Land Model Testbed: Accelerating Development, Benchmarking and Analysis of Land Surface Models

Sarat Sreepathi*, Min Xu†, Nathan Collier†, Jitendra Kumar‡, Jiafu Mao‡, and Forrest M. Hoffman†
*Computer Science and Mathematics Division,
Oak Ridge National Laboratory, Oak Ridge, TN USA
Email: sarat@ornl.gov
†Computational Sciences and Engineering Division
Oak Ridge National Laboratory, Oak Ridge, TN USA
Email: xum1@ornl.gov, collierno@ornl.gov, forrest@climatemodeling.org
‡Environmental Sciences Division
Oak Ridge National Laboratory, Oak Ridge, TN USA
Email: jkumar@climatemodeling.org, maoj@ornl.gov

***Abstract*—A Land Model Testbed (LMT), designed to provide a computational framework for systematically assessing model fidelity and supporting rapid development of complex multiscale models, offers a general-purpose workflow for conducting large ensemble simulations of multiple land surface models, post-processing large volumes of model output, and evaluating model results. It leverages existing tools for launching model simulations and the International Land Model Benchmarking (ILAMB) package for assessing model fidelity through comparison with best-available observational datasets. Increased complexity and proliferation of uncertain parameters in process representations in land surface models has driven the need for frequent and intensive testing and evaluating of models to quantify uncertainties and optimize parameters such that results are consistent with observations. The LMT described here meets these needs by providing tools to run thousands of ensemble simulations simultaneously and post-process their output files, by automating execution of an enhanced version of ILAMB with site-specific benchmarks and multivariate functional relationships, and by offering ensemble diagnostics and a customizable dashboard for displaying model performance metrics and associated graphics. We envision the LMT capabilities will serve as a foundational computational resource for a proposed user facility focused on terrestrial multiscale model–data integration.**

***Index Terms*—land surface models, model benchmarking, ILAMB, model testbed, machine learning**

I. Introduction

The Land Model Testbed (LMT)[1] was developed to provide a computational framework for systematically assessing model fidelity and supporting rapid development of complex multiscale models. Leveraging existing tools for launching model simulations and a well-established framework for model–data comparison, the LMT offers a general-purpose workflow for conducting large ensemble simulations of multiple land surface models (LSMs), post-processing large volumes of model output, and evaluating model results to support model benchmarking, uncertainty quantification, parameter estimation and analysis. The LMT computational infrastructure and workflow tools were designed and optimized for high performance computing (HPC) and cloud resources, and they provide an extensible and easy-to-use platform for rapid development and assessment of complex multiscale LSMs.

Process-based LSMs simulate the exchange of energy, water and carbon between the land surface and the atmosphere, and they incorporate biogeochemical and ecological processes as well as interactions with human systems. The complexity of LSMs has increased significantly since the first of such models, which had very simple representations of energy, mass and momentum transfer between the atmosphere and land [1]. LSMs have evolved to represent a large array of mechanistic processes, including soil moisture dynamics, photosynthesis, vegetation dynamics, carbon and nutrient cycling, fire, crops, land cover management and urban environments. While incremental addition of these processes over generations of LSMs has improved the accuracy of individual terrestrial process representations, it has also led to increased complexity and potentially to increased uncertainties [2]. Model uncertainty continues to be one of the biggest challenges in Earth system science, and it is not clear that increased model complexity reduces that uncertainty [3]. Increased complexity and the proliferation of uncertain parameters in process representations has driven the need for frequent and intensive testing and evaluating of models to quantify uncertainties and optimize parameters such that results are consistent with observations. To conduct the factorial simulations required, an efficient and scalable framework for executing simulations with perturbed parameters in parallel on HPC platforms is needed. In addition, to systematically evaluate and benchmark model results of ensemble simulations, the International Land Model Bench-

[1]Presented at Gateways 2020, Online, USA, October 12–23, 2020. https://osf.io/meetings/gateways2020/

marking (ILAMB) package provides, as a starting point, a set of internationally accepted metrics and observation-based data sets [4], [5]. The LMT outlined below was designed to meet these needs by providing tools to run thousands of ensemble simulations simultaneously and post-process their output files, by automating execution of an enhanced version of ILAMB with site-specific benchmarks and multivariate functional relationships, and by offering ensemble diagnostics and a customizable dashboard for displaying model performance metrics and associated graphics.

II. LMT Computational Infrastructure

The computational infrastructure of the LMT, summarized in Figure 1, shows that multiple land models can be executed by the workflow, which enables the design and conduct of simulation experiments on HPC platforms like Summit [6], the fastest supercomputer in the world. The LMT is presently deployed to facilitate execution of site-level, regional and global simulations, as well as large ensemble runs for rapid model benchmarking, uncertainty analysis, and model parameter optimization on the Summit supercomputer. Due to cybersecurity constraints with respect to web services and two-factor authentication requiring user intervention on Summit, the parameter generation, subsequent post-processing, analysis and model benchmarking are conducted on an institutional cloud server that can be scaled up as needed. Presently, our cloud instance is comprised of 40 compute cores (Intel Xeon CPU E5-4620 2.60 GHz), 248 GB of RAM and 1,240 GB of flash-based storage to successfully handle (see Section IV) large ensemble generation and big data processing workloads.

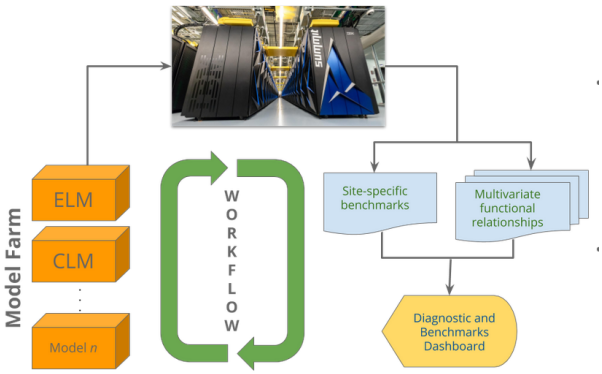

Fig. 1. The Land Model Testbed (LMT) provides a computational framework and workflow for executing complex multiscale models and assessing their fidelity to support uncertainty analysis, parameter optimization, and rapid model development.

A JupyterHub instance was configured and deployed on the LMT server to facilitate model output data processing, analysis and visualization. For example, as shown in Figure 2, through a Jupyter notebook and the ILAMB library, we can quickly define a model ensemble based on a regular expression pattern in the filenames, and then extract a time series of the gross primary productivity (GPP) for visualization (Figure 3), which includes an ensemble mean (solid line) and the standard deviation (grey shading). This interface allows for an interactive exploration of the model ensemble locally in a web browser, while the computation and data remains on the server.

```
lmt = ModelResult('path/to/output',name='LMT')
lmt.findFiles(group_regex='.*clm2_(.*)\.h0\..*')
gpps,mean_gpp = lmt.getVariable("GPP",mean=True)
fig,ax = plt.subplots(figsize=(18,3))
mean_gpp.convert("g m-2 d-1")
mean_gpp.plot(ax)
```

Fig. 2. This Python code was run in a Juypter notebook with the ILAMB library to post-process ensemble output for gross primary productivity.

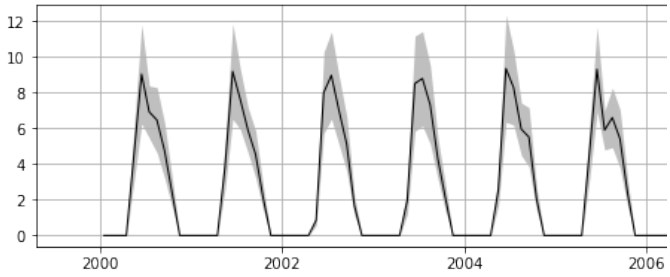

Fig. 3. The ILAMB library enables easy and interactive visualization and analysis of ensemble time series of gross primary productivity.

III. Capabilities

A. LMT Workflow

We established a reliable and effective workflow to implement the LMT capabilitis. It leverages the Offline Land Model Testbed (OLMT) [7] to launch model simulations, ILAMB [8] for comprehensive model–data comparison and benchmarking, and the Earth System Grid Federation (ESGF) and other data centers for archiving output. We built a containerized variant of the LMT software stack using Docker to encapsulate the workflow and the land models for easy deployment on HPC and cloud environments.

The LMT includes an interactive variable mapping web interface, a post-processing toolkit, a Python-based parallel lightweight CMOR-ization tool (to rewrite of model output in compliance with climate community standards), and a unified dashboard [9]. The variable mapping tool enables users to collaboratively add, modify, and update mapping relationships between model and CMOR variables requested by different Model Intercomparison Projects (MIPs) using a web browser. Users can save the mapping information as a JavaScript Object Notation (JSON) file locally or archive it in a repository for version control. The post-processing toolkit converts raw model output, including time serialization (aggregation of output to single variable files from all time records) and grid remapping, as necessary. Two options are provided for grid remapping: *ncremap* from the NCO toolkit [10] and an in-house remapping tool. Our remapping tool, designed with a performance-first objective utilizes SciPy sparse matrix methods and Numba Just-in-Time compilation capabilities for

better numerical performance, as well as parallel NetCDF for enhanced I/O performance The lightweight CMOR-ization tool uses the variable mapping information in the JSON file to rewrite the outputs in parallel after time serialization and grid remapping, following the MIP standard. The CMOR-ized outputs can be used by ILAMB directly for model intercomparison and be published to the ESGF and other data centers.

B. Perturbed Parameter Ensembles

The LMT infrastructure can rapidly launch thousands of ensemble simulations simultaneously while perturbing the parameter spaces of land models. Such perturbed parameter ensembles can be used to optimize model parameters and quantify model variability and uncertainties by comparing with in-situ measurements and observational datasets [11] and build artificial intelligence models using machine learning methods [12]. The parameters for ensemble simulations are generated using the Monte Carlo method, i.e., parameter selection through random sampling from a uniform distribution.

C. Ensemble analysis

We leveraged our collective experience in model benchmarking algorithms [5] to create an analysis framework for benchmarking model ensembles, which present several challenges. Due to the extensive computational and data requirements of the model, model ensemble simulations are often focused at select locations, where detailed observational records are available for forcing and benchmarking simulations. In addition, the simulation results from large ensemble members represent a large volume of high dimensional data, making their interpretation and analysis difficult with traditional methods. The benchmarking methodology and metrics in ILAMB were originally envisioned for comparison of long time series of global gridded models with observational records. We adapted and extended ILAMB protocols for site-level simulations of LSMs, focusing our analysis of simulation results on measures of bias and Root Mean Square Error (RMSE). For each model ensemble, we also compute and report the distributions of the bias, RMSE, bias score, RMSE score, and overall score (Figure 4). These metrics quantify the effects of perturbed parameters (in this case $flnr$, $slatop$, and $leafcn$) on benchmarking scores.

The ILAMB ensemble analysis leverages the Plotly [13] library, using the Javascript interface to embed interactive plots into a webpage (Figure 4). Individual ensemble members can be selected by a mouseover on the datapoints in any plot, which triggers an update of the text information in the middle panel, as well as updating the annual cycle plot in the bottom middle panel to highlight the ensemble member to which it corresponds.

D. Unified dashboard

An unified dashboard was designed to summarize key model benchmarking results and facilitate further interactive exploration of data. The dashboard provides a responsive web application, using HTML5, Cascading Style Sheets (CSS) and JavaScript (JS) front-end technologies. It extensively uses the jQuery and Tabulator JS libraries to implement various interactive features, including moving, hiding/showing table columns, expanding/collapsing nested benchmarking metrics, sorting and highlighting results, and tool tips.

The benchmarking results are typically multi-dimensional variables and could be treated as a function of region, metric, model and statistical score. The dashboard enables users to select and filter results in different dimensions by manipulating the JS object that is loaded from the JSON files generated by benchmarking software packages.

E. Machine learning analysis

To enhance the diagnostics capability of the LMT, we developed machine learning (ML)-based benchmarking workflow, mechanistically evaluating the LSM's performance in capturing the nonlinear and complex interactions between multivariate model variables. As a use case, we designed new relationship metrics employing major ML methods to benchmark the LMT-generated ensemble simulations against site measurements of key ecosystem variables (e.g., gross primary productivity, evapotranspiration, and sensible heat fluxes). Specifically, the lead-lag interactions among interested carbon, water and energy fluxes, and the responses of these fluxes to selected environmental drivers (e.g., temperature, precipitation and CO2 mole fraction) were systematically assessed by applying five ML algorithm variants including the random forests, support vector machines, artificial neural networks, least absolute shrinkage and selection operators, and gradient boosting machines, onto the flux observations and simulations. Each ensemble member was then evaluated in terms of the best-ML-based importance score, a semi-qualitative metric of relative importance of individual affecting factors derived from the ML techniques. Mechanistic agreement was then measured by the Spearman's rank correlations of the importance scores between observations and ensemble members. Finally, the overall performance of each ensemble simulation was defined by the square-root-average of the mechanistic rank R^2 and regular R^2 based on monthly model versus observed outputs. This ML-based evaluation framework can be further adapted to leverage current ILAMB to reinforce process-based quantification of model biases for high-dimensional large simulations.

IV. Performance

A. Background: Large Ensemble Simulations

During the first phase of this effort, we evaluated the capability of OLMT to setup, build, and submit large ensemble simulations; however, the OLMT did not scale efficiently for the large ensemble simulations, including perturbed parameter ensembles (see Section III-B). The OLMT launches ensemble simulations by cloning a base case and subsequently changing model parameters for each case, making it sub-optimal for generating even hundreds of ensembles.

The OLMT builds the model only for the base case and links the executable file to other cases to save model build

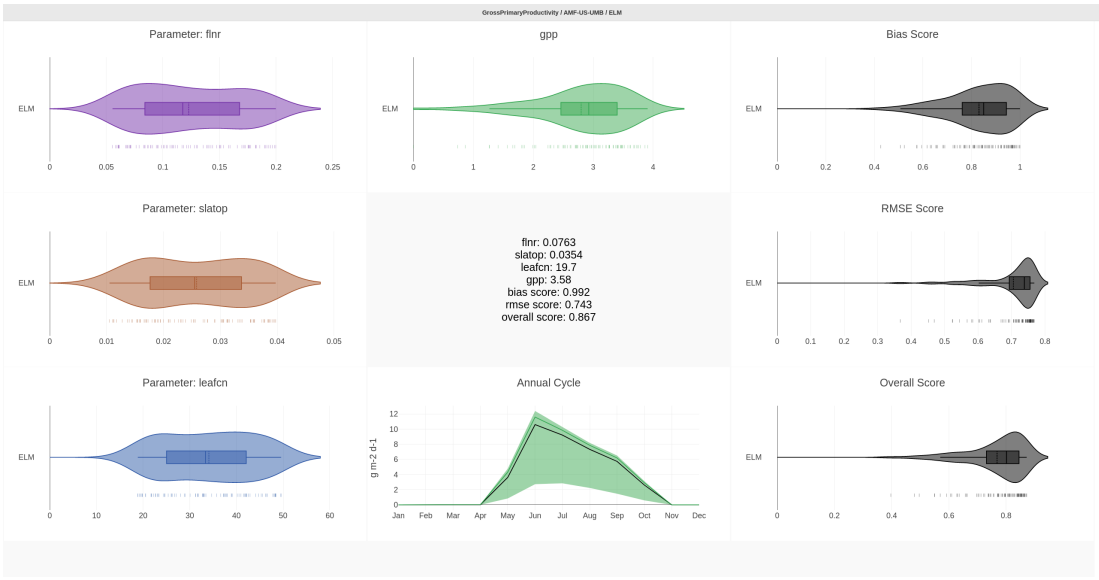

Fig. 4. Sample ensemble analysis output for a 84 member ensemble depicting the effect of varying the leaf nitrogen ratio (flnr), top specific leaf area (slatop) and the leaf carbon-nitrogen ration (leafcn) on the gross primary productivity (gpp) at the US-UMB Ameriflux flux tower.

time. Nevertheless, it takes a significant amount of time in the experiment setup and file copy stages. It bypasses the model's default framework that controls parallelism and job submission, relying instead on externally maintained scripts. Finally, the simulation results are spread across numerous directories, which makes post-processing cumbersome. Due to the aforementioned design and implementation choices, the execution of large ensembles using the OLMT is fragile, error-prone and computationally inefficient.

To overcome these limitations, we redesigned and implemented the perturbed parameter ensembles in the LMT infrastructure by directly using the multi-instance functionality within the core framework. The same common-core framework is employed by two large climate modeling efforts namely, the Energy Exascale Earth System Model and the Community Earth System Model and their respective land component models are used in the testbed. In this framework, the multi-instance functionality can enable execution of multiple simulations simultaneously in a single job; however, the framework is still unsuitable for large ensemble simulations because it generates thousands of model component and I/O namelists, and data stream description files in simulation experiment setup, build and submission steps. For example, if we run a 1000-instance simulation, it will generate more than 54 thousand files in the experiment directories during the setup, build and run phases. This is highly inefficient, taking more than 12 hours on Summit in the above scenario to finish generating the requisite files.

B. Workflow Optimization

To alleviate this bottleneck, we optimized the framework to rapidly launch several thousands of ensemble simulations by (1) eliminating the file generation during the experiment setup phase; (2) consolidating the numerous data stream description files by reusing a single file for the whole ensemble; (3) optimizing code associated with the multi-instance loops by refactoring loop invariant and lifting instance-irrelevant (unrelated to a specific ensemble instance) out of the loops; (4) multi-threading capability to generate requisite files in parallel; (5) mitigating file copies from the build and experiment directories to the run directory through direct generation of files in the run directory; and (6) eliminating file generation in the experiment submission step.

These optimizations resulted in a 72× improvement in the overall workflow performance, bringing the overall ensemble generation time from 12 hours to 10 minutes on a single node of Summit. Using the multi-instance capability, the LMT exhibits good weak-scaling characteristics on leadership computing systems like Summit as each ensemble instance is "embarrassingly parallel" and communicates only within its local group of processes.

V. Summary

The LMT was developed to provide computational infrastructure and workflow tools for execution and systematic assessment of complex multiscale models. The LMT enables performance of large ensemble simulations, post-processes large volumes of model output, and evaluates model results to study model parameter sensitivity and quantify model uncertainties. By integrating workflow tools with an enhanced model benchmarking package (based on ILAMB) and an interactive analysis and visualization platform, the LMT seeks to accelerate the model development cycle through rigorous assessment of model fidelity and analysis of simulation results. We envision the LMT capabilities will serve as a foundational computational resource for a proposed user facility focused on terrestrial multiscale model–data integration.

ACKNOWLEDGMENT

This research was sponsored by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC, for the U.S. Department of Energy under contract DE-AC05-00OR22725. This research used resources of the Compute and Data Environment for Science (CADES) and the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory. This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

REFERENCES

[1] P. J. Sellers, Y. Mintz, Y. C. Sud, and A. Dalcher, "A Simple Biosphere model (SIB) for use within general circulation models," *J. Atmos. Sci.*, vol. 43, no. 6, pp. 505–531, 1986.

[2] R. A. Fisher and C. D. Koven, "Perspectives on the future of land surface models and the challenges of representing complex terrestrial systems," *J. Adv. Model. Earth Sy.*, vol. 12, no. 4, p. e2018MS001453, 2020.

[3] K. Carslaw, L. Lee, L. Regayre, and J. Johnson, "Climate Models Are Uncertain, but We Can Do Something About It," *Eos Trans. AGU*, vol. 99, Feb. 2018.

[4] F. M. Hoffman, C. D. Koven, G. Keppel-Aleks, D. M. Lawrence, W. J. Riley, J. T. Randerson, A. Ahlström, G. Abramowitz, D. D. Baldocchi, M. J. Best, B. Bond-Lamberty, M. G. De Kauwe, A. S. Denning, A. R. Desai, V. Eyring, J. B. Fisher, R. A. Fisher, P. J. Gleckler, M. Huang, G. Hugelius, A. K. Jain, N. Y. Kiang, H. Kim, R. D. Koster, S. V. Kumar, H. Li, Y. Luo, J. Mao, N. G. McDowell, U. Mishra, P. R. Moorcroft, G. S. H. Pau, D. M. Ricciuto, K. Schaefer, C. R. Schwalm, S. P. Serbin, E. Shevliakova, A. G. Slater, J. Tang, M. Williams, J. Xia, C. Xu, R. Joseph, and D. Koch, "International Land Model Benchmarking (ILAMB) 2016 workshop report," U.S. Department of Energy, Office of Science, Germantown, Maryland, USA, Tech. Rep. DOE/SC-0186, Apr. 2017.

[5] N. Collier, F. M. Hoffman, D. M. Lawrence, G. Keppel-Aleks, C. D. Koven, W. J. Riley, M. Mu, and J. T. Randerson, "The International Land Model Benchmarking (ILAMB) system: Design, theory, and implementation," *J. Adv. Model. Earth Sy.*, vol. 10, no. 11, pp. 2731–2754, Nov. 2018.

[6] "Summit Supercomputer," https://www.olcf.ornl.gov/olcf-resources/compute-systems/summit/, 2020.

[7] "Offline Land Model Testbed," https://github.com/dmricciuto/OLMT, 2020.

[8] "International Land Model Benchmarking (ILAMB) repository," https://github.com/rubisco-sfa/ILAMB, 2020.

[9] "Land model testbed unified dashboard," https://github.com/climatemodeling/unified-dashboard, 2020.

[10] C. Zender, P. Vicente, Hmb1, D. L. Wang, Wenshanw, JeromeMao, Dywei, Filipe, I. Fernando, B. Couwenberg, Jedwards4b, J. Hegewald, O. Poplawski, J. Hamman, and H. Oliveira, "nco/nco: Delirium," Feb. 2020.

[11] D. Ricciuto, K. Sargsyan, and P. Thornton, "The Impact of Parametric Uncertainties on Biogeochemistry in the E3SM Land Model," *J. Adv. Model. Earth Sy.*, vol. 10, no. 2, pp. 297–319, 2018.

[12] G. J. Anderson and D. D. Lucas, "Machine Learning Predictions of a Multiresolution Climate Model Ensemble," *Geophys. Res. Lett.*, vol. 45, no. 9, pp. 4273–4280, 2018.

[13] C. Sievert, *Interactive Web-Based Data Visualization with R, plotly, and shiny*. Chapman and Hall/CRC, 2020. [Online]. Available: https://plotly-r.com