

# ストリーム処理における情報源の動的選択機能の評価

大喜 恒甫<sup>†</sup> 渡辺 陽介<sup>††</sup> 秋山 亮<sup>†</sup> 北川 博之<sup>†,†††</sup> 天笠 俊之<sup>†,†††</sup>  
川島 英之<sup>†,†††</sup>

<sup>†</sup> 筑波大学システム情報工学研究科 〒305-8573 茨城県つくば市天王台 1-1-1

<sup>††</sup> 科学技術振興機構 戦略的創造研究推進事業

<sup>†††</sup> 筑波大学計算科学研究センター

E-mail: <sup>†</sup>{ohki,watanabe,akiyama}@kde.cs.tsukuba.ac.jp, <sup>††</sup>{kitagawa,amagasa,kawasima}@cs.tsukuba.ac.jp

あらまし 近年、ネットワーク環境の発展に伴い、ライブカメラなどによる映像ストリームを容易に取得できるようになった。多数のカメラ映像の中から、利用者の探したい人物が映った映像の提供や、カメラ映像とセンサーデータとの統合等の処理を行うアプリケーションの開発要求が高まっている。このようなアプリケーションを、より効率的に開発できるようにするため、大規模な映像ストリームの統合の基盤となるシステムが求められている。大規模な映像ストリーム環境に対応するために、我々のグループが研究開発しているストリーム処理エンジンをベースに構築した情報を統合して利用できる基盤システムにおいて、情報源への接続を動的に選択する手法を提案した。本論文では、大規模な映像ストリーム環境を仮想的に構築し、提案手法の有用性を評価実験を用いて示す。

キーワード データストリーム、異種情報源結合、情報源の動的選択

## Evaluation of dynamic source selection in stream processing

Kosuke OHKI<sup>†</sup>, Yousuke WATANABE<sup>††</sup>, Ryo AKIYAMA<sup>†</sup>, Hiroyuki KITAGAWA<sup>†,†††</sup>, Toshiyuki  
AMAGASA<sup>†,†††</sup>, and Hideyuki KAWASHIMA<sup>†,†††</sup>

<sup>†</sup> Graduate School of Systems and Information Engineering, University of Tsukuba 1-1-1 Tennoudai,  
Tsukuba-shi, 305-8573 Japan

<sup>††</sup> Japan Science and Technology Agency, Core Research for Evolutional Science and Technology  
(JST/CREST)

<sup>†††</sup> Center for Computational Sciences, University of Tsukuba

E-mail: <sup>†</sup>{ohki,watanabe,akiyama}@kde.cs.tsukuba.ac.jp, <sup>††</sup>{kitagawa,amagasa,kawasima}@cs.tsukuba.ac.jp

**Abstract** The recent development of network technology has made easy to get some video streams. A demand for applications providing video streams requested by users and integrating video streams and sensor data is increasing. Thus we need an integration system for numerous video streams. Our system is based on the stream processing engine which our research group developed. In this paper, we propose a framework to integrate video streams with other streams or databases. We extend our engine to realize video stream integration by adding original data type for video data, and functions to extract metadata from video streams. Moreover, we propose a new query processing operation to select data sources automatically from numerous video streams. To evaluate proposed method, we made fictitious extensive world and made assessment experiment.

**Key words** Stream Processing, Heterogeneous Information Source, Dynamic Source Selection

### 1. ま え が き

近年、デバイス技術やネットワーク技術の普及に伴い、実世界から自発的に配信されるストリームと呼ばれるデータに対す

る高度利用要求に注目が集められている。ストリームの例としては、情報配信サービスや実世界からの温度、映像、音声などのデータがある。これらストリームに対する利用要求としては、フィルタリングや異なる情報源との統合利用などの問合せ処理

が挙げられる。ストリームに対する問合せを実現するシステムとしてストリーム処理システム [1] があり、本研究室においても StreamSpinner[2] を研究・開発中である。

ストリームに対する問合せは情報源からストリームデータが入ってくるたびに連続的に処理されるが、ユーザの興味のある情報源が移り変わる場合がある。例えば、ネットワークカメラが整備された環境において、位置情報を発するセンサを身に付けた特定の人物を追跡したいという要求があるとする。ここでは、登下校中の子供の安全確認のための追跡を考える。全てのカメラに接続し、子供の映っている映像を解析することで、先の要求を満たすことができるが、この方法では子供の映っている映像を解析するためのコストと、常時カメラに接続することによるストリーム処理システムへの映像蓄積のための負荷が大きくなってしまふ。

限られた資源の中で子供の映っている映像を効率的に得るためには、子供に付けられているセンサからの位置情報を用いて、子供の最寄りのカメラからの映像のみを取得することが考えられる。ここで、子供は移動をするので、その位置情報の変化に伴って接続するカメラを動的に切り替えることが必要となる。これを本研究では、情報源の動的選択機能と呼ぶ。情報源の動的選択機能により、子供の近くのカメラのみに接続することで、子供が映っている可能性の高いカメラからの映像を取得することが出来き、これにより映像解析のコストを削減することが出来るとともに、ストリーム処理システムへの負荷を減らすことができる。

本研究では、[3] において、ストリーム処理システムである StreamSpinner をベースに既に構築している映像統合処理システムを大規模な映像ストリーム環境に対応できるように、必要最低限の情報源のみに接続をする動的選択機能を提案した。[3] では、問合せ記述に ASSIGN 節を新しく定義し、ASSIGN 演算において、入力タプルから接続する情報源を動的に選択することができるようにした。提案手法は大規模映像ストリーム環境において、人物追跡はネットワークカメラが充実した環境で、大多数の人の中から追跡対象者の映像を得ることを目的としているが、実際に想定環境を整えるにはハードウェアなどの資源が多く必要となる。そこで、本稿では、大規模な映像統合利用環境を仮想的に構築した。そして、仮想的な大規模映像ストリーム環境における情報源の動的選択機能の有用性を確かめるための評価実験を行った。

本稿の構成は以下の通りである。まず、2 では映像統合処理システムに関して述べる。3 では大規模映像ストリームに対応するために、提案した情報源の動的選択を実現するための問合せ記述と処理の仕組みを述べる。そして、4 でプロトタイプシステムの概要を説明し、5 で仮想的な大規模実験環境の構築について述べ、6 で動的選択機能の有用性を示す評価実験について説明する。7 に関連研究、最後の 8 でまとめと今後の課題について述べる。

## 2. 映像統合処理システム

StreamSpinner を用いた映像統合処理基盤のアーキテクチャ

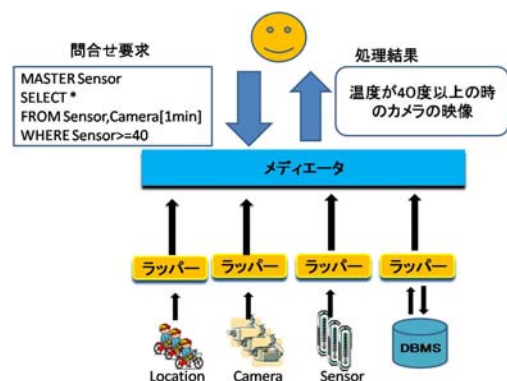


図 1 映像統合処理システムのアーキテクチャ

は図 1 である。ラッパーは、ストリームを受け取るモジュールであり、ストリームからデータが到着すると、受け取ったデータを表形式に変換して、メディエータに送る役割を持つ。カメラからの映像は一定のフレーム数ごとに 1 タプルに変換される。DB ラッパーの場合は、ストリームから配信されたデータの蓄積や、ストリームデータと DB 中のデータの結合などに利用される。例えば、カメラからの映像ストリームの中からユーザが取得したい人物の映像を解析して DB に保存することや、あらかじめ DB に映像データを蓄積しておいて、カメラから配信されてくる映像ストリームと比較して、同一人物であるかどうかの判定などを行うことができる。このような問合せを行うためには、映像データから人物判断や顔認識、特徴抽出などのメタデータを抽出する必要がある。このために、外部ライブラリをユーザが独自に定義できる関数を設けており、映像からメタデータを抽出するための関数を作成することができる。

メディエータは、ラッパーから受け取ったデータに対して、問合せ要求に応じた問合せ処理を行うモジュールである。StreamSpinner では、ストリームと DB の両方を対象とした連続的問合せ記述を提供している。連続的問合せ [4] とは、ストリームからデータが到着するたびに問合せ処理を行い、前回までの処理結果との差分を結果として与える問合せ方式のことである。図 1 の問合せは温度センサからストリームが届くたびに、問合せを繰り返し行い、温度センサの値が 40 度以上の時に温度とともにカメラからの最新の一分間の映像を付けて配信する要求である。FROM 節の [1min] 記述は、ウィンドウ幅を示しており、ウィンドウ幅とは、問合せ処理の対称となる時間の幅のことである。このように異なる情報源を統合し、利用できる点が映像統合処理システムの大きな特徴である。

## 3. 情報源の動的選択機能

ネットワーク上では、複数のライブカメラやセンサーなど、多数のストリームが利用可能である。多くのストリームを同時に扱うような問合せ処理は、相応の計算機資源を必要とする。例えば「ネットワーク上のライブカメラのどれかに、人の姿が映ったら教えて欲しい」という要求では、利用可能なすべての映像ストリームに対する人物検出処理を行わなければならない。このような要求の実現には、複数台の計算機での分散処理

```

MASTER Location
SELECT *
FROM(
SELECT Camera1.Video
FROM Location[1msec],Camera1[1min],CamLoc
WHERE CamLoc.Name = 'Camera1'
AND
dist(CamLoc.X,CamLoc.Y,Location.X,Location.Y)<5
UNION
...
UNION
SELECT CameraN.Video
FROM Location[1msec],CameraN[1min],CamLoc
WHERE CamLoc.Name='CameraN'
AND dist(CamLoc.X,CamLoc.Y,Location.X,Location.Y)<5
)AS AllCamera
WHERE person(AllCamera.Video)='A'

```

図 2 UNION を用いた全カメラ接続を行う問合せ記述

が必須となる。また、全てのカメラを問合せ記述に記述する必要があり、問合せ記述が肥大化する。

一方、別の例として「ある人物Aの近くに設置されているライブカメラに、その人物の姿が映っていたら教えてほしい」という特定人物を追跡する要求を考える。このような要求の具体的なアプリケーションとしては、子供の登下校見守りシステム[6]などが挙げられる。この場合には、すべての映像ストリームを人物検出の対象にするかわりに、人物Aの位置情報に基づいて、近隣のライブカメラの映像ストリームだけを処理対象とすることで、システム内で使用される計算機資源を少なく抑えられる可能性がある。[3]では、そのような処理を実現するために、処理対象とすべきストリームを他のストリームやデータベース中のデータに応じて選択的に切り替える、動的情報源選択機能を導入した。一方、別の例として「ある人物Aの近くに設置されているライブカメラに、その人物の姿が映っていたら教えてほしい」という特定人物を追跡する要求を考える。この場合には、すべての映像ストリームを人物検出の対象にするかわりに、人物Aの位置情報に基づいて、近隣のライブカメラの映像ストリームだけを処理対象とすることで、システム内で使用される計算機資源を少なく抑えられる可能性がある。本研究では、そのような処理を実現するために、処理対象とすべきストリームを他のストリームやデータベース中のデータに応じて選択的に切り替える、動的情報源選択機能を導入した。本機能は次の二つの仕組みで成り立っている。

(1) 入力データに応じて問合せ処理に用いる情報を変えることができる ASSIGN 演算

(2) 情報源への動的接続を行うラッパーを実行中に追加・削除する機能

これらの機能により、子供がある一定距離カメラに近づいたらカメララッパーを作成し、さらにカメラに近づく利用者への配信を ASSIGN 演算で行い、一定距離離れたらカメララッパーを削除することで、ストリーム処理システムへの負荷を少なくすることが出来る。3.1 ですべてのカメラを対象とした場合の問題点について記述し、3.2, 3.3, 3.4 で動的選択機能について述べる。

### 3.1 全カメラに接続する問合せの問題点

通常の SQL ベースの問合せ記述の枠組みにおいては、目的

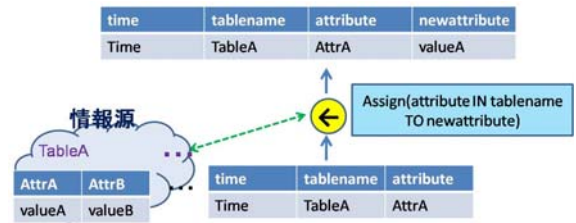


図 3 ASSIGN 演算の動作例

```

MASTER Location
SELECT *
FROM(
SELECT CamLoc.Name,CamLoc.Attribute
FROM Location[1msec],CamLoc
WHERE
dist(CamLoc.X,CamLoc.Y,Location.X,Location.Y)<5
)AS AllCamera
ASSIGN CamLoc.Attribute IN CamLoc.Name TO Video

```

図 4 ASSIGN 節を用いた動的問合せ

とする追跡対象の映像データを得るために、全カメラからの映像ストリームの和集合を指定する必要がある。これに基づく記述は図 2 である。しかし、この方式では、以下のような問題が生じてしまう。

#### (1) 問合せ記述の肥大化

カメラ一台ごとに副問合せにおける位置情報と座標データベースとの統合処理を記述する必要がある。そのため、カメラ台数が膨大になるとそれにあわせて副問合せの記述が長くなってしまふ。また、カメラ台数が変動するたびに、問合せの書き換えが必要になる点も問題である。

#### (2) カメラへの常時接続

この問合せは全カメラを情報源として明確に指定しているため、全カメラからの映像ストリームが利用可能な状態でなければ論理的に正しい動作ができない。よって、常に全カメラに接続した状態となってしまう、人物Aの位置情報に基づいて近隣のカメラの映像ストリームだけを選択的に取得することは困難である。

### 3.2 動的情報源接続

人物の位置情報に基づいて使用すべきライブカメラを特定し、その情報から接続先のカメラを動的に変えることが問合せ記述として指定できるようになれば、上記の問題は解決できるはずである。動的に情報源を選択するための仕組みを説明する。まず、新たに ASSIGN 演算を以下のように定義する。

$$\begin{aligned}
 &ASSIGN_{R, A_i, \{R, A_1, \dots, R, A_j\}}(R) \\
 &= \bigcup_{r \in R} \{r \times t \mid t \in \pi_{attr(r, A_1, \dots, r, A_j)}(rel(r, A_i))\} \quad (1)
 \end{aligned}$$

$rel(r, A_i)$  はタプル  $r$  の属性  $A_i$  の値に対応するリレーションを表し、 $attr(r, A_1 \dots r, A_j)$  はタプル  $r$  の属性  $A_1 \dots A_j$  の値に対応する属性名の集合を表す。これらに対応するリレーションや属性がなかった場合は、タプル  $[r, ]$  を出力するものとする。ASSIGN 演算の処理の具体例を図 3 で表す。

入力タプルの Tablename 属性の値により、情報源中の該当するテーブルを選択し、そのテーブル中の AttrA 属性に含まれ

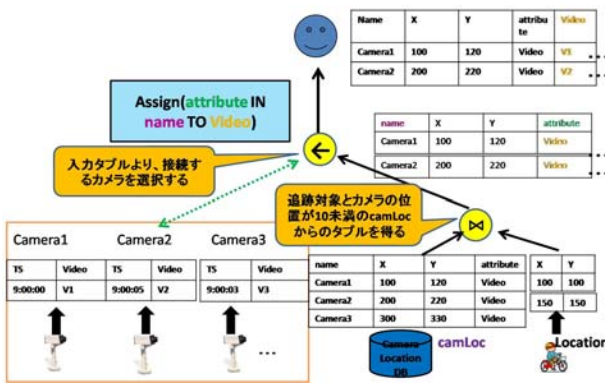


図5 問合せ処理

```

MASTER Location
SELECT create_wrapper(ConfDB.Name,ConfDB.Conf)
FROM Location[now],CamLoc,ConfDB
WHERE
dist(CamLoc.X,CamLoc.Y,Location.X,Location.Y)<10
AND CamLoc.Name = ConfDB.Name

```

図6 ラッパーの動的作成

```

MASTER Location
SELECT delete_wrapper(CamLoc.Name)
FROM Location[now],CamLoc
WHERE
dist(CamLoc.X,CamLoc.Y,Location.X,Location.Y)>=10
AND CamLoc.Name = ConfDB.Name

```

図7 ラッパーの動的削除

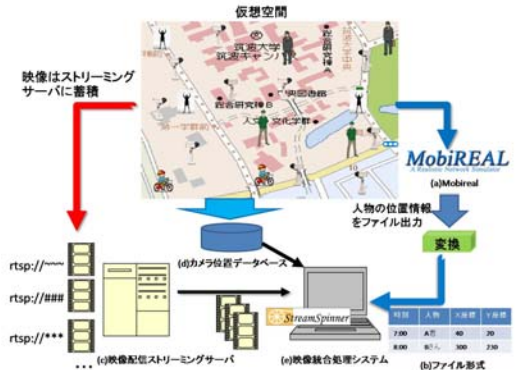


図8 仮想的な大規模映像処理環境

る値を取得する。出力結果の newattribute 属性に取得した値である valueA を格納する。

問合せ記述中では ASSIGN 節は以下のように記述する。

ASSIGN attribute IN table-name TO newattribute

attribute は取得する属性であり、table-name は選択するテーブル名であり、newattribute は出力結果に付加する属性名である。

### 3.3 問合せ処理

位置情報を用いて、子供の近くにいるあるカメラからの映像が欲しいという要求は、ASSIGN 節を用いて図4のように記述できる。複数のカメラの中から子供の位置との距離が5未満のカメラを動的に選択し、その映像を得るという問合せである。

問合せ処理の流れは図5である。あらかじめカメラ位置DB (CamLoc) にカメラ名と映像データの属性名を入れておく。追跡対象の位置情報とカメラの位置情報との距離が一定距離以内であるカメラを結合演算で計算する。その後、ASSIGN 演算にて、カメラの動的選択を行う。

### 3.4 ラッパーの動的作成・削除

ストリーム処理の負荷を減らすことを目的として、必要最低限な情報源に接続するために、ラッパーの動的作成・削除の機能がある。図4の問合せでは、子供との距離が5未満のカメラからの映像を取得することになる。そこで、図6のように、子供とカメラの距離が10未満の時に、あらかじめ create-wrapper 関数にて動的にラッパーを作成する。ConfDB.Name はラッパーの名前であり、ConfDB.Conf はラッパー作成のための設定である。図7は、追跡対象とカメラの距離が10より遠ざかった時に、ラッパーを delete-wrapper で動的に削除する問合せである。

## 4. プロトタイプシステム

StreamSpinner をベースとした映像統合処理システムの

プロトタイプシステムについて説明する。本システムは JavaSDK1.6.0.03 によって実装されている。開発プラットフォームは WindowsXP であり、映像データの変換等に JMF[7](Java Media Frameworks) のクラスを使用している。映像を取得するラッパーには QuickTime for Java[8] を使用して、プログラム上で QuickTime7.1 を起動し、再生された映像を StreamSpinner の内部形式に変換している。現段階では、ネットワーク上の複数のカメラからの映像を同時に受け取り、タプルに変換することが可能となっている。また、ユーザ定義関数として映像から色ヒストグラムなどのメタデータの抽出は可能である。

## 5. 仮想的な大規模映像ストリーム統合環境

実世界で大規模な映像ストリーム環境を整えることは多くの資源が必要となり、難しい問題である。そこで、仮想的な大規模映像ストリーム統合利用のための環境の構築を行った図8である。地図などの2次元の仮想空間上に、多くのカメラを配置し、位置情報を発信するセンサをつけた人が大勢存在する図8のような環境をシミュレートする。位置情報を発信する人の移動には、mobireal[9] というネットワークシミュレータを利用した。mobireal は携帯電話や PDA, MobilePC などの移動端末を持つ多数の人や車の現実的な行動モデルに基づいて大規模なネットワークシミュレーションを行うための理論的な枠組みを提供してくれるものであり、時間とともに移動するノードの位置情報を生成することができる。シミュレート時間や移動端末を持つ人物の数などのパラメータは自由に調整できる。各ノードの位置情報はファイル形式で出力されるが、位置情報ファイルは映像統合処理システム内で処理できるようなファイル形式に変換した。

次に映像ストリームを配信する仮想ネットワークカメラについて述べる。カメラを仮想空間の地図上に配置し、その上で各カメラが設置されている位置情報をあらかじめカメラ位

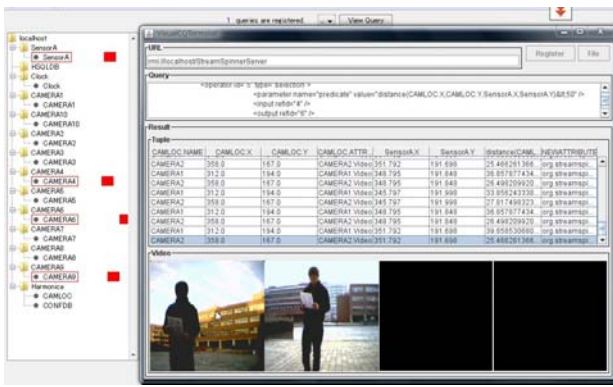


図 9 情報源の動的選択機能を用いた人物追跡

置データベースに登録しておく。仮想空間上の各地点のカメラからの映像はあらかじめ USB カメラで撮影したものを用いる。ストリーミング配信をするストリーミングサーバに各カメラと対応する URL をつけて蓄積し、要求に応じて映像を配信する。ストリーミングサーバ構築のためのソフトウェアは DarwinStreamingServer[10] を用いた。ストリーミング配信のプロトコルは rtsp で、実際の配信方法と同じであり、現実空間に近い形となっている。

映像統合処理システムは mobireal からの人物の位置情報ストリームとストリーミングサーバからの映像ストリームとカメラの位置情報が登録されたデータベースとそれぞれ接続してデータを取得している。ユーザからの問合せ処理をそれらのデータを用いて行う。

大規模映像ストリーム環境のシミュレーションの流れを説明する。シミュレートを開始すると、映像統合処理システムが追跡対象の位置情報ストリームを毎秒取得する。次に取得した位置情報とデータベースに登録されているカメラ位置情報との距離をとり、一定距離内のカメラを見つける。最後にカメラに対応したストリーミングサーバに置かれている映像を取得する。今回は仮想的な環境として、シミュレート時間は 10 分間で、カメラ 100 台を仮想空間上に配置し、それぞれの位置を位置データベースに登録している。移動端末を持つ人数 1000 人を仮想的にシミュレートしており、毎秒 (x,y) 座標の位置情報を配信している。ストリーミングサーバに配置しているカメラに対応した映像は QuickTime 形式であり、解像度は 320 × 240 の約 10 分間の映像である。図 9 は、仮想的な大規模映像ストリーム環境において、動的選択機能を用いた人物追跡のアプリケーションを実行している画面である。

## 6. 評価実験

大規模映像ストリーム環境に適応するために、提案手法が有効であることを証明するための評価実験を仮想的に構築した環境の上で行った。実験は以下の 2 点に関して行った。

- (1) 単独マシンに映像統合処理システムを起動させ、どの程度カメラに常時接続ができるか
- (2) 映像統合処理システムの負荷を減らすために提案した情報源の動的選択機能を用いた場合と、全てのカメラに接続し

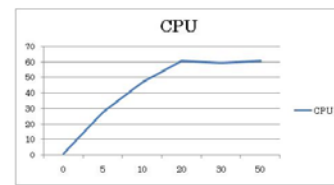


図 10 実験 1:CPU 占有率 (デュアルコア)

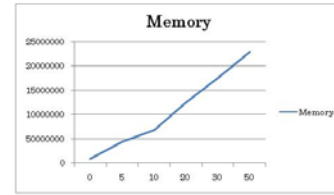


図 11 実験 1:Memory 使用量 (デュアルコア)

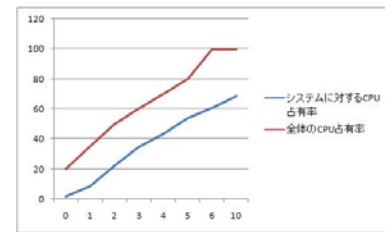


図 12 実験 1:CPU 占有率 (シングルコア)

た場合のシステム負荷について

実験マシンについて、ストリーミングサーバには OS に MacOSX10.4.11,CPU に 1.25 GHz PowerPC G4,Memory に 512MB DDR SDRAM を搭載したマシンとなっている。映像統合処理システムが動作しているマシンは OS に windows vista,CPU に AMD Athlon 64 × 2 Dual Core Processor 3800+2.00GHz,Memory に 2GB である。

### 6.1 実験 1

単独マシン上で徐々にカメラ接続を増やしていった時のシステムが稼働できなくなる限界カメラ接続台数を調べた。java 測定ツールである jconsole[11] を用いて、CPU 占有率と memory 使用率を測定した。実験結果は図 10,11 であり、デュアルコアで行った。横軸がカメラ台数であり、縦軸が cpu 使用率と memory 占有率である。各カメラ台数における cpu 使用率と memory 占有率の値は一分間の平均である。memory に関しては、カメラ台数が増えるに従い、映像ストリームが蓄積されていくので、増加の一途をたどっている。cpu 使用率に関してはカメラ台数が 20 台を超えた付近から変化が見られなくなった。これは、デュアルコアの処理にバス幅からの映像データが追いつけなくなったことが原因であると考えられる。

そこで、シングルコアに切り替えて、同様の実験を行った。実験結果は図 12,13 である。CPU 全体の負荷はカメラを 6 台つないだ時点で 100 % となり、システム稼働の限界であると分かった。

この実験結果より、単独のマシン上では利用できる計算機資

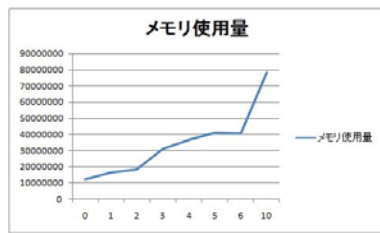


図 13 実験 1:Memory 使用量 (シングルコア)

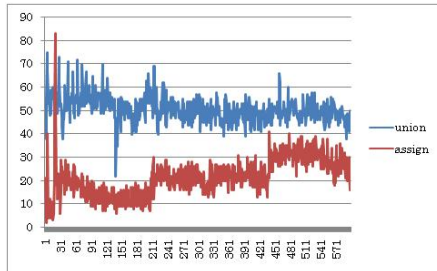


図 14 実験 2:CPU 占有率

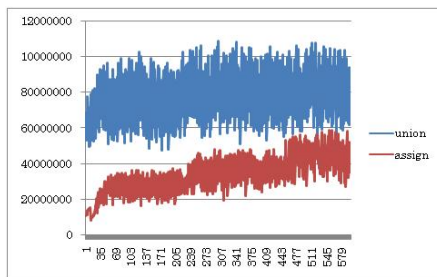


図 15 実験 2:Memory 使用量

源が限られており、多くのカメラを接続してしまうと、映像処理が破たんしてしまうことがわかった。これにより、提案手法で必要最低限の情報源に接続し、そこから必要な情報を動的に選択することが必要であると分かった。

## 6.2 実験 2

大規模映像ストリーム環境を想定し、限られた資源の中で提案手法が有効であることを示す。指標には先ほどと同じように CPU 占有率と Memory 使用量を用いた。実験環境はデュアルコアで行い、カメラは仮想空間上に 10 台配置した。全てのカメラに接続した UNION 演算を用いた問合せと ASSIGN 演算を用いた動的選択の問合せの比較は図 14,15 である。ここで、横軸は時間(秒)、縦軸は memory 使用量と cpu 占有率である。この結果より、提案した ASSIGN 演算を用いた問合せがメモリ使用量、CPU 占有率の双方で優れていることが分かった。

## 7. 関連研究

以下では、他の関連研究と比較した上での、本研究の特徴について述べる。ストリームデータの処理を行う Aurora[1] は、主にセンサー情報などの、数値やテキストデータで得られるストリームを取り扱ったシステムであり、映像等のマルチメディアデータへの対応に関しては、特別言及されているわけではな

い。これに対し、本システムは、映像に特化したデータ型やメタデータ抽出機能を提供することで、様々な映像ストリームと、他の情報源からの情報との統合利用をより容易にするものである。本稿で述べた映像ストリーム統合のアプローチは、これらのシステムにも応用可能である。内山らの提案した DMS[5] は、大規模な映像ストリーム環境における映像監視システムを実現可能としており、本研究と類似している点も多いが、彼らの手法は、データ処理の分散に注目したものであり、実現する為には、分散環境を実現するための膨大なリソースの確保が必要となる。一方、本研究で提案する手法は、映像ストリーム以外の様々なストリームや DB も活用し、接続する情報源を取捨選択することで、限られたリソースの中でも動作することを目的とする。

大規模な映像ストリーム環境を提供するシステムとしては、Argos View[12] などが、最大 10000 台のカメラ映像の統合環境を実現できるシステムとなっている。規模としてはかなり大きなものだが、こちらはカメラから送られてくる映像データのみ注目しており、他のセンサー情報との情報統合などを目指すものではない。これに対し、本研究の提案システムは、映像以外のストリーム、更には DB に保管された情報なども統合することで、多岐にわたるシステム開発の支援が可能となっている。

情報源を動的に切り替える技術に関連するものとしては、FISQL[13] と呼ばれる問合せ言語、およびそれに対応した処理の最適化手法等が提案されている。これは、各 DB のスキーマ情報を問合せ処理の中で参照できるようにすることで、多様なスキーマで保管された複数の DB の情報を一括して扱うことを目的としている。情報源の名称や属性名にあたるデータを変数として参照するという点では、本研究の提案手法と似ている部分があるが、FISQL が対象としているのはデータベースであってストリームではない。本研究における目的は、データを受け取るストリームの動的な切り替えであり、情報源が常に変化していくというところが大きく異なる。

## 8. まとめと今後の課題

本研究では、ストリーム処理システムである StreamSpinner において、情報源の動的選択を可能にし、大規模映像統合利用環境において実験により情報源の動的選択の有用性を示した。現在、単独の PC 上で Streamspinner を動作させ、人物追跡要求を処理しているが、一台では、大規模環境に対応することが難しい。そこで、単独の問合せ要求処理を行うのではなく各ノードで処理の分散化を行うことで、一台のマシンの負荷軽減をすることになり、大規模環境に適応することができると考える。今後は問合せ処理の分散化を行う予定である。

謝辞 本研究の一部は科学研究費補助金特定領域研究 (# 19024006), 科学研究費補助金基盤研究(A) (# 18200005), 科学技術振興機構 CREST「自律連合型基盤システムの構築」による。

## 文献

- [1] D.J.Abadi et al. "Aurora: a new model and architec-

- ture for data stream management ",VLDB Journal Vol.12,No2,pp.120-139,2003
- [2] StreamSpinner,<http://www.streamspinner.org/>
  - [3] 秋山亮, 渡辺陽介, 大喜恒甫, 北川博之, "実世界情報管理基盤システムにおける大規模映像データ統合方式 ",DBWeb2007
  - [4] 渡辺陽介, 北川博之, "連続的問合せに対する複数問合せ最適化手法", 電子情報通信学会論文誌, Vol.J87-DI, No10, pp.873-886, 2004年10月
  - [5] 内山寛之, 赤間浩樹, 西岡秀一, 内藤一兵衛, 長谷川知洋, 谷口展郎, 兵藤正樹, 三浦史光, 山室雅司, 櫻井紀彦, "分散データストリーム処理アーキテクチャの提案" DBWS2007 .
  - [6] ドコモ・システムズ株式会社 . "登下校情報連絡サービス Kids in Feel ", <http://www.docomo-sys.co.jp/products/kidsinfeel/>
  - [7] JMF,<http://java.sun.com/products/java-media/jmf/>
  - [8] QuickTime Player,<http://www.apple.com/jp/quicktime/>
  - [9] mobireal, <http://www.mobireal.net/index-j.html>
  - [10] Darwin Streaming Server, <http://www.apple.com/jp/quicktime/streamingserver>
  - [11] jconsole,<http://java.sun.com/j2se/1.5.0/ja/docs/ja/guide/management/jconsole.html>
  - [12] Argos View Network Camera Manager, <http://argosview.jp/>
  - [13] Catharine M. Wyss, Felix I. Wyss. "Extending Relational Query Optimization to Dynamic Schemas for Information Integration in Multidatabases ", SIGMOD '07, June 12-14, 2007.