

適応型ソフトウェアの高速化に関する一考察*

4H-9

蔭山 克禎†

藤崎 智宏†

浜田 雅樹†

NTTソフトウェア研究所†

1 はじめに

近年のインターネットの普及により、特に Web や Java 等に代表される技術をもちいた新しいタイプの C/S コンピューティングが開発・利用され始めている。そのようなソフトウェアには、自ら動作する環境に適応するための能力が必要である。

しかし、一般に適応性を持つソフトウェアは動的に判断する処理が多く、処理スピードが遅くなる問題がある。

本稿では、筆者らが開発したコンピュータネットワーク管理システム (NMS: Network Management System) を例に、適応的なシステムの高速化技術について考察する。

2 ネットワーク管理システムにおける適応性

NMS とはネットワーク機器を管理プロトコルを用いて遠隔操作し、状態管理するソフトウェアである。

マルチメディアネットワークでは、ネットワーク機器によって管理プロトコルが異なる。また、同じ機器でも処理内容に応じて適したプロトコルを用いる必要がある。NMS はこのような機器の管理プロトコルに適応する能力が必要である。

筆者らの NMS[1] は、機器管理エージェントが1つのネットワーク機器を管理し、そのエージェントの協調作業によって、ネットワーク全体の管理を行っている。

この NMS は、管理するネットワーク機器と管理内容に適した管理プロトコルを動的に選択して処理する適応性を持つ。これは、特定の管理プロトコル処理が異常等により利用できない場合、代替プロトコルを選択する機能を含む。

個々の機器管理エージェントは、この適応性を実現する MPH(Multi-Protocol Handling Mechanism) を実装している [2]。

図1に MPH の構成と動作例を示す。

図1に示す3つの Handler(PH) は、各々管理プロトコル処理用のオブジェクトである。SNMPHandler は SNMP に必要な処理を実装し、SNMPv2Handler には SNMPv2 に必要な処理を実装している。

PC(Protocol Controller) は、処理要求に対して転送先 PH を決定するオブジェクトであり、プライオリティが付けられた PH(Protocol Handler) のリストを持つ。

機器管理エージェントは機器にアクセスする必要がある時、内部にある PC に処理要求を出す。PC は処理要求を受けると、各 PH に対し順に処理可能かどうかを確認し、適した管理プロトコルを選択する。

図1には選択方法例を示している。PC はまずプライオリティが最も高い SNMPHandler に処理可能かどうか問

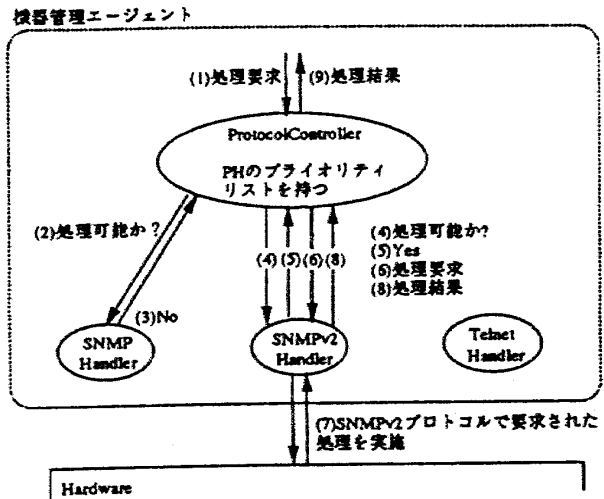


図1: MPH の構成と動作

い合わせる。SNMPHandler は処理不可能なので、PC に No を返却する。それを受けて PC は次にプライオリティが高い SNMPv2Handler に問い合わせる。SNMPv2Handler は処理可能なので PC に Yes を返却する。Yes を受けた PC は、SNMPv2Handler に処理要求を出す。SNMPv2Handler は、SNMPv2 プロトコルで要求された処理を実施し、処理結果を PC に返却する。

機器管理エージェントが持つ複数の PH は、管理プロトコルが異なっても同じ処理を持つ事が多い。そこで、選ばれた PH で処理しようとして応答がない時、その PH のプライオリティを一番低くして、次のプライオリティの PH で処理を試みる。もし、次のプライオリティの PH でも同じ処理が出来れば処理を続行する。このようにして、機器の特定の管理プロトコルを処理する部分で障害が発生した場合に対応している。

3 MPH の分析

MPHM は、処理内容に応じてプロトコルを使い分ける部分をハードコーディングした場合と比較して処理スピードが遅い。一様に PH に処理要求を出す条件では、15倍～30倍時間を要し、PH 数の増加に伴って急激に処理スピードが低下する事を実験により確認している(図2の(●)に示す)。その事から、パフォーマンスの低下には、全ての処理に PC-PH 間のネゴシエーション(処理が可能かどうかの確認処理)が必要である事が大きく影響していると考えられる。そこで、確認処理を減らし高速化する方法を検討する。

*A Study on Adaptive Software Performance

†Katsuyoshi KAGEYAMA, Tomohiro FUJISAKI, Masaki HAMADA

†NTT Software Laboratories

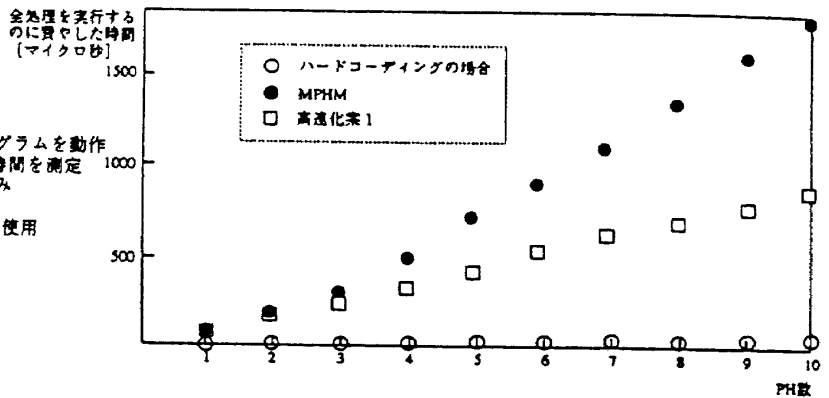


図 2: PH 数の増加に対する処理時間の変化

4 高速化方法

SNMP をベースとした NMS はネットワーク監視のため、ポーリングを定期的に行う。これは一定の時間間隔毎に機器にアクセスを行う処理で、同じ処理を繰り返す特徴を持つ。これを考慮すると、一度処理したメッセージのネゴシエーション情報を PC 側のキャッシュに蓄積する事により確認処理を削減できると考えられる。

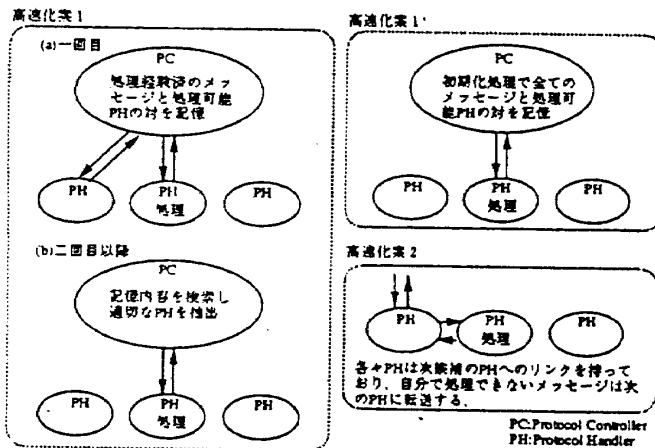


図 3: MPHM の高速化案

高速化案 1 PC は、ネゴシエーション情報として、処理要求名をキーとして処理可能 PH の対をハッシュテーブルに記憶する。PC は処理要求を受けると、ハッシュテーブルの検索を行う。ハッシュテーブルに処理要求名のキーが無ければ、MPHM と同様の方法で処理可能 PH を見つける。そして、ハッシュテーブルに処理要求名をキーとして処理可能 PH を登録する。これにより、同じ処理要求を受けた場合、ハッシュテーブルの検索結果から処理可能 PH を見つけるため、確認処理は減少する。図 2 に示した処理例は、同図で示す (□) のように改善される。

更に高速化案 1 には以下のバリエーションが考えられる。

高速化案 1' 高速化案 1' はリスト内の全 PH が処理できる全ての処理名をハッシュテーブルに登録する初期化処

理を行う。これにより、確認処理は高速化案 1 より更に削減される。しかし、ハッシュテーブルのサイズやヒット率、初期化処理にかかる時間を考慮し評価する必要がある。

さらに筆者らは、開発した NMS を分析した結果、以下の特徴があることがわかった。

1. 処理の多くはプライオリティが最も高い PH (第一 PH) で処理される
2. 1 つの機器管理エージェントの中に必要な PH 数は 3 つ以内である。

例えば、ある主要なルータ管理エージェントの例では、PH 数 2 (SNMPHandler、TelnetHandler) で、SNMPHandler のメソッド数は 20、TelnetHandler のメソッド数は 5 であり、第一 PH で処理される可能性が高い事が伺える。これを考慮すると、以下の高速化案が考えられる。

高速化案 2 各 PH が次候補の PH へのリンクを持つ。各 PH は自分で処理できない要求を受けると、次候補の PH に処理要求を転送する。これを繰り返して処理可能な PH まで辿りつき処理されると、反対経路で返答されて結果を得る。

本方式は並列動作するオブジェクトで構成するシステムでよく用いられる方式であり、単一プロセス内で本方式を適用した場合、戻りによるオーバーヘッドが考えられる。しかし、第一 PH への集中度が高い場合は、本家で平均処理スピードを上げる事ができる。この案についても今後の評価が必要であると考えている。

5 今後の予定

今後は以上の案について、評価をすすめていく予定である。

参考文献

- [1] Takashi Arano et al. "A Computer Network Management System Platform based on Distributed Objects". *IFIP/IEEE DSOM '95*.
- [2] 藤山 克禎 他. "ネットワーク管理システムにおけるマルチプロトコルハンドリングについて". 情報処理学会 分散システム運用技術研究グループ 研究報告 No2, 1996.