Parallel Computing 46 (2015) 78–97

Contents lists available at ScienceDirect

Parallel Computing

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

Performance modeling for hierarchical graph partitioning in heterogeneous multi-core environment

Siew Yin Chan [a,*], Teck Chaw Ling [a], Eric Aubanel [b]

[a] Department of Computer System and Technology, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia
[b] Faculty of Computer Science, University of New Brunswick, Fredericton, N.B., Canada

ARTICLE INFO

Article history:
Received 28 January 2013
Received in revised form 6 April 2014
Accepted 7 May 2014
Available online 19 May 2014

Keywords:
Performance model
Benchmark
Graph partitioning
Memory hierarchy
Multicore processing

ABSTRACT

Considering application behavior in graph partitioning is an arduous task because of the chicken-and-egg problem: the application behavior depends on how the graph is decomposed while achieving load balance requires the knowledge of how the application utilizes the underlying resources. Advances in multi-core processors further complicate the endeavor by introducing hardware diversity and intra-node contention. As an attempt to quantify performance for partitioning refinement, we propose a model that predicts execution times of iterative mesh-based applications running on heterogeneous multi-core clusters. Apart from considering resource heterogeneity, the model takes into account hierarchical communication characteristics, overlap between computation and communication, as well as performance penalties due to intra-node contention. We present a detailed methodology on how to obtain key parameters from a real system and highlight potential pitfalls of conventional approaches in obtaining the parameters. Experiments were conducted using a synthetic application benchmark solving a partial differential equation. Evaluation shows a good agreement between actual time measurement and the performance model.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Over the decades, graph partitioning has been applied in a wide range of applications including parallel scientific computations [1–3], VLSI design [4–6], image rendering [7,8], and database storage [9–11]. One of the most common applications of graph partitioning is the finite element method (FEM) to support numerical analysis. FEM describes physical phenomena of interest on bounded domains and is used to solve partial differential equations (PDEs). The domain is discretized into a finite element (FE) mesh and the PDE is transformed to a linear system which is solved using iterative methods. Accurate approximation to the original problem often requires many iterations and a huge number of elements. Given p_n processors with identical capacity, the FE mesh is partitioned into p_n subdomains such that every processor is assigned an equal number of elements. Each processor executes the same FEM code on its own subdomain. For elements situated at the subdomain boundary, dependencies on external elements necessitate information exchange with neighboring processors. Apart from balancing workload, graph partitioning ensures that these communication costs are kept minimal so as to reduce total execution time.

While conventional graph-partitioning algorithms perform adequately on homogeneous CPUs, the recent advances in parallel architectures poses new challenges from multiple dimensions. On one hand, the high variation in processor capacity

* Corresponding author. Tel.: +60 379676390.
E-mail addresses: siewyin_chan@nrl.fsktm.um.edu.my (S.Y. Chan), tchaw@um.edu.my (T.C. Ling), aubanel@unb.ca (E. Aubanel).

http://dx.doi.org/10.1016/j.parco.2014.05.001
0167-8191/© 2014 Elsevier B.V. All rights reserved.

causes computational powers (speed of computation and memory access) to differ significantly. Inferring the computational powers prior to graph partitioning, however, is not straightforward because the *actual* processor capacity also depends on how the FEM application exercises the processor's subsystems. On the other, the emergence of multi-core machines further complicates the endeavor by introducing resource contention on the horizontal memory hierarchy. In addition to local area networks, communication now spans across various interconnects ranging from that within chip multiprocessors (CMPs), between CMPs, among compute nodes, and among clusters. The complex dependence on processor capacity and interconnect performance can prevent application developers from harnessing the power of multi-core processors. Thus, the interaction between workload and resources must be taken into account for more efficient parallelization. To accomplish this goal, one needs to understand the application requirements on target machines and form informative analysis for decision making.

Essentially, to achieve increased parallelism in FEM applications and the others that use graph partitioning involves addressing two questions:

1. How to characterize application requirements in the presence of resource heterogeneity and contention?
2. How to use such information to guide graph-partitioning algorithms?

This paper focuses on Question 1 – a critical issue which is mandatory for workload-aware graph partitioning and application parallelization. One way to answer the question is through an analytical model which predicts the performance of applications prior to or during graph partitioning. Although much work has been undertaken, the proposed metrics [12] are formulated based purely on the application side and are independent of the hardware platform. Meanwhile, some existing models disregard the impact of processor heterogeneity and contention on multi-core systems [13,14]. In this paper, we propose a resource- and contention-aware model for a *hypercluster*[1] of heterogeneous multi-core processors. In this environment, the multi-core machines are grouped into clusters and are interconnected via a hierarchy of networks with varied capacities. Since processes compete with one another for shared resources, we agree with Zhong et al. [13] that time measurement is a better candidate than a combination of parameters to induce the effects of intra-node (within and between CMPs) contention. Our model predicts execution times of the FEM application based on system properties and application characteristics – a white-box approach adopted at Los Alamos National Laboratory [16]. It is only with predicted information available that the performance model may be used to help explore Question 2.

1.1. Motivation

The chicken-and-egg phenomenon between system and application raises the need to instill the awareness of application behavior into graph partitioning. A typical approach "adds" together seemingly important parameters such as computational speed, network capacity, and memory bandwidth. In spite of simplicity, the combination of parameters takes little consideration of collective impact as a whole. This is particularly true for multi-core processors when processes interfere with one another for shared resources. Nevertheless, quantifying the contention effect of applications with different memory-access patterns based on this assumption may prove futile in the absence of judicious investigation.

Essentially, the major obstacle to understanding application behavior is the unknown correlation between parameters. Further, the correlation is most likely machine and application specific due to inter-dependency between hardware properties and application characteristics. An alternative strategy to describe this complex scenario is via performance modeling [16]. Coupled with time measurement, a performance model that predicts execution time offers an invaluable tool for consolidating the impact of all parameters. We opt for this approach in this paper. To acknowledge resource contention on multicores, we design a contention-aware benchmark methodology which estimates the model parameters more accurately than simple benchmarking. The influence of non-blocking communication between neighboring processes is also considered to better support the interpretation of performance characteristics.

1.2. Contribution

The contribution of this work is as follows. We develop a model to estimate the execution time of an FEM application running on small-scale clusters of heterogeneous shared-memory multi-core processors. Such a model is invaluable for performance analysis throughout a system's life cycle [16]. Lastovetsky and Reddy [17], for instance, used their models to optimize the performance of compute-intensive kernels on heterogeneous multi-core platforms. The type of applications on which we focus here is FEM which involves data dependencies between vertices of irregular meshes. Hence, our model can also be generalized to stencil-based applications implementing finite difference method and finite volume method.

To achieve higher prediction accuracy of our model, we make the assumptions that (*i*) communication routines are implemented as asynchronous non-blocking operations, (*ii*) there exists overlap of computation and communication in FEM applications, as well as overlap of communications between levels in memory hierarchy, and (*iii*) network and memory contentions are non-negligible. Note that these assumptions differ from those commonly made in the literature, which include synchronous communication, the absence of computation/communication overlap, and insignificant resource

[1] Cappello et al. [15] define a *hypercluster* as an aggregation of clusters into multi-level clusters of processors.

Download English Version:

https://daneshyari.com/en/article/523878

Download Persian Version:

https://daneshyari.com/article/523878

Daneshyari.com