

大規模 M2M サービスのための複合イベント処理システムと評価

小林 佑嗣[†] 磯山 和彦[†] 喜田 弘司[†]日本電気株式会社 クラウドシステム研究所 〒211-8666 神奈川県川崎市中原区下沼部 1753[†]E-mail: {y-kobayashi@nj, k-iso@bc, kida@da}jp.nec.com[†]

1. はじめに

近年、センサなどのデバイスの情報に基づき、人手を介さずリアルタイムな実世界状況の変化に追従したデバイス制御を行う M2M (Machine to Machine) が注目されている。例えば、車両情報や路上センサ情報などに基づき、走行案内や交通制御をリアルタイムに行うテレマティクスサービスがある。

現行の M2M サービスでは、デバイス情報はサービス毎にクラウドであるのが一般的であるが、将来はオープン化されると予想される。例えば、2013 年 5 月にはトヨタ自動車が“ビッグデータ交通情報サービス”としてプローブ情報の提供を開始するなど、企業がデバイス情報を共有、活用する気運が高まっている。

我々はこのような M2M サービスのプラットフォームとして、リアルタイムな処理が可能な CEP(Complex Event Processing/複合イベント処理)を大規模かつスケラブルに実現する方式を提案し、その効果を示した[1]。これにより M2M サービスの大規模化に伴う(1)デバイス情報量の増大と、(2)サービス数の増大への対応という課題を解決した。

本稿では、提案方式を拡張し、M2M サービスのオープン化に伴って新たに生じると予想される、(3)デバイスの種類の増加に対応する CEP の分散処理方式を提案し、同方式の効果を評価にて示す。

2. CEP

CEP では、実世界の状況をイベントという情報で表す。イベントは属性という key-value 型の情報を複数組み合わせた情報である。例えば、自動車のイベントは CAR{ID=12345, 時速=47, 位置=道路 A} などとなる。“ID”, “時速”, “位置”がそれぞれ属性である。

また、CEP では実行したい処理の内容と処理を行うべき状況を、IF-THEN 形式のルールで定義する。例えば、渋滞の迂回を提案するルールは、「道路 A を走る車両の多くが低速走行していたら(IF), 道路 A に向かう車両に迂回経路を通知する(THEN)」などと定義される。IF 項は複数のイベントに関する条件であり、典型的には、イベントの属性の値に関する条件式とそれらイベントの発生する順序や組み合わせである。

CEP は、受信イベント毎に IF 項を満たすかを確認し、満たす場合は直ちに THEN 項の処理を実行する。

1 節で述べた M2M サービスの 3 つの課題は、CEP においてはそれぞれ(1)膨大なイベント数、(2)膨大なルール数、(3)膨大な属性の種類と置換できる。

3. 従来の CEP の分散処理方式の課題

CEP では全てのイベントと全てのルールとを比較する。(1), (2)の環境で分散処理をスケールさせるには、分散処理を行う各サーバにルールとイベントを分担させる。

方法として、Fig. 1 に示すような、サーバがルールを分担しディスパッチャが各サーバの処理に必要なイベントを振り分ける構成が考えられる。しかし本構成では、ルールの分担によっては、分散処理はスケールしない。

例えば Fig. 2 のように、サーバ 1 は A, B のイベントに関するルール 1 と X, Y のイベントに関するルール 3 を、サーバ 2 は B, C のイベントに関するルール 2 を担当するとすれば、ディスパッチャはイベント B を受信する度にイベントをコピーして両サーバに転送し、それぞれがイベント B を処理しなければならない。

このように複数のサーバで同一のイベントが処理されることを本稿では重複処理と呼ぶ。重複処理はイベントの分担を妨げるためスケラビリティ低下の要因である。

我々はこれまでに、イベントの重複処理の発生を抑えるようルールの分担を決定するルール分配方式を提案し(1), (2)の課題を解決した。従来のルール分配方式は、下記の 3 つのポリシーに従い各ルールの分担を決定する。

- ① サーバが担当するルールの数を均等にする。
- ② 特定の 1 属性をキー属性とし、キー属性の条件式が重複するルールを同じサーバに担当させる。
- ③ サーバが担当するルールのキー属性の条件式の範囲を均等にする。

なお、ディスパッチャはルールのキー属性の条件式に基づき、各サーバの処理に必要なイベントを識別する。

①を満たすとき、各サーバはルールを均等に処理するため(2)が解決する。また②, ③を満たすとき、各サーバはイベントを重複なく均等に処理するため(1)が解決する。

しかし、従来方式は(3)の状況では分散処理をスケールさせることができない。(3)の状況では、サービス提供者は数ある属性を自由に組み合わせ条件式を定義するため、全てのルールが共通の属性の条件式を持つとは限らず、従来方式ではキー属性を選定できない。従って、従来方式では適切なルール分担とイベント転送を実現できなかった。

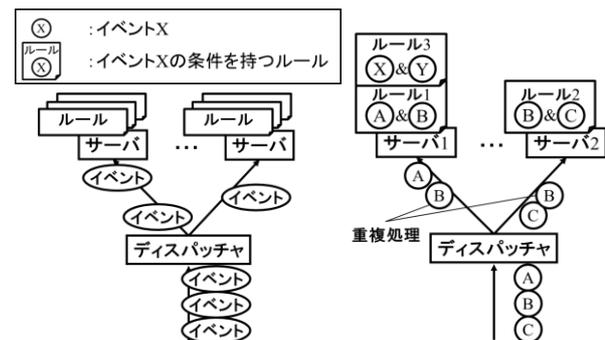


Fig. 1 CEP 分散処理構成

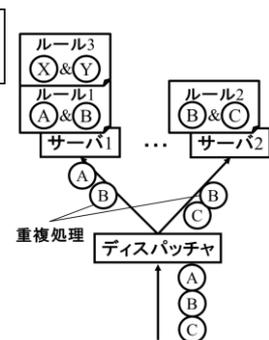


Fig. 2 重複処理

4. ルール分配方式の拡張

4.1. アイデア

従来方式の問題点は、全てのルールが共通の属性の条件を持たない場合に、ルールの処理に必要なイベントを特定できないことである。

しかし、各ルールは共通の属性は持たずとも、各固有の属性を少なくとも1つは持つ。そこで我々は、ルール毎にキー属性を選定し、異なるキー属性に基づいてルール処理に必要なイベントを特定する方式を提案する。

4.2. 複数属性でルールの分担を決める分散処理方式

提案方式では、従来方式のポリシー②を次のように拡張することで従来方式の課題を解決する。

② ルール毎にキー属性を選定し、キー属性の条件が重複するルールを同じサーバに担当させる

従来方式では特定の1属性をキー属性としルールの分担を決定していた。本方式ではルール毎にキー属性を選定し、全てのルールが必ずキー属性の条件を持つようにする。ディスパッチャは、ルール毎に選定された複数のキー属性の値に従い各ルールに必要なイベントを特定し、イベントをサーバに振り分ける。この拡張により3節で述べた課題は解決する。

一方、本拡張により新たにイベントの重複処理が生じてしまう。これは、本拡張にてキー属性が複数選定されることで、イベントが複数のルールのキー属性の値を併せ持つ状況が生じるためである。例えば、サーバ1は”速度”属性をキー属性とするルール1を、サーバ2は”位置”属性をキー属性に持つルール2を担当するとすれば、”速度”属性と”位置”属性の両者の値が各ルールの条件を満たすイベントは、サーバ1とサーバ2で重複処理される。

以上のように、ルール毎にキー属性を選定すると、重複処理が生じるため、分散処理がスケールしない。そこで、提案方式では新たに次のポリシー④を設ける。

④ ルールが他のルールと同一のキー属性の条件を持つ場合には、優先的にその属性をキー属性として選定する

④により、キー属性の選定が必要最小限に抑えられ、イベントの重複処理を抑制できるため、(3)が解決する。

5. 評価

5.1. 目的

提案方式が(3)を解決し、スケーラブルな CEP の分散処理を実現可能であることを示すため、提案方式を適用した分散処理システムのスケーラビリティを評価する。

5.2. 評価方法

指標：CEP の性能指標はスループットとする。スループットとは、システムが秒間に処理するイベントの数である。また、どの程度重複処理が生じたかを確認するため、重複処理数も計測する。重複処理数とは、1 イベントあたりのディスパッチャのイベントのコピー数である。

サーバ数：分散処理システムがスケールすることを確認するため、システムを構成するサーバの数を変化させながら計測する。サーバ数は1, 2, 4, 6, 8と変化させる。

イベントモデル：イベントは、”sensor1” ~ ”sensor4”の4属性を持つとする。また、各イベントは属性毎に1以上1000以下のランダムな整数値をとるとする。

ルールモデル：ルールは、一定時間内に特定の条件を満たす2イベントが発生した場合に通知を行うサービスを想定する。イベントの条件には、4属性のうち1つ以上のランダムな属性について数値の範囲条件を設定する。

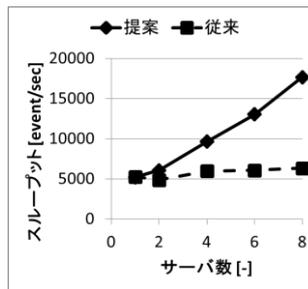


Fig. 3 スループット

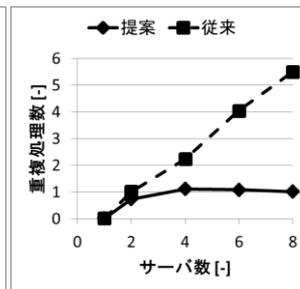


Fig. 4 重複処理数

範囲条件は、各属性につき1以上1000以下で幅50のランダムな範囲とする。また、ルールの数は10000とする。

比較対象：提案方式にてルールの分担を決定する提案システムと、従来方式にてルールの分担を決定する従来システムについてそれぞれ評価を実施し、比較する。なお、従来システムでは”sensor1”をキー属性とする。

5.3. 評価結果

Fig. 3に評価結果を示す。Fig. 3によれば、提案システムでは、サーバ数が増す毎に線形にスループットが向上しており、システムがスケールすることが分かる。一方、従来システムでは、サーバ数が増してもスループットがほぼ向上せず、システムがスケールしないことが分かる。

5.4. 考察

従来システムでは、キー属性である sensor1 の条件を持たないルールを処理するサーバにあらゆるイベントが転送されるため、イベントの重複処理が生じたと予想できる。一方、提案システムでは、ルール毎にキー属性を選定しそれら条件に基づきイベントが転送されるため、イベントの重複処理を抑えられたと予想できる。

Fig. 4に重複処理数を示す。Fig. 4によれば、従来システムでは、サーバ数が8の時に1イベントあたり5イベント以上を重複処理していることが分かる。一方、提案システムでは、サーバ数が8の時でも1イベントあたりの重複処理数は1イベントであり、予想と整合する。

以上より、提案方式が(3)の課題を解決し、スケーラブルな CEP の分散処理に貢献することを示した。

6. おわりに

本稿では、デバイス情報がオープン化された大規模 M2M サービスのための CEP の分散処理方式を提案した。

提案方式では、デバイス情報のオープン化に起因する CEP の分散処理のスケーラビリティ低下問題を、イベントの重複処理を抑制することで解決した。

また、本方式の効果を確認するため、スケーラビリティの評価を実施し、本提案方式がイベントの重複処理の発生を防ぎ、分散処理をスケールできることを示した。

謝辞

本研究の一部は、総務省の「ネットワーク仮想化基盤技術の研究開発」ならびに「ネットワーク仮想化統合技術の研究開発」による委託を受けて実施しています。

O3 プロジェクトでは、上記研究成果を世界中のコミュニティへと公開することによりさらに成果を進化・発展させてまいります。

参考文献

- [1] 磯山和彦ら, ”スケーラブルな複合処理イベント処理システムとその評価,” 信学技報(IEICE Technical Report), Vol.112, No.88, pp.31-36, 2012.