

平成 16 年度 工学研究科 情報工学専攻 博士後期満了研究 発表会 要旨

依存グラフ再構成処理による並列処理タスク生成モデルの提案

工学研究科 情報工学専攻 内田 智士 (渡邊研究室)

近年, 計算機能力やネットワーク能力の向上に伴い, PC/WS クラスタ環境が注目を浴びている. スーパーコンピュータにおいても, クラスタ環境の導入により大規模な環境の構築が盛んに行われている. 同時に, 画像や動画などのマルチメディアデータを対象とする処理など, 数値計算以外にも膨大な計算を必要とするソフトウェアが増加し, 並列処理による速度向上の期待は高まっている. このような新たなドメインに対して効率よく並列処理するためには, 各環境やソフトウェアの特徴を考慮したうえで新たに並列プログラムを開発する必要がある. しかし, 並列プログラムの開発は長年の経験と知識を必要とし, 容易ではない. そこで, 逐次プログラムを並列プログラムへと変換する並列化コンパイラは, 並列プログラムの作成を容易にさせ, 開発効率の向上のための重要な役割を果たす. しかしながら, 並列化コンパイラの構築も, 通常の逐次コンパイラの処理に加え並列化処理を取り入れる必要があり, 開発が困難で膨大な手間を必要とする. そこで, 我々は並列化コンパイラの構築支援の開発を目指す.

並列化コンパイラ構築支援として, COINS や SUIF などはコンパイラ開発用インフラを提供する. しかし, これらのツールは異種環境におけるコード変換やステップ予測を提供する程度の支援にとどまり, 並列化処理を直接的に支援していない. また, 従来, 並列化コンパイラは並列タスク生成処理とタスクスケジューリング処理の問題に分けられて研究がなされてきた. 各処理に対して盛んに多くの研究が行われているが, それらの構築を支援するツールは存在せず, 各々の戦略を各自が実現する必要があった. そこで, 本研究は, 並列タスク生成処理の構築支援を目的とする.

我々は, 並列タスク生成処理を依存グラフの再構成処理としてモデル化し, 並列化戦略を再構成ルールとして定義する. コンピュータプログラムとは計算仕様書であり, データの加工とデータの授受で構成されている. この関係をグラフ表現しているのが依存グラフであり, プログラムの特徴を表現するのに適している. そこで, 依存グラフを基に並列タスク生成処理をモデル化する. 依存グラフのノードを処理単位と定義することで, 逐次プログラムと並列プログラムを表現することができる. 逐次プログラムは, 依存グラフ上の全ノードを 1 つのノードにまとめた形である. 並列プログラムは, ノードを結合, 分割することにより, 適度な処理粒度のノードに変形した形である. このとき, 互いに非依存なノードが並列実行可能となる. そして, 並列タスク生成処理は, 逐次プログラムから元の依存グラフを再構築し, さらに並列処理に適応するようにノードを再構成する処理といえる. また, 並列化知識は, 再構成のためのルールといえる. そこで, 我々は, 並列タスク生成処理を依存グラフの再構成に基づくルールベースシステムとしてモデル化する. これにより, 並列化コンパイラ開発者は, 自らの戦略を再構成ルールとして表現するのみで並列タスク生成機構を容易に実現することができる.

最初に, 再構成ルールの表現法を定義する. 条件部は制約付き依存グラフパターンで, 操作部は基本再構成操作のセットで表現する. 条件部には, 並列化知識の対象となるプログラム断片を検出するための条件が表現される. 依存グラフ再構成モデルにおいては, 依存グラフ上のサブグラフを抽出することにあたる. そこで, グラフパターン表現により抽出するサブグラフを特定する. グラフパターンの各ノードやエッジに対して制約を設定することにより, 更なる制約を設ける. 操作部には, 並列化知識の本質であるプログラムの構成方法が表現される. 依存グラフ再構成モデルにおいては, 抽出された依存グラフの再構成方法にあたる. 我々は, 再構成のための必要な操作として, 結合, 分割, 複製, 消去の 4 つの基本再構成操作と再構成途中を識別させるためのラベリング操作

を定義する．これらを組み合わせることにより複雑な操作を表現することができる．このように表現される再構成ルールをさらに組み合わせることで並列化戦略を自由に変更することが可能となる．

再構成ルール適用による問題として，再構成ルール適用時に生じる再構成ルールの競合と，適用後に生じるデッドロックを引き起こす依存関係の生成とがある．次に，これらの解決手法を説明する．

再構成ルールの競合問題は，優先度の導入と再構成操作の振舞い定義により解決する．異種ルールにおいては，優先度を導入することで解決できる．通常，並列化知識間においては，その洗練さによって一意に順序付けすることができる．すなわち，洗練された知識の方が，より高い並列性を抽出することができる．そこで，再構成ルールに優先度を設けることで，その適応順序を一意に決定付ける．しかし，同種ルールにおいては，同一優先度となり競合問題は解決されない．そこで，我々は，再構成ルールに用いられた再構成操作の振舞いを定義することで解決を図る．例えば，結合操作を利用している場合は，細粒度ノードをまとめることによって，粒度を大きくすることが狙いである．そこで，競合している結合操作が対象としている全ノードを結合対象として1ノードにまとめる．同様に，分割，消去，ラベリング操作は捜査対象に対して1度の振舞いを行い，複製操作は競合していないと捉える．

デッドロックを引き起こす依存関係とは，再構成されたノード間の依存関係が環状になる場合である．我々は，このような環状依存関係を LOOP 依存関係と呼ぶ．LOOP 依存関係は，それを構成しているノードの一部を分割することで解消される．再構成された結果は並列化コンパイラ開発者の意図が反映された結果である．そこで，依存グラフ構成への影響を少なくすることで，その意図への影響を少なくする．すなわち，分割対象としては，全体の構成への影響を少なくするため，多くの LOOP 依存関係を解決可能なノードから選出する．もし，LOOP 依存関係の解決可能数が等しい場合は，分割後に細粒度ノードを生成しないノードを選出する．これは，再構成された粒度にまとめたいという意図を反映するためである．

依存グラフ再構成モデルで，容易に並列タスク生成機構を実現可能であるが，ツールとして提供する場合，インタフェースも重要な役割を果たす．そこで，我々は，XML に基づく並列化知識インタフェースを提供する．前述の再構成ルールを XML 表現することにより，その取扱いを容易化している．また XML 表現により，XML オーサリングツールや XML DBMS などと提携することが可能となり，容易に再構成ルールの記述や管理を行うことができる．同時に，我々は，再構成ルールエディタを開発した．これは，グラフパターンを CAD のように描写により記述可能にし，各制約も入力ウィンドウを用意することで，記述を容易にしている．

我々は，本モデルを実装したプロトタイプシステムにより，並列化知識の記述の容易性とその適用効果について評価している．多くの並列化知識は依存グラフの再構成として表現することができ，さらに，各並列化知識も数ルール/手法，十数タグ/ルールで表現することができた．更に，数種の再構成ルールを NPB (NAS Parallel Benchmark programs) を適用することで，その効果を確認した．並列化ルールは数十箇所，基本ブロック作成ルールは数百箇所適用され，数十～数百個の並列タスクを生成できた．これにより，再構成ルールの表現は容易であり，プログラムに対して適切に適用され並列性を抽出しているといえる．