A Framework for Analyzing the Community Land Model within the Communty Earth System Models

D. Wang, T. Janjusic, W. Ding, F. Winkler,
N.Forrington, O. Hernandez
Climate Change Science Institute &
Computer Science and Mathematics Division
Oak Ridge National Lab
Oak, Ridge, TN, USA
{wangd,janjusict,dingw,winklerf,
forringtonnr,oscar}@ornl.gov

W. Wu, Y. Xu
Department of Computer Science &
Department of Geography
University of Tennessee
Knoxville, TN, USA
{wwu12,yxu30}@vols.utk.edu

***Abstract*— As environmental models (such as Accelerated Climate Model for Energy (ACME), Parallel Reactive Flow and Transport Model (PFLOTRAN), Arctic Terrestrial Simulator (ATS), etc.) became more and more complicated, we will need new tools to expedite integrated model developments and facilitate the collaborations between field scientists, environmental system modelers and computer scientists. In this poster, we present our methods and efforts to analyze the Community Land Model (CLM), a terrestrial ecosystem model within the Community Earth System Models (CESM)). Specifically, we demonstrate our objectives, methods and software tools to support interactive software structure exploration and automatic functional testing code generation, compiler analysis and interesting works on code porting preparations for pre-ExaScale computers. We believe that our experience on the environmental model, CLM, can be beneficial to many other scientific research programs which adapt the integrated, component-based modeling methodology on high-end computers.**

Keywords—Climate Modeling, Community Land Model, Functional Testing, Performance Analysis, Compiler Analysis, Visual Analytics

I. INTRODUCTION

As the environmental models (such as Acclearted Climate Model for Energy (ACME), Parallel Reactive Flow and Transport Model (PFloTran), Arctiv Terristrial Simulator (ATS), etc.) became more and more complicated, we will need new tools to expedite integrated model developments and facilitate the collaborations between field scientists, environmental system modelers and computer scientists. In the poster, we present a framework for analyzing the Community Land Model (CLM) within the Community Earth System Models. Over the past several decades, researchers have made significant progress in developing high fidelity earth system models to advance our understanding on earth system, and to improve our capability of better projecting future scenarios. The Community Earth System Model is one of the US leading earth system models. CESM is being actively developed under the "Accelarted Climate Model for Energy (ACME)" project to support Department of Energy's climate and environmental research. Within the CESM framework, the CLM is designed to understand how natural and human changes in ecosystem [1]. affect climate. The model represents several aspects of the land surface including surface heterogeneity and consists of submodels related to land biogeophysics, the hydrologic cycle, biogeochemistry, human dimensions, and ecosystem dynamics. Currently, the offlice CLM simulation system contains of more than 1800 source files and over 350,000 lines of source code. (The graphic presentation of CLM software structure is shown in Figure 1).It is well known that the software complexity of the Community Land Model becomes a barrier for rapid model improvements and validation, as well as efficient code porting to next generation HPCs [2,3].

The main objectives of our efforts shown in this poster include: 1) Create web-based interactive views for new module development via BER's MODEX approach; 2) Overcome software complexity barriers via automatic mutliscale ecosystem mechanistic modular design; 3) Develop advanced tools for code analysis and porting on next generation HPCs; and 4) Engage board user communities via web services and cloud computing. Specifically, this poster presents detailed information in following sections.

II. VISUAL ANALYTIC SYSTEM

A web-based visual analytic system with 3-tier architecture (data/middle/client) has been developed to support dynamic query and interactive visualization of CLM software structure. The data tier hosts the information of software structure extracted from CLM source code using Fortran-syntax analysis. The client tier was developed based on HTML5, JavaScript and open source libraries such as D3.js (d3js.org) and jQuery (jquery.com). The middle tier, which was implemented using Python, exchanges information between the data tier and the client tier. The system provides three main functionalities, which are CLM overview, query of functions/variables and submodel structure, shown in Figure 2. Users can search any function/variable in a particular CLM version, and see how it accesses or is accessed by other functions/variables. They can also explore the call tree of any subroutine or the interrelationships among different components at the submodel level. Dramatically improved from previous effort [4], the system provides much needed interface for software structure exploration, and further benefits model interpretation and new module development (URL: http://cem-base.ornl.gov/CLM_Web/CLM_Web.html).

III. AUTMATIC FUCNTIOANL TESTING SYSTEM

We have implemented an automatic ecosystem function testing system, which is able to extract a specific subroutine/module from CLM and to generate a standalone functional testing model for the given subroutine/module. Using this testing system, we have successfully tested most ecosystem modules, and it can be extended to all submodels in CLM or even CESM. This effort is a significant improvement for the functional test platform [5], which is designed to create direct linkages between site measurements and key ecosystem functions within CLM. It provides much needed integration interfaces for both field experimentalists and ecosystem modelers to improve the model's representation of ecosystem processes within the CESM framework without large software overhead. Figure 3 in the poster presents the general workflow for the automatic function unit testing module generation as well as two typical usages of the functional testing: user defined computational experiments configuration and observation-inspired computational experiments setup for direct model-data comparison.

IV. CODE ANALYSIS AND PORTING PREPERATIONS

Several efforts have been conducted to better understanding the CLM software performance, internal structure similarity as well as scientific data workflow. Specially, We also use a compiler-based porting tools [6] was used to analyze the internal data structure similarity to identify efficient ways to port CLM to new hybrid computing platforms. We used Geinpir [7], a memory cache performance tool to identify the memory access patterns of each CLM function. We also used Vampir Framework combined with TAU instrumentation to trace and profile the each CLM function [8] and explored the possibility to porting CLM on Accelerators using OpenACC [9]. The exemplar results of the analysis and porting preparations are shown in Figure 4 of the poster.

V. COMMUNITY ENGAGEMENT EFFORTS

We believe that better engagement with large and diverseuser communities (such as field scientist, observational datasets provider, modelers and computer scientists.), who have difference research focus and computing background, is the key to future CLM success. We have developed web-based services provide convenient, interactive interfaces for computational experiment configuration and results exploration. Moreover, a cyberinfrastructure is being developed within a could computing environemnt at ORNL (Comput and Data Envrionmnet for Science (CADES) cades.ornl.gov) to accelerate CLM development by further reducing the computational barriers, since the cloud computing environment will minimize user efforts on compiler and tool installation, library configuration, as well as databases setups.

VI. CONCLUSIONS

We have demonstrated our objectives, methods and software tools to support interactive CLM structure exploration and automatic CLM functional testing code generation, compiler analysis and other preliminary CLM code immigration preparations for pre-ExaScale computers. We believe that our experience in the design of software tools and modular modeling and testing platform for the environmental models can be beneficial to many other scientific research programs which adapt the integrated, component-based modeling methodology on high-end computers.

ACKNOWLEDGMENT

This research was partially funded by Terrestrial Ecosystem Sciences (TES) Program and Climate Sciences for Sustainable Energy Future (CSSEF) Program under the Biological and Environmental Research (BER) under the Office of Science of the U.S. Department of Energy (DOE). This research used resources of the Oak Ridge Leadership Computing Facility at Oak Ridge National Laboratory, which is managed by UT-Battelle LLC for the Department of Energy under contract DE-AC05-00OR22725.

REFERENCES

[1] Oleson, K., Lawrence, D., Gordon, B., Flanner, M., Kluzek, E., Peter, J., Levis, S., Swenson, S., Thornton, P., and Feddema J., 2010, Technical description of version 4.0 of the Community Land Model (CLM).

[2] Wang, D., Post, W., Wilson, B., 2011. Climate Change Modeling: Computational Opportunities and Challenges, IEEE Computing in Science and Engineering, Vol 13(5), pp36-42

[3] Wang, D., Schuchart, J., Janjusic, T., Winkler, F., and Xu, Y. 2014a. Toward Better Understanding of the Community Land Model within the Earth System Modeling Framework, International Conference on Computational Science, Cairns, Australia, 2014,

[4] Xu, Y., Wang, D., Janjusic, T., Xu, X., A Web-based Visual Analytic System for Understanding the Structure of Community Land Model, International Conference on Software Engineering Research and Practice, June, 2014.

[5] Wang, D., Xu, Y., Thornton, P., King, A., Gu, L., Steed, C., Schuchart, J., 2014b. A Functional Testing Platform for the Community Land Model, Environmental Modeling and Software, Vol. 55, pp25-31, 10.1016/j.envsoft.2014.01.015

[6] Ding, W., Hsu, C. H., Hernandez, O., Chapman, B., & Graham, R. (2013). KLONOS: Similarity-based planning tool support for porting scientific applications. *Concurrency and Computation: Practice and Experience*, *25*(8), 1072-1088

[7] Tomislav Janjusic, Christos Kartsaklis, Dali Wang, Scalability Analysis of Gleipnir: A Memory Tracing and Profiling Tool on Titan, Cray User Group, May, 2014, Switzerland

[8] Domke, J., Wang, D., Runtime Tracing of the Community Earth System Model: Feasibility Study and Benefits, 12th Workshop on Tools for Program Development and Analysis in Computational Science, Omaha, Nebraska, June 2012, procedia CS 9: pp1950-1958, 2012

[9] Wang, D., et al, Feasibility Study on Porting CLM onto Acclerations using OpenACC, in preparation.