異機種並列分散コンピューティングのためのメタ・スケジューリングの構想

小出 洋* 武宮 博* 今村俊幸* 太田浩史* 川崎琢治* 樋口健二* 笠原博徳† 相川裕史*
†早稲田大学 理工学部 電気電子情報工学科　*日本原子力研究所 計算科学技術推進センター

2J－10

1 はじめに

本論文では，種々の並列計算機群をネットワーク接続した並列分散環境のもとで，事前あるいは実行時に収集した計算機やネットワークの負荷情報 (資源情報) を利用して，利用者，あるいはコンパイラが並列分散プログラム中に挿入したスケジューリングコードが，計算時間最小化を目的とし，実行時に動的負荷分散を行うメタ・スケジューリング方式を提案する．

筆者らが計算科学技術推進センターに設置されている複合並列計算機システム (COMPACS，図 2) 上に開発した STA 基本ソフト第 2 版 (STA_2，[3]) は，科学技術計算の並列分散化と実行に必要な MPI2 を基本とした計算機間通信基盤，および，開発ツールを統合的に利用できる Java アプレットにより実装された機種に依存しない GUI (グラフィカル・ユーザ・インターフェース) を備えた統合環境である (図 1)．STA_2 では，資源情報を常時収集し，ユーザプログラムに対して，提供するための枠組みを導入している．これを並列分散プログラムのスケジューリングに利用すると，計算時間の最小化が可能である．

提案するメタ・スケジューリング方式は，OSCAR コンパイラで実装されているマルチグレイン並列処理法におけるコンパイラにより生成される動的スケジューリングコードとして実現可能である．

2 資源情報サーバ

既存の並列分散環境である STA_2 に資源情報を専門に常時収集し，並列分散プログラムの実行時間の短縮化の目的のため，ユーザプログラムに提供するための枠組み (資源情報サーバ) を導入する．

並列分散環境において考慮する必要のある資源情報は，各並列計算機の演算処理，記憶領域，ネットワークの容量や負荷，各並列計算機とユーザプログラムの組み合せによる実行効率の相違など多岐にわたり時刻とともに変化する特徴を持つ情報である (表 1)．過去の資源情報を統計的にまとめたもの (統計情報)，これからの資源情報を予測したもの (予測情報)，各並列計算機で利用可能なモード，パーティション，時間などの運用情報も資源情報に含まれる．

情報提供サーバは STA_2 の情報提供サービスの一部として実装される [3]．メタ・スケジューリングのために並列分散プログラムが資源情報を参照する他，利用者が実行計算機を指定するときにも資源情報を参照できる．資源情報サーバは，STA_2 の実行計算機指定ツールから起動される各並列分散プログラムや STA_2 の計算機間通信基盤の状態を常時監視し，資源情報をデータベース化する (図 1)．

表 1: 資源情報サーバが提供する資源情報

種別		内容
静的情報		各並列計算機のカタログ性能 各プログラムの特徴，必要な資源
動的	観測情報	現時点での並列分散環境の状態
	統計情報	過去の並列分散環境の状態
	予測情報	予測される将来の並列分散環境の状態
その他		運用に関する情報

3 メタ・スケジューリング

従来，並列化コンパイラを使用した並列処理では，Do-all,Do-across などのループ並列化 [1] が自動的に行なわれている．このような並列化手法のみでは，ループが複雑なイタレーション間データ依存を持つと並列化できないし，ループ以外の部分の並列性を利用できなかった．このため，さまざまな粒度の並列処理を階層的に組み合わせて並列化を行なうマルチグレイン並列処理法が提案されている [2]．マルチグレイン並列処理法のコンパイル時の処理のうち，粗粒度の並列化を行なうマクロデータフロー並列化の概略を次に示す．粒度が小さい逐次ループ内並列化と近細粒度並列化は，実行時のスケジューリング負荷を低く抑えるために，静的スケジューリングを採用している [1, 2]．

1. プログラムを基本ブロックや最外側ナチュラルループ程度の粒度の粗粒度タスク (マクロタスク) に分解．
2. 各マクロタスク間のコントロールフロー，データフロー解析を行ない，マクロフローグラフを生成．
3. 最早実行可能条件解析を行い，マクロタスクグラフを生成．
4. マクロタスク間の条件分岐などの実行時不確定性に対処するため，実行時にマクロタスクをプロセッサに割り当てる動的スケジュールルーチンを生成．

マクロタスクの粗粒度並列処理のコンパイラが生成する異機種並列計算機間動的スケジューリングルーチンとして，メタ・スケジューリング方式を使用することが可

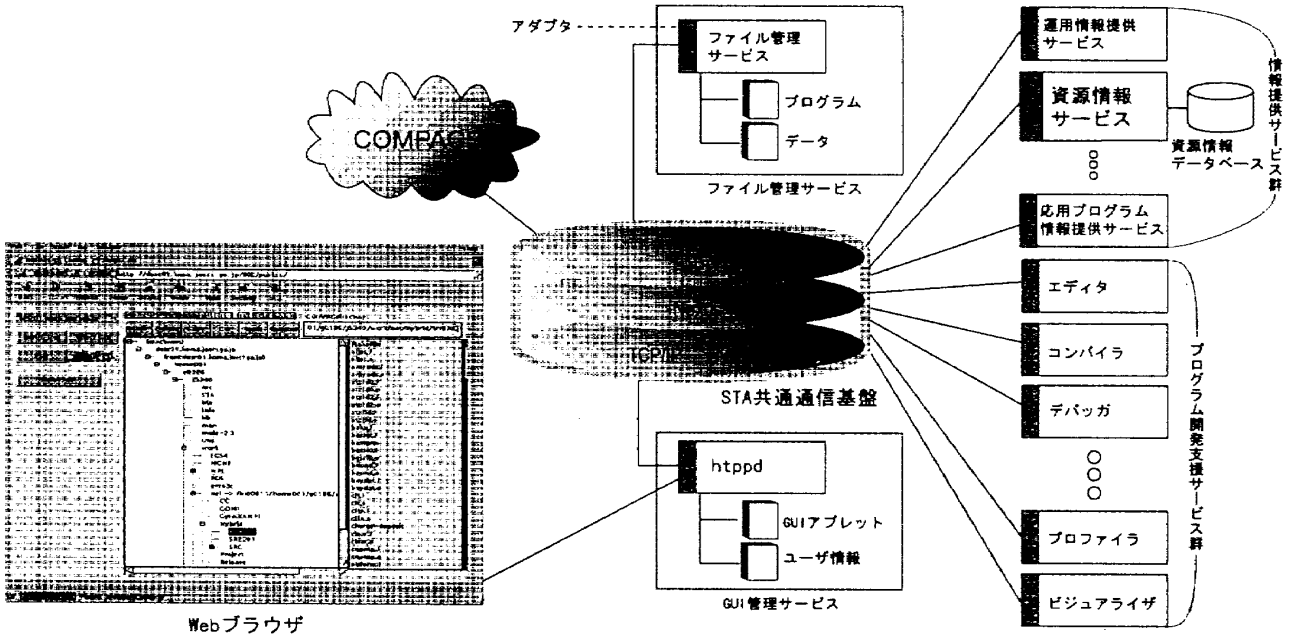

図 1: STA_2の構成.

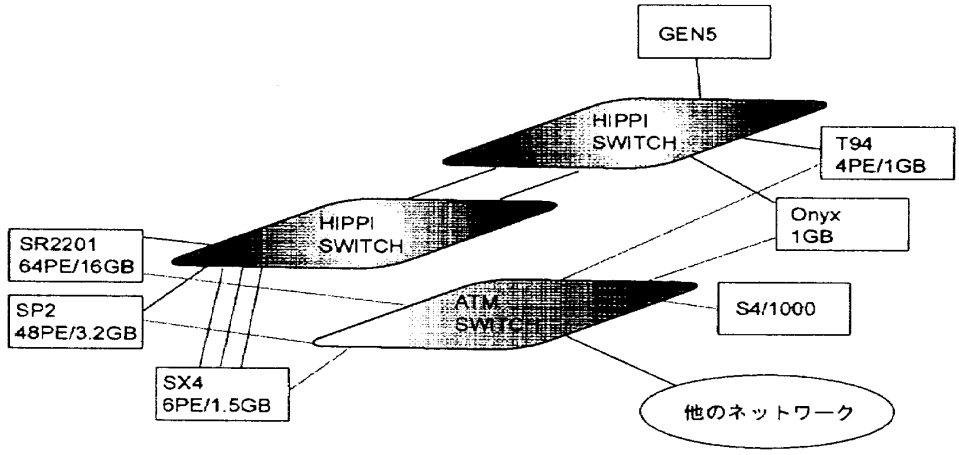

図 2: COMPACS の構成.

能である．具体的には，コンパイラが各マクロタスクに対して，複数の並列計算機用の並列化コードを生成し，動的スケジューリングルーチンは，資源情報サーバから提供される資源情報を参照しながら，各マクロタスクと並列計算機の組合わせを決め，各マクロタスクをネットワーク上の適切な並列計算機に割り当てる．

なお，本論文で提案したメタ·スケジューリングは手法の性格上，動的スケジューリングになるが，静的スケジューリングを行なう場合であっても，各基本ブロックやデータ転送に要する時間の見積りが必要であり，これに資源情報を利用できる．

4 まとめ

異機種並列計算機をネットワークで接続した並列分散環境において，ネットワークや並列計算機の負荷など並列分散環境の動的な変化を考慮して，計算時間が最小になるように処理を自動的に分配するメタ·スケジューリングの枠組を提案した．今後，メタ·スケジューリング方式で使用する動的スケジューリングルーチンをCOMPACS上で設計，実装し評価する予定である．

参考文献

[1] 笠原: 並列処理技術, コロナ社 (1991).

[2] Kasahara, H., Honda, H. and Narita, S.: Parallel Processing of Near Fine Grain Tasks Using Static Scheduling on OSCAR, *IEEE ACM Supercomputing '90*, pp.856-864 (1990).

[3] 武宮, 今村ほか: 複数の並列計算機上での科学技術計算のための統合利用環境の構築, 情処研報, 97-HPC-67, Vol.97, No. 75, pp.97-102(1997).