Language and Library Support for Climate Data Applications[1]

Eric Van Wyk[†], Vipin Kumar[†], Michael Steinbach[†], Shyam Boriah[†], and Alok Choudhary[⋆]

[†]Department of Computer Science and Engineering
University of Minnesota
evw,kumar,steinbac@cs.umn.edu

[⋆]Department of Electrical Engineering and Computer Science
Northwestern University
choudhar@ece.northwestern.edu

Ecosystem scientists now have petabytes of data available for analysis; one source of such data is from Earth orbiting satellites. Effective analysis of this data can help us understand how the Earth's climate is changing, and determine factors that cause these changes, in turn, providing an opportunity for predicting and preventing future ecological problems by managing the ecology and health of our planet. For example, change detection algorithms [1] can be applied to the EVI time-series data to discover the loss of forest cover, and in turn help determine their impact on carbon emission.

Performing the necessary analysis on this data is difficult. Although spatio-temporal data sets can be analyzed at various scales, many phenomena of interest become accessible only at a finer scale, making it critical to develop capabilities for large-scale data analytics. For example, it is difficult to detect slow changes (such as logging) in land cover at coarse resolutions. But higher resolution data sets such as EVI have billions of data points just for one time instance, making change-point detection on a global scale extremely compute intensive.

Writing efficient, scalable, and portable data-intensive applications that deal with data on this scale is immensely challenging. In practice, programmers get bogged down in the low-level details of managing parallel processes and parallel file I/O and thus spend more time focused on performance issues than on the core computational problem. This significantly increases the time required to build these applications and in many cases it is so much of a burden that problems that scientists would like to address are not even implemented since it is too difficult to achieve their solutions within the time constraints.

We are beginning a new project to address these issues. Our central hypothesis is that an *extensible language framework*, backed by a rich and expressive collection of high-performance libraries, will provide an application development environment in which several domain- and application-specific *language extensions*, added to a *host* language such as C, allows programmers and scientists to more easily and directly specify solutions to data-intensive problems as programs written in domain-adapted languages. In our approach to extensible languages [3], language extensions provide new language constructs (syntax) based on notations from the domain along with semantic analysis and, most importantly, optimizations. For data-intensive applications, an extension may add language support for the notions of *map* and *reduce* found in MapReduce and optimizations over such constructs based on existing work in functional programming.

Libraries provide the computational building blocks for language extensions in the highly tuned implementations of the core operations, such as map or reduce. Programs written in domain-adapted languages are optimized at the domain-specific level and translated into implementations that utilize the high-performance APIs provided in libraries such as PnetCDF [2]. Thus, a key challenge in this effort is in building efficient and scalable library-based operations at the right level of abstraction.

Based on our past work in extensible languages, data-mining, and efficient parallel libraries we expect that an extensible language and library framework that targets the domain of data-intensive climate analysis and modeling applications can provide tools that ecosystem scientists can use to more effectively address the challenges of climate change.

References

[1] S. Boriah, V. Kumar, M. Steinbach, C. Potter, and S. Klooster. Land cover change detection: A case study. In *Proc. of the 14th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 857–865, 2008.

[2] J. Li, W. Liao, A. Choudhary, R. Ross, R. Thakur, W. Gropp, R. Latham, A. Siegel, B. Gallagher, and M. Zingale. Parallel netCDF: A scientific high-performance I/O interface. In *Proc. of the ACM/IEEE Conference on Supercomputing (SC)*, pages 39–49, November 2003.

[3] E. Van Wyk, L. Krishnan, A. Schwerdfeger, and D. Bodin. Attribute grammar-based language extensions for Java. In *Proc. of the European Conf. on Object Oriented Programming (ECOOP)*, volume 4609 of *LNCS*, pages 575–599, July 2007.

[1]This work is supported by the NSF under collaborative grants #0905581 and #0905205.