A Massively Parallel Dynamical Core for Continental- to Global-Scale River Transport

[1,2,3]J. W. Larson, [4]A. P. Craig, [5] J. B. Drake, [5]D. J. Erickson, III, [5]M. L. Branstetter, [5]M. W. Ham

[1]ANU Supercomputer Facility, The Australian National University
Canberra ACT 0200, Australia
E-Mail: `Jay.Larson@anu.edu.au`
[2]Mathematics and Computer Science Division, Argonne National Laboratory
9700 S. Cass Avenue, Argonne, IL 60439, USA
[3] Computation Institute, University of Chicago
Chicago, IL, USA
[4] Climate and Global Dynamics Division, National Center for Atmospheric Research
Boulder, CO, USA
[5] Computer Science and Mathematics Division, Oak Ridge National Laboratory
Oak Ridge, TN, USA

Keywords: *Runoff and River Modelling; Climate Modelling; Software Architecture; Parallel Computing*

The practise of climate modeling has evolved from study of individual subsystems to integrated *coupled* climate or *earth system* models. Coupled climate models comprise general circulation models (GCMs) for both the atmosphere and ocean, a land-surface model, and a sea-ice model. Rivers play an important role in the Earth's hydrological cycle (Figure 1), and most climate system models now include continental-scale river transport models (RTMs) to complete the global water balance. The RTM takes as its input runoff calculated by the land-surface model, routes it through the river network and ultimately into the system's ocean component.

Many continental-to-global scale river transport models (RTM's) exist, but none are currently simultaneously able to achieve high performance and to advect tracers. We are developing a massively parallel dynamical core (dycore) for river transport modeling at the continental-to-global scale. This approach treats the world's river networks as a directed graph $\mathcal{G}$, whose vertices represent the centroids of grid cells or catchments, and whose edges represent the surface flow paths or river reaches between catchments. For RTMs using a linear reservior assumption, transport becomes a linear transformation, with the transport matrix $\mathbf{T}$ having identical structure to the adjacency matrix $\mathbf{A}$ of $\mathcal{G}$.

We describe a programming approach to the RTM linear dycore that leverages classes and methods from the Model Coupling Toolkit (MCT). MCT is a parallel computing toolkit for building message-passing parallel coupled models from message-passing-parallel codes. We provide a brief overview of MCT, with emphasis on the classes and methods necessary to our dycore design. Use of MCT yields a highly scalable RTM dycore, and offers two immediate advantages: 1) support for tracer transport with virtually no additional programming effort; and 2) coupling the RTM to other models is straightforward, as MCT's underlying coupling infrastructure is part-and-parcel of the resulting model. Furthermore, MCT's global sum methods can be used to diagnose and enforce both water and tracer mass conservation.

Some of these ideas have been prototyped in the RTM from the Community Climate System Model 3.0 and are currently undergoing testing. The resulting parallel RTM improves significantly the parallel scalability of this system's land-surface scheme through faster surface runoff transport and parallelisation of river-land coupling.

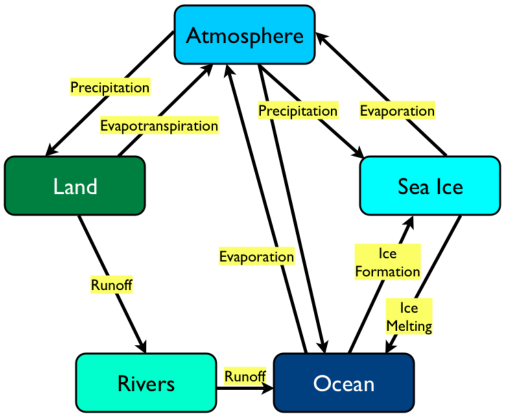

Figure 1. Hydrological cycle as represented in most earth system models.

1 RIVER TRANSPORT MODELS

The world's rivers are a crucial link in the global hydrological cycle (Figure 1) because they route fresh water fluxes from land deep within continental masses into the earth's oceans, and the distribution of these fresh water fluxes affects coastal flow and the global thermohaline circulation. From a climate modelling perspective, the river link is also important because runoff into rivers is an integrated product of atmosphere and land processes that can be validated directly with hydrograph data.

The current state of the art in climate modelling is the *coupled climate model* comprising atmosphere and ocean general circulation models (GCMs), sophisticated vegetation-atmosphere transfer schemes (SVATS) for land-surface energy, momentum, and moisture exchange, and dynamic-thermodynamic sea ice models. Each subsystem component has a high degree of computational complexity that is conquered using parallel computing in the form of 1) distributed-memory parallelism using the message-passing interface (MPI MPI Forum [1994]) standard, and/or 2) shared-memory parallelism using OpenMP (Chandra et al. [2000]). The data dependencies between the models that coupling requires constitute a *parallel coupling problem* (Larson et al. [2005]), and this functionality is often implemented in a distinct component called a *flux coupler*, or a *coupler*. River transport models (RTMs) typically reside within the land-surface component of a coupled model (Jacob et al. [2001]; Collins et al. [2006]), or as a distinct component (Washington et al. [2000]; Bettge et al. [2001]) .

Most climate modelling groups have as a near-term goal the addition of a biogeochemical cycle (a.k.a. a carbon cycle) to their models. This comprises biogeochemical modules for the land, ocean, and atmosphere, as well as enhancements to the coupling infrastructure to handle additional data traffic due to chemical species concentrations. The logical extension to RTMs to meet this goal are inclusion of 1) river chemistry, and 2) advection of trace chemicals; our primary objective in the present work is addressing the latter requirement, by which we provide a design for satisfying the former requirement. All RTMs with which the authors are familiar simply advect water, and have no mechanism for advection of chemical tracers and other dissolved matter.

We believe a next-generation RTM should include the following features

1. A separation of concerns between surface mesh generation and river routing. Mesh and/or river network generation from digital elevation models should be accomplished using a method such as that of Graham et al. [1999] or others, and be defined as input data that can be selected by the RTM at run-time.
2. A scalable and performance-portable algorithm that will allow use of very high-resolution river networks, and capable of producing more detailed (and with hope more accurate!) river flow patterns, and thus higher-quality freshwater inputs to the oceans.
3. The ability to transport dissolved matter such as chemical tracers and sediments for biogeochemical cycle and water quality modelling. The system should be sufficiently flexible that tracers can be added or removed without undue programming effort, or possibly can be set at run-time.
4. A sophisticated, extensible framework that will allow the inclusion of the effects of storages such as lakes and reservoirs
5. The ability to simulate floods.
6. A framework that can be used to model lateral groundwater flow into and out of rivers and lakes.

The present work addresses requirements 2) and 3). We believe requirement 1) is met through the use of third-party software and datasets. Requirements 4)-6) are beyond the scope of the present work, but we believe our work will provide a firm foundation upon which these requirements may be met in future work.

RTMs can be classified as watershed-based, cell-to-cell, or source-to-sink (Sushama et al. [2004]). Both watershed and cell-to-cell routing begin with discretisation of the land river networks, with the former based on watersheds and the river reaches that connect them, and the latter using cells defined by an imposed mesh with flow between nearest neighbors on the mesh. Numerous cell-based models exist, including Vörösmarty et al. [1989], Miller et al. [1994], Sausen et al. [1994], Lohman et al. [1996], Coe [1998], Arora and Boer [1999], and Sushama et al. [2004].

Many RTMs present in coupled climate models use the Miller et al. [1994] algorithm. Various implementations of this scheme can be found in the Fast Ocean Atmosphere Model (FOAM; Jacob et al. [2001]), the Parallel Climate Model (PCM; Washington et al. [2000]; Bettge et al. [2001]), and the Community Climate System Model (CCSM; Collins et al. [2006]).

Miller et al. [1994] formulate linear river transport as follows: A grid cell contains a mass of water M (kg) above the sill depth and thus capable of free flow, and the flux F (kg/s) from this river cell to its downstream neighbor is

$$F = M\frac{u}{d}, \tag{1}$$

where u is an effective flow speed determined by basin morphology and topography gradient, and d is the distance between the centroids of the grid cell and its downstream neighbor. This assumption of a linear relationship between flux and storage is sometimes called a *linear reservoir*.

2 GRAPH-THEORETIC PICTURE

A river network routes water through a series of catchments from relatively high-altitude catchment areas through intermediate catchments down to river estuaries and ultimately into the world's oceans. This process can be abstracted to a *directed graph* or *digraph* (Temperly [1981]) $\mathcal{G}$, whose vertices $V(\mathcal{G})$ correspond to catchments, and whose edges $E(\mathcal{G})$ correspond to channels connecting the catchments (Figure 2). Following this analogy, any node with only outbound edges is a *source* (nodes 1-8 in Figure 2), and any node with only inbound edges is a *sink* (nodes 19-21 in Figure 2).

An *acyclic graph* is one free of loops; that is, for for every vertex i, there is no directed path leading away from i and ultimately back to it. In principle, one could construct a set of spatial cells for a river network that could have loops, but in practise this is not done. In the algorithm we are using (Miller et al. [1994])the associated graph for a river network is acyclic.

In some cases a $\mathcal{G}$ may be divided into independent subgraphs called *partite sets* or *parts*. The digraph in Figure 2 can be divided into three parts– nodes $\{$1-3,9,14,19$\}$, nodes $\{$4-7,10-12,,15,17,20$\}$ and nodes $\{$8,13,16,18,21$\}$—making it a *tripartite graph*. Any graph that can be broken into some indeterminate number of independent subgraphs is called a *multipartite graph*. The layout of the world's river network, with portions isolated from each other by either oceans or continental divdes is thus representable by a multipartite graph, and its associated parts are an underlying structure we will exploit in the domain decomposition for a parallel RTM algorithm.

The connections in a graph with N vertices may be visualised through its $N \times N$ *adjacency matrix* **A**, whose elements are A_{ij} defined as the number of edges connecting vertex i to vertex j. Some RTMs (e.g., Miller et al. [1994]) assume at most one edge connecting any two vertices, and hence the elements of **A** are either 1 or 0.

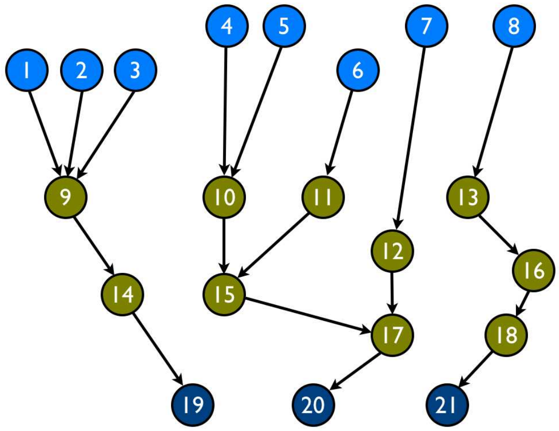

Figure 2. Directed-graph representation of a river network following assumptions of Miller et al. [1994].

The water storages at the vertices in $\mathcal{G}$ can be summarised in a *storage state vector* $\mathbf{S} \in \Re^N$. RTMs with linear transport schemes (Eq. 1) advect water mass based on the amount of water present, using constant (within the advection step) coefficients. These coefficients may be organised as a *transport matrix* **T**, whose elements T_{ij} are scaled values of the corresponding elements in **A**. The quantity $T_{ij}S_j$ is the water mass flux from vertex j to vertex i. The continuity equation for the flow requires the flux out of a cell Q_j must equal sum of the individual fluxes from a cell into its neighbors, that is

$$Q_j = T_{jj}S_j = -\sum_{i=1}^{N} T_{ij}S_j, \quad i \neq j. \tag{2}$$

Each storage will get runoff input I_j from the land-surface model. Thus, the storage at time $t+1$ is computed explicitly from its current state and input as

$$\mathbf{S}(\boldsymbol{t}+\boldsymbol{\Delta t}){=}\mathbf{S}(\boldsymbol{t}){+}\mathbf{I}{+}\mathsf{\mathbf{T}}\mathbf{S}(\boldsymbol{t}), \tag{3}$$

where $\mathbf{I} \in \Re^N$ is the vector of runoff inputs from the land model. This approach is highly configurable and can mimic any of the linear models cited above. For many models, the elements of **T** are constant, for example Vörösmarty et al. [1989], Miller et al. [1994], Sausen et al. [1994], Lohman et al. [1996], and Coe [1998]. Other linear models have time-varying elements of **T**, such as Arora and Boer [1999], and Sushama et al. [2004]. A well-designed linear RTM could be configurable at run-time to mimic the behaviour of any of the aforementioned models.

3 THE MODEL COUPLING TOOLKIT

The high level of computational complexity in the individual components of the earth system has been conquered through the use of *parallel computing*, with *distributed-memory* or *message-passing* parallelism using the Message Passing Interface (MPI; MPI Forum [1994]) library being the most-often employed aproach. Multiple subsystem models in mutual interaction lead to a *parallel coupling problem* (Larson et al. [2005]). The parallel coupling problem appears in other fields of research other than climate, and involves the description, parallel tranfer and parallel transformation of *distributed* data.

One of the authors (Larson) has co-led the development of a key piece of software infrastructure for solving the parallel coupling problem—the Model Coupling Toolkit (MCT; Larson et al. [2005]; Jacob et al. [2005]; Larson et al. [2007]). MCT provids a Fortran object-based and highly flexible programming model for constructing custom parallel coupling mechanisms. MCT's classes—our use of the term *class* in Fortran follows that of Decyk et al. [1997]—include three for data description, three for parallel data transfer, and three to implement parallel data transformation. In addition to MCT's classes and associated methods, there is a library API that manipulates these classes to perform parallel data transfer and transformation to implement parallel coupling. MCT is an open-source package and available for download via the MCT Web site. The most prominent application of MCT has been its role as the coupling middleware used in the Community Climate System Model (CCSM 3.0; Collins et al. [2006]; Craig et al. [2005]).

Below, we discuss briefly the relevant MCT classes and facilities that are leveraged to create our linear RTM dycore. A more complete description of MCT can be found in Larson et al. [2005] and Jacob et al. [2005].

The MCT programming approach requires users to describe their coupling-specific data—fields to be exchanged, the spatial meshes on which they reside, and their domain decomposition across a model's pool of processors—using MCT's data model. Physical fields are described using the AttrVect class, which is a *vector of scalar attributes* (e.g., components of the wind field, near-surface air temperature, et cetera); that is, data stored in an this class are indexed by location, and within location by attribute. This storage indexing strategy is driven by the most common coupling approach—*pointwise operations*—multivariate data processing at given points. Domain decomposition description is implemented by the GlobalSegMap class employing a strategy called *virtual linearisation* (Lee and Sussman [2005]; Bertrand et al. [2006], and references therein), whereby tuples in multidimensional index space $T \subset \mathbb{Z}^N$ are mapped into a set of integers $L \subset \mathbb{Z}$ in a one-to-one and onto fashion. The linearised index space uniquely identifies each physical location in the coupling domain. The MCT GlobalSegMap class encapsulates explicit domain decomposition of the linearised index space across processors as runs of consecutive indices. MCT also offers a class for spatial mesh description—the GeneralGrid—but it is not required for our RTM dycore.

The other MCT class of interest is the SparseMatrixPlus class, which encapsulates a parallel linear transformation cast as sparse matrix-vector multiplication. This class provides storage for nonzero transformation coefficients stored in COO format, and communications scheduling necessary to perform a parallel matrix-vector multiply $\mathbf{y} = \mathbf{B}\mathbf{x}$, where both $\mathbf{x}$ and $\mathbf{y}$ are stored in AttrVect form, making this operation a *multifield linear transformation*. This is how MCT implements intermesh interpolation of multiple fields residing on a source grid to a target grid. For sufficiently large numbers of fields, MCT's parallel sparse matrix-AttrVect multiply kernel scales well to large numbers of processors, and can achieve superlinear speedup over the initial part of the scalability curve (Larson et al. [2005]). Furthermore, this kernel is performance-portable to commodity microprocessor-based clusters and vector processor-based platforms such as NEC SX- series platforms and the Cray X-1.

4 DESIGN

The generic linear RTM dycore described in Section 1 can be implemented using MCT's data model, parallel transfer facilities, and compute kernels as follows:

1. The the river network's storage state vector $\mathbf{S}$ is instantiated in a variable of the MCT AttrVect datatype;

2. The domain decomposition is described using a variable of the GlobalSegMapdatatype;

3. The river flow is linear following the scheme of Miller et al. [1994], and thus can be cast as a sparse matrix-vector multiply. The nonzero elements of $\mathbf{T}$ stored in a SparseMatrixPlus variable. The work partition is determined by large-scale catchments (i.e., those defined by continental divides). Thus, the water-routing matrix-vector mulitplication in (3) is embarassingly parallel (i.e., no interprocessor communication is required during this operation).

4. MCT's global sum library routines can be invoked to provide with ease diagnosis of water and tracer mass conservation.

5. Coupling of the RTM and CLM land-surface models is implemented using MCT.

5 IMPLEMENTATION

Elements of the above MCT-based approach have been implemented in a modified version of the CCSM 3.0 RTM. This version of the model with the parallel RTM is still undergoing testing, but we present performance results in Section 6. The flow calculation has been parallelised using the RTM's current data structures rather than MCT's, but the performance gains will be representative of the approach outlined above. The coupling of the RTM to the land model has been parallelised using MCT classes and methods. The RTM currently uses a statically defined 0.5° latitude-longitude grid. The RTM in the publicly available version of CCSM3.0 performs the following steps: 1) perform an MPI_ALLGATHER to replicate the land model's runoff values on the land grid to all processors (note currently the land and river share a a common set of processors); 2) on all processors interpolate the runoff values onto the RTM grid; 3) on each processor, perform the full global RTM advection calculation; and 4) each processor sends a non-overlapping portion of ocean-bound runoff fluxes computed from the advection step to the coupler. Thus coupling is only partially parallelised, and the advection calculation has a high degree of redundancy.

MCT was used to parallelise steps 1) and 2) of the above algorithm–the land-to-river coupling. In the original version of the RTM, this process was more time-consuming than advection on the 0.5° grid. The advection step was parallelised as well, but not using MCT's compute kernels. Step 4) was modified so that each processor sends only the ocean-bound freshwater flux data it owns, and thus this step remained parallel.

The RTM couples with the land-surface model once every three hours, and its freshwater fluxes are accumulated over the course of one model day for input to the ocean, where these freshwater fluxes are applied incrementally. The timestep Δt in the RTM is chosen very conservatively to avoid violation of the Courant-Friedrichs-Lewy (CFL) condiiton (Courant et al. [1928]; Press et al. [1992]); that is, that a timestepping scheme cannot violate causlity. For the RTM, this condition is $|v|\Delta t/\Delta x \leq c \leq 1$, where Δx is the a characteristic distance between the centroids of two adjacent cells, v the characteristic signal velocity (for our model based on Miller et al. [1994] it corresponds to the global constant downstream flow velocity), Δt is the timestep, and c is the CFL stability parameter. The model currently operates with an arbitrarily low value of $c = 0.1$, which translates to a globally defined timestep of 30 minutes. The model is stable with this timestep. The authors believe that significantly larger timestep values might be stable, which is a topic of further investigation.

6 PERFORMANCE

We have benchmarked the prototype parallel RTM in the stand-alone version of CCSM 3.0's land component–the Community Land Model (CLM). We used CLM 3.5.07, with land mesh resolution identical to CCSM's T42 (2.8°) atmosphere, which corresponds to 3687 active grid cells. The RTM mesh was a 0.5° latitude-longitude grid, with 126631 active cells organised into 36120 river basins. The platform used for benchmarking was NCAR's bluevista cluster, which is an IBM p575 with PowerPC 5 processors running at 1.9GHz and organised into shared-memory (SM) nodes with eight processors each. Timings were performed with simultaneous multi-threading (SMT) enabled; that is, with two MPI processes per physical processor. Timings were taken over a period of thirty model days.

Timings for the RTM dycore and the land-to-river mesh transformation (L2R) are shown in Figure 3. Minimum and maximum values for timings across the whole set of processors are included to indicate load imbalance. The RTM dycore scales well from 1 to 128 processors, with the exception of poor speedup from 8 to 16 processors. The marginal speedup from 8 to 16 processors is a consequence of the the SMT configuration, as the 8 physical processors in the node are supporting 16 MPI processes. Speedup gains from 16 to 128 processors continue, but saturate slowly as load imbalances become more pronounced. The L2R transformation scales with speedup up to 8 processors, but saturates, and then degrades in performance, actually running slower on 64 and 128 processors than it does on 8. This poor scalability is a function of the relatively light workload involved in the linear interpolation embodied in L2R.

The impact of the parallel RTM in CLM is demonstrated in timings of the CLM physics driver and the wall-clock time for the CLM physics plus the RTM and L2R calculations(Figure 4). The CLM physics calculation is embarassingly parallel, and its scalability is affected only by load imbalance. The RTM calculation represents 2-5% of the total CLM physics cost at T42 resolution. The L2R transformation starts to become a relatively large cost at higher processor counts, exceeding 10% of the total CLM physics at 128 pes. Note that this is for a relatively low resolution version of CLM (there are only about 30 CLM gridcells per processor at 128 pes).

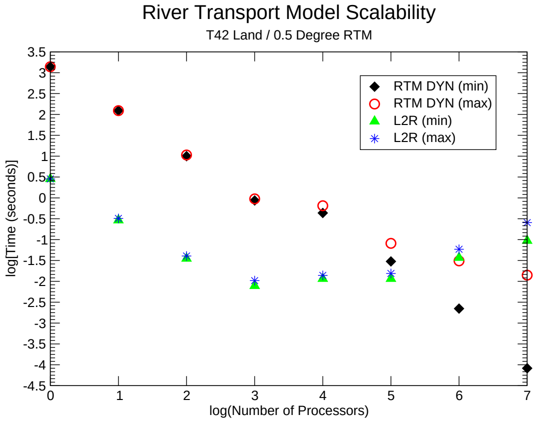

Figure 3. Parallel scalability of the RTM dycore and Land-river mesh transformation. Logarithm base is 2.

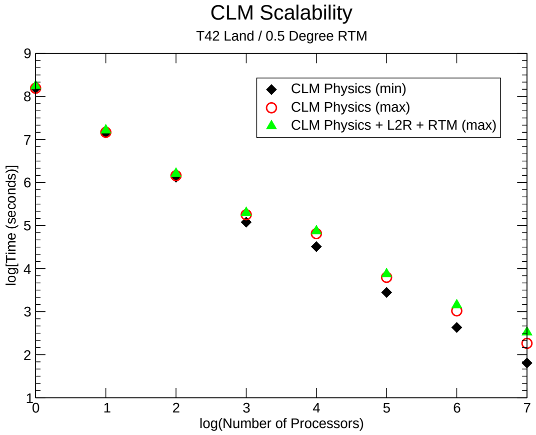

Figure 4. Parallel scalability of the CLM physics and wall-clock time for CLM incorporating the parallel RTM and MCT L2R scheme. Logarithm base is 2.

7 CONCLUSIONS AND FUTURE WORK

High-resolution of river networks is essential to the faithful simulation of the hydrological cycle in coupled climate models. The emerging need for biogeochemical cycles in earth system models creates an additional requirement for tracer transport in RTMs. These new requirements, along with the cost structure of coupling of RTMs into larger parallel coupled systems creates a genuine need for high-performance, parallel RTMs.

We have abstracted the major components of the river modelling process, and identified the RTM dycore as an algorithm in need of a flexible, high-performance software foundation. We have observed that because RTMs route water from catchment to catchment, the river network can be viewed as a directed graph $\mathcal{G}$ whose vertices are catchments and edges are the river channels. Furthermore, from our survey of the literature, we have noted that a wide class of RTMs use linear transport to advect runoff, and that this process can be cast as a sparse matrix-vector multiply, with the vector indices corresponding to catchment IDs, and the transport matrix $\mathbf{T}$ having the same sparsity and pattern of nonzero entries as the adjacency matrix $\mathbf{A}$ of $\mathcal{G}$. We have observed further that most RTMs place an additional restriction that amounts to $\mathcal{G}$ being acyclic. The multipartite nature of an RTM's associated graph provides a useful foundation for the parallelisation of its linearised dynamical core.

We have shown the efficacy of this approach through direct implementation of an MCT-based coupling algorithm in the CCSM 3.0 RTM, and parallelisation of its advection calculation.

The success of this approach leads us to believe that an RTM-application-specific toolset could be built on top of MCT technology and advance dramatically the practice of river modelling at the continental-to-global scale. Before the release of such a toolset, we will explore other issues, notably how to include storages (e.g., lakes and reservoirs) and floods into this scheme.

ACKNOWLEDGMENTS

This work has been supported by the US Department of Energy (DOE) Office of Science Climate Change Prediction Program, which is part of DOE's Scientific Discovery through Advanced Computing initiative. The CCSM development at the National Center for Atmospheric Research is supported by the US National Science Foundation. The ANU Supercomputer Facility is supported in part by the Australian Department of Education, Science, and Training through the National Collaborative Research Infrastructure Strategy. Argonne National Laboratory is operated for the DOE by UChicago Argonne, LLC, under contract number DE-AC02-06CH11357. Oak Ridge National Laboratory is operated for the DOE by UT-Batelle.

References

Arora, V. K. and G. J. Boer. A variable velocity flow routing algorithm for gcms. *Journal of Geophysical Research*, 104(D24):30965–30979, 1999.

Bertrand, F., R. Bramley, D. E. Bernholdt, J. A. Kohl, A. Sussman, J. W. Larson, and K. B. Damevski. Data redistribution and remote method invocation for coupled components. *Journal of Parallel and Distributed Computing*, 66(7):931–946, 2006.

Bettge, T., A. Craig, R. James, V. Wayland, and G. Strand. The DOE Parallel Climate Model (PCM): The Computational Highway and Backroads. In Alexandrov, V. N., Dongarra, J. J., and Tan, C. J. K., editors, *Proc. International Conference on Computational Science (ICCS) 2001*, volume 2073 of *Lecture Notes in Computer Science*, pages 148–156, Berlin, 2001. Springer-Verlag.

Chandra, R., R. Menon, L. Dagum, D. Kohr, D. Maydan, and J. McDonald. *Parallel Programming in OpenMP*. Morgan Kaufmann, New York, 2000.

Coe, M. T. A linked global model of terrestial hydrologic processes: Simulation of modern rivers, lakes, and wetlands. *Journal of Geophysical Research*, 103:8885–8899, 1998.

Collins, W. D., C. M. Bitz, M. L. Blackmon, G. B. Bonan, C. S. Bretherton, J. A. Carton, P. Chang, S. C. Doney, J. J. Hack, T. B. Henderson, J. T. Kiehl, W. G. Large, D. S. McKenna, B. D. Santer, and R. D. Smith. The Community Climate System Model: CCSM3. *Journal of Climate*, 19(11):2122–2143, 2006.

Courant, R., K. Friedrichs, and H. Lewy. Über die partiellen Differenzengleichungen der mathamatischen Physik. *Mathamatische Annelen*, 100(1):32–74, 1928.

Craig, A. P., B. Kaufmann, R. Jacob, T. Bettge, J. Larson, E. Ong, C. Ding, and H. He. cpl6: The new extensible high-performance parallel coupler for the community climate system model. *Int. J. High Perf. Comp. App.*, 19:309–328, 2005.

Decyk, V. K., C. D. Norton, and B. K. Syzmanski. Expressing object-oriented concepts in fortran90. *ACM Fortran Forum*, 16(1):13–18, 1997.

Graham, S., J. S. Famigliette, and D. Maidment. Five minute, $1/2°$, and $1°$ data sets of continental watersheds and river networks for use in regional and global hydrologic and climate system modeling studies. *Water Resources Research*, 35:583–587, 1999.

Jacob, R., J. Larson, and E. Ong. Mxn communication and parallel interpolation in ccsm3 using the model coupling tookit. *Int. J. High Perf. Comp. App.*, 19: 293–308, 2005.

Jacob, R., C. Schafer, I. Foster, M. Tobis, and J. Anderson. Computational design and performance of the fast ocean atmosphere model. In Alexandrov, V. N., Dongarra, J. J., and Tan, C. J. K., editors, *Proc. 2001 International Conference on Computational Science*, volume 2073 of *Lecture Notes in Computer Science*, pages 175–184, Berlin, 2001. Springer-Verlag.

Larson, J., R. Jacob, and E. Ong. The model coupling toolkit: A new fortran90 toolkit for building multiphysics parallel coupled models. *Int. J. High Perf. Comp. App.*, 19:277–292, 2005.

Larson, J., R. Jacob, and E. Ong. Model Coupling Toolkit. `http://www.mcs.anl.gov/mct/`, 2007.

Lee, J.-Y. and A. Sussman. High performance communication between parallel programs. In *Proceedings of 2005 Joint Workshop on High-Performance Grid Computing and High-Level Parallel Programming Models (HIPS-HPGC 2005)*. IEEE Computer Society Press, April 2005. Appears with the Proceedings of IPDPS 2005.

Lohman, D., R. Nolte-Holube, and E. Raschke. A large-scale horizontal routing model to be coupled to land surface parameterization schemes. *Tellus*, 48A:708–721, 1996.

Miller, J. R., G. L. Russell, and G. Caliri. Continental-scale river flow in climate models. *J. Climate*, 7: 914–928, 1994.

MPI Forum. MPI: A message-passing interface standard. *Int. J. Supercomputer App.and High Perf. Comp.*, 8(3/4):159–416, Fall-Winter 1994.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipies in Fortran 77: The Art of Scientific Computing*. Cambridge University Press, New York, Second edition, 1992.

Sausen, R., S. Schubert, and L. Dümenil. A model of river runoff for use in coupled atmosphere-ocean models. *Journal of Hydrology*, 155:337–352, 1994.

Sushama, L., R. Laprise, D. Caya, M. Larocque, and M. Slivitzky. On the variable-lag and variable velocity cell-to-cell routing scheme for climate models. *Atmosphere-Ocean*, 42(4):221–233, 2004.

Temperly, H. N. V. *Graph Theory and Applications*. John Wiley and Sons, New York, 1981.

Vörösmarty, C., A. Moore, A. Grace, M. Gildea, J. Melilo, B. Peterson, E. Rastetter, and P. Steudler. Continental scale models of water balance and fluvial transport: An application to south america. *Global Biogeochemical Cycles*, 3(3):241–265, 1989.

Washington, W. M., J. W. Weatherly, G. A. Meehl, A. J. Semtner, T. W. Bettge, A. P. Craig, W. G. Strand, J. M. Arblaster, V. B. Wayland, R. James, and Y. Zhang. Parallel climate model (pcm) control and transient simulations. *Climate Dynamics*, 16: 754–774, 2000.