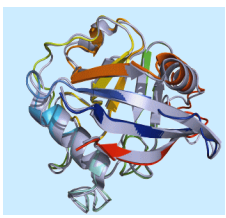
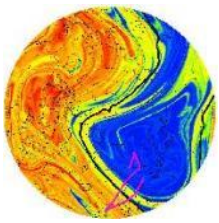
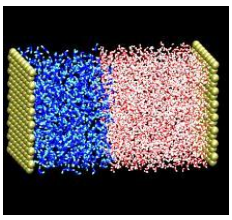
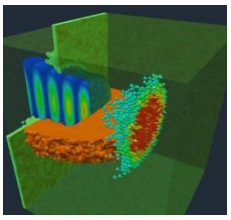
High-Level Support Initiative of the JSC Simulation Laboratories 2011

P. Gibbon, L. Arnold, D. Brömmel, V. Chihaia, S. Grießbach, R. Halver, L. Hoffmann, A. Karmakar, J. Meinke, S. Mohanty, Th. Müller, A. Schiller, G. Sutmann, O. Zimmermann

Institute for Advanced Simulation, Jülich Supercomputing Centre, Forschungszentrum Jülich

Contents

1. Introduction: SimLab Support Initiative 2011 3
2. Project summary 4
3. Highlights 5
3.1. Biology 5
3.2. Climate Science 6
3.3. Molecular Systems 8
3.4. Plasma Physics 10

1. Introduction: SimLab Support Initiative 2011

A central component of the Helmholtz Programme Supercomputing (Topic 1: Computational Science and Mathematical Methods) is the development of advanced software support through the recently established Simulation Laboratories (SimLabs) at JSC. This was motivated by the realisation that application software is lagging behind HPC hardware developments, which are now at the Petaflop-scale and beyond: it is becoming increasingly difficult to fully exploit the potential of these machines with single applications.

Four such units have now been established at JSC in the fields of Computational Biology, Molecular Systems, Plasma Physics and Climate Science, which came on stream in early 2011. All four labs have been actively engaged with user groups from their respective communities over the past two years, through various workshops, informal cooperations and 3rd-party projects. To channel these activities into a more formal structure, a „Support Call“ was issued in September 2010, inviting current and potential users of the supercomputers in Jülich from these four research communities to apply for high-level support from the Simulation Labs (see Appendix A).

Expertise offered by SimLab staff included:

1. (Re)design of computational methods needed to exploit highly parallel architectures
2. Performance analysis and scaling improvement of application codes and workflows
3. Porting of new codes to the JUROPA and JUGENE systems

In this pilot phase, proposed work packages were permitted to take up to two person-months of SimLab staff resources. The applicant or members of his/her group were also expected to contribute an equivalent amount of manpower to the project, particularly where the work involved substantial code and algorithm redevelopment.

Altogether 21 proposals were received, five of which were either postponed for lack of manpower (the Climate Science Lab comprised only 1 member at that time), or rejected as being outside the scope of the SimLabs' expertise or mandate. Selection decisions were taken in a plenary session involving all four SimLabs, which ensured that proposals which fell outside the expertise of the lab to which it was directed could potentially be transferred to another SimLab, or passed on to one of the cross-sectional teams at JSC. Due to the relatively low number of proposals per staff member (on average each lab had two full-time staff members as of January 2011), it was decided not to request external reviews and ranking for this round. A summary of all proposals can be found in the table on page 4.

After the proposal assignment, there followed a series of face-to-face consultations between the SimLabs and principle investigators of each proposal (PIs) to establish a work plan and time-frame for each project. This was largely completed by December 2010 so that work could begin in earnest in early 2011.

2. Project summary

Project #	Short Title	Code(s)	Task(s)	Scaling	Goal	PM request	PM spent
SLBIO1101	Parallel NGS de-novo assembly	Velvet, Abyss, Ray	Parallelisation	1	>200	2.0	3.0
SLBIO1102	Multi-million atom MD simulations	GROMACS	Scaling; electrostatics	1k	>10k	0.5	0.5
SLBIO1103	Molecular dynamics simulations at constant pH	GROMACS	Scaling, redesign	1k	100k	1.0	0.0
SLBIO1104	Conformational preferences of p53 peptide	PROFASI	Module design	32k		0.5	0.5
						4.0	**4.0**
SLCS1101	Stochastic atmospheric chemistry simulations	EURAD-IM	parallel scaling	200	20k	2.0	0.5
SLCS1102	CLaMS chemistry solver upgrade	CLaMS	algorithm redesign	128	>128	2.0	0.5
SLCS1103	Water vapor retrievals from SCIAMACHY measurements	SCIATRAN	basic parallelisation	1	8	0.5	0.5
SLCS1104	Optimisation of 2-D PREMIER retrievals	JuRaSSiC	single core tuning	256	256	2.0	2.0
						6.5	**3.5**
SLMS1101	Simulations of large ionic molecular systems		Electrostatics, benchmarking	N/A	4k	2.0	1.0
SLMS1102	Parallel Monte Carlo Initial Structure Generation	EMC	Parallelisation	1	>8	2.0	0.5
SLMS1103	IBIsCO Optimisation	IBIsCO	Scaling	<64	>64	2.0	3.0
						6.0	**4.5**
SLPP1101	Performance analysis of the PIC code EPOCH	EPOCH	Scaling, I/O, load balancing	1k	10k	1.0	1.0
SLPP1102	Scaling of the PIC code PICLS on JUGENE	PICLS	Scaling, I/O, 3D	1k	4k	2.0	1.0
SLPP1103	Scaling of adaptive Multigrid solver racoon	racoon	Scaling, BG/P optimization	100	10k	1.0	1.5
SLPP1104	Hybrid parallelization of the transport code RALEF	RALEF-2D	Parallelisation	8	>8	1.0	0.5
SLPP1105	Optimization of the code system EIRENE-B2	B2/B2.5	Parallelisation	1	64	1.0	2.0
						6.0	**6.0**
SLMS1104	Hydrodynamic Modeling of Macromolecules	UltraScan	Grid enabling (UNICORE)	10k	JSC	2.0	
SLPP1106	Port of AMR code racoon to GPUs	racoon	Porting GPU	N/A	hybrid	1.0	
SLCS1105	Characterization of river-aquifer interaction	EnKF3d-SPRING	Parallelisation	1	1k	1.0	
SLCS1106	2D Crosshole GPR Full-waveform inversion	2D_CH_GPR_FWI	Scaling	50	150	1.0	
SLCS1107	3D On-ground GPR full-waveform inversion	3D_OG_ GPR_FWI	Parallelisation	1	250	1.0	

3. Highlights

In this section accounts of example projects from each discipline are given as of December 2011. By this point most of the projects had either been completed or had reached a point where a second proposal would be a more appropriate way of continuing.

3.1. Biology

In the year 2011 the Simulation Laboratory Biology processed the following four support call projects:

SLBIO1101	Prof. Peter Nürnberg, Dr. Peter Frommolt	Cologne Center for Genomics	Parallel NGS de-novo assembly
SLBIO1102	Dr. Carsten Kutzner	MPI Göttingen	Multi-million atom MD simulations
SLBIO1103	Dr. Gerrit Groenhoff	MPI Göttingen	Molecular dynamics simulations at constant pH
SLBIO1104	Prof. Stefan Wallin	Lund University, Sweden	Conformational preferences of p53 peptide

Highlight: Conformational preferences of a p53 peptide (SLBIO1104)

The aim of this project was to enable the Monte Carlo code ProFASi to simulate the binding process of a 15-residue peptide from the C-terminus of p53 with 4 different proteins. The conformational preferences of this peptide are interesting because it can bind to 4 different target proteins, thereby assuming 4 different “native” structures. The software package ProFASi was chosen for this project because the force field implemented in ProFASi has been shown to successfully capture the folding and thermodynamic behaviour of many short peptide chains, through high statistics Monte Carlo simulations. Further, at least for short peptides, ProFASi is known to be relatively unbiased towards either helical or beta sheet secondary structures. These qualities are expected to be important for the simulation under consideration.

As the p53 peptide contains an acetylated Lysine residue, not present in the ProFASi model, support was needed to enable post-translational modifications in ProFASi, in particular Lysine acetylation.

The ProFASi code was extended to enable post-translational modifications. This involved redesigning the module named AA in ProFASi which contains a set of classes and functions for handling amino acids and N- and C-terminal capping groups. The new design permits post-translational modifications of amino acid side chains to be conveniently introduced as needed without touching the rest of ProFASi. For Acetylated-Lysine, ALY, geometrical parameters consistent with the bond-angle and bond-length approximations used in ProFASi were determined. A new ALY class was written, and it was tested with real simulations where ALY was part of a test peptide.

In addition an interaction potential was calibrated for acetylated Lysine. The calibration started with assigning some Lysine interaction strengths to Acetylated Lysine. ALY is uncharged, and hence dropped out of charged side-chain side-chain interactions. Hydrophobic strength was slightly increased relative to Lysine and the C atom in the new Acetyl group was introduced as one of the reference atoms for the contact based hydrophobicity term of ProFASi.

Simulations using modified code are currently running in Lund, Sweden and a joint paper is in preparation. The planned 0.5 PM were fully spent.

3.2. Climate Science

In the year 2011 the following four support call projects were executed by the Simulation Laboratory Climate Science:

SLCS1101	Dr. Hendrick Elbern	University of Cologne	Stochastic atmospheric chemistry simulations
SLCS1102	Dr. Jens-Uwe Grooß	FZJ/IEK-7	CLaMS chemistry solver upgrade
SLCS1103	Dr. Katja Weigel	University of Bremen	Water vapor retrievals from SCIAMACHY limb measurements
SLCS1104	Prof. Dr. Martin Riese	FZJ/IEK-7	Optimisation of 2-D PREMIER retrievals

Three additional proposals were submitted but could not be supported.

Highlight: Optimization of 2-D PREMIER retrievals (SLCS1104)

The Juelich Rapid Spectral Simulation Code (JURASSIC) is a fast radiative transfer model for the Infrared spectral region. It is used to simulate radiance observations made by satellite- or air-borne remote sensing instruments. The code is also used to retrieve atmospheric data (pressure, temperature, or trace gas concentrations) from remote sensing observations. At FZJ/IEK-7 and JSC the JURASSIC model is currently used to carry out feasibility studies for a new satellite mission proposed to the European Space Agency. Due to new instrumental techniques, the amount of measurement data to be provided by the PREMIER mission will exceed the data rates of current experiments by a factor 100. Feasibility studies indicate that the single core performance, as well as the parallelization scheme used by the JURASSIC model, both need to be improved to cope with the large amount of measurement data provided by future satellite missions like PREMIER.

Within the project the JURASSIC model was ported and optimized for usage on the supercomputer JUROPA. Single core performance and parallelization strategies applied in JURASSIC were measured and tuned based on the existing, well-tested application of PREMIER 2-D retrievals. Figure 1 illustrates outputs of function profiling and coverage testing as an example.

For this project 2 PMs were requested and 2 PMs were needed. The project goals were met and no problems were encountered. Outcomes of the performance analyses were highly valued input to European Space Agency (ESA) project reports. As a result of this work, the SimLab has been offered funding to participate in a follow-up ESA study during 2012.

Summary

The projects SLCS1103: 'Water vapor retrievals from SCIAMACHY limb measurements' and SLCS1104: 'Optimization of 2-D PREMIER retrievals' were completed successfully. An initial literature study for the project SLCS1102: 'CLaMS chemistry solver update' showed that the proposed solvers cannot provide the aspired performance improvements. The project SLCS1101 was postponed on request of the PI due to missing manpower.

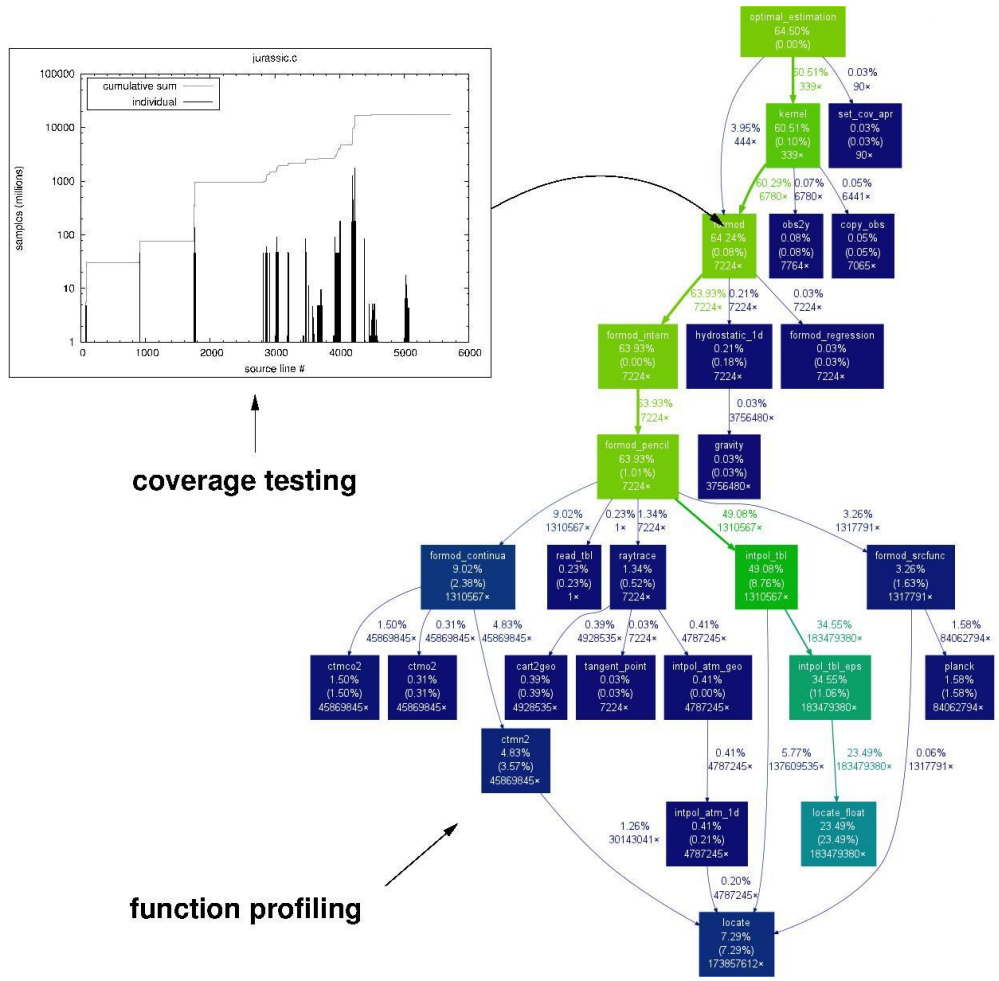


Figure 1: Outcomes of function profiling and coverage testing for the JURASSIC radiative transfer model. Both techniques were applied to optimize single core performance of the code.

3.3. Molecular Systems

The following three projects were carried out by the Simulation Laboratory Molecular Systems:

SLMS1101	Prof. Maxim Fedorov	MPI Leipzig & University of Strathclyde	Simulations of large ionic molecular systems
SLMS1102	Dr. Pieter J. in 't Veld	BASF Ludwigshafen	Parallel Monte Carlo initial structure generation for EMC
SLMS1103	Prof. Dr. Florian Müller-Plathe	TU Darmstadt	Extending the code IBIsCO to model surface interactions of polymers

Highlight: Extending the code IBIsCO for surface interactions of polymers (SLMS1103)

Tuning the code IBIsCO

The IBIsCO model uses coarse-grain molecular dynamics to simulate molecular liquids and polymers, discriminating between interphase and bulk properties and the simulation of polymerization processes (R-IBIsCO). Coarse-grained potentials are derived from radial distribution functions using the iterative Boltzmann inversion method.

We removed the reordering scheme in the subroutines *writetrj* and *average,* two routines that were very time consuming. The subroutines *takepos* and *takeen* are now called only once for the case that the subroutines *writetrj* and *average* are called in the main program in the same MD step. The communication scheme of the data between the processors was changed by assembling the different components of the positions and velocities, as well as the atom and cell indices in the communication buffers DATA*n*_EXP or DATA*n*_IMP (where *n*=3, 5 or 8). Consequently, the subroutines *communicate* and *boundary* corresponding to the three simulation schemes (Leapfrog, DPD and LA), as well as the subroutine *trans* were changed. Some redundant calls of MPI_Barrier were also removed. The subroutine *boundaryv* is not called in the code and it was excluded from compilation.

The performance enhancement of the modified code (input files reading and data preparation are not considered) compared to the original version supplied by the TU Darmstadt group for the two sets of simulations are presented in Figure 2, which shows a 4X improvement in the larger run on 512 cores. For both simulation types the list-cell is updated at each 10 MD time steps.

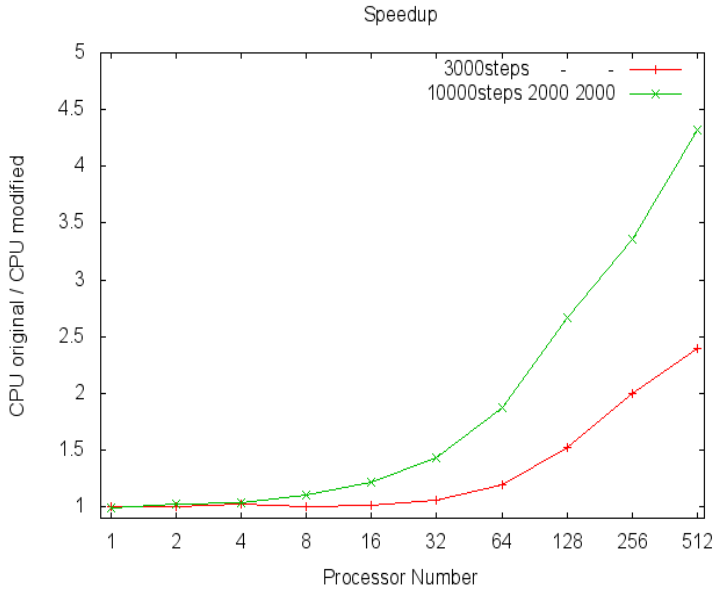


Figure 2: CPU time speed-up per MD step, for 3000 MD steps without trajectory writing and averaging (red curve); for 10000 MD steps with trajectory writing and averaging at each 2000 step (green curve).

The adaption of the code IBIsCO to surface calculations

To adapt the code to the surface calculations we chose to keep the parallel communication scheme and to modify the particle administration within the processor domains and in the link-cell list. In the current implementation, the surface slab is created only along the z direction, but in principle this scheme can be generalized to all other directions. The z direction was chosen because it is used in some other slab-type calculations that are implemented in the original version of the IBIsCO code.

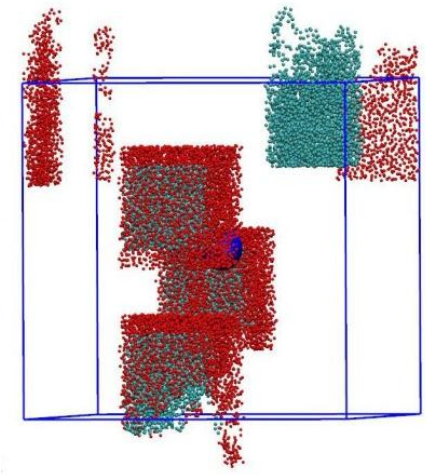


Figure 3: Altered decomposition scheme for surface calculations.

The particles are assigned or updated to the domains in the subroutines *division* and *trans*. To reduce the unbalanced distribution of the particles to the domains, the code can be adapted to work with a non-uniform size of the domains, but this scheme will increase the computing time because of the additional effort required to localize the particles to the domains. Because the average numbers of the particles assigned to the processor domains are very similar after the slab z-shift, we did not adapt the code to the non-uniform domains. For example, in a 4x4x4 processor scheme, each processor domain contains on average 2500 particles with a standard deviation of 227 particles – Figure 3.

3.4. Plasma Physics

The following five support call projects were carried out by the Simulation Laboratory Plasma Physics:

SLPP1101	Dr. Tony Arber,	University of Warwick	Performance analysis of the PIC code EPOCH
SLPP1102	Dr. Michael Bussmann	Helmholtz Zentrum Dresden-Rossendorf	Scaling of the PIC code PICLS on JUGENE
SLPP1103	Dr. Jürgen Dreher	University of Bochum	Scaling of adaptive Multigrid solver racoon
SLPP1104	Dr. Anna Tauschwitz	University of Frankfurt	Hybrid parallelization of the transport code RALEF
SLPP1105	Dr. Sven Wiesen	FZJ/IEK-4	Optimization of the code system EIRENE-B2

Highlight: Scaling of multi-grid solver racoon (SLPP1103)

The adaptive mesh refinement code racoon is developed at the Institute for Theoretical Physics I (Ruhr-University Bochum). Mainly it is used to study fundamental problems in plasma physics that require resolution of disparate scales like MHD turbulence, current sheet formation in ideal plasmas and magnetic reconnection. It has been successfully run on the JSC machines and yields formidable parallel scaling behaviour for hyperbolic systems like the compressible MHD equations on up to 256k cores. In order to simulate incompressible systems and implement implicit algorithms, a generic adaptive multi-grid (MG) solver was developed and is currently under re-design to be able to operate in staggered mesh arrangements. The original version of this elliptic solver scales to less than 1k cores.

The key goals of this support activity were to:

- analyse the MPI parallel multi-grid solver racoon
- improve scalability and execution performance
- achieve simulations on more than 2 racks (4k cores)

We planned with 1PM, however 1.5PM were invested.

An extended performance analysis, using the Scalasca tool, showed a couple of bottlenecks in the application, with large optimization potential. The first optimization aimed the volume exchange in the restriction phase of the MG solver. The application uses internal communication buffers to combine small messages into large ones and therefore achieve, in general, a better communication performance. However on the very large scales (128k cores) these buffers are much larger than the information needed to be sent and therefore there existed a large overhead by transmitting nearly empty buffers. A proper adjustment of the buffer sizes resolved this bottleneck.

The second task was to optimize the communication scheme and computational performance of the smoother. Here, again the buffer sizes had to be adjusted, as well as the way of boundary exchange on the coarsest level. The application used an overloaded strategy to solve the equations on the coarsest level, and even invoked MPI, although all computations were serial. Together with the racoon developers, we managed to achieve a simple and fast coarse grid solver for racoon. Finally, the solver was OpenMP-parallelized to reach even higher scaling.

The last optimization task aimed a proper MPI rank placement on JUGENE's torus network. Here, the ranks were placed along a Hilbert curve, according to the domain decomposition in the application.

The results of the SimLab work are summarised in Figure 4, which displays the successive reduction of the execution time (wall clock time, 128k cores) of the code after each of the above improvements had been applied.

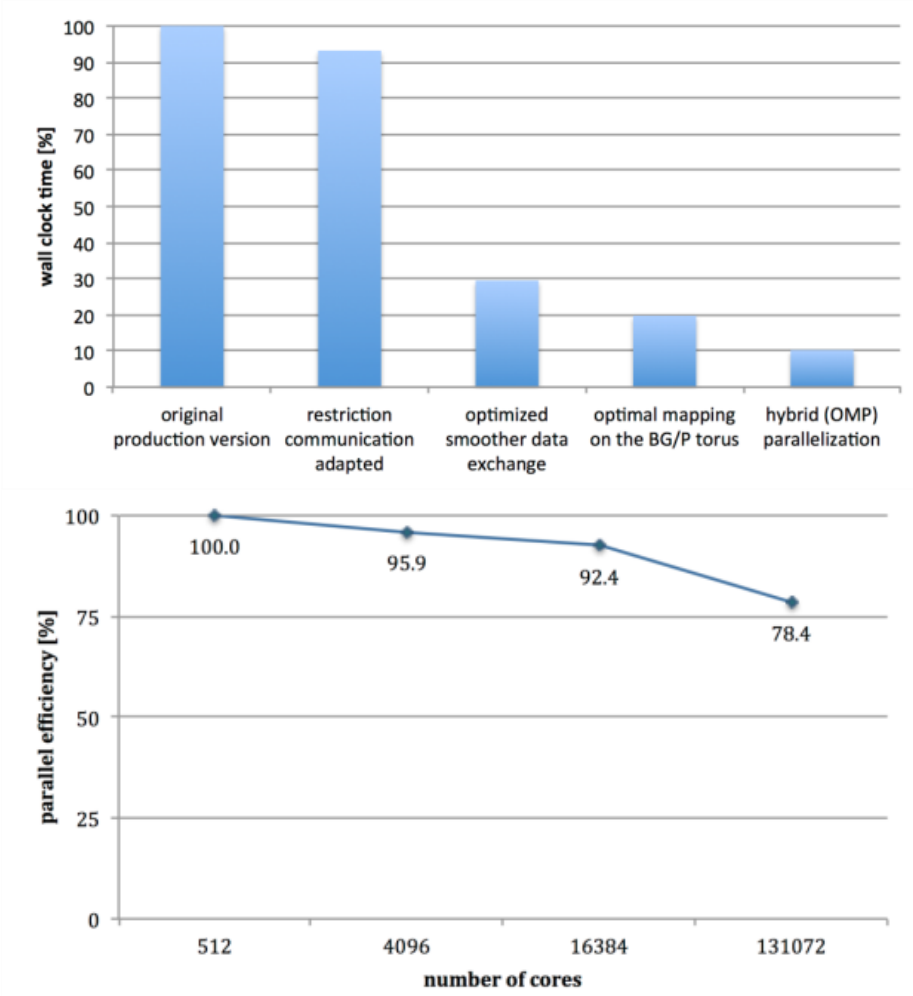


Figure 4: Top: reduction of the execution time on 128k cores BG/P due to various changes in the communication scheme. Bottom: parallel efficiency of the upgraded version.

As a lot of effort was spent on optimizing the communication, as well as the development of an hybrid parallelization strategy, the parallel efficiency has been boosted. The largest simulation scale of the original version was 4k cores, and we managed to run racoon efficiently on up to 128k cores. The following graph shows the parallel efficiency in a weak scaling benchmark.

In summary, we fulfilled all objectives in a very efficient collaborative work with racoon's developers, mainly Tobias Grafke. The only problems arose during the inclusion of OpenMP and SIMD instructions. A combination of both was not possible, as the IBM XL compiler did not produce proper SIMD instructions. As there are no fundamental reasons against the usage of both techniques at the same time, we assume that this is not possible with the IBM XL compiler. This issue was not followed further, as IBM reduced its compiler support for BG/P. We decided to use the OpenMP version, as it provides the slightly better performance.