Vol.2012-HPC-133 No.20
2012/3/26

分散制御システムにおける設備管理サーバの大規模化検討

鈴木清彦†

近年の制御システムは急速な大規模化に対応すべくオープンな技術の活用が進んでいる．近年の分散処理技術は Web 分野にて著しい発展を遂げたが，クラウドに代表される Web 系のシステムと制御システムとはデータフローが異なるため，単純に適用することが出来ない．Web システムでは集中的に殺到するリクエストを分散処理することに長けるが，これに対して，制御システムは各々のサブシステムが元々独立に動作しており，これら独立に動作するサブシステムを設備管理サーバが統一的に管理する．即ち，制御システムはデータフローが Web システムと反転しており，制御システムを大規模化するためには，ボトルネックとなる設備管理サーバの処理を分散する必要がある．即ち，制御システムの大規模化には，独立して動作するサーバ間でデータを共有でき，かつ名前空間を持たずにこれを実現する仕組みが必要である．

本論文では分散制御システムに適したオープン技術の比較を行った．その結果，要求を満足する 1 つの技術として GlusterFS が存在すると判明したため，GlusterFS を用いた分散制御システムの評価結果を報告する．

Large Scale Sizing of Facility Management Server in Distributed Automation Control System

KIYOHIKO SUZUKI†

Recently, a control system is a way which is enlarged. Therefore An open technique is utilized with a control system. The distributed processing technology developed rapidly in the field of Web. However, the control system is different from a Web system in data flow. The Web system is good at performing distributed processing of a concentrated request to do. In contrast, as for a control system, each subsystem works independently. And all subsystems are managed integrally by facilities management server. It is necessary to distribute by the processing of facilities management server becoming the bottleneck to make a control system a large scale.

In this paper, compared the open technology of a distributed control system. As a result, it became clear that GlusterFS satisfies our demand. The evaluation result of a distributed control system using GlusterFS is reported.

1. はじめに

FA(Factory Automation system)や BA(Building Automation system)などに代表されるオートメーション系制御システム(Automation Control System：ACS)はかつて集中制御システムが主流であった．しかしシステムの複雑化，受益者ニーズの高度化及び多様化や, システムが担う責任の増大に伴い, 集中制御方式から分散制御システム(Distributed Automation Control System:DACS)へと主流が移行した．この結果，ACS ではシステム構成の変更容易化が進み，そして対故障性が向上した．

近年，低炭素社会の実現や省エネ意識の向上から ACS の導入が進んでおり，加えて大規模災害に伴う電力供給量のひっ迫から，より細やかなエネルギー制御による省エネ効率の向上が求められている．その結果，ACS では各種センサなどの取付密度が高まり，システム管理点数の急速な大規模化が進んでいる．

近年の ACS では急速な大規模化の要求に早急に対応すべく, オープンな技術の活用が進んでいる. しかしながら DACS は集中制御システムの側面も未だ有しているため，クラウドに代表される Web 系のシステムとデータフローが異なり，単純な適用が出来ない．

そこで本論文では，DACS のデータフローに着目することで，DACS の大規模化に求められる要求を明らかにした．この要求の下で各種オープン技術の比較を行った．その結果，要求を満足する 1 つの技術として GlusterFS が存在すると判明した．GlusterFS を用いた分散制御システムの評価を実施したため，この結果を報告する．

2. 分散制御システム

2.1 DACS の構成

DACS の一例を図 1 に示す．近年の DACS は一般的に Web ブラウザからシステム全体の管理が可能であり，システム内の各種機器状態の確認や運用の設定などが行える．こうした管理機能はシステム全体を管理する設備管理サーバが提供するため，設備管理サーバは通常 Web サーバ機能を備える. 設備管理サーバは管理者が指示した運用設定に基づき，各 LCS(Local Control Server)ごとに適切な運用設定を行う．LCS はサブシステムを管理するサーバであり，一例として BA システムであれば，ビル内の空調，照明，防犯など，機能システムの個々のサブシステムを担当するサーバである．LCS はひとたび運用設定を受け取ると，その情報を元にサブシステム内の設備を管理・制御する．すなわち，個々のサブシステムは他のサブシステムと独立して動作す

† 三菱電機(株)　情報技術総合研究所
Mitsubishi Electric Corporation Ltd. Information Technology R & D Center.

ⓒ 2012 Information Processing Society of Japan

るため，一部のサブシステムやあるいは設備管理サーバに障害が発生しても，正常に稼動する各サブシステムの機能を阻害しない．

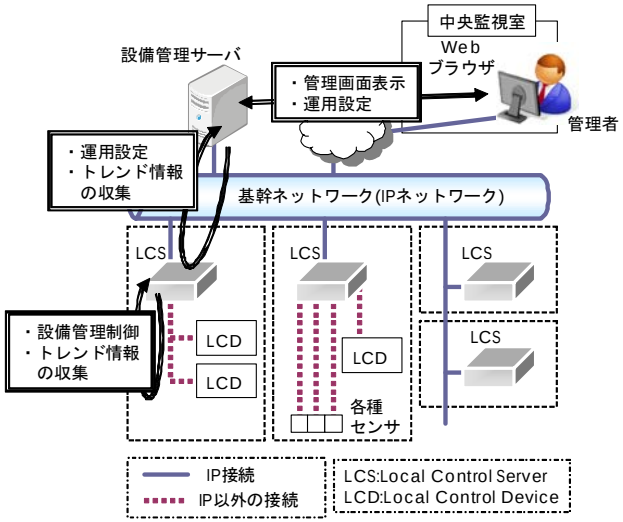


図 1　分散制御システムの構成例
Figure 1　The example of composition of a distributed control system.

一方, DACS は分散制御システムではあるが集中制御システムの特性も有している．設備管理サーバは LCS への逐次制御を行わないが，システム全体の運用設定を把握する必要があり，また場合によっては管理スケジュールに従い LCS への設定を適宜変更するなど，逐次制御に近い機能が必要となることもある．また管理者にシステム全体の状態を提示するためには，システム内のトレンド情報（センサの計測値や各種設備の動作信号など）を収集する必要がある．

通常，各 LCS はサブシステム内部のトレンド情報を数日分保持しており，設備管理サーバは一定周期で各 LCS からトレンド情報を収集する．設備管理サーバは収集したトレンド情報を数年程度保持する．なお設備管理サーバがトレンド情報を収集する周期が短いほど DACS の運用効率が向上するが，弊害として保存するトレンド情報量が膨大となる．

2.2　大規模構成の検討

近年の BA システムではきめ細やかな情報を収集するために各種センサなど管理設備点数が増加しており，加えて，省エネ効率を高めるためにトレンド情報の収集周期を数分単位と短く設定することが求められる．また FA システムにおいても，ライン状態の詳細な把握や制御効率の向上が求められている．この結果，近年の DACS は急速に大規模化が進んでいる．DACS の大規模化に対応する一案として，中長期的には設備管理サーバの高性能化も考えられる．しかし再設計には一定の開発工数を要するうえに，再設計中にも大規模化が進むことから，再設計後の設備管理サーバであっても大規模化に対応しきれない可能性もある．従い，ボトルネックとなる設備管理サーバの処理を複数のサーバに分散して，サーバの台数を増やすことで大規模化に対応するスケールアウトな方式が求められる．また設備管理サーバはシステム全体を管理することの他に，管理者に対してシステム全体の管理画面を表示する必要がある．

以上を考慮することで，DACS の大規模化に対応するためのシステム構成を検討した結果を図 2 に示す．LCS 単位のサブシステムは互いに独立しているため，いくつかのサブシステムを束ねたシステムを複数作成しても, 互いのシステムは独立している．各々の設備管理サーバは従来同様にサブシステムを管理するが，各設備管理サーバを管理する統合管理サーバを設置することで，管理者が全体を俯瞰できることを可能とする．即ち従来は設備管理サーバが管理者に提供していた管理画面表示機能を，統合管理サーバが請け負う．

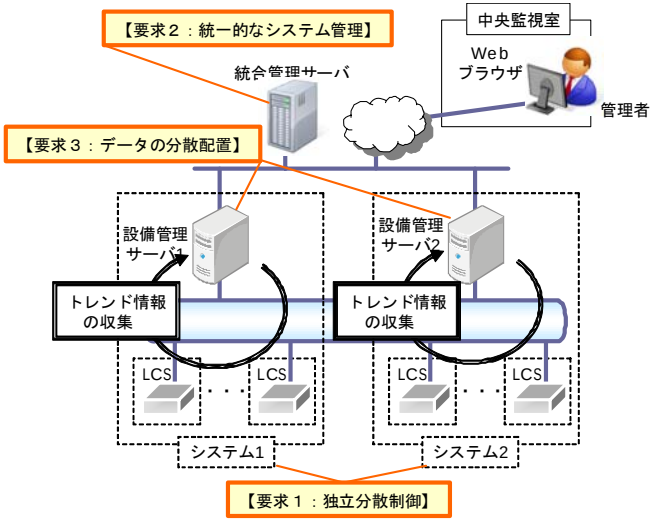


図 2　DACS の大規模化
Figure 2　The large-scale DACS.

他方，分散制御の観点から，管理者が各設備管理サーバにアクセスした際は，その

ⓒ 2012 Information Processing Society of Japan

設備管理サーバが管理する範囲においてシステム管理画面が表示できるものとする．従い管理者は統合管理サーバ，設備管理サーバのいずれでも運用管理できるものとする．この際の注意点として，各設備管理サーバ間では運用設定を共有する必要はないが，統合管理サーバと各設備管理サーバは運用設定を共有する必要がある．

統合管理サーバは管理者の要求に応じてシステム全体のトレンド情報を提示できる必要があるものの，統合管理サーバにてトレンド情報を一元的に集約してはストレージ容量がひっ迫する上に収集性能も要求されてしまい，スケールアウトの観点に反する．そのためトレンド情報の収集は各設備管理サーバが従来通りに実施し，統合管理サーバは各設備管理サーバに分散保持されたトレンド情報を頼りに，システム全体のトレンド情報を管理者に提示する．

2.3 満たすべき要求

図 2 の構成案が満たすべき要求は以下である．

【要求１：独立分散制御】
各設備管理サーバは独立して動作すること．

【要求２：統一的なシステム管理】
統合管理サーバからは全ての設備管理サーバが有するトレンド情報，運用設定情報が統一的に見えること．

【要求３：データの分散配置】
統合管理サーバはデータを一元収集せず，各設備管理サーバが独立してデータを保持すること．

3. 大規模化方式の検討

オープンな技術の活用により DACS の大規模化を達成するための検討を実施した．

3.1 データフローの特性

近年の分散処理技術は Web 系システムの分野にて著しい発展を遂げた．そこで，Web 系システム技術の巧緻な活用方法を検討すべく，Web 系システムと大規模 DACS のデータフローを比較すると図 3 の如くである．

Web 系システムは多数のクライアント端末からのリクエストがデータセンタに殺到するデータフローであり，データセンタは大量のリクエストを処理するために多数のサーバにて分散処理を行う．クライアント側からはデータセンタは単一のサーバに見えるため，予め各サーバにデータを分散配置した上で，データ位置を記憶する名前空間を管理するサーバが，クライアントのリクエストを集約した上で各サーバにて分散処理する方式が一般的である．

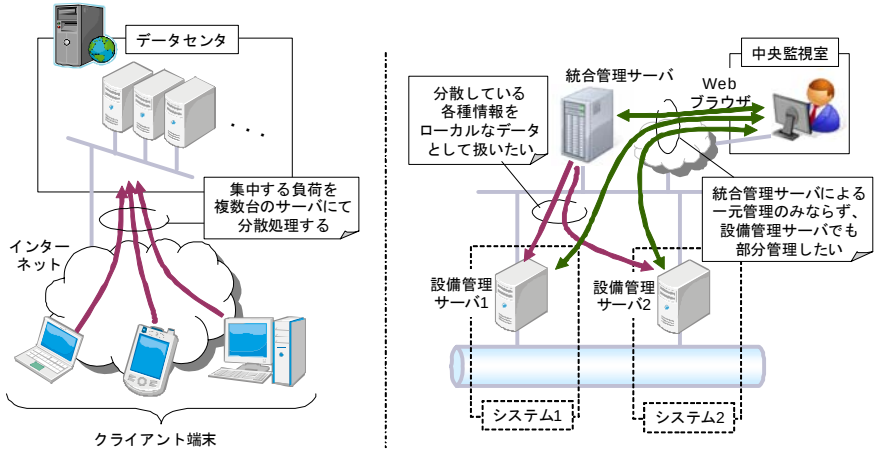


図 3 Web 系システムと大規模 DACS のデータフロー比較
Figure 3 Comparison about the data flow of a Web system and large-scale DACS.

Web 系システムと比較した大規模 DACS の特徴としては，各設備管理サーバは自身が管理するシステムの範囲において独立してトレンド情報を蓄積することが挙げられる．これは統合管理サーバに処理が殺到することを防ぐために有効だが，他方，名前空間を作ることが出来ない．統合管理サーバが名前空間を作成するためには，全てのトレンド情報を統合管理サーバ経由で各設備管理サーバに配置する必要が生じるが，これは満たすべき要求に反する．なお統合管理サーバは設備管理情報を把握するため，名前空間が無くとも必要なデータの位置を特定することは可能である．

以上より，大規模 DACS を実現するためには，独立して動作するサーバ間でデータを共有でき，かつ名前空間を持たずにデータ共有を実現する仕組みであることが求められる．

3.2 オープン系技術の比較

分散制御システムに関する多くのオープン系技術が存在するが，特徴的な技術について比較検討を実施した．

（１）HDFS(Hadoop Distributed File System)

HDFS は Apache プロジェクトの１つである Hadoop の分散ファイルシステムである．データファイルは固定長ブロックを単位として複数のデータノードに配置される．

ⓒ 2012 Information Processing Society of Japan

HDFS は Hadoop の普及により広く用いられている技術であるが，図 4 に示す如く NameNode が名前空間を管理する方式であるため，2.3 節に述べた要求を満足できない．

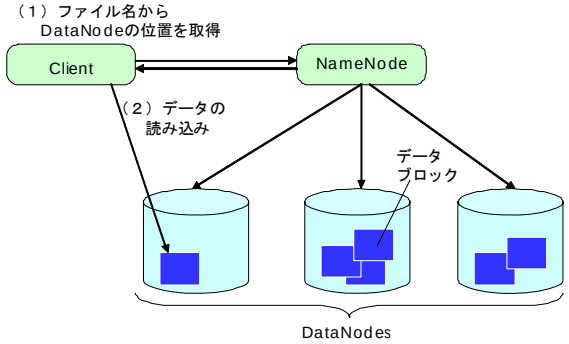


図 4　HDFS の動作
Figure 4　Operation of HDFS.

（2）Gfarm

産総研にて開発が進められている広域分散環境向けのファイルシステムである．認証関係の機能が豊富である一方，耐障害性のサポート機能は乏しい．特に，複製を作成した元のファイルに対して書込みが発生すると複製先のファイルが削除されてしまうため，常時更新されるファイルの冗長性と一貫性を維持することに適さない．図 2 に示した大規模 DACS の構成では各設備管理サーバがトレンド情報を管理することから，Gfarm ではトレンド情報の管理方法が煩雑となる．

（3）GlusterFS

Gluster, Inc.[1] により開発されている GPL ソフトウェアであり，HDFS の NameNode のような名前空間を持たずにデータ共有が実現できることを特徴とする．また個々のストレージ間は互いに通信を行わない方式であるため，ストレージ間の独立性が高い．こうした GlusterFS の特徴は図 2 の構成と親和性が高い．ゆえに本論文の以降では GlusterFS に基づく大規模 DACS の評価環境を構築し，大規模 DACS を巧緻に構築するための GlusterFS の評価を実施した．

4. 評価環境の構築

4.1 評価環境

図 2 に示した大規模 DACS が GlusteFS を元に構築できるかを評価するため，図 5 に示す評価環境を構築した．評価環境は Widows XP(SP3)搭載の PC(CPU：Core2Duo 2.4GHz，Memory：DDR2 2.0GByte)上に VirtualBox[2] にて仮想環境を構築することで実現した．仮想環境上の全ての Host のスペックは表 4-1 の通りである．

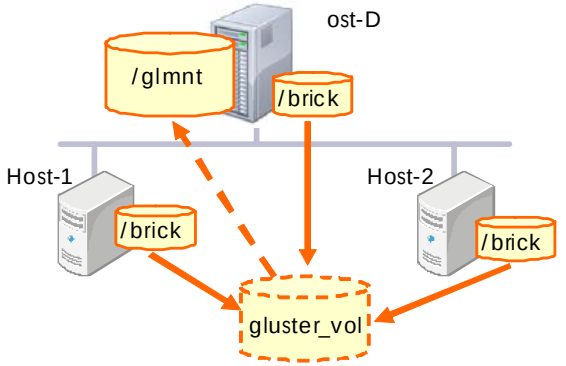


図 5　GlusterFS による評価環境の構成
Figure 5　Composition of evaluation environment by GlusterFS.

表 4-1：評価環境の条件

	スペック（Host-D/Host-1/Host-2）
Memory	384MB
OS	Debian Lenny 5.0.4(Kernel 2.6.26-2)
ストレージ	SATA 15GB
GlusterFS バージョン	3.2.4

Host-D，Host-1，Host-2 の 3 台は，各々，統合管理サーバと設備管理サーバ 1 および 2 に相当する．各 Host では GlusterFS にて共有するディレクトリとして"/brick"を作成し，全ての"/brick"を統合した仮想ストレージ"gluster_vol"を作成する．GlusterFS の機構としては，いずれの Host においても"gluster_vol"をマウントすることが可能だが，ここでは図 2 の構成に従い Host-D のみが"gluster_vol"を"/glmnt"ディレクトリとしてマウントする．

評価の視点は以下の 2 点である．

（1）2.3 節のシステム要求に対する評価
（2）GlusterFS の挙動に関する詳細評価

ⓒ 2012 Information Processing Society of Japan

4.2 評価結果

4.2.1 システム要求に対する評価

各 Host の"/brick"にファイルを配置した際の動作を図 6 に示す．Host-D は各 Host が"/brick"以下に持つファイルの全てについて，ファイルの実体を収集すること無しに"/glmnt"からアクセス可能である．アクセスする際，名前空間は特に用いない．また各 Host の"/brick"は互いに独立しているため，いずれかの Host において"/brick"内にファイルが生成・削除されても他の Host の"/brick"は影響を受けない．

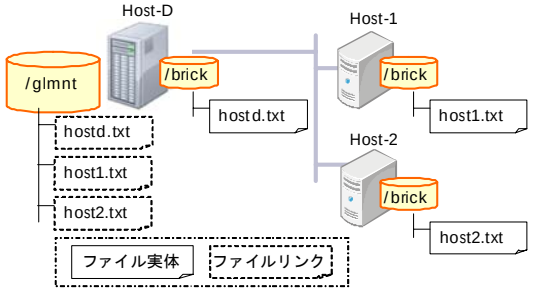


図 6　ファイル共有動作
Figure 6　Share operation of files.

"/brick"以下にディレクトリを生成する際は注意が必要である．図 7 に示す如く，全ての"/brick"に同名のディレクトリ（図 7 の"common/"）が作成されることで"/glmnt"からアクセス可能となる．ただしディレクトリ内のファイルについては，図 6 と同様に全ての Host のファイルが"/glmnt"から透過的にアクセスできる．

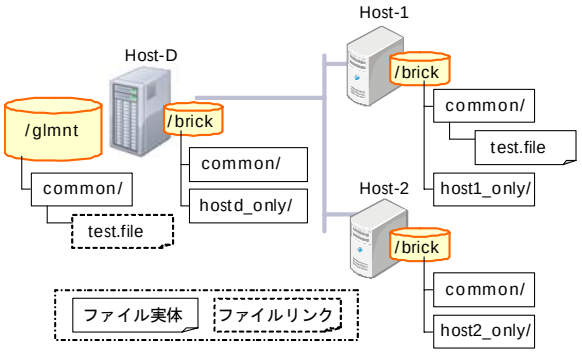


図 7　ディレクトリの共有動作
Figure 7　Share operation of directories.

"/glmnt"内でファイルを作成した場合は図 8 の如く，その実体はいずれかの Host の"/brick"に格納される．この際の格納方法は GlusterFS の設定によって，ストライプ配置や冗長配置も可能である．しかしいずれにせよ，Host を特定して"/brick"内にファイルを配置するためには，GlusterFS とは別の機構を用いることが好ましい．一例として，図 2 の構成では統合管理サーバでも各設備管理サーバでも運用設定が可能であるため，運用設定ファイルを"/brick"に配置することで設備管理サーバと統合管理サーバ間で設定内容を共有する状況が考えられる．そして統合管理サーバが運用設定を設備管理サーバに渡す場合，GlusterFS とは別の機構を用いて実現すべきであろう．

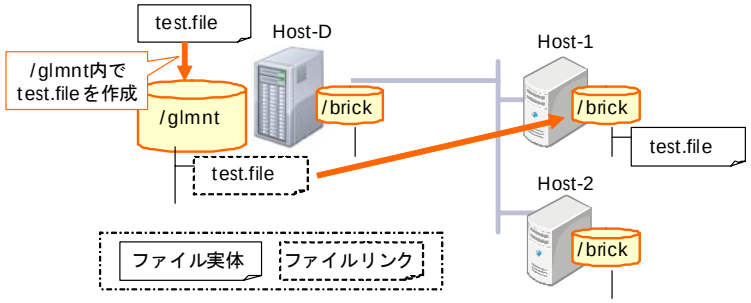


図 8　ファイル作成動作
Figure 8　Operation about file creation.

4.2.2 挙動に関する評価

GlusterFS の挙動の詳細を調査するため，図 5 の評価環境において，以下の（1）（2）に関するファイル I/O の動作を調査した．

（1）大容量ファイルの読出し動作

図 5 において，Host-1 の"/brick"に 200MB のデータファイルを配置した後，Host-D の"/glmnt"からこのファイルを読み出して/dev/null に廃棄する．その際，Linux の sar コマンドにより，CPU 占有率，ネットワーク通信量，ディスク読出し処理回数(read transaction per second : rtps)を計測した．この結果を図 9 に示す．なお CPU 占有率は常時 1%未満であったため割愛した．

Host-D が読出しに要した時間は 33 秒で，通信が終了するまでディスク読出しが殆ど発生しないことがわかる．Host-1 で同様の処理を実施すると，通信が発生しない以外は凡そ同様の動作を示し，30 秒で処理を終えた．

以上のことから，ネットワークの通信帯域がボトルネックとならない環境においては，大容量ファイルのアクセスはローカルデータへのアクセスと遜色無い性能でアクセス可能であると判る．

ⓒ 2012 Information Processing Society of Japan

（2）多量のファイルの読出し動作

Host-2 の"/brick"に 256Byte のデータファイルを 10000 個作成して Host-D から読出し，/dev/null に廃棄する．その際，上記（1）と同様の計測を実施した．この結果を図 10 に示す．この際，CPU 占有率は常時 1%未満であり，またディスク読出し処理回数もほとんど無いため，図 10 にはネットワーク通信量のみを示した．

同様の処理を Host-2 にて実行すると 1 秒未満で処理を終了した．対して Host-D では図 10 の如く 73 秒を要している．受信(RX)，送信(TX)の動作から，Host-2 へのデータの要求とその応答が頻繁に実行されていると判る．

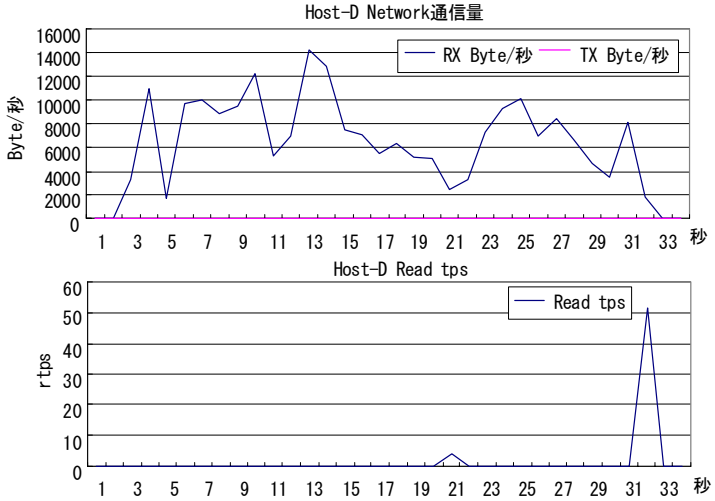


図 9　200MB のデータファイル読出し動作(Host-D)
Figure 9　200 MB of data file read-out operation.

4.3 評価結果の集積

4.2 節の評価結果を纏めると以下である．

- 2.3 節のシステム要求を満足する大規模 DACS を構築することが可能である．
- ディレクトリを共有する際には，別の Host が"/brick"内に作成したディレクトリ名を，全ての Host が何らかの方法により知る必要がある．
- 統合管理サーバが運用設定内容を設備管理サーバに渡す場合には，GlusterFS とは別の方法で渡す必要がある．
- ファイルサイズが大きいほど読出し効率が良い．逆に，小さなファイルを大量に読み出すことは不得手である．統合管理サーバから各設備管理サーバのトレンド情報を読み出す場合には，この特性に注意する必要がある．

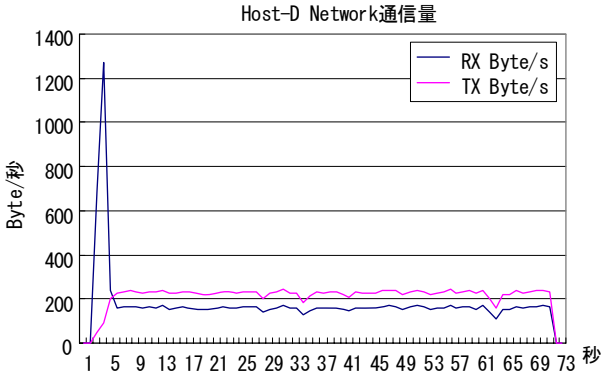


図 10　256Byte データ 10000 個の読出し動作（Host-D）
Figure 10　Operation which reads 10000 files data of 256Byte.

5. まとめ

本論文では，DACS のデータフローに着目することにより，DACS の大規模化に求められる要求を 2.3 節にて明らかにした．大規模 DACS では名前空間を必要とする分散処理方式の適用は困難である．分散制御システムを構築するためのオープン技術の 1 つである GlusterFS は名前空間が不要であることから，大規模 DACS の構築に適する技術であると判った．

4 章の評価結果より，GlusterFS は大規模 DACS に求められる要求を満足することが確認出来た．しかし実際に大規模 DACS を構築する際には，GlusterFS を基としながらも，4.3 節の評価結果を考慮した付帯的な機能の実現を図る必要がある．

また大規模 DACS の運用ではスケールアウトの容易性も重要な話題であるため，この評価は今後の課題である．

参考文献

[1]　Gluster Community
< http://www.gluster.org/ >（参照 2012-02-27）.

[2]　Oracle VM VirtualBox
< https://www.virtualbox.org/ >（参照 2012-02-27）.

ⓒ 2012 Information Processing Society of Japan