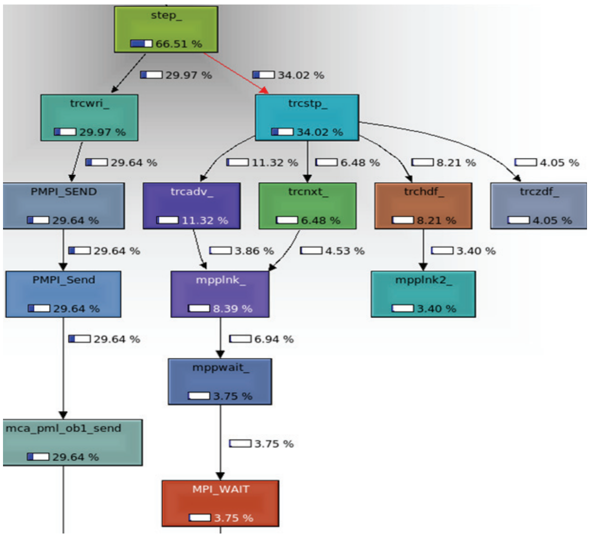


Fig. 2. A fragment of the application call graph (obtained with KCacheGrind tool of the Valgrind tool suite)

In order to proceed with the communication analysis efficiently, the phases of the application execution are to be identified. The localization of the most computation- and communication-intensive phases without the help of profiling tools is a non-trivial and quite complicated process that requires deep understanding of the application source code as well as the model basics [19]. However, a so called application call graph from a profiling tool [20] is sufficient for basic understanding of dependencies between the regions of the application as well as time characteristics of the communication events in those regions. A fragment of the application call graph for the most important execution phases is presented in Fig. 2.

Such a run profile analysis is an excellent starting point for further investigation of the message-passing communication in the parallel application. This can be done by performing profiling of MPI operations which account for the most significant application execution regions.

## IV. MEASUREMENTS AND ANALYSIS RESULTS

This section gives an overview of characteristics of the application run profile. The most communication- and computation-intensive phases of the application execution are identified for both test (3 steps of numerical solution) and standard (816 and more steps) use cases. The measurements and analysis results are presented as well.

### IV.1. Analysis of the Application Run Profile

For application run profiling we used tools from the instrumentation framework (e.g. the Valgrind tool suite [21]). Assuming that the communication mechanism implemented in the main simulation step routine does not depend on iterations, we were able to limit the number of

Table 1. Time distribution among the main phases of the execution (for the test use case)

| Phases of the execution | Operations performed | # iterations, real case | # iterations, test case | Computation | | MPI-calls | | Total | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Durat. [s] | Percent [%] | Durat. [s] | Percent [%] | Durat. [s] | Percent [%] |
| 1. Initialization, Input | Loading of input data | 1 | 1 | 939 | 99 | 6 | <1 | 945 | 80 |
| 2. Main simulation loop | Internal MPI calls, halo cells exchange | 816 | 3 | 3 | 38 | 5 | 62 | 8 | 2 |
| 3. Data storage | Storing of output and restarting data on disc, internal MPI communication | 17 | 1 | 7 | 3 | 204 | 97 | 213 | 18 |
| Total times | | | | 949 | 80 | 226 | 20 | 1175 | ~100 |

Table 3. Scalability characteristics of the application (for the test use case)

| Phases of execution | 32 nodes, with one process per node | | | 64 nodes, with one process per node | | | Scalability coefficient, $t_{64}/t_{32}$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | Comput duration [s] | MPI calls duration [s] | Total time [s] | Comput. duration [s] | MPI calls duration [s] | Total time [s] | Comput. | MPI calls | Total |
| 1. Initialization, Input | 939 | 6 | 945 | 2238 | 6 | 2244 | 2,4 | 1 | 2,4 |
| 2. Main simulation loop | 3 | 5 | 8 | 2 | 5 | 7 | 0,7 | 1 | 0,9 |
| 3. Data storage | 7 | 204 | 213 | 3 | 170 | 173 | 0,4 | 1,25 | 0,8 |
| Total | 949 | 226 | 1175 | 2243 | 181 | 2424 | 2,4 | 0,8 | 2 |

iterations that are profiled in the main simulation routine. For this purpose a special test use case which required only 3 steps of the main simulation was specified. That corresponds to 90-minute real time of the ecosystem evolution.

The main results of profiling for the test use case are collected in Table 1. The timing characteristics are followed by results for the application scalability acquired by launching the application on a larger number of nodes (Table 3). It is important to emphasize that the time distribution of the different phases compared to the total run time differs significantly for the test use case and for a real long-term simulation that requires a larger number of steps of the numerical solution [14]. This is due to the fact that only the iterative part changes but not the initialization and finalization parts. The application scalability for running on a different number of nodes is also changing according to the size of the use case. Nevertheless, the internal characteristics of the iterative phase are iteration-independent and valid not only for the test use case but for all use cases.

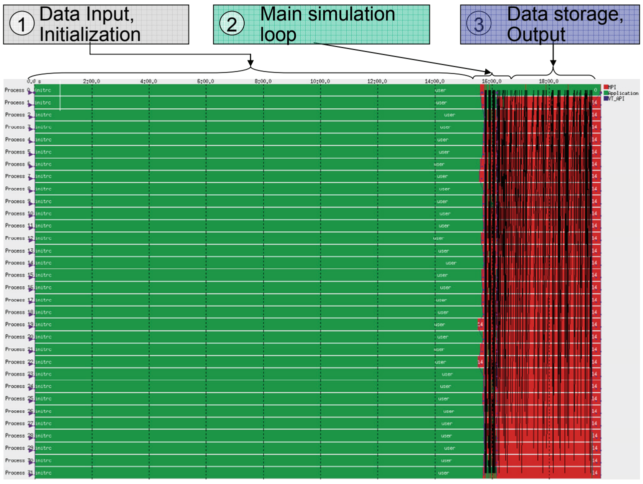


Fig. 3. Visualization of the application run profile for the test use case (obtained with Vampir tool)

While the most considerable part of the application is executed by operations of data input from the disk storage for the test use case (80% of the application execution time), the iteratively launched main simulation routine that implements a numerical solver will become a dominating part of the execution (up to 90% of the application execution) for a real production cycle use case that requires a big number of steps of a numerical solution. For example, for a two-week prediction 816 steps of numerical solution are required (see Table 1). Being launched only each 48th step of the numerical solution and at the end of the application execution, disk storage and data output operations are also an important point of the application performance optimization for both short-term and long-term forecasts.

Fig. 3 shows the time distribution among phases of the application execution for the test use case.

In the next subsection we describe in detail the profile of message-passing communication patterns in the main step simulation routine and data storage and output phase of the application execution.

## IV.2. The I/O Operations Profile

Data storage and retrieval in the application is performed in the self-describing and machine-independent NetCDF format [24]. Natively, NetCDF provides a programming interface for sequential I/O operations. Due to the lack of parallel output of contiguous data arrays in the NetCDF library, parallel applications must serialize their access to a NetCDF file. Optimally, the data input from a collection of NetCDF files should be performed in parallel from multiple processes concurrently. Although the newest implementations of the NetCDF-4 library [25] allows for such concurrent access by means of parallel I/O, the original NetCDF implementation used in the application is inadequate for most parallel applications because of the lack of an efficient access mechanism for parallel file systems. In consequence we observed very low performance of file input operations (see Table 1) that take approximately 940 seconds (80% of the execution time for the test case). Moreover, as the data input is implemented in a way that each process reads the complete data array from the NetCDF file, the input operations do not scale for a growing number of nodes the application is launched at; the duration of data input operations grows proportionally to the number of processes (Table 3).

Although the NetCDF file output duration does not show similar behavior when being launched for the double number of nodes, it also does not scale well (Table 3). In OPATM-BFM the serialization of the write-out mechanism is performed by the root process which composes the entire domain from all processes through MPI communication and then stores the complete dataset to a NetCDF file (Fig. 4). The message-passing communication pattern is described in detail in section IV.3. The OGS implementation of I/O routines is located in the functions *trcrst(), trcdit() and trcwri()* of the application source code.

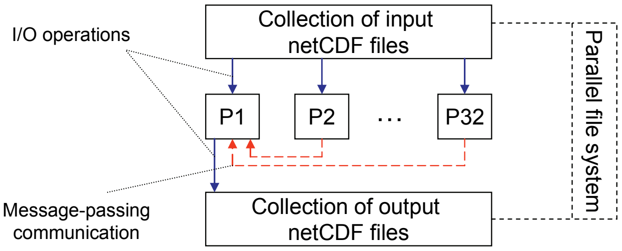

I/O operations
Collection of input netCDF files
P1
P2
…
P32
Parallel file system
Message-passing communication
Collection of output netCDF files

Fig. 4. The application I/O pattern

As time characteristics of the application execution show (Table 1), a sequential realization of the application I/O becomes therefore a considerable performance bottleneck for both short-term (input operations) and long-term (output and restart operations) simulations.

### IV.3. The Message-passing Communication Profile

The application is parallelized using the domain decomposition method. This required a message-passing mechanism for the data exchange among domains. This mechanism is implemented in the main step simulation routine (exchange of boundary elements of neighboring domains is performed at each simulation step) and in the output routine (*trcwri*) which performs the dataset storage in the NetCDF format

In the phase where the data storage and output operations are used to save the restart data, the blocking point-to-point communication is used for data gathering in the root process. It is performed by means of 612 MPI_Send calls that are executed by each of the non-root processes and a corresponding number of MPI_Recv calls in the root process (an all-to-one communication pattern). Cumulatively for the analyzed region, 18972 data messages are transmitted by means of MPI. The size of the transmitted messages varies from 4 kB (12648 messages) up to 1.125 MB (204 messages).

However, most MPI calls occur in the main step simulation routine. The frequency of message transmission in this phase of the application execution is very high (for the investigated use case a total of 252960 messages are being transmitted through MPI in the main step phase). Due to these factors, the main simulation step routine became the most important study point in our further investigations.

The exchange of data fields stored in different domains is realized by means of non-blocking point-to-point MPI operations (MPI_Isend, MPI_Irecv, MPI_Wait) in routines responsible for the simulation of the 3D advection scheme (trcadv, 816/408 MPI calls used in each non-boundary/-boundary domain), horizontal diffusion (trchdf, 7242/3621 MPI calls accordingly) and time integration (trcnxt, 102/51 MPI calls accordingly).

The size of a standard message is 72 kB for the advection and time integration routines (a total of 1821 MB and 227.6 MB of data are transmitted respectively) and 1 kB for the horizontal diffusion routine (224.5 MB). Hence, the total amount of transmitted data for one execution step amounts to 2273.1 MB even for this small test case.

Therefore, the application should be classified as communication-intensive. The optimization of the currently implemented communication pattern together with I/O pattern improvement will be important for further application performance and scalability improvement.

## V. OPTIMIZATION PROPOSALS

This section describes several optimization proposals for the current realization of application communication patterns as well as preliminary evaluation results of their impact on the application performance and scalability for the short test case. Estimation of the performance improvement expectation for the long-term forecast is given as well. We also examined the impact on the application performance of MPI I/O operations for data write/read operations.

### V.1. Parallel Realization of I/O Operations by Means of MPI-IO

While parallel file systems provide distribution of application input and output data on different storage elements of the Grid, any sequential implementation of I/O poses an obstacle for obtaining high application performance and scalability. On the contrary, parallel I/O provides many opportunities for optimized access to underlying file systems. For applications implemented with MPI the parallel I/O can be accessed through MPI-IO, the I/O mechanism specified in the MPI-2 standard [26].

Usage of I/O libraries built on top of MPI-IO for retrieving and storing data is therefore a good solution for the I/O optimization on a parallel file system. Parallel NetCDF (PNetCDF) follows this idea and is a library providing high-performance access to data arrays stored in the NetCDF format [27]. As indicated in [28], implementation of the application I/O by means of parallel operations is preferable compared with sequential operations and results in the significant improvement of I/O performance. OPATM-BFM will take advantage of PNetCDF using a number of optimization mechanisms provided by the underlying hardware-specific modules for MPI-IO [29].

PNetCDF differentiates between two distinct data modes – the collective and independent data mode. Similarly to MPI-IO, collective functions must be called by all the processes in the communicator associated to the opened NetCDF file, while the non-collective functions can be called by single processes. Independent operations provide only sequential data I/O and therefore require minimal modifications of the application source code due to the realization of the interface which is very similar to the native NetCDF API. On the contrary, using collective PNetCDF functions for accessing multidimensional datasets in a file requires essential modifications of the source code (especially with regard to the serialization of datasets for output in a file) but is necessary for obtaining high performance I/O, particularly at the large scale of processor nodes.

In our experiments, in order to evaluate the impact of MPI-IO on the application performance and scalability, the collective data mode was integrated in OPATM-BFM. Accordingly, some minor changes were required for the original OPATM-BFM source code. For function calls intended for creating/opening a NetCDF file modifications consisted mainly of adding a corresponding MPI communicator in the argument list to define participating I/O processes within the file's open/close scope that is the approach for applications using MPI. The most time consuming I/O operations of the application execution for the test case – data input and initialization operations – were ported PNetCDF using the collective data mode operations as well.

Profiling of the application execution after parallelization of the data input mechanism by means of MPI-IO has proven a dramatic improvement of the application I/O performance. As expected, due to the parallel realization the duration of the data input operations for the test use case decreased from 945 s measured for the sequential realization (see Table 1) to approximately 300 s. (more than 68% of time reduction). The difference between serial and parallel realization of I/O becomes even more significant when scaling the application to a larger number of nodes. For example, for 64 nodes the initialization I/O characteristics improved further by 84% (approximately 360 s compared to 2244 s measured for the previous non-parallel realization). Thus, usage of MPI-IO allowed the application to become more scalable for I/O operations for an increasing number of nodes (only 20% of time overhead for an increase from 32 to 64 nodes compared to 140% of overhead by the serial I/O realization).

Such significant impact on the application I/O performance and scalability characteristics shows high expediency of further usage of PNetCDF for the application I/O realization. Further improvements will concentrate also on optimization of the application output pattern (currently only done by master, see Fig. 4) through the realization of the parallel write-out mechanism to a NetCDF file from multiple processes with the support of the model developers.

### V.2. Optimization of Message-passing Communication Patterns

The message transmission mechanism that is currently used for the inter-domain communication (blocking MPI calls) is uniform – the order, structure and type of separately transmitted data do not change within the execution in each simulation step. Up to all 51 state variables are independently communicated, each using separate MPI communication calls. Due to this property the messages can be rearranged into segments transmitting more than one variable per MPI call. The current implementation foresees a transmission of only one variable per MPI operation (in the proposed implementation that would be the lower-bound segment size). Encapsulation of additional data into a segment enlarges the segment size and reduces the total number of MPI calls for data transmission. The highest data encapsulation level is reached by packing all messages into one single segment (that would be the upper-bound segment size). The effect is even larger for a network with high latency and low bandwidth. However, a procedure of packing/unpacking messages into/from a segment can require an additional computational overhead reducing fractionally the performance effect on the communication.

Therefore, a detailed study of the optimal number of segments (varying the segment size from the lower-bound to the upper-bound size) is required for both homogeneous and heterogeneous types of message encapsulation. In case of the heterogeneous encapsulation, an additional evaluation of the optimal segment size is necessary. For this purpose we have modified one of the application routines where the inter-domain communication is implemented (the time integration routine *trcnxt*) in order to allow the size of a segment to be explicitly specified by a user or an automatic tuning procedure. The first investigations for the current test configuration have proved that the data encapsulation influences positively the overall performance. The highest impact on reducing the communication time was reached using the segments with the upper-bound size (in the tested time integration routine the communication time was reduced by 50%). The message encapsulation mechanism will also be implemented for other application routines which perform the inter-domain communication (described in section IV).

As the parallel output mechanism for the application (as described in section V.1) is still under development, the data encapsulation can be also beneficial in terms of the application performance for the serialization of datasets from multiple processes to the root process that is required for the sequential data storage to a NetCDF file. This is implemented through the all-to-root communication pattern (currently implemented by means of blocking point-to-point MPI operations). However, for this pattern we have concentrated on using collectives instead of point-to-point operations.

As practical experience showed for the large-scale target platforms [22, 23], for the observed type of communication the usage of collective gather and reduce MPI operations is an advantage. We have investigated in detail the current implementation of the communication scheme and developed a new message transferring mechanism that is based on collective MPI operations and provides encapsulation of data arrays into segments similarly to the approach described above. For the test configuration the usage of collective MPI operations reduces the communication time by 50%. However, it is important to note that the optimal number of segments and the segment size for both analyzed types of MPI operations can differ for other configurations, target platforms, number of launched processes, implementations of the MPI library, etc., and this will be also an important point of our future research.

The realization of all the optimization proposals for the application communication and I/O patterns elaborated in this paper for the OPATM-BFM application allowed us to improve dramatically its performance for the test case (Table 4).

Whereas file I/O operations are dominating for the application execution for a small number of simulation steps (3 steps for the test case), the overall performance improvement due to optimization of the MPI communication becomes significant only for a long-term simulation (816 steps for the real case). As Fig. 5 shows, the total amount of realized optimizations allowed us to reduce the duration of the application execution for the real case by up to 5% (from initially measured 309 min. down to 293 min.). Furthermore, the optimization for the increasing number of nodes from 32 up to 64 grew up to 15% (from 213 min. down to only 185 min.). The application scalability grew accordingly from 145% up to 158%.

Table 4. Comparison of application time characteristics before and after optimization (for the test case, 3 steps of the numerical solution)

| Phases of execution | 32 nodes Total duration [s] | | 64 nodes Total duration [s] | |
|---|---|---|---|---|
| | Initial | Optim. | Initial | Optim. |
| 1. Initialization, Input | 945 | 300 | 2244 | 360 |
| 2. Main simulation loop | 8 | 7 | 7 | 6 |
| 3. Data storage | 213 | 207,5 | 173 | 172 |
| Total | 1175 | 514,5 | 2424 | 538 |

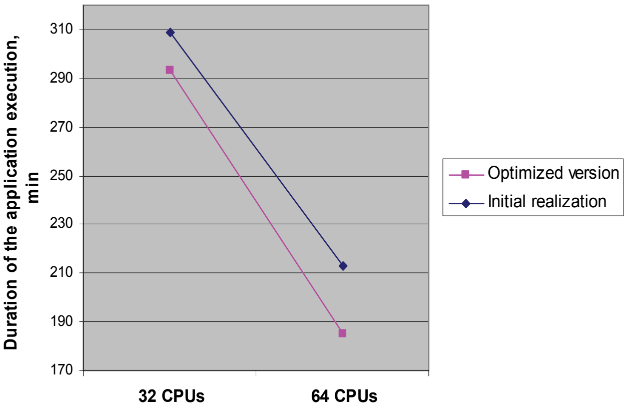


Fig. 5. The time characteristics of the application execution for the real case (816 steps of the numerical solution)

## VI. SUMMARY AND FUTURE DIRECTION OF RESEARCH

The OPATM-BFM is a scientific application that can make a strong benefit of using a Grid e-Infrastructure to increase the performance and scalability characteristics. However, the shortcomings of the current realization of the application I/O and internal message-passing communication pattern (e.g. sequentially accessing the input and output data, non-optimal size of messages transmitted by means of MPI, etc.) result into the application performance degradation that becomes especially dramatic after porting to such an e-Infrastructure. Hence, a requirement for obtaining the highest productivity of the application after porting to a Grid is the optimal realization of the internal communication and I/O patterns.

The article presents a technique of the applications performance and scalability characteristics analysis. Describing this technique, we used it to analyze in details the communication patterns of the OPATM-BFM. The technique was used for the analysis of the I/O pattern as well. This allowed us to identify the shortcomings of the application communication and I/O patterns as well as to elaborate several optimization proposals minimizing these shortcomings.

The optimization proposals for the I/O and communication patterns elaborated in this article have allowed us to increase dramatically the productivity of the OPATM/BFM application on a cluster of workstations. We proved benefits of using MPI-IO for accessing the collection of input/output files in the NetCDF format which allowed us to obtain very good application scalability for an increasing number of nodes. Combined with the optimization of message-passing communication patterns, we reduced the duration of simulation for the test use case (3 steps of the numerical solution) by up to 75% on 64 nodes. The duration of the application execution for the real case decreases as well. The application performance after porting to the grid is expected to increase accordingly.

On the other hand, the described improvements will also allow us to increase the application performance on the IBM SP-5 machine currently used for running the application’s main operation procedure. Besides that, the acquired results for the OPATM-BFM application will also be important for other scientific parallel applications that implement similar types of inter-process communication as described in the paper (e.g. all-to-one communication) and realize their I/O by means of sequential operations, in particular for applications in development. Furthermore, the presented technique used for the analysis should also support the OPATM-BFM application providers in further understanding the communication patterns in order to improve the load balance of the application for different use cases and define bottlenecks as well as work out a solution on how to resolve the shortcomings and maximize the performance and scalability of the application.

Other future research activities include further issues of the application integration within a Grid e-Infrastructure (e.g. analysis of application prerequisites for additional software packages installed on the Grid, data management, testing and debugging on Grid worker nodes with further launching through a Grid job from the user interface node, providing interactivity, etc.).

We will also concentrate our investigations on working out optimization proposals for several types of hardware and software architectures based on the obtained results (focusing especially on facilities of the currently provided DORII architecture). Furthermore, interesting possibilities of improving the communication include the development of a hybrid MPI+OpenMP programming model for the application.

Last but not least, in the future we plan to use the application as a use case for improving the internal profiling utilities of MPI implementations, e.g. Open MPI.

### Acknowledgments

This research has been performed in the frame of the DORII project. DORII is supported by the European Commission within the 7th Framework Program (FP7/2007-2013) under grant agreement no. RI-213110.

We are grateful to Dr. Giovanni Erbacci from CINECA who kindly provided the authors with access to the IBM-SP5 computer.

### References

[1] See the web page of the GMES project http://www.gmes.info

[2] See the web page of the Marine Environment and Security for the European Area (MERSEA) European Integrated project, http://www.mersea.eu.org

[3] A. Crise, P. Lazzari, S. Salon and A. Teruzzi, *MERSEA deliverable D11.2.1.3 – Final report on the BFM OGS-OPA Transport module*. 21 pp., 2008.

[4] See the description of the IBM SP5 machine on he CINECA’s web page, https://hpc.cineca.it/docs/user-guide-zwiki/SP5UserGuide

[5] See the web page of the DORII project, http://www.dorii.org

[6] I. Foster, *Service-Oriented Science*. Science 6 May 2005: Vol. 308. no. 5723, pp. 814-817.

[7] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury and, S. Tuecke, *The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific*

*Datasets*. Journal of Network and Computer Applications, **23**, 187-200 (2001) (based on conference publication from Proceedings of NetStore Conference 1999).

[8] B. Simo, O. Habala, E. Gatial and L. Hluchy, *Leveraging interactivity and MPI for environmental applications*. Computing and Informatics **27**, 271-284 (2008).

[9] See the web page of the Italian Group of Operational Oceanography, http://gnoo.bo.ingv.it

[10] M. Vichi, N. Pinardi and S. Masina, *A generalized model of pelagic biogeochemistry for the global ocean*. Part I: Theory. Jou. Mar. Sys. **64**, 89-109, 2007.

[11] See the web page of the OGS short term forecasting system of the Mediterranean Marine Ecosystem, http://poseidon.ogs.trieste.it/cgi-bin/opaopech/mersea

[12] J. J. Dongarra, S. W. Otto, M. Snir, D. Walker, *A message passing standard for MPP and workstations. Communications of the ACM.* **39 (7)**, 84-90 (July 1996).

[13] See the web page of the Mediterranean Ocean Observing Network, http://www.moon-oceanforecasting.eu

[14] A. Teruzzi, P. Lazzari, S. Salon, A. Crise, C. Solidoro, V. Mosetti, R. Santoleri, S. Colella and G. Volpe, 2008. *Assessment of predictive skill of an operational forecast for the Mediterranen marine ecosystem: comparison with satellite chlorophyll observations*. MERSEA Final Meeting, Paris, 28-30 April 2008.

[15] R. L. Graham, G. M. Shipman, B. W. Barrett, R. H. Castain, G. Bosilca and A. Lumsdaine, *Open MPI: A High-Performance, Heterogeneous MPI*. Proceeding of the conference HeteroPar '06, September 2006, in Barcelona, Spain, http://www.open-mpi.org/papers/heteropar-2006/heteropar-2006-paper.pdf

[16] *MPI: A Message-Passing Interface Standard Version 2.1*. Message Passing Interface Forum, June 23, 2008. http://www.mpi-forum.org/docs/mpi21-report.pdf

[17] See the description of the cluster “Cacau” on the web page of HLRS, http://www.hlrs.de/hw-access/platforms/cacau/

[18] A. Knüpfer, H. Brunst, J. Doleschal, M. Jurenz, M. Lieber, H. Mickler, M. S. Müller and W. E. Nagel, *The Vampir Performance Analysis Tool-Set*. Tools for High Performance Computing, Springer 139-156 (2008).

[19] R. Riesen. *Communication patterns*. IEEE 2006, http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=01639567

[20] J. Weidendorfer, *Sequential Performance analysis with Callgrind and KCachegrind*. Tools for High Performance Computing, Springer 93-114 (2008).

[21] J. Seward, N. Nethercote, J. Weidendorfer and the Valgrind Development Team, *Valgrind 3.3 – Advanced Debugging and Profiling for GNU/Linux applications*. http://www.network-theory.co.uk/valgrind/manual/

[22] O. Hartmann, M. Kühnemann, T. Rauber and G. Rünger, *Adaptive Selection of Communication Methods to Optimize Collective MPI Operations*. Parallel Computing: Current & Future Issues of High-End Computing, Proceedings of the International Conference ParCo 2005, G. R. Joubert, W. E. Nagel, F. J. Peters, O. Plata, P. Tirado and E. Zapata (Editors), John von Neumann Institute for Computing, Jülich, NIC Series **33**, 457-464 (2006).

[23] J. Pjesivac-Grbovic, T. Angskun, G. Bosilca, G. E. Fagg, E. Gabriel and J. J. Dongarra, *Performance Analysis of MPI Collective Operations*. Cluster Computing archive **10 (2)** 127-143 (2007).

[24] E. Hartnett and R. K. Rew, *Experience with an enhanced NetCDF data model and interface for scientific data access.* http://www.unidata.ucar.edu/software/netcdf/papers/AMS_2008.pdf

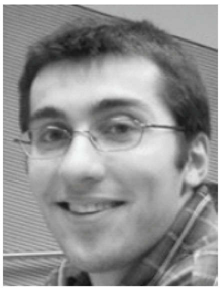
[25] R. Rew, E. Hartnett and J. Caron, 2006. *NetCDF-4: software implementing an enhanced data model for the geosciences, AMS.* http://www.unidata.ucar.edu/software/netcdf/papers/2006-ams.pdf

[26] Hakan Taki and Gil Utard, *MPI-IO on a Parallel File System for Cluster of Workstations*. IWCC, pp. 150, 1st IEEE Computer Society International Workshop on Cluster Computing, 1999.

[27] F. Hoffmann, *Parallel NetCDF*. Linux Magazin Julay 2004, http://cucis.ece.northwestern.edu/projects/PNETCDF/pnetCDF_linux.html

[28] J. Li, W. Liao, A. Choudhary, R. Ross, R. Thakur, W. Gropp, R. Latham, A. Siegel, B. Gallagher and M. Zingale, 2003: *Parallel netCDF: A High-Performance Scientific I/O Interface*. SC2003, Phoenix, Arizona, ACM. [29] R. Latham, R Ross and R. Thakur, *The Impact of File Systems on MPI-IO Scalability*. Preprint ANL/MCS-P1182-0604, June 2004.

**Dr. Alexey Cheptsov** is a research scientist at the High-Performance Computing Center Stuttgart (Germany). He attained his M.Sc. degree in Computer Science at the National University of Technology of Donetsk (Ukraine) in 2002 and his Ph.D. at the Research Institute of Simulation Problems in Power Engineering of the National Academy of Science of Ukraine in 2007. Throughout his scientific career he has conducted research and published in areas of simulation support of complex dynamic systems, deployment of technologically-oriented distributed simulation environments for diverse problem areas and parallel and high-performance simulation technology. His current research interests include providing Grid support for scientific applications, improvement of application communication patterns and parallel communication libraries. He is currently involved in the DORII project.



**KIRIL DICHEV** is a scientific employee at the High-Performance Computing Center Stuttgart (Germany). He earned his computer science degree (Dipl.-Inf.) at the Faculty of Computer Science, Electrical Engineering and Information Technology of the University of Stuttgart in 2007. He was involved in the Int.EU.Grid project and currently participates in the projects DORII and ParMA. His interests are the MPI application support on Grid systems and HPC resources as well as innovative methods of MPI profiling.

**STEFANO SALON** graduated in Physics (1998) and got a Ph.D. in Applied Geophysics and Hydraulics (2004) at the University of Trieste. Post-Doc at OGS from 2004 to 2007, he has been a researcher at OGS since December 2007. His research interests focus on turbulence in tidally-driven boundary layers, operational numerical forecasts of the Mediterranean ecosystem, dynamical downscaling of the ecosystem of the lagoon of Venice based on IPCC-SRES scenarios. He is a component of the Academic Board of the School of Doctorate in Environmental and Industrial Fluid Mechanics (Univ. of Trieste) where in 2008 he taught the course in Geophysical Fluid Dynamics. He is currently involved in the DORII project.

**PAOLO LAZZARI** got a Ph.D. in Environmental Sciences in 2008 at the University of Trieste. He is currently involved in numerical modeling at OGS as a research fellow. His activity regards the setup of 3D biogeochemical model at the Mediterranean scale. The aim of this system is to reproduce biogeochemical fluxes characterizing the ecosystem of the Mediterranean Sea, with particular focus to the lower part of the trophic web: primary production of phytoplanktonic functional groups and microbial loop. The model is based on the mathematical formulation of the transport reaction equation OPATM, where the reactive component introduced is a Biogeochemical Flux Model.