

6J-03 広域網分散オブジェクト環境における複製の一貫性管理方式について

石野竜馬 渡辺 尚

静岡大学

1 はじめに

近年のインターネットの普及によって、分散した異機種システム間の相互運用に対するニーズが高まっている。しかし、従来のクライアント・サーバ方式では、クライアントとサーバ間の接続にはさまざまなインタフェースやプロトコルが使用されており、分散したシステム間で相互運用を実現することは非常に困難である。また、オープン化の影響により、システム全体をあらかじめ確定することが困難となっており、クライアントとサーバの関係があらかじめ確定されなければならないクライアント・サーバ方式では十分ではない。これらの問題を解決する技術として、CORBA[1]などの分散オブジェクト技術に注目が集まっている。分散オブジェクト技術では、ネットワーク上に散りばめられたオブジェクトが相互に連携しながら全体として機能するものである。しかし、通信遅延による影響が、逆にパフォーマンスが低下してしまう可能性がある。RORP(Replicated Object Relocation Protocol)[2]は複製を動的に近くの間接サイトに配置することによって(図1参照)、この問題を克服している。しかし、複製を生成することはオリジナルとの一貫性問題が生じる。すなわち、ユーザに対して満足に行く結果を返すことができなくなる可能性がある。本稿では、RORPの一貫性管理に関する改良方法を提案する。

2 RORP の概要

RORPは、参照要求、返答にかかる通信遅延、オブジェクトの参照頻度、サイトの処理速度等を考慮し動的に適切な中間サイトに複製を配置するプロトコルであり、複製の再配置、複製の予約、複製の共有、複製の更新、の4つの機能を持っている。RORPでは、多数の複製をネットワーク上に配置することによって起こる複製の一貫性保持の難しさを、分散されたオブジェクトの総数規制をする規制数を利用することによって対処している。規制数とは1オブジェクトに対して幾つ複製を配置することが可能かを示す数である。

RORPの複製の更新機能は、以下の2つの方法を取り扱っている。

- lazy update

クライアントは更新間隔ごとに中間サイトを經由してサーバに対して更新要求を送信する。もし、サーバは、オブジェクトの変更が無ければ、クライアントに通知し、これを受信したクライアントは更新間隔を2倍にする。オブジェクトの変更がクライアントの更新間隔の間に2回以上あった場合、更新間隔を1/2にする。

Consistency Management for Replicated Objects on Distributed Computing System.

Ryoma Ishino, Takashi Watanabe
Shizuoka University

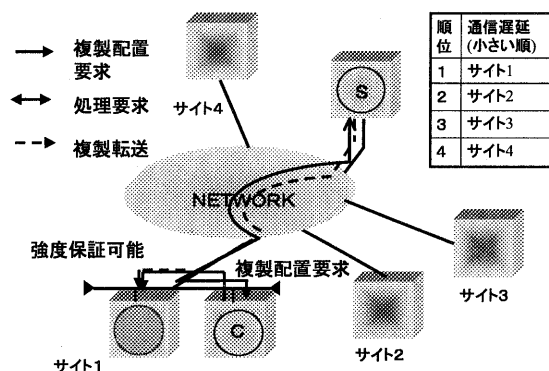


図 1: 提案方式の概要図

- immediate update

サーバが重要な更新であると判断した場合、サーバ自らが更新を行なう。

RORPでは弱い一貫性管理を行っており、一貫性が損なわれる可能性がある。理想的には、サーバオブジェクトが更新された場合、その複製も同時に更新されることが望ましいが、厳密な一貫性保証を行うとオリジナルの更新が生じるたびに複製も更新され、ネットワークトラフィックの増加を招く[3]。

3 提案方式

RORPは、弱い一貫性管理を行っているため、ユーザが望む結果が得られない可能性がある。本稿では、それらを考慮した方式の提案を行う。

3.1 提案方式の概要

ユーザに対して満足に行く結果を返すことができない弱い一貫性管理の欠点と、更新によるトラフィックの増加を引き起こす強い一貫性管理の欠点を、本方式では以下の2点で解決する。

- 一貫性強度の設定

アプリケーションによって一貫性強度(C)を設定できるようにする。この強度は、どの程度の一貫性を望んでいるのかを示すもので、 $C=R/O$ (R は複製の更新回数、 O はオリジナルの更新回数である。)

で表される。この値は中間サイトに送信される。クライアントは要求された一貫性強度を達成できる中間サイトを選び、中間サイトは、一貫性強度の管理を行う。図1は一貫性強度設定を満たせる中間サイトに複製配置要求を行い、中間サイトによって一貫性管理が行われている様子を示している。

● トラフィックを考慮した更新

中間サイトは更新を行う際トラフィックを考慮に入れる。サーバはオリジナルオブジェクトの更新が行われると、そのオブジェクトを保有する中間サイトに対して更新通知を発行するが、中間サイトは、中間サイトとサーバ間の通信遅延が閾値を超えると複製を更新しない。しかし、更新をしないことにより、一貫性強度を保証できない可能性がある。

3.2 一貫性強度を満たせなくなった際の処理

中間サイトが更新を行わないことにより、一貫性強度が保証できなくなる場合が生じる。その際クライアントとサーバでは以下の選択肢がある。

<クライアント>

- 他の中間サイトへの移動。
- 強制的に更新。
- 更新は行わず参照を続ける。

<サーバ>

- 規制数の制限。

一貫性強度が保証不可能となっても、クライアント側は通知によって対処することが可能となる。また、サーバ側でも、規制数を減らすことによって、一貫性強度を保証することが可能である。

3.3 処理の流れの例

本方式の具体的な流れを説明する。図2は、中間サイトAにオブジェクトを配置し、その後中間サイトAが複製更新を行わなかったために、一貫性強度が保証されなくなり、クライアントAは参照するサイトを中間サイトBに変更した場合である。

中間サイトに複製を配置するまでと、配置後に分けて具体的な処理の流れを示す。

<複製の配置前>

1. クライアントAは、最初サーバへ処理要求をする。
2. クライアントAは通信遅延が小さく、一貫性強度を満たすことが可能な中間サイトを検索する。
3. 中間サイトAへ複製配置要求を行う。
4. 中間サイトAはサーバへ複製転送要求を行う。

<複製の配置後>

1. オリジナルの更新により、サーバは中間サイトAへ通知する。
2. 更新通知を受けた中間サイトAは中間サイトAとサーバ間のトラフィックが閾値を超えていないため、更新する。
3. 2度目の更新通知を受けた中間サイトAはトラフィックが閾値を超えていたため、更新をしない。
4. 複製更新をしなかったため、クライアントAの一貫性強度を保証できなくなったため、中間サイトAはクライアントAへ通知する。
5. クライアントAは、他の中間サイトBへの移動を決定し、複製配置要求をする。

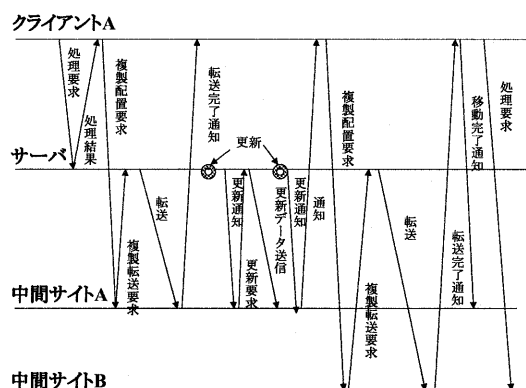


図 2: 提案方式の処理の流れの一例

3.4 提案方式の考察

本方式ではネットワークトラフィック状況に応じて更新を行うため、トラフィックの増加を防ぐことができ、トラフィックの許す限り一貫性を保証することが可能となる。しかし、サーバが保有するオリジナルオブジェクトが頻繁に更新されると、その際に発生する更新通知により、逆にトラフィックの増加を招く恐れがある。更新通知は数十バイトのオーダーであり、更新データサイズは、少なくともその100倍以上のバイト数はあるため[3]、全体的なトラフィックは減少するものと考えている。

4 おわりに

本稿により一貫性強度の設定を可能にし、トラフィック状況に応じた更新を可能にする一貫性管理方式の提案を行った。今後は、一貫性を重視するかレスポンスを重視するかの考察をすることによって、提案方式の改善をおこなう予定である。例えば一貫性強度設定に幅を持たせ、その幅の中で最速のレスポンスを得られる中間サイトに複製配置要求をするなど、さまざまな方法を考察する必要がある。また、閾値と節約できるバンド幅との関係の評価と、RORPの共有と予約機能との関連を検討する予定である。

参考文献

- [1] "Distributed Object Computing with CORBA", November 1997, Available from <http://www.cs.wustl.edu/schmidt/corba.html>
- [2] 渡辺 尚, 森 章文, 山本幸寿, "分散オブジェクト指向システムにおける動的複製配置プロトコル", 電子情報通信学会論文誌 D-I, Vol.J83-D-I, No.8, 2000 年 8 月 (予定)
- [3] J.Gwertzman and M.Seltzer, "World wide web cache consistency", in Proceedings of the USENIX 1996 Annual Technical Conference, pp.141 - 152, USENIX, Jan. 1996.