3K-3

Android OS におけるデータベースのアクセス性能に関する考察

服部　拓也[†]　　新居　健一[†]　　山口　実靖[†]

工学院大学大学院 工学研究科 電気・電子工学専攻[†]

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 の動作と性能に関する研究としては，Android OS の TCP 実装の振る舞いに関する研究[1][2]や，Linux と Android でデータベース性能に大きな違いがあることを示した報告[3]があるが，Android OS における I/O 処理時の OS の動作を俯瞰的に調査した報告は少ない．本稿では，カーネル部に Linux カーネルを用いている 3 種類の OS(Linux，Android x86，Android)を用意し，各 OS におけるデータベース処理実行時の OS 全体の動作を俯瞰的に調査し，z 違いが生じている部位について考察をする．

2．データベースアクセス性能

以下の 3 種類の測定環境を用意し，データベース処理を行い，性能を比較した．1 個目の測定環境は PC (CPU Intel Celeron 440 - 2[GHz]，メモリ 1024[MB])に Ubuntu 10.04 (Linux カーネル 2.6.32)を搭載した環境であり，2 個目は同一 PC に Android x86 1.6 (カーネル 2.6.32)を搭載した環境であり，3 個目は携帯電話 Nexus S (CPU Cortex A8 - 1[GHz]，メモリ 512[MB])に Android 4.0 を搭載した環境である．SQLite のバージョンは順に, 3.5.9，3.5.9，3.7.4 である．

評価実験は，SQLite を用いてデータベースに対する Insert 処理を繰り返し行い性能を評価した．Insert 処理では，int 型と String 型の 2 列で構成されるテーブルに，整数値と 100 バイトの文字列を 10,000 行 Insert した．測定結果を図 1 に示す．図より，Insert 処理の性能においては，PC 搭載 Android x86 の性能が PC 搭載 Linux OS に比べ約 45%低く，携帯電話搭載 Android に比べても 34%低いことがわかった．

3．I/O 解析

前章の結果より，SQLite の Insert 処理に大きな差があることがわかった．各 OS の性能差の理由を解析するために，Insert 処理実行時のアプリケーション，データベース管理システム，カーネルの処理時刻を調査できるシステムを実装し，動作を確認した．具体的には，アプリケーションが Insert 要求を行う直前と直後の時刻（App），

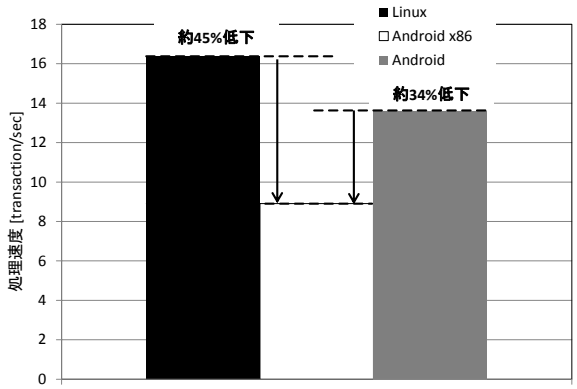

図 1　データベース Insert 性能

SQLite が Insert 要求を開始する時刻と終了する時刻（SQLite），ファイルシステムが書き込みを開始する時刻と終了する時刻（File System），ブロック I/O の I/O スケジューラへの転送が開始される時刻と終了する時刻（Block），スケジューラがリクエストを発行する直前と直後の時刻（Elevator），SCSI ドライバにて命令が発行された時刻と終了した時刻（SCSI），MMC ドライバにて命令が発行された時刻と終了した時刻（MMC）を調査した．Linux と Android x86 に関しては SCSI ドライバの調査を，Android においては MMC ドライバの調査をした．また，ファイルシステムに関しては，発行された要求の書き込みサイズと書き込み対象のアドレスと inode 番号も取得した．これらの調査システムは，各 OS の実装に履歴保持機能を追加して実現した．

図 2 に Android x86 OS におけるモニタリング結果を，図 3 に Linux OS におけるモニタリング結果を，図 4 と図 5 に Android におけるモニタリング結果示す．図 2，図 3，図 4 の横軸の幅はともに 80 [msec]である．横軸はベンチマークソフトが最初の Insert 要求を開始してからの経過時間，縦軸は上からアプリケーション層，SQLite 層，ファイルシステム層，ブロックデバイス層，スケジューラ層，SCSI 層あるいは MMC 層を表している．

図 2,3 より Linux と Android x86 では 1 回の Insert 処理で 3 回のファイルシステムへの書き込み処理が行われていることがわかる．また図 4 の Android では，1 回の Insert 処理で 14 回のファイルシステムへの書き込みが行われている．図 2，図 3 の点線内の 2 処理は共に“test.db-journal”

Performance Considerations for database access in Android OS
Takuya HATTORI[†], Kenichi NII[†], Saneyasu YAMAGUCHI[†]
Department of Information and Communications Engineering, Kogakuin University[†]

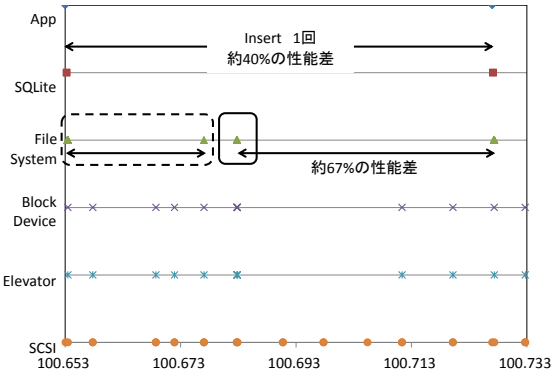

図 2　Android x86 におけるモニタリング結果

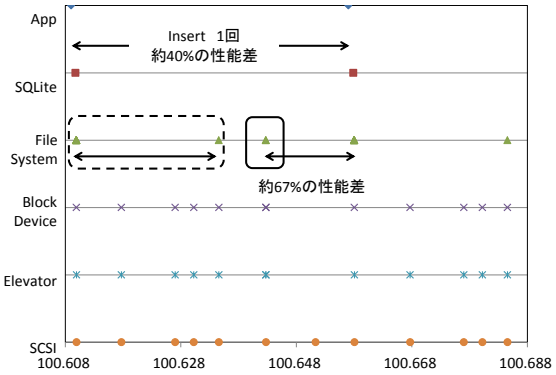

図 3　Linux OS におけるモニタリング結果

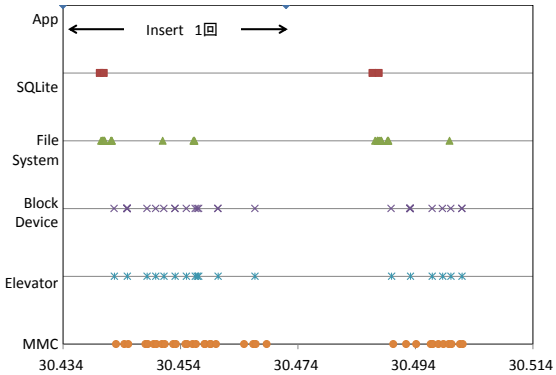

図 4　Android におけるモニタリング結果 1

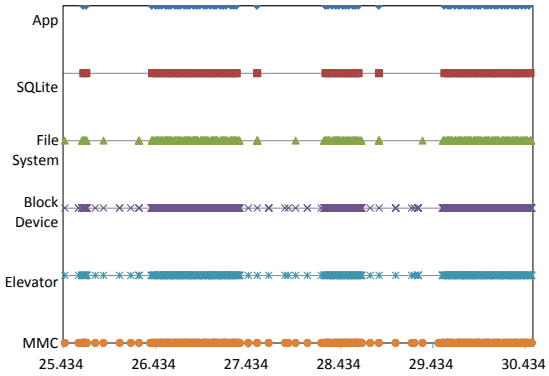

図 5　Android におけるモニタリング結果 2

(SQLite のジャーナルファイル)，実線内の処理は “test.db” (SQLite のデータベースファイル)に対する書き込みである．図 4 の Android では，最初の 11 回の書き込みはジャーナルファイル，後の 3 回の書き込みはデータベースファイルへの書き込みである．同様にブロックデバイス，I/O スケジューラにおいても処理要求の発行回数に大きな違いがあり，Linux と Android x86 の例よりも Android の例が多くなっている．

図 2，図 3 を比較すると，1 回目のジャーナ書き込みから 2 回目のジャーナル書き込みまでの時間に大きな差は無く，データベースファイルへの書き込みから次の Insert 処理が開始されるまでの時間に大きな差(約 67%)がある．さらに SCSI 層に注目すると，データベースファイルへの書き込み以降に発行される命令数が Linux より Android x86 の方が多くなっている．

図 4 を図 2,図 3 と比較すると，Android における Insert 処理 1 回の時間は他の例よりも短くなっていることがわかる．しかし，横軸の範囲を広げた図 5 を見ると要求が密の部分と疎の部分がある事が確認でき，常に性能が高いわけではなく結果として性能が低下していることがわかる．

また図 5 より，Android では I/O 要求が層間を転送されるのにかかる時間が大きいことがわかる．

4．おわりに

本稿では，Linux，Android x86，Android におけるデータベース Insert 処理性能を評価し，各 OS の動作を俯瞰的に調査した．調査の結果，1 回の Insert 要求により発行されるカーネル内命令の数に大きな違いがあることがわかった．

今後は性能向上手法について考えていく予定である．

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，小口正人，“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，小口正人，“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] 服部拓也，新居健一，山口実靖，”Android OS の I/O 性能評価と動作解析”，第 10 回情報科学技術フォーラム，2011 年 9 月