

MPI/SP における異常処理機構の設計

5K-3

落合 真一、村山 和宏、山口 義一

三菱電機(株) 情報技術総合研究所

1. はじめに

我々は大規模なデータ演算/信号処理を可能とする計算機クラスタの構築を目指している。このクラスタにおけるプロセッサ間通信のソフトウェア基盤として、従来の MPI [1] をベースに信号処理に適した通信 API を提供する MPI/SP (MPI for Signal Processing) を開発中である。

ターゲットシステムの要求の一つとして、クラスタの一部に障害が発生した場合にも、システム全体を停止することなく運用を継続し、復旧できることが求められている。本稿では計算機クラスタの高可用性、高信頼性の実現を目的として設計した MPI/SP の異常処理機構について述べる。

2. システム構成

ターゲットシステムは、データ処理用に N 枚の CPU ボードを持つ Compact PCI (CPCI) ユニットのネットワークで接続したボードクラスタ構成をとる (図 1)。各 CPU はネットワーク I/F を持ち、これによりデータ処理ネットワークを構築する。

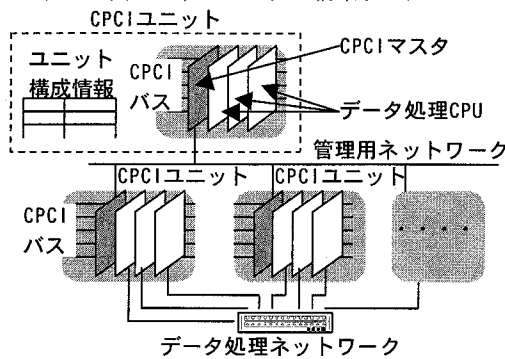


図 1: マルチユニット構成の CPU ボードクラスタ

各 CPCI ユニットにはバスマスタとなる CPCI マスタを持つ。CPCI マスタはユニット内の CPU 状態を管理する。ユニット間で構成情報を交換するために、データ処理用ネットワークと別に CPCI マスタを接続した管理用ネットワークを構築する。

3. 異常処理機構の設計

MPI/SP の異常処理機構は、MPI/SP から取外し可
The design of fault detection and recovery mechanism on MPI/SP
Shinichi OCHIAI, Kazuhiro MURAYAMA, Yoshikazu YAMAGUCHI
Mitsubishi Electric Corporation

能な拡張部として設計した。特に、異常処理機構を意識せずにアプリケーションの設計が行えることを設計の方針とした。

異常処理機構では次の機能を実現する。

- 異常検出: データ処理 CPU の障害を検出する。
- 異常伝達: 検出した障害情報を全ユニットに伝達する。
- 異常対処: 障害情報に基づき、MPI 処理およびボードレベルの制御を行う。

以下の節でこれらを実現する設計について述べる。

3.1. システム構成管理

MPI/SP では通信路ごとにクラスタを構成し、複数のクラスタを結合したシステム構築を可能にしている [2]。これを利用してシステム管理用に、各ユニット内の CPCI バスをネットワークとしたクラスタと、全 CPCI マスタを結ぶ管理ネットワークを使用したクラスタを作る。CPCI バスクラスタを使い、各 CPCI ユニットに属する CPU の構成情報を収集する。ユニット構成情報の内容を表 1 に示す。

表 1: ユニット構成情報

CPU ID	CPCI 通信アドレス	データ処理ネットワーク通信アドレス	CPU 状態 (正常/異常/停止/不在)
UNIT1_SLOT1	0x82000000	192.168.128.1	正常
UNIT1_SLOT2	0x84000000	192.168.128.2	正常
...

管理ネットワーククラスタを使い、ユニット構成情報を CPCI マスタ間で交換することにより、システム内の CPU 状態を監視し、異常に対する制御を行う。

3.2. 異常検出/伝達

異常の検出は次の 3 つの方法により行う。

(1) ユニット内ハートビート

CPCI バスクラスタを使い、ユニット内の CPU が CPCI マスタに対して定周期通信を行うことにより、CPU の異常を検出する。

(2) CPCI マスタ間ハートビート

管理ネットワーククラスタを使い、全ユニットの CPCI マスタ間で定周期通信を行うことにより、CPCI マスタの異常を検出する。

(3) MPI エラー

MPI 処理で検出した通信エラーを、異常としてユニット構成情報に登録する。

各ケースにおける異常検出とその伝達を図 2 に示

す。検出した異常は障害 CPU を持つユニットの CPCI マスタに伝達され、ユニット構成情報を更新する。さらに CPCI マスタはユニット構成情報の更新を全 CPCI マスタに通知する。

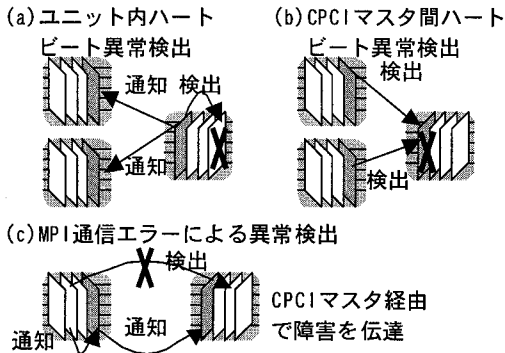


図 2: 異常の検出と他ユニットへの伝達

3.3. 障害対処機能

(1) データ処理 CPU 障害の場合の障害対処

CPCI マスタは、ユニット内の障害が発生した CPU ボードの単体リセットを行い、CPU 再起動を試みる。再起動に成功した場合はユニット構成情報を更新し、MPI 通信グループに再参加させる。システムの初期起動時に、起動完了通知がこなかった CPU に対しても同様な制御を行い、復旧を試みる。指定回数再起動を試みても障害 CPU が復旧しなかった場合は H/W 異常と判断し、その CPU を停止させ、ボード交換指示を表示する。CPU ボードの交換は活線挿抜により行い、ユニットの他 CPU の処理は継続させる。

(2) CPCI マスタ障害の場合の障害対処

CPCI マスタに障害が発生した場合は、ユニット全体の障害とし、遠隔リセット装置によりユニットリセットを行う。ユニットの再起動完了は、管理ネットワークへのユニット構成情報の通知と、CPCI マスタ間ハートビートの再開で検出する。

異常処理機構の動作を表 2 にまとめる。

3.4. MPI との統合

前節の異常処理動作を次のように MPI に統合する。

(1) 障害 CPU に対する通信

検出した異常はユニット構成情報に登録され、管理ネットワークを使い全 CPCI マスタに伝達する。MPI はこのユニット構成情報を参照し、各クラスタの通

信グループテーブルの該当する CPU 上のタスクの状態を更新する。MPI は異常と設定された CPU へは通信を行わず、エラーを返す。

(2) 障害 CPU の負荷移動

障害 CPU からの処理負荷の移動は、信号処理のアルゴリズムとして、アプリケーションレベルで receiver-initiated 方式のデータ処理の分配をすることにより自律的に行われる。アプリケーションは MPI がエラーを返した場合は、別の空き CPU を選択する。システムには、ボードレベル、およびユニットレベルで冗長 CPU を持つ。この余剰性能を使ってシステムを運用し、障害による CPU 停止に耐える。

(3) 異常情報の伝達

異常情報の伝達、ユニット構成情報の伝達は、通常のデータ交換と同様に MPI のメッセージ送受信を使って実現する。

(4) 動的構成変更への対応

MPI/SP のベースである MPI 1.2 の仕様では、通信タスクの動的参加をサポートしていない。MPI/SP では通信タスクの論理番号(rank)と CPU の物理的位置を静的に決定することにより、システム起動後の CPU 構成変更の問題を回避する。障害により停止、再起動中の CPU への通信はエラーとなるが、MPI としては一時的エラーとして管理し、その論理番号の割当は保持される。再起動が完了し CPU が復帰すると、通信コネクションを再接続し、前と同じ論理番号で通信可能にする。MPI はユニット構成情報の更新を参照して、CPU が復帰したことを検出する。

4. 今後の課題・おわりに

本異常処理機構は、MPI/SP に付加する設計としたので、MPI/SP 自体をオーバスペックとすることなく、ターゲットシステムの要求に応じて利用を選択できる。本設計に基づき実装を進める予定である。

現設計では、検出した異常は全て CPU 異常と仮定し、再起動による回復を試みているため、ネットワーク障害には対処できない。障害原因の切り分けと対応した対処の実現について検討する必要がある。

参考文献

- [1] Message Passing Interface Forum, "MPI: A message-passing interface standard", Jun. 12. 1995
 [2] 村山、落合、山口, "MPI/SP におけるクラスタ統合管理方式の設計", 情処学会第 63 回全国大会, 2001/9

表 2: 異常処理機構の動作

異常箇所	検出方法	異常伝達方法	対処
データ処理 CPU	ユニット内ハートビート	自 CPCI マスタ検出(a)	CPU ボード再起動→CPU ボード交換(活線挿抜)
	MPI エラー	CPCI マスタ間異常通知(c)	
データ処理ネットワーク	MPI エラー		-----
CPCI マスタ	CPCI マスタ間ハートビート	他 CPCI マスタ検出(b)	ユニット再起動→ユニット停止
CPCI ユニット	CPCI マスタ間ハートビート	他 CPCI マスタ検出(b)	ユニット再起動→ユニット停止