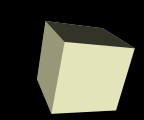
Embedded Optimization
(1) Fast Booting
(2) Starvation-Free Realtime Scheduler

Masahiko Takahashi
Akira Tsukamoto

System Platforms Research Laboratories,
NEC Corporation

2006/01/20



Embedded Optimization
(1) Fast Booting

～ 高速ネットワークブートの実装の経験から
組込み向け高速起動へのアプローチ ～

高橋 雅彦

日本電気株式会社
システムプラットフォーム研究所

2006/01/20

Outline

- 背景：個々のPCのOSなどをアップデートするのが大変！
  → ネットワークブートによる一元管理に着目
    - 最近では情報漏洩対策の意味合いも
- 通常のネットワークブートでは遅い
  → 高速起動方式の研究
    - メモリイメージを活用
    - 起動時には最小限の部分のみを取得
- x86のPC向けに作ってみた！（2004年秋）

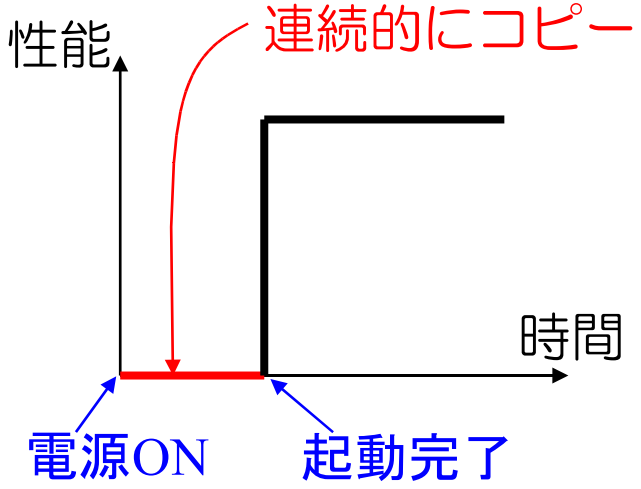
- 提案方式の特徴は、組込み向けにも有効なのではないか?
  - 今後、実装予定
- 高速ネットワークブートを実装してみて、メモリイメージ活用型高速起動方式の課題が見えてきた

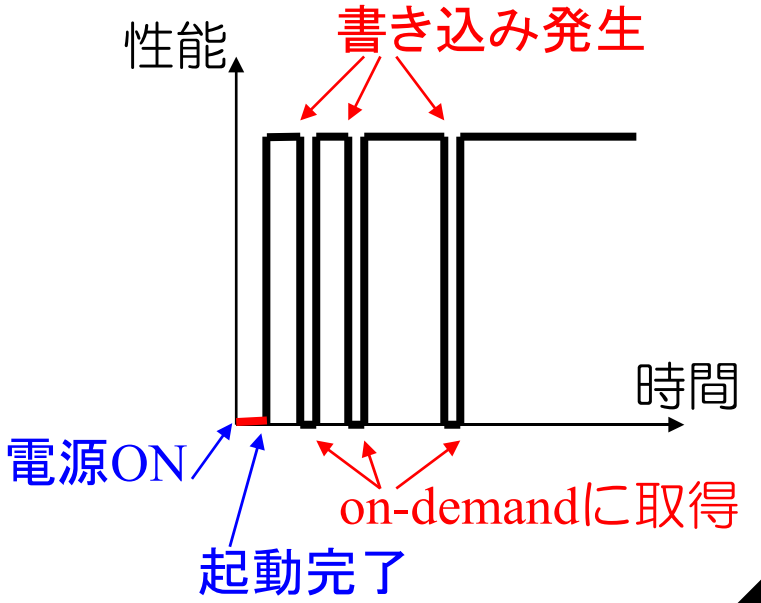
高速ネットワークブート方式とは?

- OSと必要なアプリケーション(デーモン)が起動した直後のメモリイメージをネットワーク経由で取得することでシステムを起動する方式
  - アプリケーションが固定的なシステム(サーバや組み込み機器など)に適した起動方式
- 起動時には、メモリイメージのうち起動時に必要なメモリのみを取得する→起動時の時間短縮
- メモリイメージの残りは、最初のアクセス(ページ単位)のときにon-demandで取得する
  - ページフォルトを発生させるために、これらのページはページテーブル上のpresent bitを落とす

ハイバーネーション方式との比較

- ハイバーネーション
  - 起動時にROM/HDDからRAMへのコピーが連続的に発生する
  - 使用RAM量に比例したコピー時間がかかるため起動完了までに時間がかかる
- 提案方式
  - 起動時には最小限の取得で済むので、短時間で起動が完了
  - 未取得ページへの最初のアクセスが発生したときにon-demandで取得
    - 実行時に若干の性能低下はある





実装構成

サーバ

・TFTPサーバ

・メモリイメージ

(起動時取得部分)

(on-demand取得部分)

ディスクレス・クライアント

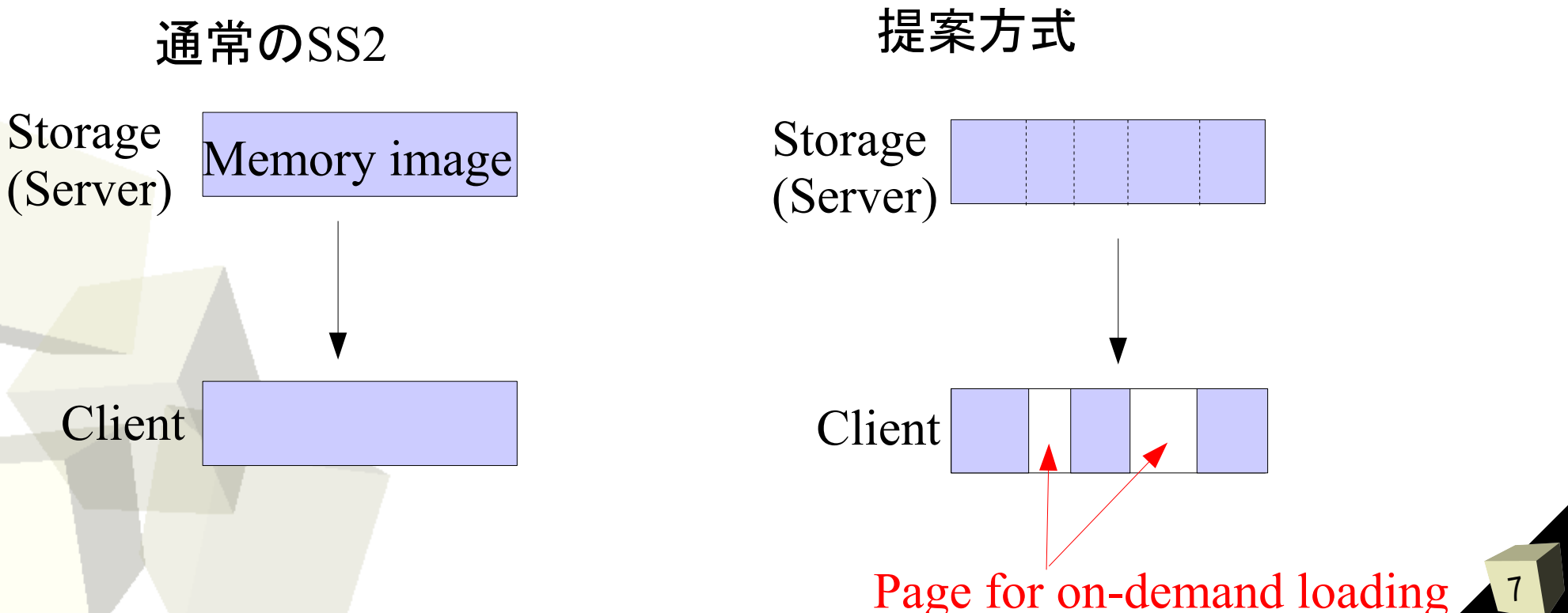
・Linux (2.6.6)
TFTPクライアント

・SoftwareSuspend2
(2.0.0.72)

注：転送スループットをかせぐため、
TFTPのブロックサイズはRFC2348の
最大値65464(61440)byteにしている。

SoftwareSuspend2の改造

- 起動時に必要なページと起動時には必要でないページとを区別する
  - 基本的にはpageset1が前者で、pageset2が後者



Linuxの改造

- ページテーブルの書き換え
  - メモリイメージ作成時、pageset2に含まれるページのページテーブルエントリのpresent bitを落とす処理
- ページフォルトハンドラの改造
  - 起動時に取得しなかったページへのアクセスが発生したときにon-demandに取得する処理

  注：正規のページフォルトと区別する必要あり
- TFTPクライアント機能の追加
  - TFTPサーバからメモリイメージを取得する処理（起動時及びon-demand）

性能測定

- 環境
  - Server：Pentium4 2GHz, 1024MB(PC133), eepro100
  - Client：Pentium4 2GHz, 512MB(PC133), eepro100
  - PXE/DHCP/TFTP/NFS root filesystem
- 「X Window + xterm + blackbox」のメモリイメージを作成
- 測定結果
  - 電源ボタンを押した直後からのキーボードが入力可能になるまでの時間
    - 提案方式(高速ネットワークブート)：25秒
    - 通常起動(ネットワークブート)：60秒

組込み向け高速起動への応用の検討

- メモリイメージを活用して組み込みシステムを高速起動する方法は既に提案されている
- 一方で、ハイバーネーションの課題は残っている
  - ″起動時のメモリコピー(イメージ転送)時間が長い″

→ 提案方式が組込みシステムでも使えないだろうか？
  - 起動時に一気に転送するのではなく、on-demandに取得することで、起動時のコピー量を削減
  - 今後、実装予定

メモリイメージ活用型のデメリット

- 顕在化する課題(例)
  - 書き込み可能なファイルシステムをマウントしている場合、ファイルシステムとメモリイメージが不整合になる
  - 個人設定などのカスタマイズに対応するのが難しい
  - アプリケーションが特定のHWの初期化を行なう場合、そのHWの初期化が出来ない可能性がある
  - ハードウェア(HW)構成が変わった場合はメモリイメージを作成し直さなければならない

→ アプリケーションも含めて、システムの作り方を考える必要あり