広島工業大学紀要研究編
第43巻（2009）pp.311-316
論　　文

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

長坂　康史*・藤井　峰夫**

（平成20年10月31日受理）

Automatic Monitoring and Diagnostic System of the Grid Environment in Accelerator Science Virtual Organization

Yasushi NAGASAKA and Mineo FUJII

（Received Oct. 31, 2008）

Abstract

Many studies for the grid computing are preformed in a lot of fields in the world. The PPJ, Particle Physics in Japan, virtual organization is one of the grid computing testbeds in Japan. It consists of 6 institutes in Japan and is based on one of the grid middlewares, that is, gLite. The system is managed and operated very well. But the monitoring and diagnostic systems are not automatic. We have, therefore, developed a monitoring and diagnostic system over the gLite middleware. The system is designed not only to check site status but also to aid site administrators by automatically navigating to appropriate documents in their site operation. The system is able to operate grid sites with lower costs.

Key words: grid computing, monitoring and diagnostic system

1. はじめに

近年，グリッド・コンピューティングは様々な分野で実用化され，利用者のニーズに応えるサービスを提供している。一方，グリッド技術の発達により，グリッドを利用したシステムの規模は大きくなり構造は複雑になった。また，分散した環境を仮想化して統合するという特徴から多くのグリッドの拠点は散在し，結果的に管理者はグリッド環境全体の管理が困難となる。

現在，グリッド・ミドルウェアの1つであるgLiteを用いてグリッド環境の構築されたサイトが国内の6研究機関に存在し，それらは加速器科学仮想組織という仮想組織として形成されている。

上記の仮想組織のグリッド環境と機能を維持するためには，適切な管理および運用を行う必要がある。そのため，多くの管理者が必要となるが，組織の多くは大学の研究室であり，そこでの専門家の確保は難しく，管理者は不足している。そこで，本研究ではグリッド・システムの運用を支援するために自動監視・診断システムを開発し，少ない管理者で，グリッド・システムのサービス全体を正常に保つことを目的とする。

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，ネットワークを介して複数のコンピュータリソースを仮想的に統合する仕組み

* 広島工業大学情報学部情報工学科
** 日立情報通信エンジニアリング株式会社

である。元来，グリッドとは格子という意味の他に送電網という意味があり，電力のように必要な時に，必要な情報サービスを，即座にまた，簡便に受けることが出来るという，情報資源をユーティリティ化した情報インフラとなる技術を目指している[2]。

2.2 グリッド環境

ヨーロッパ合同素粒子原子核研究機構（CERN）では高エネルギー加速器LHC（Large Hadron Collider）を利用した実験が行われる。この実験のデータを処理するためのグリッド環境を構築するプロジェクトをLCG（LHC Computing Grid Project）という。このプロジェクトで採用されているミドルウェアを用いて，国内でのグリッド環境を構築した。また，そのグリッド環境下で大量のデータを管理・処理することの出来る仮想組織（VO：Virtual Organization）の基盤を確立し，その基盤の元，国内6研究機関で構成される加速器科学仮想組織を形成した。

ここで定義する仮想組織とは，ネットワークで繋がれた資源（CPUなどのコンピュータ資源）を共有，もしくは，それを利用して共同作業をするグループのことである。仮想組織内の各拠点では，すべての拠点にある計算資源を自由に使い解析を行うことが出来る。

2.3 グリッド・ミドルウェア

本仮想組織ではグリッド・コンピューティング環境を構築するためのミドルウェア[3]としてEGEE（Enabling Grids for E-Science in Europe）が開発したgLiteと呼ばれるグリッド・ミドルウェアを用いている。

2.3.1 gLite

gLiteでは複数の機能が相互に連携してグリッド環境を構築する。グリッド環境へのログイン・認証には電子証明書を利用する。また，gLiteは仮想組織のユーザ管理機能や，ジョブのコントロール，アカウンティング，リソース情報の管理・共有などの機能を持つ。

2.3.2 gLiteの構成

gLiteグリッド環境で構築されるそれぞれの機能の関係を図1に示す。実線は処理データの流れ，点線は情報データの流れを表している。仮想組織内の各サイトではgLiteで構築されたグリッド環境が動作している。ユーザはUI（User Interface）からジョブの実行依頼などの操作を行う。UIは仮想組織内のRB（Resource Broker）にジョブを依頼する。RBではBDII（Barkley Distributed Information Index）を参照しジョブを実行するのに適したCE（Computing Element）を選び，そこにジョブを渡す。BDIIでは，グリッド環境の情報を収集し，これを公開している。CEはいくつかのWN（Worker Node）を管理しており，RBから渡されたジョブを実行可能なWNに渡す。WNは一般的にクラスタ構成になっており，ここでジョブが処理される。また，ストレージ資源を使う場合，WNはBDIIを参照しLFC（LCG File Catalog）へアクセスを行う。LFCはジョブが使用するグリッド上のデータファイルの統合的カタログ管理を行う。これにより，同じデータを複製し分散配置することができる。WNはLFCからの情報を元に，必要なデータを保存している自分に近いSE（Storage Element）を利用可能となる。

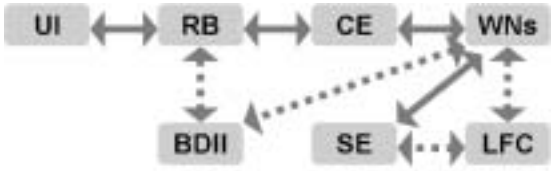

図1. gLiteの機能

3. 自動監視・診断システム

グリッド環境・機能を維持するためには，ソフトウェアの更新を定期的に行い，適切な設定がなされた上で各拠点が正しくサービスを提供している必要がある。そこで，サービスの可用性を高めるため，グリッド・システムの運用を支援する自動監視・診断システムを開発し，加速器科学仮想組織上にこれを展開する。

3.1 システムに求められる機能

グリッド環境における自動監視・診断システムに求められる機能は以下の5つにまとめることが出来る。

3.1.1 監視

監視は，各マシンが正常に動作しているかどうかの監視とグリッド・システムとして正しく機能しているかどうかの監視との二つに分類することができる。

前者は，ネットワークを介して接続されているマシンで稼動している基本的なサービスが正常か否かを監視する機能である。また後者は，gLiteで構成されたグリッド・システム環境そのものを監視するものである。

3.1.2 診断

監視しているマシンに異常を発見した場合，その異常に対する診断情報をデータベースから取得し，管理者に提供するとよいと考えられる。診断ではその機能を提供する。また，データベースに含まれない未知の異常の場合，対処

した管理者がその対策をデータベースに逐次蓄えることが出来る機能も提供する。

3.1.3 通知

通知は，監視しているマシンの異常をメールなどで管理者に伝える機能である。異常の種類やレベルにより，通知相手を自動的に選択できることが望ましい。

3.1.4 記録（アカウンティング）

障害に対しては，同じ障害を起こさないためにも，その障害が起こった過程を記録・分析し，原因の追及，そして，対応策を練ることが重要である。記録は分析・原因究明などに役立てる為の情報記録機能を提供する。

3.1.5 表示

システムの監視は通常，ネットワークに接続されたコンピュータで行う為，ネットワーク経由で Web ブラウザなどから表示出来ると良いと考えられる。表示は，ユーザが複雑な操作を必要とせず，簡潔に情報を得られるよう Web ブラウザを利用した情報提供を実現する。

4. 開発システム

4.1 システムのフレームワーク

上記の要件を満たすように提案したシステムのフレームワークを図 2 に示す。提案したフレームワークは，コストを抑えるためシステムの開発基盤として無償の総合監視ツールを利用する。総合監視ツールは，システムの様々な監視を自動で行い，監視結果を診断して正常か否かを管理者に伝えることが出来るが，一般的にグリッド環境の監視機能がない為，これをプラグインという形で開発した。また，診断機能は監視機能と連携する為，グリッド環境の監視に合わせた診断機能も開発した。データベースは監視・診断結果を記録して活用するために設置し，そのデータベースから必要なデータを検索する検索機能や，データを地図やグラフに可視化して表示する可視化機能，そして，Web 上に監視・診断情報を用いた情報共有の場を作成した。

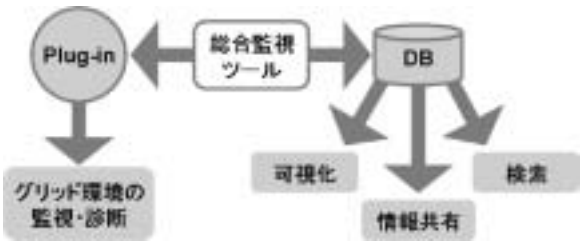

図 2. 提案フレームワーク

4.2 総合監視ツール

本研究では上記のフレームワークを元にシステムを開発した。総合監視ツールには Nagios（Nagios Ain't Gonna Insist On Sainthood Nagios）を採用した。

Nagios には監視のためのプラグインが豊富にあるため，基本的な機能の開発コストを削減できる。プラグインは Nagios が監視しているマシンに対して行う動作を決定する。Nagios ではマシンやサービスに障害が起こった場合などに通知する機能や，Web インタフェースによってマシンの状態を閲覧出来る機能がある [4]。

4.3 開発システムの機能

本システムの機能において，監視機能と診断機能は Nagios の追加機能となり，プラグインとして認識され，実際にサービスを動かすときに Nagios が読み込んで使用する。また，表示や通知は Nagios の機能を用いた。以下に本研究で開発・実装した機能を示す。

4.3.1 監視機能

グリッド・システムの監視には，そのグリッド環境に合わせた機能（コマンド）を利用する必要がある。そこで，本システムでは，グリッド環境を操作することが出来る UI 上に 4 つの監視機能を展開した。

(1) 応答監視

応答監視では，各マシンに ping などを利用して応答を確認する。また，各マシン上で必要なポートの開閉が正しく行われているかどうかも監視する。

(2) グリッド機能監視

グリッドの環境を監視する特有のプラグインとして，CE では gLite のジョブを正常に実行できるかを各サイトで監視する。また，各 RB と各 CE 間が正常に接続できているかの確認も行う。SE ではディスク容量や，SE の情報が正しく得られているか，GridFTP の機能は正常かどうかを監視する。LFC ではファイルカタログの操作が正常に行えるかを監視する。RB においても GridFTP の監視をしている。

(3) 通信監視

gLite の機能を使用したジョブにより，各 CE 間の RTT（Round Trip Time）の監視や，SE 間の GridFTP を利用したファイル転送速度の監視を行う。RTT やファイル転送速度の推移から，混雑している時間帯を避けてジョブを投入することが可能となる。

(4) 内部監視

マシンの内部情報である CPU やディスクの使用率などを監視する。内部情報の取得には gLite の機能を使用して

いる。

4.3.2 診断機能

障害が発生した場合，プラグインはその障害に対する診断情報などの対処策を記述した診断用データベースとしてwikiへ誘導する情報を提供する。これにより既知の障害ならば迅速に対処できることになる。また，wikiに書かれていない障害の場合，対処を行った管理者がwikiに記述して情報を発信することも出来る。障害時のデータは検索機能としてWebから閲覧出来る。

4.3.3 情報共有機能

各プラグインでは記録機能として，動作や処理結果をデータベースへ記録することを行っている。そのデータベースから記録を選別してwikiに表示することが出来るwikiのプラグインを作成した。このプラグインを利用し，データベースから最新の情報をwikiへ表示させることでデータベースの情報が共有可能となる。

本システムのwikiでは，編集機能を関係者のみに許可しているので，これらの情報は信頼性の高い情報と考えられる。また，wikiは誰でも閲覧することが出来るので，関係者間での情報共有と共に，システムに興味のある人などへの情報提供も可能となる。

4.3.4 可視化機能

本システムは，Nagiosによる監視・診断結果を目に見える形にし，使用者に分かりやすく提供する可視化機能を持つ。可視化機能は，以下の3つに分類できる。

(1) ステータスマップによる可視化

プラグインから得られた情報を可視化し，ステータスの一覧画面を生成する。この機能はNagiosのものだが，プラグインから出力する情報により一覧画面の表示をある程度コントロールできる。本システムでは1つのプラグインで複数個所の監視結果を表示することを可能とする。

(2) グラフによる可視化

プラグインからのデータはデータベースに記録されている。そのデータを利用してグラフを生成する。過去のデータを利用できるため，時系列にデータをグラフとして可視化することが出来る。

(3) 地図による可視化

プラグインから得られる最新の情報で，地図上に線やマーカーを表示して,サイト間の通信速度などを可視化する。これにより，監視の情報だけでは把握できなかった，サイト間の位置や距離を考慮して監視結果を評価することが出来るようになる。

5. システムの運用

本研究ではScientific Linux 3.0.8上でNagios 3.0a4を利用してシステムを構築した。監視状態の表示画面の例を図3に示す。

図3では，各マシンの監視結果や，各プラグインの設定情報をNagiosの機能で一覧表示している。各プラグインは一定間隔で指定されたマシンやサービスを監視し，それが正常に動作しているか否かを示す。管理者はその情報を元にマシンを管理できる。

また，wiki，グラフ，地図表示などで情報の可視化を行い，監視結果をより分かりやすく表示する。地図表示の例を図4に示す。図4ではgoogle maps API 2を使い，各サイト間のRTT情報を表示している。これによりRTTによる通信回線状況が迅速に確認できる。

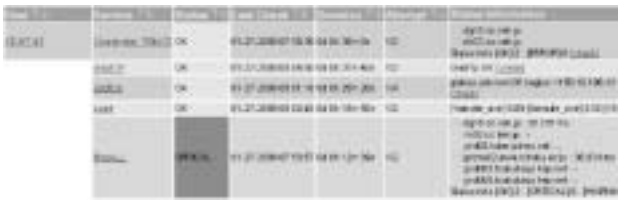

図3. Nagiosによる監視状態の表示画面例

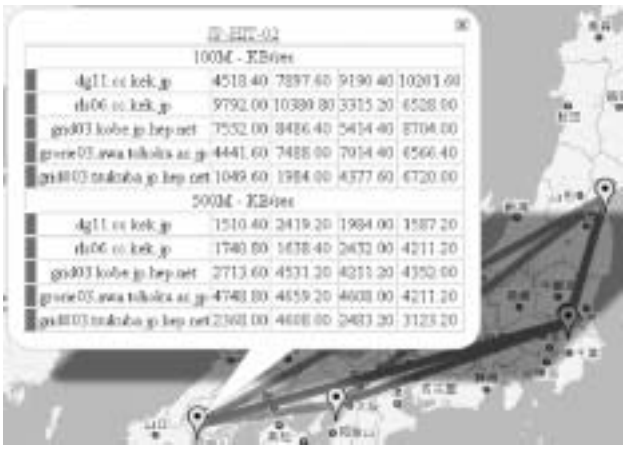

図4. google maps API 2を用いた地図の表示例

本システムでは，全ての監視を自動的に行い，異常があれば診断情報と共に通知する。このことで障害を迅速に認識し的確な処置を行うことが出来ると考える。さらに，手作業の部分を大幅に削減することができ，システムの運用に関わる人的リソースを減少させることができる。

これらのことから，管理者には各拠点の情報を簡潔に提供でき，少ない管理者でグリッド・システムの運用を行うことが可能となる。

6. 性能評価

gLite のジョブを使用して監視対象の CE から他の CE へ ping を行い，通信速度を測定するプラグインの処理時間を，ping に掛かった時間（PingTime）と，ping を除いた gLite のジョブに掛かった時間（gLiteTimeA），プラグイン全体の処理時間から PingTime と gLiteTimeA を除いた時間（PluginTime）の 3 つに分けて測定した。図 5 に，左側を縦軸に ping サイト数ごとに処理時間を積算したグラフと，右側を縦軸に gLiteTimeA に標準偏差のエラーバーを付けた折れ線グラフ（gLiteTimeB）を示す。グラフの横軸は ping する対象サイト数，両側の縦軸は処理時間である。プラグインの監視対象は広島工業大学の CE とした。

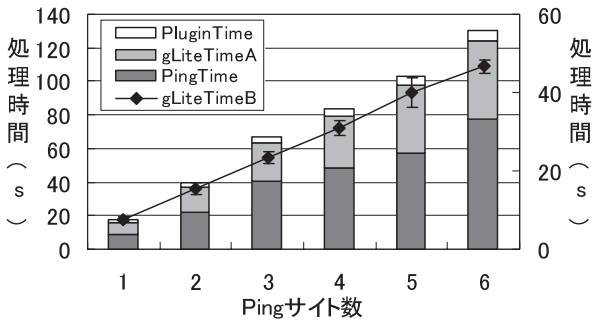

図 5. 通信速度測定プラグインの処理時間比較

図 5 より，どの処理時間も ping サイト数が増えるごとに増加していることが分かる。PluginTime は，サイト情報の処理に時間が掛る為に処理時間が増加する。PingTime は Ping サイトごとに 1 秒間隔で ping を 10 回行い，通信速度を測定しており，正常に ping が行えれば約 9 秒掛かり，パケットロスなどの異常が一度でも起きると ping コマンドの処理に時間が掛かり，さらに処理時間が 10 秒程度増加する。gLiteTimeB では，ping サイト数が 5 の場合を最大に，全体的に処理時間が変動していることが分かる。これは，gLite では仮想組織の管理やジョブのスケジューリングを考慮しており，同じ処理でも処理時間は変動する為である。

上記の結果から，ping や gLite の機能を除いたプラグインの処理は，他と比較して時間が掛からないことが分かった。時間の掛かる ping の処理は調整できるため，必ずしも必要となる処理時間ではない。よって，本システムでは監視に掛かる時間を短縮する場合，最終的にボトルネックとなる箇所は gLite の処理時間である。

7. 考察

本研究で利用した Nagios は，それ自体の機能が限られているので，いかに機能を拡張し，自動監視・診断システムの要件に近づけるかが問題となる。その点は，Nagios のプラグインの開発や Web ベースのインタフェースなどを利用して機能を拡張し，対応した。

一般的な監視では，監視時間が長いと頻繁に監視できないため監視精度が落ちる。これについては，Nagios ではプラグインの監視時間を短くし，監視間隔を Nagios で調整するのが望ましい。本システムではプラグインの監視時間を短くするには，監視に掛かる必要な時間の大半が gLite の処理時間であることから，gLite を調整する事で処理時間の改善が望めると考えられる。

本システムは Nagios を基盤に動作し，実際に gLite で構成されたグリッド環境を監視・診断している。さらに，Nagios は無償でありコスト面で優れていることから，Nagios を基盤とした自動監視・診断システムはグリッド環境を監視する上で有用であると考える。

8. まとめ

本研究では，グリッド環境を監視・診断するフレームワークを提案し，それを共にグリッド環境を自動監視し，診断を行うシステムを開発した。本システムを用いることで，障害が起こった場合にその障害を診断し，必要な情報を管理者に通知することが出来る。更に，wiki ページなどで障害に関する診断情報の提供や，グラフ，地図表示で情報を可視化し分かりやすく提供することで，迅速な対応が行えるようになった。

今後は，監視の広域化として複数のグリッド・ミドルウェアの監視を行うと共に，大規模化への対応を念頭に置いた処理時間の短縮による監視精度の強化と，さらに可視化を充実させる必要があると考える。

参考文献

[1] 藤井峰夫，長坂康史，渡瀬芳行，佐々木節，岩井剛 FIT2007 第 6 回情報科学技術フォーラム，p5-6，L-003，2007/9/5
“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”
“An automatic monitoring and diagnostic system of the grid environment in the Accelerator Science Virtual Organization”

[2] 独立行政法人　産業技術総合研究所　グリッド研究センター，丸善株式会社：産総研シリーズ　グリッド

—情報社会の未来を紡ぐ—（2004）
［3］藤井峰夫，長坂康史　FIT2006 第 5 回情報科学技術フォーラム，p53-54，L-021，2006/9/5
“グリッド技術を用いた高効率データ収集システム”
“A high efficiency data acquisition system with GRID technology”
［4］斉藤省吾 著，毎日コミュニケーションズ，2006
Nagios 2.0 オープンソースではじめるシステム&ネットワーク監視