

第5章 結論

分散処理環境内で単一な名前空間は、その環境内で作業を行う各ユーザに対し同じイメージを提供する。この単一な名前空間により、各ユーザはその共通の基盤を用いて協調作業を行うことができる。本研究では、このような特性をもった単一な名前空間に対して、一貫した方法でユーザがカスタマイズすることを可能とする多重名前空間を提案した。またその名前解決を効率的に行う機構として、クライアントにプレフィックス・テーブルを利用したキャッシュ機構、および名前空間選択テーブルを用いることで、名前解決を効率的に行う手法を示し、その性能評価を行う実験を行った。実験により、名前の更新率が低い場合において本提案方式の効率化手法が非常に有効であることが示された。名前の更新率が高い場合においては、サーバ間の一貫性情報の交換によるネットワークのトラフィックの増大が見られたが、クライアントのレスポンスタイムは向上しており、本提案方式の効率化手法はこの場合も有効であることが示された。

また、クライアントおよびサーバにキャッシュのための機構を埋め込む前述の効率化方式に対し、通信経路上の機器にキャッシュ機構を埋め込むという *intermediate caching* の手法を提案した。これは、ルータなど通信経路上の機器、またはパケットをプログラマブルにするというアクティブネットワーク技術を応用したものである。また、キャッシュ機構を内蔵した専用ルータなどが出現している今日、これら通信経路上での処理は現実的なものになってきている。本論文で提案した *intermediate caching* 方式は、クライアント及びサーバに透明な形で名前のアクセス頻度や更新頻度を取得し、キャッシュ情報の伝播を制限するという特徴をもつ。本論文では、多重名前空間システムに対し本方式を導入し、その性能評価を行う実験を行った。実験において、提案方式によってネットワークのトラフィックの減少、サーバに対する負荷の減少、および要求パケットの減少が示された。

今後の課題には以下のものが考えられる。

本論文では、多重名前空間における名前解決を効率化する手法として、名前空間選択テーブルを用いた方式を提案した。しかし、本方式では名前の更新時における名前空間一貫性情報の送信において多大なコストを生じさせていた。これに対し、名前空間の一貫性の定義を緩いものとするすることで、一貫性情報の送信を遅らせ複数の一貫性

情報の送信を1回の送信にまとめるなどの効率化が可能となると考えられる。また、一貫性情報の送信量を減らすための名前空間順序リンクは、ループを許していないためビューパスの自由度を低下させていた。これに対処するため、同じ情報が複数回到着したことを検知する機構を導入することが考えられる。

アクティブネットワーク技術を利用したキャッシュ機構に関しては、以下に述べるような研究課題が考えられる。まず、アクティブノードにおけるキャッシュの処理はさらに高度なものにすることができると考えられる。現在は、キャッシュの伝播を制御するパラメータ N_{acc} , N_{prop} および N_{info} の値は実行時は定数とされているが、これらの値を変化させてもアルゴリズム的には問題はない。これらの値を名前の更新頻度の変化に応じて適切に変化させることができれば、提案方式の性能をさらに向上させることができると考えられる。

また、本提案方式ではアクティブノードが故障した際には、その下流のノードにおけるキャッシュも全て消去されるという方式をとっていた。これに対し、名前空間の一貫性を緩めることで、キャッシュをある時間保持しその時間内に故障ノードが復旧しなかった場合のみキャッシュを消去することで、キャッシュの消去を最小限に抑えることが可能である。

ネットワークのトポロジーに関して、本提案方式は非常に単純な形態を仮定していた。すなわち、要求パケットとその返答パケットが全く同じルートを通るということである。この仮定は、現実には使用されているネットワークでは必ずしも常に真であるとは言えない。この仮定が破られた場合にも正しく動作するシステムを作成する必要がある。また、本論文で述べた名前解決システムはデータグラム型の通信である UDP/IP 上にその実装を行っていた。しかし、HTTP や FTP など、現在インターネット上のデータの獲得に多用されているプロトコルは、ストリーム型の TCP/IP[26] 上に実装されている。途中の機器において TCP/IP のストリームを再生し、その解析を行うには複雑な処理が必要となる。本論文で述べたようなパケット単位で要求が完結するシステムとは違ったアプローチが必要となると考えられる。

次に、本提案方式の実装手法には以下のような検討課題がある。現在、アクティブノードのプログラムは Java を用いている。このため、クライアントのレスポンスタイムにおいては、キャッシュ機構のオーバーヘッドによって、キャッシュを置かない場合より性能が悪化していた。また、トラフィックが非常に大きくなると考えられる基幹のルータなどに実装する場合は、低レベルな部分で実装を行うことで最大限の高速化を行い、パケットのスループットを向上させること必要である。これには、UDP/IP レベルやそれ以下のレベルのパケットを直接扱うことができるネットワークデバイスのデバイスドライバ上に本提案方式を実装することで更なる性能向上を図ることが考え

られる。

最後に、提案方式のキャッシュ手法の既存プロトコル (Domain Name System [18,19] 等) への適用があげられる。DNS は、ローカルのサーバで解決できない問い合わせをすぐにルートサーバへ転送する (もしくは、forwarder として上位のサーバを静的に設定しておく必要が有る) ため、ルートサーバへの負荷は膨大なものとなりがちである。これらをパケットの送信経路上で解決できれば、サーバの負荷やネットワーク全体の負荷を低減できると考えられる。