Empowered by Innovation NEC

About a Tool to check the Kernel image size and memory consumption

Jan. 20th, 2006
NEC Linux Promotion Center
IKEDA Munehiro
m-ikeda(_at_mark_)ds.jp.nec.com

0. Contents

1. Background (1)

- **Evaluation and analysis of LinuxTiny (July, 2005)**

Sort of Kernels	vmlinux[KB]	bzImage[KB]	free mem[MB]
vanilla-full	7709	3071	78.6
vanilla-size	2833	1061	85.0
tiny	2360	869	86.0

- **Sort of Kernels**
  - vanilla-full : Almost default setting.
  - vanilla-size : Enabling least drivers, fs, and CONFIG_CC_OPTIMIZE_FOR_SIZE.
  - tiny : Applied LinuxTiny. All configuration items added by LinuxTiny are set to reduce the Kernel size.
- **Measurement conditions**
  - Kernel 2.6.10
  - x86(PC conpatible)
  - gcc 3.3.5
  - Machine : Celeron 400MHz / 96MB RAM / 10GB HD
  - free mem was measured by free(1) just after reboot, unless daemons' running.

LinuxTiny works well to reduce the Kernel size.

But, investigation of Kernel-configuration is much more considerable.

1. Background (2)

- **How to investigate Kernel-configuration to reduce the Kernel size and memory consumption...**

  1.make menuconfig
  2.make
  3.measure size
  4.install and reboot
  5.free(1)

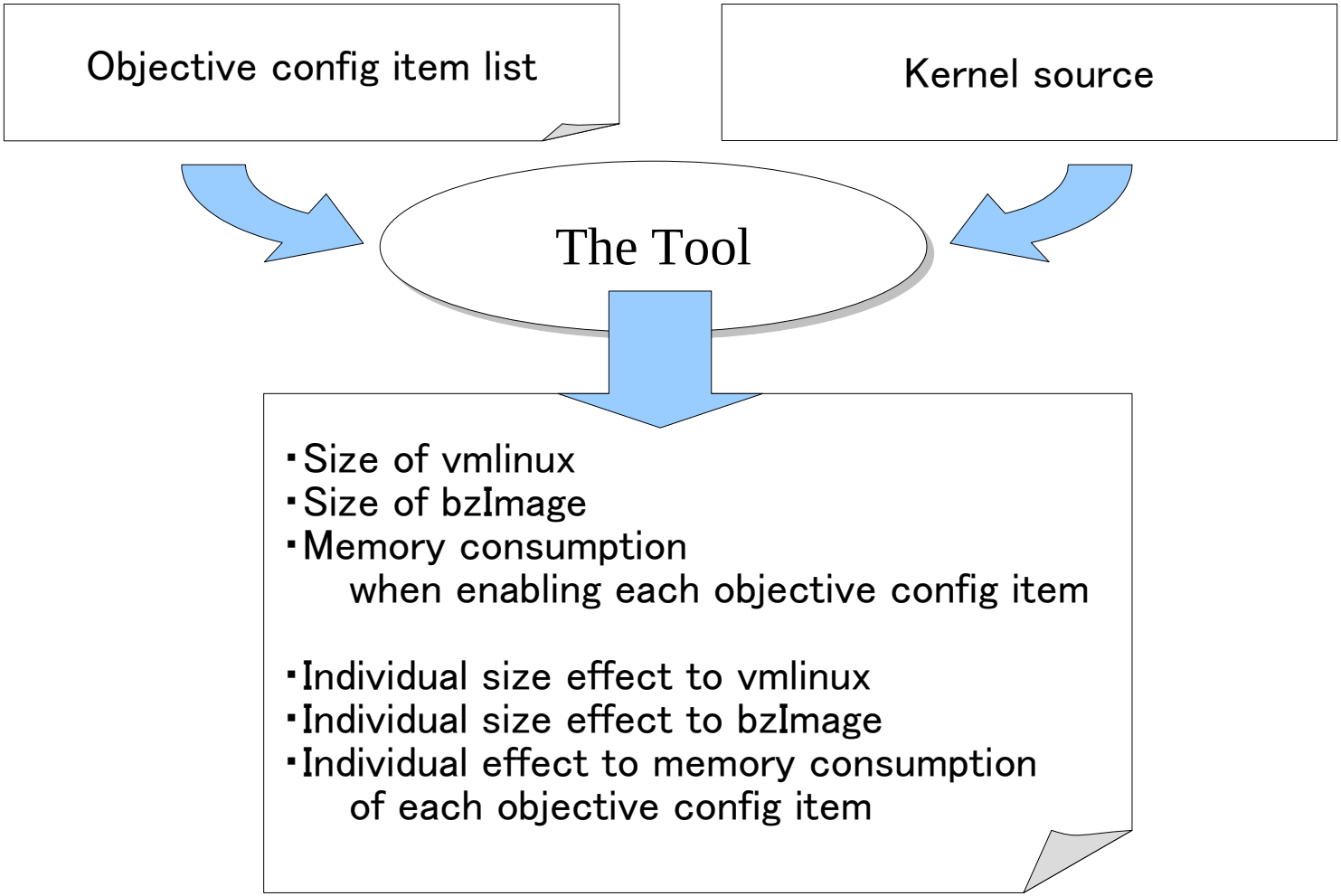
- **Num of configration items (2.6.12.3)**
  - Total : 5338 (general 3028 + arch 2310)
  - ARCH=i386 : 3187 (general 3028+ arch/i386 159)
  - ARCH=arm : 3314 (general 3028 + arch/arm 286)

It makes everyone HAPPY to develop a tool which automatically measures size and memory consumption effects of each configuration item !

Motivation to develop The Tool

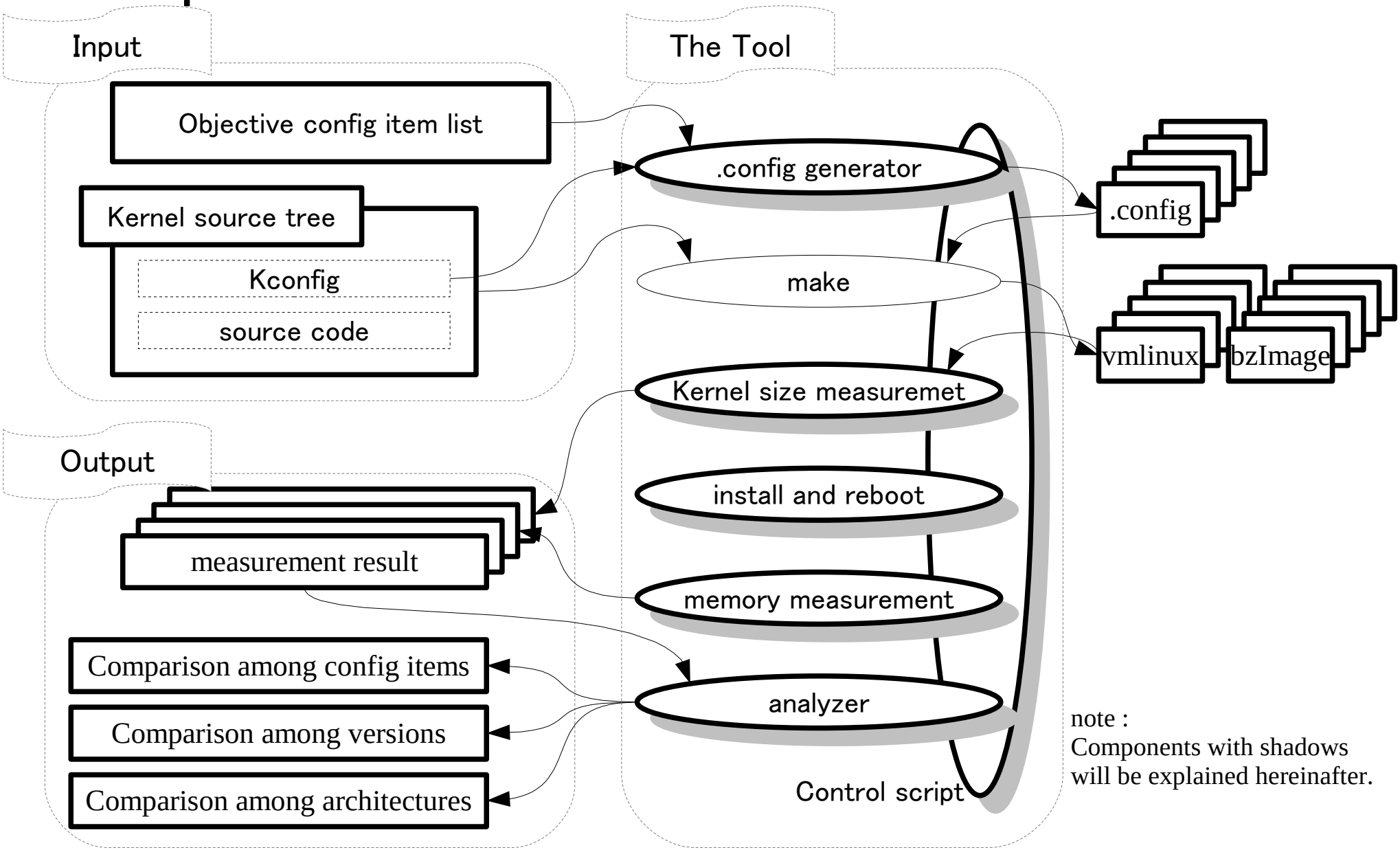
2. What is The Tool ?

- **The Tool is a combined tool to measure sizes and memory consumptions associated with specified kernel configurations.**



3. Features of The Tool (1)

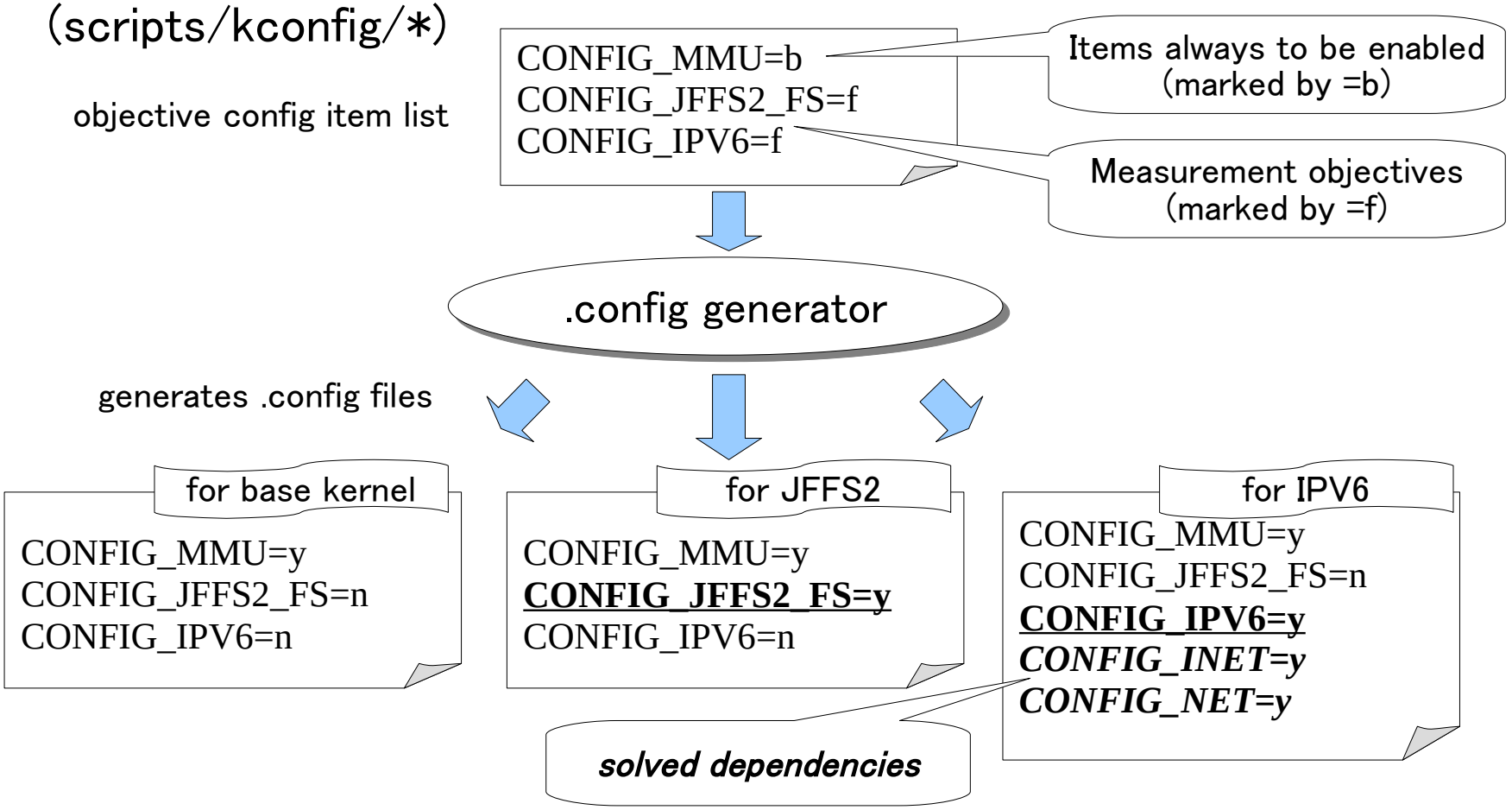
- **Components**



note :
Components with shadows will be explained hereinafter.

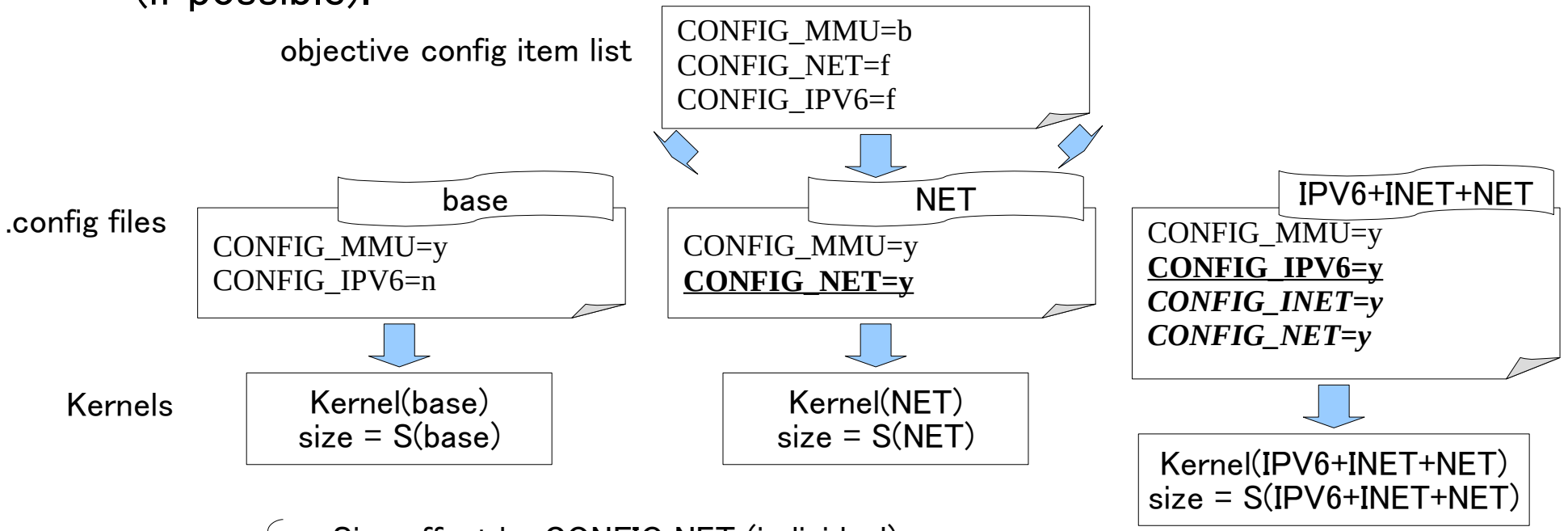
3. Features of The Tool (2)

- **.config generator**
  - Reads objective config item list, and generates .config files which contain enabled objective configuration item one by one.
  - Automatic dependency solving.
  - The code is diverted from Kconfig parser in Kernel source tree. (scripts/kconfig/*)



3. Features of The Tool (3)

- **Kernel size measurement**
  - Measures size of kernel image file(vmlinux) and compressed image file (bzImage)
  - Calculate individual size-effect by an objective configuration item (if possible).



○possible to calculate
- Size effect by CONFIG_NET (individual)
  ⊿s(NET) = S(NET) − S(base)
- Size effect by CONFIG_IPV6+CONFIG_INET
  ⊿s(IPV6+INET) = S(IPV6+INET+NET) − S(base) −⊿s(NET)

✗IMPOSSIBLE to calculate
- Size effect by CONFIG_IPV6 (individual)
  ⊿s(IPV6) = S(IPV6+INET+NET) − S(base) − ⊿s(INET+NET)
  ...impossible because ⊿s(INET+NET) is not in the result.

3. Features of The Tool (4)

- **Install and reboot**
  - To reboot, controls target power line with relay-box connected to the parallel port of host machine.
- **Memory measurement**
  - Measures memory consumption of the kernel using sysinfo(2), and records output in a result file.
    (The reason of using sysinfo instead of free(1) is for the system which doesn't support procfs)
  - Calculate individual memory-effect by an objective configuration item (if possible).
- **Control script**
  - Runs test loops for architectures, Kernel-versions and objective configuration items.
  - Recording status for resuming broken test loop by some reboot trouble.
- **Analyzer**
  - Compares sizes and memory consumptions among objective configuration items.
  - Compares sizes and memory consumptions among Kernel versions.
  - Compares sizes and memory consumptions among architectures.

4. What is The Tool for ?

- **For embedded system developers**
  - **“There are drivers or filesystems whose functionalities are similar … I want to select the smallest one. Which should I select?”**
    ➔You can select one beeing guided by the measurement data.
  - **“I have a plan to use a new version kernel… but how will it be larger than it is?”**
    ➔You can estimate their difference comparing measurement data from the versions.
- **For Kernel developers**
  - **“Where is an inefficient implementation in the kernel?”**
    ➔You can point out the part of the kernel investigating the measurement data.
  - **“How much size does the patch effect to the kernel?”**
    ➔You can evaluate the effect to run the tool automatically.

5. Measurement preparation – The base kernels – (1)

- **Base kernel**

  We need a reference kernel (=base kernel) which contains NO enabled objective configuration item to measure size and memory consumption effects of an objective item.

- **Three kinds of configuration items**

  There ara 3 kinds of configuration item below.

  1. **Mandatory items to boot.**
     ex : designation of CPU architecture, driver for the rootfs device, etc.
  2. **Independent config items**
     One whose size can be measured individually.
     ex : most of drivers, fs, etc.
  3. **Dependent config items**
     One whose size can NOT be measured individually.
     (The size is dependent on settings of other items)
     ex : SMP support, printk support, procfs, size optimization, etc.

- **Pattern of independent configuration items**

  We need to investigate and apply patterns of “independent items” to base kernels. Threfore,

  Configuration items to be enabled for base kernel
  = Configuration items to be marked by “=b” in objective item list
  = Mandatory items + Independent items

  Measurment result depends on the patterns.

5. Measurement preparation – The base kernels – (2)

- **Patterns of independent configuration items (i386)**

  The table below shows independent configuration item pattens which we currently use.

  Configuration items to be enabled for base kernel of a certain pattern
  = Configuration items to be marked by "=b" in the objective item list for the pattern
  = Mandatory items + "ENABLED" independent items shown in the table below

Patterns of independent configuration items (i386)

Independent configuration items	Pattern names: up	smp	small	cc_size	silent	module	procfs	sysctl	preempt	hotplug	pnp	pm	
SMP	×	●	×	×	×	×	×	×	×	×	×	×	-
CC_OPTIMIZE_FOR_SIZE	×	×	●	●	×	×	×	×	×	×	×	×	-
PRINTK	●	●	×	●	×	●	●	●	●	●	●	●	-
BUG	●	●	×	●	×	●	●	●	●	●	●	●	-
KMOD	●	●	×	●	●	×	●	●	●	●	●	●	Ref.1
PROCFS	●	●	×	●	●	●	×	●	●	●	●	●	-
SYSCTL	●	●	×	●	●	●	●	×	●	●	●	●	-
PREEMPT	●	●	×	●	●	●	●	●	×	●	●	●	-
HOTPLUG	●	●	×	●	●	●	●	●	●	×	●	●	-
PNP	●	●	×	●	●	●	●	●	●	●	×	●	-
PM	●	●	×	●	●	●	●	●	●	●	●	×	-

● : ENABLED
× : disabled

Ref.1 "CONFIG_KMOD=b" is enough to support kernel-module feature because CONFIG_KMOD "depends on CONFIG_MODULES".
Our .config generator can solve the dependency and enable CONFIG_MODULES automatically.

5. Measurement preparation – The base kernels – (3)

- **Patterns of independent configuration items (ARM)**

  The table below shows independent configuration item pattens which we currently use.

  Configuration items to be enabled for base kernel of a certain pattern
  = Configuration items to be marked by "=b" in the objective item list for the pattern
  = Mandatory items + "ENABLED" independent items shown in the table below

Patterns of independent configuration items (ARM)

		Pattern names												
		up	smp	small	cc_size	silent	module	procfs	sysctl	preempt	hotplug	pnp	pm	
Independent configuration items	SMP	×		×		×		×	×	×	×			Ref.1
	CC_OPTIMIZE_FOR_SIZE	●		●		●		●	●	●	●			Ref.2
	PRINTK	●		×		×		●	●	●	●			Ref.3
	BUG	●		×		×		●	●	●	●			-
	KMOD	×		×		×		×	×	×	×			Ref.1
	PROCFS	●		×		●		×	●	●	●			-
	SYSCTL	●		×		●		●	×	●	●			-
	PREEMPT	●		×		●		●	●	×	●			-
	HOTPLUG	●		×		●		●	●	●	×			-
	PNP	×		×		×		×	×	×	×			Ref.1
	PM	×		×		×		×	×	×	×			Ref.1

● : ENABLED
× : disabled

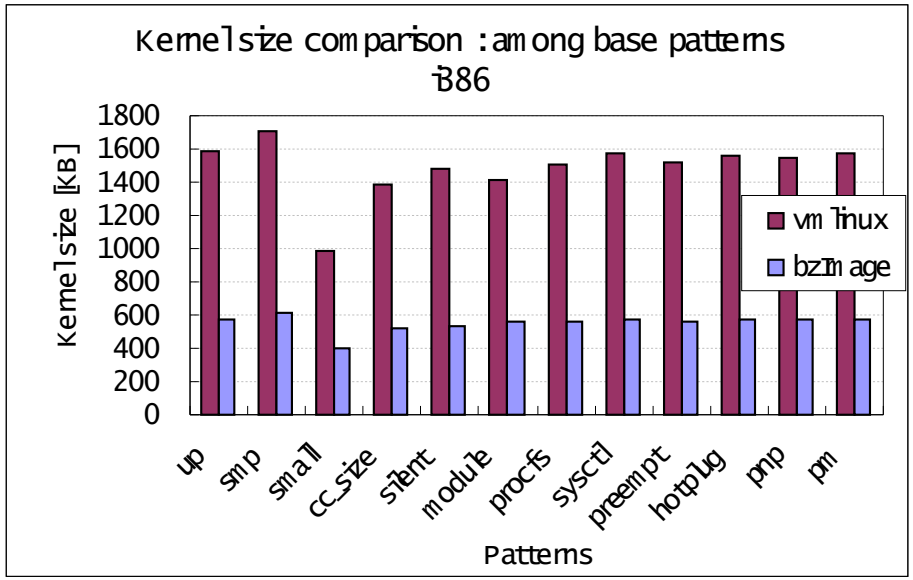
Ref.1 Compilation failure occured with enabled CONFIG_SMP, CONFIG_KMOD, CONFIG_PNP, or CONFIG_PM.

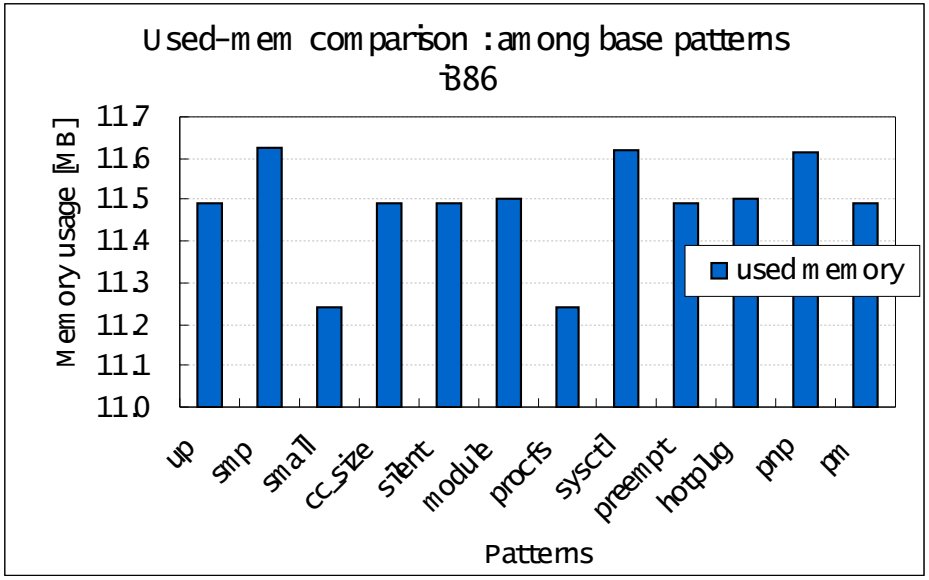
Ref.2 Compilation failure occured with disabled CONFIG_CC_OPTIMIZE_FOR_SIZE.

Ref.3 Boot failure occured with disabled CONFIG_PRINTK (see also 6.(2))

6. Measurement results (current status) (1)

- **Result of base kernels (i386)**
  - Comparing base kernel sizes and memory consumptions among independent item patterns.
  - Measurement conditions
    - Kernel : 2.6.12.3
    - gcc : 3.3.2 / binutils : 2.15
    - Target machine : Pentium IV 2.2GHz, RAM 256MB, HD 40GB

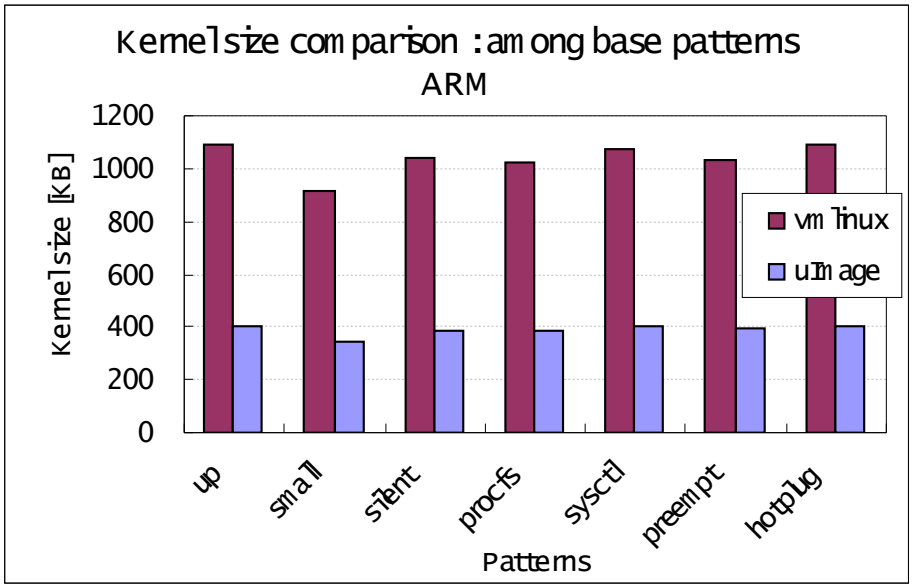


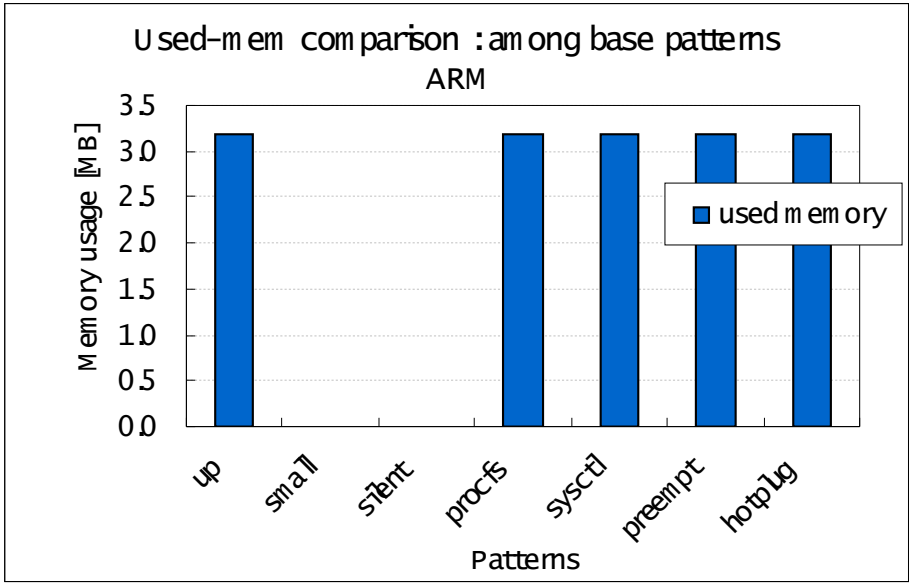


- **bzImage size of "small" pattern = 400KB**
  ... the bottom size limit of 2.6.12(i386) kernel
- **Memory consumption of "procfs" pattern is rather small**
  ... procfs relatively consumes large memory.

6. Measurement results (current status) (2)

- **Result of base kernels (ARM)**
  - Comparing base kernel sizes and memory consumptions among independent item patterns.
  - Measurement conditions
    - Kernel : 2.6.12.3
    - gcc : 3.3.2 / binutils : 2.15
    - Target machine : OSK5912 (OMAP5912(ARM926EJ-S) 192MHz, RAM 32MB, Flash 32MB)

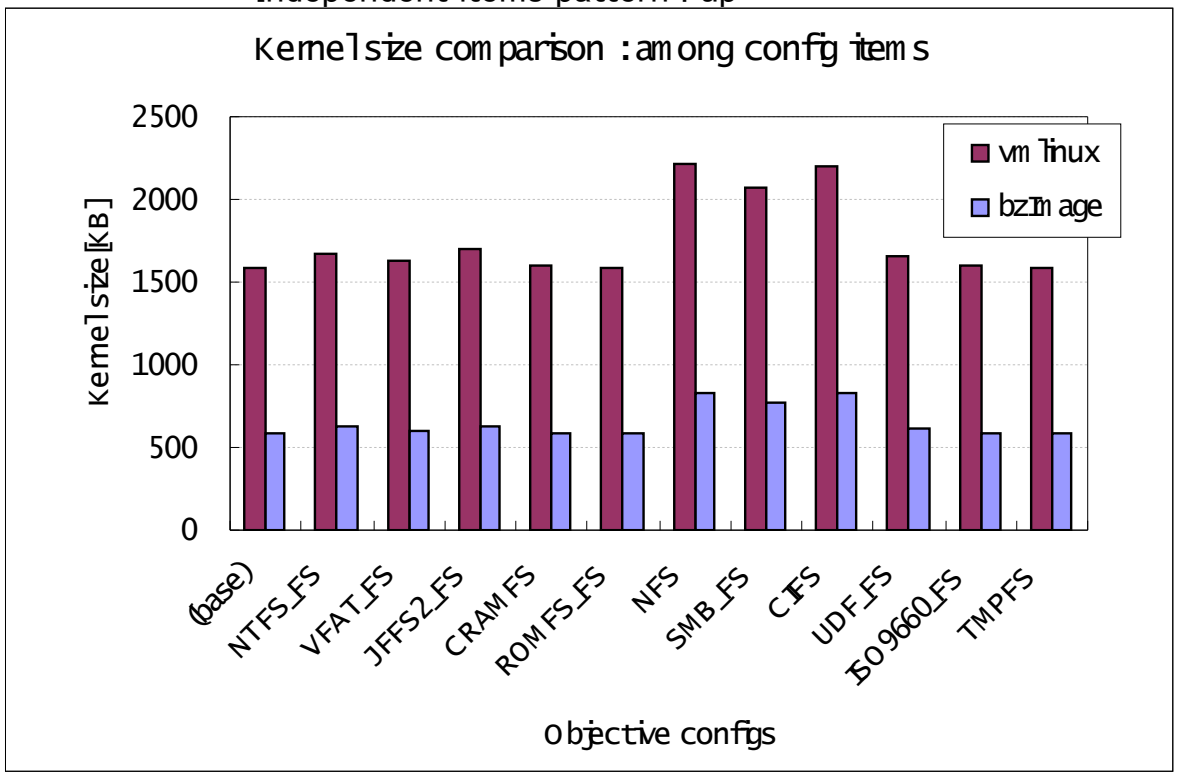




- **uImage size of "small" pattern = 340KB**
  ... the bottom size limit of 2.6.12(ARM) kernel
- **Memory consumptions are almost same.**
- **Kernels disabled printk support fail to boot (small and silent pattern)**
  kernel/printk.c : console_setup() may have to be defined...not tried yet.
  (http://www.selenic.com/pipermail/linux-tiny/2005-ugust/000216.html)

6. Measurement results (current status) (3)

- **Comparison among configuration items**
  - Comparing some filesystems
  - Measurement conditions
    - Kernel : 2.6.12.3
    - gcc : 3.3.2 / binutils : 2.15
    - Target machine : Pentium IV 2.2GHz, RAM 256MB, HD 40GB
    - Independent items pattern : up



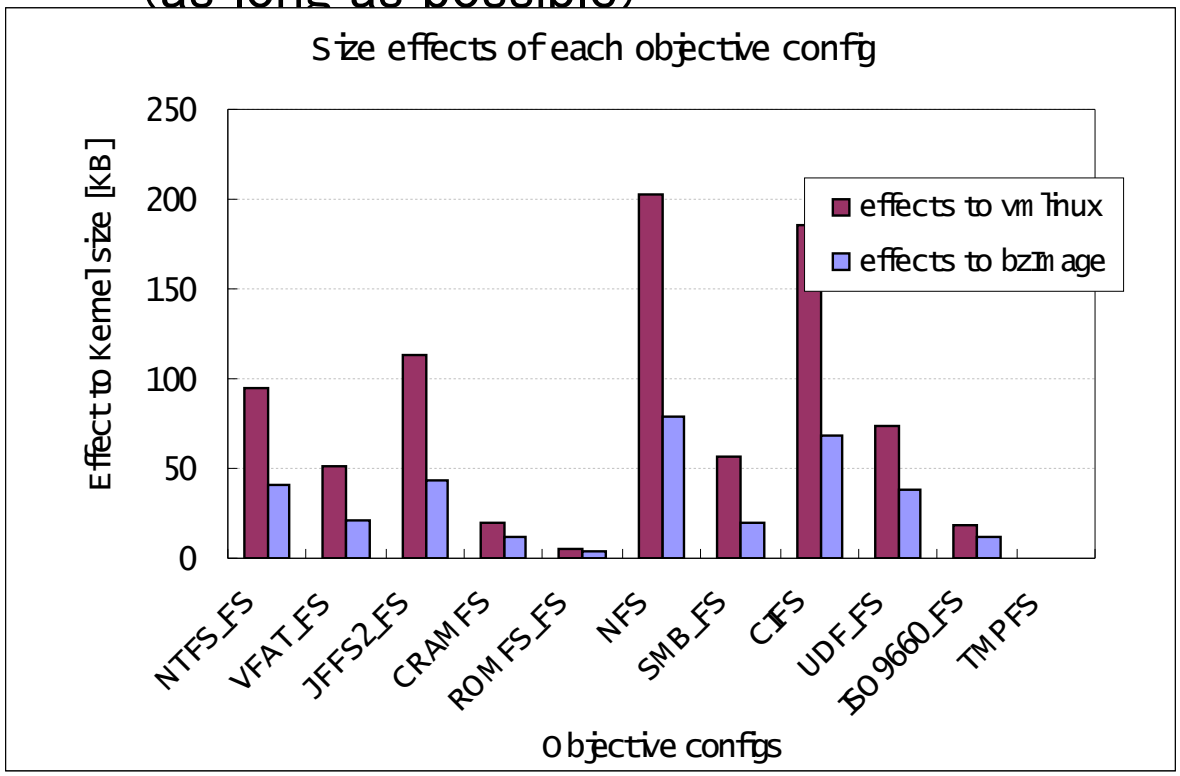
objective items	enabled configuration items in fact
(base)	(base)
NTFS_FS	NTFS_FS+NLS
VFAT_FS	VFAT_FS+NLS+FAT_FS
JFFS2_FS	JFFS2_FS+CRC32+MTD
CRAMFS	CRAMFS+ZLIB_INFLATE
ROMFS_FS	ROMFS_FS
NFS	NFS_FS+SUNRPC+LOCKD+INET+NET
SMB_FS	SMB_FS+NLS+INET+NET
CIFS	CIFS+NLS+INET+NET
UDF_FS	UDF_FS
ISO9660_FS	ISO9660_FS
TMPFS	TMPFS

- **NFS, CIFS and SMBFS seem large**

… but please note that each kernel contains enabled items other than objective one to solve dependencies. (continue to next page)

6. Measurement results (current status) (4)

- **Comparison among configuration items (cont'd)**
  - Below is the individual size effects, calculated by subtracting size of enabled items to solve dependencies from total size increments. (as long as possible)



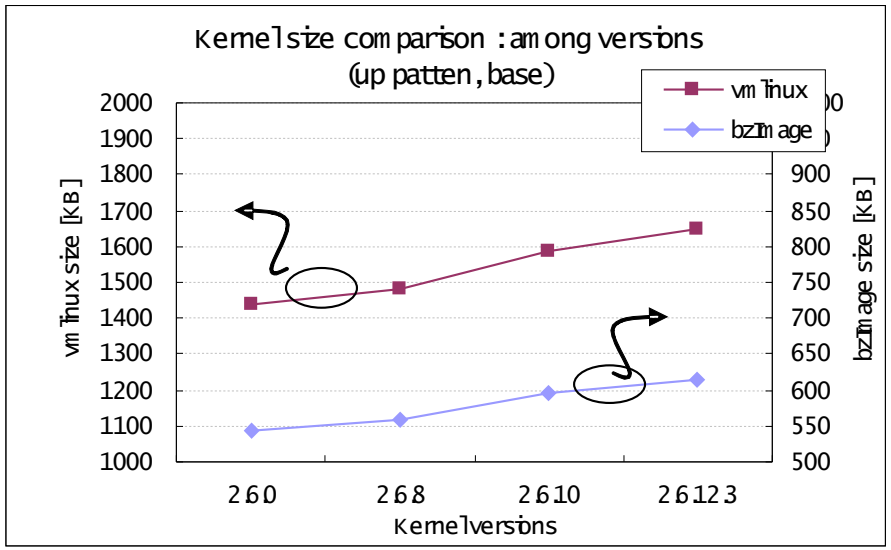
objective items	effective cnofig. items to the result
NTFS_FS	NTFS_FS
VFAT_FS	VFAT_FS+FAT_FS
JFFS2_FS	JFFS2_FS
CRAMFS	CRAMFS+ZLIB_INFLATE
ROMFS_FS	ROMFS_FS
NFS	NFS_FS+SUNRPC+LOCKD
SMB_FS	SMB_FS
CIFS	CIFS
UDF_FS	UDF_FS
ISO9660_FS	ISO9660_FS
TMPFS	TMPFS

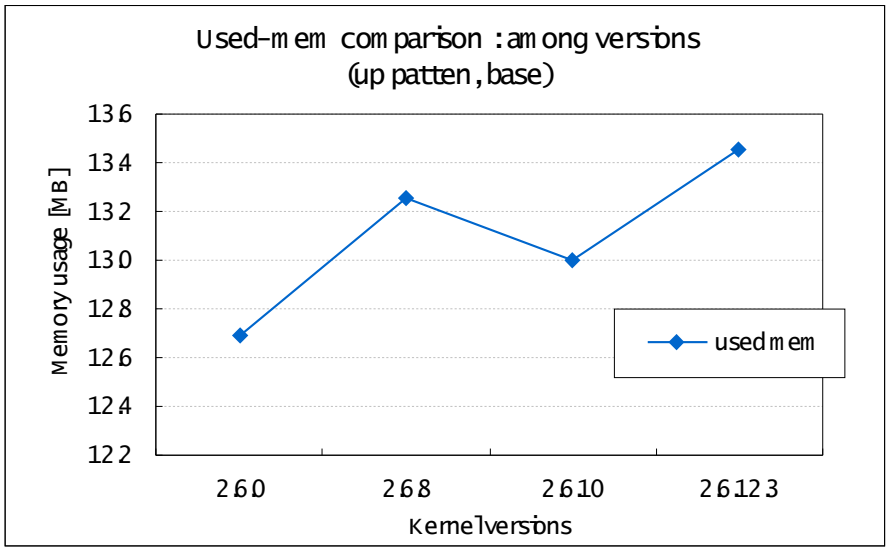
- **Individual size winners(?) are CIFS, JFFS2 and NTFS.**

… There are some items whose individual size can't be calculated because there is no individual result of enabled items to solve the dependencies. (VFAT, CRAMFS and NFS are the examples in above result)

6. Measurement results (current status) (5)

- **Comparison among Kernel versions (1)**
  - Comparing size/memory among some Kernel versions in 2.6 series.
  - up pattern / base kernels
  - Measurement conditions
    - gcc : 3.3.2 / binutils : 2.15
    - Target machine : Pentium IV 2.2GHz, RAM 256MB, HD 40GB
    - Independent items pattern : up





- **Kernel is thriving in.**

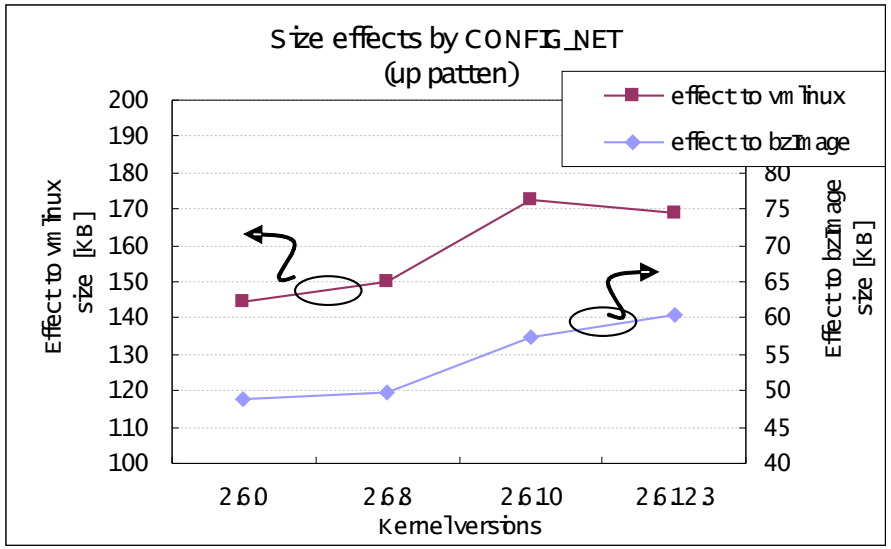
  Growth from 2.6.0 to 2.6.12 is about

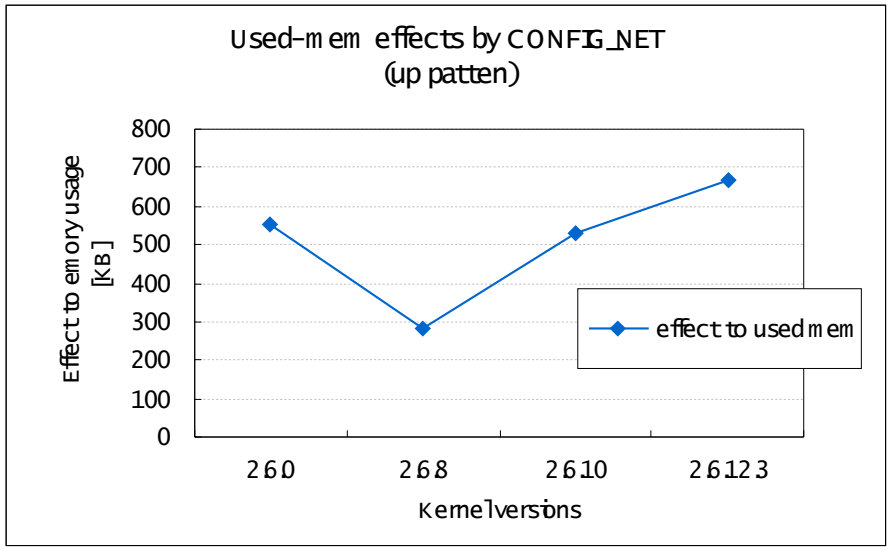
  vmlinux : 200KB , bzImage : 100KB , memory consumption : 750KB

  ... the bloating trend becomes clear.

6. Measurement results (current status) (6)

- **Comparison among Kernel versions (2)**
  - Comparing CONFIG_NET's individual size/memory effects among some Kernel versions in 2.6 series.
  - Measurement conditions
    - gcc : 3.3.2 / binutils : 2.15
    - Target machine : Pentium IV 2.2GHz, RAM 256MB, HD 40GB
    - Independent items pattern : up





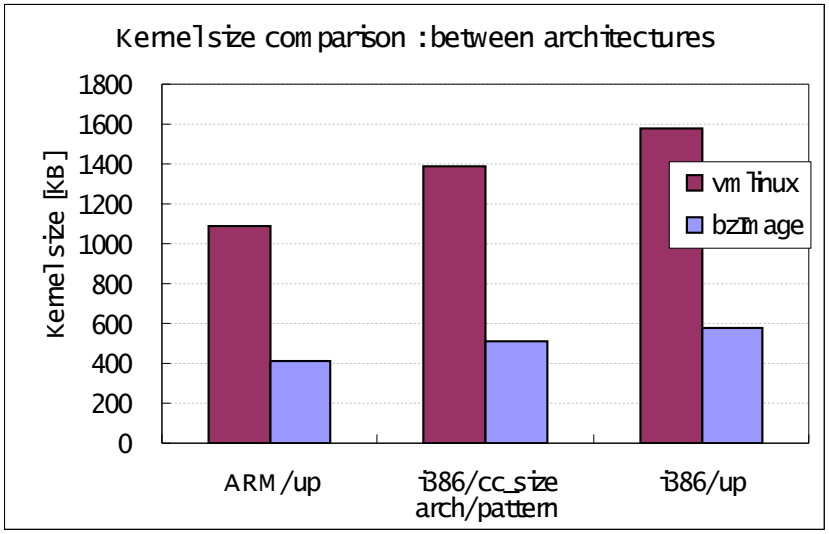
- **The Kernel part size associated with CONFIG_NET is gradually increasing.**

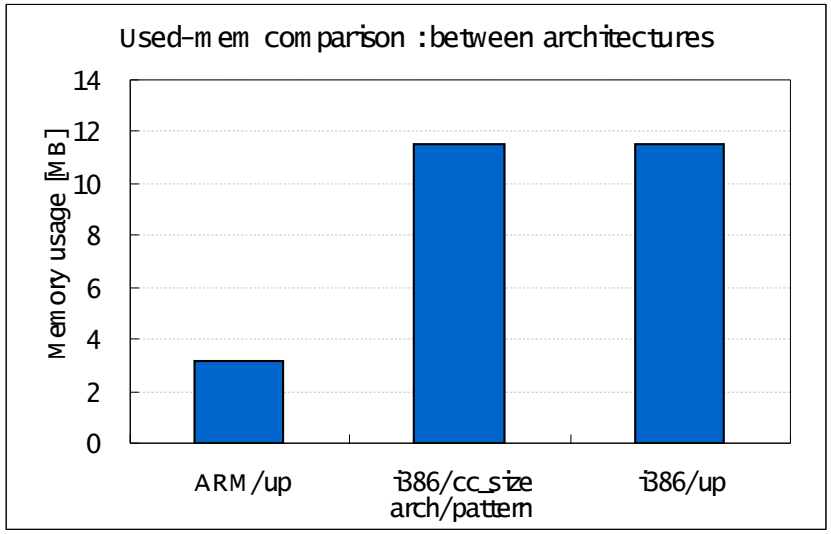
  Increasing from 2.6.0 to 2.6.12 is about
  vmlinux : 25KB , bzImage : 10KB

- **Memory consumption decreased once, and then increasing again.**

6. Measurement results (current status) (7)

- **Comparison between architectures (1)**
  - Comparing size/memory between i386/ARM.
  - Measurement conditions
    - Kernel : 2.6.12.3
    - gcc : 3.3.2 / binutils : 2.15
    - Target machine :
      - ARM : OSK5912 (OMAP5912(ARM926EJ-S) 192MHz, RAM 32MB, Flash 32MB)
      - i386 : PC (Pentium IV 2.2GHz, RAM 256MB, HD 40GB)
    - Independent items pattern :
      - ARM : up
      - i386 : up, cc_size ... because CONFIG_CC_OPTIMIZE_FOR_SIZE=y in up pattern for ARM

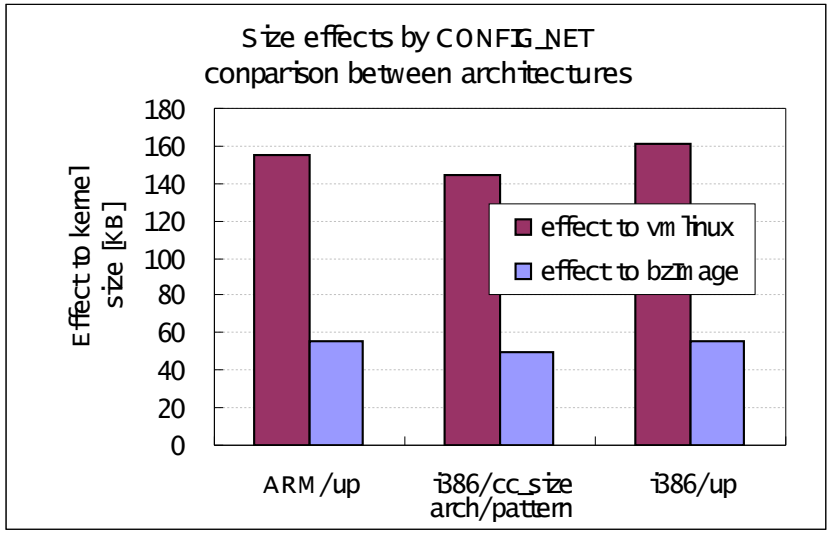


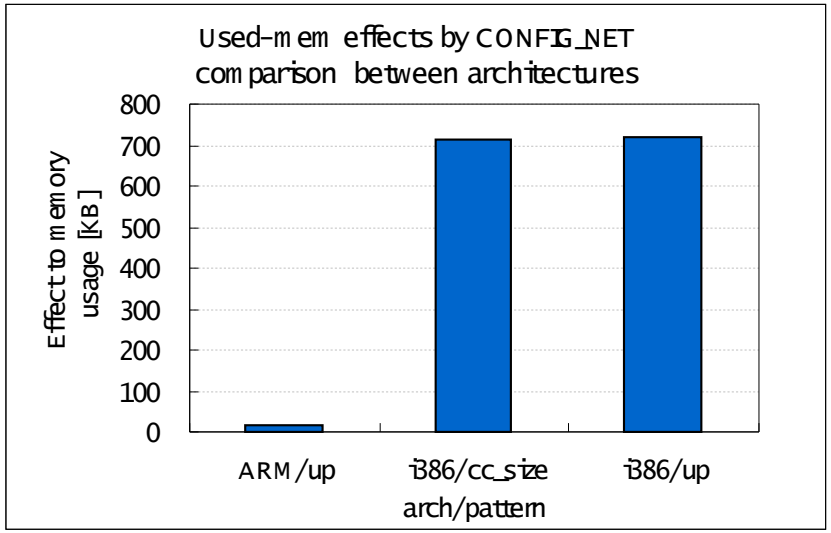


- **Both of Kernel size and memory consumption of ARM are smaller than i386.**
  (Note that they aren't directly comparable because i386 kernels contain drivers for IDE, PCI, etc.)
- **Memory consumption of ARM is extremely small, because ... ?**

6. Measurement results (current status) (8)

- **Comparison between architectures (2)**
  - Comparing CONFIG_NET's individual size/memory effects between i386/ARM.
  - Measurement conditions
    - Kernel : 2.6.12.3
    - gcc : 3.3.2 / binutils : 2.15
    - Target machine :
      - ARM : OSK5912 (OMAP5912(ARM926EJ-S) 192MHz, RAM 32MB, Flash 32MB)
      - i386 : PC (Pentium IV 2.2GHz, RAM 256MB, HD 40GB)
    - Independent items pattern :
      - ARM : up
      - i386 : up, cc_size ... because CONFIG_CC_OPTIMIZE_FOR_SIZE=y in up pattern for ARM

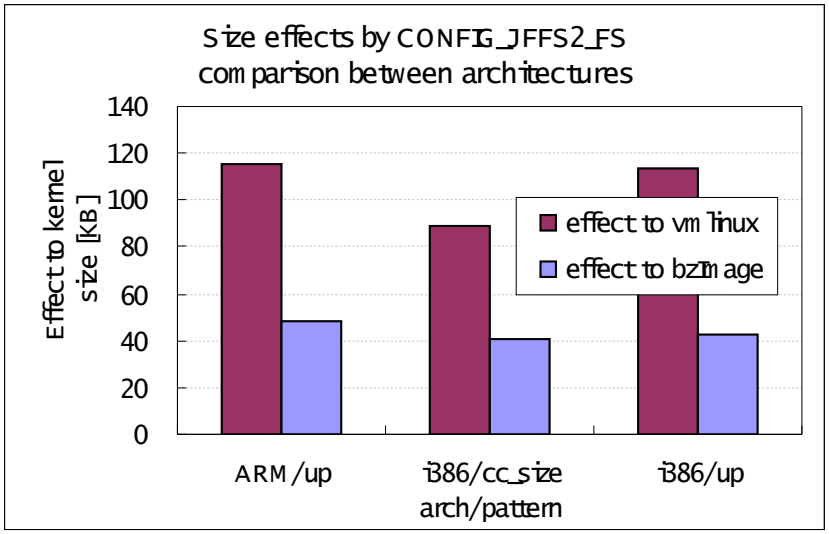


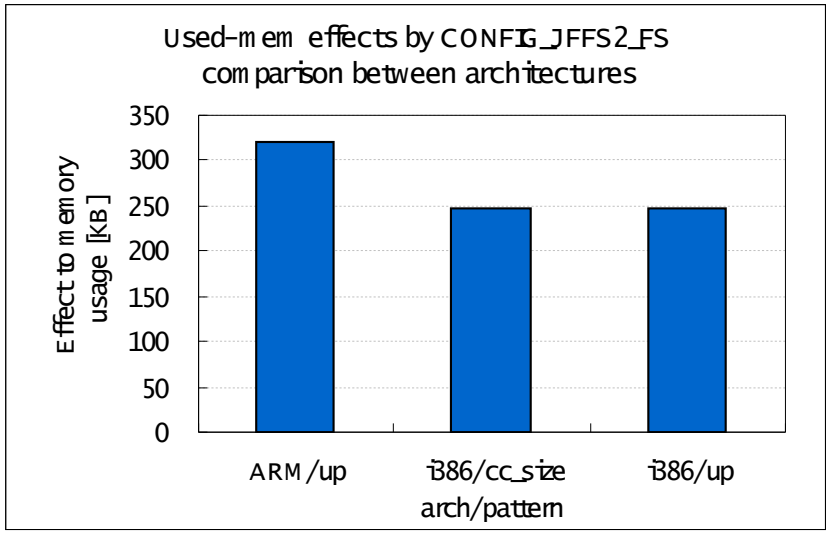


- **Size effect on ARM is slightly larger as long as comparing with CONFIG_CC_OPTIMIZE_FOR_SIZE enabled kernels.**
  (Supposed because of a difference between RISC/CICS)
- **Memory consumption effect of ARM is extremely small, because ... ?**

6. Measurement results (current status) (9)

- **Comparison between architectures (3)**
  - Comparing CONFIG_JFFS2_FS's individual size/memory effects between i386/ARM.
  - Measurement conditions
    - Kernel : 2.6.12.3
    - gcc : 3.3.2 / binutils : 2.15
    - Target machine :
      - ARM : OSK5912 (OMAP5912(ARM926EJ-S) 192MHz, RAM 32MB, Flash 32MB)
      - i386 : PC (Pentium IV 2.2GHz, RAM 256MB, HD 40GB)
    - Independent items pattern :
      - ARM : up
      - i386 : up, cc_size ... because CONFIG_CC_OPTIMIZE_FOR_SIZE=y in up pattern for ARM





- **Both of size and memory consumpstion effect on ARM are slightly larger as long as comparing with CONFIG_CC_OPTIMIZE_FOR_SIZE enabled kernels.**
  (Supposed because of a difference between RISC/CICS)

6. Measurement results (current status) (10)

- **Current conclusion**
  - The bottom size limit of 2.6.12 :
    - i386 : vmlinux = 920KB , bzImage = 400KB
    - ARM : vmlinux = 900KB , uImage = 340KB
      ※ Note that they aren't directly comparable because i386 kernels contain drivers for IDE, PCI, etc.)
  - The bottom memory consumption limit has not been concluded.
    - The cause of the memory consumption diffrence between i386 and ARM is unknown.
    - Kernel for ARM which is disabled printk support can't boot.
  - CIFS has the largest size-effect in our FS measurment, and JFFS2 follows.
    - CIFS : vmlinux 190KB, bzImage 70KB
    - JFFS2 : vmlinux 110KB, bzImage 40KB
      (Both of them are results from i386/up pattern)
  - The bloating trend becomes clear in 2.6 series.
    - Bloat from 2.6.0 to 2.6.12 is
      vmlinux=200KB , bzImage=100KB , memory consumption=750KB
      (Both of them are results from i386/up pattern)
  - Individual size effect of a certain item on ARM is slightly larger than on i386. (Supposed because of a difference between RISC/CICS)

7. Future work

- **Brushing-up features and implementation**
  - Kernel size prediction
    - Suppose that Kernel size can be calculated from .config file by suming up individual size effects of independent configuration items?
    - Result of pre-investigation :
      - Investigated with 154 independent configuration items

Base kernel size	1195.0 [KB]	(a)
Sum of individual effects	3083.4 [KB]	(b)
Size prediction	4278.4 [KB]	(c) = (a) + (b)
Size in fact	4082.9 [KB]	(d)
Error	195.5 [KB]	(e) = (c) - (d)
	4.8%	(f) = (e)/(d)

➔Guess a reasonable result?

  - Make usage simple
    - Setting up and usage are very complicated currently...
      (There is a pile of configurations to use The Tool though not mentioned in these slides)
  - Bug fix
- **Publishing tools and sharing results**
  - Plan to publish current objective item lists and results on CELF WiKi in early Feb.
  - Plan to publish tools on CELF WiKi by end of Feb.
  - Wish to discuss on CELF SZWG mailing list.
  - Wish to work with CELF Test Lab. and to measure meaningful data.

8. Let's have a discussion !

- **Please give us your opinion for example ...**
  - "I want data like as ... "
  - "I have an idea that the tool can be used for ..."
  - etc. ....
- **Please help us and work together for example ...**
  - Reviewing whether our approach is correct.
  - Reviewing whether our configuration for base kernels is reasonable.
  - Reviewing whether our selection of objective items is reasonable.
  - Doing tests
  - Investigating about causes of kernel size bloating
  - Writing patches to shrink kernel size
  - etc. ...

Empowered by Innovation