

発表概要

遅延分割型負荷分散フレームワークの試験実装

大林 竜太[†] 平石 拓[†] 八杉 昌宏[†]
馬谷 誠二[†] 湯浅 太一[†]

本発表では、逐次プログラムに遅延分割、結果の結合の機能を加えたものを並列実行するための遅延分割型負荷分散フレームワークについて述べる。遅延分割とは、要求が生じた時点でタスクを分割するという事であり、スレッド生成なども事前には行わず、通常は逐次プログラムとして動作することで、低い並列化オーバーヘッドを実現できる。各ノードではノード内のタスクを管理し、他のノードとタスクなどをやりとりするための通信/スレッドライブラリを用いる。各ノードは、拠点サーバに接続して、拠点サーバを介して他のノードと通信を行う。また、担当しているタスクが終了したり、他のノードのタスクの結果を待って一時中断したりする場合は、ノード内でタスクを分割して融通したり、他のノードに要求を出し、新たなタスクを得ようとしたりする。要求を受けたノードは自分の処理中のタスクの一部を2つに分割し、片方を要求元へ送ってから、残りを自分で継続して処理する。このとき、できるだけ呼び出し元にさかのぼってのタスクの分割を実現するため、入れ子関数を利用した。逐次プログラムへ遅延分割、結果の結合機能を加える際の労力は、変換器を利用することで軽減する予定である。本研究は、共有メモリや分散システム等の多様な環境での動作を目指している。このフレームワークを試験的に実装した結果、各ノードに分割・結合時の通信コストと比較して十分な仕事量があれば、分散環境でも良好な台数効果が得られることが確認できた。

A Preliminary Implementation of a Load-balancing Framework Based on Lazy Partitioning

RYUTA OBAYASHI,[†] TASUKU HIRAISHI,[†] MASAHIRO YASUGI,[†]
SEIJI UMATANI[†] and TAIICHI YUASA[†]

In this presentation, we explain a load-balancing framework based on lazy partitioning; sequential programs augmented with capabilities for lazy partitioning and joining can be executed in parallel on this framework. Lazy partitioning means that tasks are partitioned only when a request has arisen. The usual execution of a sequential program without prior creation of tasks or any forms of threads involves quite low parallelization overhead. Each node uses a communication/thread library to supervise tasks in that node and exchange tasks and others with other nodes. Each node keeps a connection to a central server and communicates with other nodes via central servers. When a task on some node is finished or is suspended waiting for a result of another task on any other node, that node tries to get a new task by partitioning its other tasks or sending a task-request to some other node. The node which receives the request tries to partition its running task into two tasks then sends a new task to the requester and continues the remaining task. We employ nested functions for the task partitioning with backtracking to the caller as original as possible. We are planning to develop a translator for reducing the cost of augmenting sequential programs with capabilities for lazy partitioning and joining. Our goal is to construct a framework which works in various environments, such as shared-memory systems and distributed (Grid computing) systems. A preliminary implementation of this load-balancing framework exhibits good speedup even in a distributed environment if each node has a sufficient amount of work compared to the cost of exchanging messages for lazy partitioning and joining.

(平成 17 年 8 月 3 日発表)

[†] 京都大学大学院情報学研究科通信情報システム専攻

Department of Communications and Computer Engineering, Graduate School of Informatics, Kyoto University