大規模環境を想定した階層型ワークフロー処理系の効率的な通信方式

仲　貴幸[†]　増田　嵩[†]
松本真樹[†,††]　大野和彦[†]

1. はじめに

大規模な並列処理を行う手法の一つとして，複数のプログラムをタスクとして組み合わせデータフローを記述するワークフローが使われている[1)]．一般にワークフローではタスクや通信の粒度が大きく，大規模な計算資源を安価に供給できる広域分散環境に適する．

我々はワークフロー記述のためのタスク並列スクリプト言語 MegaScript を開発している[2)]．従来の処理系は単一のマスターホストが全ワーカホストを制御する集中管理型であり，スケーラビリティに欠ける問題があった．そのため，階層型動作モデルに基づく処理系を提案し実装を進めている[3)]．しかし，従来の通信方式をそのまま適用すると効率的な通信ができず，実行性能が大幅に低下することがある．そこで，本稿では階層型処理系に適した通信方式を提案する．

2. 背　景

MegaScript では，タスクの標準入出力間をストリームと呼ばれる仮想通信路で接続することにより，タスク間のデータフローを定義する．入力端側に接続した入力側タスク群の出力はストリーム上で非決定的にマージされ，出力端側に接続した各出力側タスクにマルチキャストされる（図 **1**（a））．

処理系では，タスクが配置された各ワーカホスト上にストリームの入出力端を生成し（1）各入力端が入力側タスクの出力を受け取り（2）すべての出力を代表出力端へと収集し（3）他の出力端にも転送し（4）各出力端から出力側タスクへ入力する（図 1（b））．

階層型処理系では，ワーカホストを複数のホストグループに分け，それぞれをサブマスターにより制御する．グループによってはファイアウォールにより内外の直接通信ができなかったり，グループ間通信が低速な WAN 上で行われたりする．このため，上記の単純な実装方式では通信ができない，オーバヘッドが大きい，といった問題が生じる．

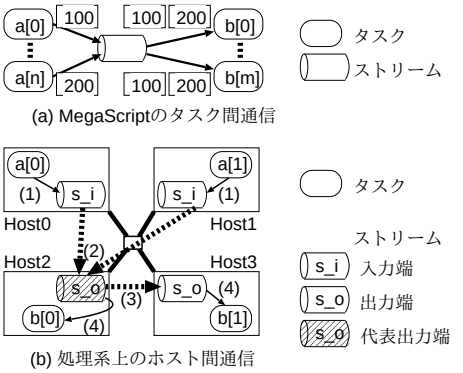

図 **1**　ストリームによるタスク間通信とその実装

3. 提案手法

図 1（b）に示すように，多対多のタスク間通信は多対一と一対多のホスト間通信で実現される．したがって，一対一，一対多，多対一の 3 通りのホスト間通信について，既存の通信効率化手法を使い分ける．

3.1 ルーティングの最適化

階層型処理系では，ファイアウォールの存在やローカルアドレスの使用により，任意のホスト間の直接通信が可能とは限らない．このため，マスター・サブマスターホストが通信を中継する必要がある（図 **2**（a））．しかし，中継経路のうち直接通信可能な部分はショートカットするルーティングを行うことで，中継段数を減らし通信オーバヘッドを削減できる．(図 2（b））．

3.2 マルチキャストの最適化

同じメッセージを複数のホストにマルチキャストする場合，各ホストに対して直接通信を行えば通信回数は最小となる．しかし，WAN を経由する場合はコストの大きい WAN 通信が複数回発生し，オーバヘッドが大きい（図 **3**（a））．受信ホスト群の上位のサブマスターホストに中継させ，LAN 内でマルチキャストを行わせることで，WAN 通信の回数を減らし全体の

† 三重大学　大学院工学研究科
†† 現在，株式会社 医用工学研究所

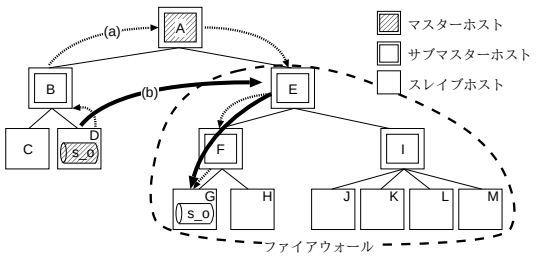

図 2　ルーティングの最適化

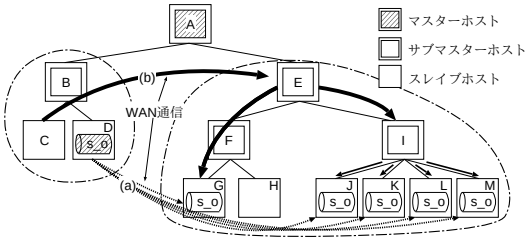

図 3　マルチキャストの最適化

通信オーバヘッドを削減できる（図 3（b））．

3.3 バッファリング

一定量のメッセージをまとめてから一度に送信することで，通信回数を削減できるが，遅延が増大する欠点もあるため，適切な粒度制御が必要となる．

4. 性能評価

階層型 MegaScript 処理系上に提案手法を実装し，簡単な予備評価を行った．評価には 28 台の PC（AMD Athlon(tm) II X2 B24(800MHz), 2GB Memory）で構成された PC クラスタ一組を使用し，次に挙げる 2 通りの階層型実行環境を構築した．

実行環境 A　マスターホスト下に 3 台のサブマスターホストがあり，各々が N 台のワーカホストを管理する．すべてのホストは同一 LAN 上に接続する．

実行環境 B　実行環境 A と同じ構造だが，1 台のサブマスターホストと配下のワーカホスト N 台は，異なる LAN に接続する．異なる LAN 間の通信は WAN 通信となり，ファイアウォールのためワーカホスト間の直接通信ができない．

物理的にはすべてのホストは同一 LAN 内にあり，WAN 通信は UNIX の `tc` コマンドで帯域制限を掛けることにより，擬似的に再現している．

これらの環境上で単純な多対一および一対多構造のワークフローを実行したときの実行時間を，それぞれ**表 1**，**表 2** に示す．ストリームの入力端側タスクは，各々が合計 30MByte を出力し，出力端側タスクはストリームの出力をすべて受け取る．

表 1　多対一のワークフローの実行時間 (s)

実行環境	N	非適用	ルーティング最適化	全て適用
A	2	195.3	52.73	4.01
	4	352.9	53.33	5.08
	8	732.1	56.58	8.54
B	2	191.0	184.41	13.28
	4	366.0	341.66	25.45
	8	761.7	714.15	49.82

表 2　一対多のワークフローの実行時間 (s)

実行環境	N	非適用	マルチキャストとルーティング最適化	全て適用
A	2	349.6	304.9	12.33
	4	646.2	502.1	19.40
	8	1134.7	956.3	31.07
B	2	389.7	252.9	9.08
	4	708.8	502.1	17.99
	8	1211.3	894.4	33.66

多対一の場合，ルーティング最適化により環境 A での実行時間は $N = 8$ で 1/13 と大きく低下した一方，B ではファイアウォールが一部のホスト間の直接通信を阻害するため，7%程度の実行時間削減にとどまった．バッファリングを併用することにより，環境 A では 1/86，B でも 1/15 まで，実行時間が短縮された．

一対多では，ルーティングとマルチキャストの最適化が可能であり，後者の効果により環境 B では A よりも大きな削減効果が得られている．また，この場合もバッファリングを併用することにより，環境 A, B ともに 1/36 まで実行時間が短縮された．

5. まとめと今後の課題

階層型ワークフロー処理系の効率的な通信方式を提案し，小規模環境の評価で基本的な有効性を示した．今後は大規模環境で評価を行うと共に，各手法の適用・非適用を制御する閾値や評価関数を決定していく．

参考文献

1) Deelman, E., Gannon, D., Shields, M. and Taylor, I.: Workflows and e-Science: An overview of workflow system features and capabilities, *Future Gener. Comput. Syst.*, Vol. 25, pp. 528–540 (2009).
2) 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 (2003).
3) 西川雄彦, 高木祐志, 大野和彦, 佐々木敬泰, 近藤利夫, 中島浩: タスク並列スクリプト言語 MegaScript ランタイムの広域分散化, 先進的計算基盤システムシンポジウム SACSIS2005, pp. 251–252 (2005).