Android OS における 演算性能と I/O 性能の評価と解析

服部拓也† 新居健一† 山口実靖†

近年スマートフォンの普及により Android OS が注目されている．Android OS のカーネルは Linux カーネルをもとに作成されており，ユーザーアプリケーションは Java 言語で構築され Dalvik VM 上で動作する．本稿では，Android OS を搭載した携帯端末および計算機(PC)の性能と，Linux を搭載した計算機の性能の調査を調査し，比較を行った．結果，現状では多くの例において Linux が性能において優れていることが確認された．また，Android カーネルのモニタリングシステムを提案，構築し，Android システムに対して適用した．結果，カーネル内部の処理時間の観察が可能であることが確認された．

Performance Evaluation and Analyses of Android OS

TAKUYA HATTORI KENNICHI NII
SANEYASU YAMAGUCHI†

Android OS is one of promising OS for mobile devices. In this paper, we present detailed performance evaluation of Android OS and Linux OS. Our results demonstrated that Linux OS provides better performance in many cases, and the current Android OS implementation can be improved more. For further analyses, we constructed Android kernel monitoring system. We evaluated it by applying it to Android OS, and the system revealed processes inside Android kernel.

1. はじめに

近年スマートフォンの登場により Android OS が注目されている．Android OS は Linux カーネルをもとに開発が進められているオープンソースの携帯端末向け OS であり，誰でも改変が可能である．アプリケーションは Java 言語で構築され，Java VM に相当する Dalvik VM と呼ばれる小メモリ環境に最適化された Android OS 独自の仮想マシンの上で動作する．Android OS はオープンソースであるため利便性や拡張性が高く，今後は新しく多様な機能を持った携帯端末の誕生が期待できる．

本稿では，Linux OS を搭載した計算機と Android OS を搭載した計算機 および携帯端末を用いた基本性能の評価と I/O 解析システムの構築と適用を行い，Android OS の動作解析を目指す．

2. Android OS

豊富なユーザーインタフェース，アプリケーション，コードライブラリ，アプリケーションフレームワーク，マルチメディアサポートなどの機能を含んでおり，カーネル部は Linux カーネルを用いて C 言語で記述されている．ミドルウェア部でも多くのオープンソースソフトウェアが採用されており，例えば DBMS として SQLite が用いられている．これらに関しては Linux と類似の振る舞いを示し，類似の性能を示すと予想される．しかし，Android Runtime やアプリケーションフレームワークは独自のものを用いており，必ずしも近い性能になると予想されるわけではない．また，多くの Linux は x86 または x64 の実装が用いられているが，多くの Android では ARM 用の実装が用いられており，これらの差異は性能に影響を及ぼす可能性がある．

3. 関連研究

Android OS の性能評価における研究としては，以下のものがある．

三木らは文献[1]，[2]において，Android OS におけるネットワークコンピューティング能力について評価および解析を行っている．彼女らの調査によれば，同性能の CPU を搭載した Android OS および Linux OS において，高遅延環境下で Android OS が受信側になった場合のスループットが，Linux OS に比べ低下することがわかっている．また，Android 端末にカーネルモニタを導入することにより，2 台の Android 端末が同時に 1 つのアクセスポイントを経由しサーバと通信を行った際の，輻輳ウィンドウの値の変化を解析することに成功している．

間嶋らは文献[3]において，各 Dalvik バイトコードの CPU 負荷量を解析するためのマイクロベンチマークの生成方法および実施方法を提案している．彼らの調査によって，アプリケーションの CPU 負荷量の正確なモデル化を行うためには，トレース中のバイトコードのみでは不十分であり，バイトコードの種類や引数を考慮する必要があることが明らかとなっている．

†工学院大学 大学院 工学研究科 電気電工学専攻
Electrical Engineering and Electronics, Kogakuin University Graduate School

4. 基本性能評価

Android OS における基本性能の評価を行った. 使用計算機と測定結果を以下に示す.

4.1 測定環境

基本性能評価のため， Linux OS と Android OS を搭載した PC，および Android OS を搭載した携帯端末を用意した．Linux OS 上で用いた Java は，バージョン 1.5.0-19 である．測定に使用した環境を表 1 に示す．各環境で使用した OS はグラフ内に示す．本稿では Server PC と Netbook のみ，同一の計算機に Linux OS と Android OS を稼動させ，性能の比較を行った．それ以外の計算機については Android OS についてのみ評価した．

表 1 使用計算機

		CPU	Memory	Network
PC	Server PC	Intel Celeron 440 (2 [GHz])	1024 [MB]	100Base TX Ethernet
	Netbook	Intel Atom N280 (1.66 [GHz])	512 [MB]	Wi-Fi 54 [Mbps]
	AZ/05M	NVIDIA Tegra 250 (1 [GHz])	512 [MB]	Wi-Fi 54 [Mbps]
携帯端末	Xperia arc	Qualcomm MSM 8255 (1 [GHz])	512 [MB]	Wi-Fi 54 [Mbps]
	Xperia	Qualcomm QSD 8250 (1 [GHz])	384 [MB]	Wi-Fi 54 [Mbps]
	HT-03A	Qualcomm MSM 720a (528 [MHz])	192 [MB]	Wi-Fi 54 [Mbps]
	Ziio 10	ZiiLABS ZMS-08 (1 [GHz])	512 [MB]	Wi-Fi 54 [Mbps]

4.2 CPU 性能

Java 言語で記述されたベンチマークプログラム LINPACK および SciMark2 を，Android Dalvik VM 上と Linux Java VM 上で実行し CPU 性能の測定を行った．測定結果を図 1 に示す．また，C 言語で記述された姫野ベンチマークをネイティブコードにコンパイルしたものと，Java 言語で記述された同ベンチマークを各バイトコードにコンパイルしたものを実行させその性能を測定した．結果を図 2 に示す．図 1 より，Android OS における性能が Linux OS における性能より低いことがわかる. 特に Server PC における SOR 法による測定結果では Android OS は Linux OS の約 14%, Netbook におけるモンテカルロ法による測定結果では Android OS は Linux OS の約 7%の性能しかでておらず大きな差が出ていることがわかる．また図 2 より，両 OS ともネイティブコードプログラムより，バイトコードプログラムの性能の方が低いが，Linux OS ネイティブと Java VM の差より，Android OS ネイティブと Dalvik VM の差のほうが大きいことがわかる.

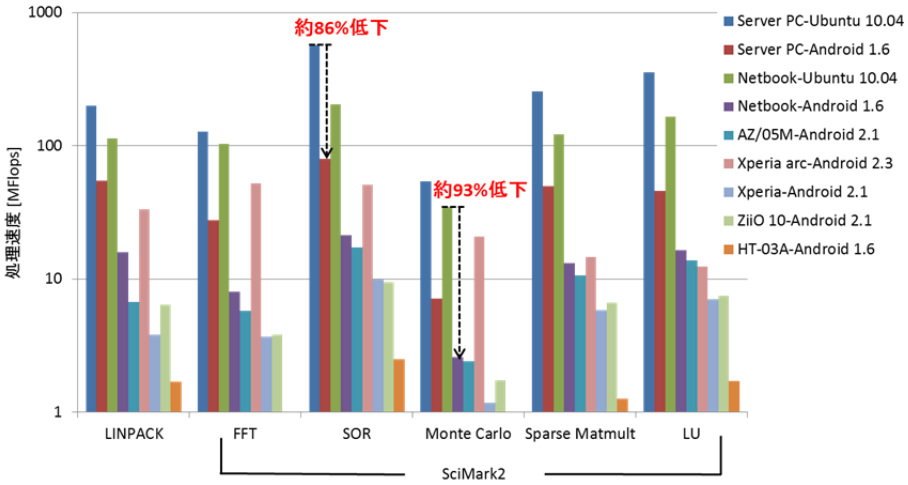

図 1 LINPACK および SciMark2 による CPU 性能評価

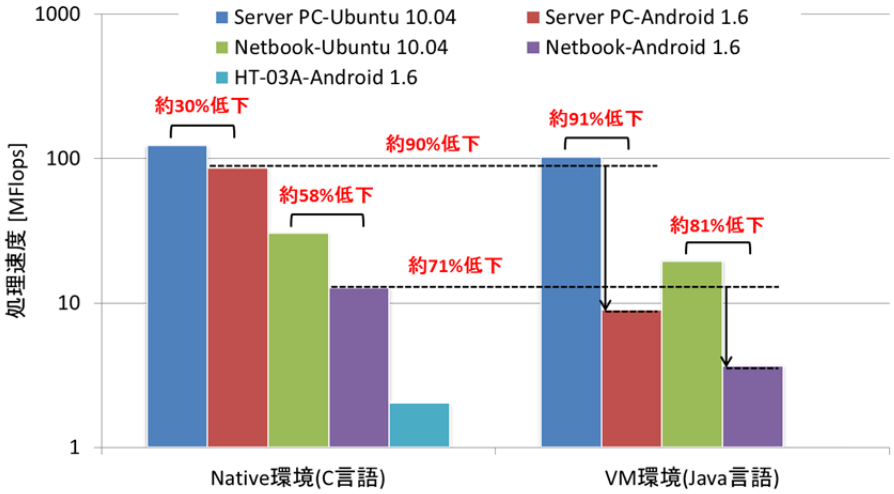

図 2　姫野ベンチマークによる CPU 性能評価

4.3 HTTP トランザクション

通信処理の性能として，HTTP トランザクションの処理速度の測定を行った．HTTP トランザクションは，Web サーバより 1 バイトのデータを 1 秒間に何回取得できるかを測定したものである． Server PC はスイッチを介して Web サーバと接続され，それ以外の実験機は無線 LAN ルータとスイッチを介して Web サーバと接続されている．測定結果を図 3 に示す. 縦軸は 1 秒間に処理した HTTP トランザクションの数である．図 3 より，HTTP トランザクションにおいても Android OS の性能が Linux OS よりも低いことがわかる．

4.4 描画性能

Java 言語にて，矩形，直線，円を繰り返し描画するベンチマークプログラムを作成し，Android Dalvik VM 上および Linux Java VM 上でその性能を測定した．測定結果を図 4 に示す．描画性能についても，矩形を除き Android OS の性能が Linux OS よりも低い性能を示している．直線の描画速度に注目すると，図 1 の CPU 性能の相対関係に似ていることがわかる．よって描画性能は CPU 性能に強く依存していると考えられる．

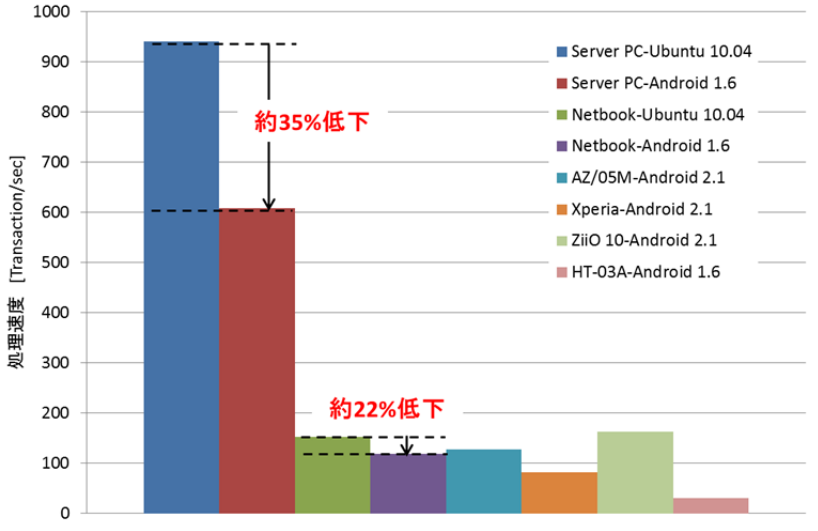

図 3　HTTP トランザクションの処理速度

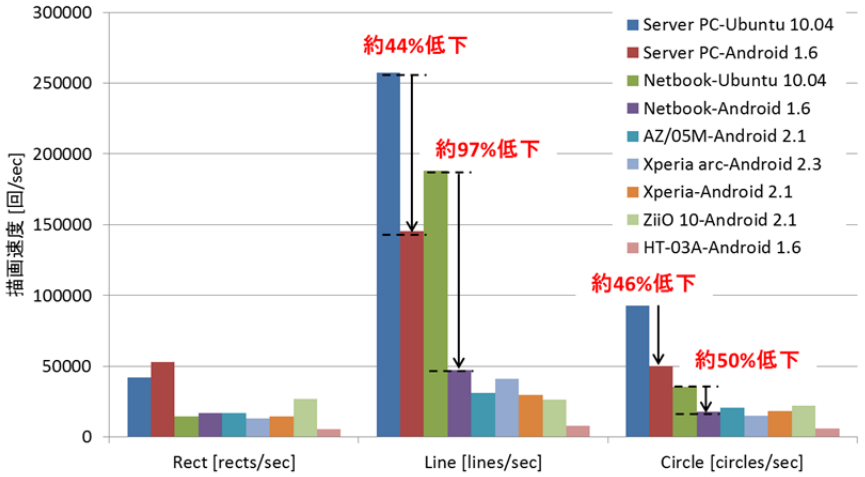

図 4　描画性能

4.5 I/O 処理性能

I/O 処理速度の評価として，シーケンシャル I/O 速度，ファイル I/O 速度，データベースアクセス速度の評価を行った．

4.5.1 シーケンシャル I/O

dd コマンドを使用し，Server PC では 1MB のデータを 512 回，Netbook 1 では 1024 回，HT-03A では 192 回，内部ストレージのファイルに書き込み，その後読み込みを行った．測定結果を図 5 に示す．外部ストレージに対し同様の処理を行った際の測定結果を図 6 に示す．Server PC および Netbook の内部ストレージは HDD であるが，HT-03A はフラッシュメモリである．また Server PC および Netbook の外部ストレージは USB メモリであるが，HT-03A は SD card である．

図 5 より，書き込み速度，読み込み速度，どちらも Android OS の性能が Linux OS よりも低いことがわかる．また図 6 においては書き込み速度が同程度の結果となった．読み込み速度については内部ストレージと同様 Android OS の性能が低い．図 6 では HT-03A 以外は USB メモリに対して読み書きしており，デバイスの限界の性能が得られていると考えられる．

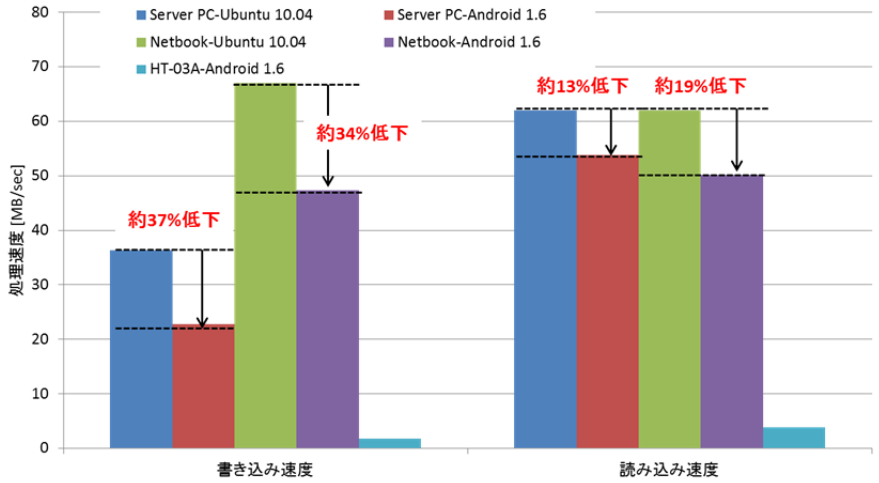

図 5　シーケンシャル I/O 性能(内部ストレージ)

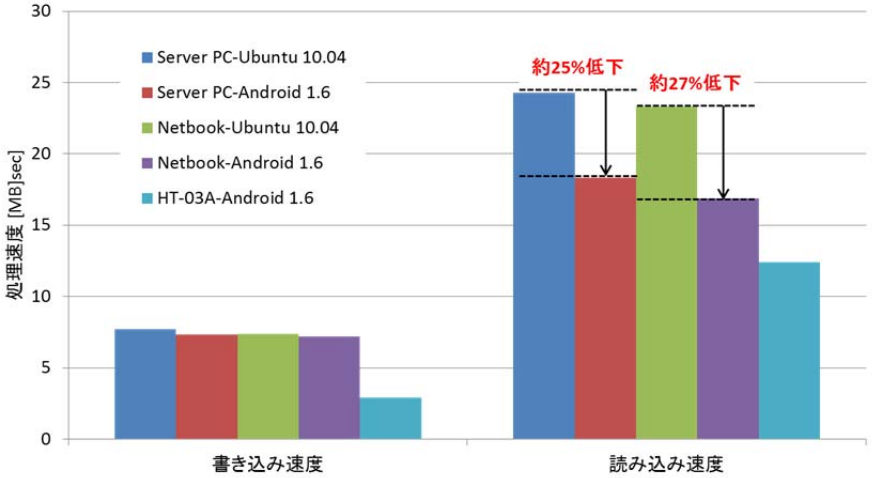

図 6　シーケンシャル I/O 性能 (外部ストレージ)

4.5.2 ファイル I/O ベンチマーク

Java 言語で作成した，任意のサイズのファイルを任意の個数書き込み・読み込みを行うベンチマークプログラムを使用しファイル I/O 性能を測定した．Server PC，AZ/05M では 1MB のファイルを 512 個，Netbook では 1024 個，その他の端末では 192 個，内部ストレージおよび USB メモリ，もしくは SD card に対し，書き込み・読み込みを行った．測定結果を図 7，図 8 に示す．この測定においても，Android OS が Linux OS よりも低い性能を示し，外部ストレージに対する書き込みは，シーケンシャル I/O 速度の測定と同様に多くの計算機で同程度の性能となった．これも使用した USB メモリの転送速度が影響したものであると考えられる．

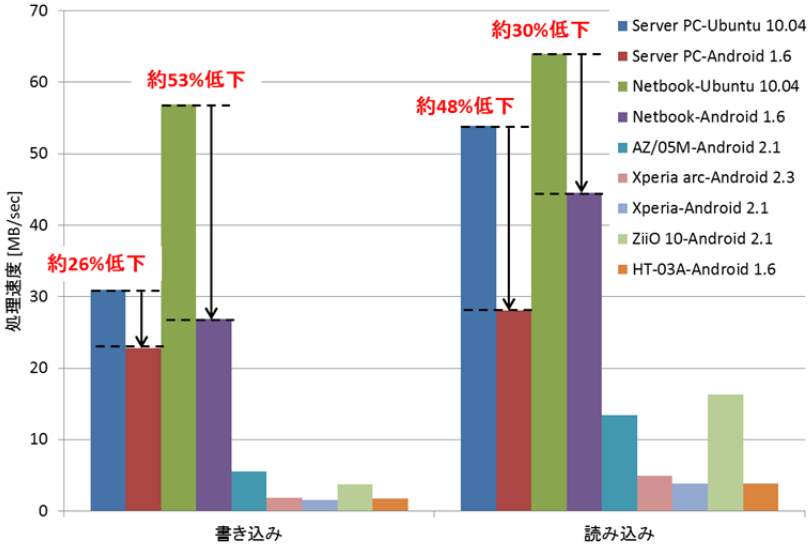

図 7　ファイル I/O ベンチマークによる性能評価(内部ストレージ)

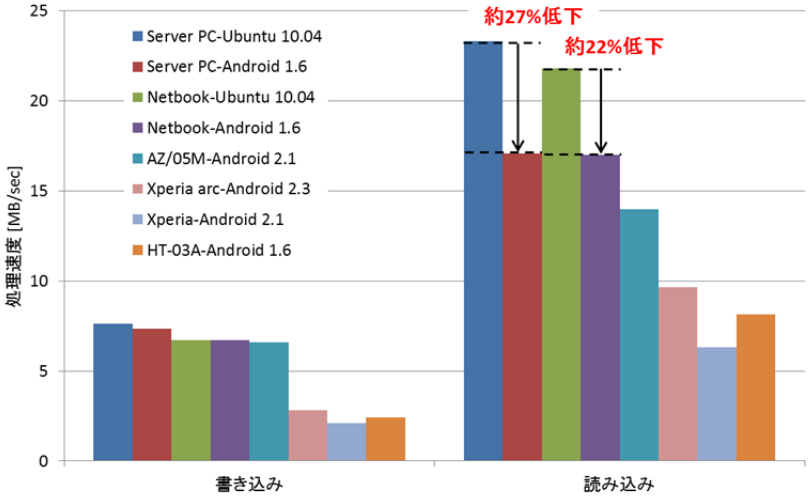

図 8　ファイル I/O ベンチマークによる性能評価(外部ストレージ)

4.5.3 データベースアクセス性能

SQLite を用いて，データベースに対する Insert 処理，Select 処理の速度を測定した．結果を図 9，図 10 に示す．SQLite による Insert 処理速度は，int 型と String 型の 2 列で構成されるテーブルに，整数値と 100 バイトの文字列 1,000 行を Insert するのに要した時間を計測した．Select 性能は，1,000 行で構成される上記テーブルに対して全行のスキャン(SELECT * FROM TABLE;)を，10,000 回行うことにより計測した．

結果より，Insert 性能に比べ Select 性能が非常に高く，Select 性能においては CPU 性能に依存していることがわかる．また，Server PC および Netbook において Linux OS と Android OS を比較すると，Android OS の性能が Linux OS よりも低いことがわかった．　Select 性能が Insert 性能を大きく上回る理由については次章で述べる

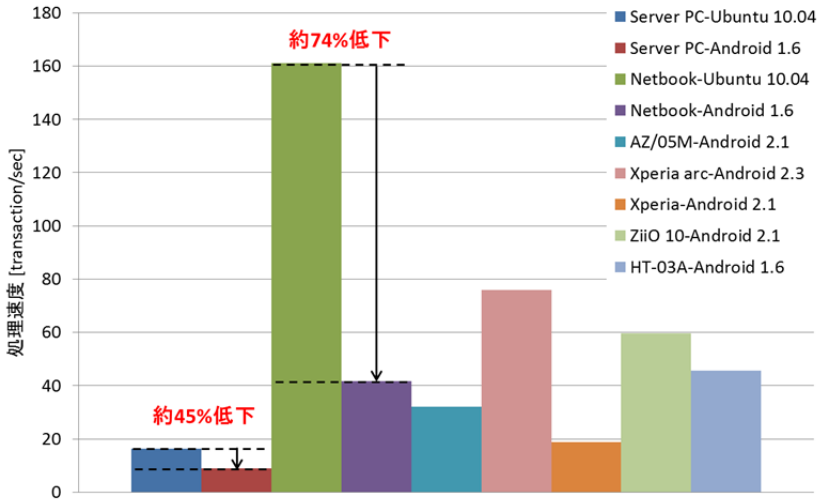

図 9　SQLite による Insert 処理速度

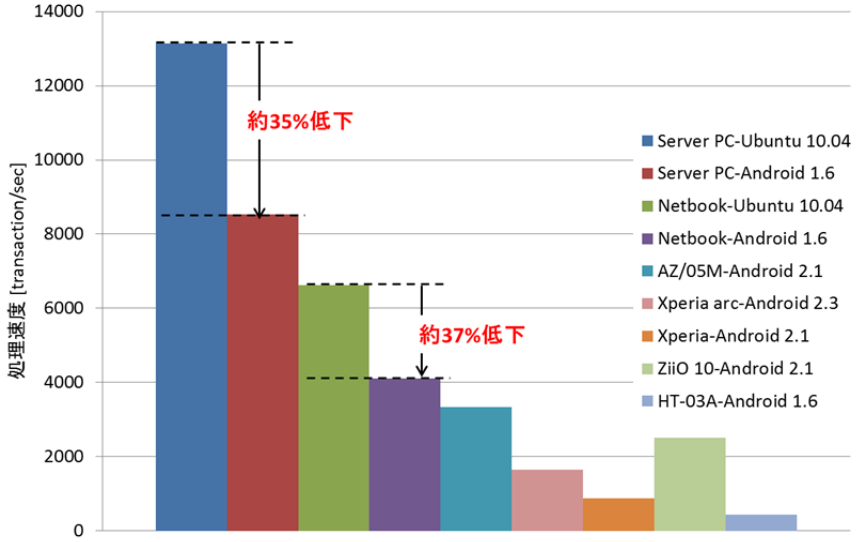

図 10　SQLite による Select 処理速度

5. I/O 解析

I/O 性能の測定により, SQLite の Select 性能が Insert 性能を大きく上回ること, Insert 処理速度において Linux OS と Android OS の性能差が大きいことがわかった．そこでこの二処理に焦点を絞り，カーネル解析ツールを構築しその動作を確認した．

5.1 Select 処理と Insert 処理の比較

Android OS のカーネルに，発行された SCSI コマンドをメモリに保持する機能を追加し，Select 処理と Insert 処理時に発行される SCSI コマンドを調査した．Insert 処理時に発行された SCSI コマンドを図 11 に，Select 処理時に発行されたものを図 12 に示す．横軸は時刻，縦軸はアクセスアドレスである．図より，Insert 処理時はストレージに対する書き込み要求が発行されているが，Select 処理時はストレージアクセスが発生しておらず，メモリキャッシュにより I/O が処理され Select 性能が Insert 性能を大きく上回ったと予想できる．

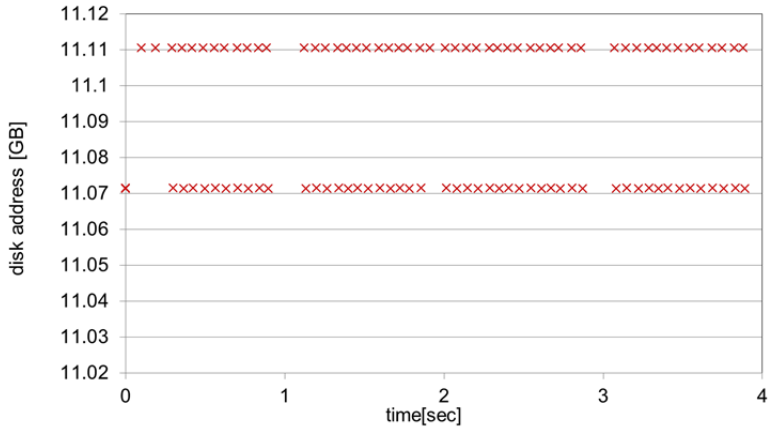

図 11　Insert 時に発行された SCSI コマンド

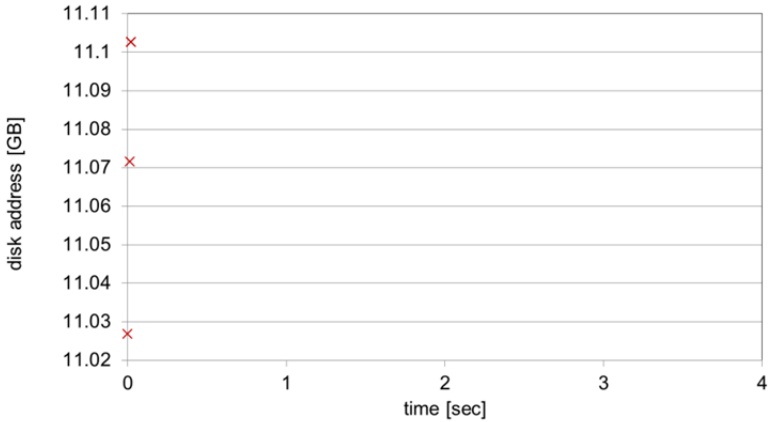

図 12　Select 時に発行された SCSI コマンド

5.2 Linux OS と Android OS の Insert 処理

SQLite を用いたアプリケーションが Insert 要求を行う関数の直前(App(start))と直後(App(end))の時刻，SQLite が Insert 要求を発行する関数に入った時刻(SQLite(start)と抜け出す時刻(SQLite(end))，ファイルシステムが書き込みを要求する関数 generic_file_aio_write に入った時刻(aio(start))と抜け出す時刻(aio(end))，ブロック I/O を Request に変換し I/O スケジューラに入れる関数 submit_bio に入った時刻(bio(start))と抜け出す時刻(bio(end))，最後に SCSI コマンドが発行された時刻(scsi)を測定し，各レイヤで要求された書き込みサイズ，書き込み先のアドレスや inode 番号などを取得した．図 13 に上記の各レイヤと処理の流れを示す．図 14 に Android OS におけるモニタリング結果を，図 15 に Linux OS におけるモニタリング結果を示す．両図の横軸の幅はともに 12 [msec]である．横軸はベンチマークソフトが Insert 要求を初めてからの経過時間，縦軸は上からアプリケーション，SQLite，ファイルシステム層，ブロックデバイス層，SCSI 層，と各レイヤを表している．

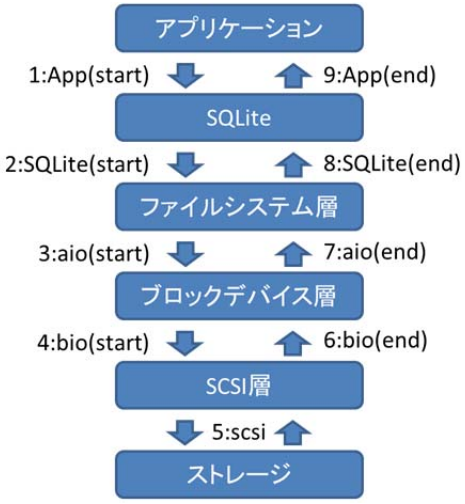

図 13　各レイヤと測定時間の関係

Android OS の 200.097 [sec]付近と Linux OS の 200.030 [sec]付近のファイルシステム層において，Android OS では 1024 [B]のデータベースに対する書き込み要求が 3 回，Linux OS においては要求が 2 回あることがわかった(図中赤線部)．さらに同時刻付近のブロックデバイス層，SCSI 層に注目すると，Android OS においては 4 [KB]の書き込み要求が 2 回と 8 [KB]の書き込み要求が 1 回，Linux OS においては 4 [KB]の書き込み要求が 2 回存在することが確認できた．このように，当該モニタリングシステムでカーネル内の I/O 処理の動作を観察できることが確認された．

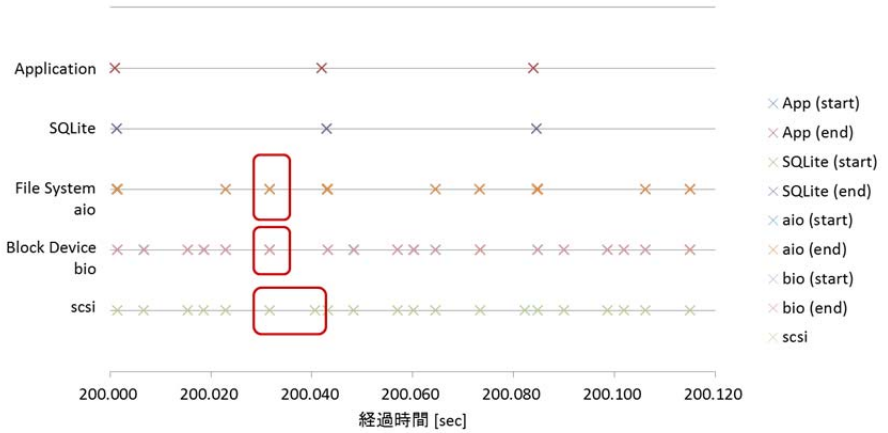

図 14　Android OS におけるモニタリング結果

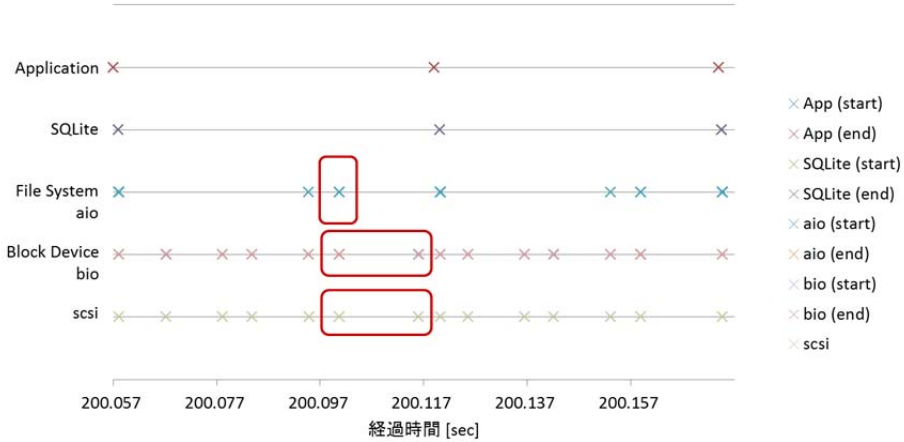

図 15　Linux OS におけるモニタリング結果

6. おわりに

本稿では，Android OS を搭載した携帯端末および計算機(PC)の性能と，Linux を搭載した計算機の性能の調査を調査し，比較を行った．結果，現状では多くの例において Linux が性能にて優れていることが確認された．また，Android カーネルのモニタリングシステムを提案，構築し，Android システムに対して適用した．結果，カーネル内部の処理時間の観察が可能であることが確認された．

今後は更なる解析を行い，Android OS における I/O 性能低下の原因を調査し，性能向上に関する考察を行なっていく予定である．

謝辞

本稿は科研費 (22700039) の助成を受けたものである．

参考文献

[1] 三木香央理，山口実靖，小口雅人，“Android 端末におけるカーネルモニタの導入”，第 22 回コンピュータシステム・シンポジウム，2010 年 11 月

[2] 三木香央理，小口雅人，“Android 端末の無線 LAN 通信時のトランスポート層の振舞に関する一検討”，情報処理学会論文誌　Vol.50　No.2

[3] 間島崇，横山哲郎，曾剛，神山剛，冨山宏之，高田宏，“Android プラットフォームにおける Dalvik バイトコードの CPU 負荷解析”，情報処理学会研究報告，2010 年 3 月