

Coupling Climate and Hydrological Models: Interoperability through Web Services

Jonathan L. Goodall^a, Kathleen D. Saint^b, Mehmet B. Ercan^a, Laura J. Briley^c, Sylvia Murphy^d, Haihang You^e, Cecelia DeLuca^{d,*}, Richard B. Rood^c

^a*Department of Civil and Environmental Engineering, University of South Carolina, 300 Main Street, Columbia, SC 29208*

^b*SGI Inc., 17141 Primavera Cir. Cape Coral, FL 33909*

^c*Department of Atmospheric, Oceanic and Space Sciences, University of Michigan, 2455 Hayward Street, Ann Arbor, MI 48109-2143*

^d*Cooperative Institute for Research in the Environmental Science, University of Colorado Boulder, Box 216 UCB Boulder, CO 80309-0216*

^e*National Institute for Computational Sciences, University of Tennessee and Oak Ridge National Laboratory, PO Box 2008, BLDG 5100, Oak Ridge, TN 37831-6173*

Abstract

Understanding regional-scale water resource systems requires understanding coupled hydrologic and climate interactions. The traditional approach in the hydrologic sciences and engineering fields has been to either treat the atmosphere as a forcing condition on the hydrologic model, or to adopt a specific hydrologic model design in order to be interoperable with a climate model. We propose here a different approach that follows a service-oriented architecture and uses standard interfaces and tools: the Earth System Modeling Framework (ESMF) from the weather and climate community and the Open Modeling Interface (OpenMI) from the hydrologic community. A novel technical challenge of this work is that the climate model runs on a high performance computer and the hydrologic model runs on a personal computer. In order to complete a two-way coupling, issues with security and job scheduling had to be overcome. The resulting application demonstrates interoperability across disciplinary boundaries and has the potential to address emerging questions about climate impacts on local water resource systems. The approach also has the potential to be adapted for other climate impacts applications that involve different communities, multiple frameworks, and models running on different computing platforms. We present along with the results of our coupled modeling system a scaling analysis that indicates how the system will behave as geographic extents and model resolutions are changed to address regional-scale water resources management

problems.

Keywords: Modeling Frameworks, Service-Oriented Architectures, Hydrology, Climate, Modeling

*Corresponding author

Email addresses: goodall@cec.sc.edu (Jonathan L. Goodall), ksaint@sgi.com (Kathleen D. Saint), ercanm@email.sc.edu (Mehmet B. Ercan), auraell@umich.edu (Laura J. Briley), sylvia.murphy@noaa.gov (Sylvia Murphy), hyou@utk.edu (Haihang You), cecelia.deluca@noaa.gov (Cecelia DeLuca), rbrood@umich.edu (Richard B. Rood)

1 **1. Introduction**

2 Projections of the Earth's climate by models provide the primary information for an-
3 ticipating climate-change impacts and evaluating policy decisions. Changes in the water
4 cycle are expected to have impacts on, for example, public health, agriculture, energy
5 generation, and ecosystem services (Parry et al., 2007). The integration of information
6 from climate-model projections with the tools used by practitioners of water manage-
7 ment is a core interest of those developing strategies for adaptation to climate change
8 (Raucher, 2011). Often a hydrological model that is formally separated from a climate
9 model is used in these applications (Graham et al., 2007). In this paradigm, climate pro-
10 jections may be used as a forcing function to drive the decoupled hydrologic simulation
11 model. These applications assume there is no significant feedback from the land surface
12 to the climate system (either regional or global), and while this assumption may be true
13 for small watersheds, as hydrologists continue to scale their models up to river basin
14 and regional systems, this assumption of no feedback loop will need to be addressed.
15 Therefore both intuitively and theoretically, we expect hydrological models to perform
16 better when they are coupled in some way to a global or regional climate model (Xinmin
17 et al., 2002; Yong et al., 2009).

R1.C3

18 A second paradigm for the coupling of hydrological models into global climate systems
19 is to allow two-way communication, so that simulating feedback loops is possible. There
20 are scientific and software challenges posed by either form of coupling. The difference
21 in spatial scales provide an intrinsic challenge when coupling climate and watershed-
22 scale hydrologic models. For a hydrological model used in agricultural decision-making,
23 intrinsic scales must adequately represent the drainage of the streams, the specifics of
24 the land and vegetation in the watershed, surface topography at accuracies of less than
25 a meter, and the surface type of the built environment. Even with the highest resolution
26 climate models likely to be viable in the next five years which promise grid cells on
27 the order of 100 km², there are differences of several orders of magnitude in the spatial
28 scales. Transference of information in a physically meaningful way across these scales,
29 large-to-small and small-to-large, is neither scientifically nor algorithmically established.

30 The work described here is forward looking in that we explore loose coupling of
31 a climate model and a hydrological model with two-way communication between the

32 models using Web Services. This type of coupling might be viewed as a first step towards
33 linking climate models to real-world applications. With the full realization that, from an
34 Earth-science perspective, the spatial resolution of the climate model might not justify
35 the coupling at this time, we propose that there are scientific and algorithmic challenges
36 that are worth addressing. Rather than waiting until the climate models are at some
37 undefined state of readiness to start the coupling, then begin to develop the coupling
38 strategies, we are co-developing the coupling with the models. This will help both to
39 define the scientific foundation of the coupling and to evolve the algorithms in concert
40 with the scientific investigation. This work is related to activities in the computational
41 steering community (e.g. Parker et al., 1998; Malakar et al., 2011) in that we use Web
42 Services to pass data between desktop and climate and weather models. As we move
43 past exploratory and prototyping work, we believe that work related with this field will
44 help to define both the scientific foundation of the coupling and evolve the algorithms in
45 concert with the scientific investigation.

46 The work advances on existing work in Earth System Modeling Framework (ESMF) R1.C1
47 and standards by exploring how two existing modeling frameworks, ESMF and the
48 OpenMI Configuration Editor (OmiEd), can be integrated for cross-framework simu-
49 lations. By leveraging a service-oriented architecture, we show that a climate model
50 implemented within ESMF can be made available as a Web Service, and that an OpenMI-
51 based client-side component can then wrap the ESMF service and use it within an OmiEd
52 configuration. We selected OmiEd (which adopts the OpenMI standard) as the client
53 application in our work because of past work to create ESMF services that could be
54 brought into OmiEd. This work builds on the proposed concept of modeling water re-
55 source systems using service-oriented architectures (Laniak et al., 2012; Goodall et al.,
56 2011; Granell et al., 2010) and extends the work to leverage ESMF models in a personal R1.C2
57 computer-based integrated model configuration. It extends on this work by specifically
58 exploring coupling across modeling frameworks, in particular modeling frameworks that
59 target different communities (climate science and hydrologic science) that have differ-
60 ent models, best practices, and histories for building computer-based model simulation
61 software. By using a service-oriented, loose-coupling approach, we are able to maintain
62 state-of-the-art community supported models within the integrated modeling system.

63 There are other aspects of this work that address the use of climate projections in
64 decision making. As discussed by Lemos and Rood (2010) and others, there are many
65 research questions to be answered in bridging scientists' perceptions of the usefulness of
66 climate information and practitioners' perceptions of usability. Co-generation of knowl-
67 edge and methodology has been shown to be an effective way to address these questions;
68 discipline scientists, software specialists, and practitioners learn the constraints that each
69 must face. This improves the likelihood of successful use of climate information. In the
70 development that we are pursuing, we will be using a hydrological model that is widely
71 used in agricultural decision-making. Thus, we are not only coupling Earth science mod-
72 els implemented for different spatial scales, but we are laying the foundation for diverse
73 communities of experts to interact in a way they have not done previously by enabling
74 bidirectional coupling of distributed models outside the scope of a single integrated cli-
75 mate model.

76 Given this motivation, the first objective of our research was to design a system ca-
77 pable of coupling widely used models in the atmospheric and hydrologic communities
78 in a way that maintains the original structure and purpose of each model but provides
79 coupling of flux and state variables between the two models. The second objective was
80 to assess the applicability of the approach by conducting a scaling analysis experiment.
81 The purpose of the scaling analysis was to quantify the performance of the coupled hy-
82 dro/climate model in terms of the hydrology model execution time, the climate model
83 execution time, and time required for transferring data between the two models. We
84 present the methodology for addressing these two study objectives in the following sec-
85 tion. We then present the results of the scaling analysis, and discuss our findings for the
86 applicability of our proposed approach for model coupling.

87 **2. Methodology**

88 Our methodology consists of two main tasks. First, we designed an overall system to
89 consist of three components: a hydrological model, an atmospheric climate model, and
90 the driver application. The design of this system, which we refer to as the Hydro-Climate
91 Modeling System, is described in the first subsection and a prototype implementation
92 of the system is described in the second subsection. Second, we devised a series of

93 experiments with the goal of estimating how the Hydro-Climate Modeling System would
94 scale as the size of the study region increases. These experiments are meant to provide
95 an approximate measure of scaling that will aid in optimizing performance of the system
96 and improve understanding of the applicability of the approach for simulating regional-
97 scale hydrologic systems. Details of the scaling analysis design are presented in the third
98 and final subsection of this methodology section.

99 *2.1. Hydro-Climate Modeling System Design*

100 Within this general service-oriented framework, the target of our prototype is a two-
101 way coupled configuration of the Community Atmosphere Model (CAM) and the hydro-
102 logical model Soil and Water Assessment Tool (SWAT) that captures the coupled nature
103 of the physical system. The intent of our coupling was not to produce realistic simu-
104 lations, but to explore the behavior of a technical solution spanning high performance
105 computing and Web Services. Thus the specifics of the configuration matter here only
106 insofar as they represent a scientifically plausible exchange, and serve as a starting point
107 for design decisions and for exploring the behavior and scaling of the coupled system.
108 We fully expect that the models used, and the specifics of the coupling, may change as
109 our investigation continues and new models and resources become available. The use of
110 models with structured component interfaces facilitates such exploration because of the
111 “plug-and-play” functionality provided through component interface standardization.

112 In the chosen configuration, CAM supplies to SWAT a set of five fields (surface air
113 temperature, wind speed, precipitation, relative humidity, and solar radiation) for each
114 30 minute interval of the model simulation. SWAT passes one field, evaporation, back to
115 CAM also on a 30 minute interval. CAM was run in a Community Earth System Model
116 (CESM) configuration that included active atmosphere, land, and ice model components,
117 as well as a data ocean representation (in place of an active ocean component). Issues
118 related to how best to incorporate output from the SWAT model into the CAM model R3.C2
119 (e.g., regridding of data exchanges) were not addressed through this work. Instead our
120 focus was on the technical issues related on data transfers between the coupled models.
121 Proof of concept runs were performed with CAM at 1 degree resolution and SWAT for
122 the Eno Basin in North Carolina (171 km²). Following this proof of concept, a scaling
123 analysis was performed and used to explore resolutions of CAM spanning 1 to 1/4 degree

124 and SWAT for a set of domains ranging in size from 171 km² to 721,000 km². This
125 technical implementation and scaling analysis is described in more detail in following
126 subsections.

127 The technical design of the Hydro-Climate Modeling System emphasizes the loose
128 coupling of models through data exchanges over a standard interface. Figure 1 provides
129 a high-level description of the system architecture. The hydrological model SWAT runs
130 on a Windows-based personal computer and had already been integrated with the Open
131 Modeling Interface (OpenMI) by the UNESCO/IHE group (Betrie et al., 2011). The
132 atmospheric/climate model CAM runs on a high-performance computing (HPC) plat-
133 form and an OpenMI wrapper is used to provide the standard interface on the Windows
134 personal computer while providing access to the climate model via a Web Service-based
135 interface. Communication between the two models is driven by the OmiEd, which pro-
136 vides a Graphical User Interface (GUI) that is used to define the link (data inputs and
137 outputs) between the two models and then execute the model run. The approach taken
138 could be generalized for other HPC component interfaces, other Web Service interfaces,
139 or other simulation models. Details of the system components follow.

140 *2.1.1. The Watershed Hydrology Model*

141 SWAT is a watershed-scale hydrologic model developed to quantify the impact of
142 land management practices in large, complex watersheds over long time periods (e.g.,
143 multiple years or decades) (Arnold and Allen, 1996). SWAT can be characterized as a
144 semi-distributed model where a watershed is divided into subbasins, and then further
145 into Hydrologic Response Units (HRUs). Each HRU is a lumped unit with unique soil,
146 land use and slope characteristics. Subbasins are connected through stream topology
147 into a network, however HRUs are not spatially located within a subbasin. SWAT was
148 selected for this project because it is a widely used watershed model for rural watersheds
149 (Gassman et al., 2007), it is under active development, and it is open source. Also, as
150 previously mentioned, past work has resulted in an Open Modeling Interface (OpenMI)-
151 compliant version of SWAT that was leveraged in this work (Betrie et al., 2011).

152 Specific submodels within SWAT used for the analysis were the Penman-Monteith
153 method for evapotranspiration, the Green-Ampt model for infiltration, and a variable
154 storage method for channel routing. We used Green-Ampt because the climate model is

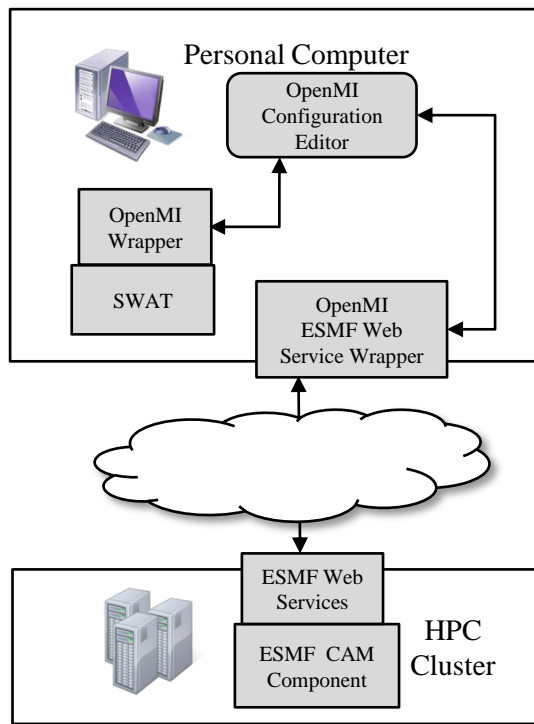


Figure 1: Diagram of the Hydro-Climate Modeling System system showing the components on the personal computer and the components on the HPC system as well as their interactions.

155 able to provide weather input data on a 30 minute-time step. The SWAT model internal
 156 time step was set to 30 minutes due to the availability of climate information. This
 157 model design was used to construct three different watershed models, chosen in order
 158 to quantify how SWAT computational scales with increasing watershed area: the Eno
 159 Watershed (171 km²), the Upper Neuse Watershed (6,210 km²), and the Neuse River
 160 Basin (14,300 km²). Additional detail on these SWAT models is provided in the Scaling
 161 Analysis section.

162 The OpenMI standard defines a sequential approach to communicate between mod-
 163 els that provides a detailed view of the method calls for the system (Figure 2). The
 164 OpenMI Software Development Kit (SDK) is a software library that provides the hy-
 165 drological community with a standardized interface that focuses on time dependent data
 166 transfer. It is primarily designed to work with systems that run simultaneously, but in a

167 single-threaded environment. Regridding and temporal interpolation are also part of the
 168 OpenMI SDK (Gregersen et al., 2007), although they were not leveraged through this R1.C5
 169 work. An OpenMI implementation must follow these fundamental steps of execution:
 170 initialization and configuration, preparation, execution, and completion. These steps
 171 correspond to methods in what OpenMI refers to as a LinkableComponent interface:
 172 Initialize, Prepare, GetValues, and Finish/Dispose. Climatological input exchange items
 173 to SWAT include air temperature, precipitation, relative humidity, solar radiation data, R3.C1
 174 and wind speed data on each model time step (Gassman et al., 2007).

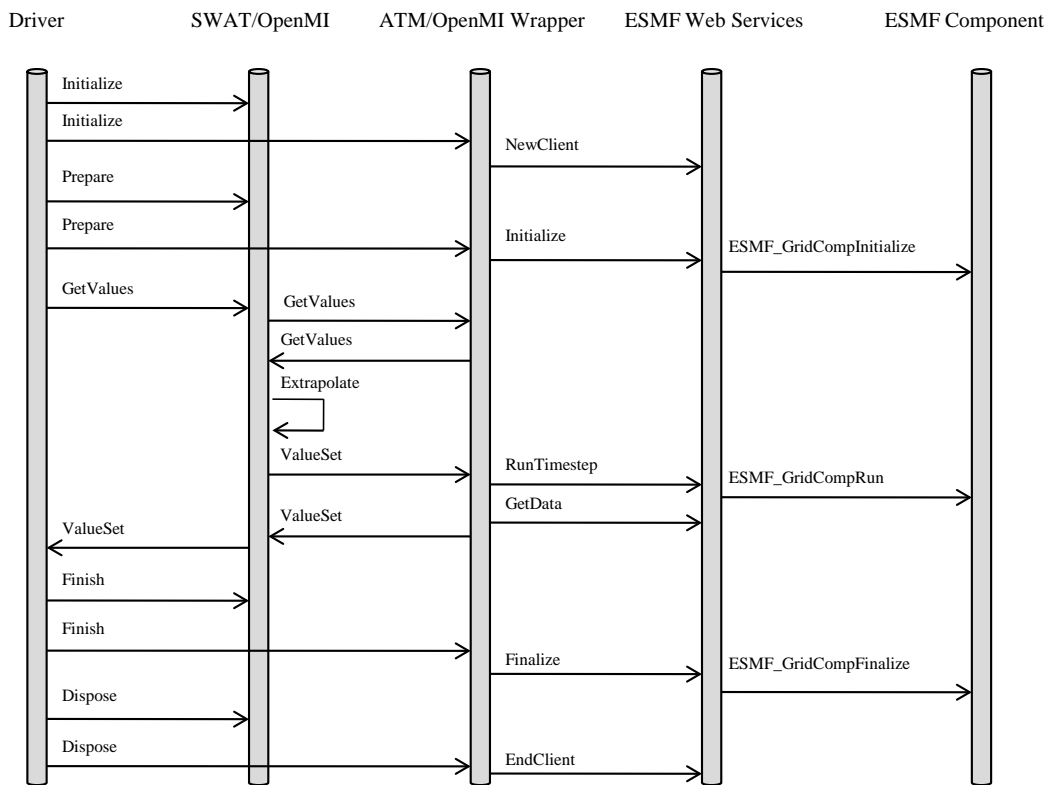


Figure 2: The method calling sequence for the entire system

175 *2.1.2. The Atmospheric General Circulation Model*

176 The atmospheric general circulation model used in this system, the Community Atmo-
177 sphere Model (CAM), is a component of the Community Earth System Model (CESM).
178 The most recent release of CAM, version 5, is documented in Neale et al. (2010). This
179 model is widely used and well documented, with state-of-the-art scientific algorithms
180 and computational performance. CAM also supports several dynamical cores, grid reso-
181 lutions and grid types, including newer grids such as HOMME (Dennis et al., 2005) that
182 can be run at resolutions that begin to approach local hydrological scales. The CAM
183 model is distributed with standard ESMF interfaces, described in more detail in the next
184 section. This combination of attributes and a community-anchored, known development
185 path make CAM a suitable choice for our research and development.

186 The high performance computing platform selected for the climate model was kraken,
187 a CRAY XT5 system with 112,896 cores located at the National Institute for Compu-
188 tational Sciences (NICS), a joint project between the University of Tennessee and Oak
189 Ridge National Laboratory. The kraken machine is part of the NSF Extreme Science
190 and Engineering Discovery Environment (XSEDE), which is an interconnected set of
191 heterogeneous computing systems. We chose this platform because the XSEDE environ-
192 ment offered a less onerous security environment than other supercomputers for the Web
193 Service prototyping work, as described later in this section.

194 The ability to remotely interface with CAM was made possible by the integration
195 of ESMF with CAM. ESMF provides an architecture for composing complex, coupled
196 modeling systems and utilities for developing individual models (Hill et al., 2004). ESMF
197 is generally used to wrap model representations of large physical domains (atmosphere,
198 ocean, etc.) with standard calling interfaces. These interfaces have the same structure
199 for each component, and enable the components to be updated or exchanged more easily
200 than *ad hoc* calling interfaces. A Web Services module is included as part of the ESMF
201 distribution and provides the ability to remotely access the calling interfaces of ESMF
202 components. This is a new feature of ESMF and this project is one of the first applications
203 that has leverage the ESMF Web Service interfaces.

204 ESMF component interfaces are supported for all major components in CESM, in-
205 cluding CAM. Each component is split into one or more initialize, run, and finalize phases.

206 Data is passed between components using container classes called States, and synchro-
207 nization and timekeeping is managed by a Clock class. The interfaces are straightforward,
208 and for an atmospheric model the “initialize” phase would be expressed as

```
209 subroutine myAtm_Init(gridComp, importState, exportState, clock, rc)
```

210 where `gridComp` is the pointer to the atmospheric component, `importState` contains the
211 fields being passed in, `exportState` contains the output fields, and the `clock` object
212 contains information about the timestep and start and stop times.

213 States may contain a variety of different data classes, including ESMF Arrays, Array-
214 Bundles, Fields, FieldBundles, and nested States. ESMF Arrays store multi-dimensional
215 data associated with an index space. The ESMF Field includes a data Array along with
216 an associated physical grid and a decomposition that specifies how data points in the
217 physical grid are distributed across computing resources. ArrayBundles and FieldBun-
218 dles are groupings of Arrays and Fields, respectively.

219 The ESMF Web Services module provides the tools to enable remote access to any
220 ESMF compliant component using standard web protocols. This module, as part of
221 the ESMF library, is comprised of several pieces: a Fortran interface to a Component
222 Server class, a Process Controller application, a Registrar application, and a set of Simple
223 Object Access Protocol (SOAP) services that, when installed with Apache/Tomcat and
224 Axis2, provide web access to the Process Controller.

225 For a climate model to be integrated with ESMF Web Services, it first must be
226 integrated with ESMF and have ESMF Components. Integration of a climate model
227 with ESMF Web Services involves modifying the driver code to enter a service loop
228 (provided as part of the library) instead of executing the initialize, run and finalize
229 routines. In addition, also using the library routines, the climate model is modified to
230 read and/or write data values for each timestep. Finally, the climate model needs to
231 be modified to accept specific command line arguments that are passed to the ESMF
232 Web Services library routines. This integration completes the creation of a Component
233 Service. To execute this component service on a High Performance Computing (HPC)
234 platform using a job scheduler, there are some UNIX shell script files that need to be
235 modified to execute the appropriate job scheduler commands to start, status, and stop
236 a batch job.

237 The remaining integration with ESMF Web Services involves software installation
238 and configuration. The Process Controller and Registrar need to be installed on the
239 login nodes. These are generic applications and do not require any code modifications to R1.C6
240 work with the climate model. Configuration files and command line arguments are used
241 to customize these applications for the specific platform (providing hostname and port
242 numbers, for example). Finally, the SOAP Services package needs to be installed in the
243 appropriate Axis2 services directory on the host that provides the web server.

244 When looking for an HPC platform to host this prototype, we ran into security R2.C1/
245 concerns from systems and security administrators. The primary issue was our need to R3.C6
246 open a port (via POSIX sockets) on the HPC/compute host. While this was considered
247 a potentially risky approach, the XSEDE team was willing to work with our team to
248 determine where the risks were and to find ways to work around them. The first step
249 was to protect the HPC host from unwanted access. The host we used, kraken, already
250 protected its compute nodes by restricting access to them from only the login nodes.
251 The Process Controller ran as an independent application and could remotely access the
252 Component Server. By running the Component Server on the compute node and the
253 Process Controller on the login node, we were able to comply with the access restriction
254 that only login nodes could access the compute nodes.

255 Access to the login nodes was also restricted, but to a wider domain; only nodes
256 within the XSEDE network could have direct access to the login nodes. To work with
257 this restriction, the XSEDE team provided a gateway host (a virtual Linux platform)
258 within the XSEDE network. This host was able to access the Process Controller socket
259 port opened on the kraken login node, as well as provide access to the XSEDE network
260 from the Internet using standard and known web technologies. Therefore, by breaking
261 down the prototype software into multiple, remotely accessible processes that could be
262 installed across multiple platforms, we were able to work with the security restrictions
263 and provide an end-to-end solution.

264 2.1.3. *The Driver*

265 The system driver controls the application flow and is implemented using the OpenMI
266 Configuration Editor (OmiEd). The Configuration Editor is provided as part of the
267 version 1.4 OpenMI distribution, runs on a Windows-based personal computer platform,

268 and provides the GUI and tools to link and run OpenMI compliant models. The version
269 of SWAT used in this system was provided as an OpenMI compliant model, but the
270 CAM model needed to be wrapped with an OpenMI interface. This was accomplished
271 by implementing the OpenMI classes on the Windows platform that, upon execution,
272 dynamically accesses the ESMF Web Services interface for the CAM Component Service.
273 The ESMF Web Services provide the bridge between the Windows personal computer
274 and the HPC platform.

275 The Configuration Editor works by loading the models as defined in OpenMI config-
276 uration files (OMI files). A Trigger is created to kick off the run, and Links are used
277 to define the data exchanged between the models. When a model is loaded into the
278 Configuration Editor, its input and output exchange items are defined. The user then
279 specifies how models exchange data by mapping output exchange items in one model to
280 input exchange items in the other model, and the Configuration Editor and the OpenMI
281 SDK provide the tools to handle the translation between the exchange items.

282 OpenMI and ESMF were the interface standards used for this project because they
283 each provide a standard interface for their respective model communities - ESMF for
284 climate models and OpenMI for hydrological models. Bridging these two standards was
285 at the heart of this coupling challenge; the ability to control execution of each model at the
286 timestep level was critical to providing a common exchange mechanism. In addition, each
287 standard provided features that allowed us to bridge the platform gap; ESMF supporting
288 access via Web Services and OpenMI supporting a wrapper construct to access external
289 services such as ESMF Web Services. Finally, the ability of each interface to allow the
290 implementor to define the data input and output formats allowed us to use the OpenMI
291 Configuration Editor to translate the formats between the two models. The features and
292 tools of both ESMF and OpenMI provided us with the ability to couple the climate and
293 hydrological models while maintaining the models' native environments.

R2.C3

294 *2.2. Hydro-Climate Modeling System Proof-of-Concept Implementation*

295 The use of an HPC environment within a distributed, service-oriented architecture
296 presented some unique technical and programmatic challenges that we had to overcome.
297 As discussed before, security was a challenge because access to the login and compute
298 nodes of an HPC platform are typically very restricted. In addition, resource utilization

299 is of primary concern to the system administrators, and they need to be confident that
 300 the compute nodes are not unnecessarily tied up. Finally, running applications on HPC
 301 platforms typically requires the use of a batch job scheduler, and running an interactive
 302 application from a job scheduler in a batch environment adds another level of complexity
 303 that must be addressed.

304 The kraken platform that we used for this work utilizes the Moab job scheduler in
 305 combination with the Portable Batch System (PBS). Figure 3 shows the architecture of
 306 the software for the service portion of the CAM implementation. The HPC platform
 307 is comprised of a set of compute nodes, on which the CAM Component Service is run,
 308 as well as a set of login nodes, from which we can access the Service. Because the
 309 HPC administrators preferred to not have a web server running on the HPC platform, a
 310 separate virtual host within the XSEDE environment was created for this purpose.

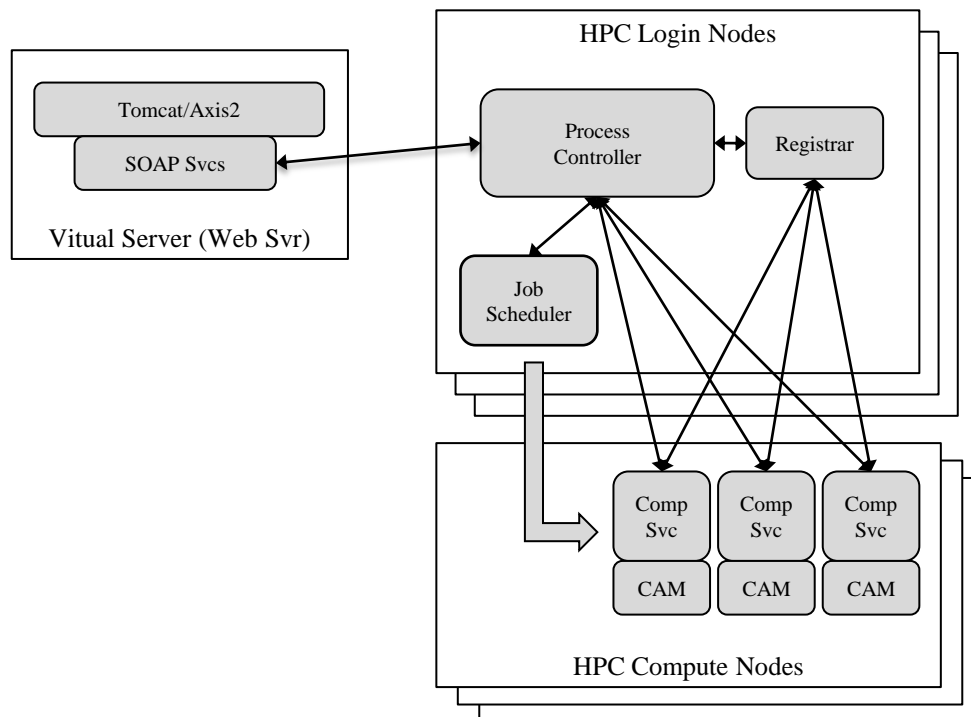


Figure 3: Architecture of the software for the service portion of the CAM component

311 The Process Controller and Registrar, both daemons that run on a login node, are

312 critical for managing the CAM Component Services within an HPC environment. The
313 Process Controller provides all access to the CAM Component Services, including startup
314 and shutdown; all communication to these Services is handled through the Process Con-
315 troller. The Process Controller is also responsible for handling resource utilization by
316 ensuring that a CAM Component Service does not sit idle for too long; it terminates the
317 Service if the client has not accessed it within a specified period of time.

318 The Registrar is needed in order to determine the state of a CAM Component Service
319 at all times. When the Process Controller starts a CAM Component Service, it registers
320 the new Service with the Registrar and sets the state to WAITING TO START. When
321 the job scheduler starts the CAM Component Service, the Service updates its registra-
322 tion in the Registrar to indicate that it is READY to receive requests. As the Service
323 enters different states (i.e., initializing, running, etc.), it updates its information with the
324 Registrar. All requests for the status of a CAM Component Service are handled by the
325 Process Controller and retrieved from the Registrar.

326 A user of the system would complete the following steps in order to run a model
327 simulation. First, the prerequisite for a user to run the system is that the Web server
328 (Apache/Tomcat), the Process Controller and the Registrar must all be running. These
329 are all daemon applications and, in an operational system, would be running at all times.
330 The first step for a user in running the system is to start up the OpenMI Configuration
331 Editor and load the simulation configuration file. This file defines the SWAT and CAM
332 models, a Trigger to kick off the run, and the Links between all of the parts. The Links
333 contain the mappings between the input and output exchange items of the two models.
334 The CAM OpenMI interface contains all of the information needed to access the ESMF
335 Web Services, so the user does not need to enter any information. To start the simulation,
336 the user simply needs to execute the Run command from the Configuration Editor.

337 The following steps describe what happens when the system is run. Figure 2 pro-
338 vides a high-level sequence diagram that also describes these steps. The first step in the
339 OpenMI interface is to call the Initialize method for each model. For the CAM model,
340 this involves calling the NewClient interface to the ESMF Web Services, which, via the
341 Process Controller, instantiates a new CAM Component Service by requesting that the
342 job scheduler add the Service to the startup queue. Each client is uniquely identified and

343 is assigned to its own Component Service; no two clients can access the same Component
344 Service. When the job scheduler does eventually start the CAM Component Service, it
345 registers itself with the Registrar as ready to receive requests. At this point, the Config-
346 uration Editor continues by calling the Prepare method for each model. For the CAM
347 model, this involves calling the Initialize Web Service interface, which in turn makes an
348 Initialize request to the CAM Component Service via the Process Controller.

349 Once the models are initialized, the Configuration Editor time steps through the
350 models. For each timestep, the SWAT model requests input data from the CAM model
351 using the OpenMI GetValues method. This call triggers the CAM OpenMI wrapper
352 to timestep the CAM Component Service (using the RunTimestep interface) and then
353 retrieve the specified data values using the GetData interface. This process is repeated
354 for each of the timesteps in the run. With two-way coupling implemented, the initial
355 OpenMI GetValues call is made to both of the models, creating a deadlock. In order to
356 break this deadlock, one of the models (the SWAT model, in our prototype) extrapolates
357 the initial data values and provides this data as input to the other model. This model
358 then uses the extrapolated data to run its initial timestep and return data for the first
359 model. The process then continues forward with the timesteps alternating between the
360 models and the data exchanged for each of the timesteps (see Elag and Goodall (2011)
361 for details). Figure 4 provides a graphical description of the data exchange process.

362 At the end of the run, the Configuration Editor cleans up the models by calling
363 the OpenMI Finish method, which is passed on to the CAM Component Service using
364 the Finalize interface. Finally, the OpenMI Dispose method is called which causes the
365 CAM OpenMI wrapper to call the EndClient interface and the CAM Component Service
366 application to be terminated.

367 The current prototype waits for updates using a polling mechanism; the client con- R3.C5
368 tinually checks the status of the server until the server status indicates the desired state.
369 This is not ideal because it requires constant attention from the client. In addition, it
370 uses up resources by requiring network traffic and processing time for each status check.
371 Ideally, this mechanism will be replaced in the future with a notification mechanism. Us-
372 ing this approach, the client can submit its request and will be notified when the server
373 is ready. The client can then handle other tasks and the system will not be burdened

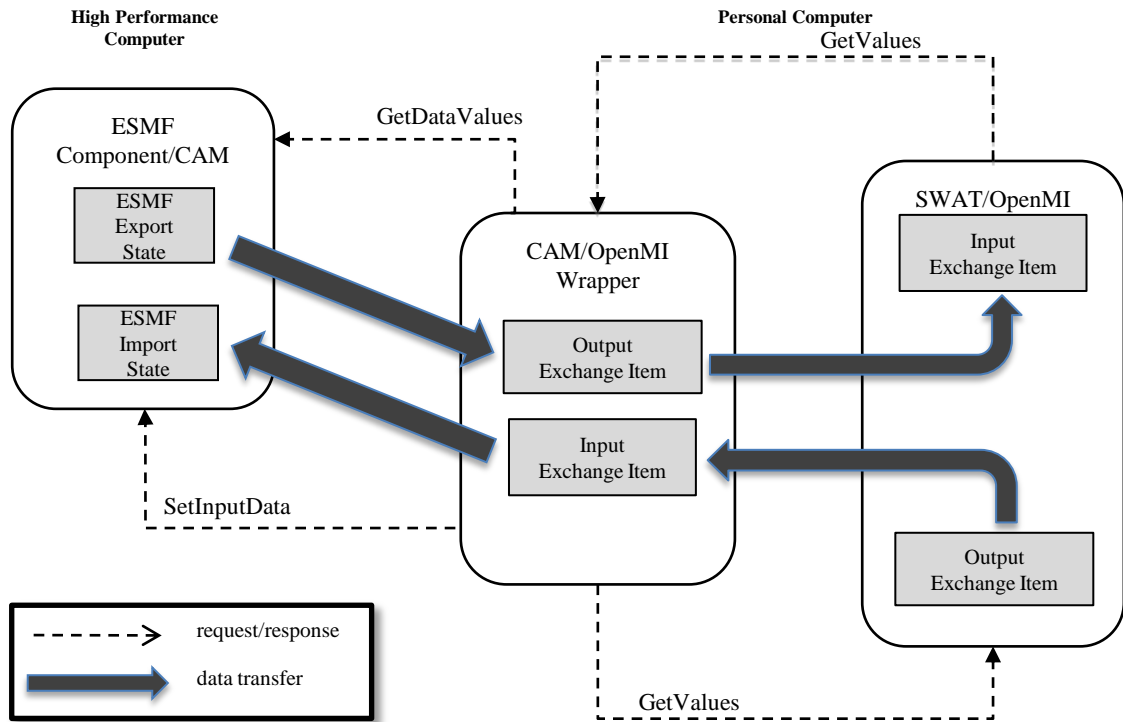


Figure 4: The flow of data through the Hydro-Climate Modeling System from the hydrology model, the atmospheric model, and the system driver.

374 again until the server is ready to proceed.

375 *2.3. Scaling Analysis*

376 A scaling analysis was performed in order to understand the current behavior of the
 377 coupled system, to inform the technical design, to predict ways in which the evolution
 378 of models and computational environment would be likely to change the behavior of the
 379 coupled system over time, and to identify the categories of scientific problems that the
 380 approach could be used to address, now and in the future. This analysis was done prior to R2.C2
 381 the completed implementation of the coupled system, and used a combination of actual
 382 model execution times along with extrapolated runtime values. It should be made clear
 383 that the goal of this analysis was not to provide a precise measurement of performance
 384 for each scale, but to provide a general overall impact of scale on the system design.

385 *2.3.1. Hydrologic Model Scaling Analysis Design*

386 To obtain baseline runtime models for SWAT, we pre-processed the SWAT model
387 input data using a SWAT pre-processing tool created within an open-source Geographic
388 Information System (GIS): MapWindow SWAT (Leon, 2007; Briley, 2010). Topography
389 data was obtained from the National Elevation Dataset at a 30 m resolution, land cover
390 data was obtained from the National Land Cover Dataset (NLCD) at 30 meter resolu-
391 tion, and soil data was obtained from the State Soil Geographic (STATSGO) Database
392 at a 250 m spatial resolution. Hydrologic Response Units (HRUs) were derived from
393 versions of land use and soil classifications generalized using 10% threshold values so
394 that we obtained approximately 10 HRUs per subbasin as suggested in the SWAT model
395 documentation (Arnold et al., 2011).

396 We did this data pre-processing work for three regions (Figure 5). The smallest wa-
397 tershed considered was a portion of the Eno Watershed (171 km²) in Orange County,
398 North Carolina. The Upper Neuse Watershed (6,210 km²) that includes the Eno Wa-
399 tershed and is an 8-digit Hydrologic Unit Code (HUC) in the USGS watershed coding
400 system, served as the second watershed. The third watershed was the Neuse River Basin
401 (14,300 km²) which consists of 4 8-digit HUCs. SWAT is not typically used for water-
402 sheds larger than the Neuse, in part because it is a PC-based model and calibration and
403 uncertainty analysis of the model can take days of runtime for watersheds of this size.
404 We then performed 10 year simulations using the 2009 version of SWAT for each of the
405 three study watersheds.

406 We did not calibrate any of our SWAT models because it was not necessary to do
407 so for the aims of this study. Because we are simply interested in understanding how
408 model execution time depends on watershed area, whether or not the model is calibrated
409 should not significantly impact the results of the study. However, other factors such
410 as our decisions of how to subdivide the watersheds into subbasin units, and how to
411 subdivide subbasin units into Hydrologic Response Units (HRUs) would be important
412 in determining model runtime. For this reason we choose typical subbasin sizes in this
413 study and kept to the suggested 10 HRUs per subbasin as previously discussed.

414 Not included in this analysis are the overhead processing times associated with the
415 OpenMI wrappers or the OpenMI driver. We expect these times to be approximately

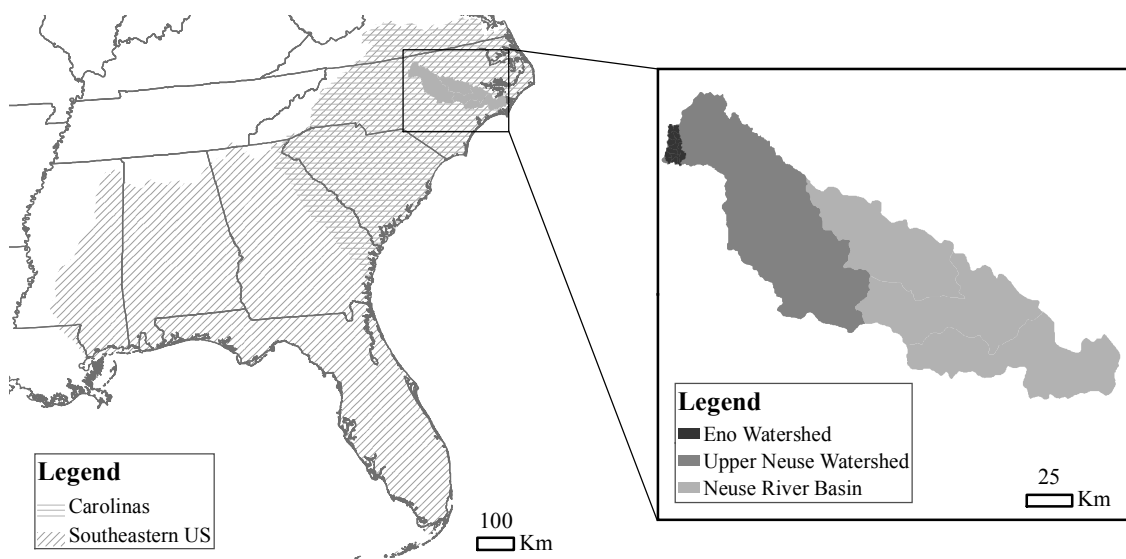


Figure 5: The regions used for the SWAT scaling analysis. The Neuse River Basin includes the Upper Neuse Watershed, and the Upper Neuse Watershed includes the Eno River Basin. SWAT models were created for the watersheds to calculate execution time. These numbers were then scaled to estimate execution times for the Carolinas and Southeastern United States regions.

416 constant for the scales we considered, and for this reason did not include them in our
417 analysis.

418 *2.3.2. Atmospheric Model Scaling Analysis Design*

419 A key computational constraint is the running time of the Community Atmosphere
420 Model (CAM). The operations count and the computational performance of a discrete
421 atmospheric model increases with the number of points used to describe the domain. To
422 a first approximation in a three dimensional model, if the horizontal and the vertical
423 resolution are both doubled then the number of computations is increased by 8, 2^3 . If
424 the time scheme is explicit, a doubling of the resolution requires that the time step be
425 reduced by half, leading to another power-of-2 increase in the number of operations.
426 Implicit time schemes, which solve a set of simultaneous equations for the future and
427 past state, have no time step restriction and might not require a reduction in time step
428 in order to maintain stability. As an upper limit, therefore, the operations increase as a
429 power of 4. This scaling analysis is based on the dynamical core defining the number of
430 operations. In practice, this is the upper range of the operations count, as the physics
431 and filters do not require the same reduction in time step as the dynamical core (Wehner
432 et al., 2008). In most applications, as the horizontal resolution is increased the vertical
433 resolution is held constant. Therefore the upper limit of the operations count for an
434 atmospheric model scales with the power of 3. When considering the model as a whole,
435 long experience shows that a doubling of horizontal resolution leads to an increase of
436 computational time by a factor of 6 to 8.

437 Not included in this analysis are the overhead processing times associated with the R3.C3
438 Web/SOAP server, the Process Controller or the Registrar. These times were consid-
439 ered constant for all scales, and we did not feel they would affect the analysis or our
440 conclusions.

441 *2.3.3. Data Communication Packets*

442 In addition to SWAT and CAM model execution times, the third component of the
443 coupled model scaling is the data transfer times for messages passed through the Web
444 Service interface between the hydrologic and atmospheric models. Assuming a two-way
445 coupling between the models, the total data transfer time includes both the request and

446 reply from SWAT to CAM and back from CAM to SWAT. Taking first the request and
447 reply from SWAT to CAM, we assumed that the request would include a 4 byte request
448 ID, an 8 byte request time, and a 4 byte request package identifier. Therefore the total
449 request data packet size would be 16 bytes. We further assumed that the reply would
450 include a 4 byte request status, the 8 byte request time, and the 4 byte request package
451 identifier along with the five values passed from CAM to SWAT (surface air temperature,
452 wind speed, precipitation, relative humidity, and solar radiation) and the latitude and
453 longitude coordinates for the point passed from CAM to SWAT. Assuming data values
454 and coordinate values are each 8 bytes, then the total reply packet size would be 16 bytes
455 (for overhead) + 56 bytes \times the number of points passed between SWAT and CAM (for
456 values and coordinates). To complete the two-way coupling, the CAM to SWAT request
457 and reply was assumed to be the same except that only one data value is passed in this
458 direction (evaporation). Therefore the data transfer from CAM to SWAT would consist
459 of a 16 byte request and a reply of 16 (overhead) + 24 \times the number of points passed
460 between CAM and SWAT (values and coordinates) bytes.

461 We understood when doing this analysis that there would be additional overhead
462 associated with network traffic. Since this effort was considered to be an approximation,
463 and since the overhead associated with the network traffic was not impacted by the model
464 scaling, we did not account for this factor in the scaling analysis.

465 **3. Results and Discussion**

466 *3.1. Hydrologic Model Scaling Results*

467 Results from the SWAT model scaling experiment for the Eno Watershed, Upper
468 Neuse Watershed, and Neuse River Basin were 7.2×10^{-3} , 1.4×10^{-1} , and 2.5×10^{-1}
469 seconds of wall time per day of simulation time (sec/d). These values were determined
470 from a 10 year simulation run. To extrapolate execution times for the Carolinas and
471 Southeastern (SE) United States regions, which were too large to prepare SWAT input
472 files for as part of this study, a linear function was fitted to these data points to relate
473 drainage area to model execution time. We assumed a linear relationship between model
474 execution time and drainage area from knowledge of the SWAT source code, past expe-
475 rience with the model, and additional tests run to verify this assumption. Results from

476 this extrapolation were that SWAT model execution for the Carolinas is estimated to
 477 be 3.8 sec/d, and execution time for the Southeastern United States is estimated to be
 478 12 sec/d. These values, which are summarized in Table 1, resulted from running SWAT
 479 2009 on a typical Windows workstation that consists of a 64-bit Intel Core i7 2.8 Ghz
 480 CPU with 4 GB of RAM.

Table 1: Measured SWAT execution times for the Eno Watershed, Upper Neuse Watershed, and Neuse River Basin. Estimated execution times for the Carolinas and Southeastern United States regions.

Basin Name	Drainage Area (km ²)	Subbasins (count)	HRUs (count)	10 yr Run (sec)	1 d Run (sec)
Eno Watershed	171	6	65	26.4	0.0072
Upper Neuse Watershed	6,210	91	1064	504	0.14
Neuse River Basin	14,300	177	1762	897	0.25
Carolinas*	222,000	-	-	-	3.8
SE USA*	721,000	-	-	-	12

* Estimated based on linear fit between execution time and drainage area

481 The SWAT scaling analysis does not consider potential techniques for performing
 482 parallel computing. One means for performing parallel tasks within SWAT is to consider
 483 each major river basin within the study domain as an isolated computational task. Using
 484 this approach, one would expect model execution times to remain near the times found
 485 for the Neuse River Basin experiment (2.5×10^{-1} sec/d). Recent work has also shown
 486 how a SWAT model can be parallelized for GRID computing by splitting a large SWAT
 487 model into sub-models, submitting the split sub-models as individual jobs to the Grid,
 488 and then reassembling the sub-models back into the large model once the individual sub-
 489 models are complete (Yalew et al., In Press). An approach like this could be used here
 490 to further reduce SWAT model execution time when scaling to larger regions. Lastly,
 491 we are aware that other hydrologic models are further along the parallelization path
 492 (e.g. Tompson et al., 1998) and another possible way to improve model performance
 493 would be to exchange SWAT for these other models within the proposed service-oriented
 494 framework.

495 *3.2. Atmospheric Model Scaling Results*

496 In order to provide empirical verification of our scaling analysis, we ran the finite vol-
 497 ume dynamical core of CAM configured for the gravity wave test of Kent et al. (2012).
 498 This model configuration does not invoke the physical parameterizations of CAM and is
 499 a good representation of the scale-limiting dynamical core of CAM. This configuration
 500 does use the filters and advects four passive tracers. The filters are a suite of computa- R1.C7
 501 tional smoothing algorithms that are invoked to counter known inadequacies of numerical
 502 techniques (Jablonowski and Williamson, 2011). The passive tracers represent trace con-
 503 stituents in the atmosphere that are important as either pollutants or in the control of
 504 heating and cooling. This model configuration is of sufficient complexity that it is a good R3.C4
 505 proxy for the scaling of a fully configured atmospheric model. On 24 processors (2 nodes
 506 of 12 processor core Intel I7, 48GB RAM per node, and 40 Gbps Infiniband between
 507 nodes), we ran 10-day-long experiments with 20 vertical levels at horizontal resolutions
 508 of, approximately, 2 degrees, 1 degree, and 0.5 degree. The results are provided in Table
 509 2. The increase of the execution time in the first doubling of resolution is a factor of 6.1
 510 and in the second doubling a factor of 7.2, both consistent with our scale analysis and
 511 previous experience. For a 0.25 degree horizontal resolution we have extrapolated from
 512 the 0.5 degree resolution using the cube of the operations count, a factor of 8.

Table 2: Measured CAM execution times for a 10-day-long experiment with 20 vertical levels at horizontal resolutions of, approximately, 2 degrees, 1 degree, 0.5 degree, and 0.25 degree. A 24 processor cluster was used for the experimental runs.

Resolution (deg)	Time Step (sec)	Execution Time (sec)
2	360	3,676
1	180	22,473
0.5	90	161,478
0.25	45	1,291,824*

* Estimated as 8 times the 0.5 degree resolution execution time

513 This scaling analysis does not consider the behavior of the model as additional pro-

514 cessors are added to the computation. As documented in Mirin and Worley (2012) and
515 Worley and Drake (2005), the performance of CAM on parallel systems is highly de-
516 pendent on the software construction, computational system, and model configuration.
517 Often it is the case that the scaling based on operations count is not realized. Mirin
518 and Worley (2012) reports on performance of CAM running with additional trace gases
519 on different computational platforms at, approximately, 1.0 and 0.5 degrees horizontal
520 resolution. They find, for example, on the Cray XT5 with 2 quad-core processors per
521 node, with the one degree configuration, the ability to simulate approximately 4 years per
522 day on 256 processor cores and approximately 7 years per day on 512 processor cores.
523 On the same machine a doubling of resolution to the half degree configuration yields
524 approximately 1.5 years of simulation per day on 512 processors. This is about a factor
525 of 5 on performance. Such scaling is representative of the results of Mirin and Worley
526 (2012) for processor counts < 1000 processors on Cray XT5. At higher processor counts
527 the scaling is far less predictable.

528 *3.3. Coupled Hydro-Climate Model Scaling Results*

529 The total execution times (Table 3; Figure 6) were determined by summing the SWAT
530 and CAM model execution times along with the data transfer times. The SWAT model
531 execution times were taken from the scaling analysis described in Section 3.1. The CAM
532 model execution time of 24 sec/d is based on 1 and 5 day CESM runs on 4.7 GHz IBM
533 Power6 processors. The atmospheric component was configured to use 448 hardware
534 processors using 224 MPI processes and 2 threads per process, with a grid of 0.9x1.25
535 and the B_2000 component set. Then the scaling factor of 8 obtained from the scaling
536 analysis described in Section 3.2 was used to obtain the higher resolution CAM model
537 execution times of 192 and 1,536. We note that Mirin and Worley (2012) obtained similar
538 execution times for CAM runs on the JaguarPF machine that, while now decommissioned,
539 had the same hardware configuration as kraken. Thus we believe these CAM execution
540 times are a reasonable estimate for execution times on kraken. We decided to use 224
541 processes in the CAM scaling analysis because this would represent a typical cluster size
542 for academic runs of CAM, fully realizing that CAM can be run on a much larger number
543 of processors.

544 The “Data Points” column in Table 3 represents the number of CAM grid nodes
545 that intersect the SWAT model domain. These values were determined by creating
546 grids of 1.0, 0.5, and 0.25 degree resolutions, and then using spatial operations within a
547 Geographic Information System (GIS) to count the number of grid nodes within 50 km
548 of the watershed boundaries. Assuming a 5 Megabits per second (Mbps) data transfer
549 rate, 30 minute time step (therefore 48 data transfers per day), and the data packet sizes
550 discussed in Section 2.3.3, we arrived at the data transfer times. We note that the 5
551 Mbps was used as a typical network rate for a DSL network, which is where much of this
552 prototyping effort was performed. Many factors other than model scale could affect the
553 network bandwidth, but since the transfer times were minimal compared to the model
554 processing times, we felt that a more detailed analysis of the network rates would not be
555 useful for this effort.

556 The results show that CAM dominates the total execution time for all hydrologic re-
557 gions included in the scaling analysis. For the case of running SWAT for the Southeastern
558 region and CAM at a 1.0 degree resolution, SWAT execution time is still approximately
559 half of the CAM execution time. For the Carolinas, data transfer time for a 0.25 degree
560 resolution CAM model is close to the magnitude of the SWAT model execution time.
561 These data provide an approximate measure of the relative influence of model execution
562 time and data transfer time as a function of hydrologic study area and atmospheric model
563 resolution. As we noted before, there is the potential to influence these base numbers by,
564 for example, exploiting opportunities to parallelize the hydrology model or to compress
565 data transfers. However we note from these results that, because CAM dominates the
566 total execution time for regional-scale hydrologic systems, the increased time required
567 for data communication between the CAM and SWAT model via Web Services does not
568 rule out the approach as a feasible means for model coupling at a regional-spatial scale.

569 **4. Summary, Conclusions, and Future Work**

570 The Hydro-Climate testbed we prototyped is an example of a multi-scale modeling
571 system using heterogeneous computing resources and spanning distinct communities.
572 Both SWAT and CAM were initialized and run, and data were transmitted on request
573 between SWAT, implemented in OpenMI, and CAM, implemented in ESMF, via ESMF

Table 3: The estimated total execution time for the coupled model simulation for difference sized land surface units. The *Data Points* value is the number of lat/lon points in the grid that are exchange points with the land surface unit (assumes 50 km buffer around land surface area). Data transfer times are estimated based on the number of exchange points, model time step, and size of data communication packets.

(a) Upper Neuse Watershed								
Resolution (degree)	Data Points (count)	Execution Time per Day (sec)				Execution Time (hrs)		
		SWAT	CAM	Data Transfer	Total	1 yr	2 yr	5 yr
1	3	0.14	24	0.02	24.2	2.4	4.9	12.2
0.5	13	0.14	192	0.08	192.2	19.5	39.0	97.4
0.25	55	0.14	1536	0.33	1536.5	155.8	311.6	778.9

(b) Neuse River Basin								
Resolution (degree)	Data Points (count)	Execution Time per Day (sec)				Execution Time (hrs)		
		SWAT	CAM	Data Transfer	Total	1 yr	2 yr	5 yr
1	5	0.25	24	0.03	24.3	2.5	4.9	12.3
0.5	23	0.25	192	0.14	192.4	19.5	39.0	97.5
0.25	95	0.25	1536	0.56	1536.8	155.8	311.6	779.1

(c) The Carolinas								
Resolution (degree)	Data Points (count)	Execution Time per Day (sec)				Execution Time (hrs)		
		SWAT	CAM	Data Transfer	Total	1 yr	2 yr	5 yr
1	37	3.8	24	0.22	28.0	2.8	5.7	14.2
0.5	154	3.8	192	0.91	196.7	19.9	39.9	99.7
0.25	612	3.8	1536	3.59	1543.4	156.5	313.0	782.4

(d) Southeastern United States								
Resolution (degree)	Data Points (count)	Execution Time per Day (sec)				Execution Time (hrs)		
		SWAT	CAM	Data Transfer	Total	1 yr	2 yr	5 yr
1	96	12.3	24	0.59	36.9	3.7	7.5	18.7
0.5	387	12.3	192	2.27	206.6	20.9	41.9	104.7
0.25	1550	12.3	1536	26 9.09	1557.4	157.9	315.8	789.5

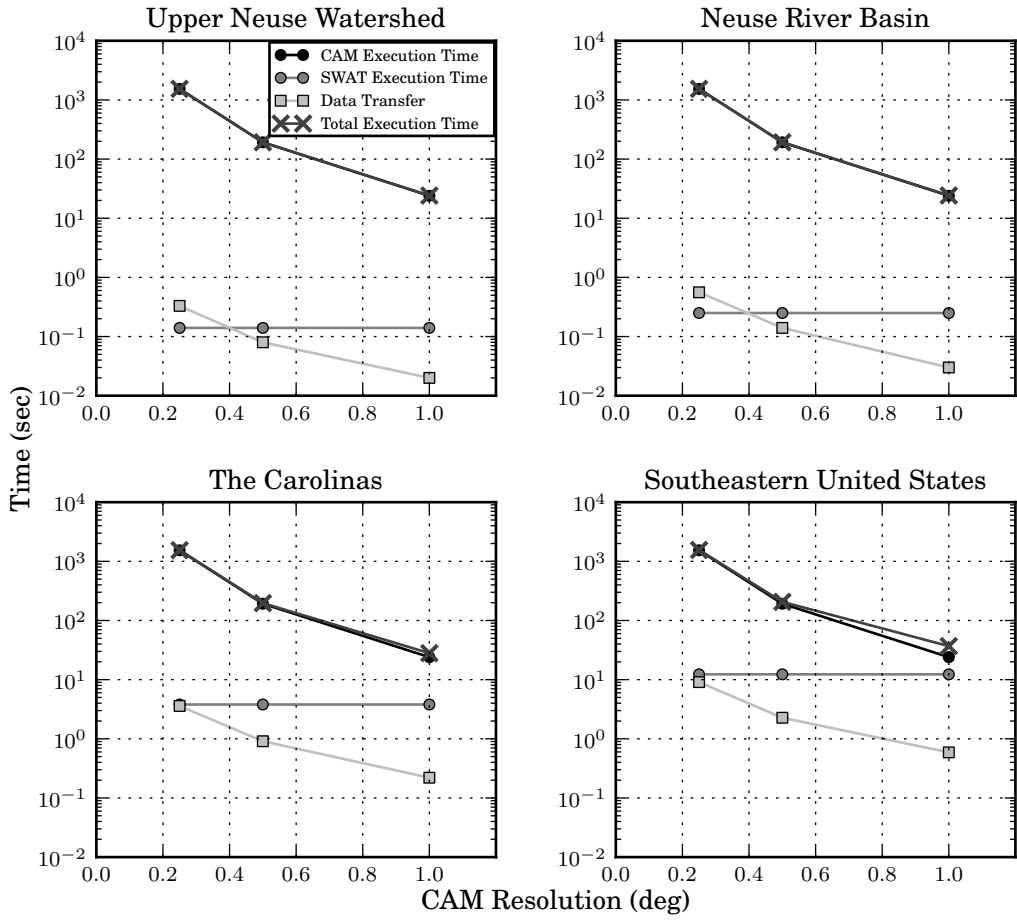


Figure 6: Results of the scaling analysis showing the time allocated to CAM and SWAT execution compare to data transfers using the Web Service coupling framework across different sized hydrologic units for SWAT and different spatial resolutions for CAM.

574 Web Services. One important result of this work is a demonstration of interoperability
 575 between two modeling interface standards: OpenMI and ESMF. These frameworks were
 576 created and used in diverse communities, so the design and development of the standards
 577 were not coordinated. Web Services proved to be a successful approach for coupling the
 578 two models. A second important result is a technical solution for coupling models running
 579 on very different types of computing systems, in this case a HPC platform and a PC.
 580 However, these results could be generalized to models running on, for example, two

581 different HPC platforms, or a model running on cloud-based services. The work required
582 to expose the HPC climate model Web Service interface highlighted the importance
583 of security policy and protocols, with many technical decisions based on the security
584 environment.

585 While we have with this work coupled computational environments with very differ-
586 ent characteristics, we have made no attempt at this point to either evaluate or exploit
587 strategies for parallelism in the hydrology model or across both modeling frameworks.
588 Our scale analysis, however, indicates the computational feasibility of our approach.
589 Currently a 0.25 degree resolution atmospheric model is considered high resolution and
590 such configurations are routinely run. At this resolution, the data transfer time and
591 SWAT computational time are approximately equal for an area the size of North and
592 South Carolina. We saw that SWAT execution time for an area the size of the South-
593 east U.S. was approximately half of the CAM execution time of the 1.0 degree CAM
594 configuration. If we run approximately 125 times the area of the Southeast U.S., the
595 computational times of SWAT and data transfer become comparable to that of CAM at
596 0.25 degrees. Assuming that a 0.25 degree atmospheric model is viable for research, then
597 with suitable strategies for parallelizing SWAT and compressing data transfer, we could
598 cover continental-scale areas with SWAT. Parallelism for SWAT is possible because if the
599 study area of each SWAT model is chosen wisely, no communication would be required
600 between the models dedicated to a particular area. The challenge comes if communica-
601 tion between the models is necessary to represent transfer, but recent work has begun to
602 address this challenge as well (Yalew et al., In Press).

603 Scientifically, we are interested in how the coupling between these two models of vastly
604 different scale impacts predictions of soil hydrology and atmospheric circulation. It is
605 well known that in the Southeast U.S. an important mechanism for precipitation is linked
606 to moisture flux from the Atlantic and the Gulf of Mexico. On a smaller scale, where
607 the Neuse River flows into Pamlico Sound the enhanced surface moisture flux is likely to
608 impact precipitation close to the bodies of water. Therefore, a logical next step in this
609 development is to build a configuration that might be of scientific interest in the sense
610 that we would be able to model impact of one system on the other. This would bring
611 focus not only to the computational aspects of the problem, but the physical consistency

612 of the parameters being passed between the models.

613 A less incremental developmental approach would be to consider regional atmospheric
614 models or regionalized global models. CAM was chosen for the initial development
615 because it is readily available, widely used, and has a sophisticated software environment
616 that was suitable. There are ESMF wrappers around all of the component models of
617 CESM, with the exception of the ice sheet model. Recently the regional Weather Research
618 and Forecasting Model (WRF) (Michalakes et al., 2001, 2004) was brought into the CESM
619 coupling environment (Vertenstein, 2012, pers. comm), creating a path to using WRF
620 with ESMF Web Services. With this advance, WRF can be brought as an alternative
621 atmosphere into the Hydro-Climate Modeling System, and work has begun in that regard.
622 Likewise, the coupling technology created for our research could support the integration
623 of other hydrological and impacts models, and models that use OpenMI with particular
624 ease. With this flexibility, we expect that the overall approach could be used to explore
625 a range of problems.

626 We have, here, demonstrated a Web Service-based approach to loosely couple models
627 operating close to their computational limits, looking toward a time when the temporal
628 and spatial scales of the models are increasingly convergent and the computational restric-
629 tions more relaxed. In addition, we have putatively coupled two discipline communities.
630 These communities have a large array of existing tools and scientific processes that define
631 how they conduct research. With such coupling we open up the possibility of accelerated
632 research at the interfaces and the support of new discoveries. In addition, we suggest the
633 possibility of more interactive coupling of different types of models, such as economic and
634 regional integrated assessment models. By controlling access to each model on a timestep
635 basis, we allow interactive reaction (via human or machine) and/or adjustment of model
636 control. Looking beyond basic scientific applications, we also suggest a new strategy for
637 more consistently and automatically (through the use of community standards and tools)
638 linking global climate models to the type and scale of models used by practitioners to
639 assess the impact of climate change and develop adaptation and mitigation strategies.

640 **Software Availability**

641 The code for this system and instructions to reproduce our results is available at
642 <http://esmfcontrib.cvs.sourceforge.net/viewvc/esmfcontrib/HydroInterop/>.

643 **Acknowledgments**

644 We thank Nancy Wilkins-Diehr of the San Diego Supercomputer Center for her sup-
645 port of this project and her assistance in gaining access to XSEDE resources. Suresh
646 Marru of Indiana University helped with the security and Web Services environment that
647 allowed the coupling to succeed. James Kent of the University of Michigan ran several
648 benchmarking experiments at the University's Flux Computational Environment. We
649 thank Andrew Gettelman and David Lawrence on the National Center for Atmospheric
650 Research for discussions of the scientific interfaces between the Community Atmosphere
651 Model and hydrological algorithms.

652 This work was supported by the NOAA Global Interoperability Program and the
653 NOAA Environmental Software Infrastructure and Interoperability group. This work
654 used the Extreme Science and Engineering Discovery Environment (XSEDE), which is
655 supported by National Science Foundation grant number OCI-1053575.

656 **References**

- 657 Arnold, J. G., Kiniry, J. R., Srinivasan, R., Williams, J. R., Haney, E. B., Neitsch, S. L., 2011. Soil and
658 Water Assessment Tool input/output file documentation (Version 2009).
659 URL <http://swatmodel.tamu.edu/media/19754/swat-io-2009.pdf>
- 660 Arnold, J. G., Allen, P. M., 1996. Estimating hydrologic budgets for three Illinois watersheds. *Journal*
661 *of Hydrology* 176 (1-4), 57–77.
- 662 Betrie, G. D., van Griensven, A., Mohamed, Y. A., Popescu, I., Mynett, A. E., Hummel, S., 2011. Linking
663 SWAT and SOBEK using Open Modeling Interface (OpenMI) for sediment transport simulation in
664 the Blue Nile River Basin. *Transactions of the ASABE* 54 (5), 1749–1757.
- 665 Briley, L. J., 2010. Configuring and running the SWAT model.
666 <http://www.waterbase.org/documents.html>.
- 667 Dennis, J., Fournier, A., Spatz, W. F., St-Cyr, A., Taylor, M. A., Thomas, S. J., Tufo, H., 2005.
668 High-resolution mesh convergence properties and parallel efficiency of a spectral element atmospheric
669 dynamical core. *International Journal of High Performance Computing Applications* 19 (3), 225–235.

670 Elag, M. and Goodall, J. L., 2011, Feedback loops and temporal misalignment in component-based
671 hydrologic modeling, *Water Resources Research* 47 (12), W12520.

672 Gassman, P. W., Reyes, M. R., Green, C. H., Arnold, J. G., 2007. The Soil and Water Assessment Tool:
673 Historical development, applications, and future research directions. *Transactions of the ASABE*
674 50 (4), 1211–1250.

675 Goodall, J. L., Robinson, B. F., and Castronova, A. M. 2011. Modeling water resource systems using a
676 service-oriented computing paradigm. *Environmental Modelling & Software*, 26 (5), 573-582.

677 Graham, L. P., Hagemann, S., Jaun, S., Beniston, M., 2007. On interpreting hydrological change from
678 regional climate models. *Climatic Change* 81 (1), 97–122.

679 Granell, C., Díaz, L., Gould, M., 2010, Service-oriented applications for environmental models: Reusable
680 geospatial services. *Environmental Modelling & Software* 25 (2), 182–198.

681 Gregersen, J. B., Gijbers, P. J. A., Westen, S. J. P., 2007. OpenMI: Open Modelling Interface. *Journal*
682 *of Hydroinformatics* 9 (3), 175.

683 Hill, C., DeLuca, C., Balaji, V., Suarez, M., da Silva, A., 2004. The architecture of the Earth System
684 Modeling Framework. *Computing in Science and Engineering* 6 (1), 18 – 28.

685 Jablonowski, C., Williamson, D. L., 2011. The pros and cons of diffusion, filters and fixers in atmospheric
686 general circulation models. *Numerical Techniques for Global Atmospheric Models*, 381-493.

687 Kent, J., Jablonowski, C., Whitehead, J. P., Rood, R. B., 2012. Assessing tracer transport algorithms and
688 the impact of vertical resolution in a finite-volume dynamical core. *Monthly Weather Review* (2012).

689 Laniak, G. F., Olchin, G., Goodall, J. L., Voinov, A., Hill, M., Glynn, P., Whelan, G., Geller, G., Quinn,
690 N., Blind, M., Peckham, S., Reaney, S., Gaber, N., Kennedy, R., and Hughes, A., 2012, Integrated
691 environmental modeling: A vision and roadmap for the future. *Environmental Modelling & Software*,
692 In Press and Available online 24 October 2012.

693 Lemos, M. C., Rood, R. B., 2010. Climate projections and their impact on policy and practice. *Wiley*
694 *Interdisciplinary Reviews: Climate Change*.

695 Leon, L. F., 2007. Step by step Geo-Processing and set-up of the required watershed data for MWSWAT
696 (MapWindow SWAT). <http://www.waterbase.org/documents.html>.

697 Malakar, P., Natarajan, V., Vadhiyar, S. S., 2011. Inst: An integrated steering framework for critical
698 weather applications. *Procedia Computer Science* 4 (0), 116 – 125, Proceedings of the International
699 Conference on Computational Science, ICCS 2011.
700 URL <http://www.sciencedirect.com/science/article/pii/S1877050911000718>

701 Michalakes, J., Chen, S., Dudhia, J., Hart, L., Klemp, J., Middlecoff, J., Skamarock, W., 2001. De-
702 velopment of a next generation regional weather research and forecast model. In: *Developments in*
703 *Teracomputing: Proceedings of the Ninth ECMWF Workshop on the use of high performance com-*
704 *puting in meteorology. Vol. 1. World Scientific, pp. 269–276.*

705 Michalakes, J., Dudhia, J., Gill, D., Henderson, T., Klemp, J., Skamarock, W., Wang, W., 2004. The
706 weather research and forecast model: Software architecture and performance. In: *Proceedings of the*
707 *11th ECMWF Workshop on the Use of High Performance Computing In Meteorology. Vol. 25. World*
708 *Scientific, p. 29.*

- 709 Mirin, A. A., Worley, P. H., 2012. Improving the performance scalability of the community atmosphere
710 model. *International Journal of High Performance Computing Applications* 26 (1), 17–30.
- 711 Neale, R. B., Gettelman, A., Park, S., Chen, C., Lauritzen, P. H., Williamson, D. L., 2010. Description of
712 the NCAR Community Atmospheric Model (CAM 5.0). Tech. rep., National Center for Atmospheric
713 Research NCAR Technical Note TN-486.
714 URL <http://www.cesm.ucar.edu/models/cesm1.0/cam/>
- 715 Parker, S. G., Miller, M., Hansen, C. D., Johnson, C. R., 1998. An integrated problem solving environ-
716 ment: The SCIRun computational steering system. In: *System Sciences, 1998.*, Proceedings of the
717 Thirty-First Hawaii International Conference on. Vol. 7. IEEE, pp. 147–156.
- 718 Parry, M. L., Canziani, O. F., Palutikof, J. P., van der Linden, P. J., Hanson, C. E., 2007. Contribution
719 of working group II to the fourth assessment report of the intergovernmental panel on climate change.
720 Assessment reports, Cambridge University Press, Cambridge, UK.
- 721 Raucher, R. S., 2011. The future of research on climate change impacts on water: A workshop focusing
722 on adaptation strategies and information needs. Tech. rep., Water Research Foundation.
- 723 Tompson, A. F. B., Falgout, R. D., Smith, S. G., Bosl, W. J., Ashby, S. F., 1998. Analysis of subsur-
724 face contaminant migration and remediation using high performance computing. *Advances in Water*
725 *Resources* 22 (3), 203–221.
- 726 Vertenstein Mariana , 2012. personal communication.
- 727 Wehner, M., Oliker, L., Shalf, J., 2008. Towards ultra-high resolution models of climate and weather.
728 *International Journal of High Performance Computing Applications* 22 (2), 149–165.
- 729 Worley, P. H., Drake, J. B., 2005. Performance portability in the physical parameterizations of the
730 Community Atmospheric Model. *International Journal of High Performance Computing Applications*
731 19 (3), 187–201.
- 732 Xinmin, Z., Ming, Z., Bingkai, S., Jianping, T., Yiqun, Z., Qijun, G., Zungang, Z., 2002. Simulations of a
733 hydrological model as coupled to a regional climate model. *Advances in Atmospheric Sciences* 20 (2),
734 227–236.
- 735 Yalew, S. van Griensven, A., Ray, N., Kokoszkiwicz, L., and Betrie, G. D., In Press, Distributed
736 computation of large scale SWAT models on the GRID. *Environmental Modelling & Software*.
- 737 Yong, B., LiLiang, R., LiHua, X., XiaoLi, Y., WanChang, Z., Xi, C., ShanHu, J., 2009. A study coupling
738 a large-scale hydrological model with a regional climate model. Proceedings of Symposium HS.2 at the
739 Joing IAHS & IAH Convention, Hyperabad, India, International Association of Hydrological Sciences
740 Publ. 333, 203–210.