

Using Virtual Grids to Simplify Application Scheduling

著者: Richard Huang, Henri Casanova, and Andrew A. Chien

出典: *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International, 25-29 April 2006, 10*

発表者: 所属: 本多・近藤研究室 学籍番号: 0953016 氏名: 西川優

1 概要

近年、グリッドアプリケーションのユーザや開発者はより多くのリソースを使う場面が多くなってきている。また、一般により多くのリソースがアプリケーションにより高い能力を発揮させようとする、それらはスケラビリティの問題を提起する。まず、最適なタスクの実行順序を決めるために複雑なスケジューリング手法を実装しても、スケジュールを計算するには法外な時間がかかるという可能性も考えられる。また、スケラブルな方法でスケジューリングアルゴリズムによって必要とされたすべてのリソース情報を集めるのは不可能ということがある。

本研究の取り組みはシンプルなスケジューリングアルゴリズムが将来の資源に富んだ環境で良い性能を達成できるということを目指している。本研究ではVG(Virtual Grids)[1]と呼ばれる単純化されたリソース環境を用いてスケラブルなスケジューリングアプローチを導入することでスケジューリングを単純化する。

2 現状のアプローチ

アプリケーションをタスク単位で分割した際、各タスクの依存性を表す図としてDAG(Directed Acyclic Graph)が挙げられる。以下の図1でDAGの一例を示す。

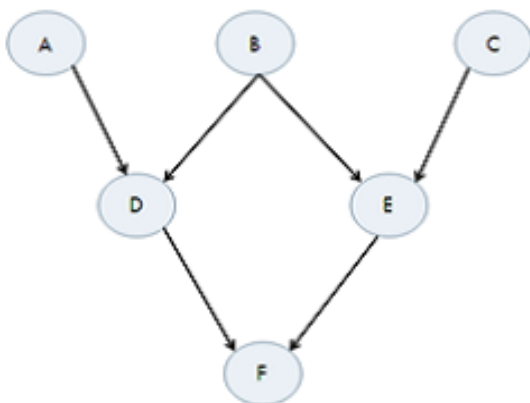


図1: DAGの構成図

DAGを用いることで各タスクの依存性を表すことができる。タスクの依存性を明確にすることでどのタスクを

先に実行すべきかを判断することができるようになる。DAGを用いたスケジューリングアルゴリズムとしてMCP(Modified Critical Path)が提案されている。

MCPは計算時間と通信時間を考慮に入れたクリティカルパスの中で最も速く実行できるタスクを実行するというアルゴリズムである。特徴はアプリケーションやリソースに関する詳しい情報を必要とすることである。また今までの研究でスケジュールを計算する時間と、タスクを実行する時間の間で良いトレードオフを達成するという明確なデモンストレーションがないということが挙げられる。

3 本研究のアプローチ

3.1 VGの導入

本研究では、VGを導入することでシンプルなスケジューリングアルゴリズムの効率化を可能にする。Virtual Gridsとはアプリケーションを実行するグリッド環境である。数多くあるリソースの中から適切なリソースを選んでグリッド環境を構築することで、スケジューリングの手間を省くことが狙いである。

VGを構築することで、膨大なリソースの中から即座に使用するリソースを選択することができるため、異なる種類のリソースが混在するような場合でも即座に実行環境が同じリソースを選択し、グリッド環境を構築することができる。また、選んだホストのパフォーマンスが出なかった場合は即座に別のホストが処理を代わるため、個々のリソースのトラブルに影響なく処理を終えることができる。

3.2 本研究のアルゴリズム

本研究で提案するアルゴリズムはタスクの依存性をはっきりさせたらすぐに処理の開始可能なホストに無作為に割り当てるだけというシンプルなアルゴリズムである。

4 実験と評価

4.1 モンタージュアプリケーション

本研究ではモンタージュアプリケーションを実行する。モンタージュは天文学に用いられる画像の合成手法であり、異なる角度から撮られた1つの被写体の距離や角度を計算することで、平面画像を立体としてとらえることが

できるアプリケーションである。以下の図 2 はモンタージュアプリケーションのタスクの依存性を表す図である。モンタージュアプリケーションはタスクの依存性が複雑であることが特徴であることから、スケジューリングの有用性を検証するには有効である。

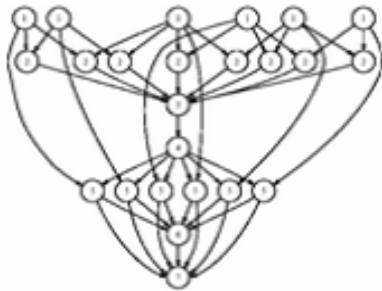


図 2: モンタージュのタスク依存性

実験結果は以下の図 3 のようになった。

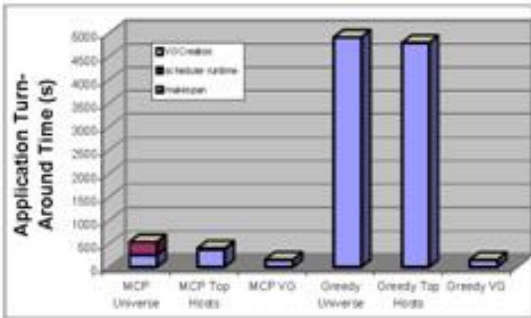


図 3: 通信コストと計算コストが等しい時の実行時間

図 3 から、VG 上で実行した場合、Greedy なスケジューリングでも MCP と変わらない時間で処理を終了させることに成功している。一方で TOP ホストのみで構成された場合や、リソース空間全体で実行された場合は Greedy アルゴリズムでは全くスケジューリングの効果が発揮されていないということがわかった。

4.2 Random DAG

本研究では様々なパラメータで実験を行うため、以下の表 1 の値に応じてランダムな DAG の集合を生成し、実験を行う。

各パラメータの説明は以下のようになる。

DAG size ..

タスクの数

CCR (Communication to Computation Ratio) ..

各タスクの通信コスト/計算コスト

Parallelism ..

DAG の幅 (1 の時すべてのタスクが並列に並ぶ)

表 1: 実験における各パラメータ

DAG Parameter	Values	Default Value
DAG size (tasks)	44, 447, 4469, 8938	4469
CCR	0.1, 0.2, 1, 2, 10	1
Parallelism	0.1, 0.2, 0.5, 0.8, 1	0.5

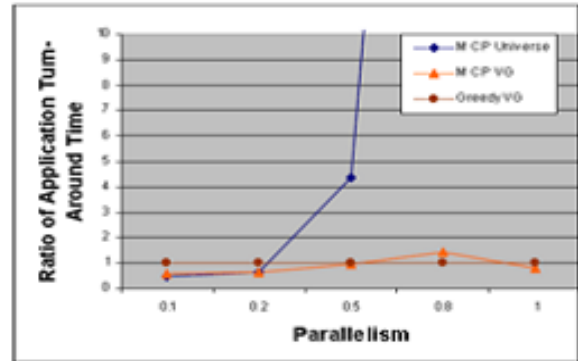


図 4: タスクの並列度を变化させた場合の実行時間の変化

図 4 の結果から、並列度が 0.5 以上の時、MCP と比べて Greedy の実行条件の方が処理時間が最大で 3 分の 2 倍に短縮したことがわかる。

5 まとめ

私たちは 2 つのスケジューリングアルゴリズムについて各パラメータを変化させて様々な実験を行った。それは結果的に VG 上で動作する greedy アルゴリズムは VG 上で動作する MCP と 3 % 以内の処理時間の違いで動作することができた。これらの実験結果から VG 環境下では greedy なアルゴリズムは MCP の代わりとして十分に機能すると考えられる。

参考文献

[1] Yang-Suk Kee, Dionysios Logothetis, Richard Huang, Henri Casanova, Andrew A. Chien " Efficient Resource Description and High Quality Selection for Virtual Grids " Cluster Computing and the Grid, 2005. CCGrid 2005. IEEE International Symposium on Publication Date: 9-12 May 2005 Volume: 1, On page(s): 598- 606 Vol.1