ネットワーク管理におけるイベントのコリレーション処理に関する一考察

1C－9

横田 英俊　千葉 和彦　浅見 徹
国際電信電話株式会社 研究所

1. はじめに

ネットワークの障害管理を効率的に行なうためには、物理的接続や、知識処理に基づいたアルゴリズム等を使用して全ての警報間での関連付けを行ない、問題の原因を識別できるようにする必要がある[1]。この警報間の関連付けをコリレーション処理と呼ぶ[2]。しかし、管理対象の数が増加するにつれて一つの障害に対する警報の数も膨大となるため、効率的なコリレーション処理が必須となる。本稿では、このような警報間のコリレーション処理を高速に実現する方式について考察する。

2. 警報の発生とコリレーション処理

ネットワークを管理するシステムは障害管理機能として、管理対象からの警報をイベントとして取り込み、警報を通知した管理対象間の接続関係やネットワーク管理における知識を用いて障害の同定を行なう[3]。しかし、一つの障害に対して非常に多くのイベントが発生する可能性があり、管理システムはこれらのイベントを取りこぼしなく処理しなければならないという実装上の問題がある。そこで、障害管理に関わる処理を図1のように階層化し、管理システムが障害の同定を行なう前にコリレーション処理を行なうことでイベントの数を絞り込み、処理の効率化を図ることができる。本稿では図1-第2層のコリレーション層に着目し、その高速な処理方式について考察する。

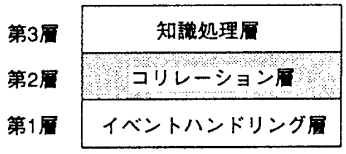

図 1: 障害管理の階層化

3. コリレーション処理の実現方式

3.1 ハッシングによるコリレーション処理の高速化

一つの障害に対して互いに関連して発生する警報(以後、イベントと呼ぶ)の集合を関係集合と定義する(図2)。このような関係集合をあらかじめ定義しておけば、発生したイベントを関係集合内のイベントに関連付けることでコリレーション処理を実現することができる。

一つの障害から発生する複数のイベントのコリレーション処理を上述の関係集合の探索と考えると、ハッシングを用いることで高速化が図れることが予想される。ここで、関係集合間の類似性を表す測度として相関係数γを定義する。Nを関係集合の個数、Cを一つの関係集合に含まれる、互いに関連するイベントの個数(今回は一定値とする)、$R_k(n)$をk番目の関係集合$(1 \leq k \leq N)$のn番目の関連イベントとして、関係集合Tの相関係数γ_Tを以下のように定義する。

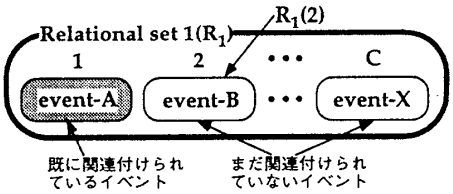

図 2: 関係集合

$$\gamma_T = \frac{1}{N-1} \sum_{k=1,k\neq T}^{N} \left[\frac{1}{C} \sum_{n=1}^{C} \sum_{m=1}^{C} R_T(n) \otimes R_k(m) \right]$$

但し、

$$a \otimes b = \begin{cases} 1 & (a = b) \\ 0 & (a \neq b) \end{cases}$$

到着したイベントのコリレーション処理(関係集合の探索)の結果、ある関係集合が持つ全てのイベントが関連付けられたとき、その関係集合は“完了した”と呼ぶ。最初の関連するイベントが到着してから、関係集合が完了するまでの時間を完了時間と定義する。相関係数をパラメータとしたときの、関係集合の個数Nと完了時間Tの関係を図3に示す。評価環境として、SPARC-server1000(2CPU、OS:Solaris2.3)を用いた。以降の評価実験では全て$C = 5$とし、ハッシュテーブルのサイズは十分大きく、ハッシュの衝突は起きないものとする。

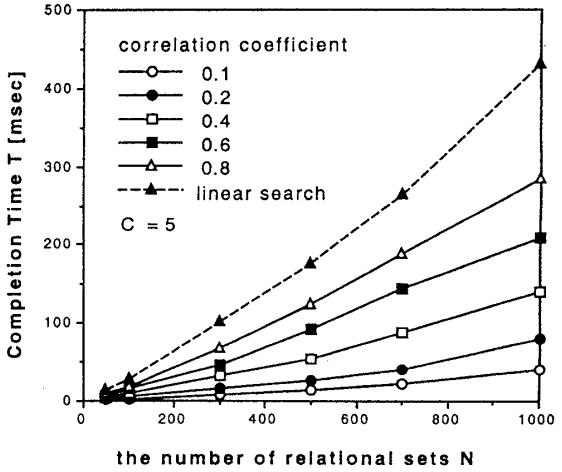

図 3: 相関係数とハッシングによる関係集合の探索の関係

図3より、相関係数が低いところでは、ハッシングによる高速化が顕著に見られるが、相関係数が高いほどそ

“Study on Correlation Processing of Events in Network Management”
Hidetoshi YOKOTA, Kazuhiko CHIBA and Tohru ASAMI
KDD R & D Laboratories

の効果が減少し、線形探索に近づくことがわかる。したがって、相関係数の高いところで、より高速にコリレーション処理を行う手法が要求される。

3.2 動的階層化ハッシングによる高速化

ハッシングによる関係集合の探索の高速化だけではなく、複数のイベントが到着する時間の相関性(時間相関)を利用してより高速化を図る、動的階層化ハッシング(Dynamically Hierarchical Hashing:DHH)を用いたコリレーション処理手法を提案する。関係集合は関連付けの効率化のため階層化し、より多くのイベントが関連付けされているものほど上層に配置する(図4)。新しいイベントが発生したら、まず上層から、自分が関連付けられるべき関係集合が存在するかどうかを探索する。存在すればイベントを関係集合に関連付け、その関係集合を一つ上層へ移動させる。関連するイベントがなければ、さらに下層の方へ探索を移行していく。上層で行われている探索ほど優先的に処理されるようスケジューリングすることで、より効率的なコリレーション処理を行うことができる。完了した関係集合は、直ちに図1-第3層の知識処理層にメッセージを通知する。また既に完了した関係集合は探索の対象から外し、参照されないようにする。

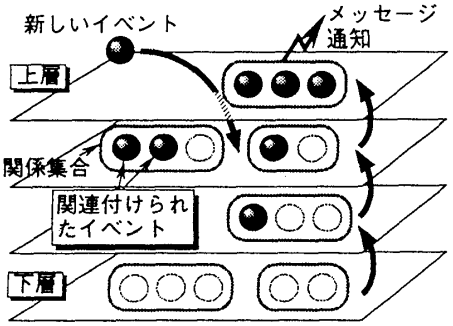

図 4: DHH によるコリレーション処理の概念図

3.1の評価環境のもとで、単純なハッシングを用いた場合とDHHを用いた場合について、相関係数γと完了時間Tの関係を図5に示す。但し、関係集合の数$N = 50$とする。

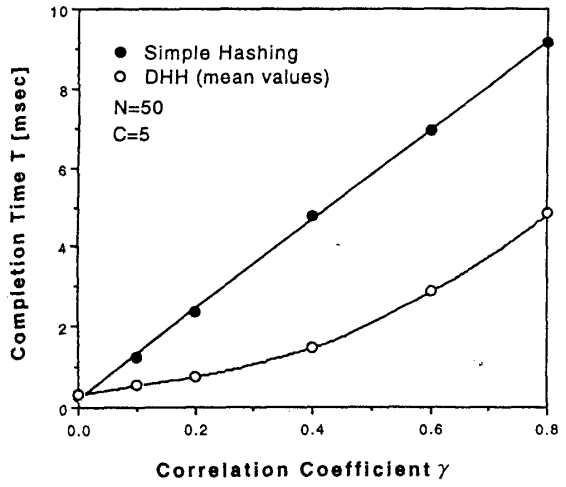

図 5: 単純ハッシングと DHH の比較

図5より、単純ハッシングと比較して、DHHによる手法の方が完了時間を最大約50%短縮できることが示された。

4. マルチプロセッシングによる高速化

DHHにおいても相関係数が高いところでは、低いところに比べて、完了時間の大きさが目立つ。相関係数の高いところでの処理をさらに高速化するために、到着したイベントが関係集合を探索する処理を一つのスレッドとして構成し、複数のイベントが並列に探索を行うようマルチスレッド化することでマルチプロセッサ環境での高速化を図る。3.1に示した評価環境のもとで、関係集合の個数Nと完了時間Tの関係を図6に示す。

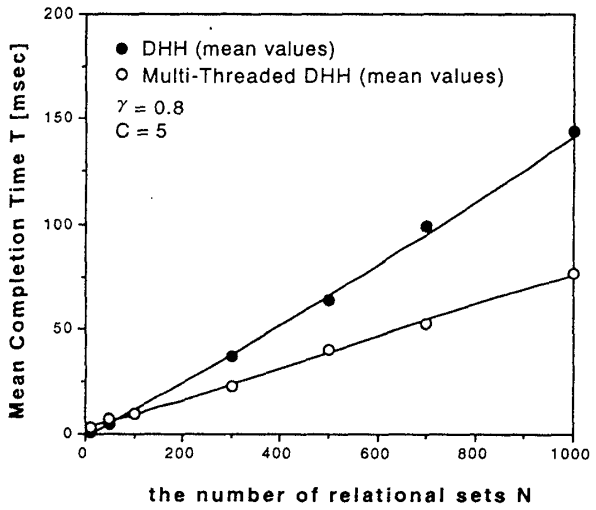

図 6: マルチスレッドによる DHH

図6より、マルチプロセッサ(2CPU)環境で関係集合の探索を並列動作させることで、完了時間を最大約47%短縮できることが示された。

5. 考察

DHHを用いることで、平均完了時間は、通常のハッシュに比べて相関係数$\gamma = 0.8$のところで、最大約50%短縮でき、さらに関係集合の探索をスレッド化することで、マルチプロセッサ(2CPU)環境において最大約47%短縮が図れることが示された。

6. おわりに

本稿ではネットワーク管理における警報間の関連付け(コリレーション処理)を効率的に行なう手法として、発生するイベントの時間的相関性を利用した、動的階層化ハッシング(DHH)による手法を提案し、その有効性を示した。最後に日頃御指導頂くKDD研究所浦野義頼所長、眞家健次次長に感謝します。

参考文献

[1] "ディスカバー・オムニポイント"、Network Management Forum、電気通信協会、1993.
[2] Gabriel Jackson and Mark D. Weissman: "Alarm Correlation", IEEE Network, pp.52-59, November 1993.
[3] 横田 他: "ネットワーク管理におけるイベントのコリレーション処理に関する一検討"、1994信学春季全大、B-627、1994.