

無線 LAN 環境向け分散オブジェクト基盤

秩父 真司† 横山 孝典† 志田 晃一郎† 兪 明連†

武蔵工業大学大学院工学研究科電気工学専攻†

1 はじめに

情報システムの複雑化、大規模化に伴い、分散オブジェクト技術を利用したシステムの開発が普及してきている。分散オブジェクト技術では、ネットワーク上の各ノードに配置されたオブジェクトを連係動作させる。一方、無線技術の発展により各ノードをつなぐネットワークの無線化が進んでいる。無線 LAN 環境では、電波状況やノードの移動により、LAN からのノードの離脱が通常的に発生する。無線 LAN 上に分散システムを構築すると、サービスを提供するサーバオブジェクトを持つノードの離脱によりサービスが中断することがあり、システムの可用性が低下する。そのため、無線 LAN 環境におけるノード離脱に対応した分散オブジェクト基盤が必要になっている。

可用性を向上させる従来研究として、分散オブジェクト基盤にオブジェクトのレプリケーション機能を付加した、DeDiSys(Dependable Distributed Systems)[1]が提案されている。DeDiSys は、レプリケーション方式として受動的レプリケーション(Passive replication)を採用している。しかし、この方式はメソッド呼び出し時に更新伝播などによって遅延が発生する可能性がある。また、レプリカを配置した全てのノードが離脱してしまうと対応できなくなるため、多くのノードにレプリカを配置しておく必要がある。しかし、多数のレプリカ間で一貫性を保持することは容易ではない。

そこで本研究の目的は、ノードの離脱が発生してもサービスの提供を継続可能な可用性の高い無線 LAN 環境向け分散オブジェクト基盤の開発である。特に、メソッド呼び出し時の応答速度の向上、および一貫性保持にかかる負荷の減少を狙いとする。

2 能動的レプリケーション方式

本研究で対象とするのは、アクセスポイントを設置したイベント会場や会議室における無線 LAN 環境である。無線 LAN を介して情報共有や協調作業などを行うことがその目的である。ノードの移動は歩く速度程度で、離脱は一時には 1 台のみと想定する。

メソッド呼び出し時の応答速度向上のため、能動的レプリケーション(Active replication)を採用する。図 1(a)に DeDiSys で採用している受動的レプリケーション、図 1(b)に本研究で採用する能動的レプリケーションのメソッド呼び出しの流れを示す。受動的レプリケーションではプライマリレプリカのみが処理を行い、他のレプリカには更新情報を伝播することで一貫性を保持している。そのため、他のレプリカからの更新終了報告の待ち時間、およびプライマリレプリカの離脱による再呼び出し時間などの遅延が発生する。

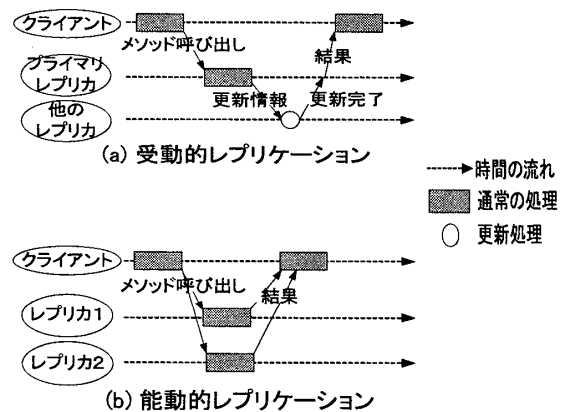


図 1: レプリケーション方式による違い

これに対し本研究で採用した能動的レプリケーションでは、複数のレプリカが同一の処理を並行して行うことで一貫性を保持する。このため、プライマリレプリカとその他のレプリカを区別する必要はない。ノード離脱が発生し、いずれかのレプリカとの接続が途絶えた場合でも、他のレプリカから結果を受け取ることができ、ノード離脱による遅延が発生しない。また、レプリカ間で更新終了を待つ必要がないため、通常時と比べても応答速度は低下しない。

また本手法では、レプリカ数を削減するために一度に生成するレプリカを 2 つのみとし、周期的な接続確認とレプリカの再生を行うことでノード離脱に対応する。これにより、LAN 内には 2 つのレプリカが存在するのみであり、一貫性の保持にかかるオーバーヘッドを低減することができる。

A distributed object platform for wireless networks

†Shinji Chichibu, Takanori Yokoyama, Koichirou Shida and Myungryun Yoo

†Graduate School of Electrical Engineering, Musashi Institute of Technology

3 能動的レプリケーションの実現方法

3.1 代理オブジェクトの拡張

能動的レプリケーションを実現するためには、2つのレプリカに対して同時にメソッド呼び出しを行う必要がある。本研究では、ORB コアには手を加えず代理オブジェクトを拡張することで実現する。代理オブジェクトは、分散オブジェクト基盤においてサーバオブジェクトとクライアントオブジェクトの間で接続の確立、メッセージの受信・解析などを行う。代理オブジェクトに対してメソッド呼び出しの二重化機能を付加することで、2つのレプリカへの同時呼び出しを実現する。

3.2 ネストされたメソッド呼び出しへの対応

能動的レプリケーションでは、オブジェクトの呼び出しがネストして発生する場合（サーバとして呼び出されたオブジェクトが、クライアントとして他のオブジェクトを呼び出す場合）、2つのレプリカが同一のオブジェクトに対して重複した呼び出しを行ってしまう。呼び出されたオブジェクトが同一の処理を2度実行してしまうと、状態を維持できなくなる。そこで本研究では、メソッド呼び出しに識別子を付加し、同じ識別子を持った呼び出しが呼び出し先の代理オブジェクトに渡されると、2度目の呼び出しに対しては処理を行わず結果のみを返すことによって、この問題を解決する。

4 レプリケーション処理の流れ

4.1 レプリカの生成

図2にレプリカ生成時の動作を示す。本研究では、アプリケーションがサーバオブジェクトを生成する際、LAN内の全ノードにレプリカの生成依頼の打診をブロードキャストする。生成依頼の打診を受け取ったノードの中で最も早く返答してきたノードに対してレプリカの生成依頼を行い、そのノード上にレプリカを生成する。依頼によってレプリカを生成したノードは、依頼元のノードと周期的に接続確認を開始する。

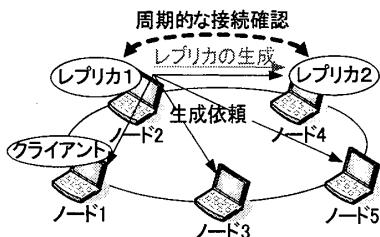


図2: レプリカの生成

4.2 レプリカの再生

図3にレプリカ再生時の動作を示す。ノードの移動などにより、いずれかのレプリカを持つノードが離脱したとする。レプリカ間の切断を検知すると、検知したノードはレプリカの再生が可能な他の近隣ノードを探索する。レプリカの再生先が見つかったノードは、再生先のノードにレプリカを再生する。一方、再生先が見つからなかったノードは、自身がLANから離脱したとみなし、自身の持つレプリカを破棄する。離脱したノードが再びLANに参加しても、すでに持っていたレプリカは破棄されているため、存在するレプリカ間の一貫性は保持される。

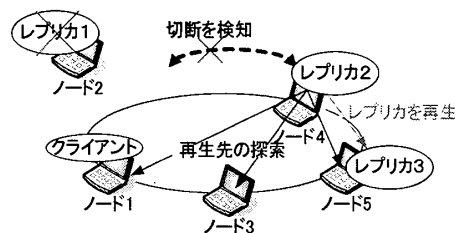


図3: レプリカの再生

5 実装方法

Java 言語で書かれた CORBA[2] 準拠の ORB である HORB[3] に対して、本論文で提案したレプリケーション機能を追加実装した。レプリケーションを行う複製機能、接続確認機能、再生機能を新規に Java 言語で開発する。HORB における代理オブジェクトである Proxy 及び Skelton に対して機能を追加する。具体的には、HORB のコンパイラである HORBC コンパイラによって Proxy 及び Skelton を生成する際に上記機能を付加することで実現する。これらのプログラムをアプリケーションと既存 ORB の間で動作させることで、提案する分散オブジェクト基盤を実装した。

6 おわりに

無線 LAN 環境を対象にノード離脱に対応するため、応答速度の優れたレプリケーション機能を持つ分散オブジェクト基盤を開発した。今後の課題として、マルチクライアントでの利用時に起きる呼び出しの順序問題への対応などがある。

参考文献

- [1] Stefan Beyer, Implementing network partition-aware fault-tolerant corba systems, *2nd International Conference on Availability, Reliability and Security*, pp. 69–76, 2007.
- [2] Object Management Group, *Common Object Request Broker Architecture: Core Specification*, version 3.0.3, 2004.
- [3] 平野聡. *You Can Fly With HORB*, 産業総合研究所, <http://www.horb.org/>, 1999.