ハイパフォーマンス
コンピューティング 93－4
（２００３．３．１１）

SNMP によるクラスタ性能計測手法の検討と評価

近藤秀樹† 石橋拓也† 建部修見††

クラスタシステムの全ノードについてネットワーク転送性能，CPU 負荷などをサブ秒単位で精密に計測することにより，ネットワーク転送のフロー制御や計算アルゴリズムの CPU 負荷の挙動を把握することができ，アルゴリズムや各種パラメータのチューニングに有用である．本論文では SNMPv1 を利用して全ノードを計測する手法を提案し，256 ノードのクラスタで評価を行った．その結果，ネットワーク積算送受信量，ディスク使用量，空き容量，CPU 負荷の 5 項目の 241 ノードに対する計測が，計測結果のディスクへの書き出しも含め 0.036 秒で可能となった．

Cluster-wide Precise Performance Measurement using SNMP

Hideki Kondo,† Takuya Ishibashi† and Osamu Tatebe††

Precise performance mesurement of network bandwidth and load average for every cluster node makes it possible to understand the behaviour of flow control of network transfer and CPU usage of applications, and to improve the performance by tuning various parameters. This paper proposes a cluster-wide performance measurement method using asynchronous SNMP requests, and evaluates on a 256-node cluster. As a result, it takes only 0.036 seconds to measure total amount of network transmit and receive data, used and available disk space, and CPU load average of 241 cluster nodes.

1. はじめに

Grid Datafarm[1] など高性能クラスタシステムの研究開発において，システムの性能向上のためクラスタの全ノードに対する高精度な性能計測が必要とされる．例えば，クラスタ間の広域高速データ転送を実現するためには，さまざまな通信パラメータの設定が必要であるが，その設定のためにはそれぞれのノードにおける精密なネットワーク性能計測が鍵となる．さらに，性能チューニングのためには計測結果は実時間で可視化されることが望ましい．性能チューニングのための性能計測手法に要求される項目をまとめると以下のようになる．

- 0.1 秒程度の短い間隔で連続的に計測が可能である．
- 256 ノード程度を同時に計測可能である．
- 計測対象に負荷をかけない．
- 計測したデータを全て保持する．
- 計測結果を実時間で可視化できる．

本稿では SNMPv1[2] を利用したクラスタの性能計測方式を提案する．この方式では，計測対象となるクラスタのノードとは別に，計測用の計算機を使う．計測プログラムは計測対象に対して SNMP 要求を発行し，各計測対象から返される SNMP 応答を非同期的に処理し，ファイルに記録する．本方式を実装した計測プログラム glogger を用いて 256 ノードのクラスタを計測し，性能を評価する．

† 株式会社創夢
SOUM Corporation
E-mail: {hkondo,takuya}@soum.co.jp
†† 産業技術総合研究所グリッド研究センター
Grid Technology Research Center, National Institute of Advanced Industrial Science and Technology
E-mail: o.tatebe@aist.go.jp

2. 関連研究

2.1 モニタリングツール

代表的なクラスタシステムのモニタリングツールとして ganglia[4] や SNAPP[7] などがある．ganglia は UCB の Millennium プロジェクトで開発されたツールである．全ての計測対象ノードに gmond というデーモンを組み込み，各ノードの gmond が個別にロードアベレージやネットワークインタフェースの統計値を計測し，マルチキャストを用いてノード間で計測値を交換する．計測値は gmetad という別のデーモンを利用し，ウェブブラウザから見られるグラフを生成する．SDSC Rocks クラスタや PrestoIII をはじめ，数百ノードのクラスタの管理にも用いられている．しかし，各ノードが個別に自身の統計値を計測し，その値を集約するという方法をとっているため，ほぼ同時刻のクラスタ全体の状態を計測することはできない．また，計測インターバルは 30 秒 ∼900 秒程度を乱数に

より変化させており，0.1 秒間隔といった短いインターバルにおける計測は考慮されていない．

SNAPP は Indiana University で TransPAC プロジェクトのために開発されたツールである．マルチスレッドで SNMPv1 を用いて計測を行う．ganglia 同様のグラフ生成機能を持ち，また計測が失敗したり計測値が threshold を越える等の事象が発生した場合に電子メイルで管理者に通報する機能がある．しかし計測インターバルは秒単位でしか設定できず，やはり時間的に精度の高い計測はできない．また，どちらのツールも Round Robin Database[6] を使用しており，過去の計測データはいずれ集約されてしまうため，必要な計測データをあらかじめ抽出しておかなければ捨てられてしまう．

こうしたツールは，定常運転中のクラスタのおおまかな動作を把握したり，ハードウェア故障等の異常事態を監視するためのものであり，経過を観察しつつチューニングを行うような場合には不適切である．TCP の挙動を計測しつつ様々な通信パラメータの設定を変更して，バンド幅のチューニングを行うような場合には，種々の設定条件を変更したときのバンド幅の変化を即時的に計測，表示できることが望ましく，既存の監視ツールではこの要求は満たされない．

2.2 SNMPv1

SNMPv1 は 1988 年に RFC1067 で決定され，1990 年に表現を訂正した RFC1157 として更新された．これに準拠したフリーソフトウェア ucd-snmp[8] 等を用いることで，クラスタのノードとなるような計算機システムでも利用することができる．クラスタのネットワークスイッチに関しても，ほとんどの高性能ネットワークスイッチが管理機能として SNMPv1 を実装している．スイッチを計測することにより，クラスタ間の通信を計測することもできる．

SNMP を用いて標準的に計測可能な項目は，RFC1158 で MIB-II として 1990 年に規定されており，前述のネットワークスイッチや ucd-snmp で計測できる項目もこれに準拠している．MIB-II ではネットワークインタフェースの送受信量の積算やノードの uptime 等が取得できると規定している．アクセス方法をベンダーが公開しているため，RFC1158 で規定されていないような機器固有の事象についても，MIB-II と同様に SNMP 経由で計測が可能である．例えば，前述の ucd-snmp は，ディスクの空き容量やロードアベレージ等を取得できるよう拡張され，SNMP 経由で取得する方法がドキュメント中で明らかにされている．

3. glogger 計測方式

本稿で提案する glogger 計測方式を図 1 に示す．glogger 計測方式は，計測プログラム glogger を実行する計算機 (以下，計測機) と計測対象ノードや計測対象スイッチ (以下，計測対象) からなる．まず，計測に SNMP を用いるため，計測対象にはあらかじめ SNMP エージェントが組み込まれ，計測機と計測対象は SNMPv1 で通信できるとする．本方式では，計測機と計測対象が次のように非同期的に通信を行いながら計測を行う．

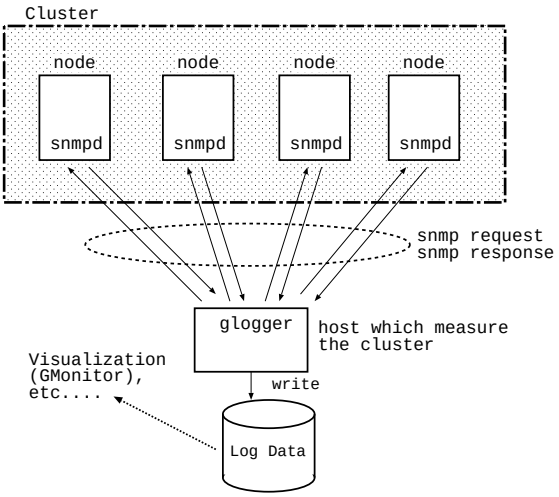

図 1 glogger 計測方式

(1) 全ての計測対象に対して，計測機が SNMP 計測要求を発行する．
(2) 計測対象は SNMP 計測要求を受信し，要求内容にしたがって自身の状態を計測する．
(3) 計測の終った計測対象は，計測値を SNMP 応答として計測機に返す．
(4) 計測機は計測対象から戻った SNMP 応答を受信し，計測値を計測機側のファイルに記録する．

これらの手続きを計測インターバル毎に繰り返し，連続的に計測を行う．計測インターバルは運用時に決定される．また，計測インターバルの終了時刻までの残り時間のことをタイムスライスと呼ぶことにする．計測インターバルが開始されると，まず最初に，計測機は全ての計測対象に対して順に SNMP 計測要求を発行する．ここでは UDP によって要求を発行するだけであり，計測対象からの応答を待つことはない．マルチキャスト等の方法で全ての計測対象に要求を発行できることが望ましいが，SNMP の通信には通常の UDP を用いなければならず，全ての計測対象について UDP を同時に送信する方法がないため，各計測対象について順に次々と要求を発行する．また，同一計測対象において複数の事象を計測する場合は，SNMP 計測要求は単一の要求にまとめられ，送信される．

要求発行が完了したところで，計測機は select() による受信待ち状態になる．1 つ以上の計測対象からの SNMP 応答を受信するか，もしくはタイムスライスを超過してしまうまで，計測機は何もしない．

各計測対象が SNMP 計測要求を受信すると，計測対象に組み込まれた SNMP エージェントがこの内容を解析し，計測を行う．複数の事象を計測するような要求であれば，要求されている全ての事象を順に計測する．全ての事象について計測が完了した後，計測対象の SNMP エージェントは SNMP 応答を組み立て，計測機に対して UDP で送信する．

select() によって受信待ち状態になっている計測機は，1 つ以上の SNMP 応答を受信すると受信待ち状態を抜け，受信した応答の解析と記録処理を行う．応答から計測値を取り出し，ファイルをオープンして計測時刻と計測値を記録し，ファイルを閉じる．その瞬間に受信している応答全てについて，この処理を行う．受信した応答をすべて記録し終ったところで，その時点での時刻を調べる．まだタイムスライスを超過しておらず，応答を受信していない計測対象が残っていた場合，計測機はタイムスライスを超過するまで引き続いて select() による受信待ち状態に戻る．

もしタイムスライスを超過していた場合は，まだ応答を受信していない計測対象を計測不可能として記録する．計測値の記録と同様，記録ファイルをオープンし，計測時刻とエラーを記録し，ファイルを閉じる．応答を受信できなかったすべての計測対象について記録を行い，このインターバルの処理を終了する．

すべての計測対象の記録を完了してもなおタイムスライスを超過していなかった場合は，次の計測インターバルまで待ち状態となり，タイムスライスを超過した時点でこのインターバルの処理を終了する．

このインターバルでの処理が終了したら，次の計測インターバルを開始する．

計測対象での計測処理に時間がかかり，インターバルをまたいで応答を返してきた場合は，単純に無視される．何故なら，直前のインターバルで応答を受信できず，間に合わなかったため，エラーとして記録されているからである．

本方式でのすべての SNMP 通信は UDP で行われるため，要求や応答が正常に到達しない可能性がある．この場合も，計測が不可能であったとして処理され，エラーとして記録される．SNMP 要求が到達しなかった場合は，計測対象の SNMP エージェントは何も処理を行わないため，結果として計測機には何の応答も返されない．いずれ計測インターバルが終了し，計測機はエラーを記録する．計測対象からの SNMP 応答が計測機に到達しなかった場合も，同様である．

典型的な処理の流れを図 2 に示す． インターバル開始時に SNMP 計測要求を全て発行し，非同期的に戻ってくる SNMP 応答を処理している様子を表したものであり，select() が受信待ちを示し，arrival 以降が受信処理開始を示す．全ての応答を受け取って処理した後，さらにタイムスライスが残っているので，usleep() で次のインターバル開始を待っている．

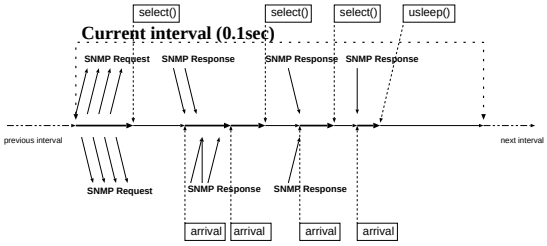

図 2 典型的な計測処理

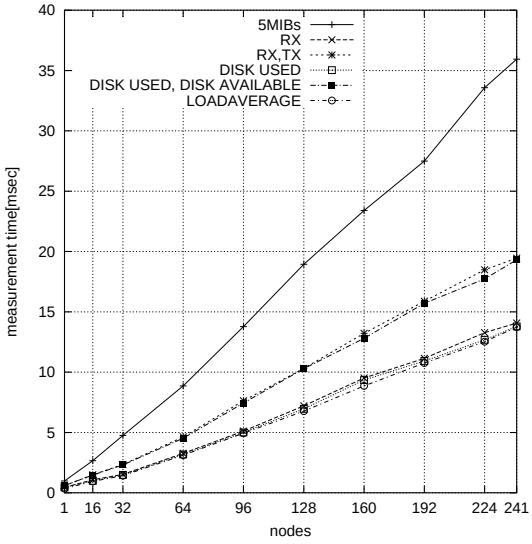

図 3 実行時間の変化

4. 評　　価

4.1 PrestoIII の計測

本稿で提案する計測方式を用いた計測プログラム glogger を実装し，性能評価を行った．SNMP による非同期通信には ucd-snmp ライブラリを利用して実装した．計測対象として東工大松岡研究室の PrestoIII を利用した．PrestoIII は 256 ノードの PC で構成されるクラスタである．各ノードは AMD Athlon MP 1900+ × 2, 主記憶 768MB，ネットワークインタフェースとして Myrinet 2000 と 100Base-TX を備えている．計測機としては，Athlon 1.2GHz，主記憶 256MB，Gigabit Ethernet を備えたホストを利用した．

PrestoIII のノードを対象として，0.1 秒間隔での計測を行った．32 ノードずつ 241 ノードまで計測対象を増やしながら計測を行った際の 1 インターバルあたりの計測完了時間の最小値の変化を図 3 に示す．

各ノードで計測した事象は次の通りである．

- RX (ネットワークインタフェースの積算受信量)
- RX, TX (ネットワークインタフェースの積算受信量，積算送信量)

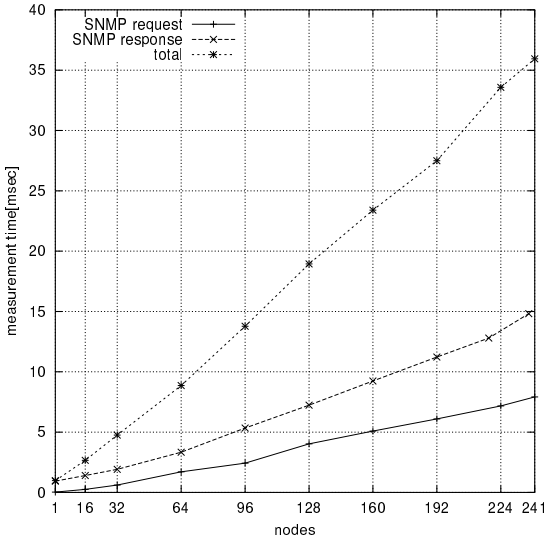

図 4 5 事象計測の内訳

- DISK USED (ディスクパーティションの使用量)
- DISK USED, AVAILABLE (ディスクパーティションの使用量と空き容量)
- LOADAVERAGE (計測対象ノードのロードアベレージ)
- 5MIBs (RX, TX, USED, AVAILABLE, LOADAVERAGE)

0.1 秒間隔で 241 ノード 5 事象を計測した場合でも，glogger はインターバル開始から 36 ミリ秒程度で計測を完了していることが分かる．これは 0.036 秒であり，240 ノード程度のクラスタに対しては，本方式が十分な性能を持っていると言える．ノード数の増加に対して処理時間は直線的に伸びており，性能が低下する傾向も見られない．受信処理を行ったうえで計測値の書き込みを全く行わなかった場合，インターバル完了は約 15 ミリ秒であった．

5 事象を計測した際の実行時間の内訳を図 4 に示す． SNMP request が要求発行終了にかかった時刻，Total が全計測値の記録完了時刻を示す．また，記録を行わず，全 SNMP 応答が到着した時間を SNMP response で示す．SNMP response の時間は 238 ノードまでしか計測していない．241 ノードの場合，インターバル開始から約 7.9 ミリ秒の時点で SNMP 要求が完了している．応答受信処理と計測値の書き込みも含めたインターバル完了は 36 ミリ秒である．インターバル完了のグラフは傾きが約 140 であるのに対して，SNMP 要求完了のグラフは傾きが約 30 となっている．計測対象ノードが増加するほど，SNMP 要求にかかる時間とインターバル完了までの時間が広がっており，SNMP 応答の記録の処理が支配的になっているのがわかる．

4.2 snmpget コマンドとの比較

ucd-snmp には snmpget コマンドが含まれており，コマンドラインで任意の対象に対して SNMP 計測要求を発行し，応答を待って計測値を出力することができる．snmpget コマンドをバックグラウンドプロセスとして利用することにより，シェルスクリプトで glogger 計測方式と同等の計測を行うことができる．

表 1 は，自ノードを含む計測対象 2 ノードに対し 10,000 回 SNMP 要求を発行し，応答を処理する時間を glogger と snmpget コマンドで比較したものである．実行した計算機は 1.8GHz Pentium 4 である．この結果，glogger は snmpget コマンドを利用するより 130 倍程度高速であった．これは glogger では C の API レベルで SNMP を利用し 1 プロセスで実現しているのに対し，snmpget コマンドを利用する場合は複数プロセスを利用しておりオーバヘッドが大きくなっているからである．

表 1 性能評価プログラムの計測結果

C 版	7.4 秒 (256 回あたり 0.1 秒未満)
シェルスクリプト版	16 分 45 秒 (256 回あたり約 13 秒)

5. 議論と今後の課題

本方式には十分な性能があると確認できたが，実際の計測にあたっては，意識しておくべき制約があることもわかった．以下に本方式による計測時の制約と，検討すべき課題を示す．

5.1 超大規模クラスタの計測

PrestoIII 256 ノードのうち，241 ノードを使って評価を行った．本方式はこの規模のクラスタを計測するには十分な性能であることが分かった．計測時間の伸びも直線的であり，256 ノード程度であれば問題なく計測できると言える．このまま計測時間が伸びると仮定すると，700 ノード程度までは現状のまま計測可能である．また，以下に議論するように glogger の実装にもまだ問題があると考えられる．本方式の限界を調べるために，さらに大規模なクラスタシステムを計測する必要がある．

5.2 計測対象の計測値更新間隔の問題

SNMP では計測対象の更新間隔に関する規定がなく，実装の差により精度が異なることがある．UNIX ライクな OS では ucd-snmp を共通に利用することができるものの，利用するハードウェアや OS によって，精度に差がある．ネットワーク機器に搭載されている SNMP 管理機能についても，ベンダーが独自に実装しているため，その精度は他の実装とは異なっている．実際の計測において，ノードとスイッチにおいて精度の異なる事例が観測されている．

5.2.1 ucd-snmpd によるネットワークインタフェースの計測の限界

評価では Linux 2.4.20 を用いて計測を行った．このとき，Intel Gigabit Ethernet Adapter を用いたネットワークインタフェースを計測すると，計測値は 2 秒間隔でしか更新されない．同様の環境で Intel PRO/100 Management Adapter を用いたネットワークインタフェースを計測した場合には，計測インターバル毎に計測値が更新される．Intel Gigabit Ethernet Adapter は e1000 ドライバで，Intel PRO/100 Management Adapter は eepro100 ドライバで駆動されている．ネットワークインタフェースによって計測値の更新間隔が異なるのは，ucd-snmp が次に示すような計測を行っているからである．

- Linux で動作する ucd-snmp はネットワークインタフェースに関する統計を/proc/net/dev から取得している．
- /proc/net/dev は通常ファイルではなく，linux/net/core/dev.c の dev_get_info() の出力となっている．
- linux/net/core/dev.c の dev_get_info() は，デバイスドライバの特定のエントリ get_stats() を呼び，値を返す．
- eepro100 ドライバ☆では，呼び出される毎にハードウェアから統計値を取得して返している．(speedo_get_stats())
- e1000 ドライバ☆☆では，get_stats() 呼び出しとは関係なく 2000 ミリ秒毎に統計値を更新しており，/proc/net/dev エントリではその値を返している．

e1000 で駆動されるネットワークインタフェースを計測する場合，有意な計測が可能な計測インターバルは 2000 ミリ秒単位であることが分かる．実際に計測する際には，デバイスドライバの特性によって計測の精度が左右されることを意識しておかなければならない．

5.2.2 3Com Gigabit Switch の計測の限界

クラスタのノード間を接続する 3Com 製 4924 Gigabit Switch が搭載している SNMP エージェントを利用して，ポートあたりのデータ流量を計測していたところ，計測値が変化するタイミングが不規則に変動することがあった．同じスイッチに接続されたノード間でファイルコピーを行いつつ，ノードの接続されたポートの送受信量積算値を 0.1 秒単位で取得しようとした際，概ね 1 秒単位で変化していたが，0.6 秒 ～1.2 秒で変化している事例もあった．ファイルコピー等の通信がない場合でも，概ね 1 秒単位で変化しているが，0.2 秒 ～0.3 秒で変化した場合があった．更新タイミングの規則性はみられなかった．SNMP の管理機能よりもスイッチ本来のタスクのほうが優先されているためではないかと推測しているが，詳細な検証はできていない．いまのところネットワーク機器を計測する場合は，計測精度がそれほど高くない可能性があることを十分注意しなければならない．

☆ linux/drivers/net/eepro100.c
☆☆ linux/drivers/net/e1000.c

5.3 計測値記録速度の影響

計測値の記録先を NFS マウントしてるディレクトリに指定した場合，せいぜい数秒単位での計測ができる程度であり，0.1 秒間隔ではまったく動作が間に合わなかった．このことから，現時点での計測プログラムの動作は，計測値の書き込み先の性能に非常に大きく影響されるといえる．

5.4 計測値記録処理の影響

図 4 の各グラフの傾きの違いから，計測プログラムの計測性能は，SNMP による計測時間よりも，計測値の記録先の性能に大きく左右されていることがわかった．

現在の実装では，ノードから計測結果が戻ってくる毎に記録ファイルを open(2) し，計測値を記録し，close(2) している．これは性能に大きく影響していると考えられる．これはファイル単位に事象やノードを分けて記録するという方針による．ファイル名に規則性を持たせて取り扱いを容易にするために採用した方針だが，性能に対する影響が無視できない場合があることは明らかである．

できるだけ少ないファイルにまとめて書き込むことができるよう，構造化されたファイルフォーマットを検討する必要がある．たとえば単一のファイルに 256 ノードのまる 1 日分の記録を格納する等，構造を工夫することで，open(2), close(2) の回数を減少させることが可能だと考えられる．

データフォーマットを変更しなくとも，書き込みをバッファリングして，ある程度まとまった量を書き込むことで，open(2), close(2) の回数を減らすこともできると思われる．計測の都度書き込むのをやめ，たとえば 10 インターバル毎に，10 インターバル分の計測値を書き込むようにすると，open(2), close(2) の回数が 1/10 に減ることになる．ただし，1 事象分の計測値はせいぜい数 bytes であるが，256 ノードの計測値を 10 インターバル分バッファリングすると，数百 K bytes～数 M bytes 程度になると予想できる．また，これを単純に 10 インターバル毎に書き込むとすると，そのインターバルに全ノード分の書き込みが集中することになる．ノード数に比例した open(2), close(2) が一度に実行されることとあわせて，次のインターバルの計測ができなくなる可能性があるため，インターバル毎に書き込みを分散させる必要がある．

5.5 遠隔計測の可能性と監視ツールとしての利用

glogger の出力した計測記録は，別途 Java で実装した gmonitor というグラフ化ツールで観測できる．こ

の可視化ツールは準リアルタイム更新が可能であり，1 秒毎に glogger の出力を読み取ってグラフを描画する．glogger と gmonitor を組み合わせて用いることで，クラスタの性能変化を視覚的に表現することができる．現在の gmonitor は，最新 1 時間程度と過去の任意の時点とをグラフ化することができるため，性能調整時にも過去の傾向を併せて参照することができる．こうしたツールと併用することで，glogger を ganglia や SNAPP のような運用監視用途へ転用することも検討している．

6. ま と め

本稿では，クラスタの全ノードに対する軽量で高性能な計測を目的として SNMPv1 を非同期に処理する計測方式を提案し，200 ノード程度のクラスタノード全体に対して計測の評価を行った．その結果，241 ノードに対する，ネットワーク積算送受信量，ディスク使用量，空き容量，CPU 負荷の 5 項目の計測が，計測結果のディスクへの書き出しも含め 0.036 秒で実行できた．また，デバイスドライバやハードウェアによっては有意な精度が確保できない場合があり，SNMP の実装に起因する注意点として認識することができた．

SNMPv1 は本来，ネットワークの監視のためのプロトコルであるが，ネットワークスイッチなどにも広く普及しており，クラスタの性能評価や調整のために利用価値の高い情報を提供している．フリーソフトウェアの SNMPv1 実装を用いることで，クラスタのノードへの組み込みも容易である．場合によっては必要な精度が確保できない可能性もあるものの，汎用的な枠組を用いるだけで計測ができることがわかった．

現在は準リアルタイムでグラフ化を行う可視化ツール gmonitor と併用し，ローカルネットワークから性能の計測が可能となっているが，ganglia や SNAPP などに似た定常運転時の観測システムとして転用することも検討している．

また，Grid Datafarm で必要とされる広域の複数クラスタを利用した性能調整が必要な場合など，遠隔地のクラスタの精密な計測結果を実時間で表示することが必要となる．このためには Java ベースの可視化ツール gmonitor をウェブブラウザ上で動作させ，遠隔地から計測や観測を行うようなことが考えられる．Java アプレットの実行環境を備えたウェブブラウザも広く使われている技術であり，どこにでもあるありふれた環境で遠隔地のクラスタを準リアルタイムで計測・監視することも検討している．

謝　　辞

大規模クラスタの計測にあたり，PrestoIII への snmpd の設定などで協力いただいた東工大松岡研究室クラスタチームの皆様に感謝致します．また，本稿のプログラムを作成するにあたって多くの技術的なアドバイスをいただいた，株式会社創夢遠藤知宏副社長に感謝致します．なお，この研究は経済産業省平成 14 年度重点分野研究開発依託費 (構造特別枠)「ネットワークコンピューティング技術の開発」によるものである．

参 考 文 献

1) 建部 修見, 森田 洋平, 松岡 聡, 関口 智嗣, 曽田 哲之: ペタバイトスケールデータインテンシブコンピューティングのための Grid Datafarm アーキテクチャ, **情報処理学会論文誌：ハイパフォーマンスコンピューティングシステム**, Vol. 43, SIG 6(HPS 5), pp. 184–195 (2002).
2) Case, J., Fedor, M., Schoffstall, M. and Davin, J.: A Simple Network Management Protocol, *Request for Comments* (1990) `http://www.ietf.org/rfc/rfc1157.txt`
3) Rose, M.: Management Information Base for Network Management of TCPIP-based internets: MIB-II, *Request for Comments* (1990) `http://www.ietf.org/rfc/rfc1158.txt`
4) Ganglia project: ganglia toolkit:: a distributed monitoring and execution system, `http://ganglia.sourceforge.net/`
5) Oetiker, T.: MRTG: The Multi Router Traffic Grapher, `http://people.ee.ethz.ch/~oetiker/webtools/mrtg/`
6) Oetiker, T.: RRD TOOL – About RRDTool, `http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/`
7) Fowler, L.: SNAPP Distributioin Site, `http://loadrunner.uits.iu.edu/dist/index.html`
8) NET-SNMP Project: NET-SNMP Home Page, `http://net-snmp.sourceforge.net/`