



Reduction of RAM consumption by SquashFS

TOSHIBA Corp.

Keijiro Yano

2006/01/20



Contents

- Cramfs vs. SquashFS.
- Introduction of our evaluation system overview and issues with SquashFS.
- Enhancement of SquashFS.
- Block size of SquashFS.



Environments

- Kernel

Linux 2.6.10 for MIPS

- SquashFS Patch

SquashFS 2.2

http://sourceforge.net/project/showfiles.php?group_id=63835



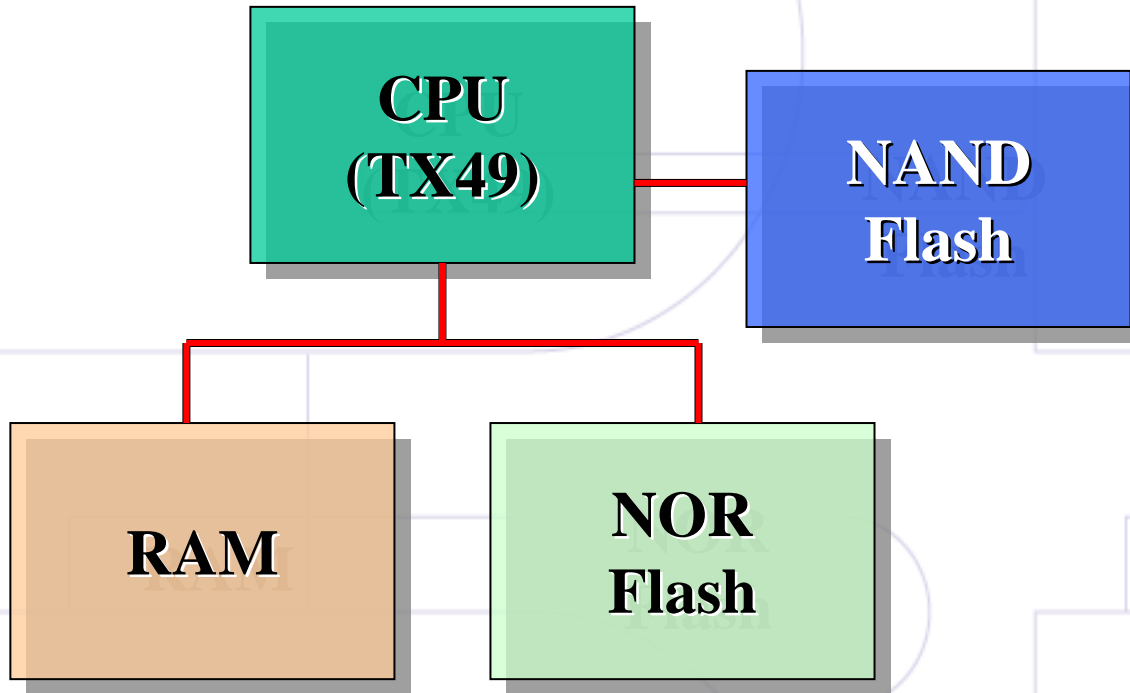
Cramfs vs. SquashFS

	Cramfs	SquashFS
Compress Library	ZLIB	ZLIB
Compressed block	4KB Fixed	0.5 ~ 64KB
Metadata compression	Not support	Support
Fragment block compression	Not support	Support
XIP support	Support	Support

Cramfs can be mounted without the block device using XIP and the linear access to RAM.



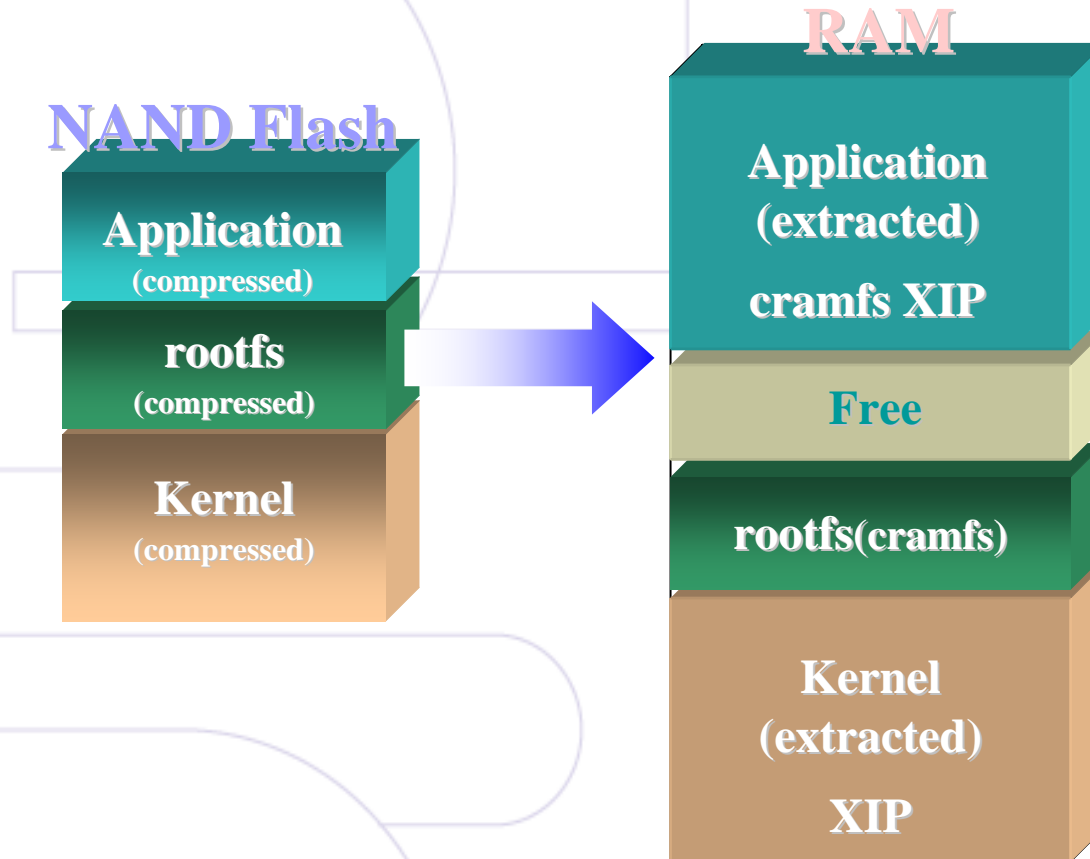
Boot sequences of our evaluation system



- ① Launch Bootloader.
- ② Read Kernel/rootfs from NAND Flash.
- ③ Extract the binary image to RAM.
- ④ Launch Kernel from RAM.



Memory structure of our evaluation system (Cramfs)





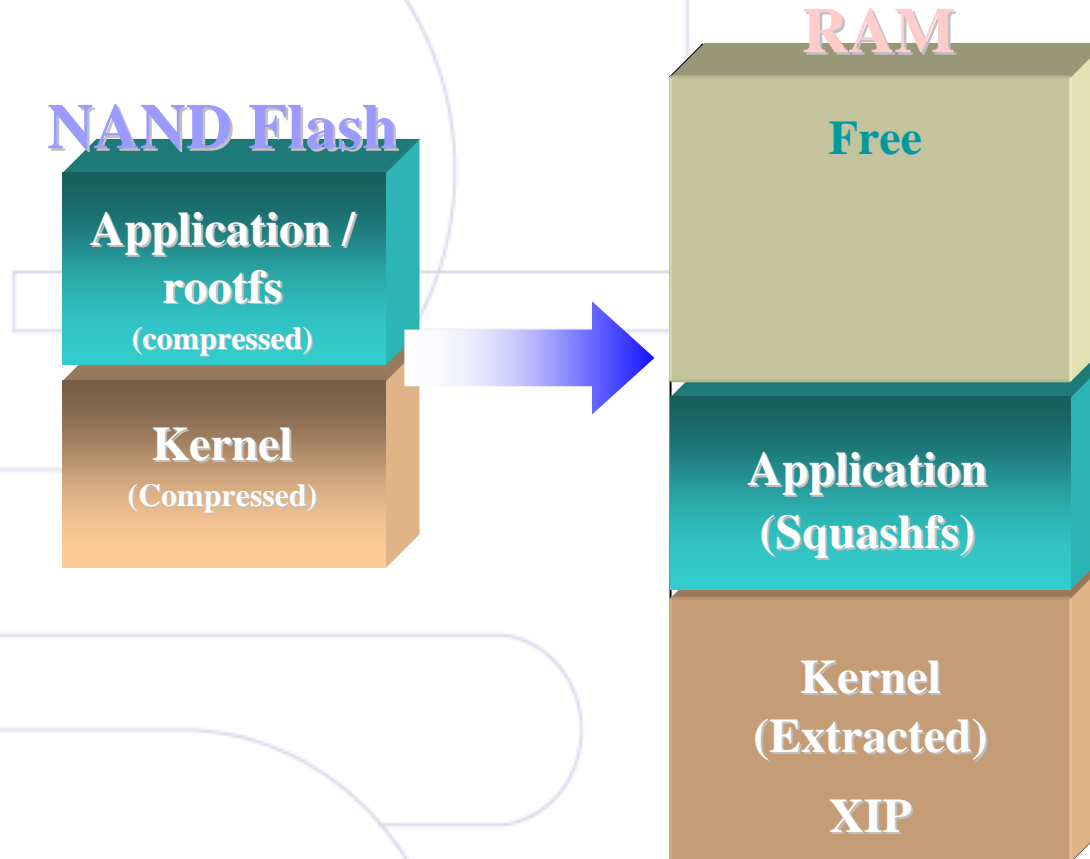
Filesystem on our evaluation system

cramfs(nonXIP)	Use compressed binary by Cramfs. The binary made by mkcramfs is stored in NAND Flash.
cramfs(XIP)	Use XIP binary (uncompressed). The compressed binary by ZLIB is stored in NAND Flash.

- We can NOT mount NAND Flash for storing the application due to our system restriction.
- We have to use cramfs(XIP) for storing the application due to the amount of NAND Flash.

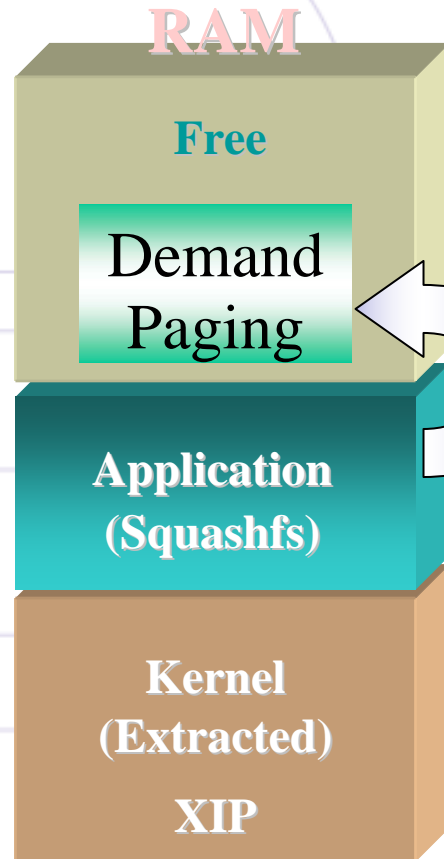


Memory structure of our evaluation system (SquashFS)





Memory structure of our evaluation system (SquashFS)



We can reduce RAM consumption by using Demand Paging for the text data of the application.

But...
The performance of the application, especially boot time, get slow...



Merit & Demerit on our evaluation system with SquashFS

Merit

Reduce the size of the binary image on NAND Flash.

vs. cramfs(noXIP)

about 1MB reduction

vs. cramfs(XIP)

same size

→ It makes boot-time faster.

Reduce the usage of RAM.

vs. cramfs(XIP)

about 5MB reduction

※ These size depends on the application.

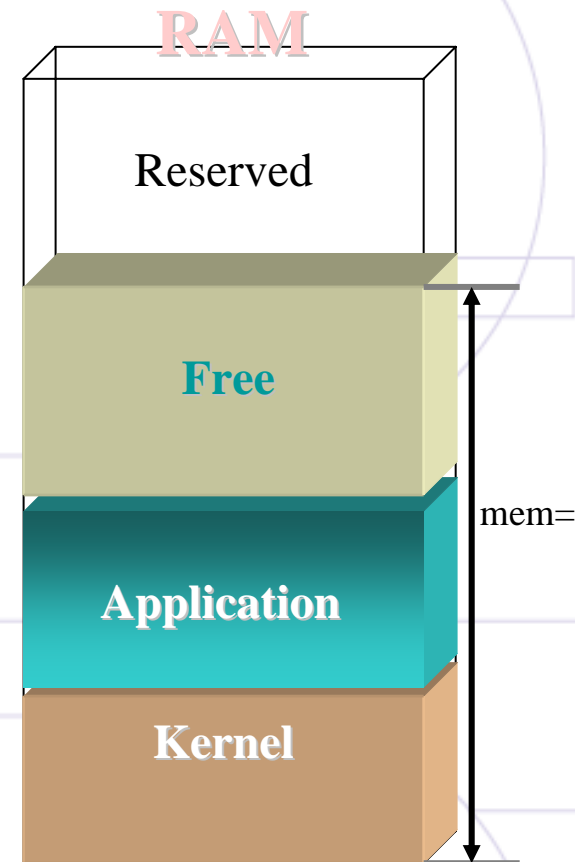
Demerit

The application performance is reduced by the overhead of demand paging.



RAM consumption

- On our evaluation system, we have to reserve the fixed area on RAM.
 - We use “mem=” option to limit the memory.
- “Reduce the usage of RAM” means “Reduce the size specified by mem= option”.
 - We evaluate this size with the test application, which allocates the memory with malloc() 1MB by 1MB.





Enhancement of SquashFS

In order to mount SquashFS binary,
We have to use;

RAM Disk

Or

MTD Uncached system RAM Driver

It makes boot-time slow,
because the kernel extracts
the binary to RAM Disk.

MTD driver handles the read
requests with kernel thread.
This overhead makes the big
impact on our system.

We need good solutions...



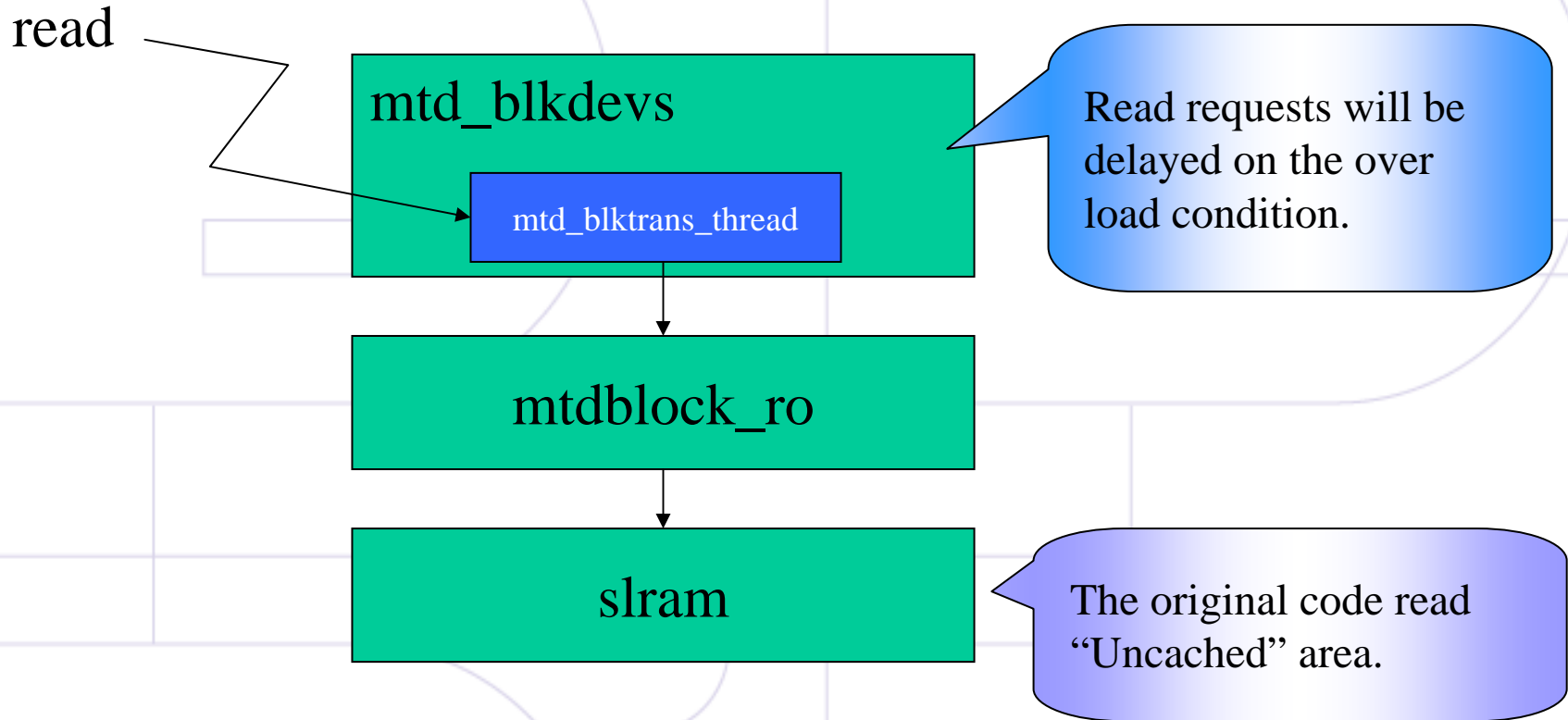
RAM Disk

- The kernel reads Initrd image and writes it to /dev/ram.
(init/do_mounts_rd.c)

```
for (i = 0, disk = 1; i < nblocks; i++) {  
    if (i && (i % devblocks == 0)) {  
        ...  
        printk("Loading disk #%d... ", disk);  
    }  
    sys_read(in_fd, buf, BLOCK_SIZE);  
    sys_write(out_fd, buf, BLOCK_SIZE);  
}
```



MTD Uncached system RAM

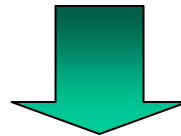




Linear access support on SquashFS

- We add the new feature to SquashFS to access RAM(cached) directly.
So we do NOT need any Block Devices for mounting SquashFS.

→ We implement it based on Cramfs source code.



It makes boot-time **about 500 msec** faster than mounting with RAM Disk on our evaluation system.



Block size of SquashFS

- Use a large block size in order to get a good compression rate.
(Max 64KB)



But...

- It will make the possibility to extract the unnecessary binary.
- This will cause a bad performance of the application with the overhead of the extraction of SquashFS.

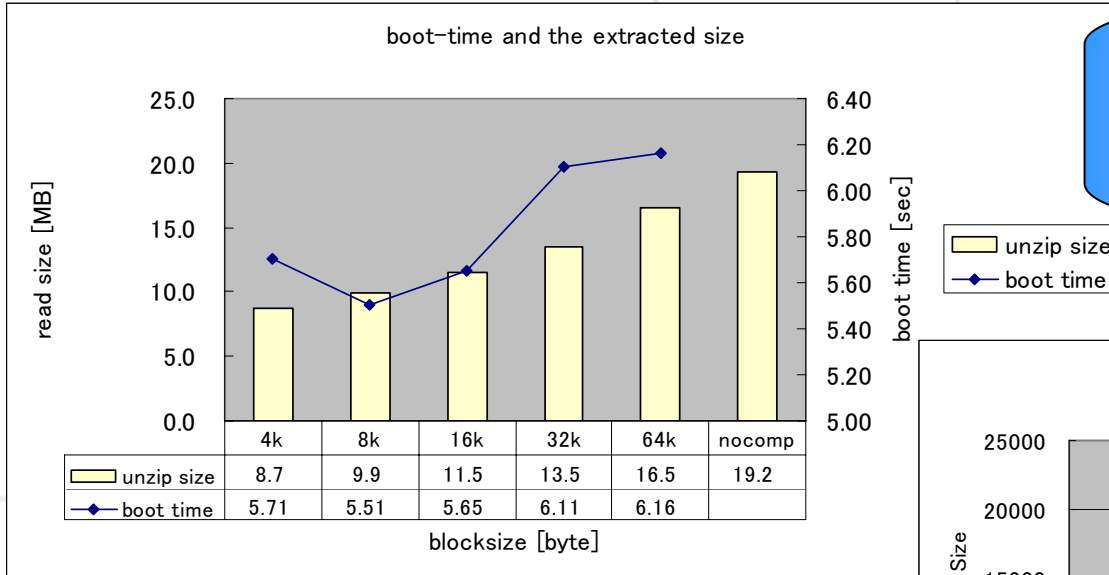


Therefore...

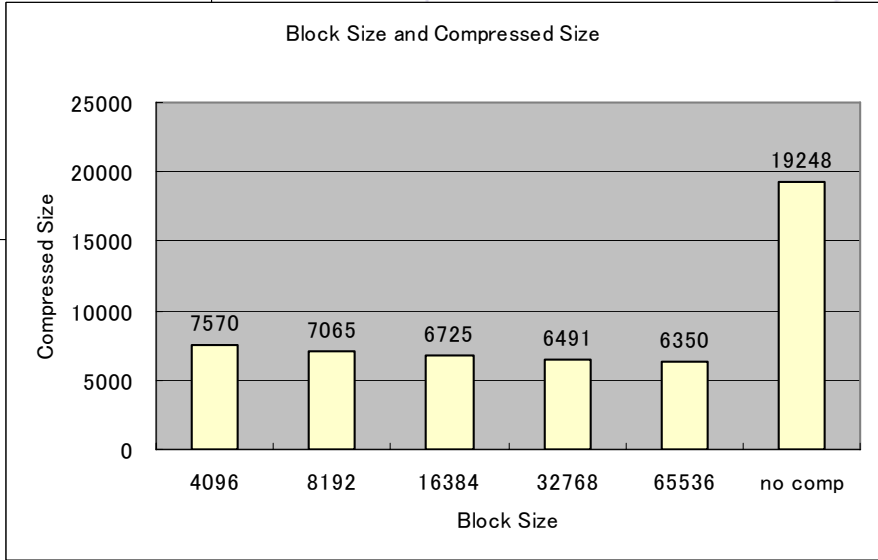
- You should decide the block size with the total balance of the compression rate and the application performance.



Evaluation result on our system. (Block size of SquashFS)



Compressed binary size



Boot time and the extraction size for each block size.



Conclusion

- We can **reduce RAM consumption** by SquashFS on our evaluation system without **increasing NAND Flash consumption**.
- We can avoid the demerit of SquashFS with our implementation of the **enhancement of SquashFS**.
- You should decide **the appropriate block size of SquashFS for your purpose, a performance of the application and/or the binary size**.



End