

クラスタモニタリングシステム

松下 知明

1 はじめに

近年, 科学技術計算やシミュレーションなどにおいて, 膨大な計算をすることが多くなり, 大規模な計算資源が必要とされている. そこで, 高い演算性能を持つクラスタが注目を集めている. クラスタは複数のコンピュータを 1 台の仮想コンピュータとして用いることにより大規模な計算が可能である. しかし, ユーザの使用中にシステムダウンなどの障害が発生するとタスクが中断してしまう可能性が出てくる. その状況を改善するため, 管理者が定期的にクラスタを監視する必要がある. このとき, 複数あるコンピュータに 1 台ずつアクセスしているのでは時間がかかる. そこで, 自動的に情報を集めて表示するシステムが必要になり, また視覚的に収集したデータをグラフ化することも必要である.

2 クラスタ

2.1 クラスタとは

クラスタは, 複数のコンピュータを相互に接続し, 複数のコンピュータを 1 台のコンピュータのように振舞わせる技術である¹⁾. 以下にクラスタのメリットを示す.

- スーパーコンピュータに並ぶ高い演算性能
- 汎用のコンピュータでできるため構築費用が安価
- 1 台が故障してもシステムの稼働が可能

2.2 問題点

クラスタは, 上記で説明したように多くのメリットを持っているが, 一部のノードに負荷が集中すると, 故障の原因, 処理速度の低下を招く. またコンピュータ資源の不足や全てのノードが平均的に使われているかなどの問題がある. これを改善するために管理者が定期的に監視しやすい環境を作らなければならない. そこで, クラスタの使用状況を視覚的に把握しやすいクラスタモニタリングシステムを作成することにした.

3 クラスタモニタリングシステム

3.1 システムの概要

クラスタモニタリングシステムは, クラスタの使用率を管理者側に表示させるシステムである. 管理者はクラスタにコマンドを送り, クラスタ内にあるコンピュータの情報を一箇所に集めて管理者に返す. 管理者側は, その使用率のデータを用いて表示する. しかし, ただデー

タを表示するだけでは, 多くの情報から必要な情報を探す手間があるので, 必要な情報だけを抽出して表示する. さらに, これまでの使用率を見ることが出来る履歴機能も備える.

3.2 システムの構成

本システムは, ディスプレイ, コレクタ, モニタの 3 つの部分で構成されている. Fig. 1 のように情報を集めるコレクタは, クラスタのマスタノードで動作させる. そこに情報を送るモニタはスレーブノードで動作させ, 管理者はディスプレイを用いてモニタリングを行う. データの送受信には, ソケット通信を用いる²⁾.

それぞれのシステムでの主な動作を以下に示す.

- ディスプレイ
 - ソケットを用意し, 接続要求を行う
 - 入力されたコマンドをコレクタに送信する
 - コレクタからのデータを受信し, 必要なデータを抽出する
 - グラフ化し表示する
- コレクタ
 - ソケットを用意し待機する
 - モニタからのコマンドを受け取り, ソケットを用意する
 - コマンドをさらにモニタに送信する
 - モニタからデータを受信し, ディスプレイに返す
- モニタ
 - ソケットを用意し待機する
 - コレクタからコマンドを受け取り, 実行する
 - 出力されたデータをコレクタに送信する

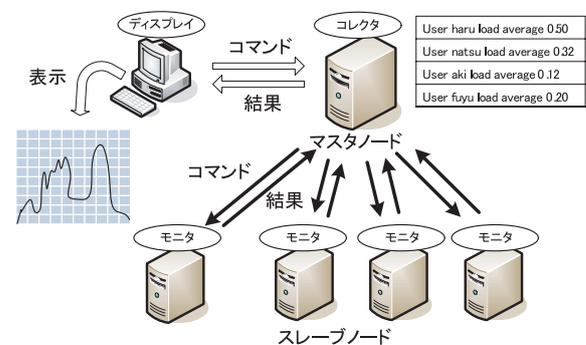


Fig. 1 システム構成 (出典: 自作)

3.3 システム環境

Table 1 に構築に利用したシステム環境を示す。

Table 1 システム環境

管理者側		
CPU	Intel Pentium4	3.00GHz
Memory		1GB
HDD		160GB
OS	Microsoft Windows XP Professional	
network		Ethernet
クラスタ側		
nodes		4
CPU	Intel Pentium	800MHz
Memory		256MB
HDD		20GB
OS	Debian GNU/Linux 3.1 (kernel 2.6.8)	
network		Ethernet

3.4 システムの動作

システムの全体的な動作を、Fig. 2 に示す。項目の番号は Fig. 2 の番号と対応している。

1. コレクタとモニタでソケットを生成する
2. ソケット登録後に待ち状態になる
3. ディスプレイでソケットを生成する
4. コレクタに接続要求を開始し、受理されればネットワークが確立される
5. コレクタはソケットを生成して4台のモニタに順番にアクセスする
6. コマンド受信後、モニタはコマンドを実行する
7. 結果をコレクタに送信後、コレクタ - モニタ間の接続を閉じる
8. コレクタは送られてきたデータを順次ディスプレイに送信する
9. ディスプレイは、送られてきたデータをノードごとに分類してファイルに保存する
10. ファイルから必要なデータを抽出する
11. グラフ化し表示する
12. 終了コマンドが入力されるまで5~11を繰り返す

3.5 インタフェース

Fig. 3 のように現在の CPU 使用率をグラフで表示している。そして、履歴の見たいノードをチェックし、履歴確認ボタンをクリックすると、そのノードの CPU 使用率のグラフが表示される。

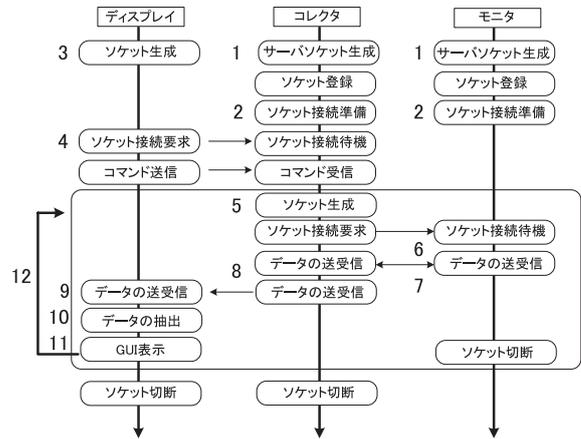


Fig. 2 全体的な流れ (出典：自作)

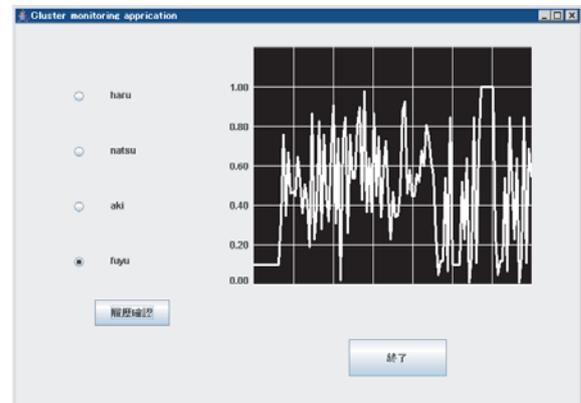


Fig. 3 モニタリング画面 (出典：自作)

4 まとめ

本報告では、ソケット通信を用いてクラスタの各ノードの CPU 使用率をリアルタイムに管理者側で監視でき、履歴も参照できるモニタリングシステムを構築した。これにより、クラスタの一部に負荷が集中したり、ダウンなどの障害が発生した場合に対処できるようになった。今後の課題としては、アクセスしているユーザ情報の表示など、管理を円滑に出来るようにモニタリングシステムを拡張する必要がある。

参考文献

- 1) 多様化するクラスタ方式
<http://www.atmarkit.co.jp/flinux/rensai/クラスタ 01/クラスタ 01.html>
- 2) ソケット
<http://research.nii.ac.jp/~ichiro/syspro98/socket.html>