

Linuxカーネルサイズ・使用メモリ検証／改善PJ

2006年6月29日
NEC OSS推進センター
池田 宗広
m-ikeda@ds.jp.nec.com



- プロジェクトの概要
- 実行ステップとステータス
- 自動検証ツールの概要
- これまでの検証結果
- 課題
- 今後の予定



プロジェクトの概要 (1)

- **組み込み領域におけるLinuxカーネルへの要求**

- サイズ・使用メモリを極力最小化したカーネルを作成したい。
- カーネルのサイズ・使用メモリ量を改善したい。

- **要求実現へのアプローチ**

- サイズ・使用メモリを最小化したカーネルを作成するためには、カーネルconfigの絞りこみが最重要。
- サイズ・使用メモリを改善するためには、改善すべき部分の明確化が必要。
「改善すべき部分」の大部分はconfig項目にマッピングできる。

→ 本プロジェクトの目的は、Linuxカーネルの各config項目がサイズ・使用メモリに及ぼすインパクトを検証・公開し、サイズ・使用メモリ最適化構成および改善の材料を提供すること。
実際の改善まで行えればなお良し。



プロジェクトの概要 (2)

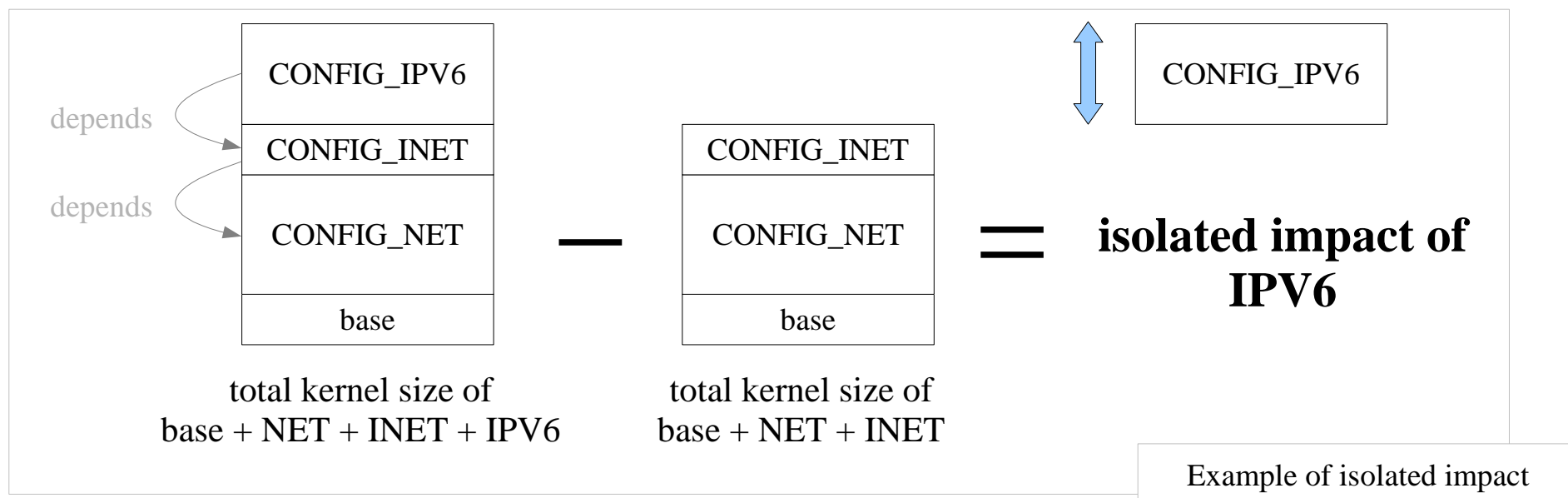
• 検証の方法

– 自動測定

- 検証対象Config項目リストにしたがって、.configファイルを生成する。
- 各.configファイルからカーネルをmakeし、サイズを測定する。
- カーネルをインストール、リブートし、ブート直後にカーネルが消費しているメモリ量を測定する。

– 「インパクト」の算出

- 測定結果から、測定対象Config項目それぞれのインパクトを算出する。
- ここで言う「インパクト」とは、下図に示す ISOLATED IMPACT のこと。



プロジェクトの概要 (3)

• アウトプット

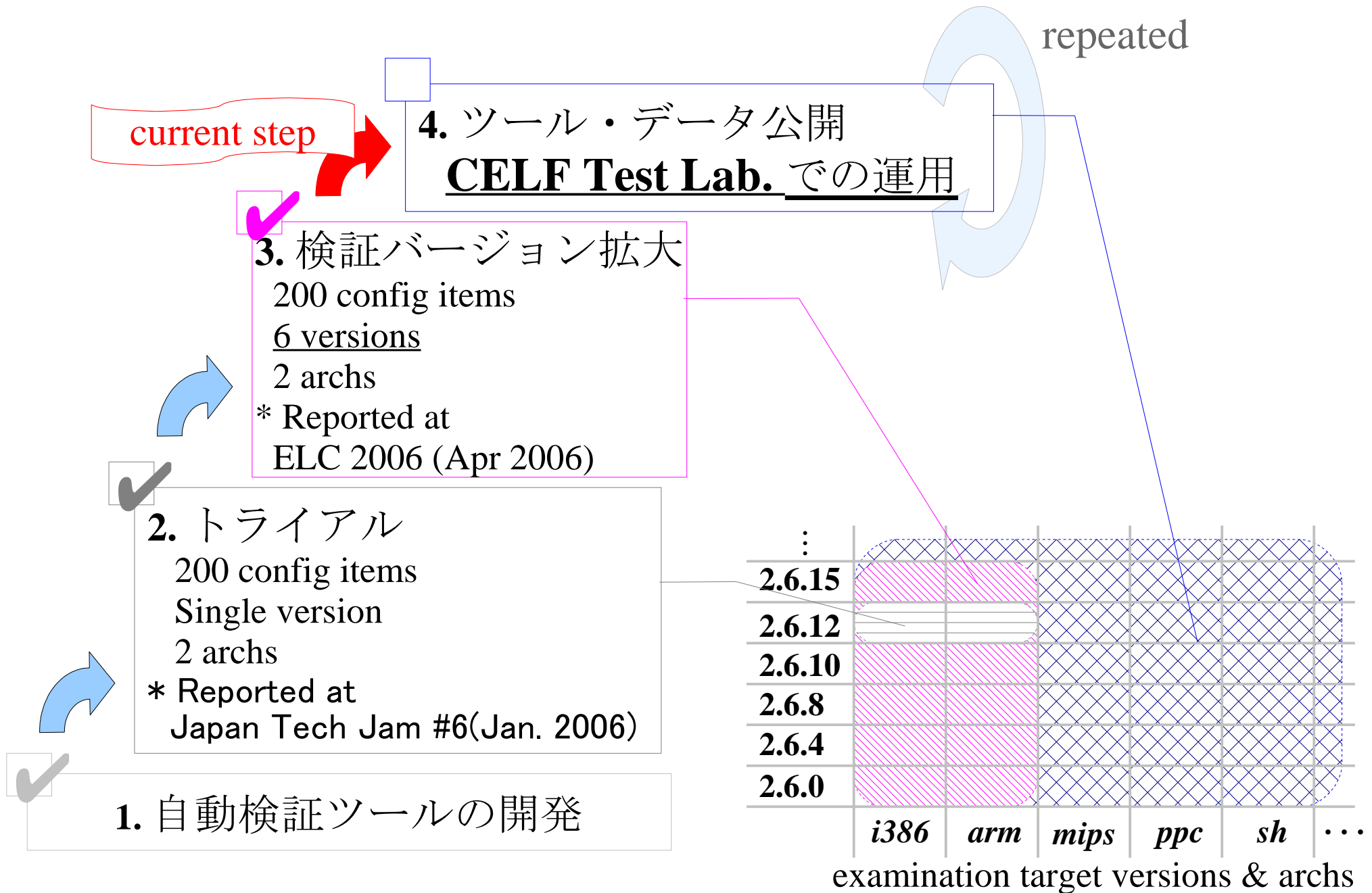
- 自動検証ツール・測定結果解析ツールを開発・公開する。
 - 組み込み用途でメジャーなConfig項目についてサイズインパクトを検証し、結果を公開する。
 - 同じくメモリ消費量インパクトを検証し、結果を公開する。
(メモリ消費量はブート直後のカーネルが消費するメモリの量)
- 全てCELF Wikiにて公開。
<http://tree.celinuxforum.org/CelfPubWiki/KernelConfigWeight>

• どう役にたつか？

- 組み込み技術者にとって
 - カーネルをビルドする前に、サイズと消費メモリを見積もることができる。
- カーネル開発者にとって
 - バージョン間の比較データから、カーネルの肥大化部分＝改善が必要な部分がわかる。

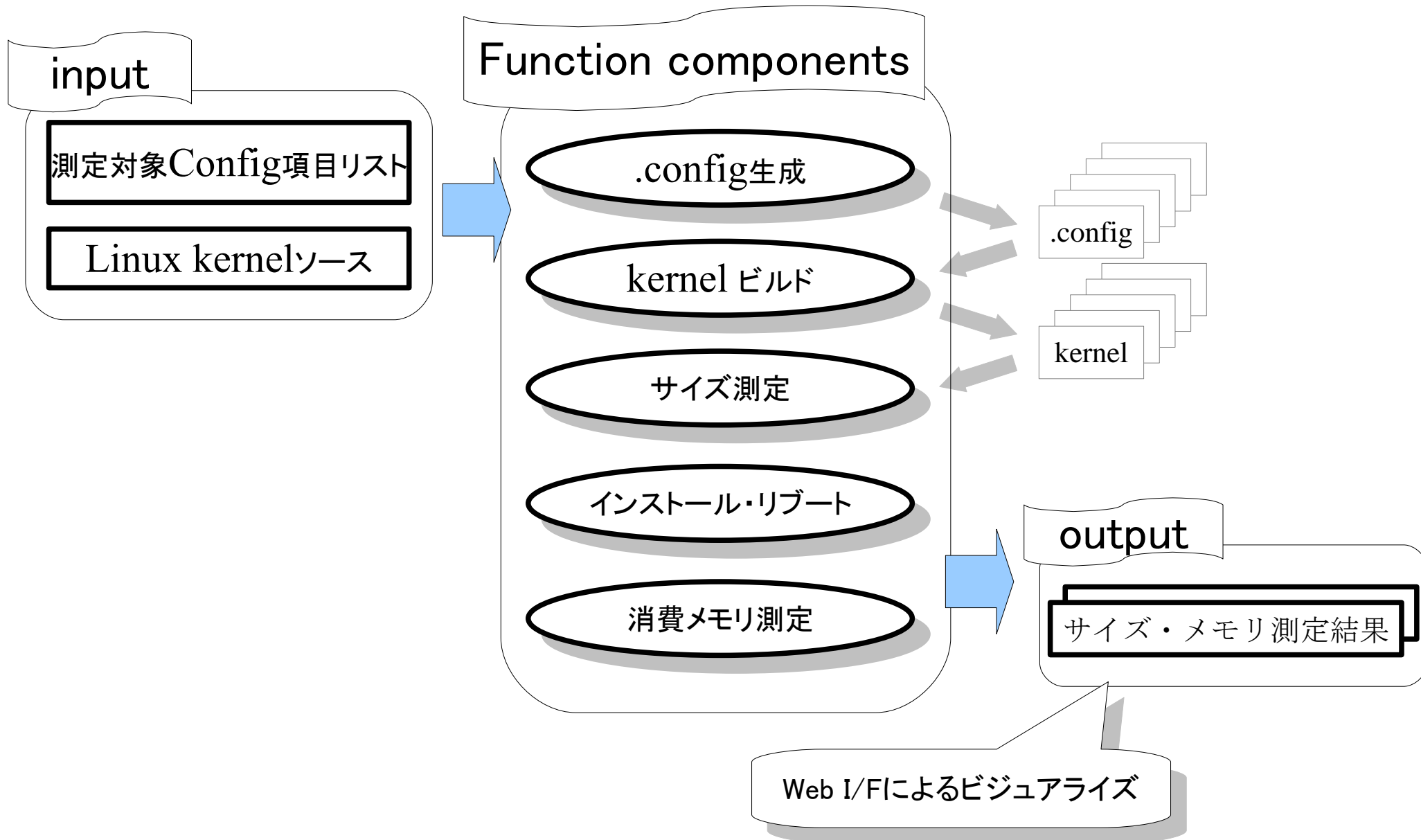


実行ステップとステータス



自動検証ツールの概要 (1)

構成



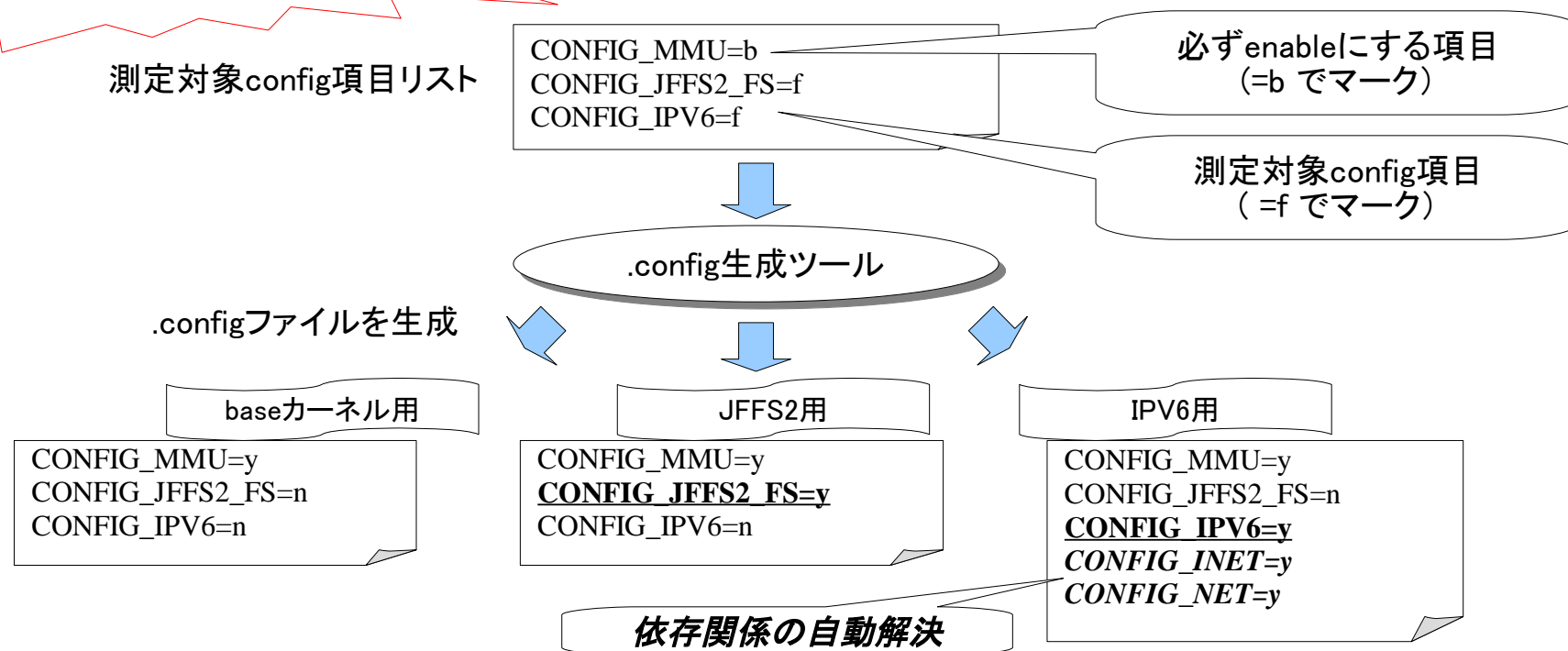
自動検証ツールの概要 (2)

測定対象config項目リストと生成されるカーネルとの関係

- .config生成ツールはリストを読み込み、測定対象config項目を一つ一つenableにした.configファイルを生成する。
 - config項目の依存関係は.config生成ツールが自動的に解決。
- 各.configファイルから測定用カーネルがビルドされる。

※.config生成ツールはLinuxカーネルソースの一部を流用して開発。

オープンソース・パワー



これまでの検証結果 (1)

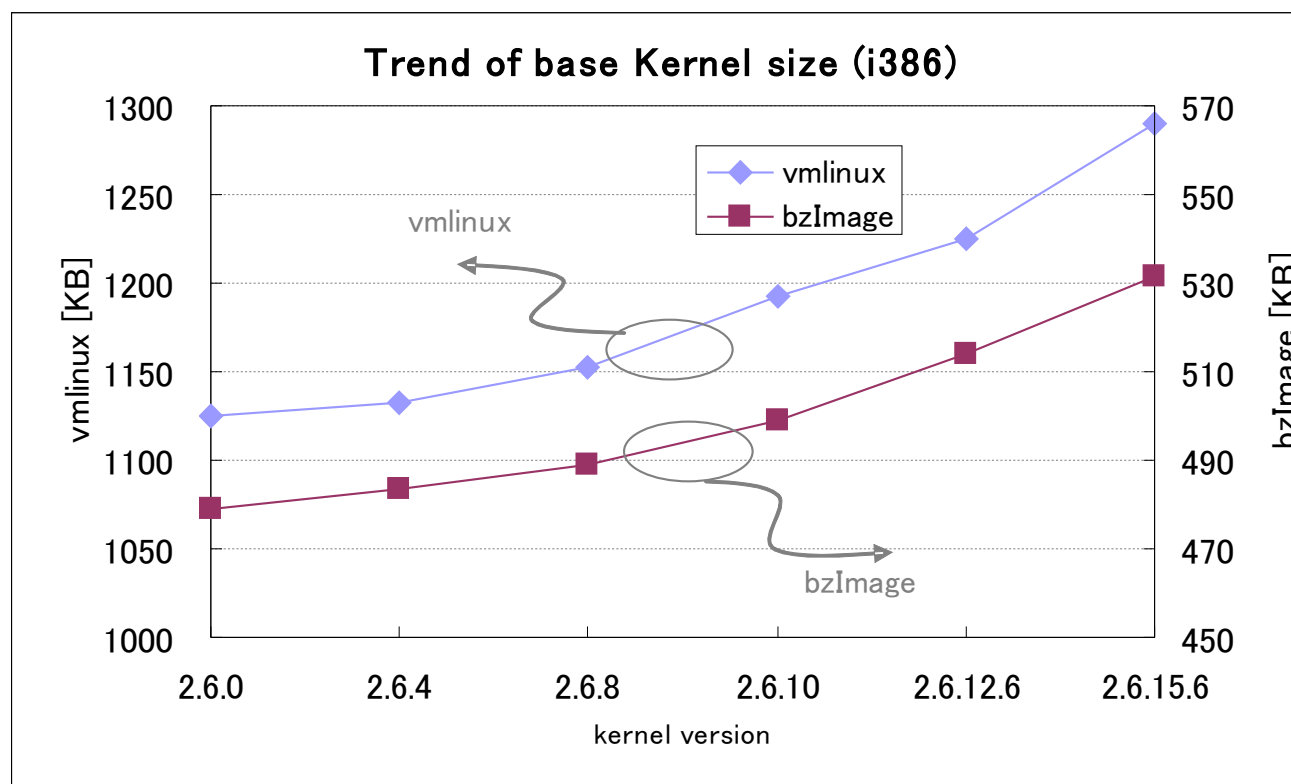
• 検証条件

- “ベースカーネル”
 - ブート可能な最小限のconfig項目のみenable。
 - Uni-Processor用。
 - ファイルシステムは ext3、sysfs、procfsのみ。
- 検証対象config項目
 - 組み込み用途でメジャーと考えられる約200項目について検証。
 - 単独の影響が抽出できないものは検証対象から除外。
(例 : smp, pm, printk support etc.)
- 対象カーネルバージョン :
2.6.0, 2.6.4, 2.6.8, 2.6.10, 2.6.12.6, 2.6.15.6
- 対象アーキテクチャ : i386, ARM



これまでの検証結果 (2)

- 静的サイズ：ベースカーネルサイズのバージョン間比較
 - 2.6.0 から 2.6.15 でサイズは約10%増加。
 - vmlinux : 150[KB]
 - bzImage : 50[KB]



Note(1) : gcc 3.3.5, binutils 2.15

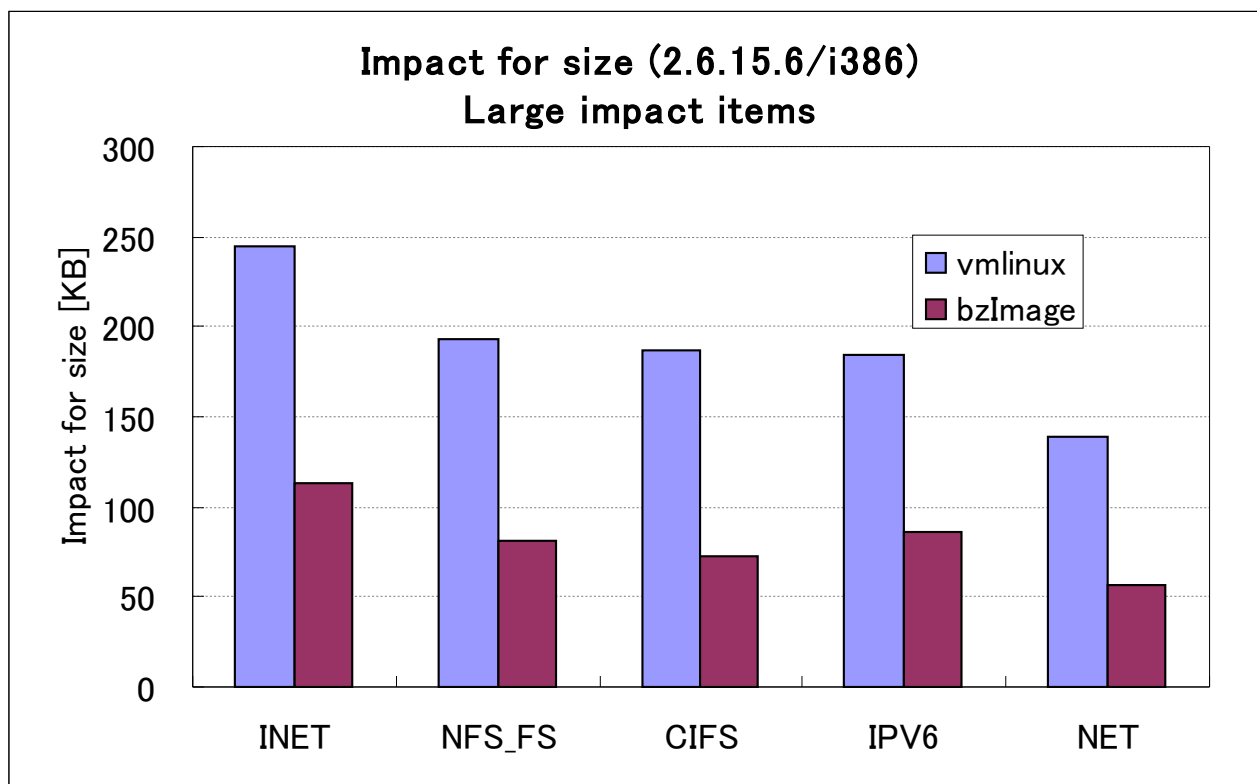


これまでの検証結果 (3)

- 静的サイズ：各config項目のインパクト

 - Top 5

 - サイズインパクトが大きいのはネットワーク関連機能。



Note(1) : gcc 3.3.5, binutils 2.15

Note(2) : Some config items are not completely isolated.

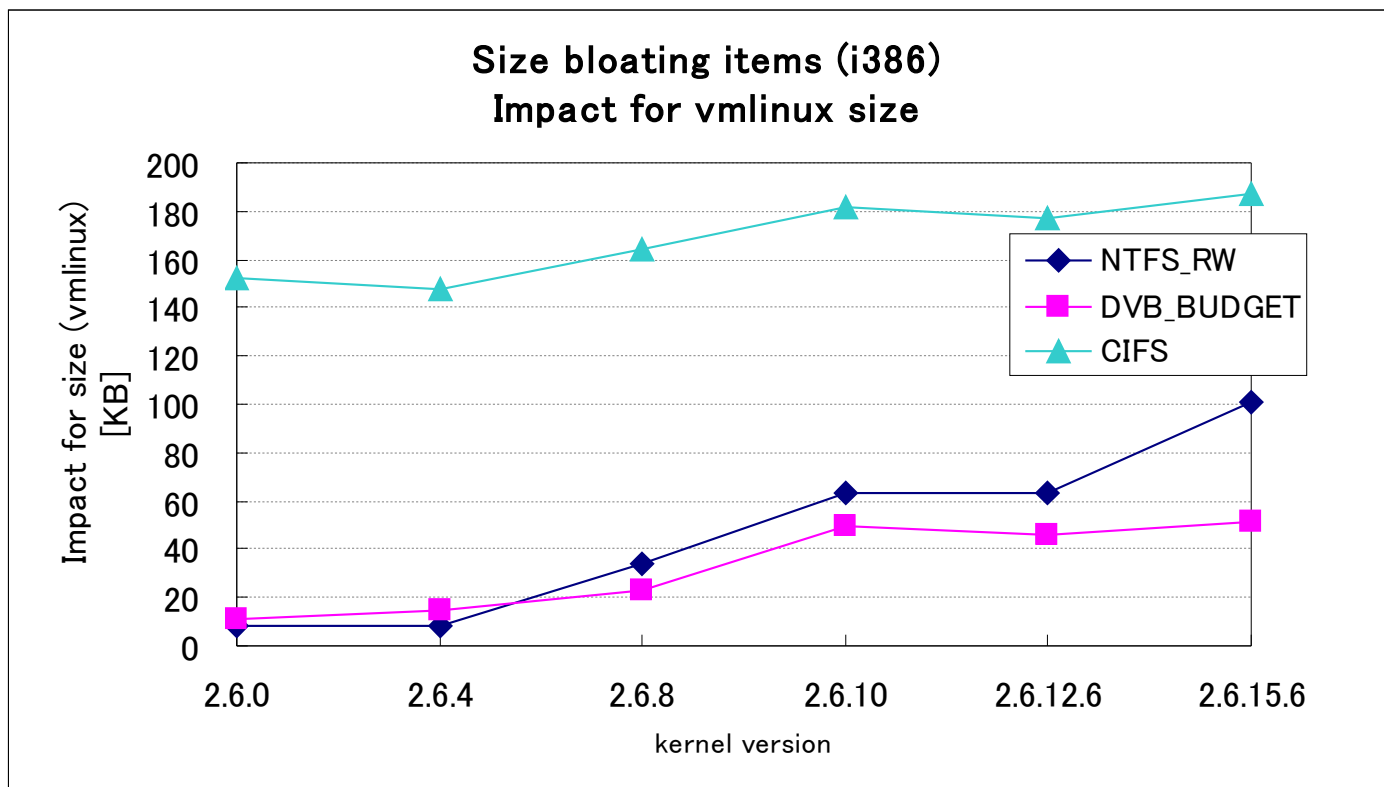
INET += FIB_HASH + TCP_CONG_BIC

NFS_FS += LOCKD + SUNRPC + NFS_COMMON



これまでの検証結果 (4)

- 静的サイズ：サイズインパクトのバージョン間比較
 - 2.6.0～2.6.15で最もサイズが増えたのは、NTFS_RW、DVB_BUDGET、CIFS。

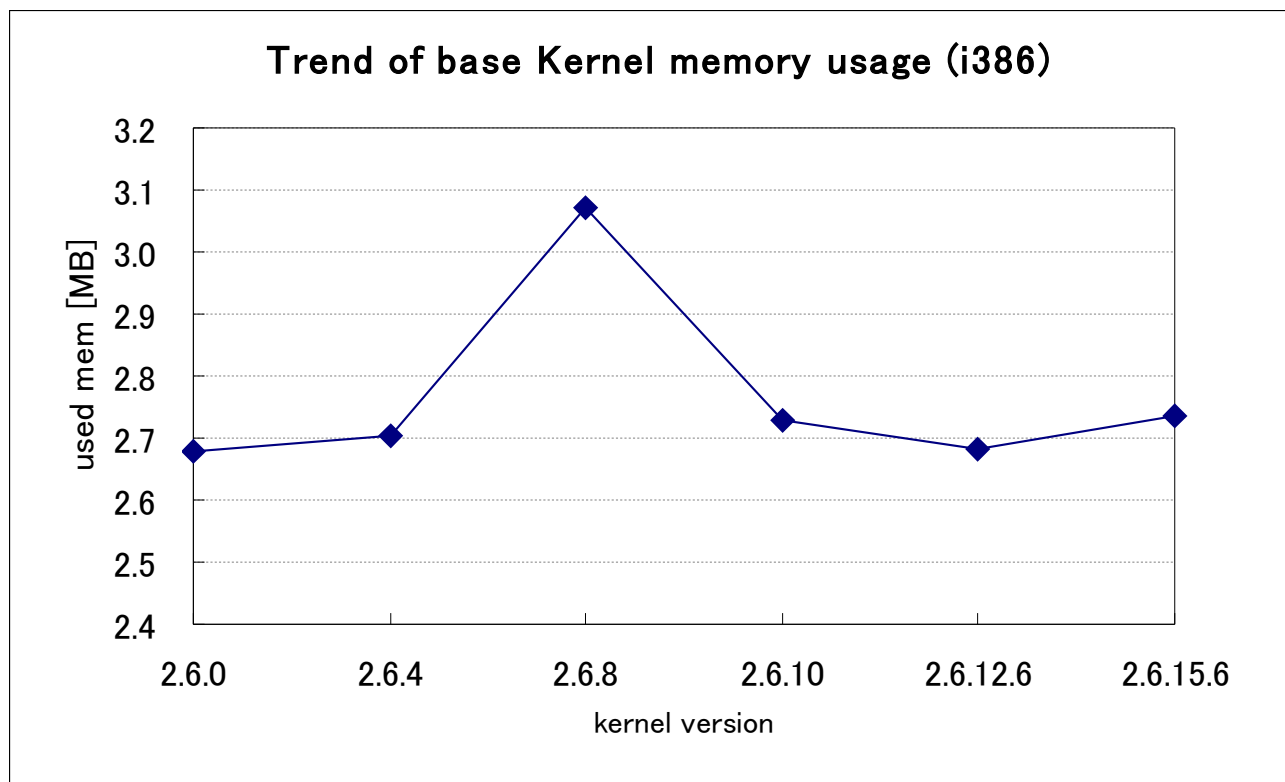


Note : gcc 3.3.5, binutils 2.15



これまでの検証結果 (5)

- メモリ消費量：ベースカーネルメモリ消費量のバージョン間比較
 - カーネルのメモリ消費量は2.6.0 から 2.6.15 でほとんど増加なし。
 - ただし 2.6.8 にピークあり。(原因は未検証)



Note(1) : gcc 3.3.5, binutils 2.15

Note(2) : Target machine ... Pentium IV 2.2GHz/RAM 256MB/HD 40GB

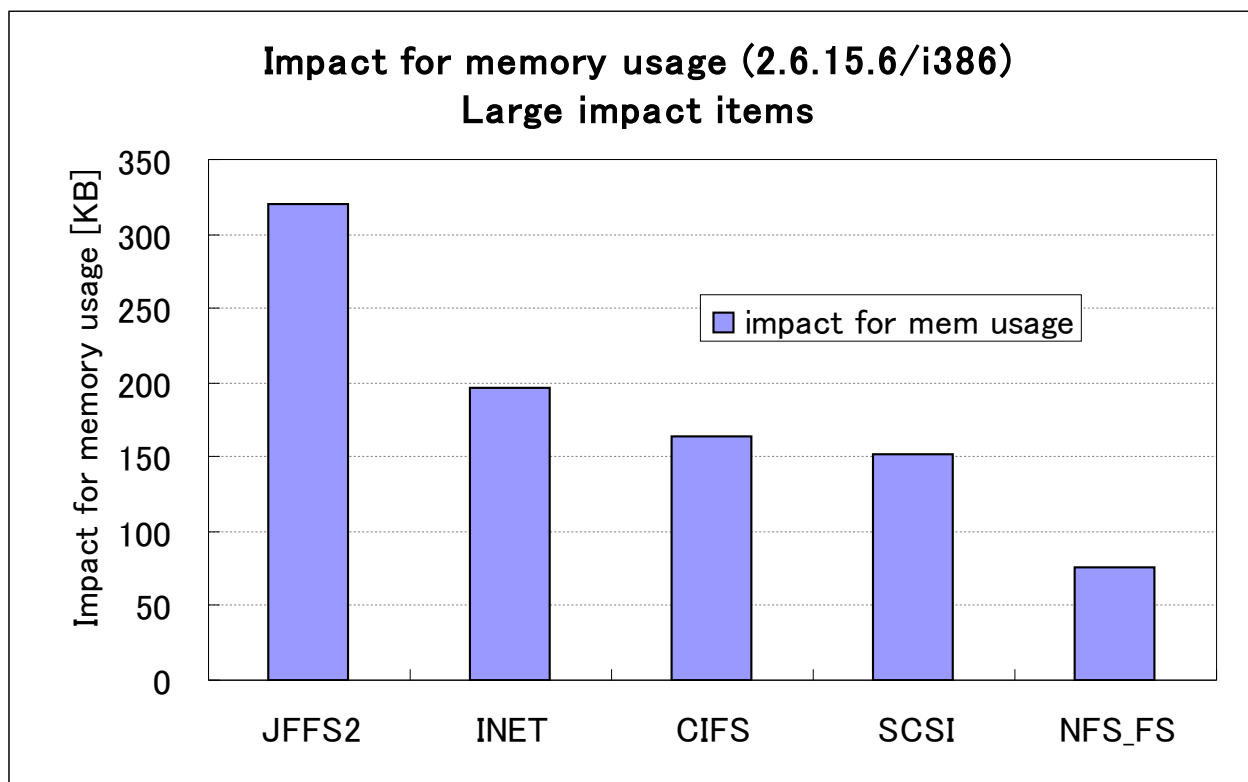


これまでの検証結果 (6)

- メモリ消費量：各config項目のインパクト

- Top 5

- JFFS2、ネットワーク関連機能、SCSIのメモリインパクトが大きい。



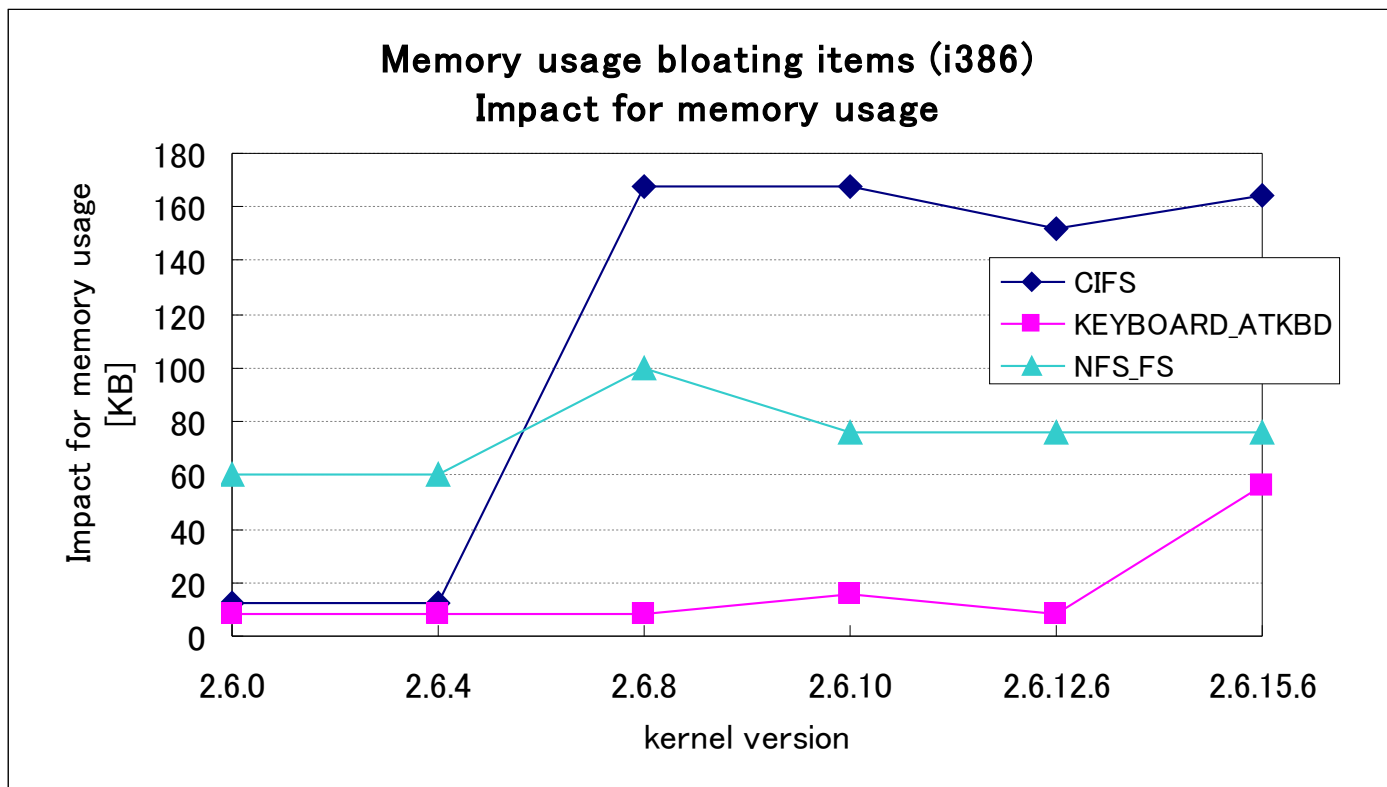
Note(1) : gcc 3.3.5, binutils 2.15

Note(2) : Target machine ... Pentium IV 2.2GHz/RAM 256MB/HD 40GB



これまでの検証結果 (7)

- メモリ消費量：メモリ消費量インパクトのバージョン間比較
 - CIFS、KEYBOARD_ATKBDのメモリ消費量が増加。



Note(1) : gcc 3.3.5, binutils 2.15

Note(2) : Target machine ... Pentium IV 2.2GHz/RAM 256MB/HD 40GB



これまでの検証結果 (まとめ)

• サイズ検証結果

- ベースカーネル(ブート可能な最小限の設定を行ったカーネル)のサイズは、2.6シリーズ内で約10%の増加。
- 機能別に見ると、最も肥大化しているのはNTFS_RW。
- サイズインパクトが大きいのはネットワーク関連機能 (INET, NFS_FS, IPV6 etc.)。

• メモリ消費量検証結果

- カーネル全体のメモリ消費量はほとんど増加なし。
- 2.6.8だけ特異的に消費量が多い。(原因は未検証)
- メモリ消費量へのインパクトが大きいのはJFFS2。



これまでの検証結果（ツールによる可視化）

- サイズインパクト、メモリインパクトをWebインターフェースでグラフ表示するツールを作成中。
(⇒ デモ)
- “Bloatwatch” を参考に作成。
 - Bloatwatch
 - 開発 : Matt Mackall (Mercurial, LinuxTiny の開発者)
 - built-in.o を size, nm で測定・集計。
 - 結果はWebインターフェースでグラフ表示。
 - サブシステムごと、シンボルごとのサイズ表示が可能。
 - バージョン間の比較も可能。
 - URL : <http://www.selenic.com/repo/bloatwatch>
- CELF Embedded Linux Conference (2006年4月)にて Matt Mackall と協議。
 - Bloatwatch : シンボルごとのデータ ... カーネル開発者視点
このPJ : config項目ごとのデータ ... 組み込み技術者視点
⇒ うまく協力して補完していこう、ということに。

オープンソース・パワー



• メモリ消費量の計測方法

- 現状の「カーネルのメモリ使用量」とは何か
 - 定義：
(バディシステムが「使用中」と認識しているメモリ) - (CPUキャッシュ)
 - 計測タイミング：
カーネルブート直後 (initプロセス起動直前) にカーネル内部で計測。
※ CPUキャッシュ : 1ページ分の確保を高速化するために、各CPUがあらかじめバディシステムから確保しておくバッファ領域。
※ カーネルのコード領域と静的データ領域は含まない。
 - ユーザープロセス実行時のメモリ使用量は、ユーザープロセスの挙動に大きく影響される。
 - メモリ確保 ⇒ プロセスメモリ空間への割付によりメモリ消費
 - ブロックデバイスへのアクセス ⇒ ページキャッシュによりメモリ消費
 - ユーザー(組み込み技術者)から見て、ブート直後の値に意味があるかどうかは疑問。
- ➔ 今後、組み込み用途でのユーザープロセスの挙動を考慮したメモリ管理手法の最適化を検討する予定。
- ユーザープロセスの挙動をモデル化して計測手法を確立
 - 組み込み向けに最適化したメモリ管理手法の開発



今後の予定

- ツールブラッシュアップ

- 検証結果可視化ツールは引き続き開発・改善。
- 組み込み技術者にとっては、

.configファイルを入力するとenableになっている各config項目のインパクトを自動的に検証する

という機能があると便利？（要否検討中）

- ツール・データ公開

- 可視化ツール完成、ドキュメント整備後CELF Wikiにて公開。

- CELF Test Lab. での公開運用

- 検証アーキテクチャの拡大、保有データの拡充を図る。



Empowered by Innovation

NEC

