

マルチグリッド環境における統合監視・診断システムの フレームワークに関する研究

長坂 康史*・小山 敦**

(平成21年10月31日受理)

A Study on Framework for Integrated Monitoring and Diagnostic System in Multi-Grid Environment

Yasushi NAGASAKA and Tsutomu KOYAMA

(Received Oct. 31, 2009)

Abstract

Grid computing is studied and developed all over the world. There are many projects of Grid computing system even in Japan. Each system is well developed and used at present without any troubles. The system, however, becomes big and complicated to get a high CPU power. A monitoring and diagnostic system is important under such a situation. The well-developed monitoring and diagnostic system is available for most of Grid systems. On the other hand, the Grid system comes to consist of multi-Grid middleware because of its large size. The automatic monitoring and diagnostic system for multi-Grid environment is, therefore, required.

We propose a framework for integrated monitoring and diagnostic system which is used in a multi-Grid environment. The framework make it possible to develop the monitoring and diagnostic system and to use easily not only in single Grid environment but also in multi-Grid one.

Key Words: multi-grid, monitoring and diagnostic system, framework

1. はじめに

近年、グリッド・コンピューティング技術を用いて大規模計算環境を構築し、そのリソースを用いることによって、様々な研究が活発に行われている。グリッド・コンピューティングを用いることによって、大量のリソースを仮想化・集約し、スーパーコンピュータに匹敵する様な計算能力を実現することが出来る。この大量のリソースを利用する研究分野としては、現在、ヒトゲノムにおける生物情報解析や、天文学における天体情報解析、高速加速器実験におけ

る物理解析等が挙げられる。これらの研究では大量のデータを扱い、それを用いた大規模な計算処理を行わなければならないため、大規模な計算リソースを低コストで用意できるグリッド・コンピューティングの利用が有効であるとされ、実用化のための研究が行われている。

また、企業においては、災害等により組織内の機器が故障した場合、速やかに別の機器へ接続を切り替えることにより、業務の中断を回避することを目的として、グリッド・コンピューティングによる自律的なリソース管理機能を用いた耐久性の高い業務システムの実用化が進められてい

* 広島工業大学情報学部情報工学科

** 筑波大学大学院システム情報工学研究科

る。

このように、グリッド・コンピューティングは各方面で利用されており、将来的に社会に大きな影響を与える技術であると言える。しかし、グリッド・コンピューティングは多数の計算機を利用するため、個別に状態を監視すると全体として大きな維持コストが発生する。そのため、従来では各ミドルウェアに特化した監視システムによってグリッド環境を監視・維持する研究が行われてきたが、異種ミドルウェアが混在する現状では、複数のミドルウェアを跨いだグリッド環境の監視を見据えた新たな監視モデルが必要となっている。

本研究では、NAREGI ミドルウェアの監視を主な機能としたマルチミドルウェアの監視をシームレスに行うことが出来る自動監視・診断システムのフレームワークを提案する。

2. グリッド・コンピューティング

2.1. グリッド・コンピューティングとは

グリッド・コンピューティングは、組織内に分散して存在する複数の計算機を仮想化・統合し、あたかも1台のコンピュータであるかのように利用するための技術である。グリッド・コンピューティングはその仮想化・統合における計算機の管理、セキュリティやユーザ管理のための技術を基本的な特徴としているが、実際に提供するサービスの内容によって数種類に分類されている。本研究で扱うシステムはコンピューティンググリッドと呼ばれるもので、仮想化し集約されたコンピュータの計算資源を用いて大規模な計算処理を行うことを目的としている。

2.2. グリッドミドルウェアの実装

グリッド環境を構築するためのミドルウェアは、それぞれ独自のサービスを持ち、内部の仕様が異なるため、監視のために専用のモジュールを用意する必要がある。本研究では、主にNAREGI ミドルウェアの監視を目的として用い、さらに、マルチグリッド環境の監視性能を検証するために、gLiteも扱う。

2.2.1. グリッド環境の概要

現在、日本では国立情報学研究所が中心となり、国産グリッドミドルウェアを開発するための超高速コンピュータ網形成プロジェクト (National Research Grid Initiative) [1]が推進されている。このプロジェクトによって開発されたミドルウェアをNAREGI ミドルウェアと呼ぶ。本研究は前年度の広島工業大学における研究[2]に引き続き、日本国内で展開するグリッドミドルウェアの監視システムに関する共同研究として行われた。そのため、NAREGI

ミドルウェアの監視と共に、gLiteに対しても同じシステム上で監視することによって監視効率の向上、維持コストの低減が期待できる。

2.2.2. NAREGI ミドルウェア

NAREGI ミドルウェアはグリッドミドルウェアとしての基本的な機能を持つ他、科学計算のための資源管理機能と、グリッドプログラミング機能を持つ。この2つの機能はコンピューティンググリッドを構成する中心となる部分である。なお、本研究ではコンピューティンググリッドのノード監視を目的としており、上記2つの機能の監視を行う。

2.2.3. NAREGI ミドルウェアの機能

NAREGI ミドルウェアを構成する各コンポーネントの関係を図1に示す。また、ユーザ管理やアカウント管理、資源管理などコンピューティンググリッドを構成するための機能を提供する一纏まりのノード群を仮想組織 (VO: Virtual Organization) と呼ぶ。

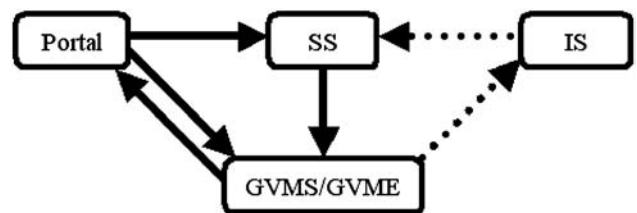


図1 NAREGI ミドルウェアの構成

図中の矢印はデータの流れを表し、実線は処理データを、点線は情報データを表す。各コンポーネントは1台のマシン上に配置されるが、GVMEは1VO内に複数存在することが出来る。また、GVMSはGVMEの情報进行管理する。ユーザはPortalにのみ接続し、全ての操作はPortal上で行われる。GVMSとGVMEは計算ノードとして計算資源を提供する。Portalから利用したいアプリケーションをGVM上に登録する。利用したいアプリケーション上で実行するパラメータサーベイなどのジョブをPortalからSSに送信する。SSはジョブを受け取ると、所属VO情報やジョブの種類に従ってジョブを実行するのに最適なGVMノードを選定する。この際、ISによって集められたGVMノードの資源情報を参照する。選定されたGVMノードにSSからジョブが投入され実行される。実行結果はPortalに送られ、ウェブブラウザ上で確認できる。

3. 統合監視・診断システム

マルチグリッド環境における統合監視・診断システムでは、VO内の各ノードに障害が発生しないか監視するだけ

でなく、ノード毎の機能やノード間におけるバージョン一致、IP 状態監視など、グリッド機能が維持されているかミドルウェア毎に監視し、障害が発生した場合に直ちに復旧に向けて活動できるよう支援することが必要である。また、マルチグリッドをスムーズに監視することが出来るような配慮が必要となる。

3.1. システムに必要な機能

統合監視・診断システムに必要な機能を纏めると、以下の5つの項目に分けられる。

3.1.1. 監視機能

監視機能の役割は2つに分類される。1つは各ノードの通常の機能が正常に動作しているか確認する定常監視である。もう1つはグリッドの機能を提供するノード群内部におけるサービスの死活確認である。マルチグリッド環境では各ソフトウェアによって使用する監視用モジュールが異なるため、これらをシームレスに監視するためには統一的なインターフェースを用いたスムーズな設定が必要である。

3.1.2. 診断機能

監視した結果は直接管理者に渡さず、システム内で自動的に異常があるかないかを診断することで労力を削減することが出来る。診断結果をデータベースに保存しておくことで、統計的な分析を加えることが出来る。

3.1.3. 通知機能

診断結果に異常が見られた場合、直ちに管理者へメールやウェブを用いて通知されなければならない。また、診断が行われた次の段階で通知が行われることから、既知のエラーであればあらかじめ対処法を登録しておき、通知と同時に自動的に起動することもサービスの早期復旧にとって有効である。

3.1.4. 可視化機能

監視システムにおける可視化機能の役割は2つに分けられる。1つ目は GUI による設定の容易化やデータのグラフ化など操作性を向上させることが出来る。2つ目は視覚による異常の早期発見、事後確認等に役立つため保守性を向上させることが出来る。経験の少ない管理者でも容易に理解できるメリットがある。

3.1.5. 記録機能

監視および診断によって得られたデータはデータベース等に保存することで、図式化などによって再利用すること

が出来ると。また、過去のデータを統計的に分析することで VO 内の障害における傾向を掴む事はグリッド環境の保守性を高める上で重要である。更に、これらのデータを VO 間で共有することでよりスムーズに障害に対応することが出来る。

4. 提案フレームワーク

本研究では、先に述べた統合監視・診断システムを開発するために必要な機能を満たすためのフレームワークを提案する。提案フレームワークのモデル図を図2に示す。図では監視対象であるマルチグリッド環境上に配置された統合監視・診断システムの各機能を表している。

(1) 管理

ウェブブラウザを用いて監視ノードや監視項目を追加する。ノード毎の監視条件の設定変更を行える他、過去データを表示することが出来る。

(2) 監視

VO 内のノードに対し、定常監視を行う。グリッド環境における監視を行うためには、一般的なりソース状態監視、IP 死活監視に加えて、グリッドミドルウェアに合わせたコマンドやモジュールを用いたサービスの状態監視が必要となる。NAREGI ミドルウェアのようにコマンドラインを持たず、ウェブ上での操作しか出来ないミドルウェアもあるため、ノード内で直接処理を実行する監視モジュールを起動する必要がある。提案フレームワークではサーバ・エージェント通信を用いてノード単体で処理を実行し、サービス状態を監視することが出来る。

(3) 診断

監視結果を設定した閾値や条件によって診断する。異常が検出された場合、自動的に管理者に通知するためのイベントを起動する。

(4) 通知

管理者に異常発生のエラーログなどの詳細をメール等を用いて通知する。また、条件に合わせてコマンドやプログラムを起動しプロセスを実行することも出来る。

(5) 記録

監視およびその診断結果をデータベースに記録する。

(6) 可視化

記録した過去のデータをグラフ表示などにより可視化することが出来る。

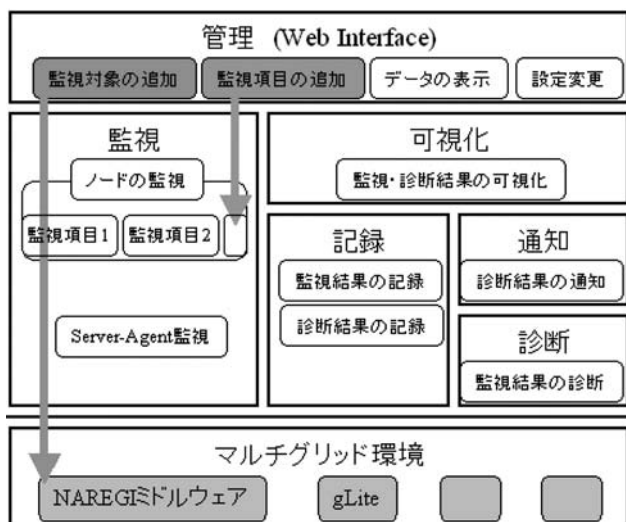


図2 提案フレームワーク

5. 開発システム

前章で述べたフレームワークを元に、実際にシステムの開発を行った。

5.1. ネットワーク監視ツール

本研究ではシステムの開発に際し、その基盤部分として、オープンソースのネットワーク監視ツールである ZABBIX を採用した。ネットワーク監視ツールとは、サーバ、ネットワーク、アプリケーションなどをオンラインで監視することが出来るソフトウェアである。オープンソースを用いることで、システム開発のスピードアップを図ると共に、無償であるためコストを低減することが出来る。ZABBIX では ping などの IP 死活状態や CPU 使用率などのリソース状態を監視できる。

5.2. 開発システムの機能

本システムでは ZABBIX の基本的な機能を利用し、フレームワークの実現に不足する機能を補う形で構築した。管理機能および通知機能は ZABBIX の機能を利用し、それ以外の機能はスクリプトを開発し、既に利用可能な部分でもプログラムレベルで改良することにより、フレームワークに適合させるようにした。

5.2.1. 監視機能

本システムにおける監視機能はネットワーク監視ツールにプラグインとして登録し、監視ツールからの設定にしたがって起動させる。各機能は役割に従って以下の項目に分けることが出来る。

(1) 応答監視

ping を用いた IP の死活監視や、グリッドサービスに必

要なポートの開閉を監視できる。

(2) ネットワーク監視

ノード間の RTT (Round Trip Time) や、GridFTP を用いたネットワーク上のファイル転送速度を計測する。マルチグリッド環境下ではミドルウェアによって操作性が異なるため、gLite ではジョブ投入時のノード選定の参考にすることが出来るが、NAREGI では SS ノードによって自動的に選定が行われるため、計算ノードの状態監視に利用することが出来る。

(3) グリッドサービス監視

GridFTP を用いて、ジョブを実行する上で必要となるファイル転送機能を確認することが出来る。また、NAREGI ではグリッドプログラミング機能を監視するため、GVM ノードにおいて簡単なジョブを実行し、状態を監視する。

(4) リソース監視

各ノードにおける CPU 使用率やロードアベレージなどのリソースの状態を監視できる。これらの情報は gLite においては効率的にジョブを処理するために利用することが出来る。

5.2.2. 診断機能

監視結果に対して設定した条件に従い、診断を行う。診断には監視ツールの閾値を利用するが、サービスの停止といった致命的な異常の場合にはエラーログをデータベースに保存することが出来る。これは開発したログビューアを用いてブラウザ上で確認出来る。

5.2.3. 可視化機能

本システムでは既存の Mapping 機能を改良することで、各ノードのグリッド機能を確認することが出来る。診断結果が正常であっても、他ノードで異常が発生した場合にいち早く影響を把握することが出来る。

5.3. システムの運用例

本研究において開発システムを実装した例を図3および図4に示す。図3は監視プラグインを起動したシステムを

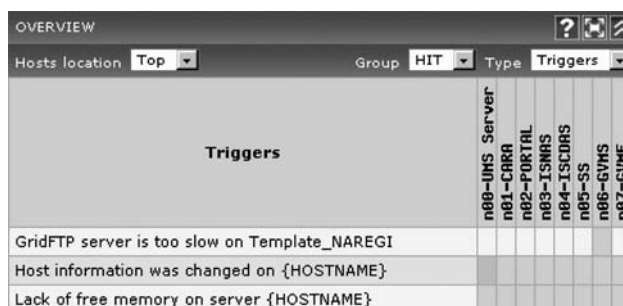


図3 ZABBIX の Web 上における監視画面例

実際の NAREGI 環境上にて実装した場合の ZABBIX のウェブブラウザにおける管理画面である。図 4 は ZABBIX 上にて実装したログビューアによる過去ログの検索画面である。

itemid	clock	num	value_min	value_avg	value_max
22169	1228834800	120	0	0	1
22170	1228834800	120	1	1	1
22168	1228834800	120	1	1	1
22162	1228834800	120	1	1	1
22163	1228834800	120	1	1	1
22164	1228834800	120	1	1	1
22167	1228834800	120	1	1	1
22166	1228834800	120	1	1	1
22165	1228834800	120	1	1	1

図 4 ログビューアによる過去ログ検索画面

6. 性能評価

本システムの性能を評価するため、GridFTP による RTT 計測プラグインによる RTT 測定を行った。比較として別の統合監視システムである Nagios 上で専用のプラグインを利用し RTT を計測した。実験環境を表 1 に示す。RTT 計測には、NAREGI は GVM 上から他 GVM ヘデータを送受信し、gLite では UI (User Interface) から SE (Strage Element) ヘデータを送受信して計測した。データは 20MB, 40MB, 60MB, 80MB, 100MB のバイナリデータを用いた。計測結果をグラフ化したものを図 5 に示す。

表 1 実験環境

	ZABBIX	Nagios
MW	NAREGI	gLite
VO	HIT	KEK
OS	Scientific Linux SL 3.0.9	Scientific Linux SL 3.0.9

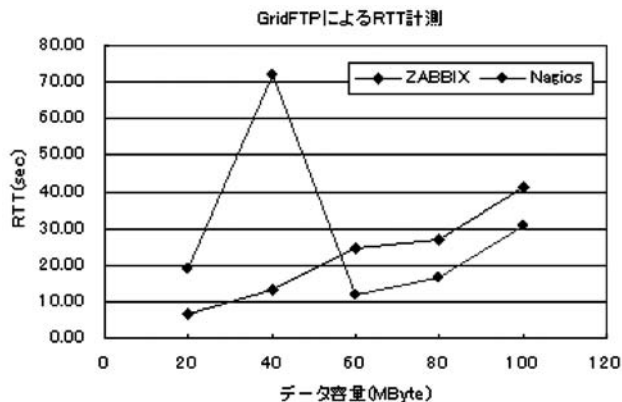


図 5 GridFTP による RTT 計測値の比較

図 5 では Nagios による監視において RTT 値が大きく乱れている。また ZABBIX による観測では終始安定した数値が得られている。しかし、60MB 以降の計測では、Nagios を使った計測値が小さくなっている。これは ZABBIX では比較的执行ジョブ数が少なくシステム負荷率の低い VO を用いたのに対し、Nagios では実行ジョブ数の多いと思われる VO を用いたため、実行ジョブが集中する時間帯に重なり数値が大きくなった物と思われる。また、ZABBIX はエージェントを用いた内部測定であるのに対し、Nagios はサーバからコマンドによる遠隔操作により測定したため、ZABBIX の方がオーバーヘッドが大きいと思われる。

7. 考察

実験によって、マルチグリッド環境に対応した監視システムと従来の監視システムの比較、また、グリッドミドルウェア自体の相違による比較を示した。グリッド環境が大勢のユーザによって共有され、リソース状態が不安定であることを除けば、若干エージェントレスである Nagios 型の監視システムが安定しているが、マルチミドルウェアに対応していることを考えれば、ZABBIX を用いた本システムが監視システムとして十分な性能を持っていると考えられる。

8. まとめ

本研究ではマルチグリッド環境のための統合監視・診断システムのフレームワークを提案した。提案フレームワークでは監視プラグインを開発した他、データを参照することが出来るログビューアの開発、視覚化のためのマップ機能の追加を行った。これによりマルチグリッド環境においてスムーズな監視が行えるようになった。今後は更にプラグインおよび機能を充実し、ユーザビリティを向上させて行く。

参考文献

- [1] 三浦 謙一「サイエンスグリッドプロジェクト “NAREGI” の概要」, Progress in informatics Vol. 3 (20060400) pp. 67-75
- [2] 藤井, 長坂, 渡瀬, 佐々木, 岩井「加速器科学仮想組織におけるグリッド環境自動監視診断システム」, FIT2007 第 6 回情報科学技術フォーラム, p5-6, L-003, 2007/9/5

