データ駆動型マルチプロセッサにおける分散スケジューリング機構の研究

1215095　福田 和馬　【 コンピュータ構成学研究室 】

A Study on Decentralized Hardware Scheduler for Data-Driven Multiprocessor

1215095　Kazuma FUKUDA　【 Advanced Computer Engineering Lab. 】

1　はじめに

近年の IoT (Internet of Things) デバイスは，様々な分野で使用されており，高機能化，高性能化が求められている．特に，ミッションクリティカルなシステムでは，マルチコア上でのリアルタイム処理が必要となる．

マルチコア上でのリアルタイム処理には，パーティショニング方式とグローバル方式がある．前者はタスク実行前に静的にタスク割り当てを決定するため，実行中の動的な負荷変動には対応できない．また，静的なタスク割り当ては NP 困難な問題である [1]．後者は，コアやタスクの実行状態に応じて動的にタスク割り当てを決定するため，スケジューリングオーバヘッドをゼロと考えると，高い稼働率を達成できる．しかし，実際にはスケジューリングオーバヘッドの大小で稼働率が左右される．また，コア数が増加するとオーバヘッドも増加し，スケーラビリティが低下するため，マルチコア化による性能向上には分散スケジューリングが必須となる．

本研究では，複数タスクを多重に処理可能なデータ駆動型プロセッサ DDP に着目し，DDP コア内のスケジューラと相互結合網内のスイッチ用スケジューラを統合した，分散スケジューリング機構 [2] を検討した．

2　設計方針

分散スケジューリングにおける主要なオーバヘッドには，マイグレーション，プリエンプション，スケジューラによるものがある．タスクのマイグレーションのためには，すべての候補コアの状態を集約する必要がある．これは，分散スケジューリングに反するため，本研究では，マイグレーションなしを前提とした．また，プリエンプションオーバヘッドを削減するためにコアに最小余裕時間優先 LST (Least Slack Time) スケジューリング方式を採用した DDP[3] を用い，スケジューラのオーバヘッドを削減するために，ネットワークによる分散スケジューリング機構を設計する．

本研究では，マルチコアのコア間ネットワークは図 1 に示す多段相互結合網で構成される．したがって，分散スケジューリング機構はネットワーク内の各スイッチモジュールが自律的に動作し，宛先を決定することで実現する．しかし，各スイッチがコアやタスクの実行状態を収集/分析すると，スケジューラのオーバヘッドが大きく，また，スケーラビリティが犠牲になる可能性がある．よって，各スイッチがローカルに持つ情報から宛先を決定することで，スケジューラ全体のオーバヘッドを小さくする．コアに直接接続されるスイッチは，過去のタスク割り当てからコアの状態を判断することができる．しかし，コアに直接接続されないスイッチはコアの状態を精密に予測できないため，過去の通過タスクの情報から経験的に宛先を決定する方針を採った．

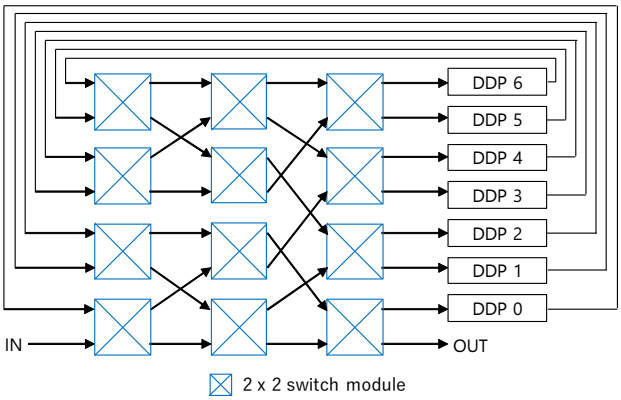

図 1　多段相互結合網の構成 (7 コアの場合)

3　分散スケジューリング機構

多段相互結合網を構成する各スイッチモジュールは，図 2 に示すようにセルフタイム型パイプライン (STP) を用いて構成した．STP は転送制御回路 (C 素子) 間のハンドシェイクによりタスクを転送するため，タスクが到着したスイッチのみが自律的に動作可能となる．以下に各スイッチのアルゴリズムを示す．

3.1　コアに直接接続されないスイッチ (図 3 (a))

タスク履歴ベースのアルゴリズムを採用した．各スイッチは履歴テーブルを持つ．タスクが到着すると，テーブルを参照し，同じタスク ID が記録されている場合は，記録されている宛先とは別の宛先にタスクを転送し，テーブルを更新する．テーブルには上限があり，到着タスクの稼働率がテーブル内の最小稼働率のタスクよりも大きい場合は到着タスクをテーブルに記録し，小さ

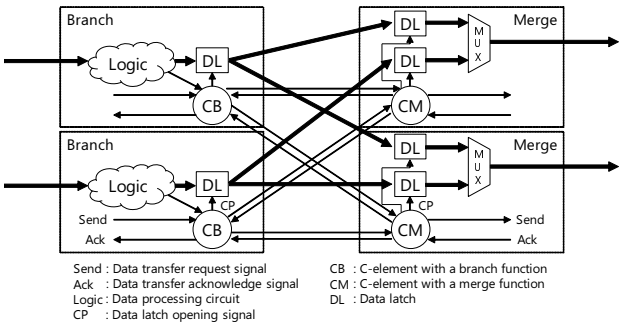

図 2　2 × 2 セルフタイム型スイッチモジュールの構成

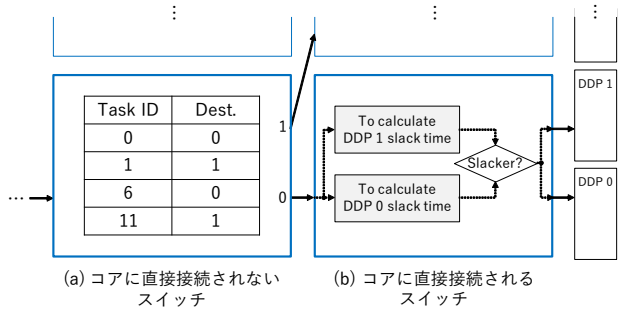

図 3　スイッチモジュールのアルゴリズム

い場合はラウンドロビンで割り当てる．これにより，稼働率の大きなタスクがあるコアに偏らないようにする．

3.2　コアに直接接続されるスイッチ (図 3 (b))

余裕時間ベースのアルゴリズムを採用した．スイッチにタスクが到着すると，到着タスクを各接続コアに割り当てたと仮定して，各接続先コアの余裕時間を計算する．計算した余裕時間を比較し，最も余裕のある (余裕時間の長い) コアに到着タスクを割り当てる．

4　評価

タスク履歴ベースのスイッチを 65nm CMOS 標準セルライブラリを用いて設計，論理合成した結果，回路面積は 0.0128mm^2 となった．また，スケジューリング性能評価のため，論理合成後の DDP の稼働率を 0.2% の誤差で見積もり可能なアーキテクチャシミュレータを作成した．システム稼働率が 60%になるタスクセットをランダムに作成し，100 試行のシミュレーションにより，平均システム稼働率とスケジューリング成功率を計測した．スケジューリング成功率は，(全タスクがスケジューリングに成功したタスクセット数) / (総タスクセット数) から算出する．

比較対象には，パーティショニング方式の LST (p-LST)，グローバル方式の EDF (Earliest Deadline First) (g-EDF) を用いる．g-EDF は，スケジューラが起動し，タスクスイッチが発生するたびに，実行状態の集約/配布オーバヘッド $\log_2 \mathrm{N} * \mathrm{weight}$ (N : コア数)，スケジューラオーバヘッド 1 (DDP1 命令実行時間) の時間分，タスク実行を中断する．また，提案方式の各スイッチは 4 個の履歴テーブルを持つ．

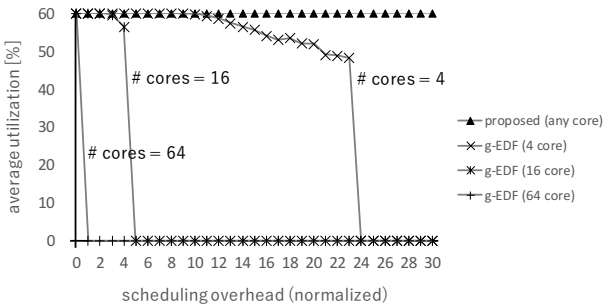

図 4　平均システム稼働率の比較

表 1　スケジューリング成功率 [%] の比較 (weight = 1)

コア数	2	4	8	16	32	64	128
proposed	99	96	95	90	80	71	58
g-EDF	100	100	100	100	100	0	0

p-LST は，スケジューリング成功率が 100%となったが，静的な方式であるため実用的には使えない．よって，g-EDF との比較結果を図 4，表 1 に示す．g-EDF は，コア数が増加するほどオーバヘッドが大きくなるため，weight = 1 の場合，64 コア以上では，平均システム稼働率，スケジューリング成功率ともに，提案方式が g-EDF に比べて高い結果となった．

5　おわりに

本研究では，データ駆動型マルチプロセッサにおける分散スケジューリング機構を検討した．提案方式は，DDP 内のスケジューラと，ローカル情報に基づき，自律的に宛先を決定可能な相互結合網内のスイッチ用スケジューラにより実現できる．オーバヘッドを考慮した場合は，コア数が多くなると，g-EDF に比べてスケジューリング性能が高いことを回路設計およびシミュレーションにより確認できた．今後は，動的な負荷変動を考慮した詳細なスケジューリング性能評価が必要である．

参考文献

[1] A. K. Singh, et al., “A Survey and Comparative Study of Hard and Soft Real-Time Dynamic Resource Allocation Strategies for Multi- or Many-Core Systems,” ACM Computing Surveys, Vol. 50, No. 2, pp. 1–40, Apr. 2017.

[2] K. Fukuda, et al., “Decentralized Hardware Scheduler for Self-Timed Data-Driven Multiprocessor,” PDPTA, pp. 256–261, July 2018.

[3] Y. Wada, et al., “Least Slack Time Hardware Scheduler Based on Self-Timed Data-Driven Processor,” PDPTA, pp. 249–255, July 2018.