

分散オブジェクトシステムにおける性能モニタリング方法

5S-2

江丸 裕教[†] 赤津 雅晴[†]

1. はじめに

近年、CORBA¹⁾による国内企業の大規模システム開発事例が発表されるなど分散オブジェクト技術が認知されつつある²⁾。しかしながら、企業情報システムへの本格適用のためには、いっそうの性能向上に加え、システム性能見積もり、チューンアップ技術の確立が不可欠である。現在、分散オブジェクト技術を用いたシステムの構築を行う上で、性能の見積もりや保証が困難であるという点が分散オブジェクト技術を用いたシステムの導入の障害となっているからである³⁾。今後、分散オブジェクト技術を用いたシステム構築が普及するためには、あらかじめシステムの性能を見積もったり、構築されたシステムの性能チューニングを行うといった技術がますます重要性を増してくると考えられる。

本稿では上記現状を踏まえ、分散オブジェクトシステムの性能モニタリング方法を提案し、そのプロトタイプ開発の結果を示す。

2. 性能モニタリングの考え方

分散オブジェクトシステムでは一つのアプリケーションが分散した複数のオブジェクトから構成される。その上、分散オブジェクトは異なったプラットフォームを持つ複数のマシン上で同時に並行的に動作することもある⁴⁾。このように複雑性、多様性を持つ大規模な分散オブジェクトシステムにおいて性能を把握するためには、スケーラビリティとポータビリティの確保が必要となる。

そのために図1に示すようなアーキテクチャを採用する。スケーラビリティを確保するため、各デ

ータ採取ポイントにおいてデータの採取を行うコレクタ、データの中継、バッファリング、コレクタの管理を行うコントローラ、GUIを提供するマネージャの3階層の構成とする。ポータビリティ確保のためには、コレクタをアプリケーション層、ミドルウェア層、OS層の三つのレイヤに分割することによって対処する。これにより、該当するコレクタを差し替えるだけで様々な環境に対応することが可能となる。

また、複雑性、多様性を持つシステムの中から効率よく性能ボトルネックを検出するために、以下の二つの機能を持たせる。

第一に大規模な分散オブジェクトシステムの特性に適応したデータ採取法を用いる。分散オブジェクトシステムでは一つの処理が複数のオブジェクト、複数のマシンにまたがって行われる。そのため、効率の良いモニタリングを行うには、多数のオブジェクトの中から着目するアプリケーションを構成するオブジェクト群を抽出することが必要である。このために、マシン、オブジェクトなどある一点に着目して得られる性能データ(以後ポイントデータと呼ぶ)に加え、フローデータという考え方を導入する。フローデータとは処理されるメッセージに着目し、メッセージの流れを追跡することによって採取され

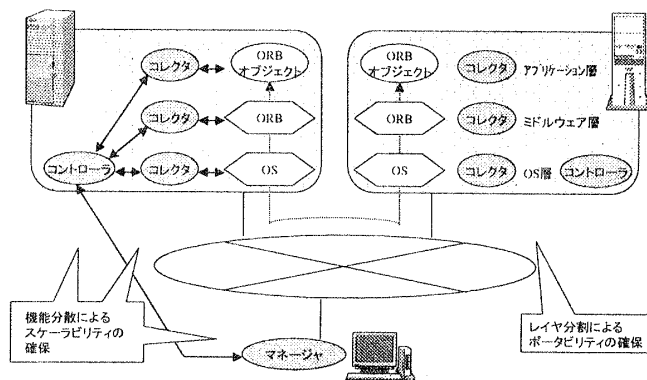


図1 アーキテクチャ

[†] 株式会社 日立製作所 システム開発研究所
Systems Development Laboratory, Hitachi, Ltd.

るデータである。これにより処理の流れをオブジェクトごとに時間軸に沿って表示することが可能となる。

第二に採取データの種類、採取ポイントをユーザの指示によって、モニタリング実行時に動的に切り替える手段を提供する。これにより、フローデータとポイントデータの連携が可能になる。例えば、大規模なシステムの中からフローデータによってオブジェクトを抽出し、そのオブジェクトからポイントデータを採取することによって、ある処理の全体とそれぞれのオブジェクトの詳細な処理を同時に調査することが出来る。

3. 実行例

上記要件を満足するモニタリングツールを開発した。図 2 はマネージャのメイン GUI である ObjectExplorer である。アクティベートされているオブジェクトをマシン毎に一覧表示する機能を持つ。

フローデータを採取した例を図 3 に示す。この図から、NumClient から SwitchingServer を介し OddServer へ 1 回、EvenServer へ 6 回の呼び出しが行われたことがわかる。フローの更新はリクエストの発生に対し、リアルタイムに行われる。

4. まとめ

大規模な分散オブジェクトシステムに適した性能モニタリングツールを開発した。スケーラビリティとポータビリティを確保した設計を行うとともに、処理の流れを可視化するフローデータを導入することによって、性能ボトルネックの検出をサポートした。

今後は採取データ、対応プラットフォームに関する拡張を行い、大規模な分散オブジェクトシステムへの適用を行うことによって更なる評価を行う。

参考文献

- 1) The Common Object Request Broker: Architecture and Specification, OMG Document Revision 2.2(1998)
- 2) 今こそ CORBA を見直せ!、日経コンピュータ、1999年2月、pp124-142(1998)
- 3) Aniruddha S. Gokhale and Douglas C. Schmidt: Evaluating CORBA Latency and Scalability Over High-Speed ATM Networks, Proceedings of ICDCS '97(1997).
- 4) Robert Orfali, Dan Harkey and Jeri Edwards: The Essential Distributed Objects Survival Guide, John Wiley & Sons(1996)

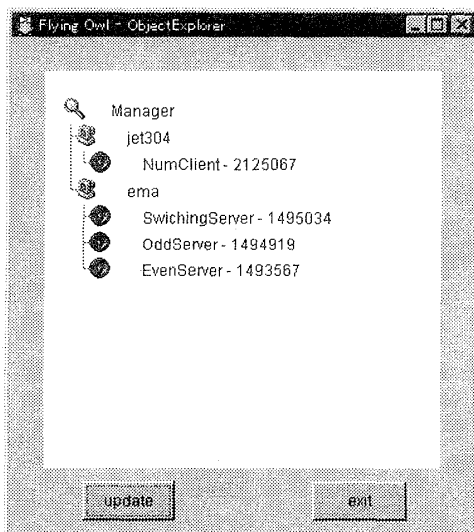


図 2 ObjectExplorer

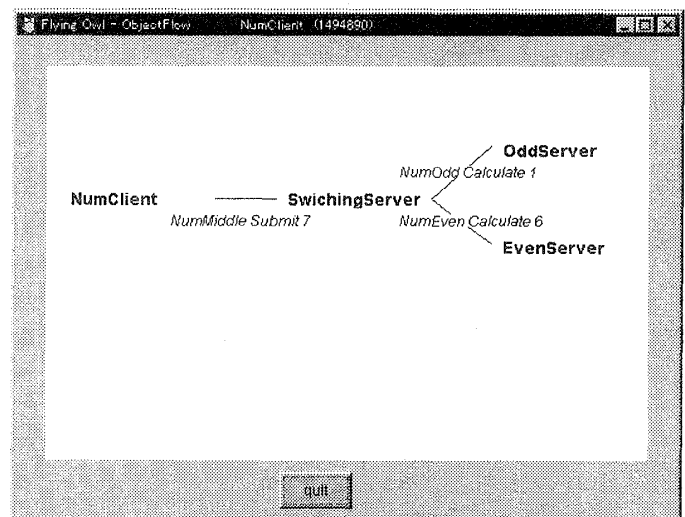


図 3 ObjectFlow