6E-8

DHT をベースとした一括更新可能な複数名前解決方式に関する一検討

山越　公洋　　関　良明

日本電信電話株式会社　NTT 情報流通プラットフォーム研究所

１　はじめに

物流管理において、RFIDタグによる業務効率化や信頼性向上を目的としたSCM（Supply Chain Management）の分野では、大量のイベント情報を確実に所定の蓄積ノードに配信するために、宛先名前解決技術が必要となる。製造業者識別番号、製品種別番号、シリアル番号からなるEPC（Electronic Product Code）を基にDNS（Domain Name Service）の仕組みを用いて名前解決する方法[1]が提案されているが、設定変更に対する柔軟性に欠けるという問題がある。

２　名前解決

２．１　DHTによる名前解決

物流ネットワークへの新規参加や離脱の自由度が比較的高いオープンな形態のSCMでは設定変更に対する柔軟性が要求される。また、導入初期は小規模システムでスタートし徐々に規模を拡大していくような用途では、システム拡張に対する柔軟なスケーラビリティが要求される。

Chord[2]に代表される分散ハッシュテーブル（DHT：Distributed Hash Table）方式では、ノードの新規参加や離脱およびシステム規模の変更に対する柔軟性を備える。DHTでは、ホスト名やファイル名といった検索対象の識別子のハッシュ値を検索キーとして、この検索キーと検索対象アドレスのペアーを複数のノード間で分散的に保持する。検索要求があると、検索対象に対応する検索キーを所有するノードを探し当てるまでルーティングテーブルに従って問合せ要求を次ノードへ転送する。当該検索キー情報を保持する担当ノードは自ハッシュテーブルから該当するキーを探し、対応するキーの登録アドレスを要求元へ送信する。

２．２　DHTによる名前解決の課題

SCMでは、イベント情報を関連する複数の事業者に向けて同時配信するような用途が想定される。例えば製品が流通経路上の最終到達点である小売店に到着したイベント情報を、製造元、倉庫、卸業者、集配センタというように複数宛先に向けて同時配信するケースが考えられる。

単一検索キーによる検索に対して単一のアドレスを返答する従来のDHTを用いて、一度の検索により複数アドレスを取得する場合、複数名前解決時に宛先個数分繰り返し検索を行う必要があり、検索メッセージ数の増加と応答時間の悪化が避けられない。

３　提案方式

一度の検索で複数アドレスを取得する方法としては、宛先名の組合せから決まるひとつの検索キーに、複数個の宛先蓄積ノードのアドレスをハッシュテーブルに登録する方法が考えられる。この方法は複数検索の効率性の点では優れるが、蓄積ノードのアドレスに変更が生じた場合に、多数の組合せからなる検索キーエントリの中から、更新の必要のある蓄積ノードのアドレス登録情報を探し出すことができない。登録元蓄積ノードのアドレスが変更された場合、複数名前解決要求に対する検索キーの蓄積ノードの登録アドレスが旧アドレスのままであることが原因で、更新が反映される以前の旧アドレス宛に配信されるケースが起こり得る。

本研究では、複数名前解決時の検索効率の向上とアドレス更新性とを両立するために、ハッシュテーブル間に連結検索キーによるリンクを導入する。ここで連結検索キーとは、複数の宛先固有名を一定の順番で並び替えた後連結した語のハッシュ値であり、検索対象の組合せが決まれば一意に決まる。ハッシュテーブルは、単一検索キーに単一の蓄積ノードアドレスを登録した参照元ハッシュテーブルと、単一の連結検索キーに複数の蓄積ノードアドレスを登録した参照先ハッシュテーブルとからなる。図１に示すように、参照先ハッシュテーブルの各エントリは、参照元ハッシュテーブルのエントリに対して、連結検索キーとインデックス番号により１対１のリンク関係を保持する。複数名前解決時は、参照先ハッシュテーブルから該当する連結検索キーを検索する。該当キーが見つかればアドレス値を返答して終了する。該当キーが見つからない場合（当該組合せについて初回解決時）は、いったん個別の検索を必要回数行い、結果を参照先ハッシュテーブルに格納すると同時に、個別検索を行っ

参照元ハッシュテーブル例

検索キー	アドレス	（参照先）連結検索キー		
:	:	:		
1E	1.2.3.4 : 8500	3B	A2	C9
		0	1	1
:	:	:		
91	5.6.7.8 : 8500	A2	C9	
		0	0	
:	:	:		

参照先ハッシュテーブル例

連結検索キー	アドレス	
:	:	
3B	0	1.2.3.4 8500
	1	2.3.4.5 8500
	2	4.5.6.7 8500
:	:	
A2	0	5.6.7.8 8500
	1	1.2.3.4 8500
:	:	
C9	0	5.6.7.8 8500
	1	1.2.3.4 8500
:	:	

【図１：　ハッシュテーブル例】

A study on Multi-name Resolution with Lump Modification
Kimihiro Yamakoshi, Yoshiaki Seki
NTT Information Sharing Platform Laboratories

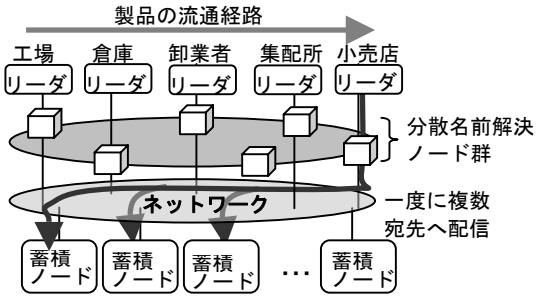
た全ての参照元ハッシュテーブルに参照先ハッシュテーブルの連結検索キーとインデックス番号を登録する。同一の複数組合せに対する２回目以降の名前解決時は、参照先ハッシュテーブルに登録された複数アドレス値を一括取得できるため、単数名前解決時と同程度の応答時間で結果を取得することが可能となる。

また、蓄積ノードのアドレスに変更が生じた場合は、参照元ハッシュテーブルの該当エントリに登録されている参照先連結検索キーと格納順番を示すインデックス番号から、関連する参照先ハッシュテーブルのエントリの登録アドレスを一括更新することが可能となる。このため、単一検索キーに複数アドレスを単純に登録する方式の欠点である、誤アドレス取得を回避することが可能となる。またアドレス変更が発生した時に限って関連エントリの更新を行うため、アドレス更新に要する帯域や処理負荷のオーバヘッドを最小限に抑えることが可能となる。

４　応答時間の考察

提案方式のSCMへの適用例を図２に示す。ここでは、あるひとつの製品の流通に着目した時の経路を切り出して示している。製造元から小売店に至る経路は一般に製品毎に異なるため、実際にはこのような流通経路が全体として多数存在する。

表１に示す条件下で、複数名前解決時の応答時間を、提案方式とハッシュテーブル間リンクを用いない単一アドレス解決方式（従来方式）とで比較した。前提条件として、製造元から小売店に至る経路中の業者数は図２に示すように5とし、工場数100、倉庫数100、卸数100、集配所数1,000、小売店数100,000とした。配送形態として、工場-倉庫間は1対1配送、倉庫-卸間はフルメッシュ配送、卸以下小売店まではTree状に分岐配送される場合を想定し、この時の流通経路数10,000,000をベース（最小値）として、これ以上の3種類と組合せ上限値を考慮した。DHTにChordを想定し、一度の単数解決に要するノード間転送メッセージ数の平均値を $A=(1/2)\log_2 N$



【図２：　物流における分散名前解決】

【表１：　評価パラメータ】

名前解決ノード数（N）	10^3
（検索対象）蓄積ノード数（M）	$\cong 10^5$（注１）
流通経路を形成する業者数（p）	5
流通経路数（s）	$10^7, 10^8, 10^9$
t_0/t_1	10

（注１）工場～小売店までの全拠点数 101300

（N ： ノード数）とし、k 個の複数名前解決に要する応答時間（平均値）を次式で近似した。提案方式は同一組合せに関する２回目以降の検索の場合を示す。

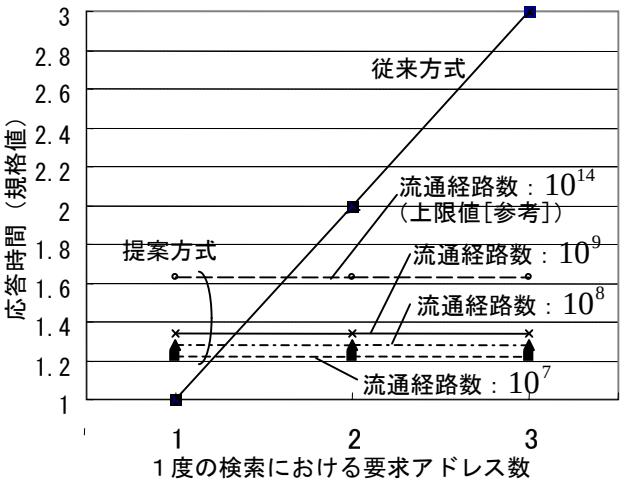
・提案方式：$t_0 A + t_1 \log_2(\mu_1 + \mu_2)$

・従来方式：$k \times (t_0 A + t_1 \log_2 \mu_1)$

ここで、t_0, t_1 を、各々、ノード間のキー検索において検索クエリが１ホップに要する時間、担当ノードが自ハッシュテーブルから検索対象キーを探すのに要するエントリあたりの時間とし、両者の比を10に選んだ。参照元および参照先のハッシュテーブルの名前解決ノードあたりのエントリ数を、各々

$\mu_1 = M/N$　$\mu_2 = p(2^{p-1} - p)s/N$（宛先は流通経路内での任意の組合せを考慮）とし、ニ分木検索によりテーブル検索を行うとした。

図３は、表１に示す条件での複数名前解決の応答時間（従来方式の検索アドレス数１の場合で規格化）を示す。提案方式は上式に示すように流通経路数（s）に依存するが、従来方式はｓに依存しない。提案方式は流通経路数（ハッシュテーブルサイズ）が増大した場合でも、検索アドレス数が２の時33%程度、３の時55%程度応答時間を短縮できる（流通経路数10^9の場合で比較）。



【図３：　複数名前解決時の応答時間】

５　まとめ

Chordをベースに蓄積ノードのアドレス変更をトリガとする登録アドレスの一括更新が可能な複数名前解決方式を提案した。RFIDタグを用いた物流管理における複数宛先配信を想定した場合の応答時間の試算により、複数名前解決時の有効性を明らかにした。

今後の課題として、複数宛先に関する効率的な初期登録方法およびハッシュテーブルメモリや更新管理コストの削減方法などの点が残っている。

【参考文献】

[1] EPCglobal Object Name Service (ONS) 1.0 working draft Apr. 15, 2004

[2] I. Stoica, et al “Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications”, Proc. of SIGCOM, 2001, P. 149 - 160