

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

宇都宮大学大学院地域創生科学研究科 工農総合科学専攻 情報電気電子システム工学プログラム

1. 研究背景と目的

研究背景

- モバイル端末の普及と高性能化
 - マルチコアプロセッサ, GPUコア, 記憶領域の大容量化, 5G → モバイル端末を計算資源として活用したい
- Androidクラスタシステム: Android OS 搭載機器をノードとしたクラスタ計算機 並列分散処理機能を実装^{[1],[2]}
- 用途: エッジコンピューティング向けプラットフォーム, 災害等非常緊急時の代替サーバ

研究目的

機械学習を行う処理基盤をAndroidクラスタシステムに構築
動作性能の評価と検証を行う

- 機械学習フレームワークとしてTensorFlowを利用

課題

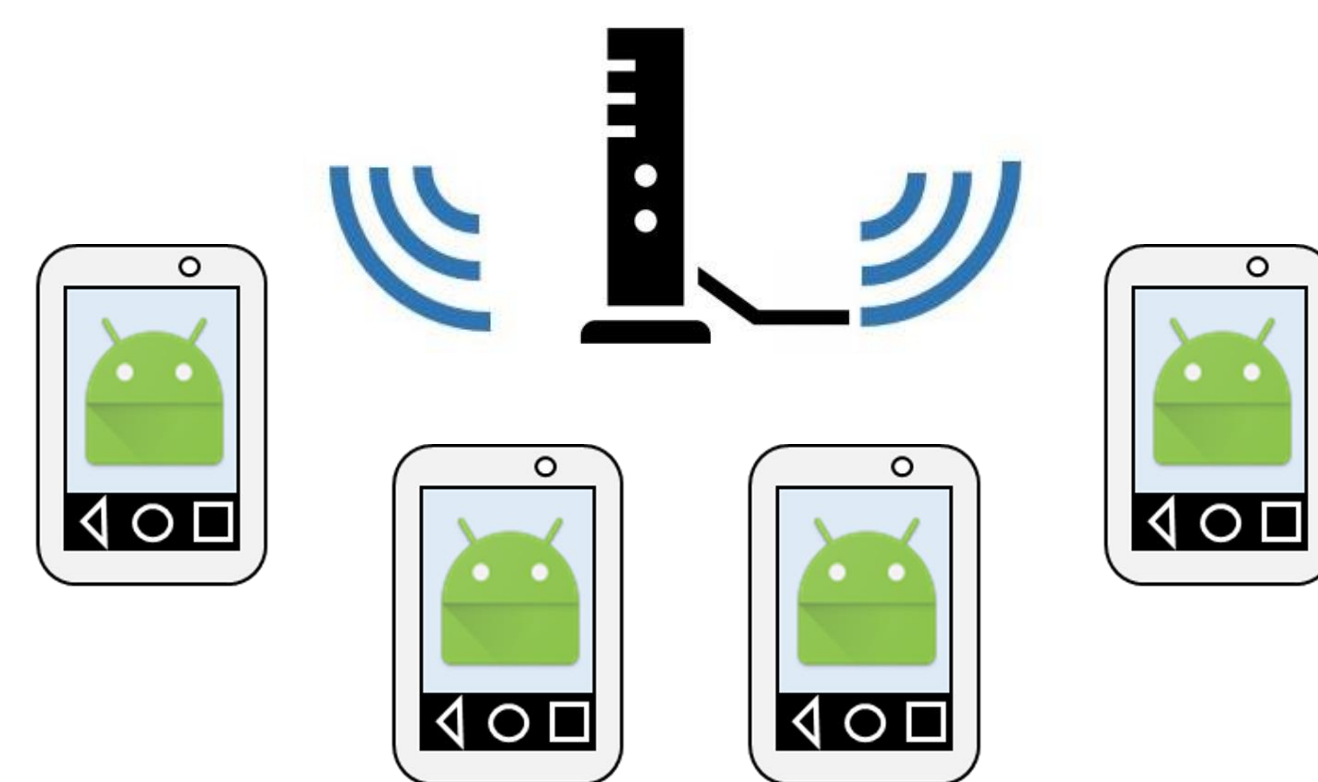
- 開発されたAndroidクラスタシステムではTensorFlowをそのまま使用することが出来ない (以下が必要)
 - TensorFlowを実行するLinux環境, Pythonインタプリタ
 - Android端末のアーキテクチャに対応するTensorFlowバイナリ

AndroidクラスタシステムにTensorFlowの実行環境を構築

*エッジコンピューティング... 中枢のサーバと端末が物理的に近距離に配置すること

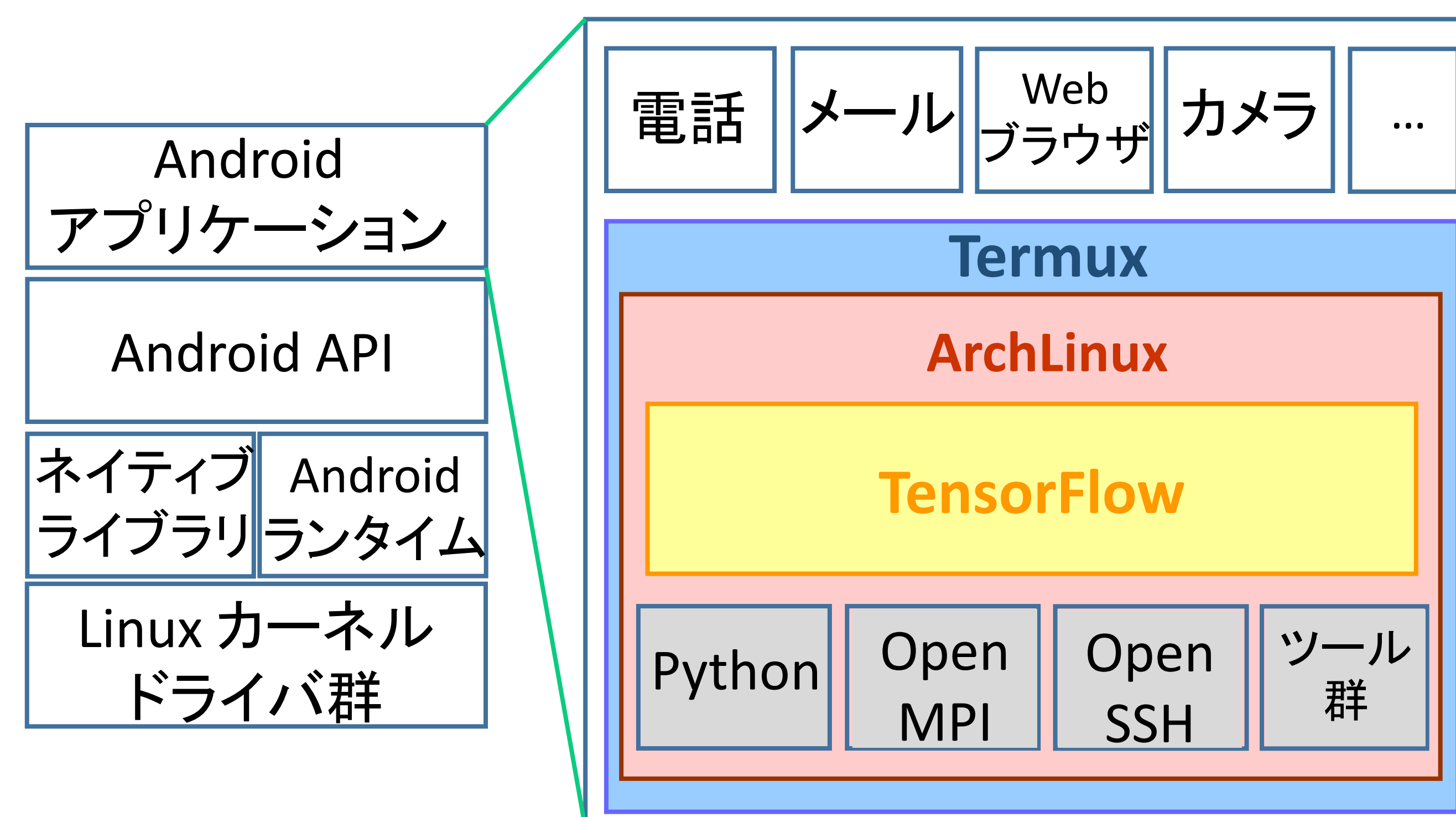
[1] Yuki Sawada, Yusuke Arai, Kanemitsu Ootsu, Takashi Yokota, Takeshi Ohkawa, "Performance of Android Cluster System Allowing Dynamic Node Reconfiguration," Wireless Personal Communication, Vol.93, Issue 4, pp.1067-1087, Feb. 2017.

2. Androidクラスタと構成する実行環境



Androidクラスタの構成例

- 本研究が想定するTensorFlow実行環境の全体構成図



[2] Masahiro Nissato, Hiroki Sugiyama, Kanemitsu Ootsu, Takeshi Ohkawa, Takashi 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.

3. TensorFlow実行環境の構築と各構成要素

Termux

Linux疑似環境を提供する端末エミュレータアプリケーション

- スーパーユーザ権限不要でLinuxシステムを操作可能
- パッケージマネージャからTermux向けにコンパイルされたソフトウェアを導入可能

Arch Linux

- Linuxディストリビューションの一つ
 - ARMアーキテクチャ版が存在
 - OSのバイナリサイズが比較的小さく, 少ないリソースに適す
- Termuxに導入し使用することで, 現状ではTermuxより豊富なアーカイブから必要な版のソフトウェアを導入可能
- ソフトウェアのビルド等ではツール間の依存関係を満たす必要がある

TensorFlow

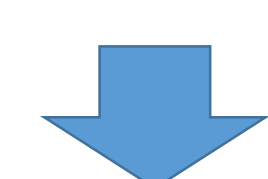
- 機械学習フレームワーク, ライブラリの1つ
 - データからアルゴリズムを学習し, 作成したモデルを実行するための実装
 - PythonのライブラリとしてTensorFlowの機能を利用することを想定

実行環境の構築手順

Termuxの導入



Arch Linuxの導入



TensorFlowのビルド・導入

- TensorFlowはAndroid OS向けバイナリが公式に公開されていない
→ ビルドする必要がある

4. TensorFlowのビルド方法の検討

TensorFlowのビルドが未完

以下の方法でビルドを行うが成功せず

TermuxのArch Linuxでネイティブ環境でのビルド

- 実行するAndroid端末上でのビルド
- ビルドツールの Bazel を使用してビルド
- Android端末単体で行うため実行時間が大幅にかかる
- 設定, オプション指定が煩雑

Raspberry Pi 4 でのビルド

- 32bitARMマシンなので生成バイナリをAndroid端末で使用可
- dockerによる仮想コンテナ環境でビルド → 失敗

X86_64, Arch Linuxでのクロスビルド

- クロスコンパイラを使用して実行
- ネイティブビルドの際と同様にビルド失敗を重ねる

QEMUでaarch64エミュレート環境でのビルド

- プロセッサシミュレータのQEMUを用いてビルド環境を構築
- CPUのみ使用で動作遅 → 作業中

その他検討中の方法

- ビルドツールを変更して実行 Bazel → CMake

5. 現状と今後の計画

- TensorFlowのビルドが未完
- TensorFlowのビルド方法を確立させる
- 実装した環境でのTensorFlowの実行性能を評価, 検証
- Termuxによるオーバーヘッドの影響を調査