Parallel Computing 40 (2014) 589–610

Contents lists available at ScienceDirect

Parallel Computing

journal homepage: www.elsevier.com/locate/parco

Region templates: Data representation and management for high-throughput image analysis

George Teodoro [a,*], Tony Pan [c], Tahsin Kurc [b,d], Jun Kong [c], Lee Cooper [c], Scott Klasky [d], Joel Saltz [b]

[a] *Department of Computer Science, University of Brasília, Brasília, DF, Brazil*
[b] *Biomedical Informatics Department, Stony Brook University, Stony Brook, NY, USA*
[c] *Biomedical Informatics Department, Emory University, Atlanta, GA, USA*
[d] *Scientific Data Group, Oak Ridge National Laboratory, Oak Ridge, TN, USA*

ARTICLE INFO

Article history:
Received 28 August 2013
Received in revised form 23 September 2014
Accepted 24 September 2014
Available online 2 October 2014

Keywords:
GPGPU
Storage and I/O
Heterogeneous environments
Image analysis
Microscopy imaging

ABSTRACT

We introduce a region template abstraction and framework for the efficient storage, management and processing of common data types in analysis of large datasets of high resolution images on clusters of hybrid computing nodes. The region template abstraction provides a generic container template for common data structures, such as points, arrays, regions, and object sets, within a spatial and temporal bounding box. It allows for different data management strategies and I/O implementations, while providing a homogeneous, unified interface to applications for data storage and retrieval. A region template application is represented as a hierarchical dataflow in which each computing stage may be represented as another dataflow of finer-grain tasks. The execution of the application is coordinated by a runtime system that implements optimizations for hybrid machines, including performance-aware scheduling for maximizing the utilization of computing devices and techniques to reduce the impact of data transfers between CPUs and GPUs. An experimental evaluation on a state-of-the-art hybrid cluster using a microscopy imaging application shows that the abstraction adds negligible overhead (about 3%) and achieves good scalability and high data transfer rates. Optimizations in a high speed disk based storage implementation of the abstraction to support asynchronous data transfers and computation result in an application performance gain of about 1.13×. Finally, a processing rate of 11,730 4K × 4K tiles per minute was achieved for the microscopy imaging application on a cluster with 100 nodes (300 GPUs and 1200 CPU cores). This computation rate enables studies with very large datasets.

1. Introduction

Distributed-memory computing systems consisting of multi-core CPUs and general purpose Graphics Processing Units (GPUs) provide large memory space and processing capacity for scientific computations. Leveraging these hybrid systems, however, is challenging because of multiple memory hierarchies and the different computation characteristics of CPUs and GPUs. Application developers have to deal with mapping and scheduling analysis operations onto multiple computing nodes and, on a node, onto CPU cores and GPUs, while enforcing dependencies between operations. They also need to

[*] Corresponding author.
E-mail addresses: teodoro@cic.unb.br (G. Teodoro), tony.pan@emory.edu (T. Pan), tahsin.kurc@stonybrook.edu (T. Kurc), jun.kong@emory.edu (J. Kong), lee.cooper@emory.edu (L. Cooper), klasky@ornl.org (S. Klasky), joel.saltz@stonybrookmedicine.edu (J. Saltz).

http://dx.doi.org/10.1016/j.parco.2014.09.003
0167-8191/© 2014 The Authors. Published by Elsevier B.V.

implement mechanisms to stage, distribute, and manage large volumes of data and large numbers of data elements across memory/storage hierarchies.

We have developed an analytics framework for on-demand, high throughput processing of very large microscopy image datasets on hybrid computation systems [63,61]. The analytics framework consists of a library of high performance data analysis methods, data structures and methods common in microscopy image analyses, and a middleware system. We reported in an earlier work on methods implemented in the middleware system for scheduling data analysis operations and analysis pipelines on hybrid machines [63,61], In this paper, we investigate efficient data representations and runtime support to minimize data management overheads of common data types consumed and produced in an analysis pipeline.

The primary motivation for this effort is the quantitative characterization of disease morphology at the sub-cellular scale using large numbers of whole slide tissue images (WSIs). This is an important and computationally expensive application in biomedical research. Investigations of tissue morphology using WSI data (also referred to here as microscopy image data) have huge potential to lead to a much more sophisticated understanding of disease subtypes and feature distributions and to enable novel methods for classification of disease state. Biomedical researchers are now able to capture a highly detailed image of a whole slide tissue in a few minutes using state-of-the-art microscopy scanners. These devices are becoming more widely available at lower price points, making it feasible for research groups and organizations to collect large numbers of whole slide tissue images (WSIs) in human and animal studies The Cancer Genome Atlas[1] project, for instance, has more than 40,000 WSIs and counting. We expect that in the next 3–5 years, research groups will be able to collect tens of thousands of digital microscopy images per study, and healthcare institutions will have repositories containing millions of images. Over the past several years, a number of research groups, including our own, have developed and demonstrated a rich set of methods for carrying out quantitative microscopy image analyses and their applications in research [26,51,37,20,45,25,15,16]. Scaling such analyses to large numbers of images (and patients) creates high end computing and big data challenges. A typical analysis of a single image of $10^5 \times 10^5$ pixel resolution involves extraction of millions of micro-anatomic structures and computation of 10–100 features per structure. This process may take several hours on a workstation.

Our earlier work has demonstrated that large numbers of images can be processed rapidly using distributed memory hybrid systems by carefully scheduling analysis operations across and within nodes in the system and that scheduling decisions can be pushed to the middleware layer, relieving the application developer of implementing complex, application-specific scheduling mechanisms. The work presented in this paper introduces a data management abstraction layer, referred to here as the region template framework, for management and staging of data during the execution of an analysis application and shows that the overhead of such an abstraction is small. Our contributions can be summarized as follows:

- A novel region template abstraction to minimize data management overheads of common data types in large scale WSI analysis. The region template provides a generic container template for common data structures, such as points, arrays, regions, and object sets, within a spatial and temporal bounding box. A *data region* object is a storage materialization of the data types and stores the data elements in the region contained by a region template instance. A region template instance may have multiple data regions. The region template abstraction allows for different I/O, storage, and management strategies and implementations, while providing a homogeneous, unified interface to the application developer.
- An efficient runtime middleware to support the definition, materialization, and management of region templates and data regions and execution of analysis pipelines using region templates on distributed-memory hybrid machines. Application operations interact with data regions and region templates to store and retrieve data elements, rather than explicitly handling the management, staging, and distribution of the data elements. This responsibility is pushed to the runtime system. Presently, the runtime system has implementations for memory storage on nodes with multi-core CPUs and GPUs, distributed memory storage, and high bandwidth disk I/O.
- An experimental evaluation of the region template framework on a distributed memory cluster system in which each compute node has 2 6-core CPUs and 3 NVIDIA GPUs. The results demonstrate that this level of abstraction is highly scalable and adds negligible overhead (about 3%).

The rest of the paper is organized as follows. Section 2 provides an overview of the motivating scenario, and the use-case application for integrative microscopy image analysis. The region template framework is described in Section 3. Implementation of global region templates data storage, which are used for inter-machine data exchange is discussed in Section 4. Section 5 presents an experimental performance evaluation of the region template framework. The related work and conclusions are presented in Sections 6 and 7.

2. Background

2.1. Motivation

While our work is primarily motivated by studies that make use of morphological information from tissue specimens, these studies belong to a larger class of scientific applications that analyze and mine low-dimensional, spatio-temporal data

[1] http://cancergenome.nih.gov.