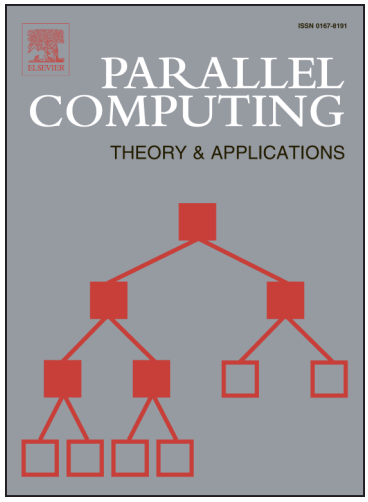

An Adaptive and Hierarchical Task Scheduling Scheme for Multi-core Clusters

Yizhuo Wang, Yang Zhang, Yan Su, Xiaojun Wang, Xu Chen, Weixing Ji, Feng Shi

PII: S0167-8191(14)00123-9
DOI: http://dx.doi.org/10.1016/j.parco.2014.09.012
Reference: PARCO 2216

To appear in: *Parallel Computing*

Received Date: 23 April 2014
Revised Date: 15 August 2014
Accepted Date: 4 September 2014

Please cite this article as: Y. Wang, Y. Zhang, Y. Su, X. Wang, X. Chen, W. Ji, F. Shi, An Adaptive and Hierarchical Task Scheduling Scheme for Multi-core Clusters, *Parallel Computing* (2014), doi: http://dx.doi.org/10.1016/j.parco.2014.09.012

An Adaptive and Hierarchical Task Scheduling Scheme for Multi-core Clusters

Yizhuo Wang[1], Yang Zhang[2], Yan Su[1], Xiaojun Wang[1], Xu Chen[1], Weixing Ji[1], and Feng Shi[1]

[1]*School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China*

[2]*School of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang, 050018, China*

E-mail: frankwyz@bit.edu.cn; zhangyang@hebust.edu.cn; {junestarvip, beiliwxjred, emailxuchen, bitjwx, bitsf}@bit.edu.cn

Abstract Work-stealing and work-sharing are two basic paradigms for dynamic task scheduling. This paper introduces an adaptive and hierarchical task scheduling scheme (AHS) for multi-core clusters, in which work-stealing and work-sharing are adaptively used to achieve load balancing.

Work-stealing has been widely used in task-based parallel programing languages and models, especially on shared memory systems. However, high inter-node communication costs hinder work-stealing from being directly performed on distributed memory systems. AHS addresses this issue with the following techniques: 1) initial partitioning, which reduces the inter-node task migrations; 2) hierarchical scheduling scheme, which performs work-stealing inside a node before going across the node boundary and adopts work-sharing to overlap computation and communication at the inter-node level; and 3) hierarchical and centralized control for inter-node task migration, which improves the efficiency of victim selection and termination detection.

We evaluated AHS and existing work-stealing schemes on a 16-nodes multi-core cluster. Experimental results show that AHS outperforms existing schemes by 11% to 21.4%, for the benchmarks studied in this paper.

Keywords work-stealing, multi-core cluster, task scheduling, work-sharing

1 Introduction

Today, most existing and new cluster systems are multi-core clusters, which present two levels of parallelism. One is shared memory parallelism within the cluster node. Another is distributed memory parallelism among the cluster nodes. How to exploit both shared and distributed memory parallelism is a critical issue to run a large application efficiently on such systems.

To exploit parallelism of the underlying architecture, the application needs to be parallelized with a particular form. Task parallelism is a popular form of parallelization of program. Most parallel programing languages and models are constructed based on task parallelism, such as Intel TBB[1], IBM X10[2], Microsoft TPL[3], OpenMP 3.0[4] and