未知のノードを含むネットワーク分散協調演算処理システム環境シミュレーション

正員 阿部 一博* 正員 柿倉 正義*

The Simulation for Network Cooperative Calculation System with Unknown Nodes

Kazuhiro Abe*, Member, Masayosi Kakikura*, Member

Recently, high speed processing computers are inevitably required for many kinds of scientific and engineering applications, for example, graphics and simulations. The solution ways for this demand are, for example, cooperative controlled multi-nodes system and centrally controlled multi-nodes system.

In this paper, we propose an idea to diffuse these two systems into one system to make much more effective and speedy system. In this system, even if the communication speed between each node is relatively slow, the total processing speed is rather higher compared with two systems. The dominance is certified using simulation results. And we consider of emergency halt to realize robustness.

キーワード：処理時間、分散制御、集中制御、通信速度、緊急停止

Keywords: processing time, cooperative controlled, centrally controlled, communication speed, emergency halt

1. ま え が き

近年グラフィックスおよびシミュレーション等の分野において、計算機は必要不可欠なものとなってきている。これに伴い計算機の処理速度の向上が求められるようになってきた。具体的な手法としては、プロセッサの並列化およびプロセッサの動作周波数の上昇化とさまざまあるが、今プロセッサの並列化に着目する。プロセッサの並列化は大型の計算機に由来より多く用いられてきた手法であり、中心的な位置にある。計算機内部のプロセッサが行なう処理として、タスクの処理およびプロセッサ間通信の処理が挙げられ、これら両者間の関係が、しばしば大型計算機を大別するための指数となっている。⁽¹⁾⁽²⁾

一般に各プロセッサにおいて、タスクの処理に要する時間が通信に要する時間を上回る場合、並列化システム全体の処理速度の向上を期待することができる。従ってプロセッサ間の通信速度が低速であったとしても、単位時間に受信可能なタスクの量が大きければ、処理の送受信に費やされる通信時間が軽減することと等価となり、同様に処理速度の向上を期待することができる。

一方プロセッサの並列化を考えた場合、ネットワークに内包される計算機を並列化することもほぼ同様であると考

えることができる。ネットワークは近年普及し現在一般家庭まで深く浸透している。従ってこれに内包される計算機の数考えたとき、この数は膨大であり、この資源を有効に活用することを考えることは重要であると思われる。これに関しては従来より多くの研究^{(3)~(5)}が試みられており、今日実用化されている一般的なものとして、地球外生命体探索プロジェクトにて運営されている SETI@Home 等が挙げられる。本研究もまた効率の良い処理分散方式の一つとしてのアルゴリズム立案およびシミュレーションによる評価をおこなうものである。

今回示すシステムは、ネットワーク内の各計算機に処理を分散させ効率よく命令を実行するものである。ネットワーク内の通信速度は確かに低速ではあるが、先までの考察により実現可能であることに期待がもてる。しかし本システム内には未知の計算機を考慮するために、従来考案されたシステムではほとんど重要視されない問題が発生する。

ここではシステムを実現する上での基盤となるアルゴリズムの考察および評価、すなわち分割し易い構造を有する処理を各計算機に分散し実行させることおよび問題に対する対処を考慮することを目的としている。

本論文では第2章にて本システムの全体構想を示す。次に第3章にて全体構想の内部詳細を示す。そこでは主として処理の分散方式および本システム構成の際の各問題点に対する対処方法についてのアルゴリズムについて示す。またこの際に行った確認のためのシミュレーション結果および検討について第4章に示す。最後に第5章にて結論およ

*東京電機大学

〒101-8457 東京都千代田区神田錦町2-2

Tokyo Denki University

2-2, Nishiki-cho, Chiyoda-ku, Tokyo 101-8457

び今後の方針について述べる。

2. システムの概要

次に本システムにて要求されるシステム概要について示す。

従来よりタスクを分割し、ネットワーク上の各ノードに分散させ処理速度を向上させることは行われてきた。その代表的手法とし集中分散系および自律分散系がある。集中分散系は、ある計算機をホスト（ノードを集中的に管理するサーバを意味する）とし、ホストに接続されるノードを集中的に管理および制御し分散処理を行う構造を有するものである。しかしこの場合一括したノード管理に伴うホストの管理能力およびネットワークの稼働率がノードの数の上昇に従い比例的に悪化し、管理可能な限界が早々に発生する。さらにホストが何らかの要因で停止した場合、これを補うシステムの必要性および復旧に係る問題が大きい。

他方自律分散系は集中分散系に挙げた集中的管理を行わないため、ホストの処理限界および停止に伴う問題は大きくはない。しかし一方各ノードで同様の処理が繰り返し行われ、タスク分散が行われるため、低処理能力のノードは通信処理および通信自体に大きな負荷がかかるため、処理速度はあまり上昇せず分散速度も低下する可能性がある。

図1に示す通り本システムではこれら2つの系の利点を生かし、中継サーバとこれに接続されるノードは集中分散系とし、中継サーバ同士では自律分散系としている。本方式を用いることにより集中分散系および自律分散系に見られる問題点自体も分散されるため、ノード数の増加に対し比例的な速度の向上が期待できる。

これらをまとめて表1に示した。

また本システムではユーザにとって未知のノードにおいてユーザ送信の処理が実行される。よってシステムの頑健

表1 システム概要

Table 1. System concept.

	ネットワーク 稼働率	サーバの負荷	1クライアント ヘッダを送信可能な量
集中分散系	小	大	大
自律分散系	大	小	小
合成系	大	小	中

性が重要となる。本システムにおける頑健性は処理の分散同様非常に重要なものであり、例えばノードの実行結果が適切なものか否かを判断するための処理およびユーザが送信した処理が誤りと判断され、その処理を緊急に停止しなければならない場合の処理など様々な考えられるが、本研究ではこれら中の特に処理の緊急停止に注目したい。

3. システムの詳細

〈3・1〉 集中分散系 はじめに集中分散系について示す。図2に示すとおり集中分散系は中継サーバおよびノードの間において用いられる優先的な分散方式である。

中継サーバは、自身に接続される各ノードの単位時間あたりに処理できる量、すなわち処理能力値に比例した処理負荷に分割をし、送信の処理をおこなう。処理能力は目安的なものとして予め算出され、中継サーバに送信されている。次に分割に関するモデルを示す。ここで N_i をノード i へ一度に送信される処理とし、

$$N_i = \frac{W}{N} J_i \dots \dots \dots (1)$$

$$L \sum_{c=1} J_c$$

ただし、 W を中継サーバが有している総処理量、 N をノード数、 J_i をノード i の処理能力値および L を分割の回数とする。

このモデルは、中継サーバが要している総処理量を一括

図1 システム構成

Fig. 1. System construction.

図2 集中分散系

Fig. 2. Centrally controlled system.

して各ノードに対し分割せず、ノードの処理能力値に応じて L 回に分けて送信することを意味する。通常 W は J_i に対し非常に大きいものを想定する。

また、分割の回数は大きな値に設定しておくことにより、ノードに対して大きな処理負荷を与えずに済み、また残りの大きな処理負荷が高速な中継サーバ間での分割に使用されるため望ましい。

〈3・2〉 自律分散系⁽⁶⁾⁽⁷⁾ 次に自律分散系について示す。図1に示すとおり自律分散系は中継サーバ間にて用いられている分散方式となっている。

今中継サーバが、図3に示すとおりに接続されているものとし、また座標 (x, y) に位置する中継サーバ $S(x, y)$ は、 $S(x+1, y)$ 、 $S(x, y+1)$ 、 $S(x-1, y)$ および $S(x, y-1)$ に対し $\frac{1}{4}$ の確率にて通信を確立することが可能であるとしている。この系における中継サーバ間の処理負荷の移動量を解析する。

今 $S(x, y)$ が隣接する中継サーバと互いに通信を確立した時 $S(x, y)$ と隣接する中継サーバ間を流れる単位時間当りの処理負荷 F は、処理負荷を R および拡散速度定数を D とし、

$$F = D \cdot \nabla_{x,y} R \dots\dots\dots (2)$$

と表現される。一方時間 δt 間に中継サーバ $S(x, y)$ の処理負荷の変化は、 $F(x-\delta t, y-\delta t)\delta t - F(x+\delta x, y+\delta y)\delta t$ となるため、

$$\nabla_t R = \nabla_{x,y} F \dots\dots\dots (3)$$

と表現される。よって、

$$\nabla_t R = D \cdot \nabla_{x,y}^2 R \dots\dots\dots (4)$$

図3 自律分散系
Fig. 3. Cooperative controlled system.

図4 自律分散シミュレーション
Fig. 4. Cooperative simulation.

図5 緊急停止0
Fig. 5. Emergency halt 0.

となる。(4)式は一般に拡散方程式として知られている。したがってある中継サーバに対して大きな処理負荷がかけられた時、図4に示すとおり、その近傍の中継サーバと処理負荷の平均化を常時おこなうことにより、先の処理負荷はすべての中継サーバに均等に分散される。

また、各中継サーバの処理速度に違いがあったとしても、それに応じ処理の平均化がおこなわれることも推察される。本方式は文献(8)~(10)にて、これら詳細および応用例についてよく論じられているが、ここでは特に拡散速度定数の変化およびノードの処理能力などの要因による各々の中継サーバが有する動的な処理負荷の分散速度にも着目する。

〈3・3〉 緊急停止 次に緊急停止について示す。

はじめに図5の状態にあるとする。今中心の中継サーバに対し緊急停止命令を送信した場合、この信号を受信した中継サーバはフラグ1をたてた後に、接続されているすべての中継サーバに対し緊急停止命令を送信し、これを受信した中継サーバは先と同様フラグ1をたてる。

次に図6に示すとおり、送信後はフラグ2をたてて、以

中継サーバ	中継サーバ オブジェクト x に 関する緊急停止信 号保有 フラグ 1	中継サーバ オブジェクト x に 関する緊急停止信 号保有 フラグ 1
中継サーバ オブジェクト x に 関する緊急停止信 号保有 フラグ 1	中継サーバ オブジェクト x に 関する緊急停止信 号送信せず フラグ 2	中継サーバ オブジェクト x に 関する緊急停止信 号保有 フラグ 1
中継サーバ オブジェクト x に 関する緊急停止信 号保有 フラグ 1	中継サーバ オブジェクト x に 関する緊急停止信 号送信せず フラグ 2	中継サーバ オブジェクト x に 関する緊急停止信 号保有 フラグ 1

図 6 緊急停止 1

Fig. 6. Emergency halt 1.

中継サーバ オブジェクト x に 関する緊急停止信 号保有 フラグ 1	中継サーバ オブジェクト x に 関する緊急停止信 号送信せず フラグ 2	中継サーバ オブジェクト x に 関する緊急停止信 号保有 フラグ 1
中継サーバ オブジェクト x に 関する緊急停止信 号送信せず フラグ 2	中継サーバ	中継サーバ オブジェクト x に 関する緊急停止信 号送信せず フラグ 2
中継サーバ オブジェクト x に 関する緊急停止信 号保有 フラグ 1	中継サーバ オブジェクト x に 関する緊急停止信 号送信せず フラグ 2	中継サーバ オブジェクト x に 関する緊急停止信 号保有 フラグ 1

図 7 緊急停止 2

Fig. 7. Emergency halt 2.

後接続されている中継サーバより送信される緊急停止命令の受信のみおこなわれる。

次に図 7 に示すとおり、送信したすべての中継サーバより同一の緊急停止命令を受信した場合、フラグおよび対象のオブジェクト消去する。

4. シミュレーション

〈4・1〉 自律分散系 図 8 および図 10 に自律分散系のシミュレーション経過を示し、図 11 に自律分散系のシミュレーション結果を示す。図 8 および図 10 では、グラフ交点を中継サーバとして、交点と交点とを結ぶ線を論理通信回線として定義し、ある中継サーバより発生した処理負荷がすべての中継サーバに分散されていく経過を示している。図 8 では図 4 と同一のネットワークの構造を想定しており、縦横 30、計 900 台の中継サーバが配置されているものとする。図 8 より 1 台の中継サーバの処理負荷がすべての中継サーバに分散される様子をうかがえる。さらに図 10 では、図 4 の配置にある中継サーバ中 450 台が抜けている状況、すなわち図 9 に示すようにネットワーク構造を変化させた

図 8 処理負荷が 1 台の中継サーバより発生した場合の自律分散経過 (規則的な配列)

Fig. 8. Cooperative Flow when only one relay server has load (Regularly).

図 9 処理負荷が 1 台の中継サーバより発生した場合の自律分散環境 (ランダム)

Fig. 9. Cooperative result when only one relay server has load (Random).

状況における自律分散経過を示したものである。

ここで図 9 において、四角の塗りつぶし [■] が各々の中継サーバを示しており、互いに隣接している中継サーバとの通信をおこなうことが可能であるとしている。

図 10 よりネットワーク構造によらずに、処理負荷がすべての中継サーバに分散される様子がうかがわれる。さらに図 11 に示すとおり集中分散に比較し今回示す分散方法が高速に処理負荷を分散できることもうかがわれる。さらに拡散速度定数の変化によりその速度が向上していく様子も観察される。さらに図 8 の環境下において拡散速度定数 1.0 にて分散完了までに費やされたステップ数は 727 となっており、図 9 の環境下において拡散速度定数 1.0 にて分散完了までに費やされたステップ数は、2772 となることを言及する。

図 10 処理負荷が 1 台の中継サーバより発生した場合の自律分散経過 (ランダム)

Fig.10. Cooperative flow when only one relay server has load (Random).

図 12 中継サーバの処理負荷が動的に変化した場合の自律分散経過 (規則的な配列)

Fig.12. Cooperative flow when relay servers have dynamical processing load (Regular).

図 11 処理負荷が 1 台の中継サーバより発生した場合の自律分散結果 (規則的な配列)

Fig.11. Cooperative result when only one relay server has load (Regular).

図 13 中継サーバの処理負荷が動的に変化した場合の自律分散結果 (規則的な配列)

Fig.13. Cooperative result when relay servers have dynamical processing load (Regular).

〈4・2〉 動的な処理負荷の変化 図 12 に各中継サーバにおける単位時間当りの処理量の違いにより発生する処理負荷の動的変化が高速に吸収されるか否かについてシミュレーション結果を示す。図 13 に示すとおり、すべての中継サーバに処理負荷が分散されるまでのステップ数が低いため、処理負荷の違いが高速に吸収されていくことが伺われる。

〈4・3〉 緊急停止 図 14 に緊急停止に関するシミュレーション結果を示した。本シミュレーションは単一の中継サーバより緊急停止信号を送信し、システム全体に伝播するまでの速度を計測した物である。図 14 に示すとおり集中的に伝播させたものに対し高速に伝播される様子が伺われる。

図 14 緊急停止

Fig.14. Emergency halt.

5. 結論および今後の方針

今回示すシステムはネットワーク内の各計算機に処理を分散させ、効率よく処理を実行させるものである。ネットワーク内の通信速度は確かに低速であるが実現可能であることに期待がもてる。また本システムの頑健性も重要な項目であり、例えを挙げるならばユーザが送信した処理が誤りと判断され、その処理を緊急に停止しなければならない場合が想定される。

本稿では、本システムを実現するための基盤とされるソフトウェアの構想、理論の検討およびそれら各シミュレーションの結果について示すと共にその妥当性について検討をおこなった。

結果より今回検討した範囲についてはシステムの如く問題なく動作することが検証された。すなわち、環境によることなく、分散に関しては集中分散系のみを使用に比較し高速におこなうことができ、緊急停止に関しても同様にネットワーク全体に緊急停止信号を高速に伝播することが可能である。しかし他方で中継サーバの配列が複雑化してくるとそれに比例して分散速度が著しく低下する。

また今回のシステム作成検討にあたり、問題も多く発見された。一つはセキュリティーに関する問題である。本システムはユーザもしくは各々の中継サーバにとって、未知の部分にて動作する要素が多い。すなわち、未知のユーザの作成した処理が未知のノードにて実行されるため、処理の結果の補償およびノードの安全性の補償に関する検討は非常に重要であると思われる。

他方、システムを開発するにあたって、そのシステムに制御可能なある種の柔軟性をもたせることもまた必要であると思われる。この柔軟性についても規則を必要とし、例えばそれを言語として体系化させることがその一つとして考えられ得る。

以上、考慮すべき点はまだ多くあり実際の起動にいたるには多くの時間を必要としている。

なお、本研究は情報処理振興事業協会、未踏ソフトウェア創造事業 10-13「未知のノードを含むネットワーク分散協調演算処理システム」の委託を受けた内の一部である。

またこの度本案件をご採択いただいた東京大学 松島 克守教授、本案件遂行にあたりサポートをしていただいた、ネイチャーランドジャパン株式会社 斎藤 昌義氏、難波 薫氏、日本無線(株) 馬場 満徳氏およびアイアンドエル(I&L)・ソフトウェア株式会社 廣澤 稔氏、近藤 充弘氏の各位に対し深く謝意を表す。

(平成 13 年 11 月 13 日受付, 同 14 年 7 月 24 日再受付)

文 献

- 飯塚 肇・緑川博子 共訳: 並列プログラミング入門 ―ネットワーク結合 UNIX マシンによる並列処理―, 丸善 (2000)
- (2) S. Brawer: "Introduction to Parallel Programming", Academic Press, (1989) (in Japanese)
大森健児 訳: 並列プログラミングの基礎, 丸善 (1990)
- (3) N. Abe, Y. Meguro, M. Nakazawa, and S. Hattori: "Dynamic Load Control by Multi-server Client Model Based on the Distributed and Autonomous Agents", *IEICE B-I*, Vol. J79-B-I, No.5, pp.310-320 (1996-5) (Japanes)
阿部倫之・日黒雄峰・中沢 実・服部進実: 「自律分散エージェントによるマルチサーバクライアントモデルの動的負荷制御」, 信学論 B-I, **J79-B-I**, 5, pp.310-320 (1996-5)
- (4) H. Yamana, M. Sato, Y. Kodama, H. Sakane, S. Sakai, and Y. Yamaguchi: "A Distributed Control Scheme of Macrotask-level Speculative Execution on the EM-4 Multiprocessor", *IPSS*, Vol.36, No.07-010 (1995) (in Japanese)
山名早人・佐藤三久・児玉祐悦・坂根広史・坂井修一・山口喜教: 「並列計算機 EM-4 におけるマクロタスク間投機実行の分散制御方式」, 情報処理学会誌, **36**, 07-010 (1995)
- (5) B. Wilkinson: "Computer Architecture design and performance second edition", Prentice Hall Europe (1996) (in Japanese)
高橋義造 監訳・渡辺 尚・小林真也・長谷川誠 訳: 計算機設計技法 マルチプロセッサシステム論 第 2 版, トップラン (1998)
- (6) G. Blom, L. Holst, and D. Sandell: "Problems and Snapshots from the World of Probability", Springer Verlag in New York, (1994) (in Japanese)
森 貞 訳: 確率問題ゼミ コイン投げからランダムウォークまで, Springer-Verlag 東京, (1997)
- (7) 寺田文行・樋口権一: 高校数学解法事典, 別冊, 旺文社 (2000)
- (8) 新 誠一・池田建司・湯浅秀男・藤田博之: 日本ファジー学会編: 自律分散システム, 朝倉書店 (1995)
- (9) T. Ishida and A. Shimizu: "Semiconductor Devices", Corona Publishing Co., Ltd. (1994) (in Japanese)
石田哲郎・清水 東: 標準電気工学講座 半導体素子, コロナ社 (1994)
- (10) 深井 有: 現代人の物理 3 拡散現象の物理, 朝倉書店 (1999)

阿部 一博 (正員) 1973 年 7 月 26 日生 1998 年 3 月東京電機大学大学院工学研究科電子工学専攻修了。同年日本無線(株)入社。現在同社モバイル研究所所属。2002 年 4 月東京電機大学大学院工学研究科電子工学専攻博士後期過程入学。現在同大学に在学。

柿倉正義 (正員) 1942 年 9 月 4 日生 1965 年東京電機大工電子卒。同年、電気試験所(現産業技術総合研究所)入所。1980 年情報制御研究室長、1988 年自律システム研究室長。1974-1975 年エジンバラ大学人工知能学科。1990 年東京電機大工学部電子工学科教授。工学博士(東京工業大学)

- (1) B. Wilkinson and M. Allen: "Parallel Programming - Techniques and Applications Using Networked Workstations and Parallel Computers-", Prentice-Hall, Inc., Pearson Education Company (1999) (in Japanese)