

1 **Implementation and scaling of the fully coupled Terrestrial Systems Modeling Platform**
2 **(TerrSysMP v1.0) in a massively parallel supercomputing environment – a case study**
3 **on JUQUEEN (IBM Blue Gene/Q)**

4 F. Gasper^{1,2}, K. Goergen^{2,3,4}, P. Shrestha³, M. Sulis³, J. Rihani³, M. Geimer⁴, and S. Kollet^{1,2}

5 ¹Agrosphere (IBG-3), Forschungszentrum Jülich GmbH, Jülich, Germany

6 ²Centre for High-Performance Scientific Computing in Terrestrial Systems (HPSC TerrSys), ABC/J
7 Geoverbund, Germany

8 ³Meteorological Institute, University of Bonn, Bonn, Germany

9 ⁴Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH, Jülich, Germany

10 Keywords: coupled terrestrial modeling, high-performance computing, parallel scaling and
11 performance, profiling and tracing, code optimization

12 **Abstract**

13 Continental-scale hyper-resolution simulations constitute a grand challenge in characterizing non-
14 linear feedbacks of states and fluxes of the coupled water, energy, and biogeochemical cycles of
15 terrestrial systems. Tackling this challenge requires advanced coupling and supercomputing
16 technologies for earth system models that are discussed in this study, utilizing the example of the
17 implementation of the newly developed Terrestrial Systems Modeling Platform (TerrSysMP v1.0) on
18 JUQUEEN (IBM Blue Gene/Q) of the Jülich Supercomputing Centre, Germany. The applied coupling
19 strategies rely on the Multiple Program Multiple Data (MPMD) paradigm using the OASIS suite of
20 external couplers, and require memory and load balancing considerations in the exchange of the
21 coupling fields between different component models and the allocation of computational resources,
22 respectively. Using the advanced profiling and tracing tool Scalasca to determine an optimum load
23 balancing leads to a 19% speedup. In massively parallel supercomputer environments, the coupler
24 OASIS-MCT is recommended, which resolves memory limitations that may be significant in case of
25 very large computational domains and exchange fields as they occur in these specific test cases and
26 in many applications in terrestrial research. However, model I/O and initialization in the peta-scale
27 range require still major attention, as they constitute true big data challenges in the light of future exa-
28 scale compute resources. Based on a factor-two speedup due to compiler optimizations, a refactored
29 coupling interface using OASIS-MCT and an optimum load balancing, the problem size in a weak
30 scaling study can be increased by a factor of 64 from 512 to 32768 processes while maintaining
31 parallel efficiencies above 80% for the component models.

32

33 **1 Introduction**

34 In studies of the terrestrial hydrologic, energy and biogeochemical cycles, integrated multi-physics
35 simulation platforms take a central role in characterizing non-linear interactions, variances and
36 uncertainties of system states and fluxes in reciprocity with observations. Recently developed
37 integrated simulation platforms attempt to honor the complexity of the terrestrial system across
38 multiple time and space scales from the deeper subsurface including groundwater dynamics into the
39 atmosphere (Anyah et al., 2008 ; Fersch et al., 2013 ; Keyes et al., 2013 ; Maxwell et al., 2007 ;
40 Maxwell et al., 2011 ; Shrestha et al., 2014). Technically, the application of these new generations of
41 terrestrial modeling systems over regional climate-scale or micro-scale (e.g. large eddy simulation)
42 requires porting of the system to supercomputing environments, while ensuring ideally a high degree
43 of efficiency in the utilization of, for example, standard Linux clusters and massively parallel resources
44 alike. With such complex applications, a systematic scaling study and performance analysis including
45 profiling and tracing is crucial for understanding the runtime behavior, to identify optimal model
46 settings, and an efficient identification of bottlenecks in the program's parallelism. On sophisticated
47 leadership-class supercomputers, such as the 28-rack 5.0 Petaflops (Linpack performance) IBM Blue
48 Gene/Q JUQUEEN of the Jülich Supercomputing Centre (JSC) (Germany) used in this study, this is a
49 challenging task, in particular, when a coupled model system consisting of an external coupler
50 integrated with different component models is to be analyzed.

51 There exist a number of studies dealing with the detailed strong and weak scaling behavior of various
52 simulation platforms in hydrology and reactive solute transport, such as Hammond et al. (2014) ; Kollet
53 et al. (2010) ; Mills et al. (2007). In these studies the focus has been placed on the parallel efficiency
54 of solution algorithms including preconditioners for various classes and systems of partial differential
55 equations in global implicit and explicit solution approaches. In the presented study, the focus is
56 shifted from the analysis of parallel solver and preconditioner performance toward the challenges and
57 parallel efficiency of coupling different component models externally as part of the development of
58 (regional) earth system models.

59 The challenges and intricacies of coupling technologies of earth system models were reviewed by
60 Valcke et al. (2012), who focused on the central features of different established systems consisting of
61 data transfers, re-gridding, time step management, and parallel efficiency. Prominent examples of
62 coupled modeling systems are the Community Climate System Model, CCSM (Gent, 2006), and the
63 Earth System Modeling Framework, ESMF (Hill et al., 2006), which have also been shown to scale to
64 processor numbers on the order of 10^4 . As a matter of fact Dennis et al. (2007) explicitly discuss the
65 application of ultra high-resolution CCSM on the Blue Gene platform and the required preparations
66 with regard to, for example, memory allocations and parallel I/O due to this unique supercomputer
67 architecture.

68 The need for high- or hyper-resolution (e.g., 1km lateral grid spacing over continental computational
69 domains), coupled simulations of the terrestrial system originates from the multi-scale, non-linear
70 processes and feedbacks of the water, energy, and biogeochemical cycles in and between the
71 subsurface, land surface, and atmosphere (Wood et al., 2011). As a matter of fact, *ab initio*
72 simulations would require spatial resolutions in the sub-millimeter and sub-second ranges, in order to
73 resolve, for example, non-local reactive transport process in porous media (Yang et al., 2013) and
74 turbulent exchange between the land surface and the atmosphere (Shao et al., 2013). Additionally,
75 heterogeneity of the terrestrial system exists at all spatial scales resulting in variances and residence
76 time distributions of system's states and fluxes spanning orders of magnitude (Kirchner et al., 2000).
77 Thus, resolving all pertinent processes at their respective support scales and adequately honoring
78 cross-scale heterogeneity of the terrestrial system constitutes a grand challenge that may be tackled
79 by efficiently utilizing massively parallel supercomputing environments (Kollet et al., 2010).

80 The issue that subsurface hydrologic models usually run on a relatively small scale with high
81 resolution, while atmospheric models operate on a very big/continental scale, leads to unsolved
82 questions regarding the coupling of those models. A solution by upscaling the hydrology model to a
83 continental scale lacks adequate scaling laws for the continuity equations of variably saturated
84 subsurface flow (e.g., Richards' equation). Also, the downscaling of the atmospheric model to a
85 regional scale remains challenging due to the representation of turbulence and the lower boundary

86 condition in atmospheric models, that is, the land surface. A straightforward way to combine both
87 models in a soil-vegetation-atmosphere system is to increase the size of the hydrology model to a
88 continental scale, but leaving the resolution high. This requires computational resources only
89 massively parallel supercomputers like JSC's JUQUEEN can provide.

90 In this study, we present our experiences from porting, tuning, and scaling the parallel Terrestrial
91 Systems Modeling Platform (TerrSysMP) (Shrestha et al., 2014) from commodity Linux clusters to the
92 massively parallel supercomputing environment JUQUEEN, the IBM Blue Gene/Q system of JSC. We
93 aim at addressing and highlighting general technical aspects that have to be considered in designing,
94 porting, or refactoring fully coupled geoscience models to highly scalable High Performance
95 Computing (HPC) architectures. The study also demonstrates how an optimal resource allocation may
96 be achieved for such a complex modeling system with heterogeneous computing loads between the
97 different component models, and gives an example for a weak scaling study of the highly scalable
98 model system TerrSysMP.

99 **2 TerrSysMP, compute environment, and experiment design**

100 In this section, the modeling platform consisting of the different component models and coupling
101 technologies is introduced, followed by a description of the hardware characteristics of the JUQUEEN
102 (IBM Blue Gene/Q) supercomputer environment used in this study. The modeling platform was
103 instrumented with performance analysis tools, which are also outlined here. The design of the
104 numerical experiments for the ensuing scaling, profiling, and tracing analyses is detailed, including
105 remarks on an ad-hoc *a priori* load balancing of the different component models.

106 **2.1 The Terrestrial Systems Modeling Platform, TerrSysMP**

107 The parallel Terrestrial Systems Modeling Platform (v1.0) consists of the numerical weather prediction
108 system (COSMO, v4.11) of the German Weather Service (Baldauf et al., 2011), the Community Land
109 Model (CLM, v3.5) (Oleson et al., 2008), and the variably saturated surface-subsurface flow code
110 ParFlow (v3.1) (Jones and Woodward, 2001 ; Kollet and Maxwell, 2006). For details with regard to the

111 different component models, the reader is referred to the aforementioned publications. In TerrSysMP,
112 these component models were integrated in a scale consistent way conserving moisture and energy
113 from the subsurface across the land surface into the atmosphere (Fig. 1). The interested reader is
114 referred to Shrestha et al. (2014) for a detailed description of the modeling system. Each component
115 model is itself parallel and has been demonstrated to scale efficiently to a large number of parallel
116 tasks (e.g., Kollet et al., 2010).

117

Figure 1

118 In order to couple differently structured component models to simulate complex systems, it is
119 necessary to match a specified interface to exchange fluxes and states. Tailoring this interface
120 exclusively for a certain model environment does not provide the flexibility and compatibility that is
121 needed for various scientific modeling platforms. The obvious solution is a coupling strategy that
122 abstracts that interface via synchronous data-exchange, time step management, grid-transformation
123 and interpolation methods, and I/O with a low cost and strong stability on different computing
124 environments.

125 In TerrSysMP, the interface abstraction relies on the Multiple Program Multiple Data (MPMD)
126 execution model, which forms the basis of the external Ocean-Atmosphere-Sea-Ice-Soil coupler,
127 OASIS (Valcke, 2013). With the MPMD functionality, which is offered by most MPI-implementations, it
128 is possible to run several executables within the same global MPI_COMM_WORLD communicator.
129 This functionality affords a coupler that has an external "view" of all component models reflecting the
130 key requirement of high modularity and is especially useful in coupling of component models with fast
131 development cycles and heterogeneous computation loads (Chang et al., 1997). The implementation
132 of the coupler is almost non-invasive. Therefore component models remain independent which allows
133 for interchangeable executables as a major advantage. Thus, OASIS links the aforementioned
134 component models as independent executables, and can be implemented in two different versions,
135 OASIS3 and OASIS3-MCT (OASIS3 including the Model Coupling Toolkit libraries). In case of
136 OASIS3, the coupler is implemented as an additional independent executable, while in case of

137 OASIS3-MCT, the coupler is attached to each individual component model as a library. The impact of
138 coupling with OASIS-3 or OASIS3-MCT in massively parallel computer environments is discussed in
139 detail in the sections below.

140 It is important to note that coupling independent executables based on the MPMD paradigm may
141 confront the developer and user with basic technical drawbacks that need to be considered in the
142 initial design of the modeling platform. For example, the MPMD-functionality might not be available or
143 well supported on every machine, especially in case of customized MPI implementations. Additionally,
144 the assigned computational resources, that is, the number of parallel tasks per component executable,
145 are fixed at run time and, thus, load balancing between them has to be performed a priori. Moreover,
146 component models with relatively small computational load, even after load balancing, are constantly
147 blocking resources and use up allocated core-hours that cannot be made available to other users.

148 **2.2 Characteristics of JUQUEEN Blue Gene/Q**

149 JUQUEEN is an IBM BlueGene/Q system with 458,752 cores and 448 TB main memory with a
150 Linpack performance of 5.0 Petaflops. This makes JUQUEEN (Nov. 2013) the 8th fastest
151 supercomputer in the world (Top500.org, 2013).

152 Supercomputers like JUQUEEN have very special characteristics. Most remarkable is the trade-off in
153 clock rate (1.6 GHz) for smaller power/cooling requirements and improved system's reliability. This
154 trade-off in clock rate is compensated by the large number of cores and also the 4-way simultaneous
155 multithreading (SMT) of the 64Bit PowerPC A2 processors. The IBM BlueGene/Q architecture is
156 based on *nodes* which contain one CPU with 16 cores and 16 GB main memory. 32 of those nodes
157 are assembled in one (water cooled) *nodeboard*, which is also the smallest allocation unit for jobs.
158 One *rack* consists of 8 I/O nodes and 2 *midplanes* containing 16 nodeboards each. Compared to
159 standard Linux clusters, the IBM BlueGene/Q series is an architecture with very low memory per core.
160 The 16 GB RAM per node are distributed to 16 (64 with SMT4) cores and have a static mapping. Thus,
161 each MPI-process can only access 1 GB (256 MB with SMT4). While there is a workaround to enable

162 more memory per core, described later in the text, this is the most challenging constraint and
163 discussed in following sections.

164 An important feature of BlueGene/Q is the very fast interconnect, which links all nodes via a 5D torus
165 (electrical signaling within a midplane, optical signaling beyond midplanes). The 512 nodes of a
166 midplane are connected in a 4x4x4x4x2 configuration and allow for a very high peak bandwidth
167 (40 GB/s per node). The mapping of requested hardware allocations is left to the LoadLeveler job
168 scheduling system, which generally prioritizes large jobs (with maximum wall clock time), but smaller
169 jobs can be placed in the gaps. The mapping to the 5D torus can be a critical task for communication
170 intensive programs, however, requesting a certain configuration (*shape*) can result in increased
171 queuing times.

172 **2.3 Performance analysis**

173 Scalasca 1.4.3 was used as a profiling and tracing tool to analyze the runtime behavior of TerrSysMP,
174 identify performance bottlenecks and determine the optimum (static) load balance (i.e. resources
175 allocation for each experimental setup). Scalasca (Geimer et al., 2012) is a portable open-source
176 toolset which can be used to analyze the performance behavior of parallel applications written in C,
177 C++ and Fortran which are based on the parallel programming interfaces MPI and/or OpenMP. It has
178 been specifically designed for use on large-scale HPC systems such as the IBM Blue Gene series, but
179 is also well-suited for small- and medium-scale systems. Scalasca supports an incremental
180 performance-analysis procedure, combining runtime summaries (profiles) suitable to obtain a
181 performance overview with in-depth studies of concurrent behavior via event tracing. A distinctive
182 feature of Scalasca is its scalable automatic trace analysis (Geimer et al., 2010), which scans event
183 traces of parallel applications for wait states that occur, for example, as the result of unevenly
184 distributed workloads. Such wait states can present major obstacles to achieving good performance.

185 The typical Scalasca workflow is as follows: Before any performance data can be collected, the target
186 application is instrumented, that is, probes are inserted into the application to intercept important
187 events. Scalasca supports various ways to accomplish this task, for example, using automatic

188 compiler-based instrumentation, library interposition, or via source-to-source transformation. At
189 runtime, these probes trigger the collection of performance events to – by default – generate a profile
190 measurement providing a performance overview. Based on the initial profile results, the measurement
191 configuration can be optimized to reduce measurement perturbation, for example, by filtering small but
192 frequently executed functions. In-depth analyses of the performance behavior can then be performed
193 by collecting and automatically analyzing event traces, which allow to distinguish between wait states
194 and actual communication or synchronization time as well as to determine their root causes and
195 activities on the critical path (Böhme et al., 2010 ; Böhme et al., 2012).

196 To obtain information about the allocated memory, only an interface provided by IBM can be used
197 (`#include <spi/include/kernel/memory.h>`). This is due to the fact that the compute-nodes of JUQUEEN
198 use a specific compute node kernel with reduced functionality that does not offer generic memory
199 interfaces making the use of conventional memory tools impossible.

200 **2.4 Scaling study experimental design**

201 To identify scalability and performance limitations of TerrSysMP when going to very large model
202 domains either by increasing the spatial resolution or expanding the model domain to, for example,
203 continental scales, a weak scaling study with an idealized test case was developed. In the scaling
204 study, the two-dimensional horizontal extent of the model domain (nx , ny) was increased by a factor of
205 4 for each scaling step (doubling every dimension). The number of cells in vertical dimension, nz ,
206 remained constant for every scaling step with ParFlow $nz=30$, CLM $nz=10$, and COSMO $nz=40$. All
207 models use a two-dimensional processor topology and in the first scaling step, one Blue Gene/Q
208 nodeboard with 32 nodes and 512 physical CPU cores was used. The allocated resources are
209 doubled in each dimension as well and, thus, the patch-size (grid-cells per task) for every MPI rank
210 remains constant throughout the scaling experiment.

211 Time stepping remains constant across all scaling steps and is based on the physical processes
212 simulated and applied solution algorithms of the different component models. In the atmospheric
213 model COSMO, the time step size, Δt , is strongly determined by the spatial discretization and was

214 fixed at $\Delta t=10$ s. Time integration of the relevant exchange fluxes with the land surface and subsurface
215 model CLM and ParFlow is performed by OASIS over a 900 s interval, which simultaneously
216 constitutes the constant time step size of CLM and ParFlow. Note that in the presented scaling study,
217 file I/O is disabled as far as possible. The reason for this is the missing parallel file I/O in some
218 component models and memory limitations in case of large domain sizes.

219 The scaling study is performed with two different setups in terms of grid size and processor allocation
220 (Table 1).

221

Table 1

222 1) In the first setup, a grid size, n , is used that is closely related to real-data test cases used by
223 Shrestha et al. (2014) for development and testing of TerrSysMP. The initial scaling step consists of
224 $n_x=n_y=288$ grid-cells for CLM and ParFlow with a lateral spatial discretization of $\Delta x=\Delta y=0.5$ km and
225 $n_x=n_y=144$ for COSMO with a lateral spatial discretization of $\Delta x=\Delta y=1$ km. An optimal hardware
226 distribution was used, which was predicted with profiles from the analysis tool Scalasca and the
227 method described in Section 3.2. The profiling showed minimal wait states (critical path) with a
228 processor allocation (starting with one nodeboard/512 MPI-ranks) of $8 \times 8=64$ for CLM and ParFlow and
229 $24 \times 16=384$ for COSMO. This results in patch sizes of $(288 \times 288 \times 30)/64=38880$ grid-cells for ParFlow,
230 $(288 \times 288 \times 10)/64=12960$ for CLM and $(144 \times 144 \times 40)/384=2160$ for COSMO.

231 2) In the second scaling setup, the grid sizes, n , and number of processors, np , are expressed as a
232 power of two to provide a more standardized experiment for better comparability. In this setup, the
233 compute resource allocation is not possible in an optimal sense, since the load of the component
234 models is roughly distributed like: 75% / 12.5% / 12.5%, which does not follow powers of two. The first
235 step has grid sizes of 256×256 for ParFlow and CLM and 128×128 for COSMO. The 512 MPI ranks
236 (one nodeboard) are distributed as: $16 \times 8=128$ for ParFlow and CLM and $16 \times 16=256$ for COSMO. This
237 results in patch sizes of $(256 \times 256 \times 30)/128=15360$ grid-cells for ParFlow, $(256 \times 256 \times 10)/128=5120$ for
238 CLM and $(128 \times 128 \times 40)/256=2560$ for COSMO.

239 In both setups, the parallel efficiency $E_{nb}(n)$ [%] in our study is defined as:

$$240 \quad E_{nb}(n) = \frac{1}{nb} \cdot \frac{T_1(n)}{T_{nb}(n)} \cdot 100 \quad (1)$$

241 Where $T_1(n)$ is the runtime with one nodeboard and $T_{nb}(n)$ the runtime with nb nodeboards and the
242 problem size n . Thus, in case of perfect parallel weak scaling without communication overhead, the
243 simulation platform would exhibit an efficiency of $E_{nb}(n)=100\%$ for nb nodeboards and the problem
244 size n .

245 **3 Results**

246 In this section, the implementation and building process of TerrSysMP is described, followed by an
247 introduction of an *ad hoc* load balancing approach for MPMD programs with the usage of performance
248 analysis tools. The execution of the designed scaling study and the reason why first attempts failed
249 due to memory restrictions are also presented in this section. This is followed by the advancements
250 with the new OASIS version with results and discussion.

251 **3.1 TerrSysMP implementation**

252 For coupled systems with independently developed model codes, it is unlikely that all components are
253 initially ready and efficient for various computing sites, compilers and libraries. In order to reach an
254 optimum single-node and component-model performance, TerrSysMP was initially ported to use IBM
255 XL compilers that may produce executables with the most efficient hardware utilization. To improve
256 the usability of the complete model system, which is developed in a standard Linux cluster
257 environment, fully automatized script-based install procedures allow for a very efficient and fast
258 application deployment. The most current release version of the TerrSysMP system is retrieved from a
259 master GIT repository and adjusted for the build environment for the machine, in our case JUQUEEN,
260 that is, little/big-endianness, library-paths and data-structures, similar to the GNU *autoconf* software
261 configuration package. Optional/experimental features (e.g., OASIS3-MCT, etc.) are also available for
262 integration during this procedure. In a second step, the complete model system is built and the run-

263 time environment (model settings, forcing data and job-scripts, etc.) is set up. In order to preserve
264 portability and legacy code, TerrSysMP does not make use of hardware intrinsics or interfaces to IBM-
265 APIs (i.e., L1P prefetcher, atomic operations, etc.). However, there are compiler options, which guide
266 the compiler to make use of architecture-specific benefits and help with constraints, in our case: `-O3 -`
267 `qhot -qarch=qp -qtune=qp`. The usage of these options enables a speedup of roughly a factor of two
268 for TerrSysMP. To allow for easy regression testing during model development and for first-time users
269 familiarizing themselves with the system, forcing data and model settings for well-defined real-data
270 and idealized test cases as well as reference results are provided.

271 **3.2 Optimum resource allocations for MPMD**

272 As already briefly mentioned in the explanation of TerrSysMP's coupling scheme in Section 2.1, in
273 most MPMD implementations, the resource allocation or association of hardware nodes to a certain
274 application is fixed during runtime. Usually, in many MPI implementations a different number of
275 executables is started through the invocation of the MPI parallel job launcher; processes are then
276 mapped onto the computational resources allocated by the job scheduler. On IBM Blue Gene/Q, a
277 mapfile has to be used in conjunction with MPMD to explicitly assign MPI ranks to the actual CPU
278 cores. This mapfile may either be set up before job submission to optimize the communication pattern
279 on the 5D torus network topology of the BG/Q, or the resources are assigned automatically by the
280 scheduler. The latter was used to define the mapfiles. In order to allocate the resources in a
281 performant way, an algorithm is used that first queries the assigned *shape* and then arranges the
282 resources in a way that an executable is distributed to adjacent nodes. This usually ensures low
283 latencies within the 5D torus interconnect.

284 This setup combined with CPU affinity means that a load balancing between the component models
285 during runtime is not possible and assigned resources are fixed. Thus, no dynamic load-balancing
286 algorithms are applicable. Since simulations may run for several hours, unbalanced resource
287 assignments have a strong impact on the parallel efficiency. Therefore, determining an approximate
288 load for every component model and applying a static load balancing in advance is a necessary

289 condition for an efficient utilization of resources. For TerrSysMP, using a profiling tool (on JUQUEEN
290 for example Scalasca) in conjunction with a graphical tool to visualize the profile (here CUBE-QT
291 (Song and Wolf, 2004)), provides a complete picture of the time spent within the individual models and
292 routines. With detailed knowledge of the synchronization and communication-structure (Fig. 2) of the
293 coupled system (or a critical-path analysis available in the newest Scalasca implementation), one can
294 identify which models are waiting for completion of others and, thus, are under- or overloaded. For
295 example, if Parflow has 30% LateSender waiting time in the corresponding receive call from CLM and
296 CLM is also waiting, it is clear, that COSMO needs about 30% more resources from, for example,
297 ParFlow. This might have to be iterated a few times, especially if the speedup saturates.

298 *Figure 2*

299 Figure 3 is a showcase for this workflow and shows two CUBE-QT screenshots of the fully coupled
300 TerrSysMP. In Fig. 3a, the load is not ideally balanced and the topology view (right) shows more cores
301 with higher load in the relevant functions than in the optimized balancing of Fig. 3b. In both
302 screenshots, the metric *Late Sender* was chosen and, thus, the displayed (accumulated) timings are
303 equivalent to this particular wait state (receiver waits for sender).

304 *Figure 3*

305 With this complete picture of TerrSysMP, it was possible to determine an improved load balance for
306 the test setup 1 in Section 2.4 and also characteristic real data test cases reacting positively to this
307 approach. For example, compared to established balancing methodologies based on component
308 intrinsic timing routines a 19% speedup was reached in this example. However, this method is only
309 precise if the actual setup is traced/profiled. In order to determine the distribution for our test setup 1,
310 24 hours were traced in scaling step 1. Since we are simulating an idealized test case (flat geometry
311 with homogeneous vegetation) we assumed negligible influence on the load distribution with
312 increasing domain sizes.

313 **3.3 Advanced coupling interface**

314 Nowadays, parallel scientific software applications are targeted mostly at architectures such as
315 commodity Linux clusters with fast interconnects, which are used regularly without major problems.
316 However, utilizing massively parallel supercomputers requires different approaches, not only because
317 of the architecture, but also because of complicated communication patterns, data-structures and
318 distinct optimization that may be possible or necessary. The individual component models, which are
319 used in TerrSysMP, are well tested at many different supercomputing sites, but coupling them
320 especially with a highly resolved hydrologic model based on an external coupler adds an additional
321 level of complexity.

322 TerrSysMP was first developed for a standard Linux cluster and then ported to JSC's IBM Blue
323 Gene/Q supercomputer JUQUEEN. A comparably small reference test case scaled reasonably well.
324 However, in order to use TerrSysMP as a model for large-scale, hyper-resolution simulations, the
325 applicability for much bigger domain sizes had to be explored. Scaling studies as described in Section
326 2.4 with resolutions from $nx=ny=288$ (CLM, ParFlow) / $nx=ny=144$ (COSMO) ideally up to $nx=ny=9216$
327 (CLM, ParFlow) / $nx=ny=4608$ (COSMO) were planned, while $nx=ny=2304$ (CLM, ParFlow) /
328 $nx=ny=1152$ (COSMO) were actually reached.

329 During initial scaling tests, an increase in problem size by a factor of four in the second scaling step
330 led to stalled simulations due to insufficient main memory. In contrast to most standard Linux clusters,
331 the IBM Blue Gene/Q uses a static memory map, which means that the nodes' memory is equally
332 distributed across the processes running on that node in MPI parallel setup (see also Section 2.2). This
333 configuration is fixed and cannot change during a simulation. Since the standalone external coupler
334 OASIS3 is only running with a single process, it can only use 1/16th of the RAM of an individual node
335 if all 16 CPU cores per node are to be used, which results in 1 GB using OASIS3 as the coupler,
336 although the rest of the node is unused (only one and the same executable may run on an individual
337 node). A workaround for enabling more memory to one CPU is to reduce the number of processes per
338 node ($nppn$), with the side effect, that this configuration obviously decreases the parallel efficiency of
339 the modeling system, especially because this process count also applies to all CPUs and thus, also to
340 all other component models.. For OASIS3, reducing $nppn$ to 4 and using only 1/4th of the nodes CPUs

341 results in 4GB of RAM which are available per process. Thus, for applications with large memory
342 requirements, such as TerrSysMP, the resource usage when coupling with OASIS3 may be inefficient
343 in non-standard supercomputer environments.

344 Investigating the memory problems further with JUQUEEN's memory-tracking interface, which
345 provides information on the actually allocated amount of memory, showed that in each coupling time
346 step, OASIS3 receives several arrays from each sending process of a certain component model. It
347 then repartitions all these local parts from the domain decomposition of each individual component
348 model into the full domain. In subsequent steps, re-gridding and also weighting algorithms are
349 performed. Then, the global domain is partitioned again into local parts and sent forward to the
350 receiving component model processes. The aforementioned memory transgression occurred due to
351 the use of arrays with the size of the complete model domain. This usually does not pose problems for
352 smaller domain sizes, especially on general-purpose Linux clusters, which usually provide more than
353 2 GB RAM per core including dynamic memory allocations. However, on JUQUEEN the allocation of
354 global domain sizes prohibits an extensive weak scaling. For example, if one would need to use just
355 one of JUQUEEN's racks, each process is allowed to store only 8192 double values as a local
356 partition in order to enable one node to gather a global domain. This limitation of the single-threaded
357 concept of OASIS3 indicates that it is (at least with regard to massively parallel supercomputers) only
358 applicable to medium grid sizes and processor counts.

359 In September 2012 CNRS/CERFACS released a new version of OASIS, namely OASIS3-MCT (since
360 May 2013 OASIS3-MCT_2.0), which now relies on the Model Coupling Toolkit, MCT (Larson et al.,
361 2005). In the new version, OASIS is not a standalone coupler, but a library that is included in the
362 different component models. The actual interface basically remains the same, which makes porting to
363 this new version straightforward. Implementing the coupling within a library leads to a parallel OASIS,
364 since the library is part of each process, which overcomes computational as well as bandwidth
365 bottlenecks. But most importantly, each process can send its data to the targeted processes without
366 the need for repartitioning a global array. This renders the coupling thinner and consumes only few

367 extra resources. Figure 4 shows an illustration of the coupling with a) OASIS3 and b) OASIS3-MCT.
368 With this newly designed coupling interface, scaling to very large model domains is possible.

369 *Figure 4*

370 **3.4 Weak scaling study**

371 By using OASIS3-MCT, the model system allows for domain sizes up to a resolution of $nx=ny=2304$
372 (CLM, ParFlow) and $nx=ny=1152$ (COSMO) grid points, which constitutes an increase in the problem
373 size by a factor of 64 as compared to the unit reference test cases applying the original OASIS3
374 coupling. A further scaling was not possible at this point because also in the component model
375 CLM3.5, arrays with global domain size are used. It appears that in newer CLM versions this bottle
376 neck has been removed. Further scaling steps might be possible after a newer CLM version has been
377 implemented into TerrSysMP.

378 The scaling plot (Fig. 5a) of setup design 1 (Table 1a) shows that the dynamic model kernels, here
379 called *driver* routines, scale well, which is essential for extended hyper-resolution runs in the context of
380 large-scale integrated terrestrial simulations. CLM has a parallel efficiency of almost 100% (98% in the
381 largest run) due to its 1D isolated column physics with no communication overhead. The driver takes
382 only a couple of seconds even in the larger runs. The COSMO driver has a parallel efficiency of
383 slightly above 92% (largest run; see dotted lines in Fig. 5a for driver efficiencies), but is the component
384 with the heaviest compute load, therefore dictating the total calculation time. The ParFlow driver
385 scales less well with about 82% parallel efficiency (largest run).

386 Figure 5 shows which bottlenecks eventually arise in the larger scaling steps preventing the coupled
387 system from efficient scaling. The initialization time of CLM increases drastically with each step. An
388 analysis of the code revealed that during initialization, the load-balancing algorithm is redundantly
389 done by every rank and dependent on the global grid size n and the number of processors np . Since
390 both grow with a factor of 4 between each scaling step, the initialization time in theory increases with a
391 factor of 16. The actual increase of the initialization time is a factor of 14.41 between the last two steps.

392 The scaling plot (Fig. 5b) of setup design 2 (Table 1b) shows a similar behavior. Only ParFlow shows
393 a decrease in parallel efficiency (68% in the largest run), which indicates a higher sensitivity to
394 communication with a larger number of MPI ranks (Kollet et al., 2010). Additionally, the initialization
395 time determined by CLM is higher because of the larger number of CLM ranks. The overall calculation
396 time is slightly higher than in setup approach 1, since the patch-size of the limiting component model
397 COSMO is larger.

398 *Figure 5*

399 **4 Summary and conclusions**

400 TerrSysMP was successfully ported to the massive parallel IBM BG/Q system JUQUEEN of the Jülich
401 Supercomputing Centre. In comparison to the domain sizes that could be run using the initial coupling
402 with OASIS3, the problem size could be increased by a factor of 64 while still maintaining very good
403 scaling factors and hence a high parallel efficiency using OASIS3-MCT. The study demonstrated that
404 an in-depth consideration of the hardware features and software environment is necessary to
405 efficiently operate fully coupled model systems based on the MPMD paradigm on massively parallel
406 architectures such as JUQUEEN. This is irrespective of the individual component model's
407 performance, as the coupling process adds significant additional complexity. Applying OASIS3 in
408 standard Linux cluster environments for external coupling is appropriate for medium domain sizes on
409 the order of 256 MPI ranks. Beyond medium domain sizes, OASIS3-MCT affords efficient coupling in
410 standard and massively parallel computer environments by overcoming mainly RAM-dependent
411 limitations. MPMD load balancing can be performed efficiently with profiling tools, such as Scalasca, to
412 optimize MPMD resource allocation and solve configuration restrictions, such as static resource
413 mapping. However, despite TerrSysMP's encouraging weak-scaling performance of the dynamic
414 kernels of the different components models, initialization and I/O need to be reconciled for processor
415 counts beyond one BG/Q midplane (8192 cores), which are required for large-scale hyper-resolution
416 simulations. Currently, the applicability of TerrSysMP is explored for fully coupled terrestrial
417 simulations over the pan European continent and simulations of a regional scale *virtual reality*.

418 **Code availability**

419 TerrSysMP is freely available for academic, non-profit research and application after registration at
420 <https://git.meteo.uni-bonn.de>.

421

422 **Acknowledgements**

423 We would like to thank the John von Neumann Institute for Computing and the Forschungszentrum
424 Jülich for providing the required compute time for the projects JICG43 ("Fractal Scaling of
425 Hydrodynamics at the Catchment Scale") and HBN24 ("Development, testing, and application of an
426 integrated soil-vegetation-atmosphere model") that were utilized in this study. The financial support by
427 the SFB/TR 32 "Pattern in Soil-Vegetation-Atmosphere Systems: Monitoring, Modeling, and Data
428 Assimilation" funded by the Deutsche Forschungsgemeinschaft (DFG) and the Geoverbund ABC/J
429 (<http://www.geoverbund-abcj.de>) is gratefully acknowledged.

430 **References**

- 431 Anyah, R. O., Weaver, C. P., Miguez-Macho, G., Fan, Y., and Robock, A.: Incorporating water table
432 dynamics in climate modeling: 3. Simulated groundwater influence on coupled land-
433 atmosphere variability, *Journal of Geophysical Research-Atmospheres*, 113, Artn D07103 Doi
434 10.1029/2007jd009087, 2008.
- 435 Baldauf, M., Seifert, A., Forstner, J., Majewski, D., Raschendorfer, M., and Reinhardt, T.: Operational
436 Convective-Scale Numerical Weather Prediction with the COSMO Model: Description and
437 Sensitivities, *Monthly Weather Review*, 139, 3887-3905, Doi 10.1175/Mwr-D-10-05013.1, 2011.
- 438 Böhme, D., Geimer, M., Wolf, F., and Arnold, L.: Identifying the Root Causes of Wait States in Large-
439 Scale Parallel Applications, *Parallel Processing (ICPP)*, 2010 39th International Conference on,
440 2010, 90-100,
- 441 Böhme, D., Wolf, F., De Supinski, B. R., Schulz, M., and Geimer, M.: Scalable Critical-Path Based
442 Performance Analysis, *Parallel & Distributed Processing Symposium (IPDPS)*, 2012 IEEE 26th
443 International, 2012, 1330-1340,
- 444 Chang, C.-C., Czajkowski, G., Eicken, T. v., and Kesselman, C.: Evaluating the performance
445 limitations of MPMD communication, *Proceedings of the 1997 ACM/IEEE conference on*
446 *Supercomputing*, San Jose, CA, 1997.
- 447 Dennis, J. M., Jacob, R., Vertenstein, M., Craig, T., and Loy, R.: Toward an ultra-high resolution
448 community climate system model for the BlueGene platform, *SciDac 2007: Scientific Discovery*
449 *Through Advanced Computing*, 78, U247-U251, Doi 10.1088/1742-6596/78/1/012030, 2007.
- 450 Fersch, B., Wagner, S., Rummeler, T., Gochis, D., and Kunstmann, H.: Impact of groundwater
451 dynamics and soil-type on modelling coupled water exchange processes between land and
452 atmosphere, *Climate and Land Surface Changes in Hydrology*, 359, 140-145, 2013.
- 453 Geimer, M., Wolf, F., Wylie, B. J. N., Abraham, E., Becker, D., and Mohr, B.: The Scalasca
454 performance toolset architecture, *Concurrency and Computation-Practice & Experience*, 22,
455 702-719, Doi 10.1002/Cpe.1556, 2010.
- 456 Geimer, M., Saviankou, P., Strube, A., Szebenyi, Z., Wolf, F., and Wylie, B. J. N.: Further Improving
457 the Scalability of the Scalasca Toolset, *Applied Parallel and Scientific Computing, Part II*, 7134,
458 463-473, 2012.
- 459 Gent, P. R.: Preface to special issue on Community Climate System Model (CCSM), *Journal of*
460 *Climate*, 19, 2121-2121, Doi 10.1175/Jcli9020.1, 2006.
- 461 Hammond, G. E., Lichtner, P. C., and Mills, R. T.: Evaluating the performance of parallel subsurface
462 simulators: An illustrative example with PFLOTRAN, *Water Resources Research*, 50, 208-228,
463 DOI: 10.1002/2012WR013483, 2014.
- 464 Hill, C., DeLuca, C., Balaji, V., Suarez, M., da Silva, A., Sawyer, W., Cruz, C., Trayanov, A., Zaslavsky,
465 L., Hallberg, R., Boville, B., Craig, A., Collins, N., Kluzek, E., Michalakes, J., Neckels, D.,
466 Schwab, E., Smithline, S., Wolfe, J., Iredell, M., Yang, W. Y., Jacob, R., and Larson, J.:
467 Implementing applications with the Earth System Modeling Framework, *Applied Parallel*
468 *Computing: State of the Art in Scientific Computing*, 3732, 563-572, 2006.
- 469 Jones, J. E., and Woodward, C. S.: Newton-Krylov-multigrid solvers for large-scale, highly
470 heterogeneous, variably saturated flow problems, *Advances in Water Resources*, 24, 763-774,
471 Doi 10.1016/S0309-1708(00)00075-0, 2001.
- 472 Keyes, D. E., McInnes, L. C., Woodward, C., Gropp, W., Myra, E., Pernice, M., Bell, J., Brown, J., Clo,
473 A., Connors, J., Constantinescu, E., Estep, D., Evans, K., Farhat, C., Hakim, A., Hammond, G.,
474 Hansen, G., Hill, J., Isaac, T., Jiao, X. M., Jordan, K., Kaushik, D., Kaxiras, E., Koniges, A., Lee,
475 K., Lott, A., Lu, Q. M., Magerlein, J., Maxwell, R., McCourt, M., Mehl, M., Pawlowski, R.,
476 Randles, A. P., Reynolds, D., Riviere, B., Rude, U., Scheibe, T., Shadid, J., Sheehan, B.,

477 Shephard, M., Siegel, A., Smith, B., Tang, X. Z., Wilson, C., and Wohlmuth, B.: Multiphysics
478 simulations: Challenges and opportunities, *International Journal of High Performance*
479 *Computing Applications*, 27, 4-83, Doi 10.1177/1094342012468181, 2013.

480 Kirchner, J. W., Feng, X. H., and Neal, C.: Fractal stream chemistry and its implications for
481 contaminant transport in catchments, *Nature*, 403, 524-527, Doi 10.1038/35000537, 2000.

482 Kollet, S. J., and Maxwell, R. M.: Integrated surface-groundwater flow modeling: A free-surface
483 overland flow boundary condition in a parallel groundwater flow model, *Advances in Water*
484 *Resources*, 29, 945-958, Doi 10.1016/J.Advwatres.2005.08.006, 2006.

485 Kollet, S. J., Maxwell, R. M., Woodward, C. S., Smith, S., Vanderborght, J., Vereecken, H., and
486 Simmer, C.: Proof of concept of regional scale hydrologic simulations at hydrologic resolution
487 utilizing massively parallel computer resources, *Water Resources Research*, 46, Artn W04201
488 Doi 10.1029/2009wr008730, 2010.

489 Larson, J., Jacob, R., and Ong, E.: The Model Coupling Toolkit: A new fortran90 toolkit for building
490 multiphysics parallel coupled models, *International Journal of High Performance Computing*
491 *Applications*, 19, 277-292, Doi 10.1177/1094342005056115, 2005.

492 Maxwell, R. M., Chow, F. K., and Kollet, S. J.: The groundwater-land-surface-atmosphere connection:
493 Soil moisture effects on the atmospheric boundary layer in fully-coupled simulations, *Advances*
494 *in Water Resources*, 30, 2447-2466, Doi 10.1016/J.Advwatres.2007.05.018, 2007.

495 Maxwell, R. M., Lundquist, J. K., Mirocha, J. D., Smith, S. G., Woodward, C. S., and Tompson, A. F.
496 B.: Development of a Coupled Groundwater-Atmosphere Model, *Monthly Weather Review*, 139,
497 96-116, Doi 10.1175/2010mwr3392.1, 2011.

498 Mills, R. T., Lu, C., Lichtner, P. C., and Hammond, G. E.: Simulating subsurface flow and transport on
499 ultrascale computers using PFLOTRAN, *SciDac 2007: Scientific Discovery Through Advanced*
500 *Computing*, 78, U387-U393, Doi 10.1088/1742-6596/78/1/012051, 2007.

501 Oleson, K. W., Niu, G. Y., Yang, Z. L., Lawrence, D. M., Thornton, P. E., Lawrence, P. J., Stockli, R.,
502 Dickinson, R. E., Bonan, G. B., Levis, S., Dai, A., and Qian, T.: Improvements to the
503 Community Land Model and their impact on the hydrological cycle, *Journal of Geophysical*
504 *Research-Biogeosciences*, 113, Artn G01021 Doi 10.1029/2007jg000563, 2008.

505 Shao, Y. P., Liu, S. F., Schween, J. H., and Crewell, S.: Large-Eddy Atmosphere-Land-Surface
506 Modelling over Heterogeneous Surfaces: Model Development and Comparison with
507 Measurements, *Boundary-Layer Meteorology*, 148, 333-356, Doi 10.1007/S10546-013-9823-0,
508 2013.

509 Shrestha, P., Sulis, M., Masbou, M., Kollet, S., and Simmer, C.: A scale-consistent Terrestrial Systems
510 Modeling Platform based on COSMO, CLM and ParFlow, *Mon. Weather Rev.*, 142, 3466-3483,
511 doi:10.1175/MWR-D-14-00029.1, 2014

512 Song, F., and Wolf, F.: CUBE User Manual, University of Tennessee, Innovative Computing
513 Laboratory, 2004.

514 Top500.org, Top 500 Supercomputer Sites: <http://www.top500.org/>, access: November 2013, 2013.

515 Valcke, S., Balaji, V., Craig, A., DeLuca, C., Dunlap, R., Ford, R. W., Jacob, R., Larson, J.,
516 O'Kuinghtons, R., Riley, G. D., and Vertenstein, M.: Coupling technologies for Earth System
517 Modelling, *Geoscientific Model Development*, 5, 1589-1596, Doi 10.5194/Gmd-5-1589-2012,
518 2012.

519 Valcke, S.: The OASIS3 coupler: a European climate modelling community software, *Geoscientific*
520 *Model Development*, 6, 373-388, Doi 10.5194/Gmd-6-373-2013, 2013.

521 Wood, E. F., Roundy, J. K., Troy, T. J., van Beek, L. P. H., Bierkens, M. F. P., Blyth, E., de Roo, A.,
522 Doll, P., Ek, M., Famiglietti, J., Gochis, D., van de Giesen, N., Houser, P., Jaffe, P. R., Kollet,
523 S., Lehner, B., Lettenmaier, D. P., Peters-Lidard, C., Sivapalan, M., Sheffield, J., Wade, A.,
524 and Whitehead, P.: Hyperresolution global land surface modeling: Meeting a grand challenge

525 for monitoring Earth's terrestrial water, *Water Resources Research*, 47, Artn W05301 Doi
526 10.1029/2010wr010090, 2011.

527 Yang, X. F., Scheibe, T. D., Richmond, M. C., Perkins, W. A., Vogt, S. J., Codd, S. L., Seymour, J. D.,
528 and McKinley, M. I.: Direct numerical simulation of pore-scale flow in a bead pack: Comparison
529 with magnetic resonance imaging observations, *Advances in Water Resources*, 54, 228-241,
530 Doi 10.1016/J.Advwater.2013.01.009, 2013.

531

532 **Table Captions**

533 Table 1: Summary of experimental design setup for scaling studies.

534 **Tables**

a)				
Design 1				
Scaling step	1	2	3	4
#gridcells per dimension (COSMO/ CLM/ ParFlow)	144 / 288 / 288	288 / 576 / 576	576 / 1152 / 1152	1152 / 2304 / 2304
#processors (COSMO/ CLM/ ParFlow)	24x16 / 8x8 / 8x8	48x32 / 16x16 / 16x16	96x64 / 32x32 / 32x32	192x128 / 64x64 / 64x64
cores	512	2048	8192	32768
nodeboards	1	4	16	64
midplanes	1/16	1/4	1	4 (2 racks)
b)				
Design 2				
Scaling step	1	2	3	4
#gridcells per dimension (COSMO/ CLM/ ParFlow)	128 / 256 / 256	256 / 512 / 512	512 / 1024 / 1024	1024 / 2048 / 2048
#processors (COSMO/ CLM/ ParFlow)	16x16 / 8x16 / 8x16	32x32 / 32x16 / 32x16	64x64 / 64x32 / 64x32	128x128 / 128x64 / 128x64
cores	512	2048	8192	32768
nodeboards	1	4	16	64
midplanes	1/16	1/4	1	4 (2 racks)

535

536 Table 1.

537 **Figure Captions**

538 Figure 1. Schematic of interaction processes between TerrSysMP component models.

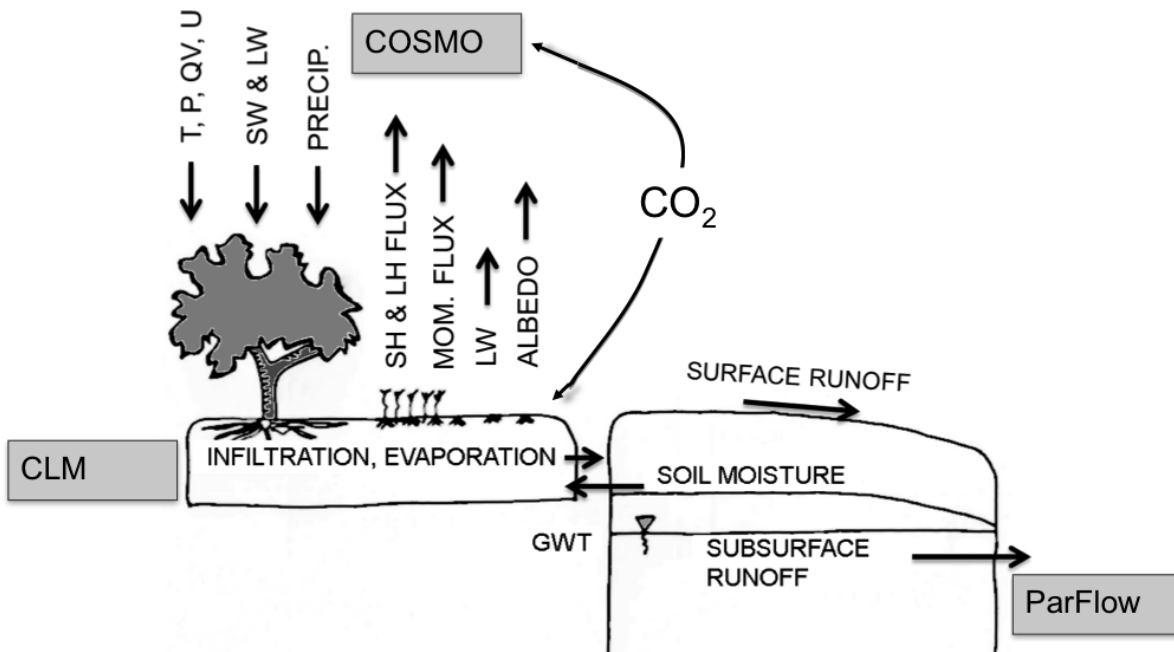
539 Figure 2. Schematic of the synchronization- and communication-structure. CLM receives first from
540 COSMO before receiving from ParFlow, thus, wait states in ParFlow are indicating an overloaded
541 COSMO. CLM calculation is very fast, but COSMO and ParFlow are idle during this time. CLM sends
542 first to COSMO before sending to ParFlow. CLM is idle during COSMO and ParFlow computation.

543 Figure 3. CUBE screenshots of the fully coupled TerrSysMP after 6 hour simulation time. In a) each
544 component model is naively distributed to one third of the resources (processor distribution: 192
545 COSMO, 160 ParFlow, 160 CLM). In b) the resources are distributed according to load, thus, the *Late*
546 *Sender* wait state is significantly reduced (processor distribution: 384 COSMO, 80 ParFlow, 48 CLM).
547 The topology view in b) shows fewer cores with *Late Sender* wait states where receivers are waiting
548 for senders in the relevant functions. The unit of the middle view is *Late Sender* waiting time
549 (accumulated over all CPUs). The units in the left and right view are percent.

550 Figure 4. Schematic of the coupling in TerrSysMP with OASIS3 (left) and OASIS3-MCT (right).
551 OASIS3 is a separate executable and coupling arrays are repartitioned to the full domain by OASIS.
552 OASIS3-MCT is part of each component model and coupling arrays only consist of the local fraction of
553 the full domain and are routed by OASIS to the destination processor.

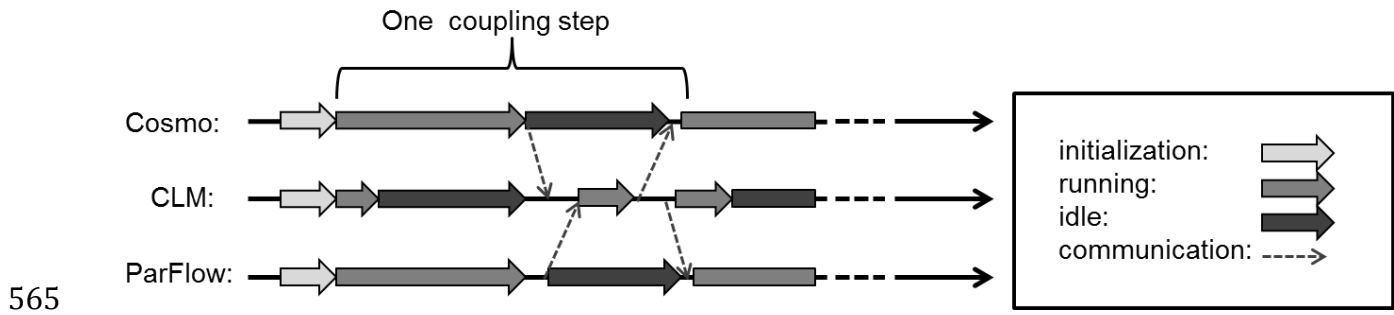
554 Figure 5. Idealized TerrSysMP weak-scaling study results with a) setup-design 1 ($nx=ny=288$, 288,
555 and 144 for ParFlow, CLM and COSMO, respectively) and b) setup-design 2 ($nx=ny=256$, 256, and
556 128 for ParFlow, CLM and COSMO, respectively). The dotted lines show the absolute timings of the
557 individual component models (green/COSMO is bounding the calculation time). The colored areas
558 show the stacked absolute timings of the calculation, initialization and finalization time. The solid lines
559 show the parallel efficiency of the relevant components on the secondary axis. The computational
560 problem size, n , as well as the assigned CPU cores, np , is increasing with a factor of 4 between each
561 step.

562 **Figures**



563

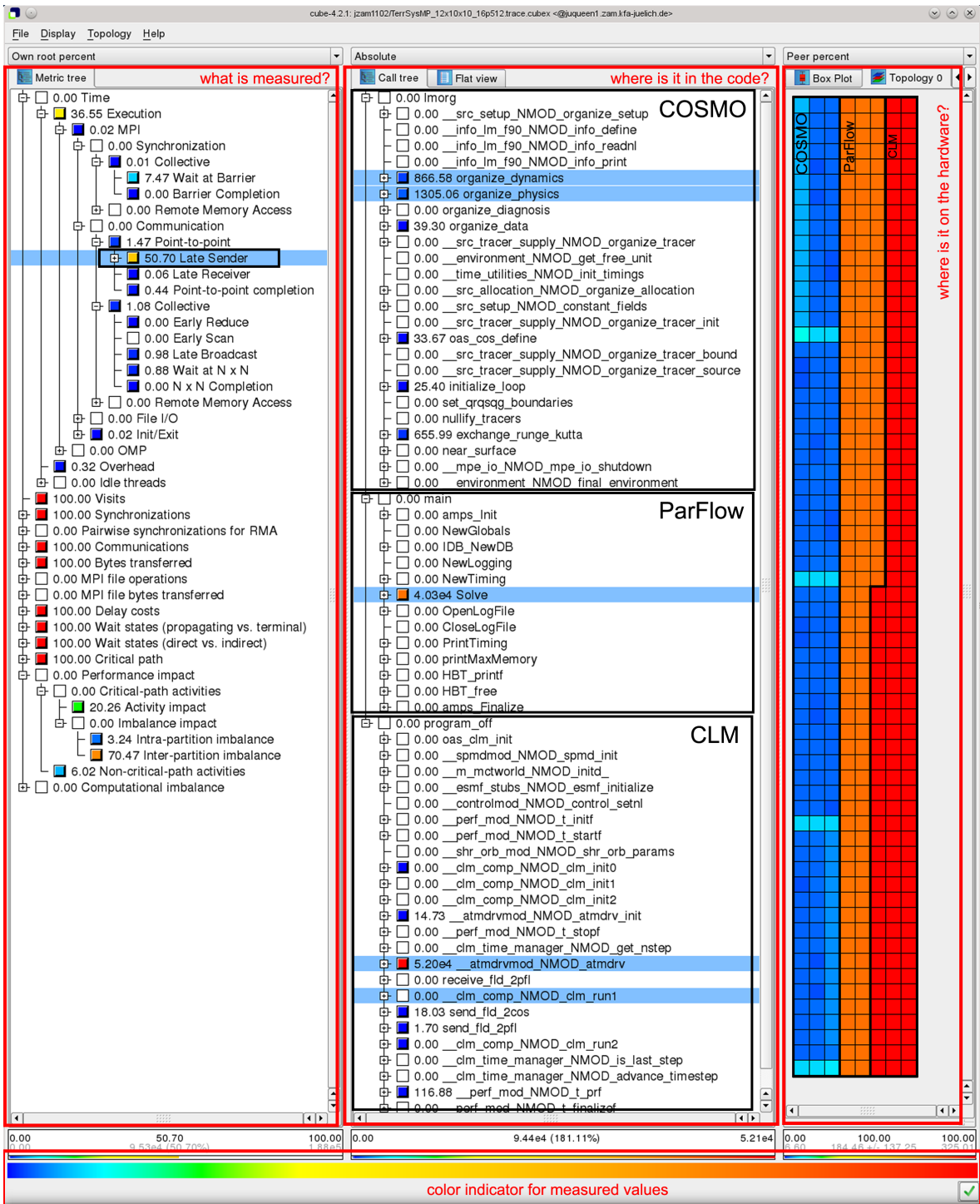
564 Figure 1.



565

566 Figure 2.

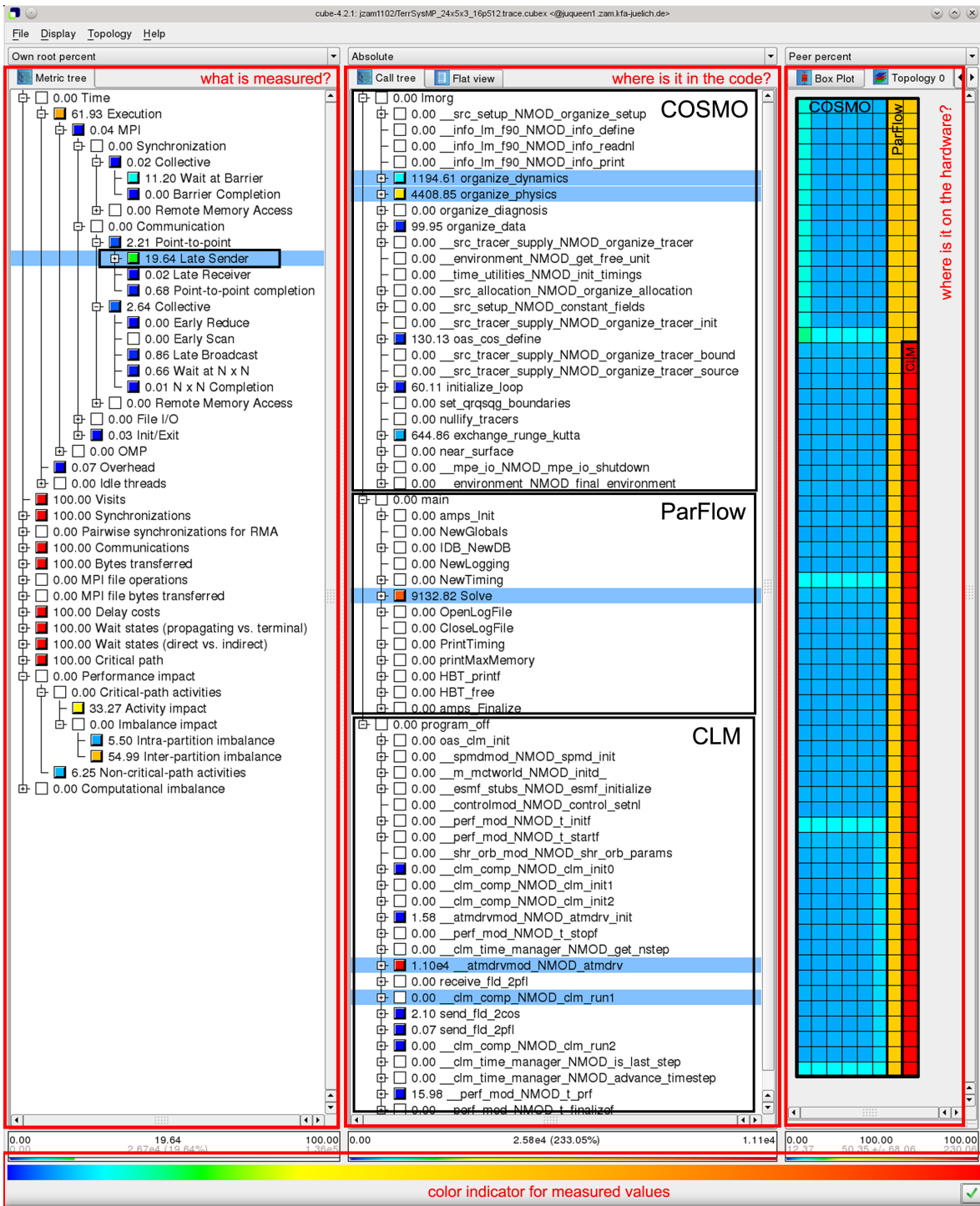
567 a)



568

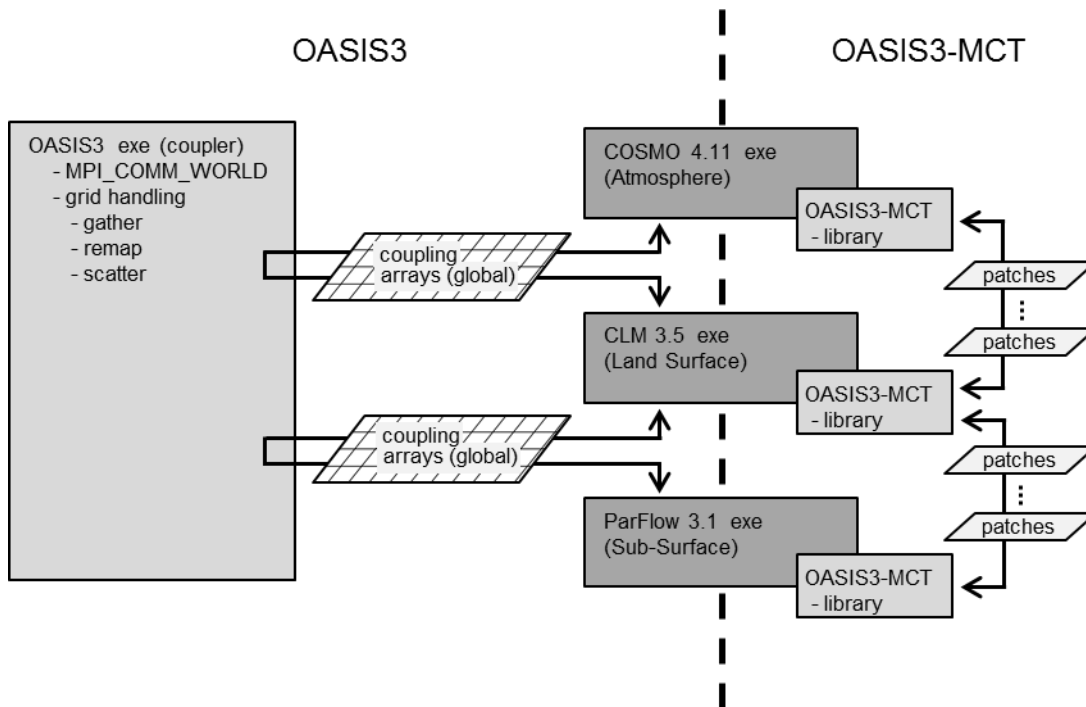
569 Figure 3a.

570 b)



571

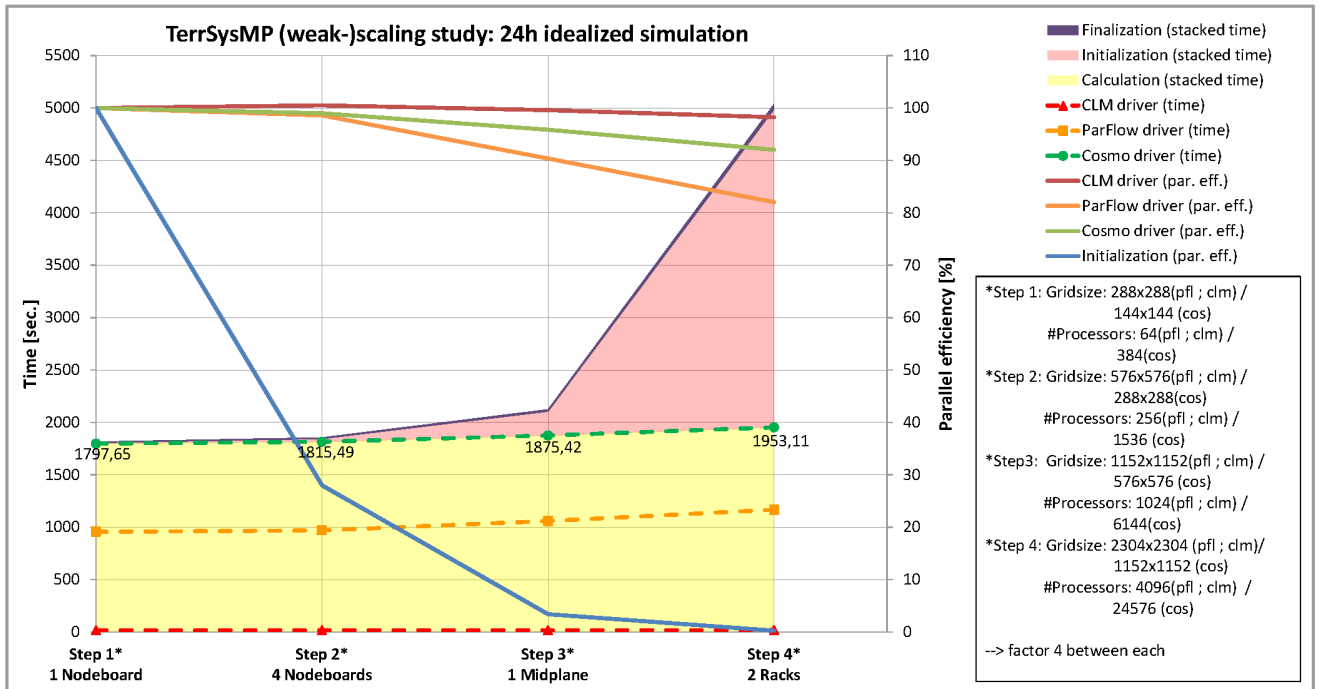
572 Figure 3b.



573

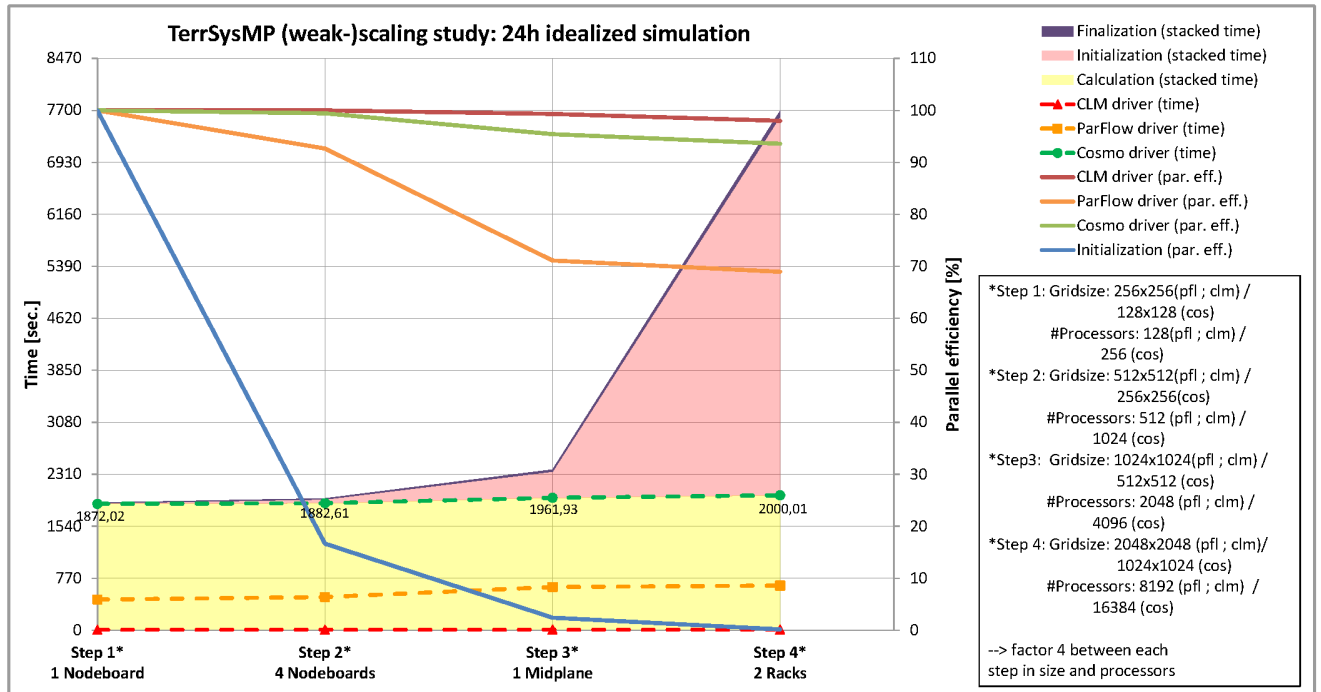
574 Figure 4.

575 a)



576

577 b)



578

579 Figure 5.