

異なる OS が混在する環境での MPI 並列実行方式の検討

寺島 祐作[†] 大津 金光[†] 横田 隆史[†] 小島 駿[†][†]宇都宮大学工学部情報工学科

1 はじめに

近年、モバイル端末の普及は急速に進み、スマートフォンにおいては国内の個人の保有率が7割近くにも及んでいる。また、性能も飛躍的に向上しており、プロセッサのマルチコア化や、メモリ、ストレージの大容量化も進んでいる。この普及率や高性能化を鑑み、個人が保有するモバイル端末を並列処理における計算資源として活用することが期待できる。我々は、Android OS を搭載したモバイル端末をノードとし、管理者権限を必要とせずに MPI 並列処理を行うクラスタシステムを開発している [1]。このクラスタシステムでは、Android OS を搭載した端末同士で MPI 並列処理を行う。しかし、計算機に搭載される OS は Android OS のみではない。また、モバイル端末周辺に存在する計算機を資源として活用するために、クラスタシステムに組み込むことも考えられる。そこで本稿では、異なる OS に対応する、クラスタシステムに周辺の計算機を組み込むという目的から、Android OS を搭載した端末をホストノードとし、種々の OS を搭載した計算機をリモートノードとした場合の MPI 並列実行方式について検討を行う。

2 Android クラスタシステム

本システムでは、Android OS を搭載したモバイル端末を使用する。ホストノードとなる端末を中心に、その他の端末がリモートノードとなり、MPI 並列処理を行う。ノード間通信には、有線通信または Wi-Fi ルータを用いた無線 LAN を利用する。また、MPI 並列処理を行う際の各ノードへのリモートログインは、OpenSSH を使用する。

MPI 並列処理は、Open MPI ライブラリを利用して行う。Open MPI ライブラリは、主要な UNIX 系 OS をサポートしており [2]、Android OS はサポートされていない。そのため、Android OS 上で Open MPI ライブラリを使用するために、先行研究 [1] では、Android OS 上で実行可能な機械語バイナリプログラムを作成するための、Android Native Development Kit (Android NDK) を用いて Open MPI ライブラリのビルドを行った。本稿では、この Android OS 用にビルドされた Open MPI ライブラリをモバイル端末上で使用する。

3 Android 端末上での MPI 並列実行

Android 端末上での MPI 並列処理の実行は、先行研究 [1] で開発した MPI 並列処理アプリケーション (以下、並列処理アプリ) を用いる。

各モバイル端末に並列処理アプリを導入すると、初回起動時にファイル展開とパーミッションの設定が行われる。ここで展開されるファイルは、OpenSSH や、ビルド済み Open MPI、MPI プログラムなどである。並列処理アプリは、最初に秘密鍵や公開鍵の登録と、ファイルパーミッションの設定を行い、SSH によるリモートログインを行う。この時、同時にノード情報収集デーモンも起動する。このノード情報収集デーモンは、各ノード間でノード情報の送受信を行い、ホストファイルとノード情報ファイルを自動生成する。ホストファイルは、各ノードのホスト名 (IP アドレス) と、CPU コア数を記述したファイルで、Open MPI を実行する際に利用する。ノード情報ファイルは、各ノードの IP アドレスとログイン時に指定するログインユーザ名を記述したファイルで、SSH でのリモートログインを行う際に使用する。

SSH によるリモートログイン後、並列処理アプリ上から MPI プログラムを起動する。MPI プログラムを起動したモバイル端末がホストノードとなり、その他のモバイル端末がリモートノードとなる。MPI プログラムの実行が完了する、またはエラーが発生し実行が中断された場合、ホストノードに実行結果もしくはエラーの内容が集約される。

4 異なる OS が混在する環境での MPI 並列実行

本稿では、ホストノードに Android OS を搭載したモバイル端末を、リモートノードに、Linux PC を用いる。Linux を選択した理由は、Android OS は Linux カーネルを使用した OS であるため、コマンドの記述方法やシェルスクリプトが実行可能など類似部分が多く、並列処理アプリから MPI 並列プログラムの実行できる可能性が高いからである。

本稿における各ノードのネットワークは、無線 LAN によって構成する。Linux PC、モバイル端末共に同じ Wi-Fi アクセスポイントに接続し、同一ネットワーク上で互いに通信できる。

Open MPI で並列処理を行うためのリモートログインに必要な SSH の公開鍵は、リモートノード上に登録する必要がある。ここで、公開鍵の登録とは、並列処理アプリのアプリケーション ID を *mpi.app* とすると、公開鍵が記述されたファイルをモバイル端末内の */data/data/mpi.app/files/.ssh* に配置することを指す。並列処理アプリでは、この公開鍵の登録を、並

Consideration of MPI parallel processing environment on heterogeneous operating systems

[†]Yusaku Terashima, [†]Kanemitsu Ootsu, [†]Takashi Yokota and, [†]Shun Kojima,

Department of Information Science, Faculty of Engineering, Utsunomiya University (†)

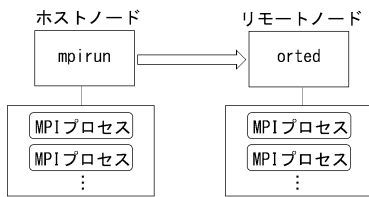


図 1: mpirun と orted の関係

列処理アプリに公開鍵が記述されたファイルを同梱し、そのファイルを SSH でのリモートログイン前に `/data/data/mpl.app/files/.ssh` ディレクトリにコピーすることで行うが、Linux PC に対しては行われない。理由は、並列処理アプリは自身が導入されているモバイル端末のみに対して公開鍵の登録を行い、他ノードに対しては公開鍵の登録は行わないからである。そのため、Linux PC 上では手動で公開鍵の登録を行った。並列処理アプリに、Linux PC に対して公開鍵の登録を自動的に行う機能を実装することは一つの課題である。

モバイル端末上で MPI 並列プログラムを実行すると、並列処理アプリが Open MPI の `mpirun` コマンドを実行する。図 1 に、`mpirun` コマンド実行からの `mpirun` と `orted` の関係を示す。`mpirun` が、ホストノードでは MPI プログラムの実行やプロセスの管理を、リモートノードに対しては SSH を利用してプロセス起動や管理を行うデーモンの `orted` を起動する。ここでは、モバイル端末で実行されている `mpirun` が、Linux PC の `orted` を起動する場合を想定する。

`mpirun` がリモートノードにプロセスを生成する際に、SSH を使用してプロセス起動を行うが、ここで機種毎に固有の処理を追加できるようにするため、SSH ラッパースクリプトを介して SSH が呼び出されるように変更している。この SSH ラッパースクリプトは、ノード情報が記述されたファイルを参照し、`mpirun` 実行開始時に指定されたオプションの内容を引数として受け取り、そのオプションを元に SSH でリモートログインを行うシェルスクリプトである。リモートログインに成功すると、MPI 並列プログラムを起動する。MPI 並列プログラムの実行が完了する、もしくはエラーによって実行が中断された場合、ホストノードに実行結果またはエラーが表示される。

動作検証の際、Linux PC に一時ファイルが作成できないことが原因のエラーが発生した。`mpirun` のオプションとして、一時ファイルの作成場所を Android OS 内で一時ファイルが作成可能なディレクトリとなるよう指定しているものがある。SSH ラッパースクリプトは、`mpirun` のオプションの内容を受け取り、リモートノードに対しても同じオプションを設定するため、リモートノードにもホストノードと同じディレクトリに一時ファイルを作成するように指定する。しかし、Android OS と Linux PC では一時ファイルが作成可能なディレクトリが異なるため、一時ファイルの作成ができない。そのため、本稿では、SSH ラッパース

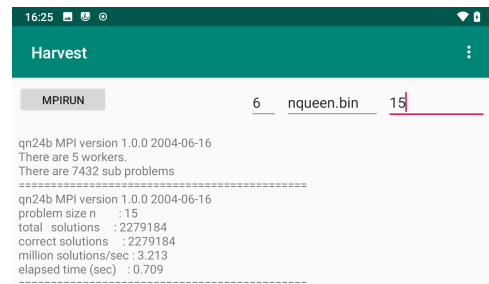


図 2: 実行結果のスクリーンショットの一部

クリプト内で、Android OS 用に指定された一時ファイルの作成場所を、OS 毎に一時ファイル作成ディレクトリを切り替えるようにする。

本稿では、MPI 並列プログラムとして、N クイーンプログラム [3] の実行を行う。図 2 に、実行結果が表示されるモバイル端末のスクリーンショットの一部分を示す。図 2 から、モバイル端末と Linux PC 間で MPI 並列処理が実行できていることが確認できる。

課題として、MPI プログラムの配置場所がある。リモートノードの MPI プログラムは、ホストノードと全く同じディレクトリに配置する必要がある。本稿では、ホストノードの MPI プログラムは、`/data/data/mpl.app/files/bin` というディレクトリ内にある。しかし、Linux PC には同じディレクトリは存在しない。そのため、Linux PC では、手動で並列処理アプリと同様の名前のディレクトリを作成し MPI プログラムを配置した。この時、フォルダの作成に管理者権限が必要となるため、並列処理アプリから MPI プログラムの配置を行うことができない。これは、並列処理アプリの操作のみで Linux PC をノードして利用する MPI 並列処理を行うために解決すべき課題である。

5 おわりに

本稿では、異なる OS が混在する環境の MPI 並列処理について、ホストノードを Android OS を搭載したモバイル端末、リモートノードを Linux PC として実行が確認できた。今後の課題として、Linux 以外にも対応可能にすることが挙げられる。

謝辞

本研究は、一部 JSPS 科研費 20K11726 の援助による。

参考文献

- [1] Masahiro Nissato, et al.: “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.
- [2] Open MPI:Open Source High Performance Computing <https://www.open-mpi.org/>, 2020 年.
- [3] 吉瀬 謙二, 片桐 孝洋, 本多 弘樹, 弓場 敏嗣: “qn24b:N-queens の解を計算するベンチマークプログラム,” FIT2004, 第 3 回情報科学技術フォーラム, 第 4 分冊, No.O-011, pp.389-392, 2004.