

# メニーコアを活用するデータフロー型プログラミング言語の開発

## シンギュラリティを目指して

2045年にはコンピュータの能力が人間を超え、技術開発と進化の主役が人間からコンピュータに移る特異点、シンギュラリティに達するとも予測されている。

シンギュラリティに向け、ハードウェアとソフトウェアの両面から並列処理を徹底して高速化する技術が必須である。

計算機ハードウェアは、逐次処理のCPUからマルチスレッド処理を活用するGPUへと発展し、今後はデータフローに基づく分散処理により「ハードウェア量(コア数)」の増加にともない性能がスケール(向上)するメニーコアへと進化する。

一方、計算機ソフトウェアは、高度化・複雑化する人工知能アプリを効率よく表現し、且つメニーコアのスケラブルな並列処理能力を活用するために、どのように進化すべきか。

- 大規模化でスケールするメニーコアへ
- ハードウェアに応じてスケールするソフトウェアへ

## 人工知能システム高速化・高度化の課題

CNN、RNN、AE、Boltzmann Machine等のディープラーニングや、ベイジアンネット、SOM、強化学習などの脳のモデルによる学習時間を短縮する高速化は、人工知能システムの進化にとって極めて重要な課題である。

C/C++, Java, Python等の従来のプログラミング言語は、汎用性が高く、並列性をプログラミングすることもできるが、メモリ集中型の逐次処理が混在しやすく、メニーコア・プロセッサ等の並列処理能力を活用しきれず、スケラブルな高速処理を実現し難い、ソフトウェアの開発・拡張・保守等の生産性が低く多くの工数を要するなど、人工知能システムの高度化に時間を要する。

- 学習時間を短縮する高速処理へ
- AIソフトウェアの生産性向上へ

## AIアルゴリズムに内在する並列性を記述

計算量が極めて多く、膨大なメモリを必要とし、また層間の通信量が多いなど、人工知能の処理には従来のアプリケーションを遥かに超える処理能力を必要とする。

GPUや性能のスケールアップが期待されるメニーコアを用いた機械学習の高速化やソフトウェアの開発効率の向上には、人工知能アルゴリズムに内在する並列性を容易に表現可能なプログラミング環境が必要とされる。

- アルゴリズムに内在する並列性をそのまま記述する

## 明示的なプログラミング・モデルの適用

C/C++, Java, Python等は、シングルプロセッサを使用するために作られた逐次処理を基本とするメモリ集中型のプログラミング・モデルに基づいている。そのため、ループ内の繰り返し処理を複数のスレッドにする、負荷の重い処理をオフロードするなど、並列化が限定的なため高速化し難い。

そこで、アルゴリズムに内在する並列性を保持したままシステムの動作を容易に表現可能なプログラミング・モデルとしてプロセス・ネットワークに基づく「DeepPN」(Deep Process Network)を定義し、これを人工知能アルゴリズムのプログラム表現に適用する。

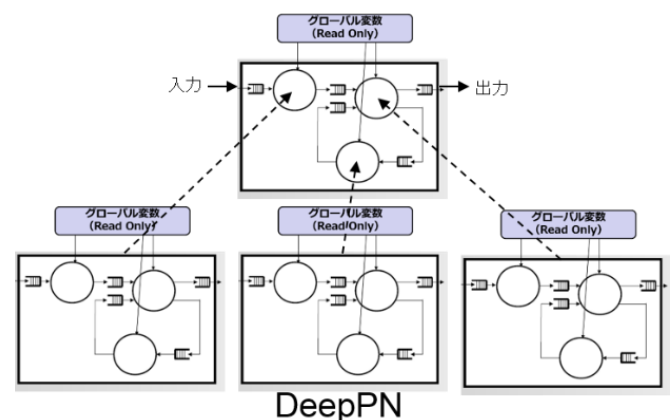
DeepPNプログラミング・モデルの基本的な考え方

- データフロー・グラフでシステム全体を表現する
- システム内の動作のコンカレンシを表現する
- 各プロセスは、ローカルメモリ処理とする
- プロセス間のPoint-to-Pointで通信する
- コンテキスト(グローバル変数)の参照を可能にする

## DeepPN

プリミティブなプロセスは、入力データ、参照データ(グローバル変数)、及び内部状態により出力データを決定する基本モデルで表現する。プログラミング上で、入力データ、参照データ、及び出力データを明示する。入力データは、外部から、あるいは他のプロセスから一方向のチャンネルを通じて受け取り、出力データは外部へ、或いは他のプロセスに一方向のチャンネルを通じて受け渡される。各プロセスは、複数の入力チャンネルと複数の出力チャンネルを持つことができ、各チャンネルには有限の深さのFIFOを内蔵する。

複数のプリミティブなプロセスを統合した1つのプロセスを上位階層で表現することにより、階層的なプロセス・ネットワークを表現する。



# メニーコアを活用するデータフロー型プログラミング言語の開発

## DeepPNを使用するメリット

### ▶ ソフトウェアの「見える化」で、仕様書として扱える

CNN等のニューラルネットワークをそのままグラフとして表現できるため、ソフトウェアの「見える化」により、可読性や保守性が大幅に高まる。

### ▶ 幅広い抽象度で記述できる

シナプスの動作を表現する相変化メモリのようなプリミティブな素子をモデル化して表現する抽象度の低いリーフレベルの表現から、CNN等の複雑なニューラルネットワークを表現する抽象度の高い上位レベルの表現まで、DeepPNを用い統一された方法で、システムをモデル化して記述できる。

### ▶ ツールによる最適化(高速化・低電力化)が容易

従来のソフトウェアは、高速化や低消費電力化のためにターゲットの計算機ハードウェア向けに人手でソースコードを変更する必要があった。

しかし、DeepPNで表現したソフトウェアは、高速化や低消費電力化の要求に従って、ツールを用いてプロセス数、プロセス間通信のチャンネル数、通信チャンネルを介して受け渡すデータの粒度などを、ターゲットの計算機ハードウェアの有する演算器数やメモリ容量に合わせて変更しやすく、最適な実装を機械的に探索できる。

具体的には、以下に示す最適化が可能となる。

#### ◇ スケジューリング

演算リソースに合わせて、プロセスの分割やコアへのマッピングを調整し、各プロセスの実行時間をスケジューリングし最適化することで、システム全体の処理時間の短縮による高速化する

#### ◇ メモリ管理

レジスタファイルやローカルメモリの容量や構成に合わせて階層構造を変更し、中間データの再利用性を高めることで、メモリ使用量を削減し低消費電力化する

#### ◇ グルーピング

サブ・プロセスをマージして、プロセス間で使用するメモリを削減することで、低消費電力化する

#### ◇ データ粒度最適化

プロセスが処理するデータの単位(粒度)をローカルメモリやレジスタファイルの容量に合わせることで、メモリ使用量を削減し、高速化や低消費電力化する

## Software Defined Sensorによる実証

### ▶ エッジに頭脳を埋め込む

### ▶ ネットやクラウド上の膨大な映像データを削減する

トプスシステムズ社は、Software Defined Sensorというコンセプトに基づいて、人工知能を含む画像認識処理をカメラ・モジュールに組み込み可能なエッジ・コンピューティングのプラットフォームを開発し、DeepPN及びSMYLEdeepメニーコアの有効性を実証してまいります。

見守り、自動運転、ロボット、監視カメラなどの用途に応じて、多種多様な機能をソフトウェアで実現する。

### ▶ ヒトの代わりに「観る」「見る」「診る」

ディープラーニング等の先進アルゴリズムによる認識

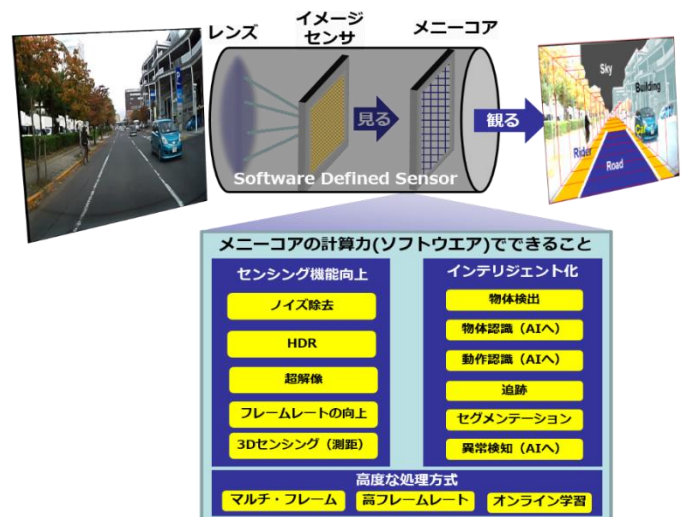
### ▶ 見えないものが観えてくる

高度な計算により、物理的・機械的な限界を越えるセンシングを可能にする

### ▶ 超小型・低消費電力・リアルタイム

カメラ・モジュールに内蔵可能、数msecの高速処理

従来技術では、実現困難な機能や性能は、トプスシステムズ社が独自に開発したスケーラブルな超高速・低消費電力メニーコアSMYLEdeep(エネルギー効率:1TOPS/W)と、DeepPNに基づくスケーラブルで高速化や低消費電力化が容易なソフトウェアにより実現する。



Software Defined Sensorのコンセプト

## 協業によるイノベーションへ

トプスシステムズ社は、協業によりこれらの技術を企業化する事で、社会に貢献する事業パートナーを求めています。

Go Next Generation