

情報爆発に対応する耐故障性 MPI フレームワークの提案

實本英之† 遠藤敏夫† 松岡 聡††

1. はじめに

大規模クラスターやグリッド等では、故障が発生しやすく長時間にわたる MPI アプリケーション実行のためには耐故障性が不可欠である。また、効率的な復旧を行うためには、環境、アプリケーション、ユーザポリシー等による異なる復旧戦略をとる必要がある。たとえば、一過性の一時故障とハードウェア故障による間欠性の一時故障では復旧戦略は異なる。後者の場合、プロセスを他のハードウェア上に移動させるべきである。しかし、すべての故障においてプロセスの移動を行うことは前者のような場合において、無意味にリソースを浪費することになる。しかし、LAM/MPI¹⁾、MPICH-V²⁾といった既存の耐故障 MPI では、故障検知/復旧戦略がハードコードされている場合が多く、柔軟な戦略をとることが難しい。FT-MPI³⁾のようにプロセスの生成/破棄等の基本的な API の提供によりプログラムコードを変更し、柔軟な耐故障性を実装できる MPI も存在するが、ユーザ負荷が高い。

我々の提案する ABARIS はアプリケーションコードの変更をおこなわずに、柔軟な復旧戦略を利用可能なフレームワークである。ABARIS は故障の検知/復旧をモジュールとして提供しており、ユーザはこれらのモジュールを選択するだけでアプリケーションを変更せず、効率のよい耐故障機能を利用可能である。

我々は以前シミュレーションを用いて ABARIS のアーキテクチャおよび最適な故障回復シナリオによる復旧オーバーヘッドの違いについて述べた⁴⁾。本論文では、ノード内ワークロードの上昇を故障と定義することにより動的負荷分散を実現し、ABARIS が柔軟なプロセス再配置を行えることを示す。

2. ABARIS フレームワーク

ABARIS は MPI プロセス状態の冗長化および一貫性保障 (PFTP: Parallel Fault Tolerant Protocol)、故障検知 (FD: Fault Detector)、故障復旧 (RS: Recovery Strategy) コンポーネントを持っている。それぞれのコンポーネントは以下の手順に従い復旧を行う (図 1)。

- (1) PFTP は定期的にチェックポイント等のプロセス状態の冗長化を行っている。さらに、定期的

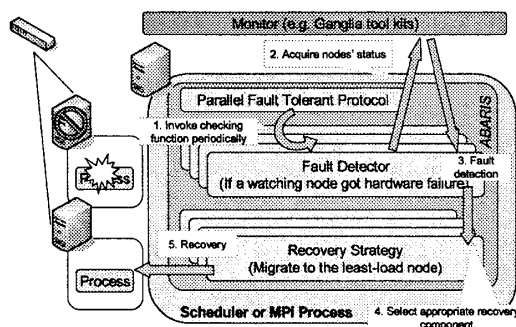


図 1 ABARIS における復旧シナリオ

に FD を呼び出す。

- (2) FD はモニタリングツールから情報を受け取り、自分の故障定義と比較、定義と一致した場合、FD は PFTP に選択した RS を通知する。
- (3) PFTP は FD から故障情報を受け取ると、選択された RS を用いて、プロセスの一貫性が保たれるように復旧を行う。

本研究では、ABARIS フレームワークを MPI の標準的な実装のひとつである MPICH-P4MPD に適用した。

3. コンポーネント実装

3.1 Parallel Fault Tolerant Protocol

PFTP は MPI プログラム全体の状態の冗長化と一貫性保障を行う。プロトタイプでは Coordinated checkpointing を実装した。この PFTP は定期的にチェックポイントを作成する。その際、MPI プロセス間の通信路に Drainage パケットを流すことにより in-flight なメッセージが無いことを保障する。PFTP は故障復旧時にすべての MPI プロセスの状態を同時に復旧する。これにより MPI プロセスの状態の一貫性が保障される。

3.2 Recovery Strategy

RS は各 MPI プロセスをどのように復旧するかを決定する。プロトタイプではもっともワークロードの低いノードにプロセスを移動する MIGRATE と故障プロセスがそれまで動作していたノードに再度プロセスを配置する RESTART を実装した。

3.3 Fault Detector

FD は故障検知を行い、故障発見時に適切な復旧手法を選択する。評価では各ノードのプロセスの集中を

† 東京工業大学

†† 国立情報学研究所

表 1 評価環境

CPU	AMD Opteron(tm) 280 Processor (2.4GHz)×2
Memory	4GB DDR SDRAM
Network	GbE
HDD	80GB SATA

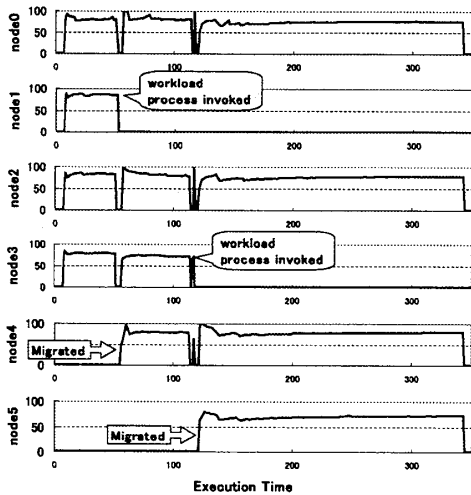


図 2 動的負荷分散を用いた CG ワークロード

検出する Performance FD を利用した。このモジュールは ps コマンドで 'R' フラグのついたプロセスをカウントし、これがノードの総 CPU コア数を越えた時故障と判定し MIGRATE RS を選択する。

4. 性能評価

4.1 評価環境

評価は 256 ノードクラスタのうち 4 台を用いて行い、各ノードには MPI プロセスを 1 つずつ配置した。各ノードの構成を表 1 に示す。また、プロトタイプは MPICH-1.2.7p1 を利用した。

4.2 動的負荷分散

ABARIS により柔軟なプロセス再配置が行えることを確認するために、Performance FD による動的負荷分散を行った。評価には NPB 2.4 CG CLASS-C を 4 プロセスで実行した。実行開始後、2 度 (45 秒後、125 秒後) ビジーループをワークロードとして投入し、動的負荷分散を行ったとき、行わなかったときの性能を比較した。また、チェックポイント間隔を 15 秒、故障検知間隔を 1 秒とした。

結果、動的負荷分散を行わなかった際の 458 秒に比べ、行った場合は 335 秒となり、実行時間は 73% に減少した。また CG のワークロードを図 2、図 3 に示した。図 2 ではまず node1 にワークロード増加が起こり、CG プロセスが node4 に移動しその後、node3 にワークロードが投入され、node5 に CG プロセスが移動していることがわかる。

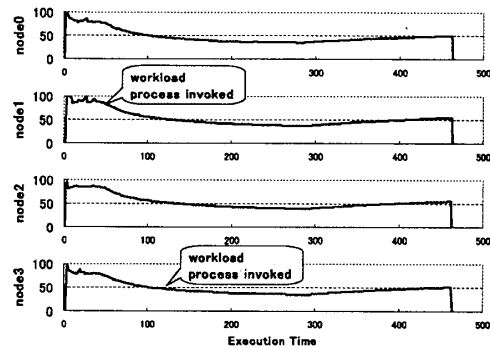


図 3 動的負荷分散を用いない CG ワークロード

5. おわりに

本研究では故障/復旧モデルを考慮した耐故障性を持つ MPI フレームワークである ABARIS を提案した。さらに、故障定義を工夫することによる柔軟なプロセス再配置の例として動的負荷分散が可能であることを示した。今後は、ABARIS コンポーネントのパラメータの自律構成や ABARIS フレームワークを利用した他システム (e.g. チェックポイントシステム) との連携について検討していく。

謝辞 本研究の一部は平成 18 年度科研費特定領域研究「情報爆発に対応する高度にスケーラブルな高性能自律構成実行基盤」による

参考文献

- 1) Squyres, J. M. and Lumsdaine, A.: A Component Architecture for LAM/MPI, *Proceedings, 10th European PVM/MPI Users' Group Meeting*, Lecture Notes in Computer Science, No. 2840, Venice, Italy, Springer-Verlag, pp. 379-387 (2003).
- 2) Bouteiller, A., Herault, T., Krawezik, G., Lemarinier, P. and Cappello, F.: MPICH-V: a Multiprotocol Fault Tolerant MPI, *International Journal of High Performance Computing and Applications*. (2005).
- 3) Fagg, G. and a Dongarra: FT-MPI: Fault Tolerant MPI, Supporting Dynamic Applications in a Dynamic World, *Euro PVM/MPI User's Group Meeting 2000*, Springer-Verlag, Berlin, Germany, pp. 346-353 (2000).
- 4) Jitsumoto, H., Endo, T. and Matsuoka, S.: "ABARIS: An Adaptable Fault Detection/Recovery Component Framework for MPIs", *12th IEEE Workshop on Dependable Parallel, Distributed and Network-Centric Systems (DPDNS'07)*, in conjunction with IPDPS2007, p.311 (2007).