CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2017; 29: e3853
Published online 19 May 2016 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cpe.3853

SPECIAL ISSUE PAPER

On the complexities of utilizing large-scale lightpath-connected distributed cyberinfrastructure

Jason Maassen[1,*,†], Ben van Werkhoven[1], Maarten van Meersbergen[1], Henri E. Bal[2], Michael Kliphuis[3], Sandra E. Brunnabend[3], Henk A. Dijkstra[3], Gerben van Malenstein[4], Migiel de Vos[4], Sylvia Kuijpers[4], Sander Boele[5], Jules Wolfrat[5], Nick Hill[6], David Wallom[7], Christian Grimm[8], Dieter Kranzlmüller[9], Dinesh Ganpathi[10], Shantenu Jha[10], Yaakoub El Khamra[11], Frank O. Bryan[12], Benjamin Kirtman[13] and Frank J. Seinstra[1]

[1]*Netherlands eScience Center, Science Park 140, 1098 XG Amsterdam, The Netherlands*
[2]*Department of Computer Science, Vrije Universiteit, De Boelelaan 1081A, 1081 HV Amsterdam, The Netherlands*
[3]*IMAU, Utrecht University, Princetonplein 5, 3584 CC Utrecht, The Netherlands*
[4]*SURFnet, Kantoren Hoog Overborch (Hoog Catharijne), Moreelsepark 58, 3511 EP Utrecht, The Netherlands*
[5]*SURFsara, Science Park 140, 1098 XG Amsterdam, The Netherlands*
[6]*STFC Rutherford Appleton Laboratory, Harwell Oxford, Didcot OX11 0QX, UK*
[7]*Oxford e-Research Centre, University of Oxford, 7 Keble Road, Oxford OX1 3QG, UK*
[8]*DFN-Verein, Alexanderplatz 1, 10178 Berlin, Germany*
[9]*Leibniz Supercomputing Centre (LRZ), Boltzmannstraße 1, 85748 Garching, Germany*
[10]*RADICAL, Rutgers University, Piscataway, New Brunswick, NJ 08854, USA*
[11]*Texas Advanced Computing Center (TACC), 10100 Burnet Road (R8700), Austin, TX 78758-4497, USA*
[12]*National Center for Atmospheric Research (NCAR), 3090 Center Green Drive, Boulder, CO 80301, USA*
[13]*MPO, University of Miami, 4600 Rickenbacker Causeway, Miami, FL 33149, USA*

SUMMARY

In Autumn 2013, we—an international team of climate scientists, computer scientists, eScience researchers, and e-Infrastructure specialists—participated in the enlighten your research global competition, organized to showcase advanced lightpath technologies in support of state-of-the-art research questions. As one of the winning entries, our enlighten your research global team embarked on a very ambitious project to run an extremely high resolution climate model on a collection of supercomputers distributed over two continents and connected using an advanced 10 G lightpath networking infrastructure. Although good progress was made, we were not able to perform all desired experiments due to a varying combination of technical problems, configuration issues, policy limitations and lack of (budget for) human resources to solve these issues. In this paper, we describe our goals, the technical and non-technical barriers, we encountered and provide recommendations on how these barriers can be removed so future project of this kind may succeed.

Received 30 November 2015; Revised 20 February 2016; Accepted 5 April 2016

KEY WORDS: distributed cyberinfrastructure; lightpaths; global climate modeling

1. INTRODUCTION

Many scientific applications are of such complexity and scale that solutions can only be obtained by using a wide variety of computing hardware all at once. The need for concurrent use of multiple

*Correspondence to: Jason Maassen, Netherlands eScience Center, Science Park 140, 1098 XG Amsterdam, The Netherlands.
†E-mail: j.maassen@esciencecenter.nl

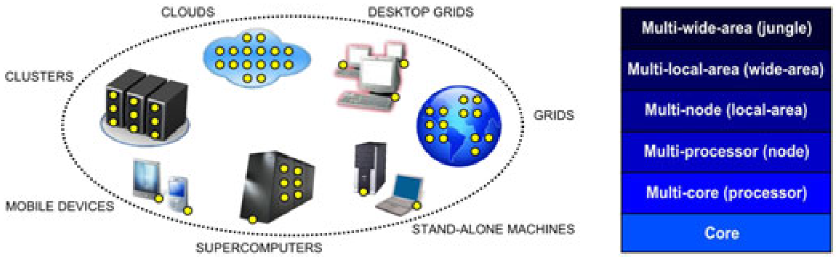

Figure 1. Left: A 'worst case' distributed cyberinfrastructure or jungle system [3], consisting of clusters, grids, clouds, supercomputers, and possibly even mobile devices. All resources may include hardware accelerators such as graphics processing units. Right: a distributed cyberinfrastructure is hierarchical, with parallelism potential at all levels.

clusters, grids, clouds, and supercomputers has spawned much innovative research into advanced distributed cyberinfrastructure [1, 2] and Jungle Computing [3, 4] (Figure 1). User-oriented programming models for such systems are under constant development (e.g., Ibis/Constellation [5], P* [6], PaRSEC [7]) with proven success (e.g., awards at AAAI'07, SCALE'08, DACH'08, SCALE'10, EYR3'11, EuroPar'14). Applications that benefit particularly from these advances are so-called multi-model/multi-kernel simulations [8], in which multiple distinct models of real-world phenomena are coupled to form a single, large simulation of a physical system. Such simulations are common in many research disciplines. In our domain of focus, climate research, distinct models of land, ocean, atmosphere, and sea-ice are typically combined to simulate the earth's climate, for example, using the community earth system model (CESM) [9].

An important application of CESM is the problem of the collapse of the meridional overturning circulation (MOC) in the Atlantic Ocean due to input of freshwater from Greenland Ice Sheet melting. State-of-the-art global climate models indicate that a weakening of the MOC would lead to a strong cooling of the North Atlantic region [10]. Constrained by compute power, most global climate models operate at a horizontal resolution of about one degree for ocean and sea-ice, and about two degrees for atmosphere and land. As a result, the ocean component does not resolve many important oceanic features, including so-called eddies (vortices with scales ranging from 10 to 100 km), requiring complex parameterization. The same holds for many atmospheric features. Recent research [11, 12] has indicated that high spatial resolution in both ocean and atmospheric model components does improve the simulation of many features in the climate system. In particular, the impact of changes of the Atlantic MOC are significantly different in eddy-resolving ocean models compared to models in which these eddies are parameterized [13].

To make significant steps towards the running of climate simulations at extreme resolution, the authors formed an international team of climate scientists, computer scientists, eScience researchers, and e-Infrastructure specialists, with the aim to participate in the enlighten your research global (EYR-global) competition, explained in more detail in Section 2. Our entry into the competition focused on the following components (explained in detail in Section 3):

- execution of CESM simulations with an extremely high resolution 0.02 degree (2-km scale) global ocean model, allowing eddies to be resolved and a new level of detail is revealed.
- utilization of trans-continental distributed computing resources, including four high-ranked Top500 supercomputers located in the Netherlands, UK, Germany, and USA;
- utilization of graphics processing units (GPUs) to improve performance of the ocean model;
- utilization of 10G lightpath technologies offered by the EYR-global organizers, to transfers intermediate state data among the participating supercomputing systems.

We have made good progress in the scaling up of the ocean modeling itself [14], and in the implementation and optimization of the distributed and hierarchical CESM simulations [15], with our work on GPU optimization receiving a best paper award nomination at CCGrid 2014 [16]. We have also made some progress in the distributed execution over a subset of the supercomputing resources available to us, the results of which are currently prepared for a separate publication.

DOI: 10.1002/cpe

However, overall, we have not succeeded in running all envisioned execution scenarios because of a combination of technical setbacks, configuration issues, policy limitations, and insufficient human resources to solve these problems (described in detail in Section 4). In Section 5, we present related work. Section 6 presents our conclusions and recommendations on which barriers need to be removed for future project of this kind to succeed.

2. ENLIGHTEN YOUR RESEARCH GLOBAL

The enlighten your research [17] series of competitions has been organized by the Dutch NREN SURFnet [18] since 2007. The main aim was to showcase and promote advanced lightpath technologies in support of state-of-the-art research performed in the Netherlands. Winners would be provided with financial means, access to dedicated lightpaths, and advanced networking expertise (manpower), to realize a proposed research project. Over the years, the competition has been extended to include further organizing partners, such as SURFsara (who maintains the Dutch national compute infrastructure), NWO (The Dutch Science Foundation), and the Netherlands eScience Center (providing support in research and engineering).

In 2013, under the name enlighten your research global, SURFnet established a collaboration with several National Research and Education Networks (NRENs): Funet [19] (Finland), Janet [20] (UK), and ESnet [21] and Internet2 [22] (USA). EYR-global became an international competition inviting researchers, and their collaborators from any discipline to submit proposals that highlight how access to world-class and international research networks would significantly improve their research and discovery process. The competition was explicitly open to a wide range of research teams, ranging from novice users to advanced experts in the use of information technology. The winners would be awarded with access to world-class network resources operated by the organizing NRENs, support and advice on end-to-end network connectivity, and NREN commitment on network resource provisioning.

All proposals were expected to be submitted by an international team of researchers but also include local network coordinators, responsible for supporting the technical implementation of the project. The research section was expected to comprise the team's research questions and the objectives of the proposed work, including the science methodology followed to solve the problem, the expected benefit of the granted network resources to the research, and the expected results in terms of research output. Finally, the technical section would have to describe the network requirements of the project, in particular, the requested network resources.

In the 2013 round, eight proposals were submitted. After a rigorous review process, including a consultation round by the organizing NRENs, four proposals were announced as winners during the SC13 conference in Denver, USA.

3. OUR EYR-GLOBAL COMPETITION ENTRY

Our EYR-global competition entry was submitted by a team of 19 members from 4 countries: the Netherlands, Germany, UK, and USA. The team was multi-disciplinary, including experts from climate science, high-performance, and distributed computing, and eScience. Team member roles ranged from coordinating tasks, to software and hardware implementation tasks, and support tasks. The current paper lists all team-members as co-authors, with additional authors from SURFnet who became full project members after the start of the project. The main ambition of the project was to use CESM to perform a 1-year climate simulation with the global ocean component at 0.02 degree (2 km) resolution on a combination of the following supercomputers available to us in 2013:

- **Stampede** [23] (#6 Top500), Texas Advanced Supercomputing Center (TACC), Austin, USA. Hardware: 102,400 cores, 6880 Xeon Phi co-processors, 128 NVIDIA GPUs, 5.2 PFlops/s.
- **SuperMUC** [24] (#9 Top500), Leibniz-Rechenzentrum (LRZ), Garching, Germany. Hardware: 155,656 cores, 2,9 PFlop/s.
- **Cartesius** [25] (not yet listed in Autumn 2013), SURFsara, Amsterdam, the Netherlands. Hardware: 13,984 cores, 271 TFlop/s.
- **Emerald** [26] (#379 Top500), STFC Rutherford Appleton Laboratory, Didcot, UK. Hardware: 1008 cores, 372 NVIDIA Tesla GPUs, 114 TFlop/s.

DOI: 10.1002/cpe

The main challenges of the project were initially expected to be in the linking up of these resources with advanced lightpath technologies, and in the development of the software to perform the simulations over the different machines. While the set aims were very ambitious, the team has ample experience running complex distributed applications in several domains, even at a world-wide scale (Section 5). As such, the project was well within the team's capabilities.

3.1. Rationale

Before describing the technical details of CESM, we like to stress the rationale underlying our work. One of the drivers is to enable execution of large-scale and extreme resolution simulations that are not easily executed on a single supercomputer. At least two valid arguments *against* this approach would be as follows:

(1) It is always possible to build a larger supercomputer such that complex distributed simulations are rendered unnecessary;
(2) A distributed simulation usually is not as fast as an identical simulation executed on a single supercomputer—in particular when significant parts of the simulation are tightly coupled.

The first argument relates to the ever increasing need for compute capacity in many research disciplines. When newer and larger machines become available to scientists, new research directions are made possible, to the effect that compute desires increase almost immediately again, often superseding the capacity of even the largest available machine. A distributed computing approach does not stop this accelerated need for speed; it may even provide further fuel to the acceleration. It does, however, allow researchers to run larger simulations and explore new research territories well before the next largest supercomputer is installed.

The second argument relates to the notion of efficiency. We agree that communication overhead, certainly at long-distance, may not contribute to the highest possible compute efficiency. In our view, however, efficiency is a multifaceted indicator of the research process as a whole, not only of the phase in which the set of simulations is executed at the largest scale. These simulations are part of a much longer pathway of scientific discovery and are often preceded by initial smaller simulations performed for testing, validation and parameter tuning, and followed by extensive analysis of the data that has been produced. The research process as a whole is significantly enhanced if all steps in this pathway can be performed somewhat interactively.

The reality is, however, that the simulations are often queued for days or weeks on large-scale production systems. However, when breaking up and distributing these simulations over a number of supercomputing system, each of the parts may be queued in waiting lines for much smaller jobs. As a result, the distributed simulation may start within minutes instead of days or weeks. This often overlooked opportunity has a major impact on the speed of the research process as a whole.

Further arguments for a distributed computing approach as applied by us are as follows:

- Many applications consist of distinct components, each requiring a different type of hardware for highest performance (CPUs, GPUs, large amounts of memory, and so on). When the needed combination of hardware is not offered in a single machine, the combined use of multiple, distributed resources may be beneficial.
- The development of large simulation codes often does not align well with the latest advances in hardware. Adapting such codes to a new hardware platform 'in one go' is often far too complex and time-consuming. Having the possibility to execute such large codes on a combination of hardware may make the transition manageable, as only part of the software needs to be adapted and optimized in each step.
- Some application components may require large I/O transfers, for example, to access data generated by distant scientific instruments. It is often beneficial to avoid long-distance pre-staging and post-staging of data, and to perform part of the calculations close to where the (distributed) data resides.

Part of the rationale underlying our EYR-global work is to research and provide further insights into the capabilities and potential of large-scale distributed cyberinfrastructure. With the arrival of high-capacity lightpath technologies, many advances in distributed computing are made possible, in particular, also for application types that are traditionally not expected to perform well in a distributed setting. CESM is an example of such an application and will be described later.

DOI: 10.1002/cpe

3.2. The community earth system model

Community earth system model is a multi-model/multi-kernel simulation system consisting of five components: (1) atmosphere, (2) ocean, (3) land, (4) sea-ice, and (5) a central coupler (Figure 2). The coupler is responsible for complex data conversions and transfer between the models. Traditionally, CESM is run on a single supercomputer, typically on thousands of cores, with reports of up to 200,000 cores [9].

The processing time required per model depends on the resolution and on the complexity of each model. For typical configurations, the ocean model requires by far the most processing time, followed by the atmosphere and sea-ice models. The land model only requires a small amount of processing due to the lower required complexity. Typically, data are exchanged between land, sea-ice, and atmosphere models every 30 simulated minutes, resulting in 48 couplings per simulated day. The ocean model only exchanges data with the other models once every simulated day. Because of data dependencies, the atmosphere model cannot run concurrently with the land and sea-ice models.

The workflow for CESM is shown in Figure 3. A model day starts by a global coupling involving all cores. Next, the ocean and atmosphere models start their computation. While the ocean model continues for the remainder of the model day, the atmosphere model stops after 30 simulated minutes. The sea-ice and land models are then run concurrently, using the same set of cores as the atmosphere model. After 30 simulated minutes, the coupler is run to exchange data between the atmosphere, sea-ice, and land models. This is repeated 48 times, followed by a global coupling. The amount of data transferred between the models depends *only* on their resolution. Changing the number of cores assigned to each model will only change the number of messages exchanged, not the total data volume. Data exchange between the models is performed using message passing interface (MPI).

3.3. Execution scenarios

Increasing the resolution from the state-of-the-art 0.1 degree (10 km) ocean and 0.5 degree atmosphere towards 0.02 degree (2 km) ocean and 0.5 degree atmosphere would require 25×–100× more processing time for the ocean model. To realize this, we proposed a combination of solutions:

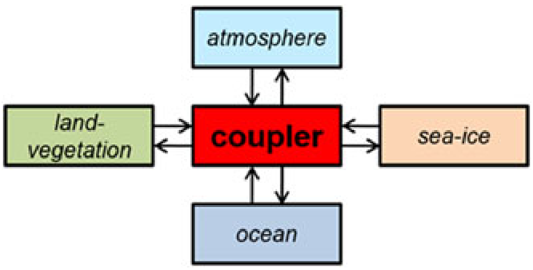

Figure 2. High-level overview of community earth system model.

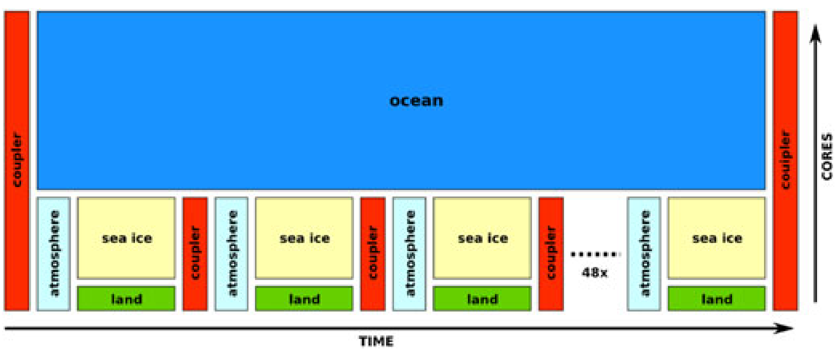

Figure 3. Community earth system model: a single simulated day (left to right) for all cores (bottom to top).

DOI: 10.1002/cpe

- **Distribute CESM over two supercomputers**. Because the ocean model is relatively loosely coupled to the other models, it can be run on a separate machine, provided the communication between the machines is fast enough for the data exchanges taking place once per model day.
- **Distribute the ocean model over multiple supercomputers**. Internally, the ocean model partitions the work into blocks which are distributed over the available cores. During the simulation, data must be exchanged between neighboring blocks, resulting in much communication. However, no exchange is needed with neighboring blocks that only contain land. Hence, by taking geography into account, the reduced communication between specific blocks can be exploited to distribute the model over multiple supercomputers [15].
- **Improve ocean model performance using hardware accelerators**. Many state-of-the-art supercomputers use accelerators such as GPUs or Xeon Phi to significantly improve their computational performance. By converting (part of) the ocean model implementation to use such accelerators, its performance can be increased significantly.

Given the proposed solutions, we defined four different execution scenarios (Figure 4):

- **Scenario 1**: A distributed simulation using Cartesius and SuperMUC, one running the ocean model, and one running the atmosphere, land and sea-ice models.
- **Scenario 2**: A distributed simulation using SuperMUC and Stampede, with the same work distribution as in Scenario 1—to show that our approach also works on a transatlantic scale.
- **Scenario 3**: A distributed simulation using Cartesius and SuperMUC to run half of the ocean model each, and Stampede to run the remaining three models.
- **Scenario 4**: A distributed simulation using Emerald to run a GPU accelerated version of the ocean model, and Cartesius to run the remaining three models.

3.4. The impact of wide-area data transfers on execution time

Figures 5 and 6 show estimates for the wide-area data transfers required per model day in each of the four execution scenarios. For Scenarios 1, 2, and 4, the amount of transferred data is identical. The difference lies in the geographical location of the machines (in Europe for Scenarios 1 and 4,

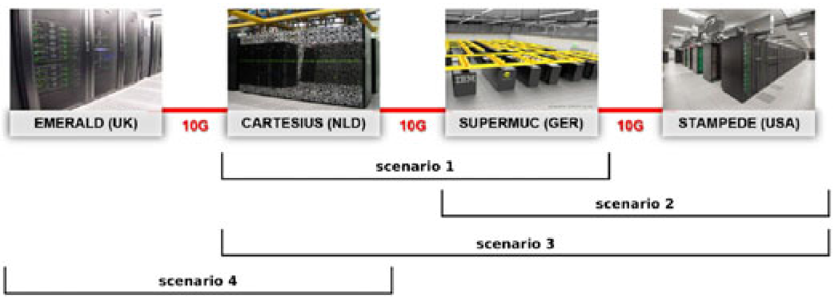

Figure 4. Supercomputer utilization in each of the defined execution scenarios.

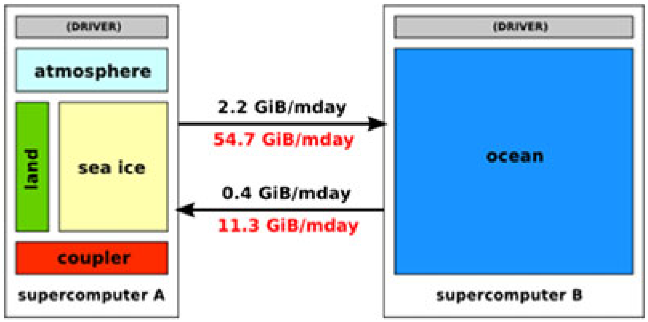

Figure 5. Estimated data transfers for Scenarios 1, 2, and 4; 0.1 degree ocean in black, 0.02 degree in red.

DOI: 10.1002/cpe

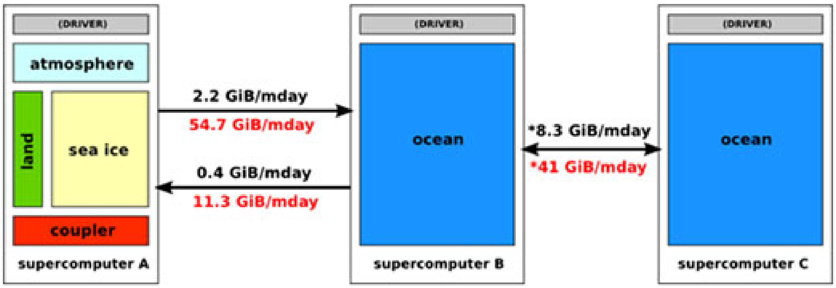

Figure 6. Estimated data transfers for Scenario 3; 0.1 degree ocean in black, 0.02 degree in red.

and transatlantic in Scenario 2), and the use of GPUs (only in Scenario 4). In Scenarios 1, 2, and 4, the wide-area communication is bursty. No wide-area communication is performed during the simulation of a single day. Once a simulated day is completed, however, data must be exchange between the models as fast as possible to minimize the communication overhead.

In Scenario 3 (Figure 6), there is additional wide-area communication taking place within the ocean model. This communication is not bursty; it continues during the full time required to simulate a single day.

To estimate the required network capacity, an estimate of the compute time per model day is also needed. For runs using 0.1 degree ocean and 0.5 degree atmosphere, Dennis *et al* [27] reported a performance of 3.23 model years/24 h on 19,812 cores of the Kraken supercomputer [28], or 73.3 s per model day (not taking into account more recent optimizations) [29]. We obtained a performance of 6.68 model years/24 h on 4176 cores of Cartesius for an ocean-only run, or 35.4 s per model day. Assuming that the relative performance of all models to be similar on Kraken and Cartesius, we estimate that a full CESM simulation on Cartesius would require approximately 6050 cores for a performance of 6 model years/24 h. With the per-core performance of Cartesius much higher than that of the older (and now decommissioned) Kraken, we used Cartesius as a reference for our network requirement estimates.

For 0.1 degree ocean simulations, in case of Scenarios 1, 2, and 4, the total amount of data transferred between the models per model day is 2.6 GiB (Figure 5), which equals 22.3 Gbit per model day. Assuming the data transfers cannot be performed concurrently (which is overly pessimistic), and also assuming that the link can reach 100% utilization (which is overly optimistic), the estimated time to transfer this data will be 22.3 s for a 1 Gbit/s lightpath, and 2.2 s for a 10 Gbit/s lightpath. Because the compute time on a Cartesius class machine is 35.4 s per model day on 6K cores, the percentage of time spent in communication will be at least

- $100 * (22.3/(22.3 + 35.4)) = 38.6\%$ (for a 1 G lightpath)
- $100 * (2.2/(2.2 + 35.4)) = 5.8\%$ (for a 10 G lightpath)

Similarly, the estimated runtime for one model year will be the following:

- $(22.3 + 35.4) * 365 = 5.9$ h (for a 1 G lightpath)
- $(2.2 + 35.4) * 365 = 3.8$ h (for a 10 G lightpath)

These numbers clearly indicate that for a 1 G lightpath the communication overhead is unacceptably high. For a 10 G link, however, the communication costs are much more acceptable.

Table I shows the estimated data transfer costs, and the percentage of communication as part of the full simulation, for all scenarios. All estimations are obtained as previously. As the computation cost for the 0.02 degree ocean runs was estimated to be between 25 times and 100 times higher than for the 0.1 degree runs, Table I gives estimations for these lower and upper bounds. For example, the lower bound estimation for the 0.02 degree runs in Scenarios 1, 2, and 4 indicates that 39.0% of the total simulation is spent in communication in case of a 1 G lightpath. The upper bound estimation for the same situation indicates a 13.8% communication overhead. These numbers drop significantly to 6.0% and 1.6% when using a 10 G lightpath. Similar numbers are provided for Scenario 3.

DOI: 10.1002/cpe

Table I. Data transfer estimations, and percentage of total simulation time, in case of 1 and 10 G networks.

Scenario/resolution	Data transfer (Gbit/mday)	Transfer time (s)		Part of total (%)		Runtime (h)	
		1 G	10 G	1 G	10 G	1 G	10 G
1, 2, 4 (0.1)	22.3	22.3	2.2	38.6	5.8	5.9	3.8
3 (0.1)	71.3	71.3	7.1	66.8	16.7	13.1	4.5
1, 2, 4 (0.02; 25×)	566.9	566.9	56.7	39.0	6.0	147.2	95.5
1, 2, 4 (0.02; 100×)	"	"	"	13.8	1.6	416.4	364.7
3 (0.02; 25×)	352.2	352.2	35.2	28.5	9.0	182.9	99.0
3 (0.02; 100×)	"	"	"	9.0	1.0	452.1	368.2

The estimates for the 0.02 degree runs all assume a fixed number of cores. With a 100× increase in processing time, however, the speed of the simulation drops from approximately 6.7 to 24 model days/24 h. Therefore, it makes sense to increase the number cores. Reducing the processing time in this manner, however, increases the communication overhead percentage. Finding the optimal balance between simulation speed and communication overhead is therefore also an important open research question of the project.

3.5. Requested network infrastructure

As indicated in Figure 4, three point-to-point connections between the available supercomputers are needed. As the communication in Scenarios 1, 2, and 4 is performed in bursts, it is important to have sufficient bandwidth available, with limited or no interference from other traffic. Scenario 3 is more tightly coupled, requiring low latency and limited jitter. For these reasons, we requested the EYR-global organizers to provide us with three 10 G lightpaths to support us in the project.

At the start of the project, it was decided that for the connection between Cartesius and SuperMUC a 10 G PRACE link would be used. The transatlantic link between Stampede and SuperMUC was replaced by a link between Stampede and Cartesius instead, as this was easier to realize and more economically viable. The only consequence for the project was that Cartesius would be used for the simulations in Scenario 2, instead of SuperMUC. The connection between Cartesius and Emerald would be set up when significant progress would have been realized for Scenarios 1, 2, and 4.

4. TECHNICAL AND NON-TECHNICAL COMPLEXITIES

Although we have made significant progress on all components of the project individually, we only have been successful in executing Scenario 1—and only at the lower resolution of 0.1 degree ocean. In the following analysis, for each of the major technical and non-technical complexities encountered during the project, it is indicated whether the problem is *solved*, *pending* (but non-critical), or *critical*—that is, representing an insurmountable problem that directly contributed to the failure of (part of) the project. As no critical technical complexities have been encountered, there were no fundamental barriers that inherently prevented the project from being successful.

Data: bathymetry and forcing files (technical-pending)

To run CESM at extremely high resolution, new input data files must be generated for each of the models, such as global bathymetry descriptions and mapping weights needed to convert fields between the various grids used by the different models. Creating these files is not a straightforward task, as it requires manually adapting a collection of shell scripts, fortran and C programs, Makefiles and IDL scripts to the new resolution, a difficult and error-prone task. Given the lack of progress on some of the other components of the project, we have not created the necessary input files yet.

Software: wide-area MPI (technical-solved)

Community earth system model is a complex application containing approximately 1 million lines of Fortran-MPI code. To communicate between distributed machines, a special MPI implementation is needed. Supercomputers typically use very fast local interconnects (e.g., Infiniband) for MPI

DOI: 10.1002/cpe

communication between the compute nodes, and a more regular IP-based network for maintenance. As a matter of policy, these networks are typically not set up to allow communication with the outside world. This is seen as security risk, also because only a handful of compute nodes could easily swamp the external network connection of the compute center. Typically, only special frontend nodes, heavily protected by firewalls, are allowed to communicate with the outside world. Several wide-area MPI implementations developed during the Grid era (e.g., MPICH-G2 [30], gridmpi [31], MagPIe [32]), where designed to run in such environments. Unfortunately, they have not been actively maintained for the last 5 to 10 years. Worse still, for many of these projects, the software has disappeared over time, and all that remains are the papers. We therefore created our own wide-area MPI implementation, using wrapper code that intercepts MPI calls and decides whether a call should be forwarded to the local MPI implementation or be sent to an external process. Such support processes or 'hubs' run on the frontend nodes and serve as a gateways to the outside world. Our MPI wrapper is made available under a free open source license [33].

Software: CESM/POP domain decomposition (technical-solved)

To let the ocean model (parallel ocean program, POP) execute efficiently in a distributed fashion, wide-area communication must be minimized. POP initially supported several algorithms for distributing the ocean grid over the available MPI tasks, but none of these take into account the hierarchical nature of modern computing hardware. To overcome this problem, we have implemented a *hierarchical block partitioning* approach that minimizes communication between the different levels in the hardware hierarchy. The algorithm takes the geographical features of the planet into account, in particular, the fact that communication with *land* blocks is not needed. Our approach is described extensively in [15] and was shown to outperform even the available strategies in traditional (non-distributed) runs.

Software: CESM/POP GPU extensions (technical-solved-partially pending)

To increase the performance of POP, we can offload the most compute-intensive parts to a GPU. Apart from the lack of real hotspots in POP, the main challenge is to overcome the PCIe bus bottleneck, by maximizing the overlap of CPU-GPU data transfers with computation. Our solutions to this problem are described in [15, 16, 34], and our GPU-enabled POP implementation is made available under a free open source license [35]. This is a stand-alone version of the ocean model, however, not integrated in CESM. Although the integration is straightforward, due to a lack of progress on the other project components, we have not (yet) done so. Also, a GPU-port of POP as a whole is expected to increase performance even further; this is also deferred as future work.

Middleware: advance reservation (technical-pending)

Some of the organizations affiliated with EYR-global became so interested in the project that we were invited to submit a PRACE/XSEDE proposal for further technical and implementation support (in terms of manpower). As part of the proposal, we requested support in the setup of an advance reservation system that would significantly ease the problem of concurrent use of compute resources at the different sites. Although the jury was sympathetic to our proposal, it was rejected primarily because it is very hard to unify advance reservation with the need for optimal resource utilization of large-scale production systems. The main issue here is that at start of time a reservation the required number of machines must be available. Enough machines must therefore become (or remain) idle in the time leading up to this start time, resulting in a decrease of utilization. In the current scheduling systems, these wasted CPU cycles cannot directly be charged to the reservation, because it has not started yet. While it is certainly possible to run our execution scenarios without an advance reservation system in place, it is a significant complicating factor.

Hardware (Network): the last mile (technical-solved-partially pending)

Creating the long-distance lightpaths from Cartesius to SuperMUC and Stampede was easy from a user perspective, although the NRENs did much work behind the scenes. The proverbial *last mile* was a problem, however. The lightpath had to be connected to the gateway node responsible for data transfer from one supercomputer's internal network over the lightpath to another supercomputer's internal network. Realizing this setup required much time and convincing explanations at some sites, as it was considered a potentially hazardous back-door into the system. Other sites were less reluctant. Part of the problem was clearly non-technical: progress depended on the specific setup and usage policies at each site.

DOI: 10.1002/cpe

Hardware (Network): network configuration I (technical-solved)

At TACC (Stampede), an issue was that private IP addresses had to be used by the gateway node for external communication. This conflicted with the private IP range in use internally at SURFsara (Cartesius). The solution was to use a SURFsara public IP address for the gateway node at TACC. This clearly illustrates that local IP assignment policies can conflict when two sites are connected using a wide area link.

Hardware (Network): network configuration II (technical-solved)

The network setup of the gateway nodes had to be optimized for lightpath usage. Theoretically, jumbo frames should be used and 'bandwidth*delay'-sized TCP buffers. The gateways at Stampede and SuperMUC, however, were configured for standard frames and buffer sizes. Changing to jumbo frames did work (initially, see in the succeeding text) for the transatlantic link. For the link to SuperMUC, this did not work. Although the exact cause has not been identified, a likely reason is that one of the intermediate hops in the wide area network does not properly handle jumbo frames. It was therefore decided to use standard frames and multiple TCP-streams instead.

Hardware (Network): a freak accident (non-technical-solved)

Once the transatlantic link between Cartesius and Stampede was correctly set up, a number of simple benchmarks were run, showing excellent performance. Before, any real CESM simulations could be started, however, the link was lost. It turned out that there was a physical break in the optical cable (quote: '*107km from Zandvoort towards Lowestoft*'), most likely because of a fishing boat hitting the cable with its anchor or nets. A ship was sent from France to repair the cable. While this problem is designated here as non-critical, it had a significant negative effect on the progress of the project.

Hardware (Gateway): disappearing and upgraded hardware (non-technical-critical)

While the transatlantic link was being repaired, the gateway node at Stampede, required for wide-area communication with Cartesius, was reclaimed for other work. We were unable to obtain a replacement and were offered a virtual machine instead, which caused a new series of configuration problems that took a long time to resolve (network configuration, jumbo frames, vlan settings). Almost immediately after having resolved these issues, however, our transatlantic link (ANA-100G) was replaced by ANA-200G [36]. After resetting the virtual machine, the network configuration problems returned. These problems illustrate that correctly configuring the gateway machines is difficult, and that this configuration is often *fragile* and prone to break when changes are made in the underlying hardware or network infrastructure. These problems are currently not yet resolved because of a lack of time on all sides (see the succeeding text).

Time: a scarce resource (non-technical–critical)

Although all people involved have been very helpful, a persistent problem was a lack of available time. It was difficult for the local system and network administrators to spend much time on the project. The time spent on setting up and debugging the network had to be accounted for somehow. Unfortunately, their specific help and expertise was part of the critical path of the project. Section 6 briefly indicates the experiments we have been able to perform, despite all of the aforementioned complexities.

5. RELATED WORK

Several related efforts towards using large-scale distributed cyberinfrastructure have been described in the literature, also including lightpath connectivity. A prominent example is CosmoGrid, aimed at distributed astrophysics simulations [37]. In this work, wide-area communication mainly consisted of bulk transfers between a single designated node at each supercomputing site, which required the simulation software itself to be adapted accordingly. In contrast, CESM causes millions of small messages to be sent between all of the nodes at the different sites. Because of our wide-area MPI implementation, no adaptations to the simulation code itself had to be made. Otherwise, the CosmoGrid project suffered from many complexities similar to those encountered in our work [38].

Further efforts are exemplified by the StarPlane project [39], which aimed to let grid applications choose a logical network topology in lightpath-connected distributed cyberinfrastructure. Focusing on a variety of application domains (a.o. model checking, digital cinema), the project differs from

DOI: 10.1002/cpe

our EYR-global efforts in that the StarPlane lightpaths were part of a relatively homogeneous, and centrally maintained distributed computing system (distributed ASCI supercomputer, or DAS).

Members of our team also have ample experience in successful deployment of scientific applications on a wide variety of hardware, even on a worldwide scale. Examples include real-time and off-line applications in the domains of distributed multimedia content analysis [40] and computational astrophysics [8, 41]. These examples have, amongst other, been demonstrated live at previous ACM/IEEE supercomputing conferences.

6. CONCLUSIONS AND RECOMMENDATIONS

In this paper, we presented our EYR-global work and the complexities we encountered during this project. Although a number of complexities have been found *critical*, the project has not been entirely unsuccessful. Most importantly, we have been able to overcome all software implementation complexities, in particular, with regard to the distributed execution, the associated load-balancing, and the efficient GPU execution. In practice, we are now capable of running distributed runs using Cartesius and SuperMUC on a regular basis, employing over 1500 cores in total, and using the 10 G PRACE link for wide area communication—thus representing the first of our four scenarios (although currently using a resolution of 0.1 degree ocean only). The obtained results follow our estimations as described in Section 3.4 and are currently prepared for a separate publication.

Nevertheless, we were not able to use the distributed cyberinfrastructure available to us in its entirety. Consequently, we have not yet been able to perform experiments at the extreme resolution of 0.02 degree ocean. The critical complexities can be attributed mostly to hardware problems and a lack of time. Because of this, we formulate the following set of general recommendations:

(1) **Improved support for wide area network connectivity:** Connecting the lightpaths to the supercomputers was harder than expected, because this is not a ‘regular’ use case for such machines. Setting up these connections required both convincing explanations and several different ad-hoc technical solutions. To prevent this in the future, we recommend that a standard setup is chosen to support distributed applications, similar to the solutions that already exist for bulk data transfers. In addition, standard libraries (such as MPI) that support these wide-area links should be provided.

(2) **Include technical support as part of the project:** In EYR-global, the wide-area network connectivity was provided by the NRENs, and the access to the different supercomputers was arranged by the project partners. Although some local system and network specialists were part of the team, there was not dedicated budget set aside for them. This made it hard for them to provide the necessary level of support, as their time had to be accounted for somehow. We recommend that a budget for support should explicitly be a part of project proposals such as ours.

(3) **Support advance reservation:** The availability of a advance reservation systems would ease the use of distributed supercomputer systems significantly. The effect on overall system utilization may be limited when advance reservation is applied correctly [42, 43].

(4) **Design for heterogeneous and distributed computing:** Although there has been a large amount of research on heterogeneous and distributed computing, very little of this work has been applied to production applications. Such applications are often still designed in a traditional C/Fortran-MPI style, even though this is not necessarily efficient on modern multi-island, multi-core, accelerator-based supercomputers. We recommend that developers start moving towards application designs that take both local heterogeneity and distribution into account. This is difficult, especially with large legacy applications, but necessary to fully utilize modern supercomputers.

We firmly believe that the aforementioned recommendations will allow future attempts in utilizing large-scale lightpath-connected distributed cyberinfrastructure to be more successful than ours have been.

DOI: 10.1002/cpe

ACKNOWLEDGEMENTS

We thank Patrick Storm, Nathaniel Mendoza (TACC), and Reinhold Bader (LRZ) for all their support.

REFERENCES

1. *The Research in Advanced Distributed Cyberinfrastructure and Applications Laboratory (RADICAL)*. (Available from: http://radical.rutgers.edu/) [Accessed on 28 April 2016].
2. Merzky A, Weidner O, Jha S. SAGA: a standardized access layer to heterogeneous distributed computing infrastructure. *SoftwareX* 2015; **1-2**:3–8.
3. Seinstra FJ, Maassen J, Van Nieuwpoort RV, Drost N, Van Kessel T, Van Werkhoven B, Urbani J, Jacobs C, Kielmann T, Bal HE. Jungle computing: Distributed supercomputing beyond clusters, grids, and clouds. In *Grids, Clouds and Virtualization*. Springer: London, 2011; 167–197.
4. Sutter H. *Welcome to the jungle*, 2012. (Available from: http://herbsutter.com/welcome-to-the-jungle) [Accessed on 28 April 2016].
5. Bal HE, Maassen J, van Nieuwpoort RV, Drost N, Kemp R, Palmer N, Wrzesinska G, Kielmann T, Seinstra F, Jacobs C. Real-world distributed computing with Ibis. *IEEE Computer* 2010; **43**(8):54–62.
6. Luckow A, Santcroos M, Merzky A, Weidner O, Mantha P, Jha S. P*: a model of pilot-abstractions. *eScience 2012*, Chicago, USA, October 2012; 1–10.
7. Bosilca G, Bouteiller A, Danalis A, Faverge M, Herault T, Dongarra JJ. PaRSEC: exploiting heterogeneity to enhance scalability. *Computing in Science & Engineering* 2013; **15**(6):36–45.
8. Drost N, Maassen J, Van Meersbergen MA, Bal HE, Pelupessy F, Zwart SP, Kliphuis M, Dijkstra HA, Seinstra FJ. High-performance distributed multi-model/multi-kernel simulations: a case-study in jungle computing. *HCW 2012*, Shanghai, China, May 2012; 150–162.
9. Worley PH, Craig AP, Dennis JM, Mirin AA, Taylor MA, Vertenstein M. Performance of the community earth system model. In *International Conference for High Performance Computing, Networking, Storage and Analysis (SC'11),* Seattle, USA, November 2011; 1–11.
10. Vellinga M, Wood RA. Impacts of thermohaline circulation shutdown in the twenty-first century. *Climate Change* 2008; **91**(1):43–63.
11. Bryan F, Thomas R, Dennis JM, Chelton DB, Loeb NG, McClean JL. Frontal scale air-sea interaction in high-resolution coupled climate models. *Journal of Climate* December 2010; **23**(23):6277–6291.
12. Kirtman B, Blitz C, Bryan F, Collins W, Dennis J, Hearn N, III JK, Loft R, Rousset C, Siqueira L, Stan C, Tomas R, Vertenstein M. Impact of ocean model resolution on CCSM climate simulations. *Climate Dynamics* 2012; **39**(6):1303–1328.
13. Weijer W, Maltrud ME, Hecht MW, Dijkstra HA, Kliphuis MA. Response of the atlantic ocean circulation to greenland ice sheet melting in a strongly-eddying ocean model. *Geophysical Research Letters* 2012; **39**(9):L09606.
14. Brunnabend SE, Dijkstra HA, Kliphuis MA, van Werkhoven B, Bal HE, Seinstra F, Maassen J, van Meersbergen M. Changes in extreme regional sea surface height due to an abrupt weakening of the atlantic meridional overturning circulation. *Ocean Science* 2014; **10**:881–891.
15. van Werkhoven B, Maassen J, Kliphuis M, Dijkstra HA, Brunnabend SE, van Meersbergen M, Seinstra FJ, Bal HE. A distributed computing approach to improve the performance of the parallel ocean program (v2.1). *Geoscientific Model Development* February 2014; **7**:267–281.
16. van Werkhoven B, Maassen J, Seinstra FJ, Bal HE. Performance models for CPU-GPU data transfers. In *14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2014),* Chicago, USA, May 2014; 11–20.
17. Enlighten Your Research. (Available from: http://www.enlightenyourresearch.net) [Accessed on 28 April 2016].
18. SURFnet (NL). (Available from: https://www.surf.nl/en/about-surf/subsidiaries/surfnet) [Accessed on 28 April 2016].
19. Funet (FI). (Available from: https://www.csc.fi/funet-network-services) [Accessed on 28 April 2016].
20. Janet (UK). (Available from: https://www.jisc.ac.uk/janet) [Accessed on 28 April 2016].
21. ESnet (USA). (Available from: http://www.es.net) [Accessed on 28 April 2016].
22. Internet2 (USA). (Available from: http://www.internet2.edu) [Accessed on 28 April 2016].
23. Stampede. (Available from: https://www.tacc.utexas.edu/systems/stampede) [Accessed on 28 April 2016].
24. SuperMUC. (Available from: https://www.lrz.de/services/compute/supermuc/) [Accessed on 28 April 2016].
25. Cartesius. (Available from: https://userinfo.surfsara.nl/systems/cartesius) [Accessed on 28 April 2016].
26. Emerald. (Available from: http://www.ses.ac.uk/high-performance-computing/emerald/) [Accessed on 28 April 2016].
27. Dennis JM, Vertenstein M, Worley PH, Mirin AA, Craig AP, Jacob R, Mickelson S. Computational performance of ultra-high-resolution capability in the community earth system model. *International Journal of High Performance Computing Applications* 2012; **26**(1):5–16.
28. Kraken. (Available from: https://www.nics.tennessee.edu/kraken/specifications) [Accessed on 28 April 2016].
29. Hu Y, Huang X, Baker AH, Tseng Y, Bryan FO, Dennis JM, Yang G. Improving the scalability of the ocean barotropic solver in the community earth system model. *SC'15*, Austin, TX, USA, November 2015.

30. Karonis NT, Toonen B, Foster I. MPICH-G2: A grid-enabled implementation of the message passing interface. *Journal of Parallel and Distributed Computing* 2003; **63**(5): 551–563.
31. GridMPI. (Available from: http://aist-itri.github.io/gridmpi/) [Accessed on 28 April 2016].
32. Kielmann T, Hofman RFH, Bal HE, Plaat A, Bhoedjang RAF. MagPIe: MPI's collective communication operations for clustered wide area systems. *PPoPP'99*, Atlanta, GA, USA, May 1999; 131–140.
33. eSALSA MPI. (Available from: https://github.com/NLeSC/eSalsa-MPI) [Accessed on 28 April 2016].
34. van Werkhoven B, Hijma P. An integrated approach to porting large scientific applications to GPUs. *eScience 2015*, Munich, Germany, August 2015; 57–66.
35. eSALSA POP. (Available from: https://github.com/NLeSC/eSalsa-POP/tree/gpu) [Accessed on 28 April 2016].
36. ANA-200G. (Available from: https://www.nordu.net/tags/ana-200g) [Accessed on 28 April 2016].
37. Portegies Zwart S, Ishiyama T, Groen D, Nitadori K, Makino J, de Laat C, McMillan S, Hiraki K, Harfst S, Grosso Paola. Simulating the universe on an intercontinental grid of supercomputers. *IEEE Computer* August 2010; **43**(8):63–70.
38. Groen D, Portegies Zwart S. From thread to transcontinental computer: disturbing lessons in distributed computing. *eScience 2015 (ERROR workshop)*, Munich, Germany, August 2015; 565–571.
39. Grosso P, Marchal D, Maassen J, Bernier E, Xu L, de Laat C. Dynamic photonic lightpaths in the starplane network. *Future Generation Computer Systems* 2009; **25**(2):132–136.
40. Seinstra FJ, Geusebroek JM, Koelma D, Snoek CGM, Worring M, Semulders AWM. High-performance distributed video content analysis with parallel-horus. *IEEE Multimedia* 2007; **14**(4):64–75.
41. Pelupessy FI, van Elteren A, de Vries N, McMillan SLW, Drost N, Zwart SP. The atrophysical multipurpose software environment. *Astronomy & Astrophysics* 2013; **557**(A84).
42. Snell Q, Clement M, Jackson D, Gregory C. The performance impact of advance reservation meta-scheduling. *JSSPP 2000, LNCS 1911*, Cancun, Mexico, May 2000; 137–153.
43. Margo MW, Yoshimoto K, Kovatch P, Andrews P. Impact of reservations on production job scheduling. *JSSPP 2007, LNCS 4942*, Seattle, WA, USA, June 2007; 116–131.

DOI: 10.1002/cpe