解　説

フォールトトレラント分散システム向けアルゴリズム

1. フォールトトレラント分散システム向けアルゴリズム概論†

萩原兼一††

1. はじめに

分散システムは，地理的に離れた地点にある複数のプロセッサ（計算機）がネットワーク（通信路）で接続され，それらが通信し合いながら，協調して動作してある共通の目的を達成するものである[29),39),45)]．

分散システムが構築される理由として以下のものが考えられる．

●情報の分散発生：取り扱う情報自身が地理的に離れた所で発生するため，発生地点で処理するのが自然な場合がある．たとえば企業のある支店の営業に関するデータはその支店で管理するであろう．そして，複数の支店にまたがる顧客処理や本社への統計データの送付など，他との連係が必要な場合にのみ該当プロセス間の協調処理を行うように設計するであろう．

●処理効率：処理を分散させることによって，比較的安価に高い効率を達成可能となる場合がある．ある種のシミュレーション問題などのように，問題がいくつかの部分問題に分割可能で，各部分問題計算中において他の部分計算との通信が比較的少なければ，各部分問題を別々のプロセッサに割り当てることで，高価な並列計算機を使わなくともある程度高速に計算できる場合がある[23)]．

●拡張可能性：システムの規模の拡張および新しい機能の追加などを比較的簡単に行える場合がある．拡張はプロセッサおよびネットワーク装置の追加により，現在動作中の他の部分に対する変更をほとんど行わずに達成できる．

●フォールトトレランス：一部のプロセッサおよび通信路に故障が存在する場合においても残った部分でシステムが所望の動作を行える可能性がある．

●セキュリティ：複数の組織からなるシステムの場合，互いに他組織にアクセスされたくない情報や機能をもちながら協調動作を行う場合がある．したがって機能および情報隠蔽のため，各組織が別々のプロセッサを持ち，互いのアクセス方法を限定した接続形態をとることがある．

しかし，上記の性質を満足するシステムの設計は一般に容易ではなく，上記の各性質を達成する手法の研究が盛んに行われてきている．

本稿ではこのうち，フォールトトレランスに関するアルゴリズム的な手法について述べる．分散システムの規模が大きくなると，接続されているすべてのプロセッサ，およびすべての通信路が同時に無故障であると期待することはほとんど不可能である．したがって，フォールトトレランスは最も重要な機能の一つである．なお，一般的な分散アルゴリズムに関するモデルについては文献 17) を参照のこと．

2. フォールトトレラント分散アルゴリズムの分類

フォールトトレラント分散システムを達成するためのアルゴリズム的手法は大きく以下の三つに分類される．

（1） 故障検出：故障を起こしている部分（プロセッサ，通信路）を検出する．

（2） 故障排除：故障の影響を排除し，無故障なプロセッサのみで正しい結果を得る．

（3） 故障復旧：故障を起こしたプロセッサを正しい状態に復帰させる．

故障が一時的なものでない（ある処理を行う最後まで故障が継続している）場合については，故

† Algorithms for Fault-Tolerant Distributed Systems by Ken'ichi HAGIHARA (Nara Institute of Science and Technology).

†† 奈良先端科学技術大学院大学情報科学研究科情報システム学専攻

障箇所（プロセッサあるいは通信路）を明確にすることが可能であれば，その部分を取り除いたシステムにおいて処理を行うことにより，正しい結果を得ることができる．すなわち，故障検出が可能である場合には故障影響の排除も可能である．よって故障検出は故障排除よりも一般に困難な課題である．また，ネットワークが非同期であるという分散システムにおける一般的な仮定のもとでは，遅延と故障による停止とを有限時間では区別できない．このため，故障検出の研究は文献 9), 11), 24), 25), 34), 48) など少数である．したがって，故障排除は一般に故障箇所を陽に検出することなく無故障な部分で正しい結果を得るための手法である．

また，フォールトトレラント分散システムを達成するためのアルゴリズムの研究を，そのアルゴリズムが解く問題をもとに分類すると以下のようになる．

（a） 対象問題特定

(ax) ベースとなる通信処理（1対1通信，1対多通信など）

(ay) 分散システムの基本問題

（b） 対象問題不特定（任意のプログラム）

対象問題を特定すると，解こうとする問題の性質を利用することにより，より厳しい故障状況に耐えられ，かつ効率のよいアルゴリズムが得られる可能性がある．対象問題を不特定とするアルゴリズムは，任意の分散プログラムと組み合わせて使用することが可能であるが，対処可能な故障状況は制限されたもので，かつその効率は低い．

よって，フォールトトレラント分散システムを達成するためのアルゴリズムの研究は上記の組合せとして以下のように分類される．

(2-ax) 通信処理に対する故障排除アルゴリズム：

1対1通信に対する故障排除については，故障発生時のルーティング問題などが研究されている[8]．放送型の通信に関しては，原子性（全宛先で受信されるかまったく受信されないかのいずれかが必ず成立する）・全順序性（全宛先で同一の順序で受信される）などの放送型通信がもつべき性質を実現するアルゴリズムについて研究されている．

(3-ax) 通信処理に対する故障回復アルゴリズム：

通信に関する故障回復としては正しく受信されなかったメッセージの再送処理が古くから研究されている[41]．

(2-ay) 基本問題に対する故障排除アルゴリズム：

分散システムにおける以下のような基本問題に対して，故障プロセッサあるいは通信路の影響を排除して動作する分散アルゴリズムが考えられている．すべてのプロセッサが同じ値を得る合意(agreement)問題[47]，プロセッサの中から唯一のプロセッサを選択するリーダ選択 (leader election) 問題[6),12),17),26),28),40),42),43]，ネットワークの生成木(spanning tree) 構成問題[22]，複数のプロセッサからの要求のうちの唯一に許可を与える相互排除(mutual exclusion) 問題，一つのプロセッサからの情報を全プロセッサに伝える情報散布 (information dissemination) 問題[19),46]，各プロセッサがもつ値を入力とする関数値計算問題[15]，プロセッサのグループ維持問題[37]，分散データベースに対する更新問題[7]などである．

(3-ay) 基本問題に対する故障回復アルゴリズム：

誤った値をもつプロセッサを復旧させながら動作する自己安定アルゴリズムが，相互排除問題などの基本問題に対して考えられている．

(2-b) 任意のプログラムに対する故障排除ルゴリズム：

プログラムに依存しない故障排除の方法としては，複数のプロセッサが同じ動作を行う冗長システムを構成し，出力の多数決を取る手法が研究されている[35]．また，ある一つの機能をプライマリ(primary) プロセッサと複数のバックアップ(backup) プロセッサの組で構成してプライマリが故障した場合にバックアップに切り替えるためのアルゴリズムが考えられている[10),18]．

(3-b) 任意のプログラムに対する故障回復アルゴリズム：

プログラムに依存しない故障回復の手法としては，動作中に途中の状況を保存しておいて故障が発生した場合に保存した状態に戻るチェックポイント・ロールバックアルゴリズムが考えられている．

本特集の各解説においては，これらの各項の中から近年よく研究されているものについて具体的

アルゴリズムを紹介する．(2-ax)からは放送型通信アルゴリズムを，(2-ax)からは分散相互排除アルゴリズムを，(3-ay)からは自己安定アルゴリズムを，(3-b)からは分散チェックポイント・ロールバックアルゴリズムを取り上げる．

これらフォールトトレラント分散アルゴリズムを考察する上での，分散システムにおけるプロセッサの故障形態として，主に以下のものが考えられている．これ以外の故障形態は文献 14) などで述べられている．

- 停止故障 (crash failure または fail-stop)：故障したプロセッサは停止し，まったくメッセージを送らない．
- 送信脱落故障 (send-omission failure)[16]：送信しなければならないメッセージが送信されないことがある．
- 一般脱落故障 (general-omission failure)[32]：送信，受信のいずれも脱落がある．
- ビザンチン故障 (Byzantine failure)[47]：故障したプロセッサが何を行うかについて，まったく仮定をおかない．この場合，故障プロセッサはでたらめなメッセージを送信することもあり得る．

ビザンチン故障および脱落故障のもとでは，合意問題に対するアルゴリズムが主に考察されている[47]．合意問題以外の基本問題に対しては考察されている故障形態は停止故障のみが多い．また，停止故障にのみ対処可能なアルゴリズムを脱落故障にも対処可能に，脱落故障にのみ対処可能なアルゴリズムをビザンチン故障にも対処可能に変換する手法も研究されている[5),30)]．

さらに，故障の特殊な場合として，ネットワークトポロジの動的な変更がある．プロセッサあるいは通信路が除去されるトポロジ変更は，プロセッサあるいは通信路に故障が発生した場合に，故障発生を周囲に伝えた後に停止する場合とみなすことができる．また逆に，プロセッサあるいは通信路が新たに参加するトポロジ変更は，故障から回復してそのことを周囲に伝えた場合とみなすことができる．いったん解が求まって終了している状態においてトポロジ変更が発生した場合に，トポロジ変更に追随して解を変更するアルゴリズムが，生成木問題[27),31),36)]，各プロセッサから他のプロセッサへの最短経路を求める最短路問題[20),33)]，プロセッサグループ情報の維持[38)]などの基本問題に対して考えられている．また，アルゴリズムの動作中にトポロジ変更が起こる場合にも対処可能なアルゴリズムも考えられている[1)~4),21),44)]．

3. おわりに

フォールトトレラント分散システム向けアルゴリズムの主な手法の分類を行った．なお，1990年ごろまでのフォールトトレラント分散アルゴリズムに関する文献リストが文献 13) および文献 17) にある．

現在主に考えられている故障形態は，現実の分散システムによく合っているとは言えない場合もある．したがって，現実のシステムにより近い故障形態のモデル化が必要であろうと思われる．そのモデル化がうまくなされれば，さらに有用なフォールトトレラント分散システム向けアルゴリズムが生まれてくるであろう．

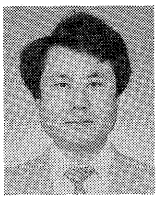
参考文献

1) Afek, Y. and Gafni, E.: End-to-End Communication in Unreliable Networks, Proc. 7th Principles of Distributed Computing, pp. 131-148 (Aug. 1988).
2) Awerbuch, B. and Sipser, M.: Dynamic Networks Are as Fast as Static Networks, Proc. 29th Foundations of Computer Science, pp. 206-220 (Oct. 1988).
3) Awerbuch, B., Cidon, I. and Kutten, S.: Communication-Optimal Maintenance of Replicated Information, Proc. 31st Foundations of Computer Science, pp. 492-502 (Oct. 1990).
4) Awerbuch, B. and Mansour, Y.: An Efficient Topology Update Protocol for Dynamic Networks, Proc. 6th Distributed Algorithms, pp. 185-202 (Nov. 1992).
5) Bazzi, R. and Neiger, G.: Simulating Crash Failures with Many Faulty Processors, Proc. 6th Distributed Algorithms, pp. 166-184 (Nov. 1992).
6) Becker, T.: Keeping Processes Under Surveillance, Proc. 10th Reliable Distributed Systems, pp. 198-205 (Oct. 1991).
7) Bernstein, P. A., Hadzilacos, V. and Goodman, N.: Concurrency Control and Recovery in Database Systems, Addison-Wesley (1987).
8) Bertsekas, D. and Gallager, R.: Data Networks, Prentice-Hall (八星訳：データネットワーク，オーム社) (1987).
9) Bondavalli, A. and Simoncini, L.: Failure Classification with Respect to Detection, Proc.

2nd Future Trends of Distributed Computing Systems, pp. 47-53 (Sep. 1990).
10) Budhiraja, N., Marzullo, K., Schneider, F. B. and Toueg, S.: Optimal Primary-Backup Protocols, Proc. 6th Distributed Algorithms, pp. 362-378 (Nov. 1992).
11) Chandra, T. C. and Toueg, S.: Unreliable Failure Detectors for Asynchronous Systems, Proc. 10th Principles of Distributed Computing, pp. 325-340 (Aug. 1991).
12) Chow, Y.-C., Luo, K. C. K. and Newman-Wolfe, R.: An Optimal Distributed Algorithm for Failure-Driven Leader Election in Bounded-Degree Networks, Proc. Future Trends of Distributed Computing Systems, pp. 136-141 (Apr. 1992).
13) Coan, B. A.: Bibliography for Fault-Tolerant Distributed Computing, Proc. Asilomar Workshop on Fault-Tolerant Distributed Computing, Lecture Notes in Computer Science, No. 448, pp. 274-298 (1990).
14) Ezhilchelvan, P. D. and Shivastava, S. K.: A Characterisation of Faults in Systems, Proc. 5th Reliability in Distributed Software and Database Systems, pp. 215-222 (Jan. 1986).
15) Goldreich, O. and Shrira, L.: The Effect of Link Failures on Computations in Asynchronous Rings, Proc. 5th Principles of Distributed Computing, pp. 174-185 (Aug. 1986).
16) Hadzilacos, V.: Connectivity Requirements for Byzantine Agreement under Restricted Types of Failures, Distributed Computing, Vol. 2, No. 2, pp. 95-103 (1987).
17) 萩原, 増澤: 分散アルゴリズム, 情報処理, Vol. 31, No. 9, pp. 1245-1256 (Sep. 1990).
18) 桧垣, 曽根岡: 分散システムにおけるフォールトトレランスのためのグループ通信,「マルチメディア通信と分散処理」ワークショップ, pp. 41-48 (Mar. 1993).
19) 五十嵐, 小保方, 三浦: Binary Jumping による情報散布方式の耐故障性について, 信学技報 COMP 92-16 (1992).
20) Italiano, G. F.: Distributed Algorithms for Updating Shortest Paths, Proc. 5th Distributed Algorithms, pp. 200-211 (Oct. 1991).
21) Kung, R. and Shacham, N.: A Distributed Limited-Depth Probing Protocol for a Network with Changing Topology, Proc. 1984 Parallel Processing, pp. 356-358 (Aug. 1984).
22) Kutten, S.: Optimal Fault-Tolerant Distributed Construction of a Spanning Forest, Inf. Process. Lett., Vol. 27, pp. 299-307 (May 1988).
23) Lynch, D. L. and Rose, M. T.: Internet System Handbook, Addison-Wesley, pp. 741-743 (1993).
24) 増澤, 萩原, 都倉: プロセッサ故障診断のための分散アルゴリズム, 信学論(D), Vol. J 70-D, No. 6, pp. 1092-1103 (June 1987).
25) 増澤, 萩原, 都倉: リンク故障診断のための分散アルゴリズム, 信学論(D), Vol. J 71-D, No. 12, pp. 2648-2658 (Dec. 1988).
26) 増澤, 西川, 萩原, 都倉, 藤田: 方向感覚付き完全ネットワークにおけるリンク故障を考慮したリーダ選択問題, 信学技報 COMP 88-98 (1988).
27) 増澤, 三浦, 朴, 都倉: ネットワークにおける最短経路木構成問題と幅優先木更新問題, 信学技報 COMP 91-6 (1991).
28) 松本, 若林, 小出, 吉田: 任意形状の非同期ネットワークにおける耐故障リーダ選挙について, 信学技報 COMP 92-63 (Nov. 1992).
29) Mullender, S.: Distributed Systems, Addison-Wesley (1989).
30) Neiger, G. and Toueg, S.: Automatically Increasing the Fault-Tolerance of Distributed Algorithms, J. of Algorithms, Vol. 11, pp. 374-419 (1990).
31) Park, J., Masuzawa, T., Hagihara, K. and Tokura, N.: Distributed Algorithms for Reconstructing MST After Topology Change, Proc. 4th Distributed Algorithms, pp. 122-132 (1990).
32) Perry, K. J. and Toueg, S.: Distributed Agreement in the Presence of Processor and Communication Failures, IEEE Trans. Software Eng., Vol. SE-12, No. 3, pp. 477-482 (1986).
33) Ramarao, K. V. S. and Venkatesan, S.: On Finding and Updating Shortest Paths Distributively, J. of Algorithms, Vol. 13, pp. 235-257 (1992).
34) Ramarao, K. V. S. and Venkatesan, S.: Distributed Problem Solving in Spite of Processor Failures, Proc. 11th Reliable Distributed Systems, pp. 164-171 (Oct. 1992).
35) Randell, B., Lee, P. A. and Treleaven, P. C.: Reliability Issues in Computing System Design, ACM Computing Surveys, Vol. 10, No. 2, pp. 123-165 (June 1978).
36) Ravindran, K., Singh, G. and Gupta, P.: Reconfiguration of Spanning Trees in Networks in the Presence of Node Failures, Proc. 13th Distributed Computing Systems, pp. 219-226 (May 1993).
37) Ricciardi, A. M. and Birman, K. P.: Using Process Groups to Implement Failure Detection in Asynchronous Environments, Proc. 10th Principles ofDistributed Computing, pp. 341-353 (Aug. 1991).
38) Ricciardi, A., Birman, K. and Stephenson, P.: The Cost of Order in Asynchronous Systems, Proc. 6th Distributed Algorithms, pp. 329-345 (Nov. 1992).
39) 坂下, 井手口, 滝沢, 水野: 分散システム入門, 近代科学社 (1993).
40) Singh, S.: Expected Connectivity and Leader Election in Unreliable Networks, Inf. Process. Lett., Vol. 42, pp. 283-285 (July 1992).
41) Tanenbaum, A. S.: Computer Networks, Pren-

tice-Hall (1989).
42) Taubenfeld, G.: Leader Election in the Presence of $n-1$ Initial Failures, Inf. Process. Lett., Vol. 33, pp. 25-28 (Oct. 1989).
43) Taubenfeld, G., Katz, S. and Moran, S.: Impossibility Results in the Presence of Multiple Faulty Processes, Proc. Foundations of Software Technology and Theoretical Computer Science, pp. 109-120 (Dec. 1989).
44) Tel, G.: Directed Network Protocols, Proc. 2nd Distributed Algorithms, pp. 1-13 (July 1987).
45) 塚本, 八田他：特集「分散処理技術」, 情報処理, Vol. 28, No. 4, pp. 369-546 (Apr. 1987).
46) 宇多, 萩原, 魚井, 首藤：プロセッサ網における効率的な情報散布方式のリンク故障耐性について, 信学技報 COMP 91-91 (1991).
47) 山下雅史：ビザンティン合意問題, 情報処理, Vol. 32, No. 6, pp. 682-693 (June. 1991).
48) Yang, C.-L. and Masson, G. M.: A Distributed Algorithm for Fault Diagnosis in Systems with Soft Failures, IEEE Trans. Comput. Vol. C-37, No. 11, pp. 1476-1480 (Nov. 1988).

(平成5年9月6日受付)



萩原 兼一

1952年生. 1974年大阪大学基礎工学部情報工学科卒業. 1979年同大学院基礎工学研究科博士課程修了. 工学博士. 同年, 同大学基礎工学部助手. 同講師, 同助教授を経て, 1993年より奈良先端科学技術大学院大学教授（ソフトウェア基礎講座）. 1992年～1993年文部省在外研究員（メリーランド大学）. 並列/分散アルゴリズム, ユーザインタフェースなどに興味をもつ. 著書「基礎 PASCAL」(岩波書店). 訳書「VLSI 計算の諸側面」(共訳, 近代科学社).