大規模システム管理のための自律分散モニタリングシステム

林原尚浩
東京電機大学 理工学部 情報システム工学科

概要

企業や組織内のシステムは，近年，ネットワーク上に多数の計算機などのリソースを有し，大規模かつ複雑なシステムを構成している．このようなシステム上で動作するサービスを停止することなく提供するためには，システム全体を常時監視する必要があり，これを人手で行っている管理者の負担は非常に大きい．
本研究では，大規模分散システムの管理を行うためのモニタリングシステムを開発する．これは，システム全体のリソースを網羅的に監視し，管理者に必要な情報を提供することを目的としている．このモニタリングシステムの初期設定やシステムの構成の変更に自律的に行うことによって，管理者の負担を軽減する．

1 はじめに

コンピュータシステムは，社会的な基盤として認知されており，様々なサービスを提供している．サービスを提供するシステムは，ネットワークインフラの充実や汎用計算機の小型化，低価格化によって，大規模化，複雑化している．このような大規模なシステムを運用するためには，常時システムの管理が必要となってくるが，多くのシステムにおいて，少人数の管理者が人手でシステムの管理を行っている．また，提供しているサービスによっては，システムが故障し，サービスが提供できなくなることによって，多大な被害を被るものもあり，システム管理者の負担は増大している．

大規模システムの管理を行うための補助的なツールとして，いくつかのシステム監視ツール [2-5] が提案，実装されている．これらは，予め対象ノードを設定ファイルに記述し，そのノードやサービスの死活判定を行う．

しかし，ノードやサービスを監視する際，クライアント・サーバ型のインタラクションを行うため，数千台規模を超える企業，大学，官庁などのネットワークリソースを網羅的に監視するためには，これらのシステム監視ツールは不向きであるといえる．また，監視対象の追加が頻発したり，ネットワークの構成が変化した場合，新たに設定ファイルを書き直す必要があり，この点では，システム管理者の負担を強いることになる．

また，サービスの監視を行うことは，サービスを継続的に提供するという目的では有効であるが，近年，組織内の計算機による情報漏洩などに代表される，意図しないサービスの提供や禁じられたサービスの不正利用によるシステムの脆弱性を監視するためには効果がない．

我々は，以上の点を踏まえた大規模システム管理のための自律分散型モニタリングシステムを開発する．

2 関連研究

システム監視ツールは，商用ソフトウェア，オープンソースソフトウェア共に数多く存在する．Nagios [2] や Big Brother [4]，Hobbit [5] は，いずれも web ベースのユーザインターフェイスを持っており，監視対象ノードやその上で動作しているサービスなどの状態をグラフなどで視覚的に表示することができる．これらは，監視サーバに監視対象ノードやサービスを記述しておき，それを元に監視対象ノードへ ICMP や SNMP を用いた問い合わせを行い，監視対象の情報を収集する．

MPG [3] は，ping の結果をグラフで表示するシンプルなツールで，タイムアウトを設定し，Round-trip time がそれを超えると，グラフの背景を変え注意を喚起する．

これらのソフトウェアは，クライアント・サーバ型のシステム監視ツールであり，監視対象ネットワークの全ネットワークリソースを網羅的に監視する場合には，スケーラビリティの問題がある．Nagios は，分散監視を行うための拡張も可能だが，単純に監視サーバを増やすため，スケーラビリティは向上するが，設定ファイルが分散し，管理がより困難になる．

3 モニタリングシステムの設計

本研究では，次の項目を満たすモニタリングシステムを提案する: (i) 自己組織的なモニタリングシステム: システム管理者が膨大な数の監視対象を設定ファイルに書き込んだりすることなく，最適なモニタリングシステムを自律的に構成できる，(ii) Accrual インターフェイス: 死活判定を従来のタイムアウトのみではなく，故障している度合い (suspicion level) を用いて表すことで，より直感的に監視対象の状態を把握することができる，(iii) 内部的な脆弱性の検索: 意図しないサービスの提供や禁

止されたサービスの利用の検索を行い，システムの内部的な脆弱性を排除する．

我々が提案するモニタリングシステムは，システム管理者が操作する端末である監視サーバと実際にネットワークリソースを監視する故障検出器という2種類のプロセスからなる．監視サーバは，故障検出器の登録と初期的な監視先の設定を行う．故障検出器は，自分のサブネット内の全てのノードを監視すると同時に，他のサブネットに配置されている故障検出器を監視する．

故障検出器同士は相互に監視を行っており，その「相互に監視する」という関係は一種のオーバレイネットワークと捉えることができる．このオーバレイネットワークを本稿では監視ネットワークと呼ぶ．システム管理者によって各故障検出器の設定が変更される場合，その情報は，故障検出器の監視ネットワークを用いて全ての故障検出器に伝搬する．

各故障検出器は同一サブネット内のネットワークリソースの監視も行う，これは，故障検出器同士の監視の様な相互的な監視ではなく，一方的なものである．それぞれの故障検出器やネットワークリソースの監視には，既存のシステム監視ツールと同様ICMPを用いる．

3.1 監視ネットワークの自己組織化

モニタリングシステムの初期的な段階においては，監視サーバのみが存在する．その後，対象のサブネットのノードで故障検出器を起動すると，監視サーバへアクセスし，自身を登録する．この時，監視サーバからは，監視対象となる他の故障検出器に関する情報（IPアドレス，ポート番号など）が得られる．この情報を元に，故障検出器同士で自己組織的に監視ネットワークを構築する．

監視ネットワークは，故障検出器が相互に監視を行う際にできるオーバレイネットワークであり，このネットワークを用いて，システム管理者からの設定変更などの情報が故障検出器へ伝搬される．一つの故障監視器が他の故障監視器を監視する数dは，スケーラビリティの観点から重要である．新規の故障検出器が追加された場合やシステム管理者がdの値を変更した場合，故障監視器は，このdを一定に保つように，自律的に監視ネットワークを再構成する．このために，オーバレイネットワークのノードの次数dを一定に保つ自己安定アルゴリズムが必要となる．Dattaらが提案した自己安定アルゴリズム [1] は無向リンクを対象としているが，監視ネットワークは有向リンクで構成されるので，このアルゴリズムを拡張した自己安定アルゴリズムを開発する．

3.2 Accrualインターフェイス

故障検出器は，設定されたタイムアウトを元に監視対象の故障検出器やサブネット内のノードの死活判定を行い，状態の変化があった場合には，監視サーバへ通知する．また，ICMP ECHO/REPLYによって得られるRound-trip timeをサンプリングし，それを元に，監視対象の現在の状態をsuspicion level(故障している度合い)として連続的な値で表示する．これによって，監視対象の故障の前兆などを予測することができる．また，suspicion levelに対する閾値を設けることによって，suspicion levelを用いた死活判定も可能である．

一般的に，タイムアウトの設定はネットワークの状態や環境によって最適な値は異なるため，監視対象のネットワークに精通している管理者でなくては設定を行うことは困難である．しかし，suspicion levelは故障している度合いを示すものであるため，管理者の経験やネットワーク環境に依存することなく設定することができる．このことから，タイムアウト(時間軸)，suspicion level(故障判定の信頼性)の両方で死活判定を行うことで，より柔軟なモニタリングシステムを構築することができる．

3.3 内部的な脆弱性の検索

故障検出器は，同一サブネット内のネットワークリソースの死活判定だけではなく，定期的にネットワークリソースのスキャンを行って，新しいリソースが加入していないか，また，既存のリソースが不正なサービスを行っていないかなどの検査を行う．これによって，近年頻繁に起こっている，システムの内部的な情報の漏洩を未然に防止するために効果があると考える．

4 まとめ

本稿では，大規模分散システムの網羅的監視を行うためのモニタリングシステムについて，その構造的と機能の概要について述べた．このシステムは主に，システム管理者が従来行っていた初期設定作業の半自動化，新規ノードなどの追加による設定ファイルの変更の自動化などを行い，管理者の負担を軽減する．また，故障検出器によるオーバレイネットワークを構築し，このネットワークを自己組織的なアルゴリズムによって構成することで，ネットワーク環境の変化に柔軟に対応することができる．今後，学術振興会 日仏交流促進事業 (Sakura program) と東京電機大学総合研究所研究 02Q631 の補助の元，自己組織的監視ネットワーク構成アルゴリズムの開発やモニタリングシステムの実装を行っていく予定である．

References

[1] A. K. Datta, M. Gradinariu, and A. Virgillito. Deterministic δ-connected overlay for peer-to-peer networks. In *the Ninth IEEE International Symposium on Object and Component- Oriented Real-Time Distributed Computing (ISORC'06)*, pages 159 – 168, 2006.

[2] E. Galstad. Nagios. `http://www.nagios.org/`.

[3] C. Schmidt. Multiping grapher. `http://software.ccschmidt.de/`.

[4] Q. Software. Big brother. `http://www.bb4.com`.

[5] H. Stoemer. Hobbit. `http://www.hobbitmon.sourceforge.net/`.