TOSHIBA

Leading Innovation

Long-term testing on accelerated Linux kernel

Yoshitake Kobayashi
Advanced Software Technology Group
Corporate Software Engineering Center
TOSHIBA CORPORATION

Mar. 5, 2010

本日の発表のアウトライン

- 問題
- Linuxカーネルの加速方法
- 実装と問題点
- 実験
- まとめ

はじめに

今回の話は・・・

- 「こんなことをやってみた」という内容です

何をしたか

- Linux カーネルの動作を加速してみました
- ．．．．というのは，少し（かなり？）”怪しい”部分を含みます

問題と解決案

問題

- 長期稼動を対象としたテストには時間がかかる

  → 手っ取り早く長期間分のテストをしてみたい

加速！

制限事項

加速できないものは多い

- CPUの最大動作周波数
- ディスクアクセススピード
- ネットワーク通信速度
- . . . . .

ハードそのものは
加速不可

加速できそうなもの

- 時間の進み方

時間の加速にあたって

Linuxにおける時間管理

- jiffies
  - システム起動時から経過したtick数
- xtime
  - 現在時刻と日付を保持

→ **jiffiesが基本**

加速の定義

- 加速 ＝ jiffies × 加速係数

実装

- 実装環境：kernel-2.6.18（Debian/GNU Linux 4.0）

1. Kconfigにパラメータを追加
   - Config SPEEDUP_RATIO（範囲：1～1000）
2. do_timer()関数でごにょごにょ

```
void do_timer(....)
{
    jiffies_64 = jiffies_64 + (1 * speedup_ratio);
    .....
}
```

   - 単に加速係数を掛けているだけ
3. 操作は procfs経由で行う

   例：echo 100 > /proc/sys/kern/accel など

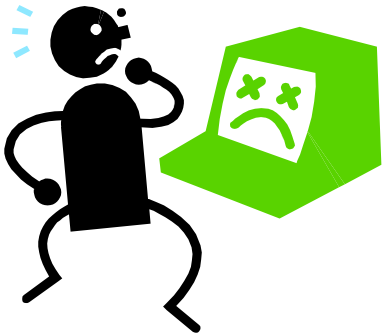
1000倍速で起動!

まともに動かない!

動作不具合現象と原因

1. 現象
   - ファイルシステムがマウントできない
   - デバイスがまともに動作しない
2. 原因
   - カーネル内部のタイムアウト処理
     - デバイスドライバ
     - ファイルシステム
   - ユーザレベルプログラムのタイムアウト

     例：udev



具体例

マウスが動かない

- 以下のカーネルメッセージが表示される
  - Mar 4 00:18:13 accel kernel: psmouse.c: Wheel Mouse at isa0060/serio1/input0 lost synchronization, throwing 1 bytes away.
  - 赤で書いたファイル名は実際には”psmouse-base.c”（コメントのバグ？）

キーボード

- キーボード入力：一瞬で数十文字入力が可能に！
- シリアルコンソールは大丈夫

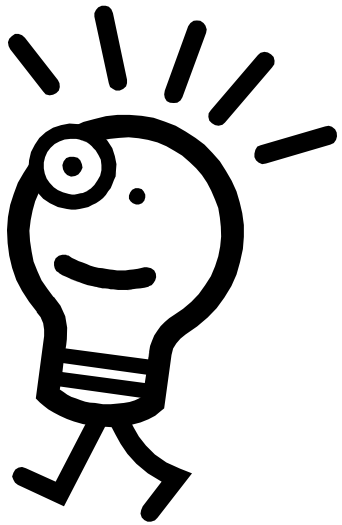
スクリーンセーバー

- 一瞬でブラックアウト

タイムアウトへの対策

1. 方針
   - 対象カーネルのタイムアウト依存部を整理
   - jiffiesを利用している部分を中心にgrepで
2. 方法
   - 加速係数に合わせてタイムアウトを調整
   - （ timeout * speedup_ratio ）でタイムアウトを長く

gnomeデスクトップ環境までそれなりに動いた



実験

★ 簡単なテストプログラムをいくつか用意

- gettimeofday()利用による時間進行確認プログラム
- times()利用による時間進行確認プログラム
- syslog, message, vmstatログ監視プログラム
- とりあえず10年程度走らせてみる（1000倍速で約4日）

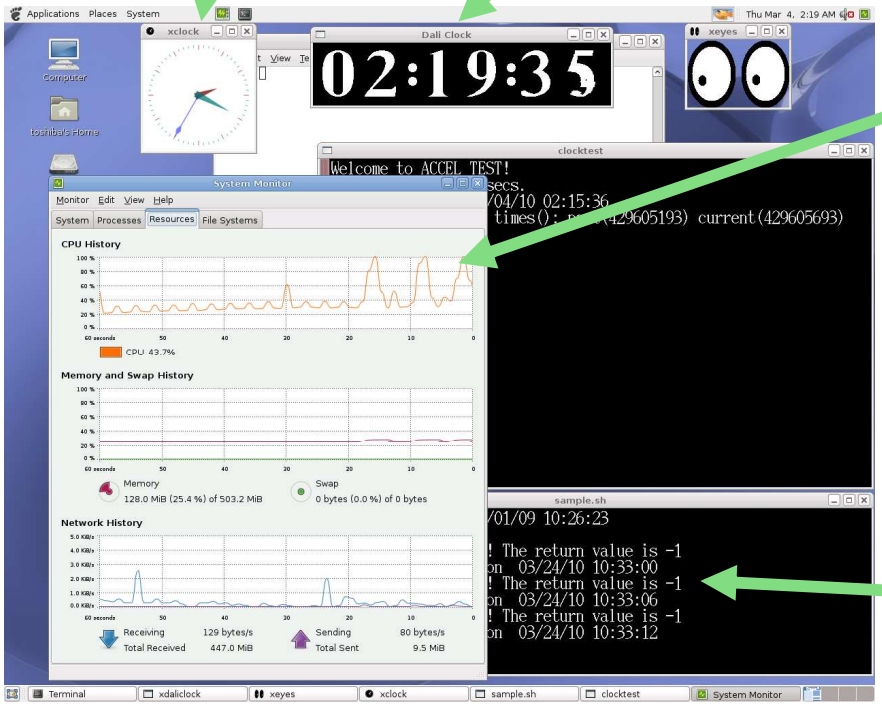
★ 結果

- gettimeofday()利用では特に問題なし
- times()利用ではclock_tオーバーフローを確認
  （マニュアルに書かれている通りの動作）
- syslog, message, vmstatログ監視では特に異常は見当たらない
- ACPIによる電源断も10年経過後も行えるようだ

スクリーンショット

xclockは針がまともに動作せず（スキップ動作）

xdaliclockはそれなりの速度で動作



40倍速程度で
CPU使用率100%に

450日弱で不正な値が
出力されている

まとめ

Linuxカーネルを加速してみた

- 本当に（ハードウェアが）速くなるわけではない
  - 物理的に加速したのではないので限界あり
  - 加速したのは時刻のみ
- ソフトウェアの実行は”見た目”で早く感じることもある
- clock_tのオーバーフロー問題は，1日程度で再現可能
- カーネル自体は10年動作させても大丈夫そう
- 長期稼動向けの試験に応用することを期待
- 他の有効利用方法についてはアイデア募集中です


TOSHIBA
Leading Innovation