

# Android における OS 可視化環境の開発

中川 裕貴<sup>†</sup> Praween Amontamavut<sup>††</sup> 西野 洋介<sup>†††</sup> 早川 栄一<sup>††</sup>

拓殖大学工学院工学研究科電子情報工学専攻<sup>†</sup> 拓殖大学工学部情報工学科<sup>††</sup>

東京都立八王子桑志高等学校<sup>†††</sup>

## 1. はじめに

Android を携帯電話や組込み OS として利用する機会が増加している。そして、Android 利用者の増加に伴い、Android のアプリケーションやオペレーティングシステムに関する学習をする者も増加している。

Android を組込みシステムに用いるためには、Linux で動作するネイティブプロセスや、DalvikVM 上で動作する Java スレッドをモニタリングしてスケジューリングの挙動を学習することが理解を深めるために有効である。しかし、Linux だけの可視化は様々な物[1]があるが、Android は Linux カーネルと DalvikVM の二つの階層で動作していることから、それらに関連付けて示すことが重要になってくる。また、組込みシステムでは開発・学習環境のセットアップが複雑になりがちであり、これらを容易にすることが必要である。

本報告では Android を対象として、階層化されたシステムにおける OS 資源のロギングおよび可視化について述べる。特に、プロセスに着目し、ロギング、分析、可視化を行う環境を作成した。また、可視化環境をブラウザで実行することでセットアップを容易にすることができた。

## 2. 設計

### 2.1 全体構成

全体構成を図 1 に示す。Android 内にある ftrace と logcat のログデータをサーバへ転送し、ログファイル生成する。データサイズが大きくなるので、ログの保存を Android 内でなく、サーバ内に保存する。そうすることにより、Android 内の保存領域を使用せずに済む。サーバ内で可視化するためにログファイルを加工する。

可視化環境は、ログファイルのデータサイズが大きくなるので、ログファイルの保存領域を

サーバ内に保存する。Android 内で保存する場合、大きなサイズのログファイルの保存領域を取することは難しい。しかし、Android 内にログファイルを保存できる保存領域を持っていない場合でも、サーバに保存することによりログファイルを生成し、保存することが可能である。

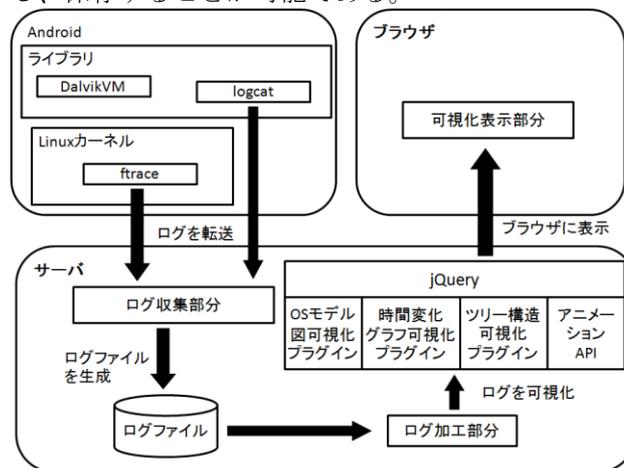


図 1 全体構成

加工したログファイルを JavaScript で可視化し、表示する。可視化のアニメーションには jQuery を使用する。jQuery は既存のプラグインを使用すること以外にも新しくプラグインを作成することができる。本システムでは、三つのプラグインを作成した。状態遷移の可視化を表示する際に使用する OS モデル図可視化プラグイン、時間変化グラフの可視化を表示する際に使用する時間変化グラフ可視化プラグイン、ツリー構造の可視化を表示する際に使用するツリー構造可視化プラグインの三種類を作成した。他にも、アニメーション API を作成した。アニメーション API は可視化プラグインで描画したものをアニメーションで表示するときに呼び出される API である。このアニメーション API は、三つの可視化表現以外に表示したい場合に可視化プラグインを変えるだけで新しい可視化の表示することが可能である。

可視化の実行画面をアニメーションで表示する。アニメーションで表示することにより、ユーザーに可視化の実行結果が見えやすくなる。動的に表現することにより利用者が視覚的に理解できる。

### 2.2 可視化手法

「Development of Visualization Environment for OS on Android」

<sup>†</sup> 「Nakagawa Yuuki・Graduate course Takushoku University」

<sup>††</sup> 「Praween Amontamavut・Faculty of Engineering, Takushoku University」

<sup>†††</sup> 「Nishino Yousuke・Hachioji Soushi High School」

<sup>††</sup> 「Hayakawa Eiichi・Faculty of Engineering, Takushoku University」

Android の可視化部分としてはプロセス管理として次の三つで可視化する。[2]

(1)状態遷移図

プロセスの状態を三種類に分けて表示していく。三つの状態は実行状態、実行可能状態、待ち状態となる。logcat でも生成されているプロセスはプロセス名の文字と色を薄くして表示している。

(2)時間変化グラフ

各プロセスの状態遷移を棒グラフによる表示をする。プロセスの実行状態、実行可能状態、待ち状態は三つの色に分けてグラフ上に表示していく。logcat でも生成されているプロセスはプロセス名の文字と色を薄くして表示している。

(3)ツリー構造

プロセス同士の繋がりを目で見えるようにするために可視化する。プロセスの親子関係やプロセスの生成、消滅をツリー構造で表示していく。logcat でも生成されているプロセスはプロセス名の文字を薄くして表示している。

2.3 プロセス情報取得

Android の可視化を行うためにプロセスの状態遷移や次に実行されるプロセスといった情報を取得する必要がある。その情報を取得するため ftrace と logcat を使用する。プログラムを実行させ、そのプログラムが実行している間のプロセスをトレースして状態遷移などの情報も取得する。

logcat で使用するデータはプロセス ID のみとなっている。logcat のプロセス ID と ftrace のプロセス ID が同じものは Java 言語と C 言語の二つの言語間で動作している。二つの言語間で動作しているプロセスは可視化画面では他のプロセスよりも色を薄くして表示している。

3. 実装

CPU ボードは KZM-A9 を使用し、Android は ver2.2 の Froyo を使用した。

実装結果を図 2 に示す。上が OS モデル図を可視化したものである。四角形のボックスにはプロセス名があり、その上には状態を表示している。プロセスの状態が変わる度にボックスが変わった状態へと移動し、その状態の色に変化する。プロセスが Java 言語とのつながりがある場合はプロセスの状態の色と文字の色を薄くして表示している。

左下が時間変化グラフを可視化したものである。縦軸にプロセス名、横軸に時間を表示している。状態を色で表しており、その色は OS モデル図の色と同じである。プロセスが Java 言語とのつながりがある場合は、プロセスの状態の色と文字の色を薄くして表示している。時間の経過に伴い、グラフを左に動かしていく。そして、時間が正確に判

断できるように自身がグラフ上で知りたい部分にマウスカーソルを合わせるとその知りたい部分の実行時間が表示されることにより、自身の知りたいプロセスの実行時間を知ることができる。

右下がツリー構造を可視化したものを表示している。四角の中にはプロセス名があり、赤い線は上下の繋がりで親子関係を表している。プロセスが Java 言語とのつながりがある場合は、プロセスの文字の色を薄くして表示している。プロセスの生成、消滅をアニメーションで表示している。

可視化の表示には、アニメーションで表示している。アニメーションは表示速度を調整することが可能である。スライダーで表示速度の調整ができるようになっている。

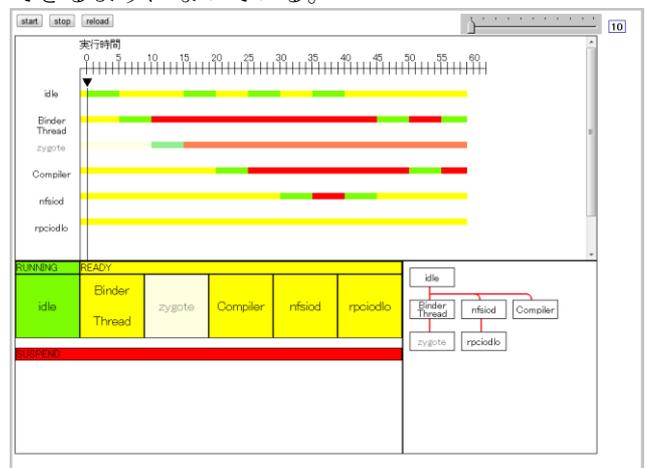


図 2 可視化画面

Android の CPU ボードから動作ログをサーバへ送り、ログファイルをサーバ内で生成、可視化をしてブラウザで表示した。

4. おわりに

Android のプロセスのログデータを取得し、可視化の表示することにより、プロセスの動作を視覚的に理解できるようになった。Android の可視化結果をブラウザ上で表示することにより、複数のマシン、ブラウザで動作が可能になった。

今後の課題としては、可視化の実行画面のユーザビリティが良くなるように改良することや改善点があるのでそれを直していく。DalvikVM 上の Java プロセスの可視化を表示することである。

参考文献

[1] 安藤 友樹、柴田 誠也、本田 晋也、富山 宏之、高田 広章：組込みマルチプロセッサシステムの設計改善支援、SWEST12 (2010) pp.54~56  
 [2] 本橋 大樹、西野 洋介、早川 栄一：組込みシステム学習支援環境「港」における Linux プロセス可視化環境の開発 (コンピュータシステム)、電子情報通信学会 (2010) pp.279~285