



垂直分散コンピューティング技術

ある車両が検知した障害物の情報を周辺車両に高速に通知するために、障害物情報を取得する処理の一部をネットワークエッジにオフロードし、処理途中の中間結果を車両に通知することで、高速な障害物通知を実現しました。しかし、都市部では、大量のコネクティッドカーが一部のネットワークエッジに接続することによるリソース不足が発生し、結果として高速な通知を実現できなくなってしまう。そこで、通知対象のコネクティッドカーの状態に応じて、処理を実行する計算機を動的に変更することで、大量のコネクティッドカーが一部のネットワークエッジに接続する環境においても、高速な障害物通知を実現しました。

まつお 松尾	かずや 和哉	たかぎ 高木	まさる 雅
なかだ 中田	りょうた 亮太	もり 森	こうや 航哉

NTT人間情報研究所

コネクティッドカー向けICT基盤の役割と課題

コネクティッドカー向けICT基盤の役割は、コネクティッドカーから車載カメラ映像やCAN (Controller Area Network) データを収集し、必要に応じて分析した結果を各車両へ送信することによって、クルマ1台のセンサーデータでは把握できない情報をクルマが取得できるようにすることです。これによって、運転手は自分からは死角になっている場所の情報を把握できるため、交通事故を未然に防ぐことが可能になります。

一方で、コネクティッドカーが普及した世界においては、大量のコネクティッドカーからデータが集まってくるため、ICT基盤の処理負荷が高くなるのが考えられます。しかし、そのような状況においても、障害物の情報をはじめとする緊急性の高い情報が検知されれば、その情報を各車両へ高速に通知できなければなりません。

トヨタ自動車との協業では、データ

処理結果をICT基盤からクルマへ通知するユースケースを複数想定して技術検証を行いました。本稿ではその中でも、より高速な通知が必要となる「障害物検知ユースケース」に沿って説明します。

本稿で紹介する垂直分散コンピューティングは、「アプリ垂直分散アーキテクチャ」と「処理ノード動的選択技術」の2つから構成されており、以降ではそれぞれの技術が解いた課題とその解決方法、および実証実験における検証結果について紹介します。

垂直分散コンピューティング

■アプリ垂直分散アーキテクチャ

(1) 従来のICT基盤における障害物検知・通知

トヨタ自動車との協業における従来のICT基盤は、ストリームデータ処理とバッチデータ処理を並列で処理することで、ストリーム処理の即時的な結果と、バッチ処理の詳細な結果の両方を得ることができるラムダアーキテクチャ^{(1),(2)}に従って構築されていました(図1)。また、障害物の検知から通知までを7

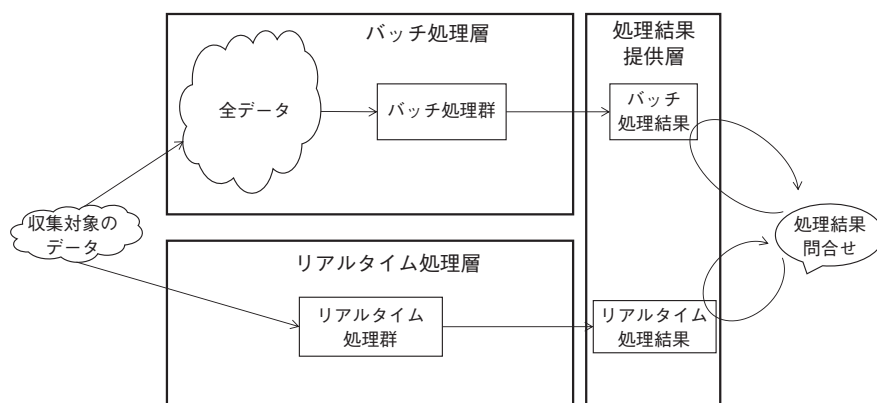


図1 ラムダアーキテクチャ

秒以内に実施するために、ストリームデータ処理として、車載カメラ映像の各フレーム（画像）から障害物の種別と位置を推定する画像データ処理を行っていました。しかし、ラムダアーキテクチャでは処理時間に関する要件は考慮されていないため、このアーキテクチャに従ってシステムを構築するだけでは、7秒以内に障害物検知・通知を実施するという要件を必ずしも満たすことはできず、実際15秒程度の処理時間を要していました。

(2) アプリ垂直分散アーキテクチャの適用

そこで、ラムダアーキテクチャに処理時間のしきい値を設定できるようにした新たなアーキテクチャであるアプリ垂直分散アーキテクチャを提案しました（図2）。アプリ垂直分散アーキテクチャは、一連の処理から合計の処理時間がしきい値を超えないように処理順に処理群を抽出し、その処理群の結果をクライアント側で参照できるようにするアプリケーションアーキテク

チャです。このアーキテクチャに従ってICT基盤を再構築することによって、物体認識と位置推定で構成される画像データ処理から、前者の処理が抽出され、障害物の位置情報を計算する前に障害物の種別のみをコネクティッドカーが参照できる構成となりました。この構成に基づいたICT基盤では、さらなる高速化をめざして、センタサーバよりコネクティッドカーに物理的に近い位置にあるネットワークエッジ*1に物体認識処理をオフロード*2し、ネットワークエッジから障害物の情報を通知できるようにしました。この構成における障害物検知・通知は以下のとおりの2段階になります（図3）。

- ① 初回通知：障害物の種別とその大まかな位置情報（当該障害物を撮影したコネクティッドカーの位置情報を周辺車両へ通知
- ② 詳細通知：位置推定から得た障害物の位置とその種別を周辺車両

*1 ネットワークエッジ：NTTの基地局や局舎に配置した計算機。
 *2 オフロード：あるシステムの負荷を他の機器などが肩代わりして軽減する仕組み。

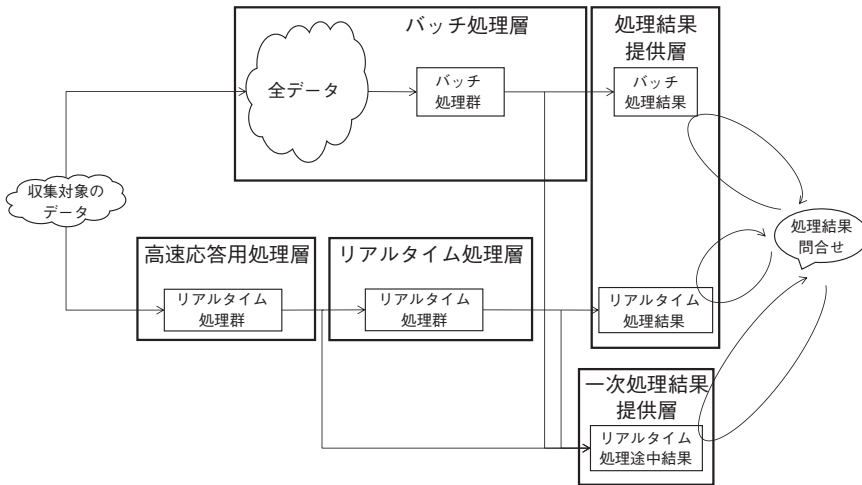


図2 アプリ垂直分散アーキテクチャ

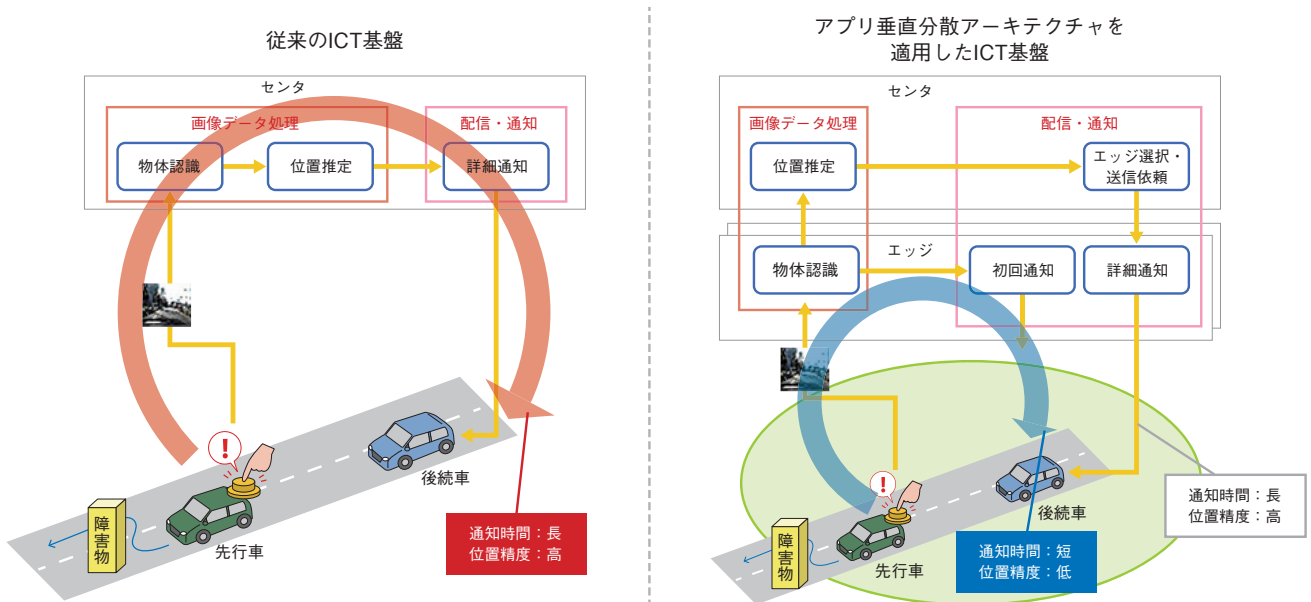


図3 アプリ垂直分散アーキテクチャ適用前後

へ通知

(3) 実証実験

実証実験では、障害物を撮影する車両と障害物情報が通知される車両を同じにして、障害物が車載カメラに写ってからその情報がICT基盤から通知されるまでに要した時間を計測しました。結果を表1に示します。表に示すとおり、初回通知は7秒以内に実施できています。一方で、前述のとおり、この初回通知では障害物の詳細な位置はまだ計算されておらず、その障害物を撮影した車両位置という、本来の障害物の位置から数m~10数m離れた大まかな位置しか分かりません。しかし、1秒で10m以上移動してしまうクルマの世界において、精度を多少犠牲にしても、従来よりも10秒以上早く障害物の情報を通知できることは、安全運転支援の観点から非常に価値があることだと私たちは考えています。

■処理ノード動的選択技術

(1) アプリ垂直分散アーキテクチャ適用後のICT基盤の問題

アプリ垂直分散アーキテクチャによって高速な通知が可能となりましたが、上記検証はネットワークエッジの計算リソースが枯渇しない範囲で計測したものでした。しかしコネクティッドカーが普及した世界における都市部では、大量のコネクティッドカーが一部のネットワークエッジに接続することが考えられるため、当該エッジでの画像処理（物体認識）が集中することでリソース不足が発生し、結果として高速な通知を実現できなくなってしまうことが考えられます。

このような一部の計算機への負荷集中を分散する技術として、システムとして許容可能な処理遅延や計算機の残りリソースを基に、最適な計算機に処理をオフロードする技術が、文献(3)をはじめ数多く提案されています。し

かし、これらの従来技術を適用するだけでは、上記問題は解決しきれません。1秒で10m以上移動してしまうクルマの世界では、その周辺状況は時々刻々と変化するため、その周辺状況によって許容できる処理遅延は異なると考えられます。例えば、見通しの悪い道路で障害物が発生した場合、事故発生のリスクが非常に高いため、周辺車両への高速な通知が必要です。一方、見通しの良い道路で障害物が発生した場合は、事故発生のリスクが低いいため、高速な通知は必ずしも必要ありません。従来技術は前述のようなクライアントの状況を考慮していないため、従来技術を適用してICT基盤の負荷分散を行うと、高速な通知が必要な車両へ高速な通知ができない可能性や、高速通知が不要な車両へ高速な通知をしてしまう可能性があります。

(2) 処理ノード動的選択技術の概要

そこで私たちは、従来の負荷分散技術に、通知対象車両の周辺状況を考慮する機能を加えた処理ノード動的選択技術を提案しました。処理ノード動的

選択技術は、下記3つの処理から構成されます。

- ① 通知対象車両の周辺状況に基づくデータの処理優先度の判定（提案技術）
 - ② オフロード対象処理を実行する計算機を選択（従来技術）
 - ③ 当該処理のオフロード（従来技術）
- 処理①については、ユースケースごとにさまざまなアルゴリズムが考えられます。実証実験では障害物検知・通知ユースケース用のアルゴリズムを採用し、その効果を検証しました。

(3) ICT基盤への処理ノード動的選択技術の適用

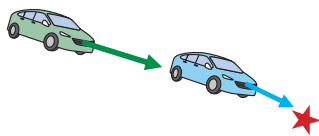
実証実験では「障害物が見えていない高速移動中の車両には高速な通知が必要である」と設定し、それに応じたアルゴリズムを採用しました。具体的には、車載カメラ映像を送信した車両付近のコネクティッドカーを通知対象車両として検索し、これらの車両に高速な通知が必要か否かを下記の2つの条件から判定します（図4）。

① 通知対象車両群の平均移動速度

表1 アプリ垂直分散アーキテクチャの実証実験結果

処理時間			
障害物検知～初回通知		障害物検知～詳細通知	
平均 [s]	最速 [s]	平均 [s]	最速 [s]
4.574	4.092	9.772	9.212

判定条件① 平均移動速度
周辺車両の移動速度がしきい値を超えているか



判定条件② 遮蔽判定
障害物を視認できない車両が存在するか

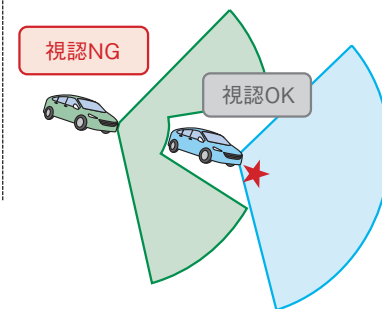


図4 高速な通知が必要な車両の有無の判定

がしきい値を超えているか否か
 ② 運転席から障害物（正確には、障害物検知の場合は障害物を撮影した車両の位置、一度検知した障害物の監視の場合、推定した障害物の位置）が見えているか否か
 なお、判定条件②については、「車両データ選択的収集アルゴリズム」と同様のアルゴリズムで遮蔽判定を計算しています。この2つの判定結果の組合せで、高速な通知が必要な場合は処

理優先度「高」、高速な通知が不要な場合は処理優先度「低」、そのどちらでもない場合は処理優先度「中」として、受信した車載カメラ映像の処理優先度を表2に従って決定します。
 続いて、上記のとおりに優先度を設定した画像処理（物体認識）をどの計算機で実行するかを決定します。処理ノード動的選択技術では、従来技術と同様にシステムを構成する各計算機の残りリソースを監視しており、その残

りリソースと処理優先度を基に、表3の判断基準に基づいて処理を行う計算機を決定します。最後に、選択した計算機に処理をオフロードし、当該計算機で処理を実行します。
 ICT基盤における計算機構成を基に、図5を用いてオフロード対象処理を実行する計算機を選択を例示します。図中の映像1～6の順番にネットワークエッジが車載カメラ映像を受信し、各映像に対する物体認識の優先度が図のとおりに設定されたとします。映像4を処理する物体認識4は、優先度「低」のため、センタサーバへオフロードします。物体認識5は優先度「中」のため、物体認識4のオフロードによりリソースが空いたネットワークエッジで実行します。これによってネットワークエッジの計算リソースがなくなったため、優先度「中」の物体認識6は、センタサーバにオフロードします。

表2 車載カメラ映像の処理優先度設定

①平均移動速度	②遮蔽判定	優先度
高速	遮蔽あり	高
	遮蔽なし	中
低速	遮蔽あり	中
	遮蔽なし	低

表3 優先度に基づく計算機選択

優先度	処理する計算機
高	残りリソースのある最寄りのネットワークエッジ
中	残りリソースのある最寄りの計算機（ネットワークエッジ or センタサーバ）
低	センタサーバ

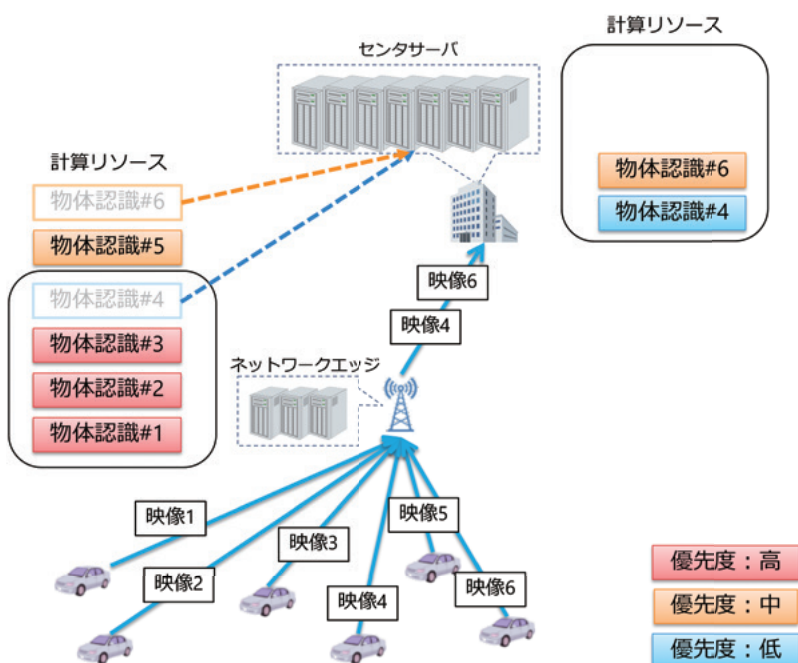


図5 処理ノード動的選択の例

(4) 実証実験
 実証実験では、ネットワークエッジへ送信する車載カメラ映像の数が、画像処理時間に与える影響を確認しました。結果を図6に示します。横軸が1秒当りに受信した映像の数（優先度「高」となる映像を1秒当り1～3個、優先度「中」となる映像を1秒当り1～9個の間で変化）を示し、縦軸は計算リソースが潤沢に使える状況での処理時間からの増加分を示しています。この図から分かるのとおり、処理ノード動的選択がないICT基盤では、毎秒3個以上の映像を受信すると、ネットワークエッジでの処理時間が増加する一方で、毎秒9個*3以上の映像を受

*3 この負荷は、コネクティッドカーが普及した世界を想定するには非常に小さな値ですが、これはトヨタ自動車との協業では各技術の有効性を確認するために最小の計算機構成で実証実験を行っているためです。今後実用化に向けてより大規模な計算機構成でICT基盤を再現した際には、各技術の有効性を示しつつ、より大きな負荷を処理できるようになるでしょう。

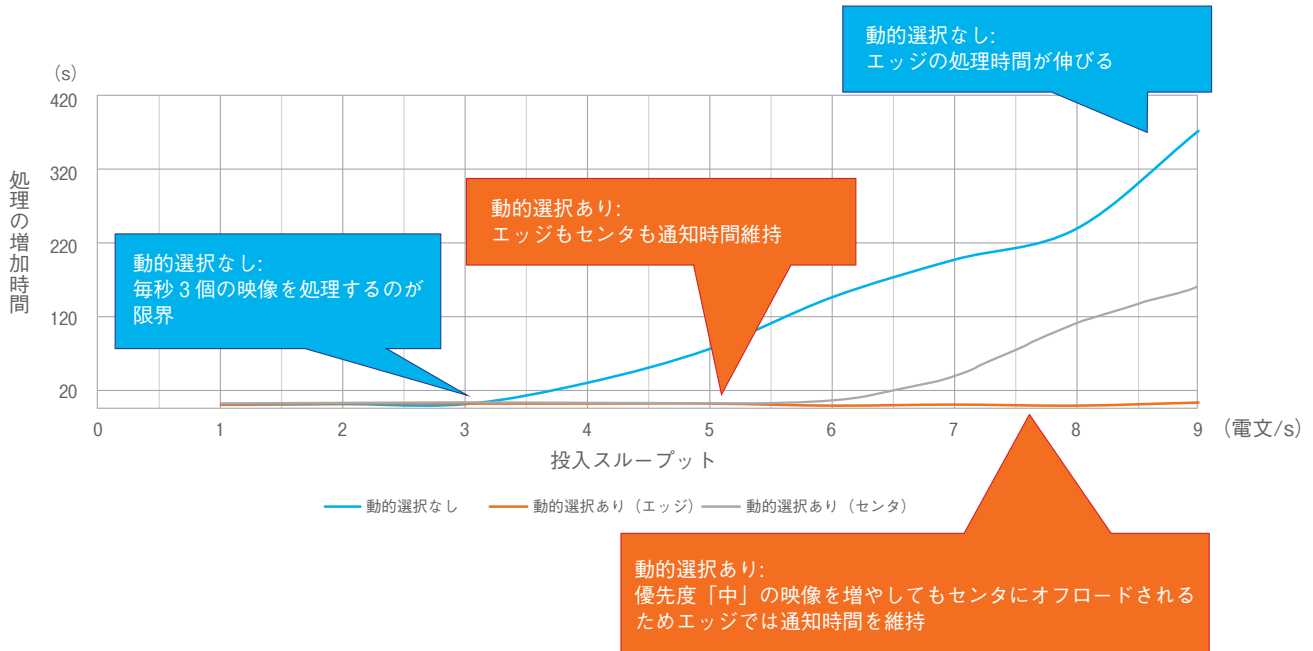


図6 処理ノード動的選択技術の実証実験結果

信するまでネットワークエッジでの処理時間を維持していることが分かります。これは、優先度が高い処理だけをネットワークエッジで処理しており、それ以外の処理をセンタサーバにオフロードしているためです。そのため、毎秒6個以上の映像を受信すると、センタサーバでの画像処理時間が増加しています。

今後の展開

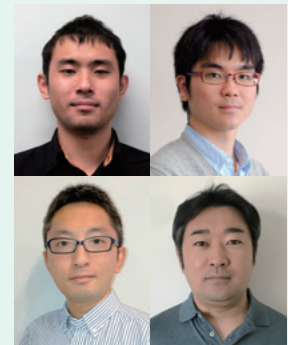
計算機は日々進歩しており、最近ではクルマ自身が車載カメラの映像を処理できるようになってきました。一方で、垂直分散コンピューティングでは、車載カメラ映像などを自身で直接処理できるコネクティッドカーは現状考慮していません。そこで今後は、ネットワークエッジ・センタサーバ側の計算リソースをさらに有効活用を目的として、コネクティッドカーに搭載されている計算リソースも利活用できる拡張を検討する予定です。

また、障害物検知・通知のユースケースでは、あるコネクティッドカー

の車載カメラ映像からシステムが障害物を検知した際に、その障害物が別のクルマのデータからすでに見つけているものなのか、新たに見つかったものなのかを識別する必要があります。すでに見つけている障害物であれば、周辺車両にはすでにその情報が通知されているため、高速な通知が必要ないためです。このように複数の異なるセンサが検知した物体が同一のものか否かを判断する技術のことを同定判定と呼びますが、私たちはこの同定判定を実現する技術についても検討を進めています⁽⁴⁾。

参考文献

- (1) <http://www.databasetube.com/database/big-data-lambda-architecture/>.
- (2) K. Dutta and M. Jayapal: "Big data analytics for real time systems," Big data analytics seminar, pp.1-13, 2015.
- (3) Y. Dai, D. Xu, S. Maharjan, and Y. Zhang: "Joint load balancing and offloading in vehicular edge computing and networks," IEEE Internet of Things Journal, Vol.6, No.3, pp.4377-4387, June 2019.
- (4) 松尾・高木・中田・森: "高精度ダイナミックマップ作成のための点群位置合わせを応用した動的物体検知・同定システムに関する検討," 研究報告ヒューマンコンピュータインタラクション, Vol.2021, No.6, pp.1-8, 2021.



(上段左から) 松尾 和哉/ 高木 雅
(下段左から) 中田 亮太/ 森 航哉

垂直分散コンピューティングは、「高速なレスポンス」と「負荷分散」という観点から、クラウドに集中しがちなデータを地方に分散させ、データの地産地消を実現するための第1歩の技術です。今後は、本技術の実環境への適用に向けた課題解決や技術の拡張に取り組んでいく予定です。

◆問い合わせ先

NTT人間情報研究所
デジタルツインコンピューティング研究センタ
E-mail dtc-office-ml@hco.ntt.co.jp