B-011

HPC システムにおける障害監視および動的再構成機能
Failure Detection and Automatic Re-configuration of HPC System

齊藤隆之 † 善甫康成 ‡
Takayuki Saito Yasunari Zempo

1. はじめに

筆者らは，HPC 計算機クラスターの計算資源管理ミドルウェア ShareTask を開発してきた．[1, 2, 3, 4, 5]

このソフトウェアでは，計算ノードに常駐するエージェントが主導権をもち，ジョブ，入力ファイル，制御命令を管理サーバーからダウンロードし，計算ノード側で発生したイベントと出力ファイルを管理サーバーにアップロードすることにより，ジョブ制御から計算資源監視までをプル型で実現している．(図 1) 一般的な計算機クラスターから，物理的に分散した計算機群，あるいはクラウド環境までを統一的に扱うために，通信プロトコルとしては HTTP(HTTPS) のみを使用している．計算ノード数の動的な増減への対応が容易なため，クラウド環境における仮想マシンの動的生成・消滅や，計算ノードの異常停止への対応において，この特徴が有利であることが実証されている．

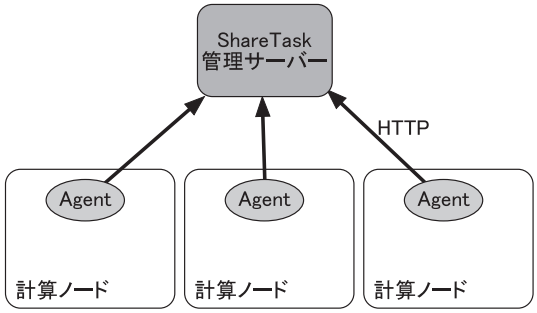

図 1: ShareTask の構成

計算機クラスターは，そのノード数の増大とともにハードウェア／ソフトウェア両面で発生する障害に対処するための運用コストが上昇する．エージェント・プル型の特徴を活かして ShareTask では，障害により機能停止した計算ノードは自然に計算ジョブ割当対象から除外されジョブスケジューリングへの影響がない．しかし，機能停止まで至らなくても，CPU，メモリ，ローカルストレージなどのリソースの逼迫や機能不全により計算ジョブを正常に実行できず異常終了させてしまうことがたびたび経験されている．そのような機能不全の計算ノードは，多数のジョブを異常終了させてしまうことが多いため，できるだけ早期に事態を検出しジョブ割当対象から除外することが求められる．

このようなリソース逼迫や異常を管理者が把握するために，リソース監視機能を内蔵しているが，その回復操作は管理者が行う必要があるために解決まで時間がかかる．経験上，リソース逼迫の原因は，過去に実行したジョブの影響（リソース解放漏れ）であることが多く，その回復操作には共通性が多い．一方，ハードウェア，ネットワーク，ファイル I/O の一部障害は，ジョブ実行の異常の原因を調査して判明することが多い．

これらの経験から，障害監視（検出）とその回復操作をできるだけ自動的に行い，障害の内容によっては，計算ノードをジョブ割当対象から自動的に除外することによってジョブの異常終了を防ぐしくみの開発に取り組んでいる．

2. 従来技術

2.1. 従来の監視ソフトウェア

サーバーとそのサービスを監視し，その健全性とリソース使用状況を監視するソフトウェアはさまざま開発され広く運用に活用されている．それらは，Web，データベース，メールなど，IT インフラを構成するサービスとそのサーバーのリソース監視 (記録) を主目的とし，以下のような監視を行う．

- CPU，メモリ，ストレージなどのリソースが平常範囲内にあるか
- サービス (サーバープロセス) が健全な状態か

管理者は，監視データの目視あるいはメール通知によってリソース逼迫やサービス停止を把握し，回復操作を行う．監視ソフトウェアとジョブスケジューリングの連携は想定されていない．

2.2.IPMI

サーバーのハードウェア監視のために，IPMI (Intelligent Platform Management Interface) と呼ばれる仕様と実装がある．これによりサーバー筐体内の消費電力，温度，冷却ファン回転数などの情報を取得することができる．温度異常をトリガーとするジョブ実行抑制やシャットダウン制御などのために有用である．

3.HPC システムにおける監視項目

HPC システムの計算ノードでは，定常的にサービスを実行するのではなく，バースト的に発生する計算ジョブプロセスを実行する．したがって，IT インフラを構成するサーバーと異なり，リソース監視は 3 つの段階にわけて考えるべきである．

1. ジョブ実行直前のリソース余裕の確保
2. ジョブ実行中のリソースの逼迫
3. ジョブ実行終了後のリソースの解放

リソース余裕がない状態ではジョブを実行することは避けるべきである．つまり，ジョブ実行終了後には，アプリケーションプロセスが使用していたリソースはすべて解放されるべきであるが，プロセスやスクラッチファイルの残存などが起こりうる．このようなリソー

†(株) アンクル, ANCL, Inc.
‡法政大学 情報科学部

Copyright © 2015 by Information Processing Society of Japan and
The Institute of Electronics, Information and Communication Engineers
All rights reserved.

スの解放漏れは，従来の監視ツールで発見することは可能であるが，その影響を受けるジョブをなくすためには，自動的にその削除を行うことが望まれる．このような観点から，主要な監視項目として以下を挙げることができる．

- 残存プロセス
- CPU，メモリの逼迫
- ローカルストレージの容量逼迫，I/O 性能低下
- 共有ストレージのマウント喪失，I/O 性能低下
- システムログ (syslog) の重大な警告
- 過大なネットワークトラフィック，エラーパケット

4.ShareTask のアプローチ

上記の監視項目について異常検出時の自動対応とジョブ実行抑制について，エージェント・プル型の特徴を活かしてつぎのようなアプローチをとった．(図 2)

4.1. エージェントによる自律的監視

エージェントが，各監視項目について情報収集，計測，判定を行うスクリプトを定期的に実行する．判定結果によって，つぎのジョブを取得するかどうかを制御する．スクリプトにより収集したデータと判定結果は，管理サーバーに報告されるので，管理サーバー側で各エージェント内の監視状況を確認できる．

4.2. テストジョブ実行による健全性確認

計算ノードの健全性は，計算アプリケーションを実行し，その結果を検証することによってより確実に確認できる．そこで，ジョブ実行の間隙にテストジョブを自動的に実行することとした．

テストジョブは，特別なジョブキューでエージェントに配信される．このジョブキューには，実行ノードが指定されたジョブが管理サーバーによって自動的に投入される．エージェントがジョブを取得すると，同じジョブが自動的に再投入される．これにより，エージェントがテストジョブを反復実行でき，実行結果を管理サーバー側で確認できる．また，テストジョブを管理サーバー側で随時変更できるという利点もある．

テストジョブを実行するタイミングはエージェント側で決定されるために，通常のユーザージョブへの影響を抑えることができる．

テストジョブには，計算結果の妥当性をチェックするスクリプトも含める．テストジョブが異常終了あるいは計算結果が妥当でない場合には，通常のジョブの取得を停止し，その旨を管理サーバーに通知する．

4.3. ハードウェア異常に関する情報収集

消費電力，温度，ファン回転数などハードウェアに関する情報は，IPMI を通じてエージェントが取得・蓄積し，異常値判定を行うとともに管理サーバーに送信する．管理サーバーから各ノードに対して IPMI によるデータ収集を逐一行うのではなく，ノード側で一定期間蓄積されたあと集約された形で管理サーバーに送信するため，ノード数の増大に伴う管理サーバーの負荷上昇を抑えることができる．

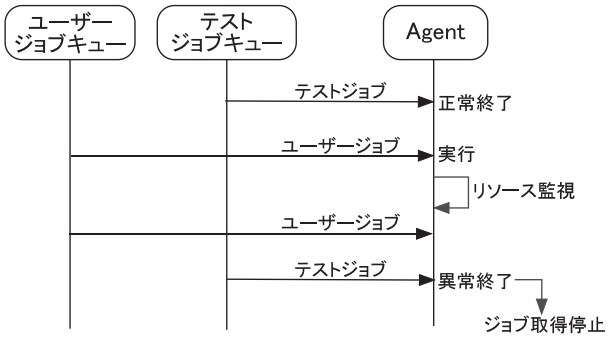

図 2: エージェントによる自律的監視

5. まとめと今後の展開

HPC システムにおいては，リソースと障害を監視し，ジョブの実行時エラーを未然に防ぐ努力が必要である．管理者が障害の回復操作のすべてを行うのではなく，典型的で単純な障害については計算リソース管理システムが自動的に問題を解消し，複雑な障害についてはジョブ実行を抑制することによって信頼性の高い運用を実現できる．

エージェント・プル型の特徴を活かすことによって，エージェントが監視から回復操作とジョブ実行の抑制までを自律的に行う仕組みが実現できた．CPU，メモリー，ストレージなどの個々のリソース情報による障害判定だけでなく，テストジョブによる総合的な判定を組み込むことによって確度の高い障害検出が可能になった．

一方で，OS レベル情報 (CPU，メモリ，ストレージ，プロセス) から，IPMI 情報 (マザーボードなど筐体内環境) まで多様なデータが収集できるが，障害とそれらデータとの相関についての知見の蓄積が重要と考えている．今後，その知見にもとづいて，障害判定ロジックとテストジョブの改良を行う．

参考文献

[1] 齊藤隆之, 平松和剛, 善甫康成, ”HPC 計算資源管理におけるソフトウェア動的配備技法について”, FIT2014 B-015.

[2] 齊藤隆之, 善甫康成, ”プル型ジョブスケジューラにおける動的通信量制御方式”, FIT2013 B-016.

[3] 齊藤隆之, 善甫康成, ”Web ベース計算リソース管理ミドルウェア：ShareTask”, FIT2012 B-012.

[4] 善甫康成, 齊藤隆之, 岡戸晴彦, 近野利信, 千田範夫, ”自律プル型制御方式によるインターネットワイドなジョブスケジューリングの性能試験”, 九州大学情報基盤研究センター 先端的計算科学研究プロジェクト成果報告会 (2010).

[5] 善甫康成, 齊藤隆之, 岡戸晴彦, 近野利信, 古賀良太, 千田範夫, ”メタスケジューリングを指向した計算環境”, 日本コンピュータ化学会 2010 年春季年会研究展示 RX02 (2010).

Copyright © 2015 by Information Processing Society of Japan and
The Institute of Electronics, Information and Communication Engineers
All rights reserved.