

マルチプロセッサ汎用計算機における 7L-4 中断処理の救済方式

吉岡 正孝郎 山田 尚文
(株)日立製作所

1. はじめに

オンラインシステム等の普及により、中枢となる汎用計算機システムの高信頼化への社会的要求が高まっている。また、より大きな処理能力を得るため、共有メモリ型マルチプロセッサ方式が採用されるようになってきた。このマルチプロセッサシステムにおけるプロセッサの冗長性を利用して、従来から、一台のプロセッサの故障時にも正常な別のプロセッサを利用することによって、故障を発生したプロセッサ上で実行を中断されてしまった処理を再実行し、処理の異常終了、システムの異常停止を回避する方式が開発されてきた¹⁾。本稿では、汎用計算機Mシリーズのアーキテクチャに基づいた中断処理の救済方式、P R A P (Process Resuming on Alternate Processor) に関して述べる。

2. P R A P方式の考え方

マルチプロセッサの冗長性により、たとえ一台のプロセッサが故障により停止しても、同じメモリ、入出力環境下にある他のプロセッサは正常に動作を続けることが可能である。そこで、障害発生プロセッサで実行していた処理の中断状態が正確に保証できる場合、その状態を引継げば正常なプロセッサで処理を続行できる。これがP R A P方式の考え方である。

従来、Mシリーズアーキテクチャでは、プロセッサ障害時、障害発生プロセッサ自身に、マシンチェックを報告していた²⁾。しかしこれでは、障害発生プロセッサ上で、障害処理と中断した処理の継続とを行うことになり、二重障害となってシステムが停止してしまう可能性があった。P R A P方式では、他の正常なプロセッサで処理の継続実行を行なうことにより、この危険を回避する。

3. 救済の対象

プロセッサの障害は、間欠障害と固定障害とに分けられる。間欠障害はリトライによって救済し、P R A P機能による救済は固定障害を対象とする。但し、マルチプロセッサシステムであっても、障害その他の原因によ

て一台のプロセッサのみが動作している(他のプロセッサは停止している等)場合には、処理を救済するプロセッサが存在しないため、P R A P処理は行わず、通常のマシンチェック処理を行なう。

4. ハードウェア動作

P R A P方式では、従来のマシンチェックや誤動作警報割込みインタフェースを利用して制御を行う。そのハードウェア動作は、マシンチェック機能を拡張したものであり、

- ① 命令リトライ処理
- ② 処理救済可能性の判断
- ③ 状態情報退避領域への状態情報の退避
- ④ チェック停止

から構成される。

プロセッサ障害が検出されると、命令リトライが開始される。命令リトライが何度か失敗すると、障害が固定障害であると判断する。そしてに障害発生直前の状態が保証可能である時、P R A Pの動作では中断処理が救済可能と判断し、プロセッサ状態の退避を開始する。状態が保証できない場合は、従来通りマシンチェック処理を行なう。

P R A Pに必要な情報の退避には、専用の情報退避領域を利用する。障害を発生したプロセッサから障害発生時のP S W、C P Uタイマ、クロックコンパレータ、A R、F P R、G P R、C R値が読み出され、格納される。これは従来マシンチェック報告時に行われていた処理と同じである。

障害発生プロセッサは、状態退避後、チェック停止する。その結果、誤動作警報外部割込み要因が他の動作中のプロセッサに対して発生する²⁾。誤動作警報割込みはシステムを構成するプロセッサの一つが異常停止(チェック停止)したことを全プロセッサに連絡するものである。誤動作警報割込みでは、異常停止したプロセッサのC P Uアドレスが割り込まれたプロセッサのP S A 1 3 2 - 1 3 3番地に格納される。

5. ソフトウェア処理

誤動作警報割込みを受け付けたプロセッサでは、格納された障害発生プロセッサの状態情報を読み込み、アフィニティチェック後、誤動作警報割込情報とともに編集し、中断された処理の継続実行を行なうための制御テーブルに情報を設定する。通常のユーザタスクの場合には、それはタスク制御テーブルであり、状態設定後レディキューに登録する。

ここで注意すべきことは、CPUアフィニティを持った処理、即ち走行する命令処理装置を限定した処理は別プロセッサでは実行すべきでないということである。このような処理は、一般に走行するプロセッサが変わった場合、異常な処理結果となると考えてよい。

6. PRAP機能有効/無効の制御

処理には、アフィニティを持ち、他のプロセッサには引継げない処理、マシンチェック処理等のように他プロセッサに引継いで実行しても意味のない処理がある。これらの場合には、処理の救済が可能であっても、PRAP処理を行わないようにする必要がある。また、IPL中など、制御プログラムにPRAP処理を行なう準備ができていない場合や、PRAP処理をサポートしていない

いOSを動作させる場合のため、プロセッサ毎にPRAP機能の有効/無効の切り替えを行えるようにした。

7. 効果

プロセッサの障害は、その大部分がリトライ可能である。中断されたすべての処理が継続実行可能と仮定すると、CPUハードウェア障害に起因するシステムダウン中、プロセッサの障害に起因するものの大部分がノードダウン化され、システムとしての可用性が向上する。

8. おわりに

コンピュータシステムの高信頼化に対する社会的要求に対応するため、マルチプロセッサシステムの持つプロセッサの冗長性を利用し、一台のプロセッサの障害時にも、正常なプロセッサで中断された処理を継続実行するPRAP方式を開発した。その結果は、プロセスセッション機能として、汎用計算機M-880、M-860で応用されている。

参考文献

- (1) 田代 他：プロセッサリリフを用いた耐故障コンピュータ
電子通信学会論文誌 Vol. J65-D No. 8 (1982. 8)
- (2) HITACHI：Mシリーズ処理装置(M/ASAモード)
8080-2-146 (1991. 8)

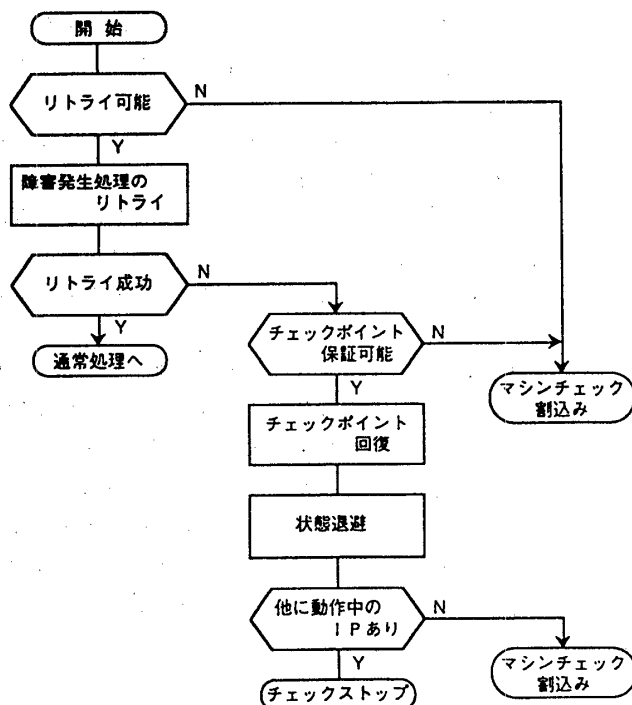


図1 PRAPハードウェア動作

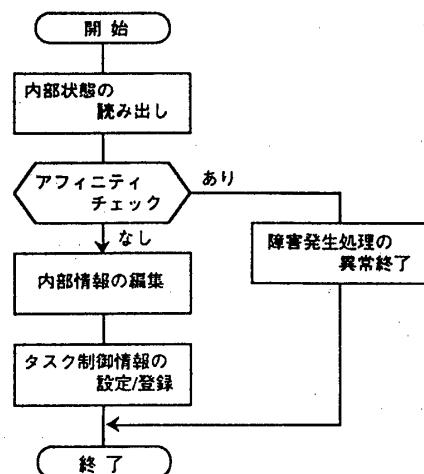


図2 PRAPソフトウェア処理