

## Android 端末を使用したクラスタ計算機システムの構築

荒井 裕介<sup>†</sup> 大津 金光<sup>†</sup> 大川 猛<sup>†</sup> 横田 隆史<sup>†</sup> 馬場 敬信<sup>†</sup>  
<sup>†</sup>宇都宮大学工学部情報工学科

### 1 はじめに

今日, AndroidOS を搭載したスマートフォンやタブレット端末といった携帯端末が急速に普及している. これらの携帯端末は近年急速な高性能化を遂げており, 最近ではデュアルコアやクアッドコアといったマルチコアプロセッサを搭載した端末も普及しつつあり, 単体での処理性能が大幅に向上している.

Android 端末は, Wi-Fi や Bluetooth といった無線通信機能を持っており, これらの通信機能を利用することで複数の端末間での相互通信が可能である. このような端末間での無線相互通信を利用することで無線接続型のクラスタ計算機システムを実現することができ, 多くの端末が集まるような機会を利用して大規模な演算を行うことが可能となる.

本稿では, ネットワークによって相互に無線接続された複数の Android 端末を使用して構築したクラスタ計算機システムについて検討する.

### 2 無線接続型クラスタ計算機

#### 2.1 システムの概要

クラスタ計算機は, 独立した CPU とメモリからなる計算機をネットワークで接続した分散メモリ型の構造をしている. そのため, 分散メモリ型並列計算機での並列処理において複数の端末を用いてクラスタ計算機を構築する場合, 端末間の通信性能が実行性能に大きく影響する. AndroidOS は携帯電話網をはじめ, Wi-Fi や Bluetooth といった多くの無線通信機能を備えている. 本システムでは端末間のネットワーク接続を無線通信によって行うが, もっとも一般的で高速な無線 LAN 規格である Wi-Fi を通信手段として用いる.

本システムの, システムソフトウェア全体の構成を図 1 に示す. AndroidOS は Linux カーネルを用いており, Linux アプリケーションを動作させることが可能である. しかし, ライブラリは Linux ベースではないため, そのままでは Android 端末上で動作させることはできない.

そこで, 本システムでは Linux-on-Android を用いて Ubuntu(ver.12.04) を導入して利用する. Android アプリケーションは, 通常 DalvikVM という仮想マシン上で動作するが, Linux-on-Android は DalvikVM を介さずに, AndroidOS と同時にカーネル上に展開される. そのため AndroidOS の機能を使用しながら, 同

時に Linux のアプリケーションを動作させることが可能となる. また, 並列分散処理を行うための標準仕様である MPI(Message Passing Interface) を使用して並列プログラムを動作させる. Ubuntu では MPI の実装の一つである OpenMPI[1] がパッケージとして提供されているため, これを利用する. OpenMPI を Ubuntu 上に導入することで, MPI 並列プログラムを Android 端末上で処理させることができる.

本システムの概略を図 2 に示す. 本システムでは Wi-Fi に複数の Android 端末同士を相互に接続し, OpenMPI で並列分散処理を行う. ネットワーク接続された複数のノード間で, MPI を用いて並列分散処理を行う場合, 並列処理を行うノード間の接続は, 通常各ホストに静的に割り付けられた IP アドレスを用いて行われる. この割り付けられた IP アドレスと, 各端末で利用可能な CPU コア数の情報を記載した設定ファイルを使用して, 各端末へのプロセス割り当てを制御する. しかし, Android 端末のようなモバイル性の高い機器をネットワークに接続する場合, DHCP (Dynamic Host Configuration Protocol) を用いて動的に IP アドレスを割り付けているため, IP アドレス情報を MPI に動的に受け渡す手段の開発が必要となる.

以上のように, Wi-Fi による無線ネットワーク接続, Android 端末上での Linux-on-Android の展開, Linux-on-Android 上に導入された OpenMPI による並列分散処理を Android 端末で行うことによって, Android 端末を使用した無線接続型クラスタ計算機のシステムを実現することができる.

#### 2.2 実行モデル

本システムの実行モデルは, 分散メモリ型モデルとなる. MPI プログラムが起動すると, すべてのノードでプロセスが起動する. 各ノードで起動したプロセス

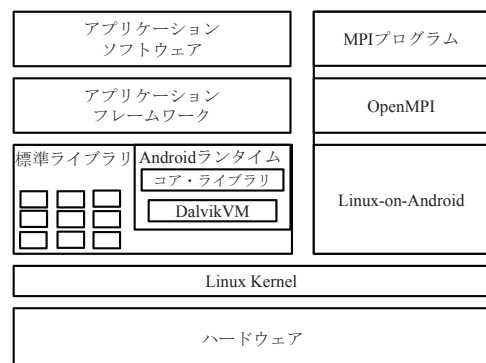


図 1: システムソフトウェアの構成

Construction of Cluster Computing System using Android Devices

<sup>†</sup>Yusuke Arai, Kanemitsu Ootsu, Takeshi Ohkawa, Takashi Yokota and Takanobu Baba

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

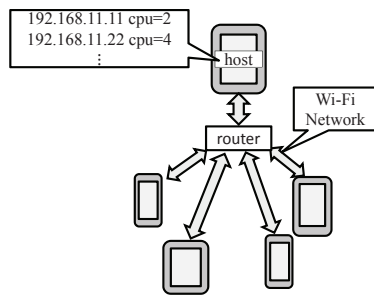


図 2: 無線接続型クラスタ計算機システム

はノード間でデータの送受信を行いながら、並列処理を行う。

### 3 初期評価

今回構築したクラスタ計算機システムの性能特性を明かにするため、各種プログラムによる処理時間の計測を行う。

#### 3.1 評価方法

性能評価は N クイーン問題プログラム (クイーン数 16), 巡回セールスマン問題 (TSP) プログラム (都市数 14) を用いて行う。評価で使用した Android 端末の性能を表 1 に示す。本稿では表 1 の端末を 2 台, あるいは 3 台使用した場合の処理時間を計測し, 1 台で実行した場合の処理時間と比較したときの速度向上率を示す。使用 CPU コア数は 1 台での実行時が 4 コア, 2 台での実行時が 8 コア, 3 台での実行時が 12 コアとなっている。また, プログラム実行時の各 Android 端末への実行プロセスの割り当ては, 1 台での実行では 4 プロセス, 2 台での実行では 4+4 の 8 プロセス, 3 台での実行では 4+4+4 の 12 プロセスで並列処理を行う。なお, 評価環境での無線ネットワークの帯域は, ネットワーク帯域ベンチマークツール Iperf による測定の結果, 17Mbps であった。

#### 3.2 評価結果

N クイーン問題プログラムでは 1 台での処理時間が 84.15 秒, 2 台での処理時間が 38.23 秒, 3 台での処理時間が 25.36 秒であった。1 台 (4 プロセス) 実行時と比較した速度向上率は, 2 台の時の 2.2 倍, 3 台の時の

3.32 倍となり, 速度向上が確認された。

TSP プログラムでは, 1 台での処理時間が 873.67 秒, 2 台での処理時間が 946.32 秒, 3 台での処理時間が 943.52 秒となった。1 台 (4 プロセス) 実行時と比較した速度向上率は, 2 台の時の 0.92 倍, 3 台の時の 0.93 倍となり, いずれも 1 台実行時に比べて性能が低下した。

### 3.3 考察

N クイーン問題はすべてのプロセスが独立して部分解の計算を行い, それぞれのノードでの部分解をいずれかのノードに集約し最終的な解を求めるプログラムであるため, ノード間での通信が部分解の送受信以外に発生しない。そのため通信によるオーバーヘッドが性能上の問題にならず, 速度向上したと考えられる。TSP プログラムでは, 100Byte 程度のデータを, ノード間で約 33 万回送受信している。通信 1 回当たりに要する時間は少ないが, ノード間の通信の回数が非常に多いため, 結果として通信時間が増大し速度低下したと考えられる。以上のことから, 本システムでは, 通信量の少ないプログラムでの並列処理で高い性能を発揮できるといえる。

### 4 おわりに

Android 端末の Wi-Fi 接続機能を用いて, 無線接続型のクラスタ計算機システムの構築と概要を述べた。構築したシステムで N クイーン問題プログラム, 巡回セールスマン問題プログラムを実行した結果, 通信量が少ない N クイーン問題プログラムでは速度向上し, 頻繁に通信を行う TSP プログラムでは速度低下した。

現状の本システムでは, DHCP に対応したシステムを構築する必要がある。現在は設定ファイルをあらかじめ用意しておく必要があるため, DHCP によって割り当てられている IP アドレスが変わった際に DHCP と自動で対応することができない。そのため今後の課題として, IP アドレスが変わった際に DHCP と連動して自動で対応できるシステムを構築する必要がある。具体的には, IP アドレスを監視するプログラムをバックグラウンドで実行させておき, 定期的にホスト情報ファイルを書き換える, といった方法が考えられる。このようなシステムを今後の課題として実装することにする。

#### 謝辞

本研究は, 一部日本学術振興会科学研究費補助金 (基盤研究 (C)24500055, 同 (C)24500054) の援助による。

#### 参考文献

[1] Richard L. Graham, Galen M. Shipman, Brian W. Barrett, Ralph H. Castain, George Bosilca: "Open MPI: A High-Performance, Heterogeneous MPI", The 2006 IEEE International Conference on Cluster Computing, 2006.

表 1: 使用機器情報

Android 端末	
使用端末	ASUS 社 TF201
CPU	NVIDIA Tegra3 モバイルプロセッサ
コア数	4
動作周波数	1.4GHz(クアッドコア稼働時 1.3GHz)
メモリ	1GByte
無線 LAN 規格	IEEE 802.11 b/g/n
無線 LAN ストリーム	1 ストリーム
ルーター	
メーカー	BUFFALO 社
型番	WHR-G301N