スケジューリング手法の分類法

概要 ⊘

サーベイ資料をまとめる時や、手法について議論する際には「対象の問題」と「アルゴリズムのおおまかな特徴」が一目で分かると便利である。本ページではこの目的のために使用される表記法を説明する。本ページの表記法を使用することで例えば以下のように書ける。

paper	3-field	approach
2014_ECRTS [1]	$P \mid da\underline{g_i}, T_\underline{i=D_i}, prll_\{moldable\}, prec, pmtn, mlrt \mid CAB$	heuristic, online, federated

3-field notation ${\mathscr O}$

スケジューリング手法が解いている問題を曖昧性なく分類し、一目で分かるようにするために、Ronald L Graham によって three-field notation が提案された [2]。フォーマットは以下である。
Resource | Properties of tasks | Criterion

その後、この表記法は様々な著者によって拡張されている。ここでは Anna Minaeva 氏の three-field notation [3] を Atsushi Yano がさらに拡張したものをリストアップする。行の背景色は以下のように決められている。

- 白: Ronald L Graham 氏の分類 [2]
- 緑: Anna Minaeva 氏が追加した分類 [3]
- 黄色: Atsushi Yano が修正or追加した分類

Resource 🔗

Symbo Is	Descriptions
1	シングルコアプロセッサ
Р	ホモジニアスマルチコアプロセッサ
Q	Machines in parallel with different speeds. ヘテロジニアスマルチコアプロセッサであり、各マシンの速度がタスクとは無関係
R	Unrelated machines in parallel. ヘテロジニアスマルチコアプロセッサであり、各マシンの速度がタスクに依存して変わる
СР	クラスタ型ホモジニアスマルチコアプロセッサ
CQ	クラスタ型マルチコアプロセッサであり、同一クラスタの速度は均一だが、異なるクラスタ間では速度 が異なる

Properties of tasks 🔗

Symbols	Descriptions
r_i	リリースタイム・オフセットあり
T_i	周期・散発タスク
D_i	任意のデッドライン
T_i=D_i	暗黙的デッドライン
T_i<=D_i	制約付きデッドライン
D_{e2e}	エンドツーエンドデッドライン
dag_i	1つのタスクがDAGとして表される(タスク内 並列)
prll_{rigid}	タスク内並列あり、並列タスクに割り当てるプロセッサの数がスケジューラの外部から事前に指定され、実行中に変化しない
prll_{moldabl e}	タスク内並列あり、並列タスクに割り当てるプロセッサの数がスケジューラによって決定され 実行中に変化しない
prll_{malleabl e}	タスク内並列あり、実行中にスケジューラが並 列タスクに割り当てるプロセッサの数を変更で きる
gang	ギャングタスク
prec	同一周期のタスク間に優先順位制約がある
prec_{mr}	異なる周期のタスク間に優先順位制約がある
pmtn	常にプリエンプションが許可
pmtn_{mix}	特定のタスクのみプリエンプションが許可
pmtn_{cprtv}	特定のタイミングでのみプリエンプションが許 可

Criterion 🔗

Symbol s	Descriptions
-	最適化基準なし
C_{max}	メイクスパンの最小化
T_{max}	最大tardinessの最小化。ジョブのtardiness = max(0,開始時間+実行時間-相対デッドライン)
L_{max}	最大latenessの最小化。ジョブのlateness = 実行時間・相対デッドライン
R	リソース最小化
ctrl	制御性能の最適化
n_{pmtn}	プリエンプション数の最小化
energy	消費電力の最小化
ctrl	制御性能の最適化
DM	デッドラインミス率最小化
wcrt	WCRTの最小化
UB	Utilization Bound の最大化
CAB	Capacity Augumentation Bound の最大化

Approaches *∂*

スケジューリングアルゴリズムの特徴をおおまかに分類するための表記法

Symbols	Descriptions
heuristic	ヒューリスティックアルゴリズム
ILP	線形計画法による最適化
СР	制約プログラミングによる最適化

マイグレーションが許可されている

SMT	SMTyJu/~
fixed-priority	固定優先度
dynamic-priority	動的優先度
offline	事前にシミュレーションによってスケジュール (各タスクの開始時間・実行コア) を確定させ、実行時にはスケジュール通りにタスクを実行する
online	動的にスケジュールを決める
global	ジョブレベルでコアの移動を許可
partitioning	タスクが実行されるコアを固定
semi-partitioning	globalとpartitioningのハイブリッド
	例:コア集合をクラスタと呼ばれるサブセットに分割し、クラスタ内でのみコアの移動が許可
federated	フェデレートスケジューリング

参考文献 ⊘

- [1] Li, Jing and Chen, Jian Jia and Agrawal, Kunal and Lu, Chenyang and Gill, Chris and Saifullah, Abusayeed, "Analysis of federated and global scheduling for parallel real-time tasks," 26th Euromicro Conference on Real-Time Systems (ECRTS), 2014
- [2] Ronald L Graham, Eugene L Lawler, Jan Karel Lenstra, and AHG Rinnooy Kan. "Optimization and approximation in deterministic sequencing and scheduling: a survey." Annals of discrete mathematics, 5:287–326, 1979.
- [3] Minaeva, Anna and Hanz{\a}lek, Zden{\v{e}}k, "Survey on periodic scheduling for time-triggered hard real-time systems," ACM Computing Surveys (CSUR), 2021