4P－3

フォールトトレラントに対応したＤＳＭ機構に関する一考察

荒木　信行，　林　徹，　東　潤一郎

ＮＴＴコムウェア株式会社

1. はじめに

近年、PC の低価格化及びノード間ネットワークの高速化により、大型コンピュータ相当の性能を実現するようなクラスタリングシステムが注目されている。そのために必要な機構のひとつに分散共有メモリ（Distributed Shared Memory、以降 DSM）機構があり、性能の向上とコヒーレンシの維持を中心に研究されている。しかしこのＤＳＭ機構は、メモリ情報を分散して持つため、任意のノードダウンがシステム全体のダウンに繋がることから信頼性が低い。

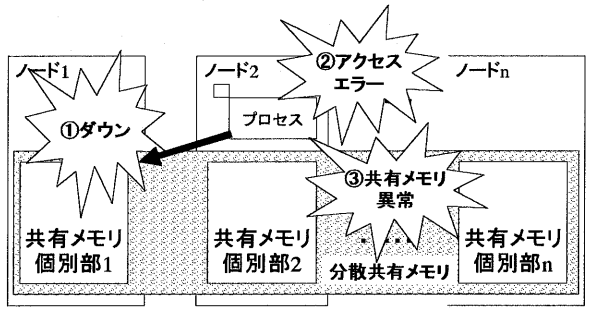
本稿では、DSM 機構を実装した並列処理型のクラスタリングシステムの信頼性を高めるため、フォールトトレラントに対応した DSM 機構に関する一考察を報告する。

2. DSM機構の課題

2. 1　現状の問題点

ＤＳＭ機構はクラスタリングする複数ノードに共有メモリを分散して配置し、使用する際に自ノードのキャッシュ領域にデータをコピーして使用する方法が取られている。しかし、このような構造では、共有メモリ情報を持つノードに何らかの障害が発生すると、そのノードに配置されていた共有メモリ情報を使用することができなくなるため、システム停止に繋がる恐れがある（図１）。

また、データベース（DBMS）を使用した OLTP プログラムを例に考えてみると、処理の途中で DSM が消滅することによってプログラム側への通知結果（OK）とデータベースの内容にデータの不整合が生じてしまう（図２）。



①ノードが何らかの原因でダウンすると、②ノード1の所有する共有メモリをアクセスしようとするプロセスはエラーとになるとともに、③共有メモリ管理異常となる

図１　ノードダウンによるシステム異常例

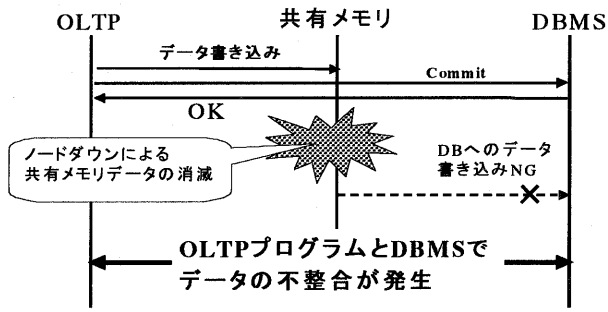


図２　データ不整合の発生例

2. 2　フォールトトレラント対応方法

ノードダウンによる問題が発生した場合でも継続動作を保障するためには、共有メモリのバックアップを配置するとともに、クラスタリングされたシステムからノードダウンを速やかに検出し、ダウンしたノードの切り離し、バックアップメモリを持つノードへ切り替えることが必要である。DSM 機構としてこれらを考える場合、クラスタリングによるシステムの特性から、共有メモリの有効利用を目的としたバックアップデータの適切な配置とバックアップ契機と、DSM 機構と連動した障害検出とリカバリ処理について考える必要がある。

A consideration of DSM structure adjusted to fault tolerant
Nobuyuki ARAKI, Tooru HAYASHI, Junichirou AZUMA
NTT COMWARE CORPORATION
1-6 Nakase Mihama-ku Chiba-shi, Chiba 261-0023 Japan

3．考察

3．1　バックアップデータの配置とバックアップ契機

バックアップデータの配置に関しては、既存のDSM 機構に機能追加する考えから、共有メモリを複数のノードに配置する方式（案1）と、各ノードの共有メモリを分割して複数ノードに配置する方式（案2）が考えられる。信頼性、性能及びメモリ資源の有効利用という3点から考察した。

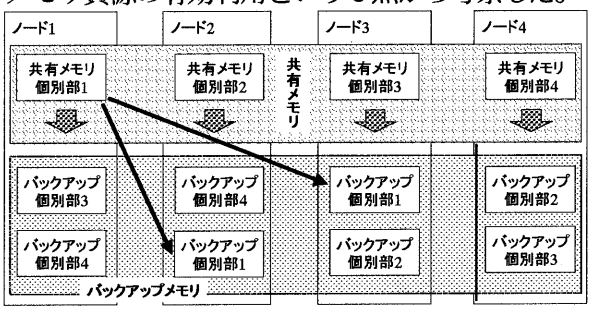


図3　共有メモリ配置構成（案1）

複数ノードに配置する方法（案1）は、バックアップ数を多くするほどメモリ消費量が増大する。性能面でも、通信量の増大による性能劣化を招く。信頼性の面は、バックアップしたノード数の同時ダウンに対応できる。

分割して複数ノードに配置する方法(案2)は、分割しないよりも均等にメモリを使用するため、メモリの集中使用による単一ノードのメモリ不足を回避できる。しかし、性能面では案1に比べ分割して通信ノードが増加する分、管理上のオーバーヘッドの増大を招き、信頼性の面では分割したものの一部を失うとに復旧できないため、1ノードのダウンにしか対処できない。

以上より、バックアップの配置は、案1（図3）の方法が複数ノードのダウンに対応できるため信頼性が高く、フォールトトレラントに適していると言える。性能やメモリ資源とのバランスを取ってバックアップするノード数を設定すれば、複数ノードのダウンに対応できる。

また、バックアップの契機については、データを失う可能性を極力減らすため、共有メモリアクセスと同時にバックアップを行う。ただし、ノード間のネットワーク遅延間のダウンに対するリカバリは出来ていないため、検討の必要がある。

3．2　障害検出とリカバリ処理

障害ノードの検出は、共有メモリやバックアップのアクセス時のデーモン間通信で、検出することが可能である。しかし、ノードダウンは共有メモリへのアクセス時に限らず、ハードウェアの故障及びOS を含むソフトウェアの異常によるノード停止や通信断等も検出することが必要であることから、共有メモリへのアクセスがない場合でも定期的にデーモン間でノードの正常性を確認する機構が必要である。

次にリカバリ処理は、ノードダウンがシステムダウンへ繋がらないようにするため、ノードの切り離し後も、共有メモリアクセスが継続できる機構が必要である（図4）。

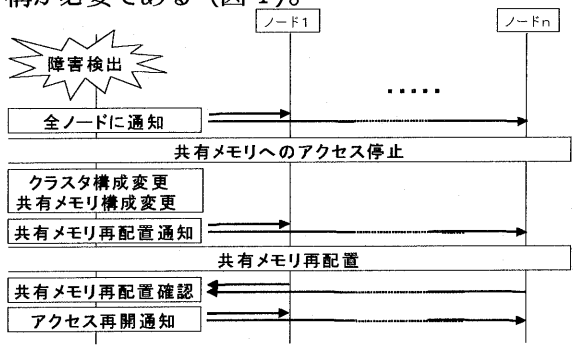


図4　リカバリ処理シーケンス

障害検出ノードは、全ノードへアクセス停止を通知する。障害ノードのバックアップを持つノードのうちリカバリを行うノードがリカバリ処理を管理し、複数ノードの同時ダウン確認も含めた共有メモリの再配置確認、アクセス再開の通知を行う。複数ノード同時ダウンの時には、リカバリを行うノードが各々のリカバリ処理を管理する。

もともと動作していたプロセスについては、アプリケーションに完了が返った時点に共有メモリ情報をロールバックする。

また、データベースを使用したOLTP ならば、バックアップへの書き込みはトランザクション単位で行うため、リカバリについてもジャーナルファイルなどは用いず、トランザクション単位でロールバックして対処する。

4．おわりに

本稿では、フォールトトレラントに対応したDSM 機構の実現を目的に考察を行った。今回の検討では、クラスタリングシステムとして信頼性を確保できることがわかった。

今後は、課題となっているバックアップデータの保証及びリカバリ、性能の面からフォールトトレラント対応について検討していく。