配列の縮退表現による大規模並列タスクネットワークの実装効率化

大野 和彦† 張 鉄群† 佐々木 敬泰† 近藤 利夫† 中島 浩‡
†三重大学工学部情報工学科 ‡京都大学学術情報メディアセンター

1 はじめに

我々は，粗粒度の大規模並列処理を目的としたスクリプト言語 MegaScript を提案し [1]，研究を進めている．MegaScript は多数のタスクからなるタスクネットワークを記述する言語であるため，このタスクネットワーク構造を保持するためのメモリ消費量や，タスクスケジューリングに必要な計算コストが，実行性能を大きく左右する．

そこで本研究では，同種タスクの一括生成に使われるタスク配列を縮退表現で実装することにより，タスク配列の消費メモリやスケジューリングコストを削減する手法を提案する．

2 背景

MegaScript は既存言語で記述されたプログラムをタスクとし，各タスクに与えるパラメータやタスク間のデータフローを表すタスクネットワーク，タスクの計算・通信コストを表すメタプログラムを，スクリプトで明示的に記述する．一方で，各ホストに対するタスクのスケジューリングや実行時のプロセス管理などは処理系が自動的に行う．これにより，並列処理を記述するユーザの負担軽減と実行性能の確保を，両立させることを目指している．

MegaScript は Ruby [2] をベースとするオブジェクト指向言語である．ユーザは個々のタスクを表すタスクオブジェクトを生成して，このオブジェクトのメソッド呼び出しにより，パラメータやネットワーク構造の設定を行う．処理系はこれらのオブジェクトを操作して，実行制御を行う．したがって基本的には，扱うタスクと同数のオブジェクトを生成・操作する必要がある．

そこでユーザの利便性を考え，異なるパラメータを持つ同種のタスクを多数生成する場合を想定したタスク配列クラスを用意している．タスク配列は要素であるタスク群に対してまとめて操作を行うことができ，配列の大きさにかかわらず一定のコード量で記述できる．

Reducible Array for Efficient Implementation of Massively-Parallel Task Networks.
†Kazuhiko OHNO †Chou TETSUGUN †Takahiro SASAKI †Toshio KONDO ‡Hiroshi NAKASHIMA
†Department of Information Engineering, Mie University
‡Academic Center for Computing and Media Studies, Kyoto University

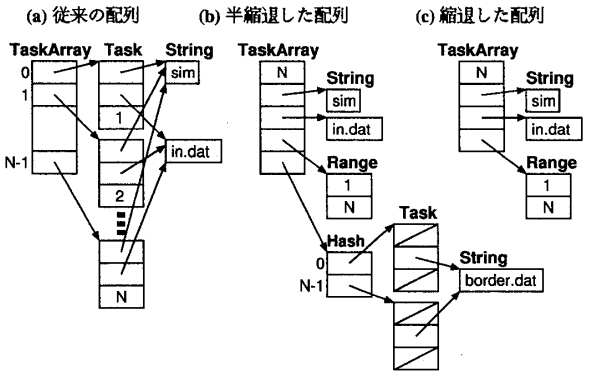

図 1: タスク配列の縮退実装

たとえばプログラム sim を用いて，入力データ in.dat に対し 1〜N のパラメータサーベイを行うためのタスク群は，以下のように一行で記述できる．

```
t = TaskArray.new(N, 'sim', 'in.dat', 1..N)
```

しかし従来の処理系では，そのままタスクオブジェクトの配列として実装しており (図 1(a))，消費メモリ量が配列の大きさに比例して増加する．このため，スクリプトを実行するマスターホストのメモリ量により，扱えるタスクネットワークの規模が制約されてしまう．また，処理系のスケジューラは個々のタスクオブジェクトに対し，計算・通信コストや依存を考慮して実行ホストや実行順を決定する．このため，タスク数が増えるとスケジューリングオーバヘッドが急増する．

3 提案手法

一般に同じタスク配列内のタスク群は共通する属性が多いため，図 1(c) のように配列サイズおよび属性を一つのオブジェクトで表す縮退表現を用いることで，メモリ消費を一定量に抑えることができる．しかしユーザが一部またはすべてのタスクに対し個別に属性を設定することもあり得るため，すべてのタスク配列を無条件に縮退表現で実装することはできない．かといって，縮退・非縮退のタスク配列を明示的に使い分けるのはユーザの負担となる．そこで，縮退・非縮退の状態をユーザに隠蔽し，処理系内で自動的に切り替えるような実装方式を提案する．

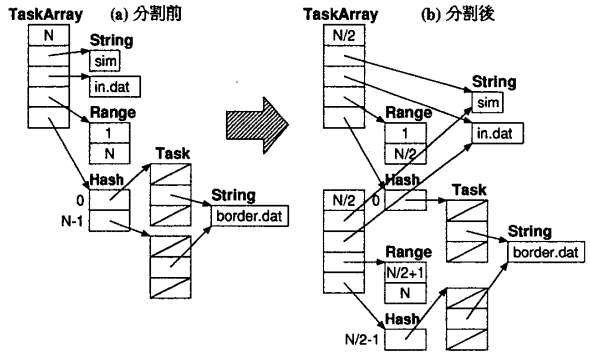

図 2: 縮退したタスク配列の分割

3.1 ユーザの操作に対する処理

本手法では以下のようにして，ユーザが記述する任意のコードに対して縮退状態を切り替える．Ruby にはメンバ変数がなく，オブジェクトの属性はすべてメソッドによりアクセスされるため，本手法はタスク配列クラスのコード変更のみで実装できる．

1. 図 1(c) の縮退状態で生成する．
2. 属性参照時は共通属性から該当する値を生成する．
3. 属性代入時は該当タスクのみオブジェクトを生成し，図 1(b) の半縮退状態に移行する．この状態では一部のタスクオブジェクトのみが独自の属性を保持しており，以後のタスク属性参照時は該当タスクがハッシュにあればその値を使用し，なければ状態 (c) と同様に共通属性から値を生成する．
4. 一定量のタスクが生成された時点で，図 1(a) の状態に移行し，タスクオブジェクトを配列で保持する．

これにより，ユーザが意識することなく，可能な場合には内部で縮退表現を用いることができる．サイズの大きいタスク配列の多くは，生成時にまとめてパラメータを設定し，個別の属性代入は境界条件など一部のタスクにしか行わない．したがって状態 (b), (c) にとどまり，メモリの消費量が大幅に削減できる．

3.2 スケジューリング時の処理

タスク配列を縮退したままスケジューリングを行うことで，オーバヘッドを大幅に削減できる．通常，タスク配列内のタスク群は外部に対して同じデータ依存を持つため，タスク群内の実行順序は性能に影響しない．したがって，スケジューリング時には配列全体を適切な計算コスト比で分割して割り振ればよく，個々のタスクオブジェクトにアクセスする必要がない．そこで以下の機能を実現した．

- (半) 縮退したタスク配列より，指定範囲内の計算コストの総和を求める
- (半) 縮退したタスク配列を指定位置で分割する

タスクの計算コストはメタプログラムより求められ，通常，実行時パラメータ等を変数とする積和や指数関数など比較的単純な式で表される．したがって，計算コスト式を積分公式で変形することにより，規則的に並ぶパラメータを持つタスク群の総コスト値は，定数時間で求められる．このため，N 個のタスクオブジェクトを一定のコスト比で分割する処理が $O(N)$ の時間がかかるのに対し，サイズ N の縮退したタスク配列ならば積分式と 2 分探索を組み合わせることで $O(\log_2 N)$ の時間で分割できる．

縮退したタスク配列の分割処理は，たとえば図 2(a) を 2 等分する場合であれば図 2(b) のように書き換えるだけでよく，分割処理のコストと分割後のメモリ消費量増加はわずかである．また，分割後の縮退タスク配列はそのまま実行ホストに送り，プロセス生成時に個々のタスク情報を取り出す．これによって，ホスト間でタスク情報を転送するときの通信量を削減できる，実行ホストでのメモリ消費量を減らすことができるといった利点も得られる．

4 おわりに

本稿では，並列タスクネットワークを記述する言語 MegaScript において，多数のタスクを効率的に扱うことのできる実装手法を述べた．現在，本手法を処理系上に実装中であり，今後，実プログラムによる性能評価を予定している．

謝辞

本研究の一部は文部科学省科学研究費補助金 (特定領域研究，研究課題番号 19024041,「高性能計算の高精度モデル化技術」) による．

参考文献

[1] 大塚保紀, 深野佑公, 西里一史, 大野和彦, 中島浩. タスク並列スクリプト言語 megascript の構想. 先進的計算基盤システムシンポジウム SACSIS2003, pp. 73–76, may 2003.

[2] まつもとゆきひろ, 石塚圭樹. オブジェクト指向スクリプト言語 Ruby. ASCII, 1999.