

大規模コンテキストウェアサービスのための Rete アルゴリズムを用いた地理的コンテンツ配信手法について

川上 朋也¹ 寺西 裕一^{3,1} 春本 要² 下條 真司¹

概要: 本研究では、大規模コンテキストウェアサービスのための地理的コンテンツ配信手法を提案する。提案手法では、コンテンツやセンサデータを地理的な情報に基づいて分散管理しつつ、IF-THEN 形式のルールに従ってコンテキスト導出や有効なコンテンツの配信を行う。提案手法では領域全体を地理的なブロックに分割し、ルールを Rete アルゴリズムに基づいて処理する。その際、対象とするブロック単位で処理を P2P ネットワーク上で分散させる。さらに、Rete アルゴリズムにおける処理を中間結果も含めてすべて分散管理することで、同様の問い合わせに対する処理を削減する。提案手法はシミュレーションにより評価し、処理をネットワーク上の端末に分散し、処理結果を共有することで処理数を削減できることを確認した。

TOMOYA KAWAKAMI¹ YUICHI TERANISHI^{3,1} KANAME HARUMOTO² SHINJI SHIMOJO¹

1. はじめに

近年、PC などの固定端末をはじめ、携帯電話などのモバイル端末や多種多様なセンサなど、さまざまな端末や機器がネットワークで接続されつつある。こうした環境ではモバイル端末などからいつでも・どこでも情報を発信することができ、膨大な量のコンテンツが発生する。ユーザやコンテンツ数が増加するにともない、膨大なユーザ、コンテンツ、ユーザをとりまく環境に関する状況情報（コンテキスト）に基づいて、必要な情報を必要なユーザにのみ提供することが可能な、大規模コンテキストウェアサービスの実現に対する期待が高まっている。

単純なコンテンツ検索の例では、ユーザが地域 A のコンテンツを検索する場合、地域 A に関係するすべてのコンテンツが返却されるのみである。一方、コンテキストに応じたコンテンツ検索では、地域 A のコンテンツを抽出したうえで、各コンテンツの位置周辺に設置されたセンサなどが

ら周辺環境のコンテキストを把握する。例えば「オープンカフェに関するコンテンツは晴れならば推薦する」など、各コンテンツの有効性をシステム側が個別に判断し、ユーザに適するコンテンツを選別することも可能となる。本研究ではコンテキストウェアサービスにおいて重要な位置づけとなる地理的コンテンツ、「地理的コンテンツ」の配信を想定する。想定するコンテンツ配信においては、ユーザは地理的な領域をあらかじめ指定しておき、その領域で有効なコンテンツが Publish/Subscribe (Pub/Sub) モデルで逐次配信される。コンテンツやコンテキストは時間変化によって追加や更新、有効性の変化などがあり、それらの変化はユーザへの配信コンテンツに即時反映されるものとする。

コンテキストに基づくコンテンツ配信においては、ユーザの位置情報やコンテンツの近くに設置されたセンサなどを活用し、適切なコンテンツを選択する必要がある。また、センサデバイスなどから連続的に発生されるデータを解釈した上で意味があるコンテキストを導出し、それに基づいてコンテンツを選択する処理が必要になる。従来の Web に基づく情報システムのアーキテクチャでは、基本的に分散して存在するセンサデバイスなどからのデータをサーバなどに集約し、集中処理する形態が取られる。しかし、セ

¹ 大阪大学サイバーメディアセンター

Cybermedia Center, Osaka University

² 大阪大学大学院工学研究科

Graduate School of Engineering, Osaka University

³ 独立行政法人情報通信研究機構

National Institute of Information and Communication Technology

ンサデバイスから発生する膨大かつ連続的なデータをすべて一つのサーバに集約し、処理することには限界がある。本研究では、こうした背景のもと、連続的なデータの処理をネットワーク上で分散実行することで大規模コンテキストアウトウェアサービスを収容可能とすることを目指す。

センサデータを解釈した上で意味があるコンテキストに変換する方法に関する関連研究は多数あり、多くは IF-THEN 形式のルールに基づいて処理される [1], [2], [3], [4], [5], [6], [7]。本研究でも、IF-THEN 形式のルールに基づいてセンサーデータを処理することを考える。

IF-THEN 形式のルールはプロダクションシステム [8] によって扱うことができる。プロダクションシステムにおける代表的な照合アルゴリズムに Rete アルゴリズム [9] があり、Rete アルゴリズム自体を分散処理する既存研究もある [10]。しかし、多くのユーザが同じ場所に集まっている場合などに同じルールが重複して用いられると、ルール内の特定の処理を担当する端末に負荷が集中してしまう問題がある。また、文献 [10] は Rete アルゴリズムの 1 入力ノードにおけるフィルタリング処理を分散させることを目的としている。そのため、2 入力ノードにおける結合処理は分散および共有されておらず、同一の処理を複数のユーザが個別に行うなどの冗長な処理が実行されてしまう。

そこで本研究では、コンテンツやセンサデータを地理的な情報に基づいて分散管理しつつ、IF-THEN 形式のルールに従ってコンテキスト導出や有効なコンテンツの配信を行う手法を提案する。提案手法では領域全体を地理的なブロックに分割し、ルールを Rete アルゴリズムに基づいて処理する際、対象とするブロック単位で処理を P2P ネットワーク上で分散させる。さらに、Rete アルゴリズムにおける処理を中間結果も含めてすべて分散管理することで、同様の問い合わせに対する処理を削減する。

以下、2 章で IF-THEN 形式のルールを処理する既存技術を紹介する。3 章で提案手法を説明し、4 章でシミュレーションによる評価を行う。最後に 5 章で本稿をまとめる。

2. IF-THEN 形式のルールを処理する既存技術

“IF-THEN” 形式のルールに基づいて推論するプロダクションシステムは、ルールを記憶するプロダクションメモリ (PM)、ワーキングメモリエlement (WME) と呼ばれるデータを記憶するワーキングメモリ (WM)、推論エンジンと呼ばれる制御プログラムから構成される。

プロダクションシステムでは条件照合の高速化に関するさまざまな研究が行われており、Rete アルゴリズムはその代表的な条件照合アルゴリズムである。Rete アルゴリズムは Rete ネットワークと呼ばれるネットワークをルールに基づいて作成し、推論の中間結果を α メモリ、 β メモ

リとして保存する。 α メモリは Rete ネットワークにおいて入力が 1 つの親からのノード (α ノード)、 β メモリは入力が 2 つの親からのノード (β ノード) で、それぞれ条件を満たした WME の組みが保存される。この β ノードでの処理は “join” と呼ばれる。Rete アルゴリズムは α メモリや β メモリを複数のルール間で共有することで処理を削減する。Rete ネットワークは Drools^{*1} や JESS^{*2} など、多くのプロダクションシステム構築ツールやルールエンジンで用いられている。

Rete アルゴリズムを分散環境の Pub/Sub システムへ適用した手法として、DHT (Distributed Hash Table) の一種である Chord [11] を用いた手法が提案されている [10]。文献 [10] では WME の主語、述語、目的語をそれぞれ Key とし、購読者のリストを Value として Chord で分散管理する。すなわち、1 つの WME につき、Key と Value の組みを Chord に 3 つ登録する。また、多用される Key を担当するピアの負荷を分散するため、WME が有効となる時間的な情報 (期間) を Key に付加することを議論している。しかし、すべての WME が時間的な情報を持つとは限らず、時間的な情報の付与および Chord 上での探索について具体的な手法を示していない。また、文献 [10] は類似する複雑な問い合わせを複数のユーザが行う環境を想定しておらず、あくまで Pub/Sub システムにおいて α メモリを分散管理することを目的としている。そのため、Rete アルゴリズムにおける join 処理は複数の α メモリを購読するユーザ自身が行い、処理結果の β メモリは分散管理されない。つまり、共通の join 処理が含まれるルールを複数ユーザが用いる環境では、各ユーザが同様の処理を個別に行うことになり、冗長となる。

3. ルールおよび地理情報に基づく地理的コンテンツ配信手法の提案

3.1 提案手法の概要

本研究では、センサデータを含む地理的コンテンツを複数のコンピュータ (ピア) から構成される P2P ネットワーク上で分散共有するとともに、ルールの処理をピア間で分担することにより、コンテキストアウトウェアサービスを構成するための手法を提案する。センサデータなどを用いたコンテキストの導出には Rete アルゴリズムを用いる。提案手法においては、処理の分散は地理的な情報に基づいて担当ピアを決定することでを行い、各ピアの処理結果を共有することで、同一の領域における同様の処理を別々に行う冗長性を排除し、負荷を削減する。

本研究では、ネットワークに接続する端末を以下に定義する。

- ピア

*1 <http://www.jboss.org/drools/>

*2 <http://www.jessrules.com/>

コンテキストウェアサービスを実現するための情報の共有、処理を行うためのリソースとしてネットワーク接続された物理的コンピュータ。ピア間で DHT (Distributed Hash Table) を構築し、情報の共有やルールに基づく導出処理を分散して行う。担当ピアの探索は 3.3 節で述べる Key によって行い、ルールから生成される Rete ネットワークの各ノードを分担する。

● ユーザ端末

任意の地理的な領域を指定し、コンテンツの配信をピアに依頼する。ユーザが必要とするコンテンツの地理的な領域 (例えば、ユーザの現在位置から半径 3km 以内など) を指定し、領域内にあるコンテンツの配信をピアに依頼する。

● コンテンツ生成端末

店舗に関する口コミ情報など、特定の位置に結び付いたコンテンツを生成する。また、生成コンテンツの管理を P2P ネットワーク上の担当ピアに依頼する。

● センサ

設置された地域の気温や雨量などを計測し、センサーデータを生成する。コンテンツと同様、センサーデータの管理を P2P ネットワーク上の担当ピアに依頼する。

また、コンテンツやセンサーデータを以下に定義する。

● コンテンツ

ユーザに返信する対象となる情報である。本研究では「位置 (x_1, y_1) にあるオープンカフェはケーキがおいしい」など、特定の住所や位置座標を持つことを想定する。また、屋内 (INDOOR)、屋外 (OUTDOOR) などの属性情報を持つ。例えば、建物内にあるレストランは屋内属性、オープンカフェは屋外属性となる。

● センサーデータ

ユーザに返信するのではなく、「領域 *area1* の雨量計の値は 0 ml」など、ユーザやコンテンツ、環境のコンテキストを導出するために用いる。計測値や値が有効となる地理的な範囲、「気温」「雨量」などの属性情報を持つ。

コンテキストを導出するためのルールはあらかじめ作成および共有され、ルールから生成される Rete ネットワークの情報は全ピアが把握しているものとする。ルールの例を図 1 に、図 1 のルールから生成される Rete ネットワークを図 2 に示す。図 1 は「屋内のコンテンツ」もしくは「値が 0 の雨量センサーが近隣に存在する屋外のコンテンツ」に関するルールを擬似コードで記述している。例えば「領域 *req_area* で有効なコンテンツが欲しい」というユーザの場合、領域 *req_area* で「屋内」のコンテンツ、もしくは「屋外かつ近隣の雨量センサーの値が 0」のコンテンツが配信される。

コンテンツが領域 *req_area* やセンサーデータの有効範囲に含まれるかの判別は、3.2 節で述べる地理的なブロック

```

for user u, content c, sensor s
if c.status = 'INDOOR' or
   (c.status = 'OUTDOOR' and
    s.type = 'RAINFALL' and
    s.value = 0)
then
  deliver content c to user u
    
```

図 1 擬似コードによるルールの例

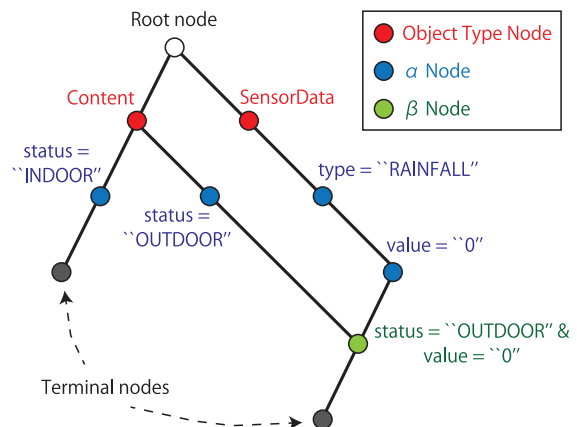


図 2 Rete ネットワークの例

単位で行う。また、ルールを満たすコンテンツの抽出は、図 2 の最上流に「Root node」で示したノードから情報を流すことで行われる。

情報はまず種類に応じて、下流の Object Type ノードと呼ばれる「Content」もしくは「SensorData」へと振り分けて流される。同様に各ノードで示した条件を満たす情報のみをさらに下流のノードへ流すことで、最終的にルールを満たす情報が図 2 の最下流に「Terminal nodes」で示したノードに流される。図 2 の「Terminal nodes」では、左側に「屋内」のコンテンツが、右側に「屋外かつ近隣の雨量センサーの値が 0」のコンテンツが流れる。

3.2 地理的な情報に基づいた分散管理

本研究の想定では、コンテンツは位置情報を、ユーザが指定する領域やセンサーデータは地理的な範囲を持つ。そこで提案手法では、領域全体を複数のブロックに分割し、地理的な点や範囲をブロック単位でピアに分散管理させる。ブロック単位で扱うことにより、分割しない場合に比べて多くのピアに負荷を分散できる。また、無関係な範囲のピアへの問い合わせや情報の収集も削減できる。

提案手法におけるブロックへの分割の例を図 3 に示す。図 3 の各ブロック中央の番号は識別するための ID (ブロック ID) である。図 3 の例では、*sens_data1* は type が「RAINFALL」のセンサーデータである。*sens_data1* の有効範囲 *area1* はブロック「12」、*30* に重なっており、各ブ

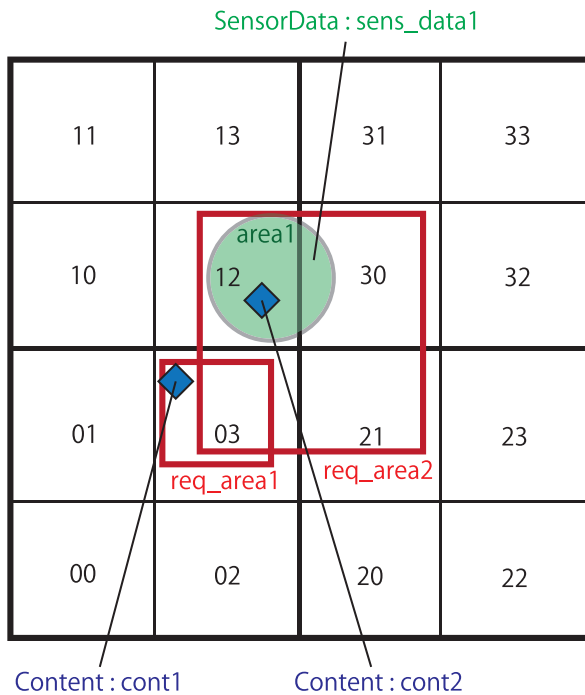


図 3 地理的なブロックへの分割の例

ロックごとに処理する。ユーザの指定領域についても同様に複数のブロックに分割され、コンテンツの場合は単一のブロックで処理する。ブロックへの分割には Distributed Quadtree [12] で用いられている四分木などの空間インデックス手法も適用でき、階層的に分割することでブロックの数や大きさを柔軟に変更できる。

3.3 β メモリの共有

本研究では Rete ネットワークの各ノードに ID を割り当て、各ノードの ID は全ピアが把握しているものとする。そして、ノードの ID および 3.3 節で述べたブロック ID からハッシュ値を計算して Key とし、担当ピアを P2P ネットワーク上で探索する。発見されたピアは当該ブロックかつ当該ノードの処理およびメモリを担当し、当該ブロックの他ノードを担当するピアとの間に Rete ネットワークのリンクを構築することで、必要な情報を収集する。また、P2P ネットワーク上で一度発見された担当ピアは記憶し、Rete ネットワークのノード間は 1 ホップでメッセージを送受信できるようにする。さらに、 α メモリだけでなく β メモリも共有するため、各ノードを担当するピアは処理結果を記憶する。 β メモリも共有することで、本来はさらに上位の 2 ノードに対して必要な処理や問い合わせを削減できる。

図 2 および図 3 の例において、 β ノードも含む Rete ネットワークをブロック単位でピア間に構築する例を図 4 に示す。図 4 の例では、左側がルールから生成される Rete ネットワークを表わしている。複数のブロックに分割された“Content”や“SensorData”などの各ノードは図 2 の

Rete ネットワークに対応している。図 4 の右側は、各ノードおよびブロックから計算されるハッシュ値の Chord における担当ピアを表わしている。提案手法では Rete ネットワークの各ノードをさらに地理的なブロックに分割し、ブロック単位でハッシュ値を計算することで、多くのピアで負荷を分散している。また、Rete ネットワークの各ノードの ID およびブロック ID は全ピアが把握しており、P2P ネットワーク上を探索できることを想定する。そのため、Root node および Terminal node はピアに担当させず、コンテンツ生成端末やユーザは“Content”や最終的な結果を配信する“status=INDOOR”ノードなどに直接依頼する。

提案手法においては、Rete ネットワークのリンクはコンテンツもしくはセンサデータの登録に伴って構築する。図 4 のセンサ“Sensor 1”の例ではまず、“SensorData”ノードのブロック“12”、“30”に対して $h(\text{“30”, “SensorData”})$ のようにハッシュ値をそれぞれ計算し、それらハッシュ値の担当ピアを探索する。発見されたブロック“30”の担当ピア *peer3* は下流の“type = RAINFALL”ノードのブロック“30”に対して、同様に $h(\text{“30”, “type=RAINFALL”})$ のようにハッシュ値を計算し、そのハッシュ値の担当ピアをさらに探索する。この処理を Rete ネットワークの末端ノードまでブロックごとに繰り返す。他のセンサデータやコンテンツでも行うことで、必要な Rete ネットワークのリンクを構築する。図 4 のセンサ“Sensor 1”が含むブロック“30”の場合、図 4 の右側が示すように、以下のハッシュ値と担当ピアの順に情報が流れる。

- (1) $h(\text{“30”, “SensorData”})$: *peer3*
- (2) $h(\text{“30”, “type=RAINFALL”})$: *peer4*
- (3) $h(\text{“30”, “value=0”})$: *peer2*
- (4) $h(\text{“30”, “status=OUTDOOR&value=0”})$: *peer1*

ユーザがコンテンツの配信を依頼する場合は、図 3 の例ではユーザ“User 2”の指定領域 *req_area2* に重なるブロック“03”、“12”、“21”、“30”の 4 つに関して、情報の配信を依頼する。図 4 では依頼する対象は「“INDOOR”」のノードと「“OUTDOOR”かつ“RAINFALL”の値が 0」のノードであり、あらかじめ知っている Rete ネットワークのノード ID およびブロック ID のハッシュ値によって探索し、発見されたピア *peer1* などにコンテンツの配信を依頼する。図 4 のユーザ“User 2”の場合は、 $h(\text{“03”, “status=INDOOR”})$ など、8 つのハッシュ値の担当ピアを探索する。依頼されたピアは、自身まで流れてきたコンテンツのうち、条件に合うコンテンツをユーザ“User 2”へ配信する。ユーザの指定領域が変化した場合には、コンテンツやセンサデータの配信依頼ブロックを逐次変更する。

4. 評価

4.1 評価環境

本研究では、3 章の提案手法をシミュレーションにより

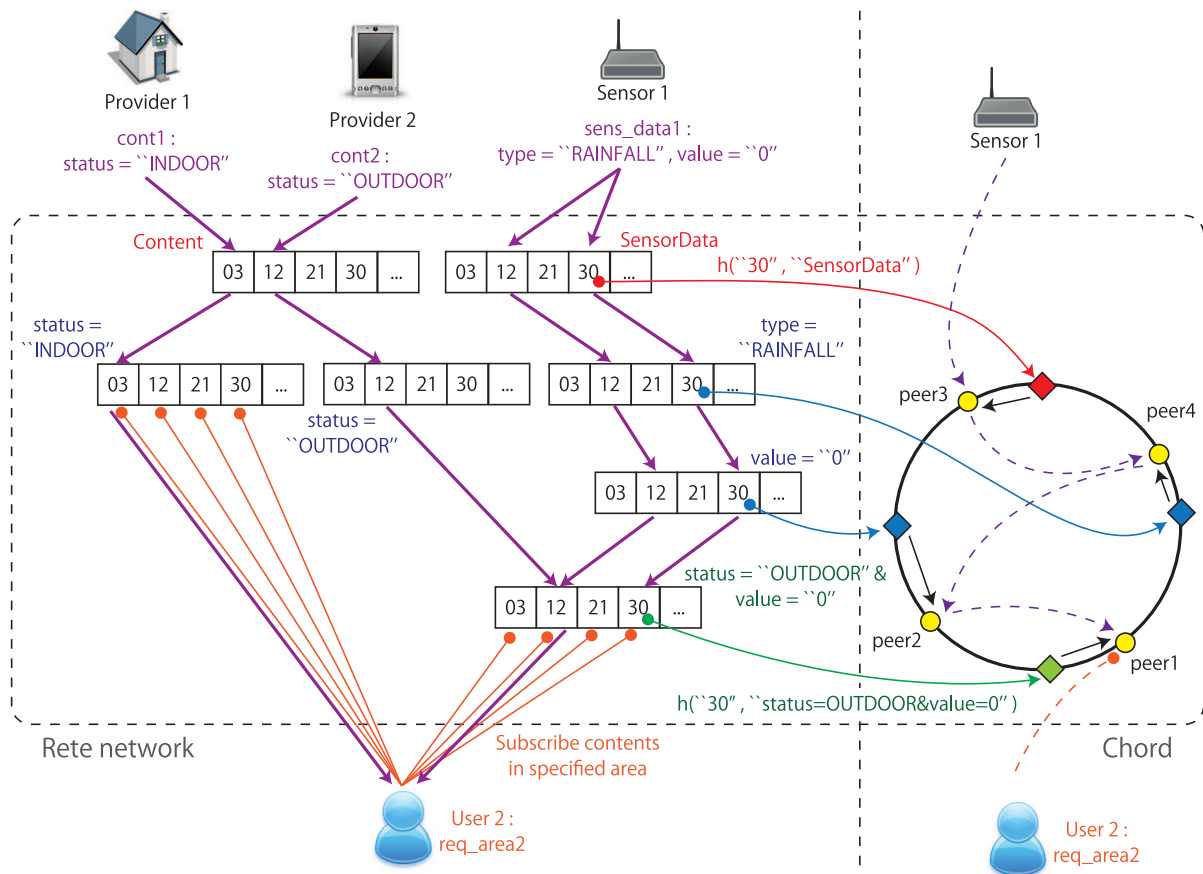


図 4 Rete ネットワークのブロック単位での構築の例

評価した。本シミュレーションのシナリオおよびルールについては、図 2 の例を用いた。

シミュレーション環境を表 1 に示す。また、ユーザの指定領域やコンテンツ、センサの配置の例を図 5 に示す。図 5 では全体領域が $32km \times 32km$ で、円がユーザの指定領域を、薄い点がセンサである雨量計、濃い点がコンテンツを表わしている。コンテンツおよびセンサは密度を $1km^2$ あたり 0.5 個とし、場所は各試行でランダムに配置した。コンテンツには「屋内」「屋外」の 2 種類があり、割合は 50% ずつに設定した。また、センサは半径 $2km$ を計測値の有効範囲とした。図 5 の明るい範囲は「晴れ」である $0ml$ を、それ以外のセンサは「雨」である $10ml$ を計測値としており、本シミュレーションでは割合を 50% ずつに設定した。ユーザがコンテンツを要求する領域は中規模の都市程度の半径 $4km$ とし、本シミュレーションでは 1 ユーザがランダムな場所のコンテンツを要求するとした。さらに、提案手法におけるピア数は 256, 512, 768, 1024 の場合で行い、ピアを探索するためのオーバーレイネットワークには Chord を用いた。また、提案手法のブロックへの分割については、分割しない場合と一辺の長さが $16km, 8km, 4km, 2km$ の固定長のブロックに分割した場合を行った。このとき、全体領域のブロック数はそれぞれ、1, 4, 16, 64, 256 となる。試行はそれぞれの場合で 50 回ずつ行い、

シミュレーションの結果として平均値を算出した。

4.2 ブロックの大きさによる比較

ユーザが最初にコンテンツの領域を指定し、有効なコンテンツを受信する際について、提案手法のブロックの大きさによるメッセージ数（ホップ数）を比較した。比較では、Chord のピア数は 512 で行った。

ブロックの大きさによるメッセージ数を図 6 に示す。図 6 の横軸はブロックの一辺の長さで、縦軸はオーバーレイネットワークで発生したメッセージ数（ホップ数）の合

表 1 シミュレーション環境

全体領域	$32km \times 32km$
ユーザの指定領域	半径 $4km$ の円
ユーザ数	1
コンテンツ数	512
コンテンツの位置	ランダム
コンテンツの種類割合	INDOOR: 50%, OUTDOOR: 50%
センサデータ数	512
センサデータの位置	ランダム
センサデータの有効範囲	半径 $1.5km$ の円
センサデータの値の割合	$0ml$: 50%, $10ml$: 50%
ピア数	256, 512, 768, 1024
オーバーレイネットワーク	Chord
ブロックの一辺の長さ	$16km, 8km, 4km, 2km$

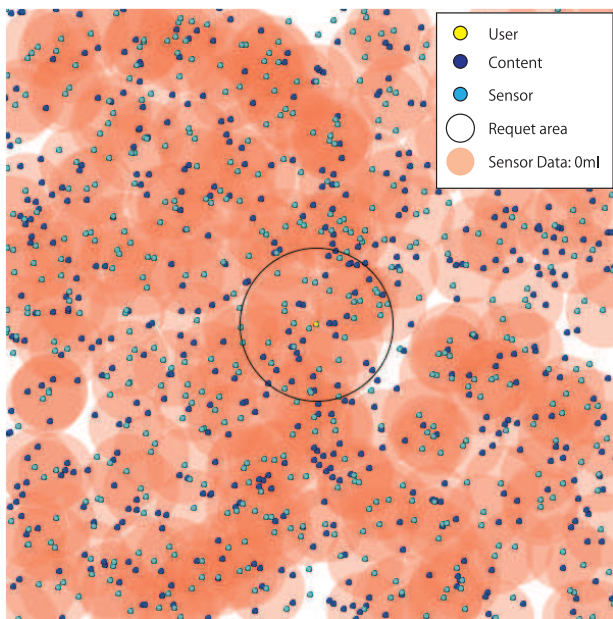


図 5 シミュレーションにおける配置例

計を示している。メッセージ数については、コンテンツに関係する数、センサデータに関係する数、1 ユーザによる Subscribe 処理（コンテンツ配信の依頼）による数ごとに示している。コンテンツおよびセンサデータに関する数には Rete ネットワークのリンクを構築する際のメッセージが含まれ、本シミュレーション環境でユーザからの Subscribe を受け付けるための処理となる。図 6 より、ブロックの一辺の長さを $1/2$ 、つまりブロック数を 4 倍にすると、小さなブロック単位で扱うことで不要な地域への問い合わせは削減するが、ネットワーク全体のメッセージは指数的に増加する。これは、センサデータが有効範囲によって複数のブロックを含み、ブロックが小さくなると含まれる数は大きくなるためである。ユーザの指定領域についても同様にブロック数は大きくなるため、Subscribe 処理のためのメッセージは増加する。一方、コンテンツ自体はブロックが小さくなっても関係するのは 1 ブロックのみであり、コンテンツあたりのブロック数には影響しない。しかし、一度発見したノードかつブロックの担当ピアは 1 ホップでメッセージを送信できる想定であるため、ブロックが細かくなると担当ピアを新たに発見するためのメッセージが増加する。

各ピアのメッセージの偏りについて、ピア数が 512 の場合の結果を図 7 に示す。図 7 の横軸はピア数を、縦軸はメッセージ数を示している。ピアは横軸に従ってメッセージ数が低いものから順に加えてゆき、それまでのピアのメッセージ数を縦軸に示した累積度数である。つまり、グラフが直線に近いほど、ピア全体の偏りが小さい。地理的なブロックに分割しない場合は Rete ネットワークのノードを担当するピアが少なく、コンテンツやセンサデータの配信、処理結果の管理などが特定のピアに集中する。その

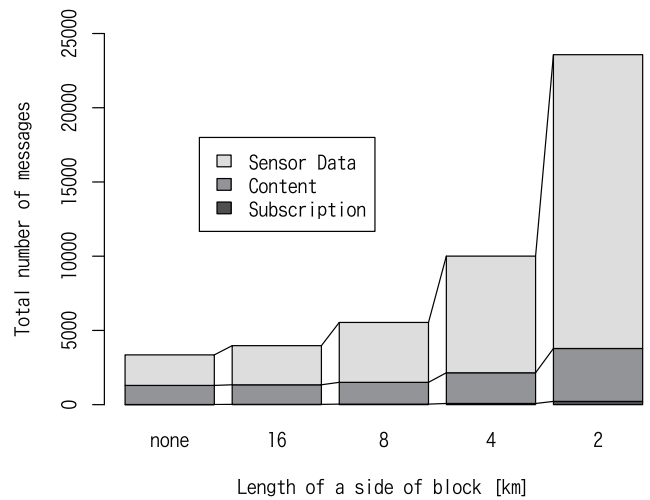


図 6 ブロック数によるメッセージ数の変化

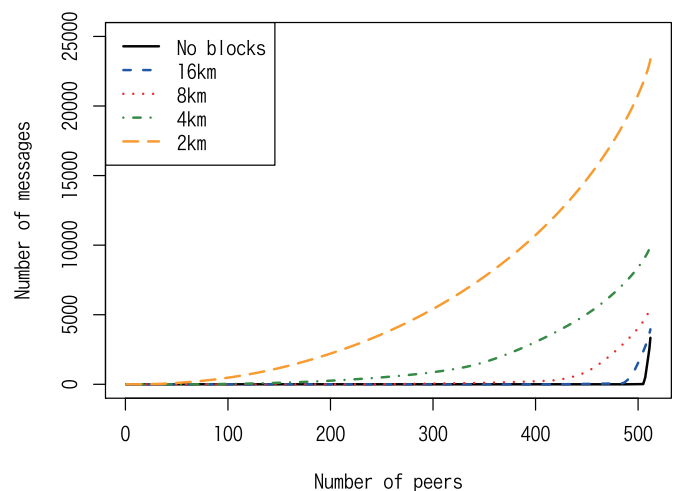


図 7 ピア数によるメッセージの累積度数の変化

ため図 7 では、ブロックに分割しない場合はメッセージ送信が 512 ピアのうちの上位約 5% に集中している。一方、ブロックに分割する場合、一辺が 16km では約 20%、8km では約 60%、4km では約 85%、2km ではほぼすべてのピアがメッセージ送信を行っている。また、ブロックが細かくなると、メッセージ数が大きいピアが全メッセージのうちで占める割合も低くなり、メッセージ送信がネットワーク全体で分散されている。図 6 のようにブロックの分割数を大きくするとメッセージ数が大きくなる一方、細かなブロック単位で扱うことで多くのピアへ負荷を分散できる。

4.3 ピア数による比較

ブロック数による場合と同様に、ピア数によるメッセージ数の比較を図 8 に示す。図 8 の横軸はピア数で、縦軸はネットワーク全体のメッセージ数の合計をコンテンツとセンサデータごとにそれぞれ示している。また、ブロックへの分割は一辺の長さを 4km で行った。

図 8 より、担当ピアを探索する際のメッセージ数が増加するため、ピア数に伴ってメッセージ数が増加する。ただ

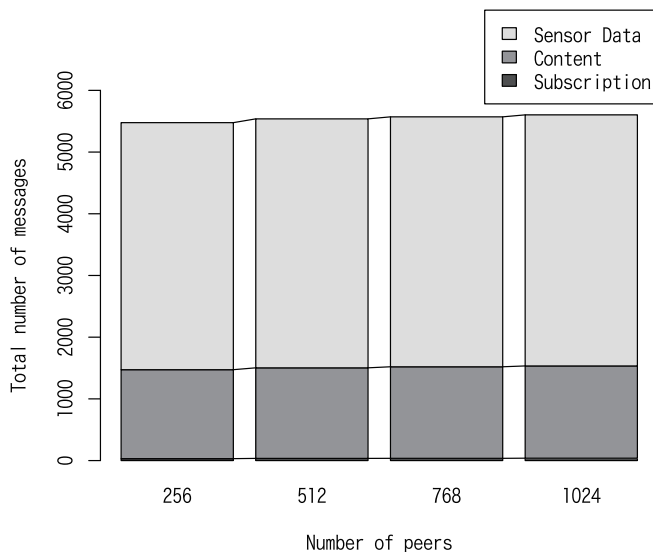


図 8 ピア数によるメッセージ数の変化

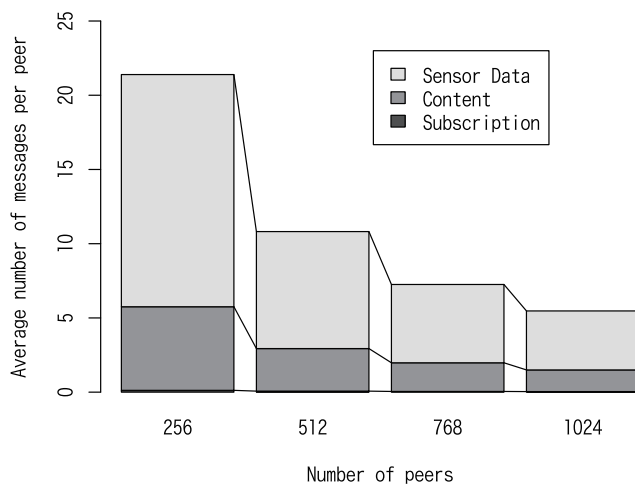


図 9 1 ピアあたりの平均メッセージ数

し、一度探索されたピアは 1 ホップでメッセージを送信するため、増加量の割合はメッセージ数全体のうちで小さい。また、図 8 から 1 ピアあたりの平均メッセージ数を算出した結果を図 9 に示す。ピア数が少ないと一つのピアが多くブロックを担当するため、1 ピアあたりの負荷が大きい。1 ピアあたりの平均メッセージ数は、ピア数にほぼ反比例する。

4.4 ユーザの追加による評価

Rete アルゴリズムでは、ルールによる導出の途中も含めて処理結果を保存し、再利用することで処理を削減する。本シミュレーションでは処理結果の再利用の有効性を評価するため、コンテンツおよびセンサデータを固定した状態でユーザ数を変化させた。本試行では、シミュレーション開始時にコンテンツおよびセンサデータを発生させ、Rete ネットワークを構築する。その後、ユーザを順次追加し、ネットワーク全体における以下の処理数の合計を 10 ユーザごとに比較した。

- メッセージ送信
- メッセージ受信
- α ノードにおけるフィルタリング
- β ノードにおける join

各ユーザについては、半径 4km の円のランダムな領域を指定する。また本試行では、提案手法である「 α ノードおよび β ノードを分散管理する場合」を「 α ノードのみ分散管理する場合」「 α ノードおよび β ノードを分散管理しない場合」と比較した。 α ノードおよび β ノードを分散管理しない場合は図 2 の “Content” および “SensorData” ノードにコンテンツやセンサデータを集約し、ブロック単位で担当ピアに割り当てる。また、ブロックへの分割は一辺の長さが 4km で、ピア数は 512 で行った。

ユーザ数を 0 から 50 まで追加した場合の処理数の累計を図 10 に示す。図 10 の横軸はユーザ数の累計で、縦軸は処理数の累計である。図 10 より、まずコンテンツおよびセンサデータが発生した時点では、Rete ネットワーク内をコンテンツやセンサデータが流れる。このとき、 α ノードや β ノードのブロックを担当するピアでフィルタリングや join 処理が行われるため、 α ノードおよび β ノードを分散管理する場合が最も処理数が多い。一方、 α ノードおよび β ノードを分散管理しない場合は、“Content” もしくは “SensorData” ノードのブロックを担当するピアとのやり取りで送受信されるメッセージのみである。しかし、ユーザを順次追加し、各ユーザが Subscribe 処理を行うにしたがって、累積の処理数は α ノードおよび β ノードを分散管理しない場合が大きく増加する。コンテンツおよびセンサデータの数がそれぞれ 512 である本試行では、 α ノードおよび β ノードを分散管理する提案手法は、 α ノードのみを分散管理する場合をユーザ数が 20 の時点で、 α ノードおよび β ノードを分散管理しない場合をユーザ数が 30 の時点で、累積の処理数がそれぞれ下回る。

それぞれの直線の傾きを示す要素については、まず α ノードおよび β ノードを分散管理する場合は、ユーザと α ノードもしくは β ノード間のメッセージの送受信のみである。一方、 α ノードのみを分散管理する場合は、ユーザと α ノード間のメッセージの送受信に加え、ユーザ側で β ノードの join 処理を行う。さらに、 α ノードおよび β ノードを分散管理しない場合は、ユーザと “Content” および “SensorData” ノード間のメッセージの送受信に加え、ユーザ側で α ノードのフィルタリングと β ノードの join 処理を行う。また、 α ノードおよび β ノードを分散管理する場合と比べ、ユーザは多くのコンテンツやセンサデータを収集している。各ユーザは多数の同様の処理を個別に行っており、冗長となる。

5. まとめ

本研究では、コンテンツやセンサデータを地理的な情

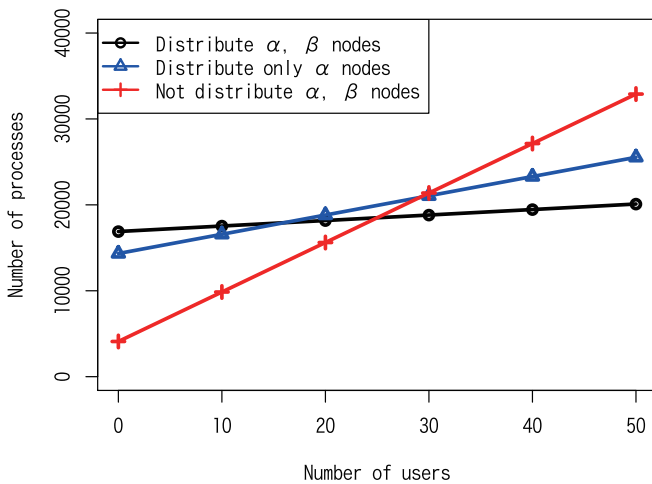


図 10 ユーザ数による処理数の変化

報に基づいて分散管理しつつ、IF-THEN 形式のルールに従ってコンテキスト導出や有効なコンテンツの配信を行う手法を提案した。提案手法では領域全体を地理的なブロックに分割し、ルールを Rete アルゴリズムに基づいて処理する際、対象とするブロック単位で処理を P2P ネットワーク上で分散させる。さらに、Rete アルゴリズムにおける処理を中間結果も含めてすべて分散管理することで、同様の問い合わせに対する処理を削減する。シミュレーションにより提案手法を評価し、地理的なブロック単位で多くのピアに処理を分散でき、処理結果を共有することで全体の処理数を削減できることを確認した。

今後の課題としては、管理や検索で生じる負荷を削減するため、適切な粒度のブロックに分割するアルゴリズムが必要である。最適なブロックの分割数はコンテンツやセンサデータ、ピアの存在密度などによって変化する。実世界では分布が地域によって異なるため、ブロックの大きさを動的に変更することが有効であると考えられる。また、提案手法では、Rete ネットワークのノードや地理的なブロックごとにコンテンツやセンサデータを記憶する。それらデータはネットワーク上のピアに分散管理されるが、全体的なデータ量は増加する。膨大な数のコンテンツやセンサデータを扱う大規模なコンテキストアウェアサービスを実現する上で、データ量のスケーラビリティについても評価する必要がある。さらに、提案手法は異なるルール間においても共通のノードの処理結果を共有する。そのため、多種多様なコンテンツやセンサデータ、ルールが存在し、動的に更新される環境において、提案手法の有効性を評価する必要がある。

謝辞 本研究の一部は、科研費 (22500057) の助成、および、NICT・大阪大学共同研究「大規模分散コンピューティングのための高機能ネットワークプラットフォーム技術の研究開発」による成果である。

参考文献

- [1] Chen, H., Finin, T. and Joshi, A.: An Ontology for Context-Aware Pervasive Computing Environments, *Knowledge Engineering Review*, Vol. 18, No. 3, pp. 197–207 (2003).
- [2] Neto, R. F. B., Macedo, A. A. and Camacho-Guerrero, J.: Configurable Semantic Services Leveraging Applications Context-Aware, *Proceedings of the 11th Brazilian Symposium on Multimedia and the Web (WebMedia)*, Vol. 125, pp. 1–9 (2005).
- [3] Ni, L. M., Zhu, Y., Ma, J., Li, M., Luo, Q., Liu, Y., Cheung, S. C. and Yang, Q.: Semantic Sensor Net: An Extensible Framework, *Proceedings of the 3rd International Conference on Computer Networks and Mobile Computing (ICCNMC)*, pp. 1144–1153 (2005).
- [4] 広田 裕, 川島英之, 佐竹 聡, 梅澤 猛, 今井倫太: セマンティック・センサネットワークの実現に向けた実世界指向メタデータ管理システム MeT の設計, *情報処理学会研究報告: 知能と複雑系*, Vol. 2005, No. 109, pp. 29–36 (2005).
- [5] Guo, B., Satake, S. and Imai, M.: Sixth-Sense: Smart Reasoning for Potential Objects Detection in Smart Sensor Rich Environment, *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT)*, pp. 191–194 (2006).
- [6] Dey, A. K., Sohn, T., Streng, S. and Kodama, J.: iCAP: Interactive Prototyping of Context-Aware Applications, *Proceedings of the 4th International Conference on Pervasive Computing (Pervasive)*, pp. 254–271 (2006).
- [7] Imai, M., Hirota, Y., Satake, S. and Kawashima, H.: Semantic Connection between Everyday Objects and a Sensor Network, *Proceedings of the Semantic Sensor Networks (SSN) Workshop*, pp. 1–13 (2006).
- [8] 小林重信: プロダクションシステム, *情報処理*, Vol. 26, No. 12, pp. 1487–1496 (1985).
- [9] Forgy, C. L.: Rete: A Fast Algorithm for the Many Patterns/Many Objects Match, *Artificial Intelligence*, Vol. 19, No. 1, pp. 17–37 (1982).
- [10] Shvartzshnaider, Y., Ott, M. and Levy, D.: Publish/Subscribe on Top of DHT Using RETE Algorithm, *Proceedings of the 3rd Future Internet Symposium (FIS)*, pp. 20–29 (2010).
- [11] Stoica, I., Morris, R., Liben-Nowell, D., Karger, D. R., Kaashoek, M. F., Dabek, F. and Balakrishnan, H.: Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications, *IEEE/ACM Transactions on Networking (TON)*, Vol. 11, No. 1, pp. 17–32 (2003).
- [12] Tanin, E., Harwood, A. and Samet, H.: Using a Distributed Quadtree Index in Peer-to-Peer Networks, *The International Journal on Very Large Data Bases (VLDB)*, Vol. 16, No. 2, pp. 165–178 (2007).