

過負荷によるノード切離しを抑制する障害監視機構

小比賀 亮仁, 佐々木 盛朗, 菅原 智義, 田中 淳裕

日本電気株式会社 システムプラットフォーム研究所

A failure detection mechanism to control node isolation

Akihito KOHIGA, Shigero SASAKI, Tomoyoshi SUGAWARA and Ashuhiro TANAKA

System Platforms Research Laboratories, NEC Corporation

1 はじめに

本稿では、「過負荷によるノード切離しを抑制する障害監視機構」を提案する。従来のハートビートを用いる障害監視機構は、監視対象となるノードが過負荷になり、ハートビートの送信に遅延が発生した場合に、当該ノードを切離すため、クラスタの性能が急激に劣化するという問題を持っていた。我々は、過負荷状態は処理が進むと解消され、通常の状態に戻るという点に着目し、過負荷状態にあるノードの切離しを抑制する障害監視機構を提案する。本機構により、クラスタの急激な性能劣化を防ぐことができる。

2 従来の障害監視機構における問題点

我々は、ノード数が1000台以上の大規模クラスタシステムを想定し、スケーラビリティがあり、かつ信頼性の高いクラスタシステムを実現するための研究を行っている。その中でも、障害監視機構は、システムの停止時間を短縮するための重要な機能の一つである。

クラスタにおける障害監視機構の役割は、障害の発生しているノードを検出すること、および当該ノードを除去することである。クラスタシステムにおける従来の障害検出方式としては、モニタリング、またはハートビートを使用する方式がある。

モニタリングとは、クラスタ管理ノードがCPUやメモリ使用率など、各ノードの詳細な内部情報を取得することを指す。モニタリングは、ノードの詳細な情報を取得しているため、障害原因の特定に役立つが、監視対象となるノード数が増えると、ネットワークや、クラスタ管理ノードの処理を圧迫することになる。ganglia[1]は、クラスタ管理ノードを階層化することによって、スケーラビリティの高いモニタリングを実現している。

ハートビート[2]とは、各ノードからクラスタ管理ノードに送信される数バイト程度のパケットであり、クラスタ管理ノードは、各ノードから定期的にハートビートを受け取ることによって、各ノードの障害を検出する。ハートビートは、高々数バイト程度のパケットを送受信するだけなので、ネットワークやクラスタ管理ノードの処理を圧迫することはない。

障害の除去には、フェイルオーバを用いる。フェイルオーバとは、冗長ノードを用意し通常利用しているノードに障害が発生した場合に、冗長ノードに通常ノードの処理を引き継がせることを言う。

大規模クラスタシステムでは、障害検出にハートビート、障害の除去にフェイルオーバを用いる。これには、次の理由がある。ノードを豊富に所有する大規模クラスタは、つねに予備ノードを配置することができ、何らかの障害の疑いのあるノードは直に切離して、新しいノードを追加する。障害の疑いのみを検知するのであれば、モニタリングのようにノード内部の詳細な情報は必要なく、ハートビートによる検知機能が適している。

しかし、従来のハートビートを用いる障害監視機構には、クラスタが過負荷状態に陥ると、ハートビートが遅れ、ノードの切離しが頻発し、クラスタの性能が急激に劣化するという問題がある。理由は以下の通りである。

ユーザからのリクエストは複数のノードから構成される処理グループによって処理され、レスポンスが返される。ユーザからの急激なリクエスト増加によって、処理グループが過負荷状態になると、ハートビートの送信が遅れる。そして、クラスタ管理ノードは、ハートビートの遅れているノードを切離すことになる。ノードが切離された後は、処理グループへノードが新たに追加される。

ここで、ノードが切離されてから、新しいノードが追加されるまでの間の負荷の変動を考える。

切離されたノードで処理していたリクエストは、ユーザからの再リクエストという形で新たな負荷となる。さらに、ノードが追加されるまでの時間内に到着したリクエストは、その他の切離されていないノードで処理しなければならない。例えば、 N 台のノードが、切離しによって $N-1$ 台になったとすると、到着する N 台分のリクエストを $N-1$ 台のノードで処理することになる。よって、ユーザからの再リクエストによる負荷と、切離したノードに到着していた従来の負荷が、生き残った他のノードに上乘せられることになる。これにより、さらにノードの切離しが発生し、クラスタ(正確には処理グループ)の性能が急激に劣化することになる。

3 ノードの切離しを抑制する障害監視方式

我々は、過負荷によるノードの切離しによってクラスタ性能が急激に劣化するという問題を解決するために、過負荷状態にあるノードを完全には切離さない方式を提案する。完全には切離さない方式とは、新規のリクエストは受け取らず、処理中のリクエストに対してのみレスポンスを返す方式である。この方式は、処理グループとは別に「診断グループ」を導入することによって実現する。仮に、診断グループに所属するノードがリクエストを返さなければ、ユーザが再びリクエストを出すことになるので、ノードを切離したのと同じ状態を作ってしまう。負荷の上昇をできるだけ抑えるために、診断グループに所属するノードもリクエストを返す必要がある。

過負荷は障害ではない。過負荷は処理が進むにつれて解消され、通常の状態に戻る。しかし、ノードの切離しを遅延させただけでは、当該ノードにリクエストが到着し続け、障害と過負荷状態の切り分けが正確に行えない。ハートビートの送信に遅延があったノードを診断グループに移動させることによって、障害と過負荷状態の切り分け精度を高めることができる。

図1は、我々の設計した障害監視方式を示す概要図である。

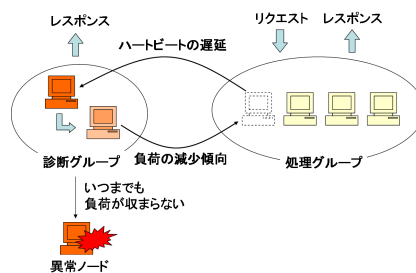


図1: 切離しを抑制する障害監視機構

4 設計と試作

4.1 システム構成

前節で説明した障害監視方式を、クラスタ管理サーバ、負荷分散サーバ、障害監視サーバ、障害監視クライアントから構成される障害監視機構として実現した。図2に、我々の提案する障害監視機構の全体構成を示す。各ノードはLANによって接続され、相互に通信を行う。クラスタ管理サーバは、クラスタ内の全てのノードの状態を管理し、負荷分散サーバの要求に対して、待機中のノード群から処理グループを構成する。負荷分散サーバは、クラスタ管理サーバから処理グループの構成情報を受け取り、ユーザからのリクエストを処理グループ内の各ノードに配布する。クラスタ管理ノードには、障害監視サーバが配置さ

れ、リクエストを処理するノード（図中の処理ノード A および処理ノード B）には障害監視クライアントが配置される。障害監視クライアントは、ハートビートメッセージを一定時間毎に障害監視サーバへ送信する。

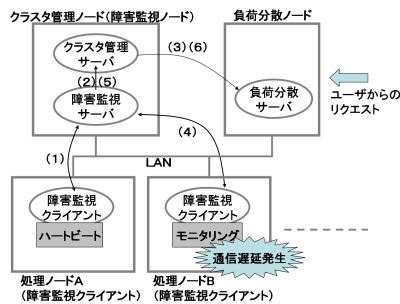


図 2: システム構成図

4.2 動作順序

前節で説明した障害監視機構は、次のように動作する。箇条書き番号は、図 2 に示される番号に対応する。動作のポイントは、ハートビートの遅延を契機に、グループの組み替えを行うところ、診断グループにて負荷の減少を把握するために、モニタリングを実行しているところである。現在は CPU 使用率のみを負荷減少の指標としているが、その他にも、メモリの利用状況や、その他のハードウェア資源の利用状況を観察することによって、より精度の高い障害と過負荷の切り分けを実現することができる。

- (1) 障害監視クライアントは、障害監視サーバに向けて一定時間毎にハートビートを送信し続ける。
- (2) 障害監視クライアントからのハートビートの受信に遅延が発生したら（図中、処理ノード B）、障害監視サーバは、クラスタ管理サーバに対して、当該処理ノードを診断グループへ移動させるように命令する。
- (3) クラスタ管理サーバは、当該処理ノードを診断グループへ移動させ、処理グループのグループ構成に変更があったことを負荷分散サーバに報告する。負荷分散サーバは、報告のあった処理ノードに対するリクエストの配布を停止する。
- (4) 障害監視サーバは、当該処理ノードに対して、モニタリングを開始させると同時に、一定時間毎に当該処理ノードの負荷状況を確認する。ここでの負荷とは、CPU 使用率を指す。モニタリング情報の送受信にタイムアウトは設定せず、取得できた情報から当該処理ノードの負荷状況を推測する。
- (5) 一定時間後に、負荷の減少が見られた場合は、当該処理ノードを元のグループへ移動させるようにクラスタ管理サーバに対して命令する。一定時間後も負荷の減少が見られないようであれば、障害が発生したとして同ノードを診断グループから切離すようにクラスタ管理サーバに対して命令する。
- (6) クラスタ管理サーバは、障害監視サーバからの命令を受け取り、診断グループから元のグループへ当該処理ノードを移動させる。そして、処理グループのグループ構成に変更があったことを負荷分散サーバに報告する。障害監視サーバからの命令が、診断グループからの切離しであれば、負荷分散サーバへの報告は行わない。

5 評価

我々は、従来のハートビートを用いた障害監視機構と比較し、提案方式がクラスタの急激な性能劣化を抑制していることを確認するために、クラスタに負荷をかけた時のスループットの経過を測定し、評価した。スループットは、単位時間あたりのレスポンス数としている。

評価環境を表 1 に示す。表に示すノードを障害監視ノード（1）、負荷分散ノード（1）、処理ノード（5）、負荷発生ノード（1）に割り当てた。括弧内の数字は、ノード数である。そして、1 バイトのハートビートを 0.5 秒間隔で送信している。

図 3 に評価結果を示す。図 3 に示されるグラフは、それぞれ、リクエストの到着傾向、既存のハートビートによる障害監視機

表 1: 評価環境

CPU	Xeon 2.4GHz × 8
メモリ	1GBDDR266 SDRAM,FSB533MHz
NIC	Intel PRO1000/TX (最大転送速度 1Gbps)
OS	Redhat Linux7.3 kernel-2.4.20-28
NW スイッチ	Extreme Summit 7i (1 ポートあたりの最大転送速度 1Gbps)

構、我々が設計・試作した障害監視機構のスループットの経過を示している。負荷発生プログラムは、TPC-W[3] に記載されている電子書店のウェブサイトにかかる負荷をモデル化したものを作成し利用している。最初は、1 秒間に 48 リクエストを発生させ、30 秒毎に 2 倍にし、それを 4 回繰り返す。最終的には、1 秒間に 192 リクエストが発生することになる。ただし、今回の評価では、ノード切離し後のリクエスト再送は行っていない。

グラフを見ると、従来方式では、負荷を発生させてから 87 秒後にスループットの低下が始まり、それから 5 秒で一気に性能が低下しているのに対し、我々の提案方式は急激なスループットの低下が見られることはなかった。

以上の結果から、過負荷によるノードの切離しを抑制する我々の障害監視機構は、既存のハートビートを用いた障害監視機構と比較して、過負荷状態においてもクラスタの性能を急激に低下させないことを確認できた。

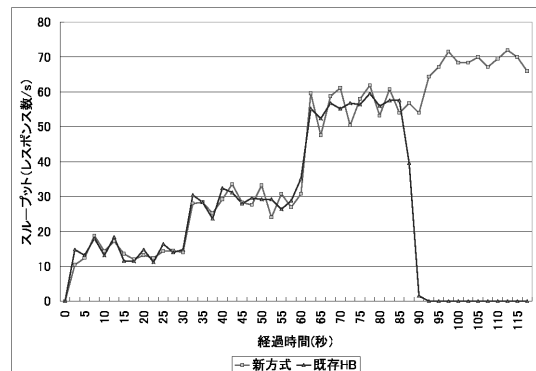


図 3: 従来方式と提案方式の負荷に対するスループットの変化

6 おわりに

本稿では、ハートビートを用いた「過負荷によるノード切離しを抑制する障害監視機構」を提案し、設計と試作について述べた。従来のハートビートを用いた障害監視機構と我々の提案した障害監視機構を比較評価をおこなうことによって、過負荷状態においてもクラスタの性能を保持し続けるという利点があることを示した。本提案方式により、ユーザからの急激なリクエストの増加が発生しても、クラスタ性能の低下を引き起こすことなく、負荷の増加に追従することが可能になる。

7 謝辞

本研究は、独立行政法人 新エネルギー・産業技術開発機構 基盤技術研究促進事業（民間基盤技術研究支援制度）の一環として委託を受け実施している「大規模・高信頼サーバの研究」の成果である。

参考文献

- [1] Matthew L.Massie, Brent N.Chun, David E.Culler. The ganglia distributed monitoring system: design, implementation, and experience, *Parallel Computing*, vol.30 pp817-840,2004
- [2] Marin BERTIER, Oliver MARIN, Pierre SENS. Implementation and performance evaluation of an adaptable failure, *Dependable Computing and Communications*, 2005.
- [3] TPC Benchmark W Standard Specification Version 1.8, http://www.tpc.org/tpcw/spec/tpcw_V1.8.pdf,Feb 2002.