社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2006－DPS－129（1）
2006／11／16

広域分散システムのためのデータベースクラスタの構想

八木 哲　長谷川知洋　長谷川雅一
NTT サイバースペース研究所

本稿では，インターネットワイドに分散した多数のサイトが連動する広域分散システムの基盤として利用できるデータベースクラスタとして，複合クラスタ（MLC:Multi-Layered Cluster）を提案する．MLC の特長は，(a)情報の柔軟な共有(b)高い可用性(c)低コストで利用できるスケーラビリティである．MLC では，(a)(b)のために，各サイトに複数のデータベースを配置し，サイト内ではサイト内の全情報を共有するように束ね，サイト間では任意のサイトが任意の情報を共有するように束ねる．(c)のために，MLC 自体を分散実装し，無償で利用できる OSS のデータベースを無改造のまま使用する．本稿では，3 層構造を持つ MLC の概要と各層の詳細を示す．

A Database Cluster for facilities of Globally Distributed Systems

Satoru Yagi, Tomohiro Hasegawa and Masaichi Hasegawa
NTT Cyber Space Laboratories

This paper proposes the MLC (Multi-Layered Cluster). MLC is a database cluster for facilities of globally distributed systems in which a lot of sites distributed over the Internet cooperate. MLC has three features: (a) flexibility in information sharing; (b) high availability and (c) scalability with low cost. For (a) and (b), it allocates some databases on each site. In the site, databases share all information in the site. Among sites, databases may share arbitrary information. For (c), it uses a distributed implementation and free open-source databases with no remodeling. This paper describes the design of MLC which has three-layer construction.

1. はじめに

組織の広域分散化・災害対策や，センサーネットワーク [1,2]・ITS[3,4]などのユビキタス・システム [5,6]の実用化が進んでいる．本稿では，このようなインターネットワイドに分散した多数のサイトが連動する広域分散システムの基盤として利用できるデータベースクラスタとして，以下の特長を持つ複合クラスタ（MLC:Multi-Layered Cluster）を提案する．

(a) 情報の柔軟な共有．
(b) 高い可用性．
(c) 低コストで利用できるスケーラビリティ．

MLC では，(a)(b)のために，各サイトに複数のデータベースを配置し，それらを束ねて情報を共有する．サイト内ではサイト内の全情報を共有するように束ね，サイト間では任意のサイトが任意の情報を共有するように束ねる．(c)のために，MLC 自体を分散実装し，無償で利用できる OSS のデータベース（MLC のプロトタイプでは機能性に定評がある PG:PostgreSQL を選択）を無改造のまま使用する．結果として，データベース開発コミュニティの成果も容易に取り込める．本稿では，３層構造を持つ MLC の概要（2 章），各層の詳細（3～5 章），今後の課題（６章）を示す．

2. 概要

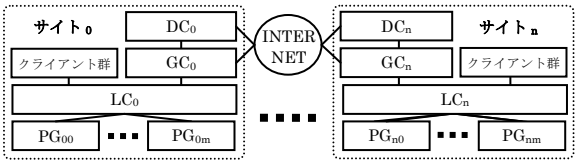

図 1　MLC の構成

MLCの構成を図 1 に示す．第 1 層のローカルクラスタ（LC:Local Cluster）は，データベースを複製する．すなわち，サイト内で束ねたPGがサイト内の全情報を等しく持ち，サイト間で束ねたPGがサイト間で共有する情報を等しく持つように複製する．第 2 層のグローバルクラスタ（GC:Global Cluster）は，サイト間で束ねたPGにサイト間で共有する情報を配信する．すなわち，配信元サイト（更新内容を配信できる更新権を持つサイト）で情報が更新されると，その更新内容を他のサイトに素早く確実に配信する．第 3 層の分散配信制御モジュール（DC:Distributed Delivery Controller）は，GCが用いる配信経路を管理する．すなわち，各サイトからの要求を調停したうえで，参加・脱退要求に応じて配信経路を構築し，配信元サイトの移動要求（配信権の移譲要求）に応じて配信経路を変更する．

3. ローカルクラスタ

3.1. 概要

LC に用いるデータベースの複製方式は，以下の要件を満たす必要がある．

(a) PG の改造が不要である．

(b) サイト内で束ねた PG（同期複製する）と，サイト間で束ねた PG（遅延を考慮して非同期複製する）に適用できる．

(c) オーバーヘッドが小さい．

(d) 動作中に新規の PG を接続できる（保守作業に起因する可用性の低下を防ぐ）．

一般に，データベースの複製方式は以下の 2 系統に大別できる．

- ジャーナルログ・ベース：複製元のデータベースのジャーナルログ（データベースファイルの更新差分）を，複製先のデータベースのデータベースファイルに反映する．
- クエリ・ベース：複製元と複製先のデータベースにおいて，同じ更新系クエリを実行する．このとき，両データベースにおいて，更新対象の表ごとに更新系クエリの実行順序が同一になるように制御する必要がある．この制御を，実行順序制御と呼ぶことにする．

上記の 2 系統の複製方式を上記の要件(a)(b)(c)(d)に照らせば，以下のようである．

- ジャーナルログ・ベース：(a)改造必須．(b)可能．(c)一つの更新系クエリが多数の行を更新すると，ジャーナルログが増大する．ジャーナルログの反映が逐次的である．(d)実装に依存．
- クエリ・ベース： (a)実行順序制御方式によっては改造不要．(b)実行順序制御方式によっては可能．(c)実行順序制御方式に起因して，同時並行実行可能な更新系クエリ数が制限される．(d)実行順序制御方式と実装に依存．

ジャーナルログ・ベースの複製方式では要件(a)を満たせないために，LC ではクエリ・ベースの複製方式を用いる．次節以降では，LC の複製方式について，上記の要件を満たす実行順序制御方式と，それに基づいた新規 PG 接続方式を示す．

3.2. 実行順序制御方式

PG を束ねるノードを設け，更新系クエリを逐次的に PG に発行するシンプルな方式がある．しかし，集中制御になるために，要件(b)を満たせない．更新系クエリが同時並行実行できないために，要件(c)を満たせない．また，全体的なロックマネージャを設け，クエリ実行に伴う PG の内部ロックの取得順序を同一にする方式がある [8]．しかし，ロック取得の許可を求める処理を PG に付加する必要があるために，要件(a)を満たせない．集中制御になるために，要件(b)を満たせない．

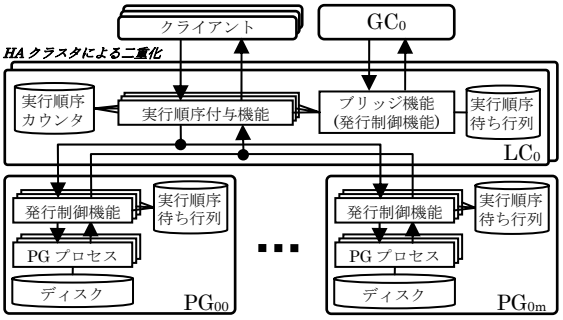

図 2　実行順序制御方式の概要

これに対してLCでは，PGを束ねるノードに置いたプログラムが，更新対象の表ごとに実行順序を更新系クエリに付与し（半順序関係になる），PGに置いたプログラムが，付与された実行順序にしたがって更新系クエリをPGに発行するという分散型の実行順序制御方式を用いる．概要を図 2 に示す．PGはクライアントとの接続ごとにプロセスを起動するために，“実行順序付与機能”と“発行制御機能”をマルチスレッドで実装し，1 接続に 1 スレッドを割り当てる．“実行順序付与機能”は，“実行順序カウンタ”から得た実行順序を更新系クエリに付与し，“発行制御機能”に送信する．“発行制御機能”は，更新系クエリの実行順序を確認し，実行順番であればPGに発行し，実行順番でなければ“実行順序待ち行列”に登録して停止する．PGから更新系クエリの実行結果が返されると，“実行順序待ち行列”を確認し，次の実行順番の“発行制御機能”が停止中であれば起動する．また“実行順序付与機能”は，実行結果の変則多数決と接続状態監視によりPGの障害を検出し，該当PGを切離す．本方式は一種の表ロックであるために，更新系クエリの系列によってはPGの表・行ロックと競合してデッドロックが発生する．そこで，以下のデッドロック制御を“実行順序付与機能”に付加する．

(a) 表・行ロックとの競合対策：タイムアウトによりデッドロックを検出し，実行中の更新系クエリのキャンセルを PG に要求する．

(b) 表ロックとの競合対策：表ロック取得済みの表に対して更新系クエリを実行する場合は，実行順序制御対象から外す．

(c) 行ロックとの競合対策（オプション）：表ロック未取得の表に対して更新系クエリを実行する場合は，表ロックを取得して表ロックとの

競合の問題に帰着させる．これは，クエリに書かれた条件式から行ロックの競合を検出することが困難なためである．

本方式は，PG の改造が不要なために要件(a)を満たす．“発行制御機能”が同時並行動作できるために要件(b)を満たす．要件(c)について，全体的なロックマネージャを設ける方式を用いる製品（PG は 7.3 系）と本方式を用いるプロトタイプ（PG は 7.4 系）とをベンチマーク（単純な参照・単純な参照更新・TPC-C）で比較すると，ベンチマークごとに優劣が異なる．共に参照処理は負荷分散している．その傾向から，処理量は本方式を用いるプロトタイプが有利である．同時並行実行可能な更新系クエリ数は，ベンチマークモデルに依存するロックの競合率と期間，デッドロック制御(c)の設定の有無に依存しており，明確な優劣はない．なお，要件(a)を緩和して小改造を容認するなら，本方式のオーバーヘッドは“発行制御機能”を PG に埋め込むことで削減できる．

3.3. 新規 PG 接続方式

上記の実行順序制御方式を利用すれば，LC の動作中に新規 PG を接続する処理を，クエリの水準で実現できる．PG の改造は不要であり，異なる版の PG にも適用できる．また，同一サイト内の PG だけではなく，異なるサイト間の PG にも適用できる．新規 PG の接続手順を以下に示す．

1. 新規 PG に初期化用クエリを発行する．
2. 既存 PG へのクエリの中継を，トランザクションの境界で一時停止する．
3. 既存 PG へ直列化可能隔離水準のトランザクションを開始するクエリを発行し，データベースファイルのスナップショットを作成する．さらに，既存 PG に発行される更新系クエリの複製をファイルに蓄積する処理を開始する．
4. 既存 PG へのクエリの中継を再開する．ただし，以下の 5，6 の処理が進むように遅延を入れて中継する．
5. 上記 2 で作成したスナップショットを新規 PG に複製する（select で抽出し insert で挿入）．
6. ファイルに蓄積し続けている更新系クエリを，実行順序制御したうえで新規 PG に発行する．
7. ファイルが空になれば，既存 PG へのクエリの中継をトランザクションの境界で一時停止し，既存 PG に発行される更新系クエリの複製をファイルに蓄積する処理を終了する．さらに，ファイルに残っている更新系クエリを，実行順序制御したうえで新規 PG に発行する．
8. ファイルが空になれば，LC の内部状態を初期化して通常運転に移行する．

本方式は「既存 PG のデータを変換したうえで新規 PG に複製し，既存 PG を切り離す」というデータ保守方式に発展させることも容易である．

4. グローバルクラスタ

4.1. 概要

GC は，各サイトに LC と 1 対 1 で存在する．配信元サイトの GC を根として複数のサイトの GC を根から葉への方向を持つ有向木形状に接続し，これを配信木として，配信元サイトの PG の更新内容を，配信木を構成する全てのサイトの PG にマルチキャスト配信する．配信木は，PG の表ごとに作成する．MLC が適用される広域環境において上記のような DB 複製用のマルチキャストを実現するには，以下の課題がある．

(a) 配信元サイトの LC から受信した更新内容を，受信した順序のまま配信木に接続された全てのサイトの LC に素早く配信する．
(b) 配信元サイトを任意のサイトに動的に切り替え，任意のサイトから更新内容を配信する．
(c) 配信木を構成する GC や通信路の故障により更新内容を配信できない場合に，代替手段を用いて確実に配信する．

4.2. DB 複製用マルチキャスト方式

図 3 にGCの構成例を示す．図 3 を用いて，課題(a)(b)(c)を解決できるアプリケーションレイヤのDB複製用マルチキャスト方式を示す．

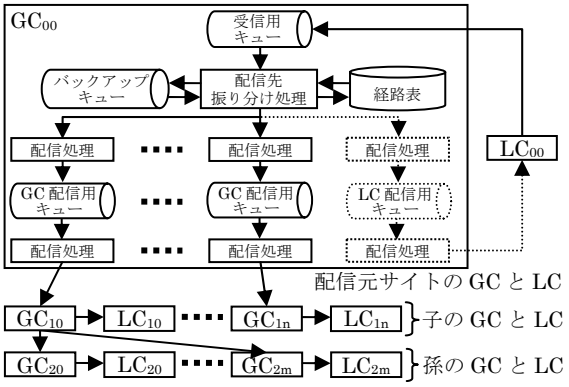

図 3 GC の構成例

課題(a)を解決するには，各 GC において，自サイトの LC と子の GC への配信を逐次処理すればよい．しかし，複数のあて先に対する配信を逐次処理すれば，配信性能が問題になる．そこで“受信用キュー”・“LC 配信用キュー”・子の GC ごとの“GC 配信用キュー”・あて先ごとの“配信処理”を用いて，更新内容の順序関係を維持した

まま更新内容の配信を並列処理することで，配信性能を向上させる．配信手順を以下に示す．

1. 配信元サイトの GC は，自サイトの LC から“受信用キュー”で受信した更新内容を“GC 配信用キュー”へ同時並行して配信する．さらに，子の GC の“受信用キュー”へパイプライン的に配信する．
2. 子の GC は，親の GC から“受信用キュー”で受信した更新内容を“LC 配信用キュー”と“GC 配信用キュー”へ同時並行して配信する．さらに，自サイトの LC と孫の GC の“受信用キュー”へパイプライン的に配信する．
3. これを葉の GC に至るまで繰り返す．

さらに，上記の各キューと“配信処理”の組を複数用いて，配信木を単位として多重処理することで配信性能を向上させる方法も考えられる．

課題(b)を解決するには，更新内容の配信を完全に停止してから，配信元サイトのGCが所有している配信権を，配信権を要求する任意のサイトのGCに移譲すればよい．しかし，更新内容の配信を完全に停止すれば配信性能が問題になる．そこで，アクティブネットワーク[8]の考え方に基づいて，配信権の移譲を指示するコマンド（配信権移譲コマンド）を更新内容の配信経路と同じ配信経路で配信し，配信権移譲コマンドにしたがった配信権の移譲と，配信権移譲コマンドを受信するまでに受信した更新内容の配信を，同時並行処理する．図 4 を用いて，配信権の移譲手順を示す．

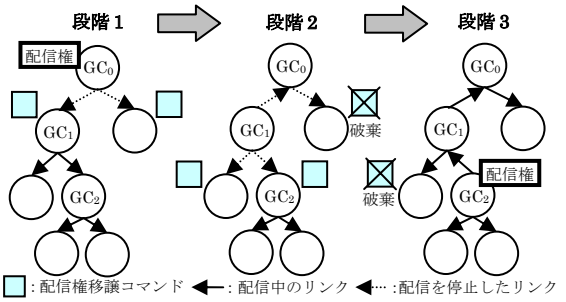

図 4　配信権の移譲手順の例

1. 更新内容を配信したい GC_2 が，配信権の移譲を DC に要求する．DC は，配信権を持つ GC_0 に配信権移譲コマンドを配信する．GC_0 は，配信権移譲コマンドを受信すると，配信権を破棄し，更新内容の配信経路と同じ配信経路で子の GC に配信権移譲コマンドを配信する．
2. GC_1 などの GC_0 と GC_2 を結ぶ配信経路上の GC が配信権移譲コマンドを受信すると，親の GC との間の配信木の弧を逆向きにし，子の GC に配信権移譲コマンドを配信する．GC_1 と GC_2 を結ぶ配信経路上にない GC が配信権移譲コマンドを受信すると，これを破棄する．
3. GC_2 は，配信権移譲コマンドを受信すると，親の GC との間の配信木の弧を逆向きにし，即座に配信を開始する（配信権の移譲完了）．

課題(c)について，DB 複製用マルチキャストでは，配信する更新内容の欠損が許されない．そこで，一時的な故障に対しては，更新内容を単位として再送を試みる．恒久的な故障に対しては，配信木を再構成し，欠損した更新内容を再送する．この欠損対策の手順を以下に示す．

1. 子の GC への配信に失敗した場合は，“GC 配信用キュー”に蓄積していた更新内容を保持したまま更新内容の再送を試みる．
2. 規定回数まで再送しても配信できない場合は，“GC 配信用キュー”に蓄積していた更新内容を消去し，配信できない子の GC（故障 GC）の切り離しを DC に要求する． DC の指示にしたがい，故障 GC の親と子の GC をバイパス接続して配信木を再構成する．
3. バイパス接続された親と子の GC がこの時点までに受信した更新内容には，差分（欠損）がある可能性がある．そこで子の GC は，配信木ごとの自身が受信した最新の更新内容の識別子を親の GC に通知する．親の GC は，通知された識別子と自身が受信した最新の更新内容の識別子とを比較する．一致しない場合は，更新内容の差分を“バックアップキュー”から子の GC に配信する．この一連の処理をリカバリ処理と呼ぶ．
4. リカバリ処理中に配信元サイトの GC から新たな更新内容が配信されてきた場合は，親の GC の“受信用キュー”に蓄積し，リカバリ処理が完了してから配信する．

4.3. 配信特性

プロトタイプを作成して，上記のDB複製用マルチキャスト方式の配信特性を評価した．プロトタイプでは，LCとGCをストリームソケットで接続し，GCとGCをHTTP経由のJMSで接続している．今回の評価では，配信元サイトのGCに，深さを優先して直列に 1～3 個の子孫のGCを接続した配信木と，幅を優先して並列に 1～3 個の子のGCを接続した配信木とを対象として，1, 10, 100, 1000 件の更新内容を配信したときの平均配信時間を測定した．配信時間の測定区間は，配信元サイトのLCが，自サイトのGCに更新内容を配信してから，配信元サイトのGCの子のGCが，

自サイトのLCに更新内容を配信し終わるまでである．並列にGCを接続する場合には複数の測定区間が存在するが，測定結果に差はなかった．測定結果を図 **5** に示す．

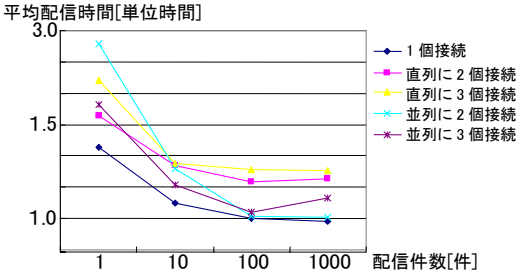

図 5　配信木別の平均配信時間

各配信木とも，配信件数の増加に伴い平均配信時間が減少している．これは，更新内容の配信をパイプライン処理している効果であると考えられる．また，GC の接続個数を並列に増やしても平均配信時間はさして増大していない．これは，更新内容の配信を同時並行処理している効果であると考えられる．

5.　分散配信制御モジュール

5.1.　概要

DC は，GC が更新内容とコマンドの配信に用いる配信木を管理する．以下の操作を行なう．

(a) 配信木作成操作：GC からの参加・脱退要求に応じて配信木トポロジを計算し，関連する GC に計算結果を通知する．

(b) 配信権移譲操作：GC からの配信権の移譲要求に応じて配信木トポロジを計算し，関連する GC に計算結果を通知する．

MLC が適用される広域環境において操作(a)(b)を 1 台の DC で行うと，以下の問題が生じる．

- 規模の拡大に伴う負荷・遅延の増加に対応できない．
- DC が故障すると配信木の管理情報が完全に消失するが，この影響範囲が大きい．

これらの問題を解決するために，DC では以下の方針を用いる．

(a) Round Trip Timeなどに基づいたネットワーク距離にしたがい，GCを複数のエリアに分割する．エリアごとにDCを配置し，GCを分割管理することで負荷分散する．図 **6** 参照．

(b) GC が DC の故障を検出した場合に，故障した DC の担当エリアを隣接エリアの DC が引き継げるように，全ての DC が等しく配信木の管理情報を持つ．

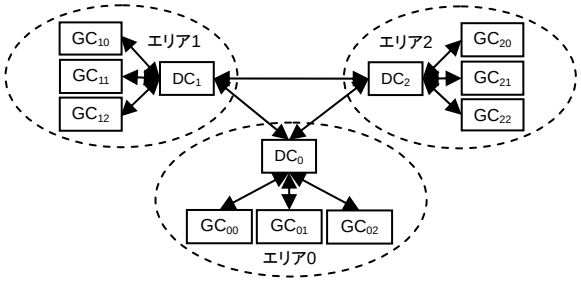

図 6　DC の構成例

次節以降では，上記の方針(a)(b)に基づいた上記の操作(a)(b)を示す．

5.2.　配信木作成操作

3 段階からなる配信木作成操作の内容を，以下に示す．

1. 配信木トポロジを計算：方針(a)にしたがって，担当エリア内の配信木の部分木のトポロジを個別に計算する．加えて，トポロジの変化が最小になるように計算する．このような計算方針をとれば，各エリア内の配信木の部分木の最適化しかできず，配信木全体の最適化ができないために，配信効率が低下する．しかし，DB 複製用マルチキャストでは配信する更新内容の欠損が許されないために，トポロジを変更するとリカバリ処理が必要になる．リカバリ処理は更新内容の配信を一旦停止させるために，配信効率を低下させる．したがって，上記の計算方針が一概に配信効率を低下させるとはいえない．さらに，適切にエリアを分割することで，ある程度の配信効率が見込める．なお，エリア間を結ぶ場合は，配信木の葉に近いエリアの DC から，配信木の根に近いエリアの DC に，GC からの参加・脱退要求を中継し，中継先の DC が計算する．
2. 配信木の管理情報を同期：方針(b)にしたがって，計算結果を相互に反映する．各エリアの DC は担当エリアに閉じて配信木の部分木のトポロジを計算するために，計算結果を相互に送信すれば，計算結果を各々が持つ配信木の管理情報に単純にマージできる．エリア間を結ぶ場合も同様である．
3. 担当エリア内の GC に計算結果を通知：担当エリア内の配信木の部分木の根の GC に，計算結果を含む配信木変更コマンドを配信する．配信木更新コマンドは，GC の“経路表”を書き換えながら，新旧の配信木に沿って流れる．要求元の GC は，配信木変更コマンドの到達により操作の終了を知る．

5.3. 配信権移譲操作

配信権移譲操作は影響範囲が配信木全体に及ぶ可能性があるために，方針(a)にしたがって各エリアの DC が配信木トポロジを個別に計算した結果を，方針(b)にしたがって相互に反映しようとしても，単純にはマージできない．1 度に 1 台の DC で計算する必要がある．この DC を決定するには，公平な調停が必要である．公平に調停するには，以下の課題に留意する必要がある．

(a) 要求の発生順番どおりに割り当てる．

(b) 常に要求を受付ける．

調停方式として 2 相ロック方式[9]が考えられる．しかし，課題(a)について，拡張フェーズが衝突しなければ保証されるが，衝突すれば保障されない．加えて，要求数が増加するほど衝突しやすくなる．課題(b)について，配信権の移譲要求をキューイングするなどの処理の併用が必要である．これに対して，DC ではシンプルなキューイング方式を用いる．この方式に用いた 3 段階からなる配信権移譲操作の内容を，以下に示す．

1. 配信木トポロジを計算：配信木の配信元サイトの GC を配下に持つ DC が，その配信木の配信権の移譲要求をキューイングするとともに，配信木トポロジを計算する．配信木単位に分割管理するという意味において，方針(a)にしたがった方式である．また，配信権の移譲要求をキューイングすることで，課題(a)(b)を保証できる．
2. 配信木の管理情報を同期：方針(b)にしたがって，計算結果を相互に反映する．しかし，配信権移譲操作の間にも，各エリアの DC が配信木作成操作のために配信木トポロジを個別に計算している．このために，計算結果を各々が持つ配信木の管理情報に単純にはマージできない．そこで，配信木作成操作の計算結果を反映する場合には，最初に配信元サイトの GC を配下に持つ DC に反映を要求し，計算結果の反映要求の衝突を検出する．衝突を検出した場合には，配信木作成操作の計算結果の反映要求を，配信権の移譲要求と共にキューイングする．配信木作成操作の計算結果の反映要求の順番がくると，配信木作成操作を再実行させ，計算結果を反映する．なお，操作の性質から衝突は少ないと思われる．
3. 配信木を構成する GC に計算結果を通知：旧配信元サイトの GC に，計算結果を含む配信権移譲コマンドを配信する．配信権移譲コマンドは，GC の“経路表”を書き換えながら，旧配信木に沿って流れる．要求元の GC は，配信権移譲コマンドの到達により操作の終了を知り，自エリアの DC に配信権の獲得を通知する．この DC は，キューイングされている要求を引き継ぎ，配信元サイトが変更されたことを全ての DC に通知する．

6. おわりに

本稿では，インターネットワイドに分散した多数のサイトが連動する広域分散システムの基盤として利用できる 3 層構造を持つデータベースクラスタ（MLC）を提案した．その特徴は，分散型の実行順序制御方式に基づくクエリ・ベースのデータベース複製方式（第 1 層：LC），アクティブネットワークの考え方に基づく DB 複製用マルチキャスト方式（第 2 層：GC），エリア分割・情報冗長化の考え方に基づく配信木の管理方式（第 3 層：DC）である．既にプロトタイプを作成して基本動作を確認している．LC については，配下の PG に表を重複かつ分散配置し，更新処理を負荷分散することも考慮している．GC については，ある配信木の葉から根に情報を収集し，その情報を別の配信木の根から葉へ配信することも考慮している．今後はプロトタイプを拡充し，障害発生時の 3 層間の連携動作・トポロジ変化が最小という条件のもとでの最適な配信木トポロジの計算方法・性能の検討を進めるとともに，事例検討により適用性を評価する．

参考文献

1) 防災情報提供センタ．http://www.bosaijoho.go.jp/
2) 白石　陽：センサーネットワークのためのデータベース技術，情報処理学会学会誌，Vol.47,No4,pp387-393 (2006)
3) 国土交通省　国土技術総合研究所：次世代道路サービス提供システムに関する共同研究報告書，http://www.nilim.go.jp/japanese/its/1top/kyouken/index.html (2006)
4) 目黒浩一郎：ITS/カーエレクトロニクス技術の現在，情報処理学会学会誌，Vol.47,No7, pp748-754 (2006)
5) g コンテンツ流通推進協議会．http://www.g-contents.jp
6) 石田　了：デジタルシティの現状，情報処理学会誌，Vol41, No.2, pp.163-168 (2000)
7) United States Patent Application 20040030739
8) 山本　幹：アクティブネットワークの技術動向，電子情報通信学会論文誌 B，Vol.J84-B No.8, pp.1401-1412 (2001)
9) Bacon, J.：並行分散システム，トッパン (1996)