

N-28

## アクティブネットワークの負荷分散

\*小澤悠一, \*中西崇, \*渡辺友基, \*松浦康治, \*吉川佳寛, \*\*小林幸也

\*沼津高専・電子制御工学科 5年 \*\*沼津高専・電子制御工学科

## 1. 目的

アクティブネットワークの特徴を生かし、アクティブネットワークシミュレータ上で、アクティブノードの負荷分散を行う学生実験システムを開発し、評価、考察する。

## 2. 実験システムの概要

## 2.1 AN(アクティブネットワーク)フレーム

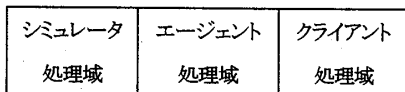


図 1. AN フレーム

アクティブネットワークでは AN フレーム(図 1)が送受信される。シミュレータ処理域には発信アドレス、宛先アドレスが格納される。エージェント処理域には、判別や、コマンド、プログラムなどが格納される。そしてクライアント処理域ではエージェント処理に必要なファイルなどが格納されている。

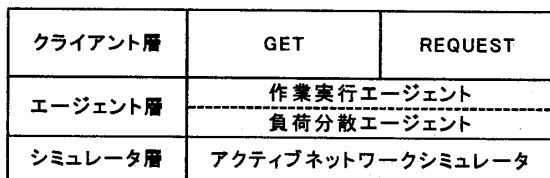


図 2. シミュレータ層、エージェント層、クライアント層

## 2.2 実験システムの特徴

実験システムでは、全ノードにまったく同一の負荷分散エージェントを格納し、動作させる。この種類のエージェントで負荷分散の指示を出したり、この指示にしたがって負荷分散のためのプログラムを実行したりする。したがって、あるノードが負荷分散の指示を出していたとしても、次の処理ではそのノードが負荷分散のためのプログラムを、与えられた指示にしたがって実行するノードになることができる。つまり、この複数のノードからなるシステムでは特定のクライアント、サーバを置かず、状況に応じて各ノードがクライアントになったり、サーバになったりして負荷分散を実現する。これにより、障害に強いシステムを構成することができる。

## 2.3 負荷分散のシミュレーション環境

アクティブノード3台と監視をする PC(監視 node) 1台で構成する。

Load-Balancing for Distributed Management in Active Networks

Y. Ozawa University of Tokuba,

T. Nakanishi Japan Broadcasting Corporation,

Y. Watanabe Toyohashi University of Technology,

Y. Matsuura Japan Telecom CO., LTD,

Y. Yoshikawa Fujitsu Support And Service INC,

K. Kobayashi Numazu College of Techninology \*

表 1. 各アクティブノードの性能

Node	name	CPU	メモリー
(A)	dkk01	Pentium Pro 200MHz	80MB
(B)	dkk02	i486 DX4 100MHz	56MB
(C)	dkk03	Pentium 150MHz	48MB

## 3. 負荷分散の流れ

負荷分散の機能の流れを図3に示す。

- ① Active node(以下 Anode) (A)から、Anode(B)に、作業付きの AN フレームを送信する。
- ② ①で送信した AN フレームを受信した Anode(B)は、Anode(B)自体の CPU 利用率を調査した後、全 Anode に CPU 利用率調査依頼 AN フレームを送信し、CPU 利用率を調べさせるようにする。  
(AN フレームに付いていた作業は(B)がファイルに保存する)
- ③ ②で送信した AN フレームを受信した Anode(A)(C)は、自分の CPU 利用率を調べてファイルに保存し、そのファイルを AN フレームに付けて送信元の Anode(B)へ送信する。
- ④ ③で送信される AN フレームを全 Anode から受信した Anode(B)は、全 Anode(自分も含めて)の CPU 利用率の値を比較し、値が一番低い Anode(例えばこの場合は Anode(C))に②で保存した作業ファイル付きの AN フレームを送信する。
- ⑤ ④で送信された AN フレームを受信した Anode は、作業ファイルを実行エージェントに渡し、実行依頼する。実行エージェントは、その Anode 上で作業を実行する。このとき、実行結果を他の Anode に送らなければならないときは、指定された Anode に実行結果を送信する。

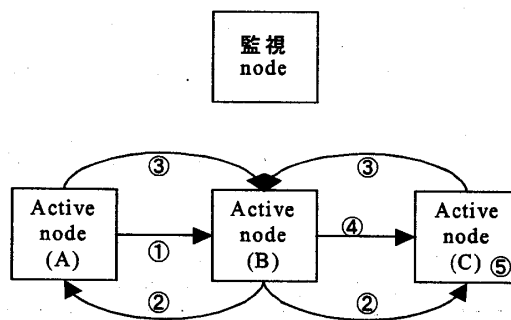


図 3. 負荷分散の機能の流れ

## 4. 学生実験

## 4.1 実験目的

学生が実験者となりアクティブネットワークを用いた実験をする。これにより学生に、アクティブネットワークや、アクティブネットワーク上

での負荷分散について理解させる。

4.2 実験方法

事前に、学生がアクティブネットワークの概要、負荷分散の方法、ネットワークの構成などの説明を受けて、以下に示す負荷分散を行う。

4台のノードを用いて負荷分散実験を行う。以下の(1)~(4)の4つのケースを行い、その後考察をする。

- (1) 全ノードに負荷をかけない場合
- (2) 初めに作業を依頼されるノードに負荷をかけた場合
- (3) 作業を依頼されていないノードのうち、一つに負荷をかけた場合
- (4) 全ノードに負荷をかけた場合

考察はシミュレータでの負荷分散がどのように行われているかをログデータを解析して行う。

4.3 実験により得られたこと

学生による実験自体はスムーズに行われた。アンケートの結果や学生を見ていてわかったことは、次の通りである。

- (1) 負荷分散については良く理解できるが、アクティブネットワークについては難しい。
- (2) スムーズに出来すぎるので、もっと実験や考察を増やした方がよい。
- (3) この実験を行うにあたっては、UNIXの基礎知識やネットワークに関するある程度の知識が必要である。このため、この実験を行う実験者は知識のあるものに限られる。(例: 専攻科2年生)

5. 連続実験に対する負荷分散実験

5.1 実験の目的

この負荷分散プログラムを、長時間使用するとき、どのように負荷分散がされるかを試し、現実の処理が問題なくできるかを確認する。また、この負荷分散がネットワーク全体に関してどのような影響を与えるのか調べる。

5.2 実験方法

10秒ごと dkk01→dkk02→dkk03→dkk01... という順番にそれぞれのノードに同じANフレームを送信し、作業を依頼するプログラムを dkk01 で5分間動かし、負荷プログラムを評価する。

5.3 実験結果

dkk01のノードで負荷分散した場合と負荷分散しない場合の、CPU利用率のグラフを図4に示す。また、5分経過後のプログラムの処理数を表2に示す。

表2. 5分経過後のプログラムの処理数

	5分経過後の作業中のプログラム数			5分間で実行終了したプログラム数
	dkk01	dkk02	dkk03	
負荷分散しないとき	5	9	9	7
負荷分散したとき	2	7	9	12

5.4 考察

- (1) 図4よりCPU利用率の時間変化をそれぞれのノードについて見ると、dkk01では負荷分散をしたときのCPU利用率は、負荷分散をしないときよりも低い値となっている。
    - (a) これは負荷分散をしない場合、作業を次々に依頼するため、多重に処理する作業の数が増えてしまう。よってCPUを効率的に使用していないため、このような結果になったと考えられる。
    - (b) 負荷分散をした場合に、CPU利用率の変化が大きいのは、負荷分散プログラムにより依頼された作業を、一定のCPU利用率より低くなったときに実行するためである。加えて負荷分散をするために、様々なエージェントやサーバントを実行していることも理由として挙げられる。
  - (2) dkk02, dkk03では負荷分散をしてもしなくてもほとんど変わりが見られない。これはこのノードの性能が低いいため、一つの作業を実行するのに多くの時間を割いてしまうからだと思う。
  - (3) dkk01, dkk02において、負荷分散したときのCPU利用率の立ち上がりまでの時間が長いのは、負荷分散を行うことによって作業を他のノードに送りだしたためである。
- また、表2より、負荷分散を行った場合は5分経過後に実行中のプログラム数が全体で少なくなっている。

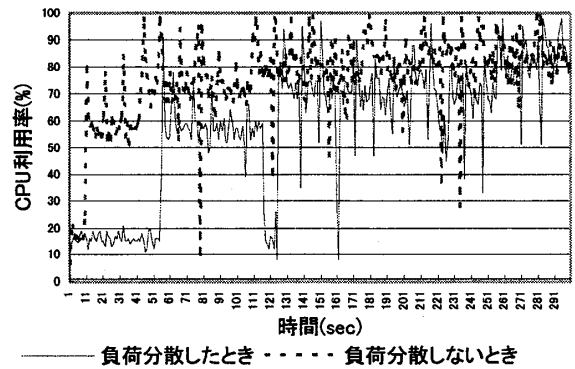


図4. CPU利用率の時間変化(dkk01)

6. まとめ

- (1) アクティブネットワークの負荷分散について、学生実験のテーマとして、実験できる目途がたった。
- (2) ネットワークプロトコルについて身近なものとして理解できる工夫が必要である。
- (3) アクティブネットワークの負荷分散のアルゴリズムについて改善し改版する予定である。