

# Android クラスタによる機械学習フレームワークの並列分散処理の検討

星 拓磨<sup>†</sup> 大津 金光<sup>††</sup> 横田 隆史<sup>††</sup>

<sup>†</sup>宇都宮大学工学部情報工学科 <sup>††</sup>宇都宮大学大学院地域創成科学研究科

## 1 はじめに

近年、モバイル端末の普及が世界的に進み、国内においても世帯別スマートフォン保有率が8割を超えた。モバイル端末はマルチコア化、メモリやストレージの大容量化による高性能化も進み、比較的大規模なプログラムの処理も可能になってきている。このモバイル端末を並列処理用の計算資源として有効活用することが期待される。また機械学習の活用は第5世代移动通信システム(5G)をはじめとするデジタル基盤の普及にも相まって、今後のさらなる発展が期待されている。そこで我々はAndroidOSを搭載したモバイル端末をノードとして、スーパーユーザ化を必要とせずに複数ノードによるMPI並列処理が可能なクラスタシステムを実現した。本研究では、機械学習フレームワークの1つであるTensorFlow[1]をクラスタシステム上で並列分散処理する環境の構築に向けて検討する。

## 2 Android クラスタによる機械学習実行環境

本システムはAndroidOSを搭載したモバイル端末を用いて構成する。Wi-Fiルータを用いた無線LANに各ノードを接続しノード間通信を行う。MPI通信を行う場合、ホストノードが1台とその他がクライアントノードとなる。

本研究ではMPI並列処理において複数ノード間で通信を行う際に、Secure Shell(セキュアシェル,SSH)を用いた。SSHは暗号や認証により安全性を考慮したコンピュータ間通信を定めたプロトコルである。Termux上でノード間通信を実行するために、SSHの実装の1つであるOpenSSHを導入した。TermuxではSSHのポート番号がデフォルトで8022に設定されており、本研究でもこの番号を使用した。また、MPI(Message Passing Interface)は分散メモリ型並列計算機において、並列処理を行うために標準化されたプロトコルであり、本研究ではMPIの実装であるOpen MPI[2]を利用した。

本研究では現在主要な機械学習フレームワークの1つであるTensorFlowを導入する。TensorFlowはGoogle社が開発、公開している汎用数値計算ライブラリであり、多次元配列(テンソル)で表されるデータを扱いニューラルネットワークによる深層学習を行うことが出来る。TensorFlowの多くの機能をAPIを呼び出す

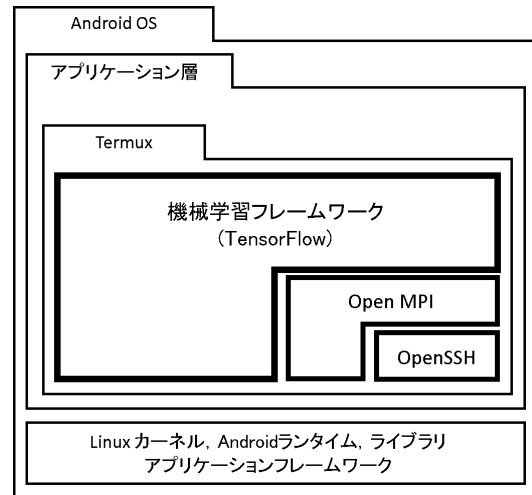


図1: 機械学習に用いるソフトウェアの構成

ことで利用でき、高レベルのAPIを使用することで処理をより簡潔に記述することが出来る。TensorFlowは画像や音声認識、自然言語処理等で幅広く活用され、多くの応用事例があるため、本研究で利用する機械学習フレームワークとして採用した。

先行研究[3]ではMPI並列処理基盤を組み込んだAndroidアプリケーションを開発したが、本研究ではTermux[4]というAndroidアプリケーションを使用することでMPI並列処理を実行した。TermuxはLinux環境を提供する端末エミュレータアプリケーションであり、オープンソースで配布されている。Termuxはスーパーユーザ化無しでファイルの実行やパッケージファイルのインストールが可能である。図1に本システムの構成図を示した。TermuxはAndroidアプリケーションの1つである。本研究ではTermuxのストレージ領域内に、MPI並列処理に用いるOpenSSHとOpen MPIライブラリ、機械学習に用いるTensorFlowを格納して使用する。図1のTermux内において、機械学習をMPI並列分散で処理するためにOpen MPIライブラリを導入した。また、複数ノード間でMPI並列分散処理をするためにOpenSSHを導入した。

TermuxはAndroidアプリケーションとして実行するため、通常のLinuxディストリビューションと構成上異なる点がある。通常のLinuxディストリビューションではディレクトリ構成としてファイルシステム階層標準(Filesystem Hierarchy Standard,FHS)を採用している。通常のディストリビューション向けにビルドされたソフトウェアはFHSのファイル構造を想定して共有ライブラリへのハードリンクを記述してあるため、

Consideration on Parallel Distributed Processing for Machine Learning Framework on Android Clusters

<sup>†</sup>Takuma Hoshi

<sup>††</sup>Kanemitsu Ootsu and Takashi Yokota

Department of Information Science, Faculty of Engineering, Utsunomiya University (<sup>†</sup>)

Graduate School of Regional Development and Creativity, Utsunomiya University (<sup>††</sup>)

基本的には Termux にインストール出来ず、設定ファイル等の修正を施す必要がある。Termux では他のディストリビューションと似た操作が可能なパッケージ管理ツールが提供されており、これを使用することで登録されているソフトウェアを容易にインストール出来る。しかしパッケージ管理ツールに未登録のソフトウェアを使用する場合、手動でビルドする必要がある。

Termux のコンパイル環境について説明する。Termux では GCC(GNU Compiler Collection) が提供されておらず、LLVM コンパイラ基盤と C 言語系向けのコンパイラフロントエンドである Clang が利用できる。また、Android NDK(Native Development Kit) は C, C++ で記述されたソースコードを Android アプリケーション向けにビルドするためのツール群であり、Android NDK を利用して既存のライブラリ等を Android 端末の CPU アーキテクチャ向けにビルドすることが出来る。

### 3 動作検証

Termux 環境で MPI 並列処理を行うプログラムの動作テストを行う。動作テストで用いるプログラムは、ノード間でメッセージパッシングを行う。表 1 に動作テストで使用した Android 端末の仕様を示す。動作テストで使用した 4 台の端末は全て表 1 の仕様である。

表 1: Android 端末の仕様

端末名	HUAWEI MediaPad M3
端末モデル番号	BTV-W09
SoC	Hisilicon Kirin 950 Scores (1.81GHz x4, 2.30GHz x4)
メモリ	LPDDR4 SDRAM 4GiB
OS	Android 7.0
ネットワーク	IEEE 802.11 a/b/g/n/ac

MPI 並列処理の実行時に内部で SSH を利用してノード間通信を行う。動作テストでは N クイーン問題の解の総数を求めるプログラムを実行した。動作テスト実行時の出力を図 2 に示す。図 2 では nqueen.bin というプログラムを 32 プロセスで実行しており、計算結果と実行時間を示した後プログラムが正常に終了していることが分かる。以上から、Android 端末のアプリケーション Termux 上でプログラムを MPI 並列処理で実行した結果が得られ、MPI 並列処理プロセスが複数起動していることが確認できた。

### 4 機械学習並列分散処理の検討

Termux は Linux 環境を提供しており、MPI 並列処理が可能であることが確認出来た。この上で機械学習フレームワークである TensorFlow を動作させる。現在では機械学習におけるモバイルデバイスは主に推論を行うために利用されているが、一般的により高負荷な処理を伴う機械学習での学習においてもモバイルデバイスで処理が可能となると考える。

```
~/works $ mpirun -n 32 -machinefile hostfile-4
p.txt --mca plm_rsh_agent "ssh -p 8022" `pwd`/
nqueen/nqueen.bin 16
qn24b MPI version 1.0.0 2004-06-16
There are 31 workers.
There are 9844 sub problems
=====
qn24b MPI version 1.0.0 2004-06-16
problem size n      : 16
total solutions     : 14772512
correct solutions   : 14772512
million solutions/sec : 4.569
elapsed time (sec)  : 3.233
=====
```

図 2: MPI プログラム実行時の出力画面

TensorFlow は複数の言語向けに API を提供するが、TensorFlow は Python のパッケージとして提供されているため、ビルド済みパッケージをインストールする場合もソースからビルドする場合も開発環境に Python が導入されている必要がある。Termux へ導入可能なソフトウェアパッケージは最新のバージョンのみ対応しているため、本稿執筆時点で Python と TensorFlow の依存関係が未解決である。また、機械学習は多量のデータを扱うため処理に時間がかかる。本研究では TensorFlow における学習の実行時間を MPI 並列処理によって短縮することを検討する。

### 5 おわりに

本稿では Android 端末を用いて構築したクラスタシステム上において、端末エミュレータの Termux 上で MPI 並列処理を実現し、動作テストを行った。また、クラスタシステム上で TensorFlow を動作させる検討を行った。Termux へ導入可能な Python と TensorFlow の依存関係を解決し、TensorFlow を動作させ、MPI 並列処理で実行することが今後の課題である。

謝辞

本研究は、一部 JSPS 科研費 20K11726 の援助による。

### 参考文献

- [1] Martín Abadi, et al.: “TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems”, <https://www.tensorflow.org>, 2015.
- [2] Open MPI: Open Source High Performance Computing: <https://www.open-mpi.org/>, 閲覧 01/05/2021.
- [3] M.Nissato, H.Sugiyama, K.Ootsu, T.Ohkawa, T.Yokota: “Realization and Preliminary Evaluation of MPI Runtime Environment on Android Cluster”, Proc. 33rd International Conference on Advanced Information Networking and Applications, AINA-2019, pp.407–418, 2019.
- [4] Termux: [https://wiki.termux.com/wiki/Main\\_Page](https://wiki.termux.com/wiki/Main_Page), 閲覧 01/05/2021.