

# 広域分散環境における「環視」システムの構築 Construction of "Monitor Ring" system in distributed wide-area environment

釘井 睦和<sup>†</sup>

廣安 知之<sup>‡</sup>

三木 光範<sup>‡</sup>

谷口 義樹<sup>†</sup>

Yoshikazu Kugii

Tomoyuki Hiroyasu

Mitsunori Miki

Yoshiki Taniguchi

## 1. はじめに

近年、高速な計算機やネットワークの普及に伴い、広域ネットワークに接続されたあらゆる計算資源および情報資源を統合し、ユーザが手軽に利用できるような環境を目指すグリッド技術が注目されている [1]。

グリッドのような技術によって、ユーザはどのサイトを利用するかを一切考慮せずにそれらが提供するサービスを享受することが可能になると期待されている。しかしながら、グリッド上のサイトは広域に分散しており、ユーザのタスクは通常複数のサイトをまたがって処理されることから、一部のサイトにシステムダウン等の問題があった場合にタスクの処理が中断される可能性がある。そのため、サイトの管理者は不特定多数のユーザに対する責任を負うことになり、管理コストは増大する。すでに計算資源のモニタリングツールはいくつか存在するが、これら主にサイト管理者が使用するためのものであり、管理体制の良い常時モニタリングを行うには管理者に大きな負担がかかる。

そこで、本研究ではサイトを利用するユーザ自身が適宜モニタリングすることを考える。多数のユーザが協調的にモニタリングを行えば、サイトの管理体制が向上すると考えられる。本研究ではユーザが常時サイトの状況をチェックできるような「環視」システムを提案し、その特徴や詳細について述べる。そして、PC用、PDA用、および携帯端末用のソフトウェアについて紹介する。

## 2. 広域分散環境でのモニタリング

広域分散環境下では、計算資源の状態が動的に変化する。このような環境では資源を有効に活用するために、サイトに発生する障害からの迅速な復旧を行う必要があると考えられる。そのためには、物理的に分散した計算機を利用する環境において、アーキテクチャ、CPU、メモリ、ディスク、ジョブの状況、ミドルウェアやアプリケーションの動作状況を含む様々な情報を収集、管理および提供を行うシステムが必要不可欠となる。こうしたモニタリング情報は、ジョブスケジューラの判断材料となるだけでなく、広域分散環境のユーザや管理者、サポートスタッフにとってはデバッグや障害検知に有益な情報源になる。

モニタリング情報の収集は、一般に全ての計算ノード上で行う必要があるが、その情報の管理、提供は全てのノードで行う必要はない。また、モニタリング情報を利用するアプリケーションは、基本的に情報収集を行うノード、管理・提供を行うノードとは独立に任意のノード上で実行され得る。また、広域分散環境では一つのノード

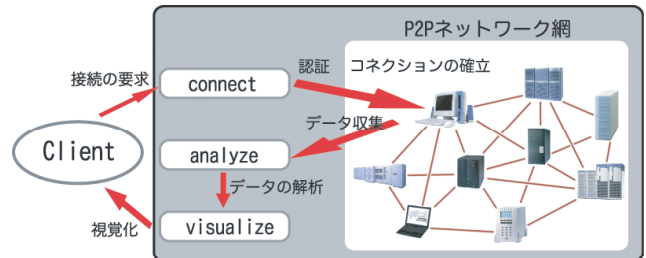


図 1: 提案システムの構成

が中央集権的にモニタリング情報を管理・提供することはほとんど不可能である。

## 3. 「環視」システム

既存のモニタリングシステムは、一つのノードが中央集権的にモニタリング情報を管理・提供することを想定して実装されているため、グリッドのように計算機ノードが広域に分散しているような環境ではアプリケーションを利用することが困難である。このような問題点を解決するために、本研究では「環視（協調的モニタリング）」システムを提案する。「環視」とは、サイト管理者のみならずサイト利用者などが無意識のうちにサイトを監視することを意味する。以下に提案システムについて述べる。

### 3.1 概要

提案システムは、DNAS (Distributed Network Application System) [2] とモニタリングクライアントソフトウェアから構成される。図 1 にシステムの構成を示す。

DNAS は、複数のノード（サイト）を再構成可能なツリー構造として扱い、ロードバランシングや効率的な情報の送受信を提供するための P2P フレームワークである。ツリー構造を用いることにより、通信の相手を階層によって明示的に制限することが可能であるため、ピア P2P モデルの問題点であるネットワーク全体への負荷を軽減することができる。また、P2P 技術を利用することにより、近接したサイトの情報収集や異常の発見を中央サーバに依存することなく行うことができる。端末クライアントは DNAS から得られる情報を解析し、視覚化してユーザに提供する。図 2 のように、従来のモニタリングシステムでは、ノードの情報を収集するためのメタサーバが存在し、メタサーバの情報を中央集権的に管理するサーバを設け、このサーバにクライアントがアクセスし情報を得るのが一般的である。しかし、情報を一括管理するサーバがシステムダウンしてしまうとクライアントは情報を取得することができないという問題点が存在する。また、サーバの負荷等を考慮するとスケラビリティにも限界が存在する。グリッド環境では、ど

<sup>†</sup>同志社大学大学院工学研究科知識工学専攻

<sup>‡</sup>同志社大学工学部知識工学科

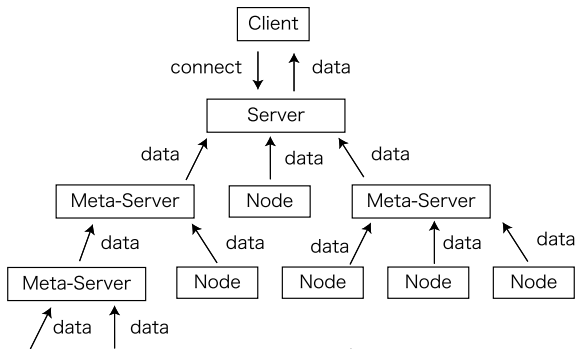


図 2: 従来のモニタリングシステムの構造

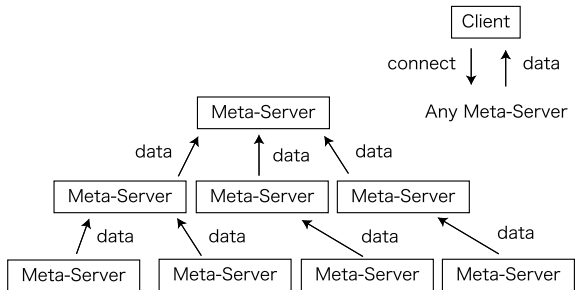


図 3: 提案システムの構造

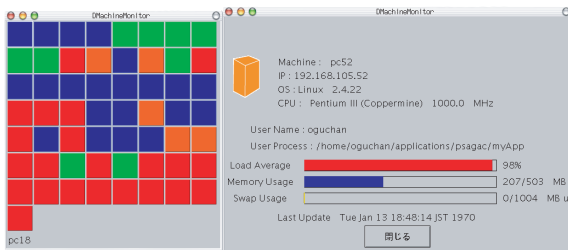


図 4: PC, PDA 用ソフトウェア

の計算ノードからでも情報が取得できること、無限のスケラビリティを有することが望まれる。図 3 に示す提案システムの構成では、ノードは全てメタサーバとして扱われ、クライアントはどのノードからでも情報が取得可能であり、DNAS を用いた P2P 通信により理論上無限のスケラビリティを有することが特徴である。

### 3.2 モニタリングクライアントソフトウェアの実装

モニタリングを行う端末クライアントの実装の 1 つとして、PC, PDA 共に動作するクライアントソフトウェアを作成した。アプリケーション表示画面を図 4 に示す。まず、接続するホスト名、ポート番号を入力しコネクションを確立させる。接続に成功すればホストに収集されているデータが取得でき、CPU の負荷状況に合わせて色が異なる全体画面が表示される(図 4)。詳細を表示したければマス目の領域をクリックする。ホスト名、プロセスの状況、ロードアベレージ、メモリ使用量などを中心にデータを取得しており、アプリケーションに表示された値と計算ノードでの実際の値が一致していることを確認した。

また、作成した携帯端末クライアントソフトウェアの表示画面を図 5 に示す。携帯端末クライアントには、より多くのユーザが頻繁にソフトウェアを利用できるようにゲームインターフェースを利用したものも同時に作成

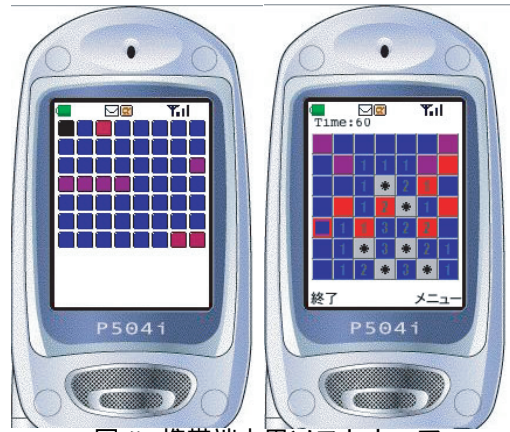


図 5: 携帯端末用ソフトウェア

した。実装したゲームインターフェースはマインスイーパーである。ゲームインターフェースを用いることにより「環視」を行う回数を増やすことが狙いである。

## 4. 実験

作成したソフトウェアの動作、機能確認のため同志社大学知的システムデザイン研究室が管理する複数の PC クラスタを用いた実験環境を構築した。広域分散環境に近づけるため、WAN と LAN が混在するような環境にした。詳細は以下の通りである。

- Supernova(Opteron 1.8GHz × 2PE × 256 ノード) より 30 ノード
- Xenia(Xeon 2.4GHz × 2PE × 64 ノード) より 20 ノード
- Gregor(PentiumIII 1.0GHz × 2PE × 64 ノード) より 60 ノード

実装したシステムが正常に動作することを確認し、モニタリングソフトウェアにより異常ノードの早期発見を行うことができた。

## 5. まとめと今後の課題

本稿では、モニタリングシステムの必要性を述べ、提案システムと既存のモニタリングソフトウェアとの比較を行い、その利点などについても述べた。テスト環境を構築し、システムの有用性も確認した。今後は提案システムを大規模なグリッドテストベッドのような大規模な広域分散環境で運用し、クライアントソフトウェアを複数の既存モニタリングアプリケーションに対応させる予定である。

## 参考文献

- [1] Foster I. and Kesselman C. The grid: Blueprint for a new computing infrastructure. *Morgan Kaufmann*, 1998.
- [2] Junichi Uekawa, Tomoyuki Hiroyasu, Mitsunori Miki, and Yusuke Tanimura. A dynamic hierarchical system for large scale distributed applications. *Proceedings of the 14th IASTED International Conference, Parallel and Distributed Computing and Systems*, pp. 422-427, 2002.