FUJITSU

shaping tomorrow with you

Linux File System Analysis for IVI Systems

July 1, 2014

Mitsuharu Ito <itom@jp.fujitsu.com>
Fujitsu Computer Technologies, Ltd

1279ka01

Get Excited about the World's Largest Football Tournament!

FUJITSU

- I like Football very Much!
- Many Kinds of Embedded Devices are used
  - e.g. High-Definition Video Transmission System, Goal Decision System, Offside Signal Transmission Devices on Flags, ...
- Videos, Sounds and Sensing Data are stored as Files on File Systems

Fujitsu IP-9500

- Embedded Devices have been used more and more in Sports, Healthcare, Agriculture, **Automotive Industry**, ...
- Dramatically Increasing Big Data

→ Core Technology of **File System** will be More Important

Agenda

FUJITSU

- Background
- File System Comparison for IVI
- Evaluation of File System Requirements
  - Robustness
  - Boot Up Time
  - Performance
- Conclusions

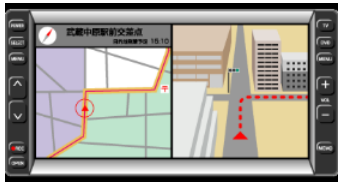
Who am I?

- Embedded Software Engineer
  at Fujitsu Computer Technologies
  - Embedded Linux Distribution and Driver Development (In-House Use), Technical Support, Training

- Our Distribution is used for Fujitsu's Products
  - Server System Controller, Network Equipment, Printer, IVI, and many other systems







What is File System?

FUJITSU

- One of the Features of Operating Systems
  to Store and Organize **Data** as **Files**
  on **Media**, such as HDD, CD/DVD/BD, Flash Drive
- Linux Supports many kinds of File Systems
  - Disk File Systems : Ext2/3/4, XFS, ReiserFS ZFS, Btrfs, ...
  - Flash File Systems : JFFS/JFFS2/YAFFS, UBIFS, LogFS, F2FS, ...
  - Network File Systems : NFS, Samba, AFS, ...

- Stored Data as Files in IVI Systems
  - 2D/3D Maps, Videos
  - Sounds of Voice, Music, Buzzer, ...
  - Information about Traffic, Shops, Disaster, ...
  - Sensing Data
  - System Logs

Motivation

FUJITSU

- **AGL Requirements Version 1.0** is Planned to be Released Soon
  - **Robust File System**
  - References to **Btrfs**, Ext2/3/4, Vfat, UBIFS

- Fujitsu has been Contributing to **Btrfs** from an Earlier Time
  - No.2 Contributor (No.1 in 2013)

- How Suitable are Btrfs and other File Systems for IVI?
  - Functional Requirements?
  - Non-Functional Requirements?

[git://git.kernel.org/pub/scm/linux/kernel/git/stable/linux-stable.git]

```
$ git log
Author: Linus Torvalds
Date:   Sat Jun 21 19:02:54 2014 -1000
```

Linux 3.16-rc2

```
$ git log fs/btrfs/ | gitdm
Top changeset contributors by employer
Oracle                    1166 (30.0%)
```

Fujitsu 634 (16.3%)

```
Red Hat                    394 (10.1%)
jbacik@fusionio.com        265 (6.8%)
Novell                     202 (5.2%)
fdmanana@gmail.com         134 (3.5%)
sbehrens@giantdisaster.de  103 (2.7%)
list.btrfs@jan-o-sch.net    93 (2.4%)
viro@zeniv.linux.org.uk     92 (2.4%)
idryomov@gmail.com          79 (2.0%)
yanzheng@21cn.com           55 (1.4%)
sensille@gmx.net            50 (1.3%)
hch@lst.de                  48 (1.2%)
Intel                       35 (0.9%)
sage@newdream.net           35 (0.9%)
```

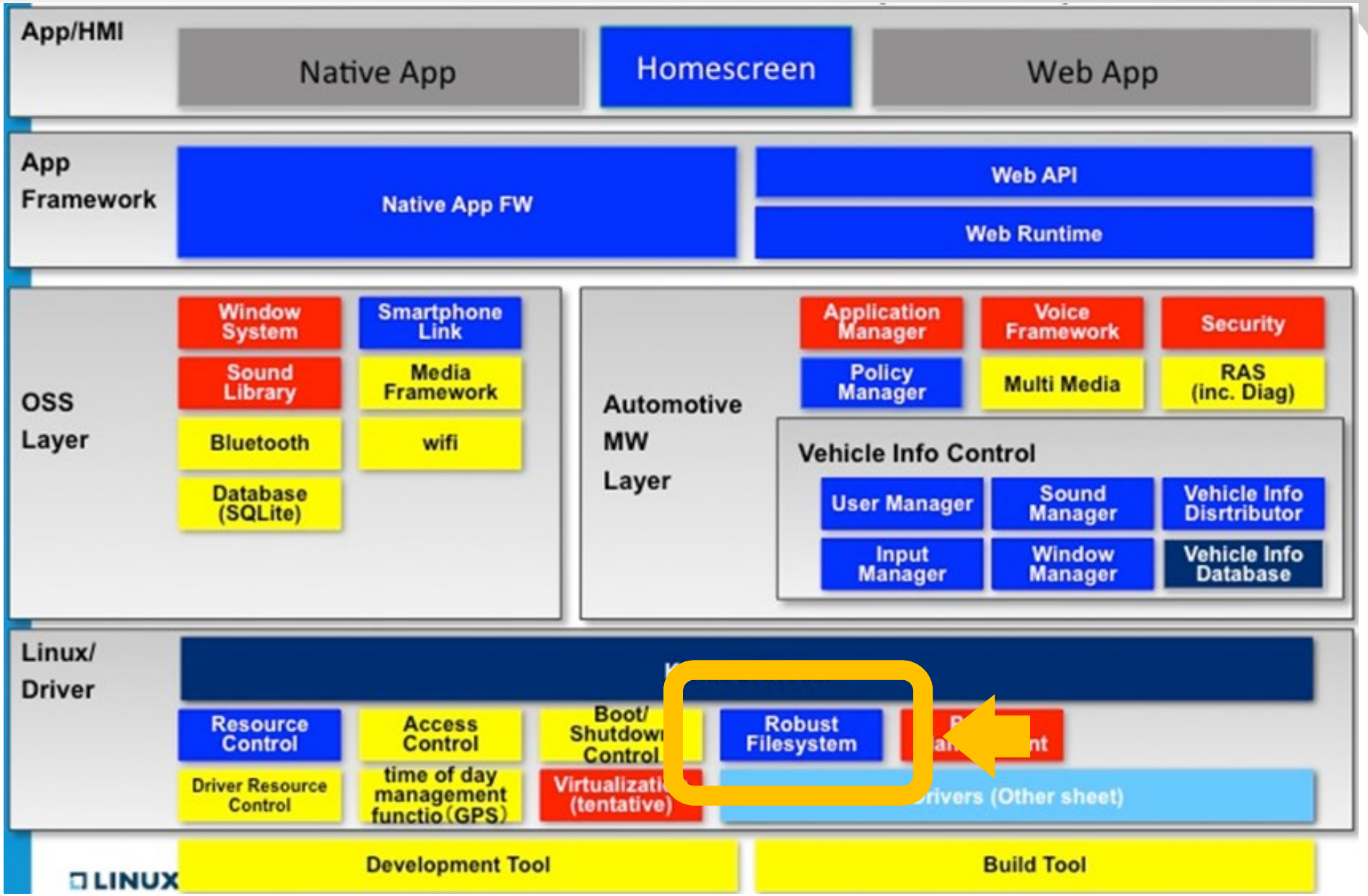
→ **FS Suitability Analysis for IVI**

FUJITSU

File System Comparison for IVI

AGL Requirements (Architecture version 0.8.2)

FUJITSU



AGL Requirements (P1:Mandatory version 0.8.2)

FUJITSU

Bold : Embedded System Specific

1. File Systems

1.1. Robust File System for Managed Internal Storage (SSD, eMMC, etc)

- 1.1.1. **Power Failure Tolerance**
- 1.1.2. **Quick Recovery After Power Loss**
- 1.1.3. Multi-threaded I/O
- 1.1.4. On-demand Integrity Checker
- 1.1.5. Read-only Mode
- 1.1.6. Non-blocking Unmounting

1.2. File System for Non-managed Internal Storage (raw NOR and NAND FLASH memory)

- 1.2.1. All P1 Requirements from FS.1.1.x List
- 1.2.2. **Wear Leveling**
- 1.2.3. **Error Detection/Correction**
- 1.2.4. **Tolerance to Flipping Bits**
- 1.2.5. **Read/Write Disturb Awareness**
- 1.2.6. **Bad Block Management**

1.3. File System for Removable Storage (USB stick, SD card)

- 1.3.1. Restricted Functionality from Security Point of View
- 1.3.2. Automount/Autounmount

AGL Requirements (P2:Optional version 0.8.2)

FUJITSU

Bold : Embedded System Specific

1.1. Robust File System for Managed Internal Storage (SSD, eMMC, etc)

- 1.1.7. Means for Optimizing I/O Performance if It May Degrade under Certain Conditions
- 1.1.8. File Space Pre-allocation
- 1.1.9. Meta-data Error Detection
- 1.1.10. File Data Error Detection
- 1.1.11. Online Integrity Checking
- 1.1.12. Write Timeout Control
- 1.1.13. **Compression support**
- 1.1.14. Quota Support
- 1.1.15. I/O Process Priority
- 1.1.16. File System Event Notifications
- 1.1.17. Logical Block Size Control
- 1.1.18. Snapshots

1.2. File System for Non-managed Internal Storage (raw NOR and NAND FLASH memory)

- 1.2.7. As Many P2 Requirements from FS.1.1.x List as Possible
- 1.2.7. **Wear Leveling Statistics**

1.3. File System for Removable Storage (USB stick, SD card)

- 1.3.3. Automatic synchronous flushing of modified data to physical media

Functional Comparison for P1

FUJITSU

Type of Storage Device	ID	Name	Btrfs		Ext2/3/4						FAT	UBIFS
			Btrfs	rfsck	**Ext2**	E2defrag	**Ext3**	**Ext4**	4defrag	E2fsck	**Vfat**	**UBIFS**
			R-FS.1	R-FS.1.1	R-FS.2	R-FS.2.1	R-FS.3	R-FS.4	R-FS.4.1	R-FS.4.2	R-FS.5	R-FS.6
Internal Managed (**SSD**, **eMMC**, etc.)	FS.1	File Systems	✔		✔		✔	✔			✔	
	FS.1.1	Robust File System for	✔		✔		✔	✔				
	FS.1.1	**Power Failure Tolerance**		A		N/A	✔	✔	N/A	N/A		
	FS.1.1.2	Quick Recovery after power loss	✔				✔	✔				
	FS.1.1.3	Multi-threaded I/O	N/A		N/A		N/A	N/A			N/A	N/A
	FS.1.1.4	On-demand integrity checker	✔		✔							
	FS.1.1.5	Read-only mode	✔	N/A	✔	N/A	✔	✔	N/A	N/A	✔	
	FS.1.1.6	Non-blocking unmounting *	✔		✔		✔	✔			✔	
		Number of Checks	**7**		**5**	N/A	**7**	**7**	N/A	N/A	**3**	
Internal Non-managed (**raw NOR and NAND FLASH memory**)	FS.1.2	File System for non-managed internal storage										✔
	FS.1.2.1	All P1 requirements from FS.1.1.x list										N/A
	FS.1.2.2	Wear leveling										✔
	FS.1.2.3	Error detection /correction					N/A					✔
	FS.1.2.4	Tolerance to flipping bits										
	FS.1.2.5	Read/write disturb awareness										
	FS.1.2.6	Bad block management										✔
		Number of Checks										4
removable managed (**USB stick**, **SD card**)	FS.1.3	File Systems for removable storag	✔		✔		✔	✔			✔	
	FS.1.3.1	Restricted functionality from security point of view	✔	N/A	✔	N/A	✔	✔	N/A	N/A	✔	N/A
	FS.1.3.2	Automount/autounmount **	✔		✔		✔	✔			✔	
		Number of Checks	3		3	N/A	3	3	N/A	N/A	3	

* implemented in VFS Layer ** supported by udev

Functional Comparison for P1 (contd.)

FUJITSU

- Btrfs and Ext3/4 are the Most Suitable Candidates
  for Internal Managed Storage Devices (eMMC, SSD, ...)
- Btrfs and Ext3/4 are also Available
  for Removable Managed Storage Devices (USB Stick, SD card, ...)
- Ext4 is the Successor to Ext3

→ We Focused on Btrfs and Ext4 as Target of Evaluation

- All AGL Requirements are Functional

→ We started to Evaluate "Power Failure Tolerance" 
  as the one of Most Important Requirements of IVI

Other Requirements of File Systems

FUJITSU

- Short Boot Time 
  - Time to Show Splash Screen, Home Screen, and Play Startup Sounds (within a few seconds in most cases)
- Performance 
  - I/O Throughput
  - Application QoS (Quality of Service) : Constant Performance under High Load Not to Keep HMI Applications Waiting for a Long Time
- Security
  - Permission Control, Encryption, …
- Scalability

Overview of Ext4

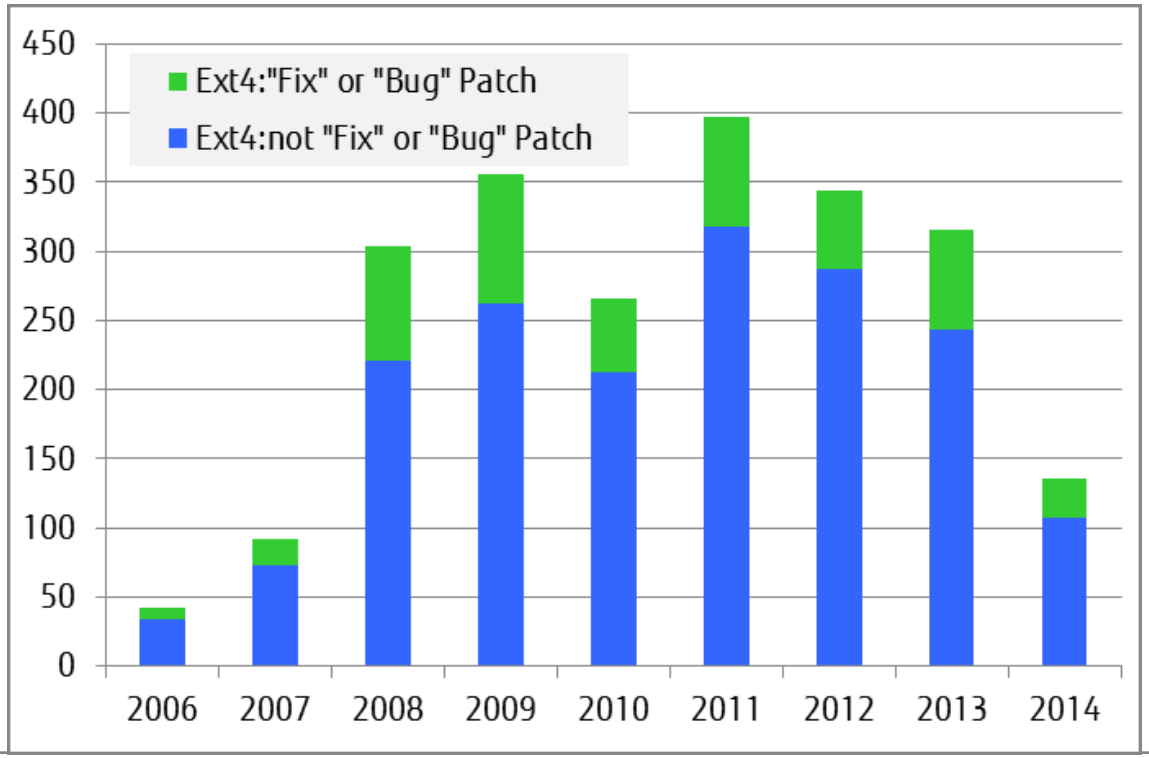
FUJITSU

- Journaling File System
  Developed as the Successor to Ext3
- Merged in Mainline **Kernel 2.6.19 in Nov 2006**
- Key Features
  - Large Volume and File Size
  - **Journaling** and Journal Checksum
  - Persistent pre-allocation, ...
- **Standard File System** for Many Major Linux Distros
  - Fedora 11+
  - RHEL 5.6+
  - Ubuntu 9.10+
  - Debian 6.0+

Overview of Ext4 (contd.)

FUJITSU

- Development Status
  - Mature Enough for Production Use
  - Principal Developer of the ext3/4, Theodore Ts'o, [from wikipedia] stated that although ext4 has improved features, **it is not a major advance**, **it uses old technology**, and **is a stop-gap**. Ts'o believes that **Btrfs is the better direction** because "it offers improvements in scalability, reliability, and ease of management".



Overview of Btrfs

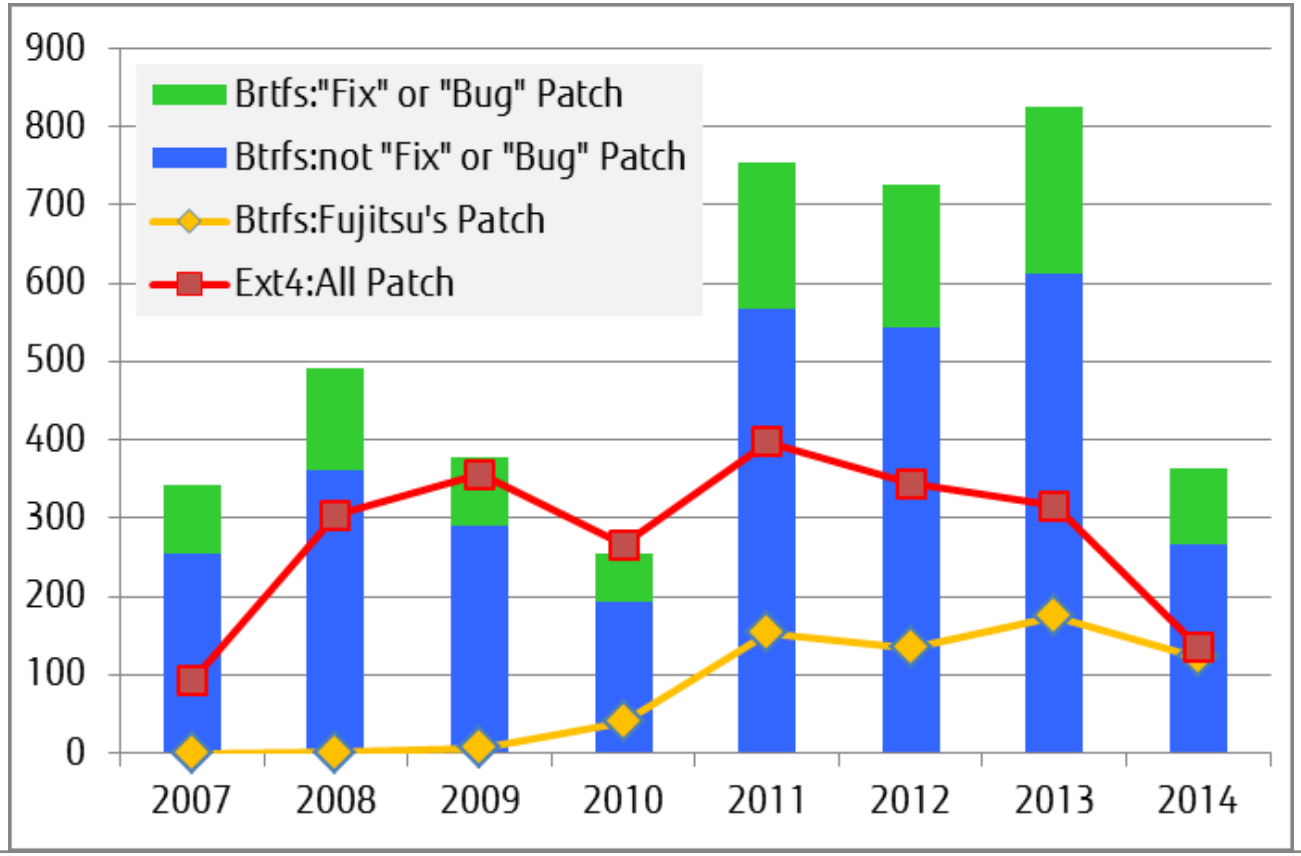
FUJITSU

- File System aimed at implementing Advanced Features while focusing on Fault Tolerance, Repair and Easy Administration
- Development began at Oracle in 2007, Merged in Mainline **Kernel 2.6.29 in Jan 2009**
- Key Features
  - Btree Data Structures, Copy on Write (**CoW**)
    Logging All Data and Metadata (→ Data Consistency and Easy Snapshots)
  - Writable and Read-only **Snapshots**, **Transparent Compression** ,RAID, …
- Supporting Distributions
  - MeeGo as Standard File System since 2010
  - OpenSUSE 13.2 using Btrfs by **Default** will be released in Nov 2014
  - Oracle Linux since 2012
  - RHEL 7 as a Tech Preview → Btrfs may be supported by Next Version of RHEL
- Facebook
  - Uses Btrfs on their Web Servers

Overview of Btrfs (contd.)

FUJITSU

- Development Status
  - Some Features are Under Development
  - Development has been More Active in the Last Few Years (Twice as Many Patches as Ext4)



FUJITSU

Evaluation of File System Requirements

Evaluation Plan

FUJITSU

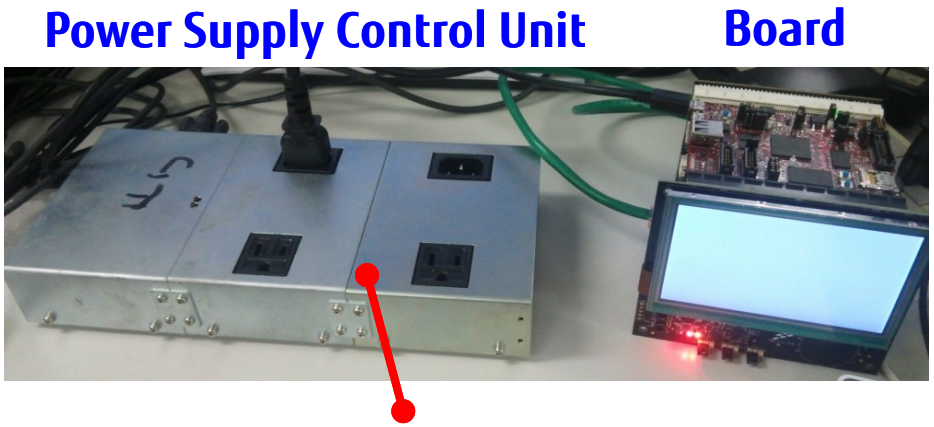
- Evaluated Characteristic Requirements of IVI
- Target File System : Btrfs and Ext4

- Eval 1 : Robustness 
  - Power Failure Tolerance
- Eval 2 : Boot Up Time 
  - FS Mount Time
- Eval 3 : Performance 
  - Basic File I/O Throughput
  - File I/O Throughput under High Load

Eval 1 : Robustness

FUJITSU

- Tolerance to Unexpected Power Failure while Writing to Files
- Eval Environment
  - Board
    - Name : Freescale TWR-VF65GS10
    - Processor : Vybrid VF61NS151CMK50
      500MHz Cortex-A5 +
      167MHz Cortex-M4 (not used)
    - Memory : 1GB DDR3
    - Storage : 16GB Micro SD Card



  - Software
    - Yocto based Fujitsu In-House Distro with Kenel 3.15-rc7
- Tools
  - Power Supply Control Unit
    - Periodically Turns On and Off DC Power Supply every Minute
  - File Writing Application
    - Continuously Creates 4KB Files and Writes to it

Eval 1 : Robustness (contd.)

FUJITSU

Analysis of Results

	Number of Power Failure	Results
Btrfs	1,000+	No Abnormal Situation Occurred
Ext4	1,000+	**Corrupted inode** had increased up to 32,000 and Finally Fell into Abnormal **Disk Full** State

- CoW of Btrfs showed Very Strong Power Failure Tolerance

Abnormal State of Ext4

Normal

```
# df -k -T
Filesystem           Type 1K-blocks      Used Available Use% Mounted on
/dev/mmcblk0p4       ext4 7206100    148172  6668836   2% /media/mmcblk0p4
```

Abnormal

```
# df -k -T
Filesystem           Type 1K-blocks      Used Available Use% Mounted on
/dev/mmcblk0p4       ext4 7206100    7189712         0 100% /media/mmcblk0p4
```

fsck.ext4

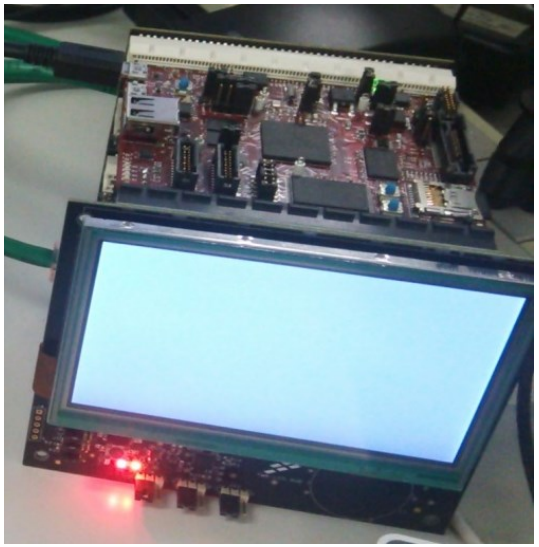
- Needed to Finish Fsck for **3 Minutes** and Recovered to Normal State

Eval 2 : Boot Up Time

FUJITSU

- Time Length of File System Mount
  - Measured after Boot Sequence was Completed in order to measure Real Mount Time
- Eval Environment (Almost the Same as Eval 1)
  - Board : Freescale TWR-VF65GS10
  - Software : Fujitsu In-House Distro with Kernel 3.15.1
- Tools
  - Kernel Code Customization
    - to Record Timestamps at Start and End Point of Mounting Process
- Conditions
  - Number of Files : about 4000, 70000, 1000000

Board

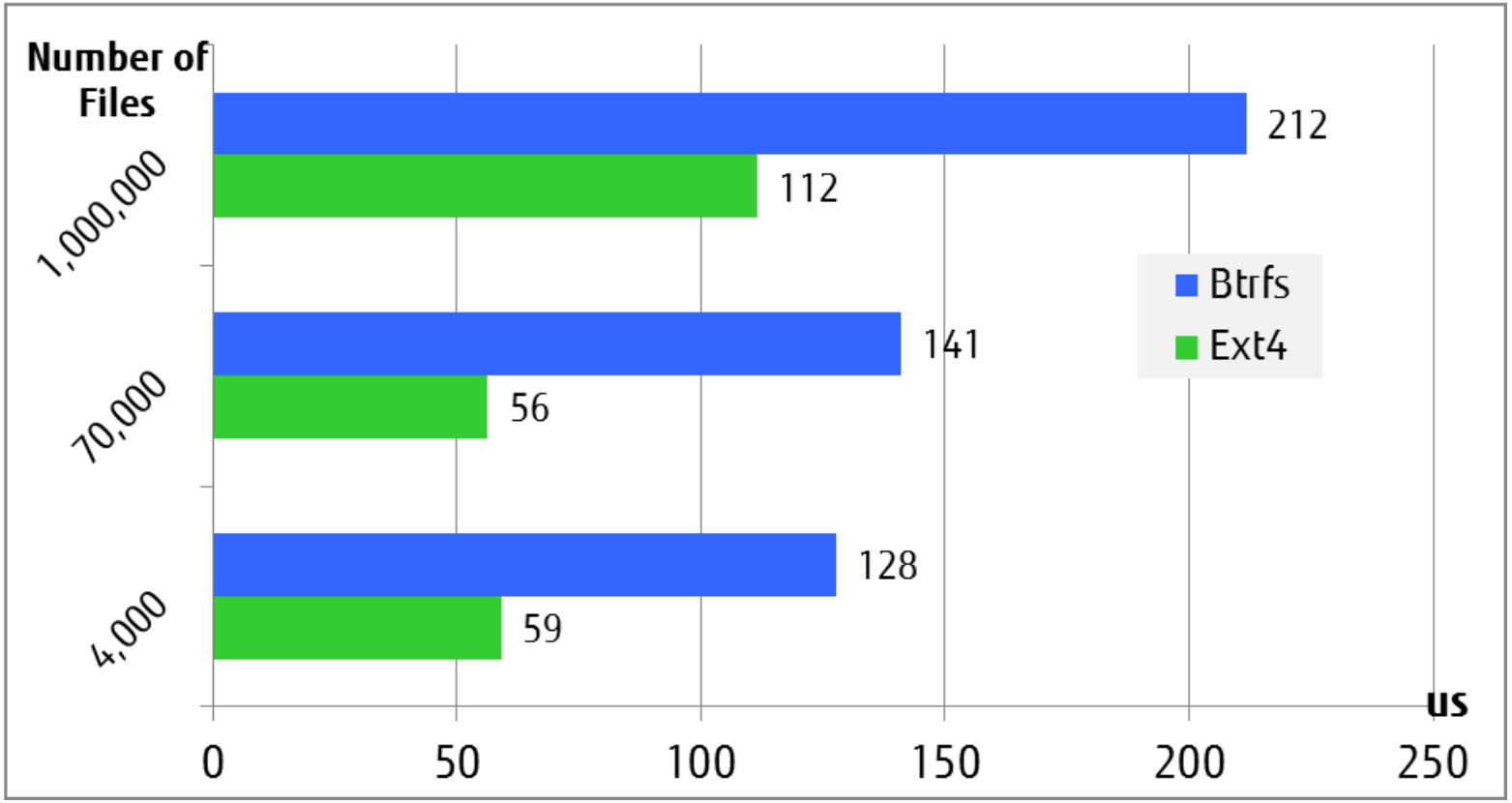


Eval 2 : Boot Up Time (contd.)

FUJITSU

Analysis of Results

- Btrfs options : rw,relatime,ssd,space_cache
- Ext4 options : rw,relatime,data=ordered



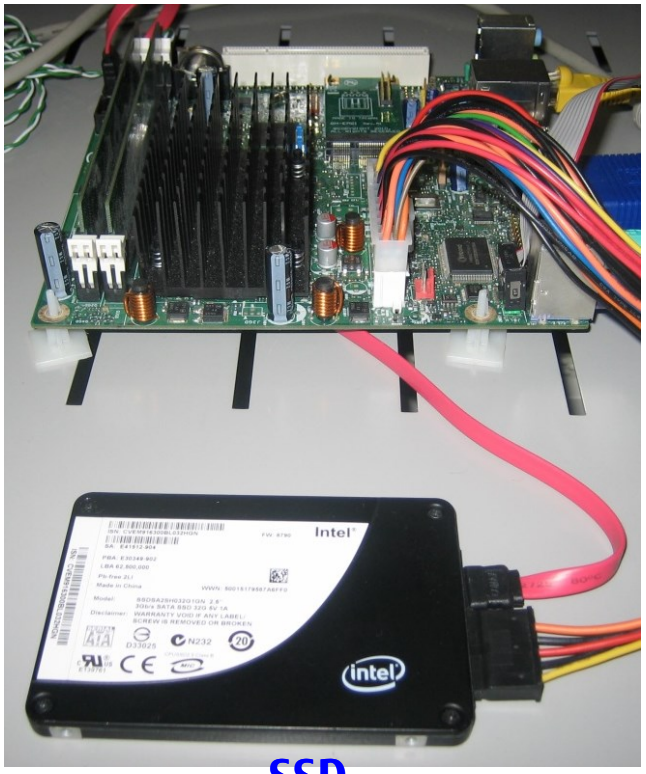
- Ext4 was mounted twice or 3 times Faster than Btrfs
  → Considering Cause : Btrfs has Dynamic inode Allocation
- Mount Time < 250us may have Tiny Impact on a Few Sec Boot Time Reqs

Eval 3 : Performance

FUJITSU

- Basic File I/O Throughput and Throughput under High Load
- Eval Environment
  - Board
    - Name : Intel Desktop Board D510MO
    - Processor : 1.66 GHz Dual-Core Atom (4 Core with HT)
    - Memory : 1GB DDR2-667 PC2-5300
    - Storage : 32GB Intel X25-E e-SATA SSD
      Sustained Seq R: up to 250MB/s, Seq W: up to 170MB/s
  - Software : Fedora 20 (x86_64) with Kenel 3.15.1
- Tools
  - FIO : to Benchmark and to Make High Load
    (with "yes >> /dev/null" for Userspace Load)
- Conditions
  - Single (for Basic) and Multiple (for High Load) FIO Running
  - FIO makes One Large File (R:2GB, W:1GB) and Reads from/Writes to the Same File with Small Block Size (Seq:64KB, Rand:4KB) (to Simulate DB like Behavior)
  - Some Combinations of Throughput-Related Mount Options

Board

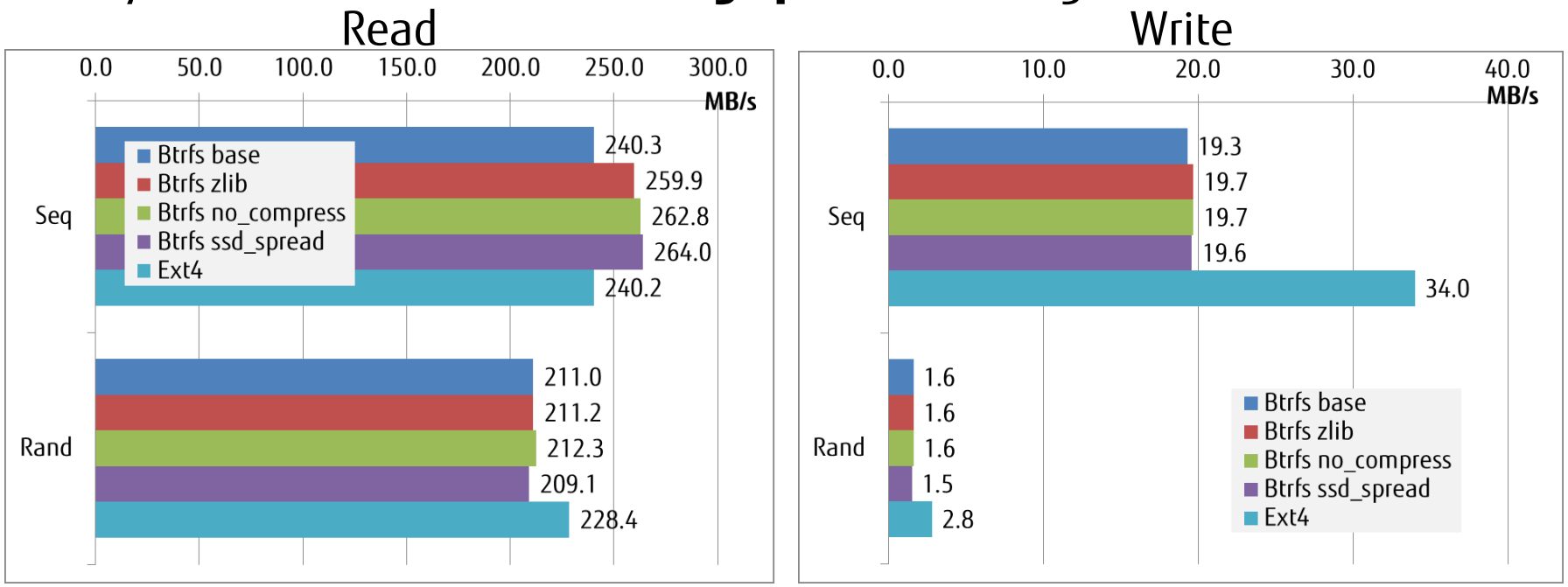


SSD

Eval 3 : Performance (contd.)

FUJITSU

Analysis of Results : **I/O Throughput** with Single FIO



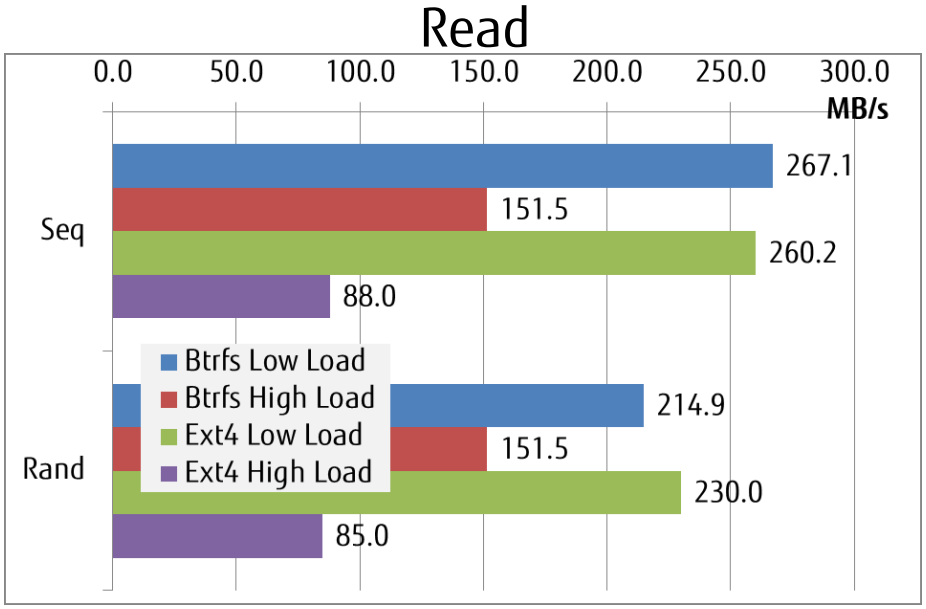
- Read : in Seq, Btrfs was Slightly Faster than Ext4, Some Opts were Effective, in Rand, Results were Reversed
- Write : Ext4 was almost Twice Faster than Btrfs → Considering Cause : Btrfs uses CoW
- Every FS has Advantages and Disadvantages, We could see the Other Results on Other Use Cases
- phoronix.com's Benchmark Results show Btrfs was the Overall Winner

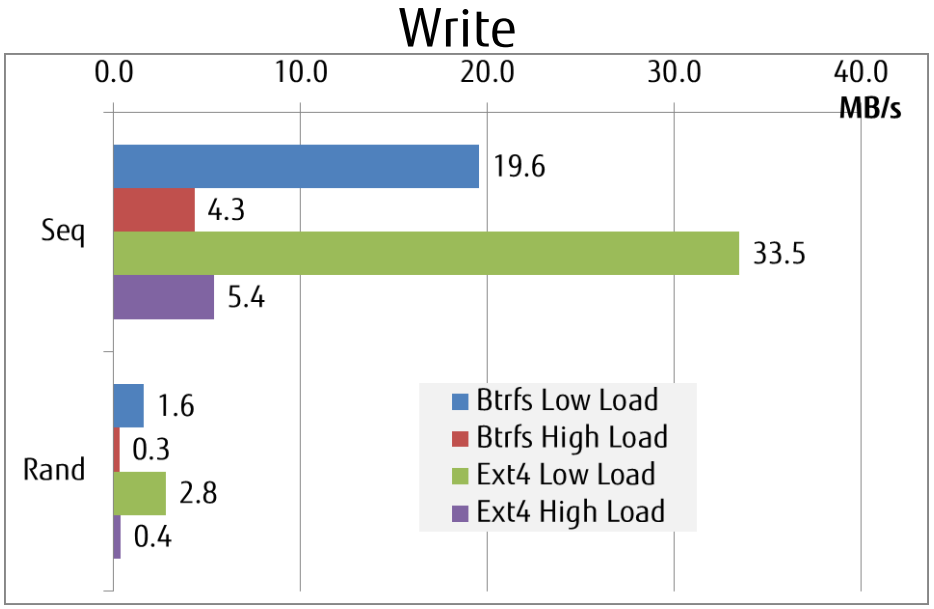
- Btrfs base options : rw,noatime,compress=lzo,ssd,discard,space_cache,autodefrag,inode_cache
- Ext4 options : rw,noatime,discard
- File Open with O_SYNC flag, Block Size : Seq 64KB, Rand 4KB, I/O Scheduler : noop
- Average of 3 Attempts

Eval 3 : Performance (contd.)

FUJITSU

Analysis of Results : **I/O Throughput** under **High Load**





- Ext4 : Every I/O Throughput Decreased Significantly under High Load
  Btrfs : Decreased Less than Ext4
  → Considering Cause : Kernel Threads of Btrfs used CPU Resource Effectively

- Btrfs options : rw,noatime,compress=lzo,ssd,discard,space_cache,autodefrag,inode_cache
- Ext4 options : rw,noatime,discard
- File Open with O_SYNC flag, Block Size : Seq 64KB, Rand 4KB, I/O Scheduler : noop
- Average of 3 Attempts
- to Make High Load : FIO Seq Read x 2 + Rand Read x 2 + "yes >> /dev/null"

Conclusions

FUJITSU

- Suitability for AGL Requirements
  - **Ext4** and **Btrfs** are Most Suitable FS from Functional Aspects
  - Other FS (XFS, NILFS2, ...) may Need to be Evaluated
- Evaluation Results
  under Some Specific Environments and Conditions (Like This Study)

	Power Failure Tolerance	Mount Time	I/O Throughput
Btrfs	**5**	**4**	**Read:5, Write:2, HighLoad:3**
Ext4	**2**	**5**	**Read:5, Write:4, HighLoad:2**

Values: 5=Excellent, 4=Very Good, 3=Good, 2=Fair, 1=Poor

  - Effective Mount Options of Btrfs
    - Base : rw,noatime,compress=lzo,ssd,discard,space_cache,autodefrag,inode_cache
    - for Throughput compress : no compression > zlib > lzo
      SSD awareness : ssd_spread > ssd
- More Evaluations will be Needed for IVI,
  like phoronix.com's Great Work

For the Future

FUJITSU

Forecasting the Future ...

- HW Specs will become more Rich
  → Priority of Requirements may Change
  such as CPU/Memory Usage, Compression, Boot Up Time, ...
- Development of EVs/FCVs may Cause a Change for Requirements of File Systems
  → Power Failures may Almost Never Occur on EVs/FCVs?
- We have to Adapt File Systems to Those Changes Flexibly and Rapidly

- Fujitsu will Continue to Improve Btrfs
- Let's Use and Evaluate Btrfs with Various Requirements, Environments, and Conditions to Make Btrfs more Suitable for IVI!

FUJITSU

Questions?

Thank you for your Attention!

FUJITSU

References

Btrfs

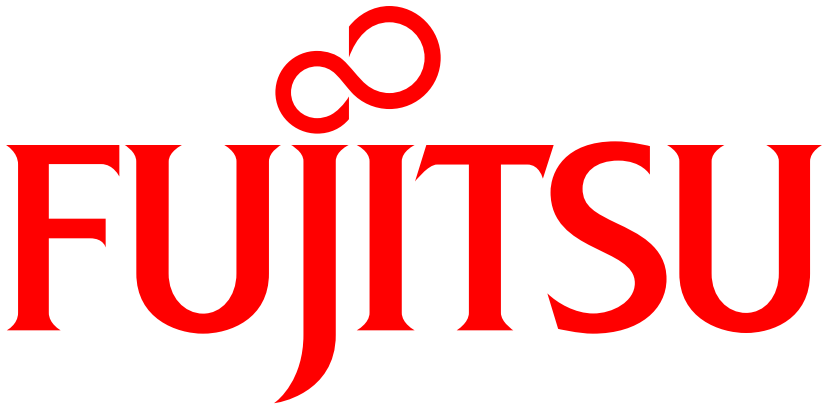
- https://btrfs.wiki.kernel.org/

Ext4

- http://en.wikipedia.org/wiki/Ext4

Benchmarking

- http://freecode.com/projects/fio
- http://www.phoronix.com/

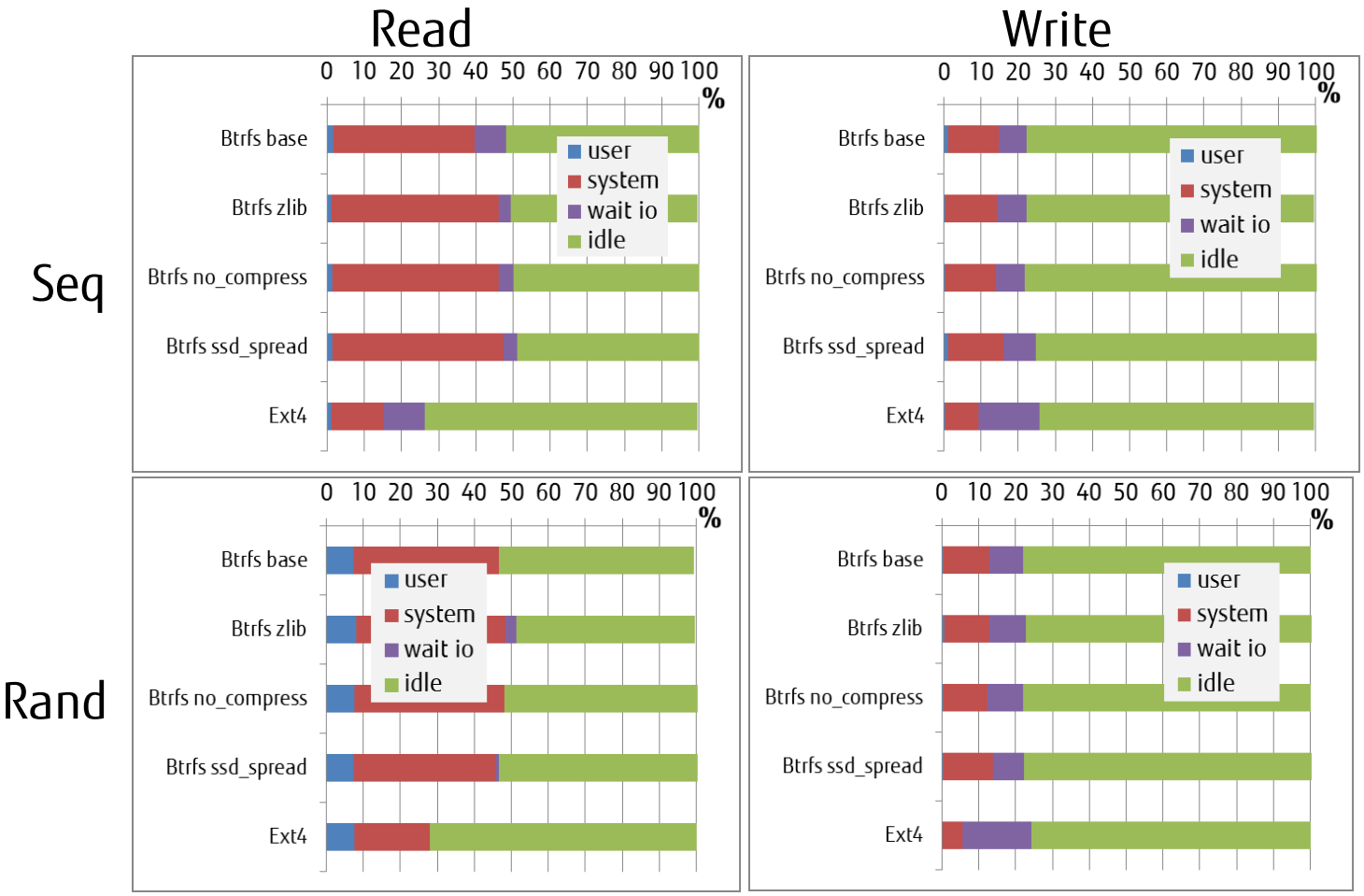


shaping tomorrow with you

Eval 3 : Performance

FUJITSU

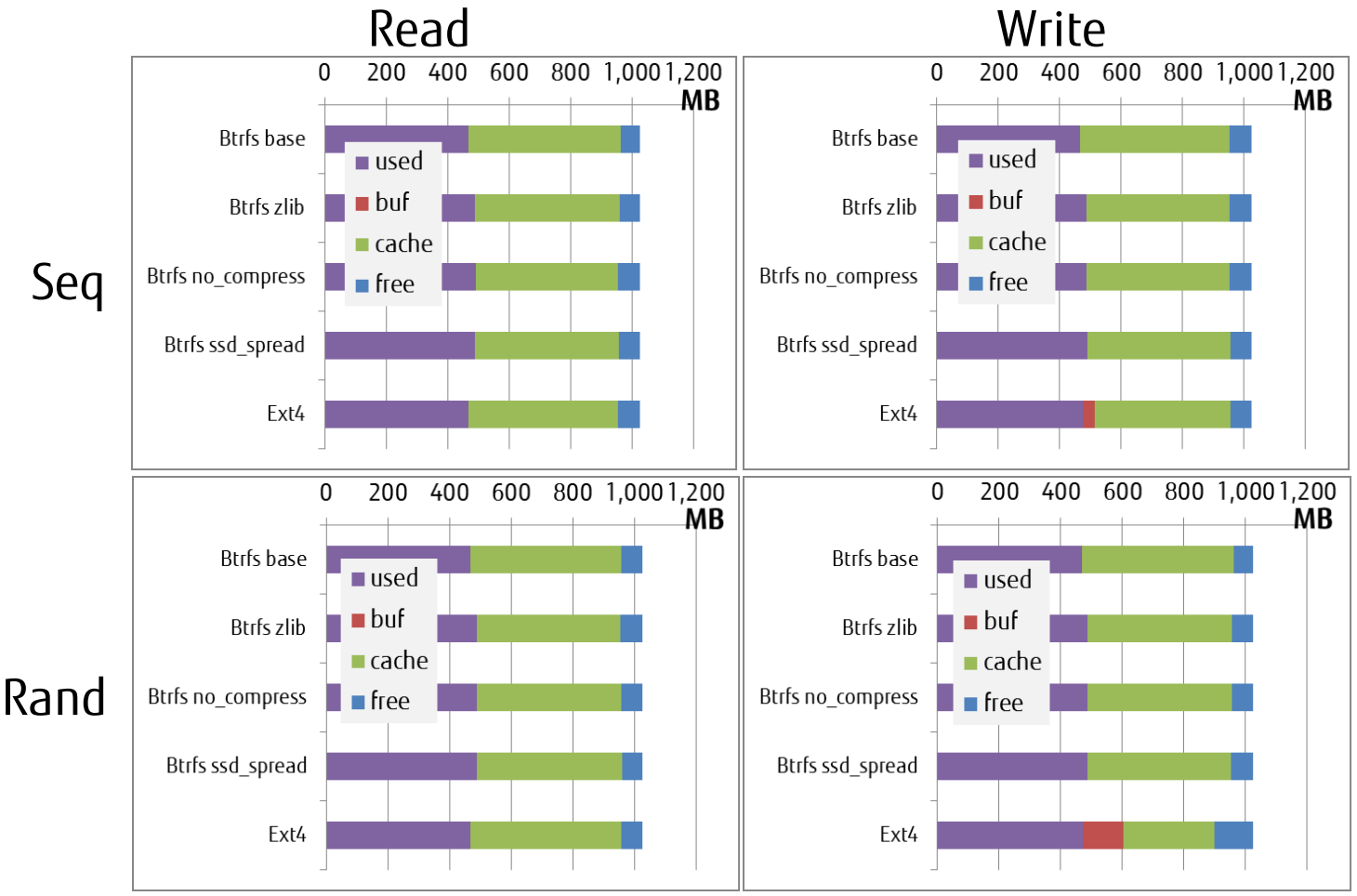
- Analysis of Results : **CPU Usage** with Single FIO



Eval 3 : Performance

FUJITSU

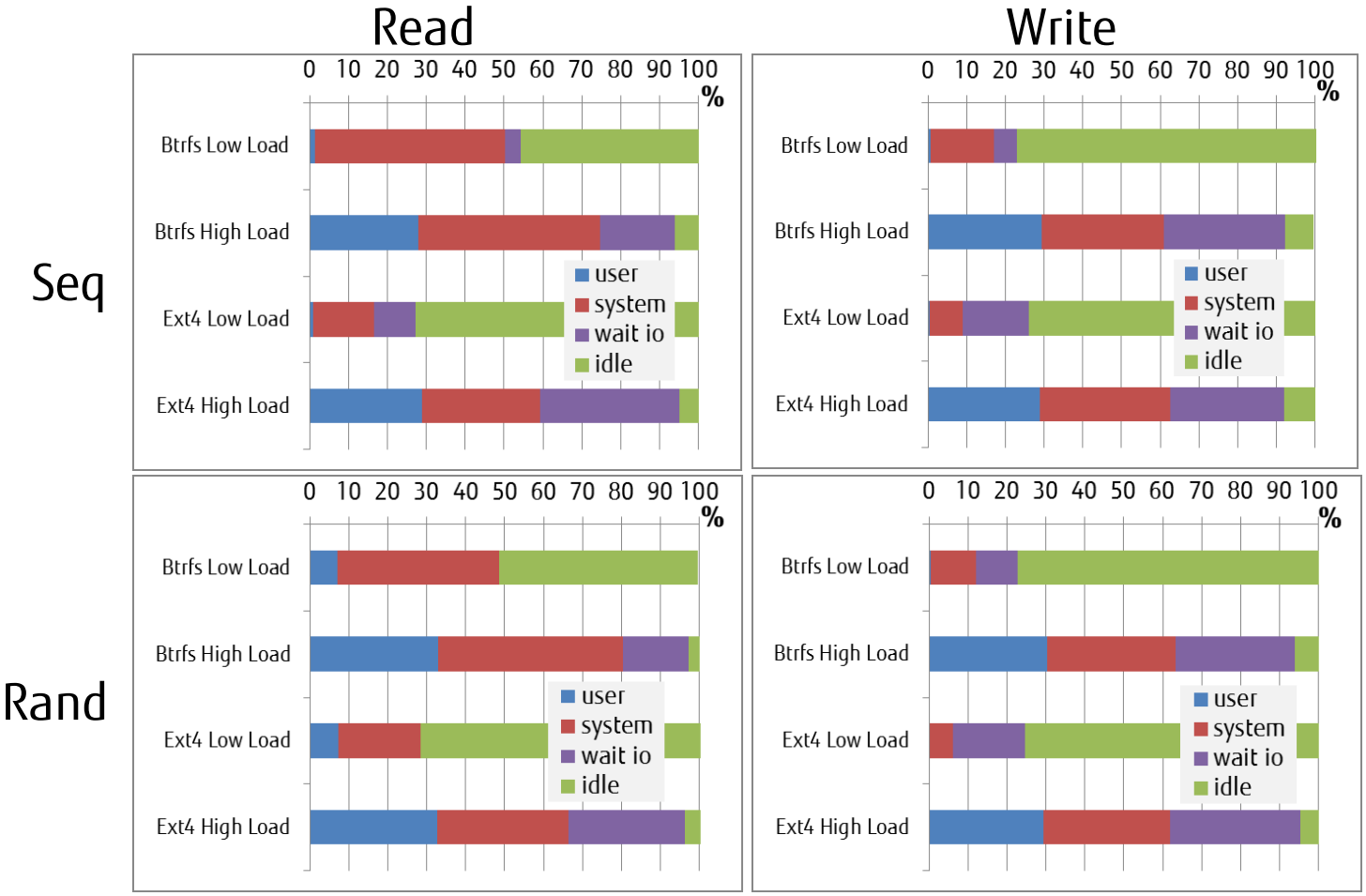
Analysis of Results : **Memory Usage** with Single FIO



Eval 3 : Performance

FUJITSU

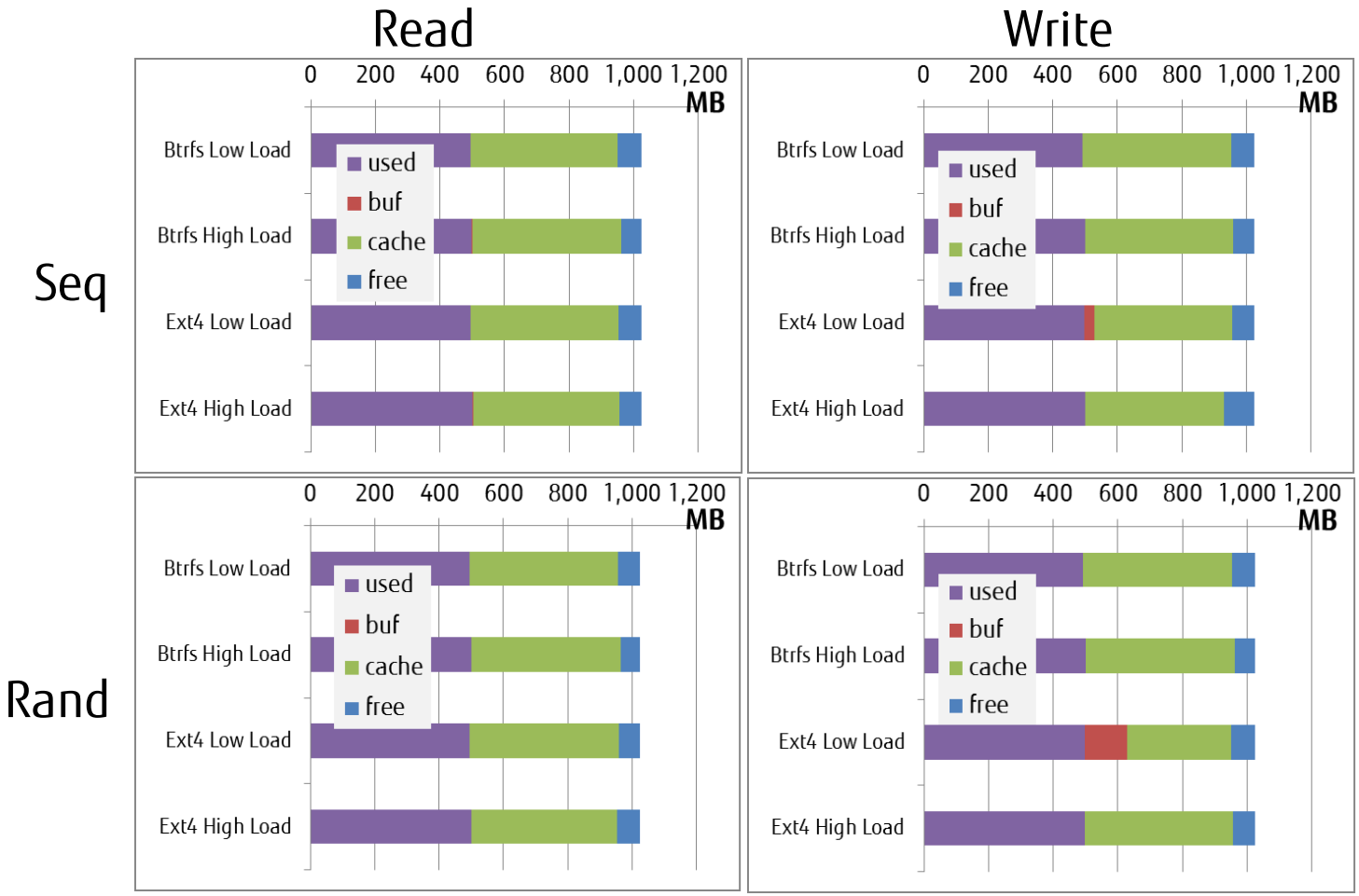
- Analysis of Results : **CPU Usage** under **High Load**



Eval 3 : Performance

FUJITSU

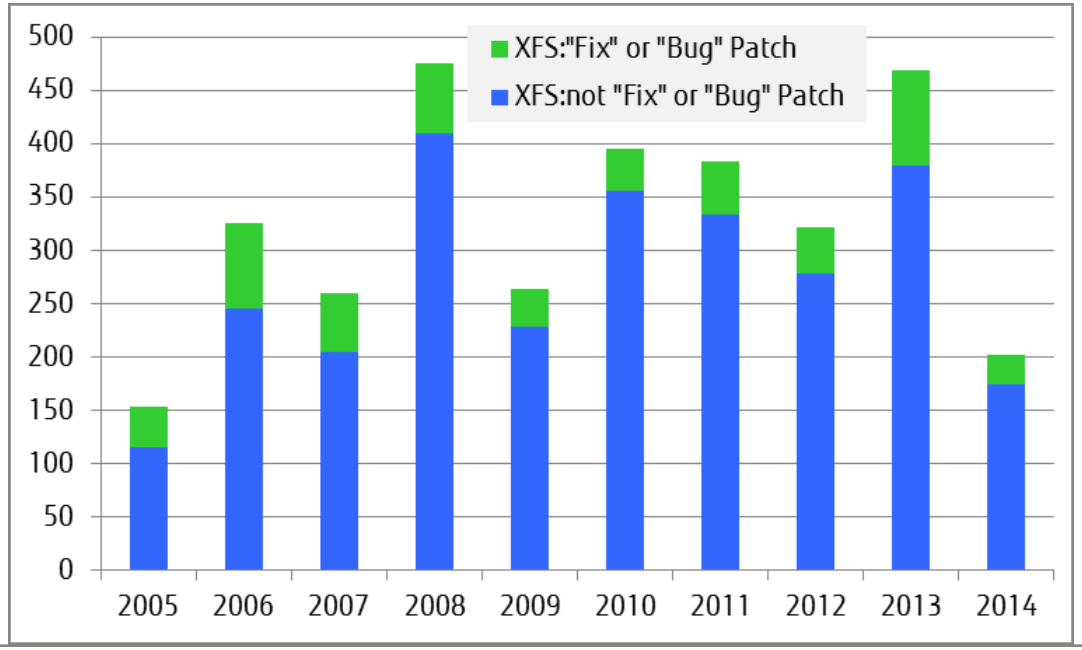
- Analysis of Results : **Memory Usage** under **High Load**



XFS : Another Candidate for IVI?

FUJITSU

- High-Performance 64-bit Journaling File System Created by Silicon Graphics
- Merged in Mainline **Kernel 2.4 around 2002**
- Key Features
  - B+tree Data Structures, Journaling, Allocation Groups, Striped Allocation, Delayed Allocation, Snapshots, Online Defragmentation/Resizing, ...
- Supporting OS : IRIX, Linux, FreeBSD, **Default FS in RHEL7**
- Development Status
  - Mature Enough for Production Use



NILFS2 : Another Candidate for IVI?

FUJITSU

- Log-structured File System
- Developed by NTT and Merged in Mainline **Kernel 2.6.30 in June 2009**
- Key Features
  - B-tree based Management, Quick Crash Recovery on Mount,
    Support Many Files/Large Files/Large Disks, Snapshots, Background Garbage Collection, ...
- Supporting OS : IRIX, Linux, FreeBSD
- Development Status