03-07 Dec 2012
Brasilia, Brazil

SCALABLE AND PROVENANCE-ENABLE SCIENTIFIC WORKFLOWS FOR PREDICTING DISTRIBUTION OF SPECIES

Gadelha Jr., L.M.R. 1
Stanzani, S.L. 2
Corrêa, P.L.P. 2
Dalcin, E. 3
Gomes, C.R.O. 1
Sato, L. 2
Siqueira, M.F. 3

1 Laboratório Nacional de Computação Científica, Brazil
2 Escola Politécnica, Universidade de São Paulo, Brazil
3 Jardim Botânico do Rio de Janeiro, Brazil

Session: Ecoinformatics tools for conservation of biodiversity (Chair: Susan P. Worner, Lincoln University, New Zealand)
7th December 2012, room 2, 12h20-12h40 (ISEI-92)

Abstract

Biodiversity data is becoming increasingly available and its volume is growing rapidly. Integration of different related repositories is also advancing through a number of successful initiatives. This data can be given as input to sophisticated computer models for predicting potential distribution of species. As the amounts of manipulated data increase, the execution of these models require more powerful computational resources and the use of parallel and distributed computing techniques to become scalable. At the same time, it is increasingly difficult to manage simulations given by various computational tasks that explore, for instance, different modeling algorithms or different parameter configurations. In this work we explore applications of scalable and provenance-enabled scientific workflow techniques to ecological niche modeling.

1 Introduction

Scientific data is being produced at an exponential growth rate by increasingly available scientific sensors. This, coupled with sophisticated computational models that consume this data, has demanded new techniques [4] for managing computational scientific experiments in a scalable way. These experiments are often given by many computational tasks that exchange data through production and consumption relationships, and are usually specified as scientific workflows [1]. Scientific workflow management systems provide features such as fault-tolerance, scalable execution, scalable data management, data dependency tracking, and provenance recording, that greatly reduce the complexity of managing the life-cycle of these experiments. Provenance [1], in particular, can support the analysis of the outcome of a computational scientific experiment since it records the history of its execution. Biodiversity follows the same trend of rapidly increasing production of data. Currently, biodiversity data is being integrated in a global scale through initiatives such as the Global Biodiversity Information Facility (GBIF) [2]. Applications for the prediction of species' distribution, such as openModeller [5], use these biodiversity data sets along with environmental (like climatology) data to predict geographic distribution of a particular species, or of multiple species, at each geographic region. In this work, we describe different implementation strategies that can be used to scale the execution of openModeller using Swift [6], a scientific workflow management system that focuses on parallel and distributed execution of computational tasks. We also show how provenance recording, as provided in Swift by MTCProv [3], can enable scientific workflow execution analysis.

2 Scalable Management of openModeller Scientific Workflows

openModeller [5] is a framework that implements various tasks related to modeling the distribution of species. It implements different algorithms for species' distribution and supports a variety of data formats. It uses a correlative approach, where the ecological requirements of a species are inferred from the environmental characteristics of known ocurrence sites. The modeling process receives data sets about species' ocurrence points, environmental characteristics of the regions containing these points, and parameters for tuning the algorithms. It generates a model that associates environmental conditions with suitability for the existence of species. This model is applied to each cell of a geographic region to generate a potential distribution map. For implementing openModeller workflows we use Swift [6], a parallel scripting system that supports the specification, execution, and analysis of scientific workflows. Component applications of a scientific workflow can be executed transparently and concurrently on computational resources that use commonly used job schedulers and grid toolkits. The Swift language is used for specifying scientific workflows by defining the data sets that will be manipulated, and the application programs that will consume and produce these data sets. The language also supports conditional flow control and loop constructs, allowing for more complex data flows. The foreach construct, for instance, processes all elements of a collection in parallel. Parallelism is implicit and pervasive in Swift, all expressions whose data dependencies are met are evaluated in parallel. This convenient approach to achieve parallelism can be applied in different scenarios for improving the scalability of openModeller scientific workflows. For instance, one could concurrently experiment with the different niche modeling algorithms implemented in openModeller. If the input data is relative to different moments in time, one could also run the modeling concurrently for each time step. Finally, one could use geographic partitioning to produce independent input data sets that can be processed in parallel. The following Swift pseudo-code applies these approaches simultaneously:

```
foreach species in selected_species[] {
  foreach timestep in timesteps[] {
    foreach algorithm in selected_algorithms[] {
      foreach parameter in selected_parameters[] {
        run_openmodeller(species, timestep, algorithm, parameter, rasters); }}}}
```

This simple nested foreach construct in Swift would generate (#selected species) × (#timesteps) × (#selected algorithms) × (#selected parameters) computational tasks that would be executed in parallel on the available computational resources. In addition to scalability, another benefit of using Swift would be the recording of provenance information in a relational database, through the MTCProv [3] component. This allows for analyzing the execution of scientific workflow through queries for information such as data set derivations and parameter values.

3 Concluding Remarks

Since openModeller computational tasks are usually loosely coupled, there are many opportunities for scaling their execution through Swift, through its implicitly parallel

ISEI8 – 8th Internacional Conference on Ecological Informatics | 2012
1

language and its execution engine, that supports many parallel and distributed environments. Through querying of provenance records, the analysis of scientific workflow execution can be improved, avoiding manual examination of log files and output data sets. We are currently implementing openModeller scientific workflows in Swift for execution in high performance computing clusters, and plan to analyze the execution performance of different parallelization strategies. We also plan to evaluate their ease of management through scientific workflow and provenance management techniques.

References

[1] E. Deelman, D. Gannon, M. Shields, and I. Taylor. Workflows and e-science: An overview of workflow system features and capabilities. Future Generation Computer Systems, 25(5):528–540, 2009.

[2] J. L. Edwards. Research and societal benefits of the global biodiversity information facility. BioScience, 54(6):486–487, June 2004.

[3] L. Gadelha, M. Wilde, M. Mattoso, and I. Foster. MTCProv: a practical provenance query framework for many-task scientific computing. Distributed and Parallel Databases, 30(5-6):351–370, 2012.

[4] T. Hey, S. Tansley, and K. Tolle, editors. The Fourth Paradigm: Data-Intensive Scientific Discovery. Microsoft Research, 2009.

[5] M. Muñoz, R. De Giovanni, M. de Siqueira, T. Sutton, P. Brewer, R. Pereira, D. Canhos, and V. Canhos. openModeller: a generic approach to species' potential distribution modelling. GeoInformatica, 15(1):111–135, 2011.

[6] M. Wilde, M. Hategan, J. Wozniak, B. Clifford, D. Katz, and I. Foster. Swift: A language for distributed parallel scripting. Parallel Computing, 37(9):634–652, 2011.

ISEI8 – 8th Internacional Conference on Ecological Informatics | 2012

ISEI
2012

Eighth Internacional Conference on Ecological Informatics

Informing decisions on biodiversity and natural resources conservation

03 December 2012

1