TOSHIBA
Leading Innovation

Collaboration with LTSI Testing

Yoshitake Kobayashi
Advanced Software Technology Group
Corporate Software Engineering Center
Toshiba Corporation

Overview

- Basic requirements
- What do we need to test?
  - Case studies
- Collaboration with LTSI Testing

Basic requirements for Linux kernel

- Stable
- Able to run as long as possible
- Able to migrate from one version to another

Basic requirements for Linux kernel

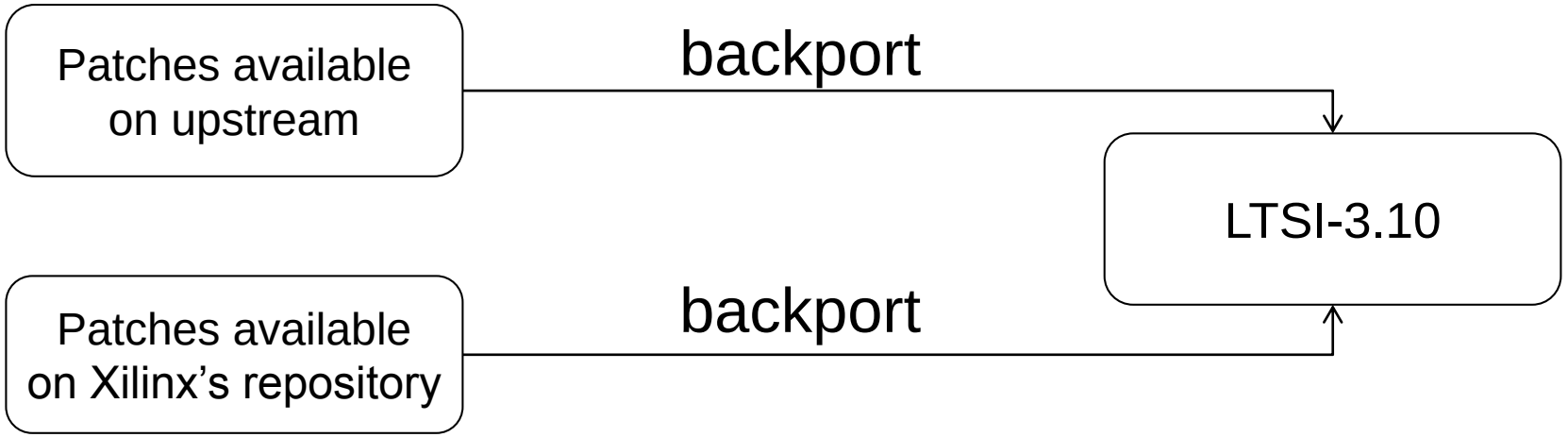
- Stable
  - No bug
  - Continue to fix bugs
- Able to run as long as possible
  - Already have some experience
- Able to migrate from one version to another
  - Evaluated migration effects
  - Fixed all compatibility issues

Required test case

- Categories of test case
  - Functionality (APIs)
  - Performance
  - Quality
  - Compatibility
- Example of test case
  - Functionality
    - LTP
    - . . .
  - Performance test
    - Cyclictest, Iozone
    - . . .
  - Quality test
    - Hardware resource isolation
    - Data reliability
    - Heat run
  - Compatibility
    - . . .

Case study: Backport Zynq support for LTSI-3.10

- Posted Xiinx Zynq (ZC702) support patches to LTSI-3.10



- Preparation
  - Backported required patches for Zynq to 3.10
  - Ask to Xilinx developer

Case study: Backport Zynq support for LTSI-3.10

- The following test has been done on ZC702
  - LTP
  - POSIX Testsuite
  - Devices
    - Ethernet
    - General Purpose I/O
    - I2C Controller
    - QSPI Flash Controller
    - SD Card
    - Timer
    - UART
    - Watchdog
- The following tests has not been done yet
  - Performance test
    - Cyclictest
  - Compatibility test

- Required test cases depend on the usage of the kernel

(Required test case)

- Categories of test case
  - Functionality (APIs)
  - Performance
  - Quality
  - Compatibility
- Example of test case
  - Functionality
    - LTP
    - . . .
  - Performance test
    - Cyclictest, Iozone
    - . . .
  - Quality test
    - Hardware resource isolation
    - Data reliability
    - Heatrun
  - Compatibility
    - . . .

Results of LTP on multiple kernels

- Evaluation environment
  - LTP
  - Userland from Debian 4.0

- Results

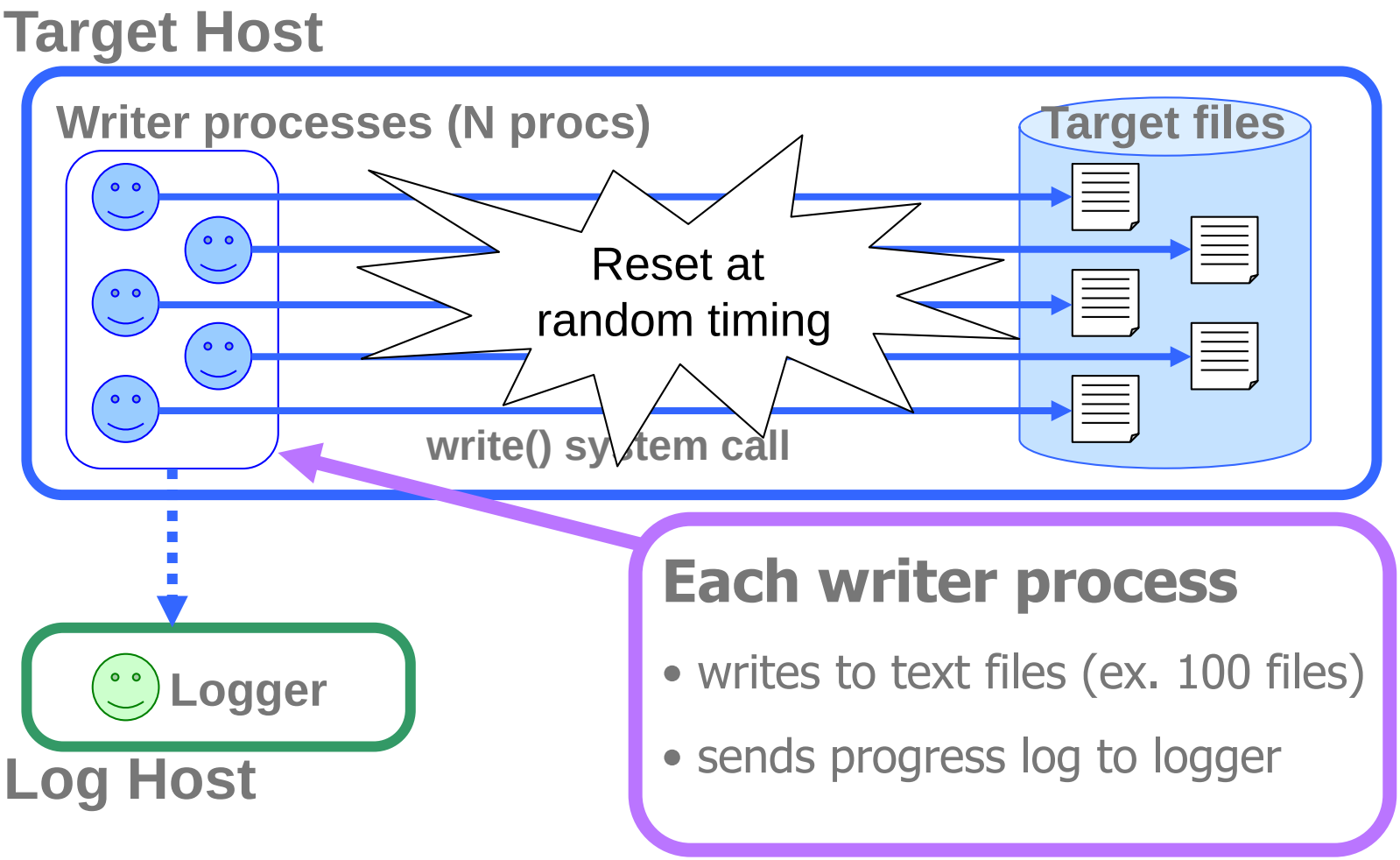
Version	Number of errors	Test case name
2.6.18	1	Cron2
2.6.26	3	getcpu01, stime01, cron02
2.6.32	7	execve04, getcpu01, swapon03, sched_cli_serv, clock_gettime03, timer_create04

- Reference: Moving Forward: Overcoming from Compatibility issues BoFs , ELC2011

Overview of data reliability test

This test case available at the following URL:

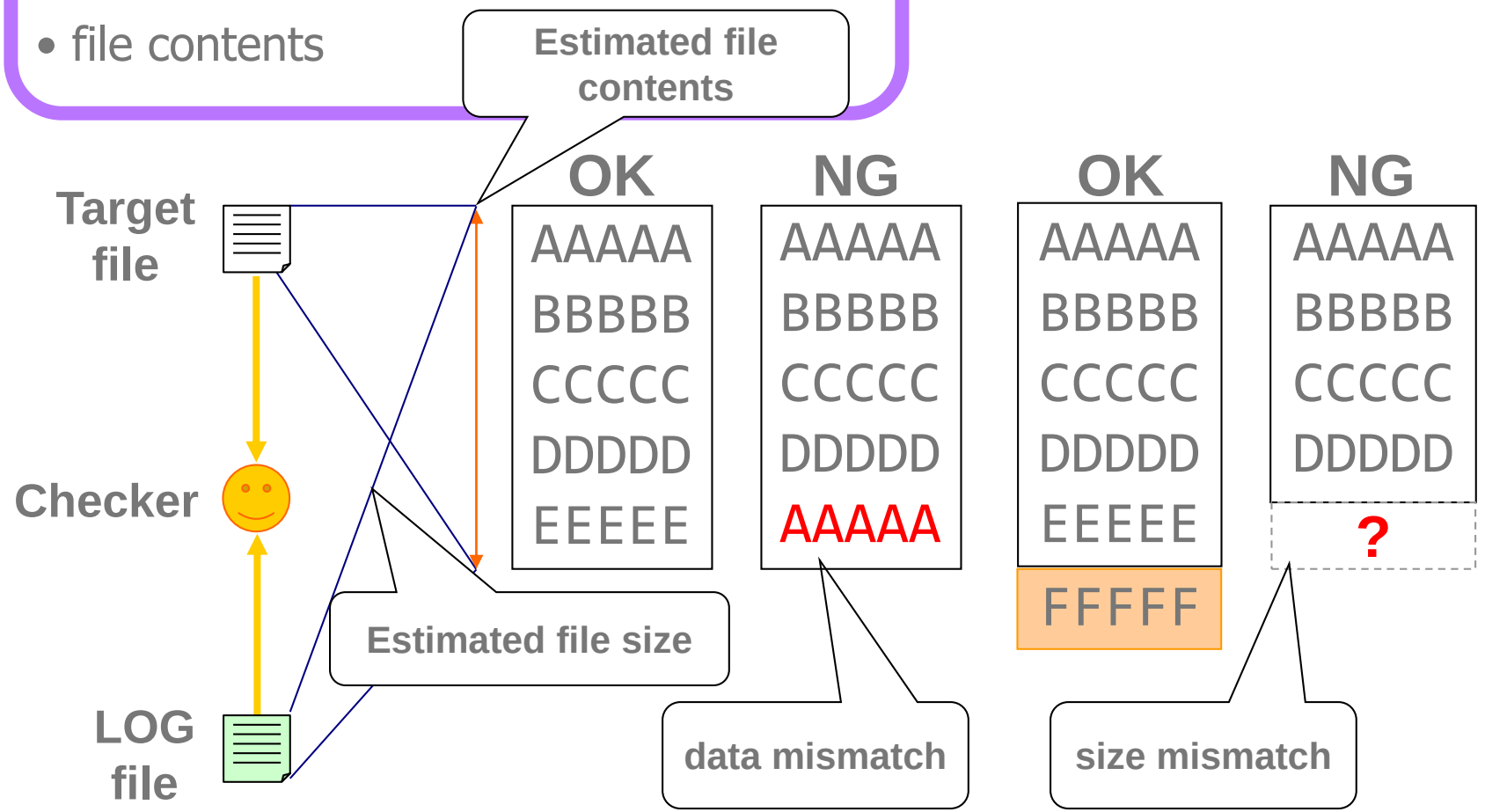
https://github.com/ystk/fs-test



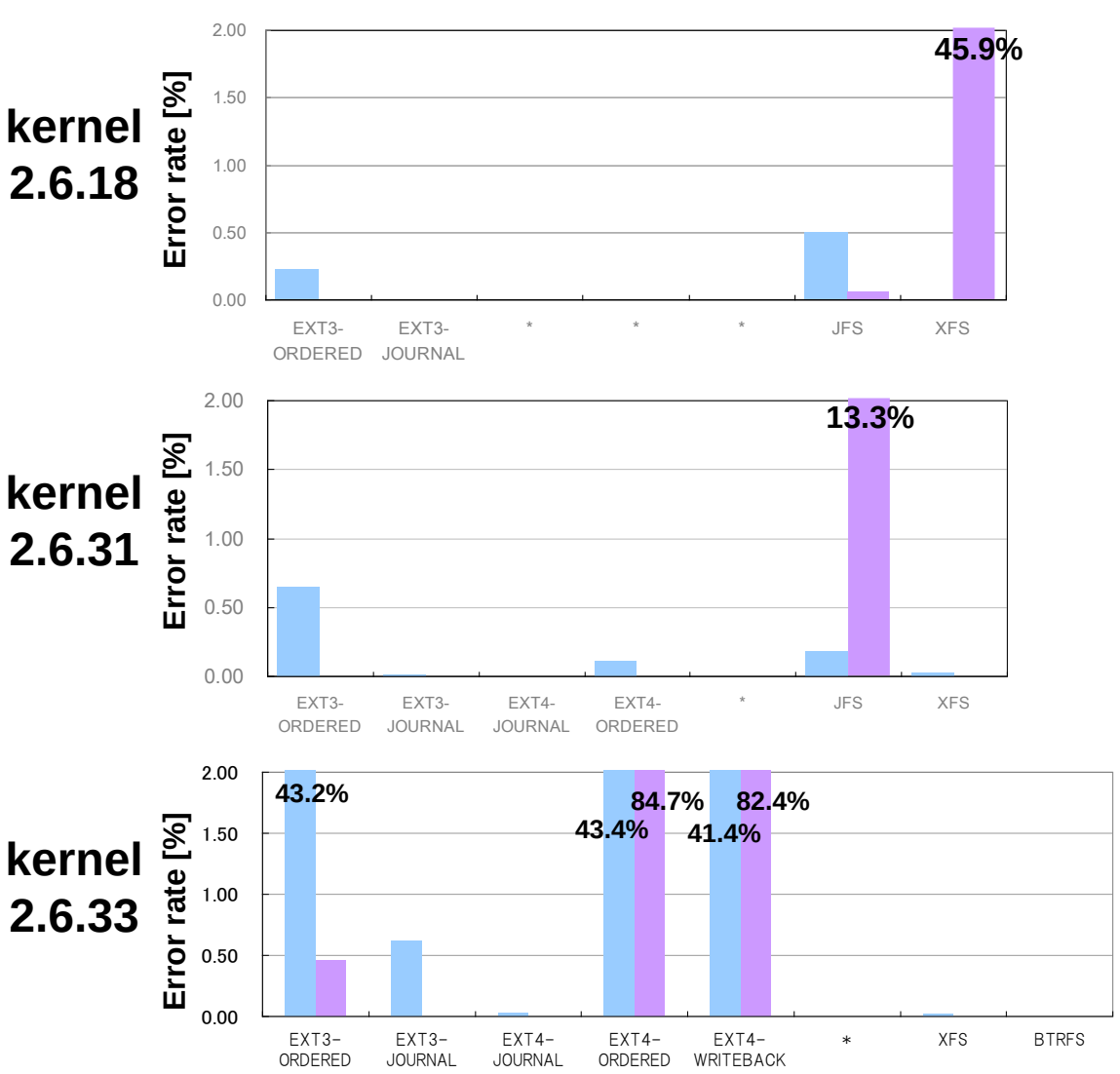
Verifying the data reliability

Verify the following metrics

- file size
- file contents

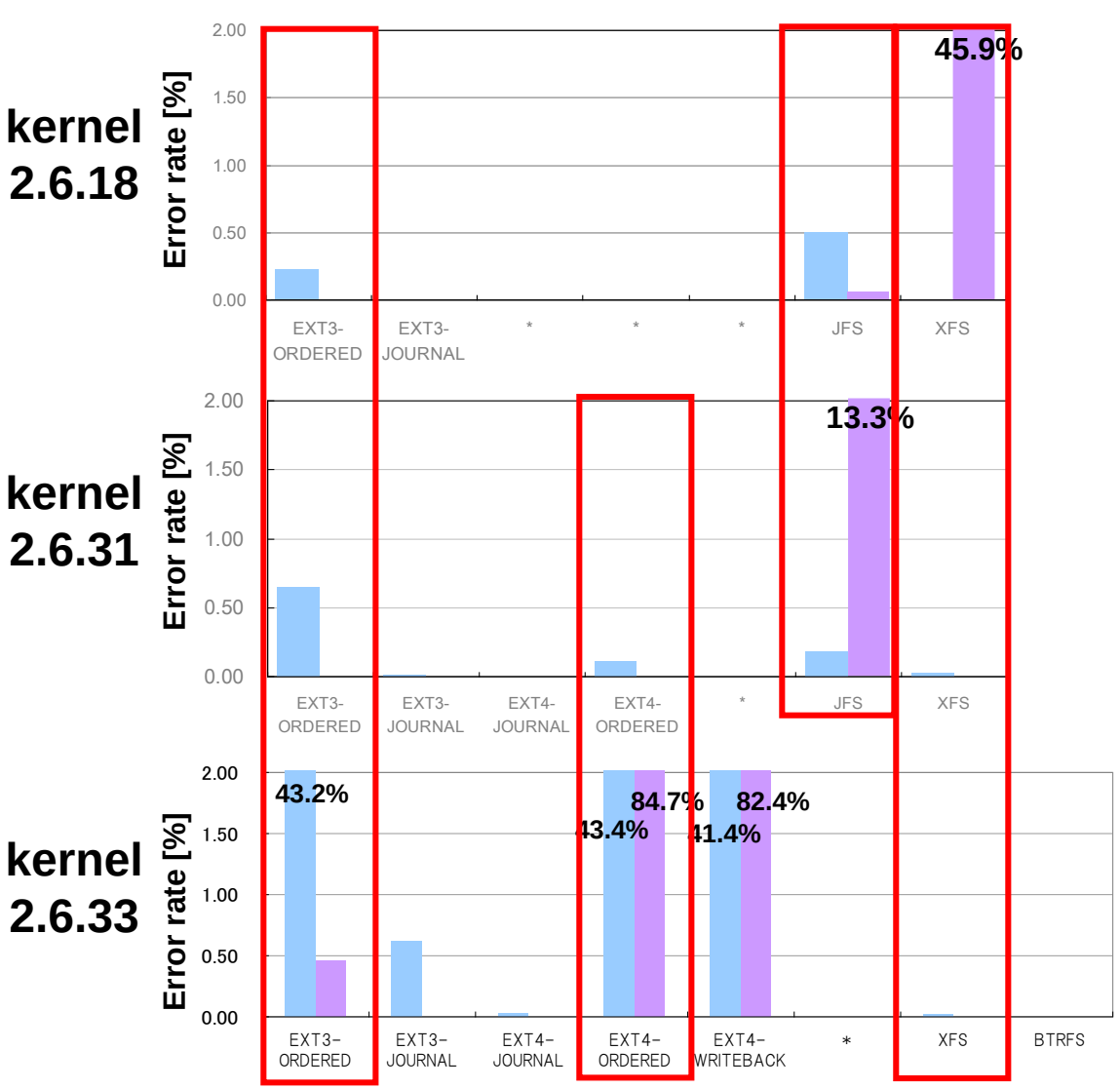


Results of data reliability test



- Reference: Evaluation of Data Reliability on Linux File Systems, ELC2010

Results of data reliability test

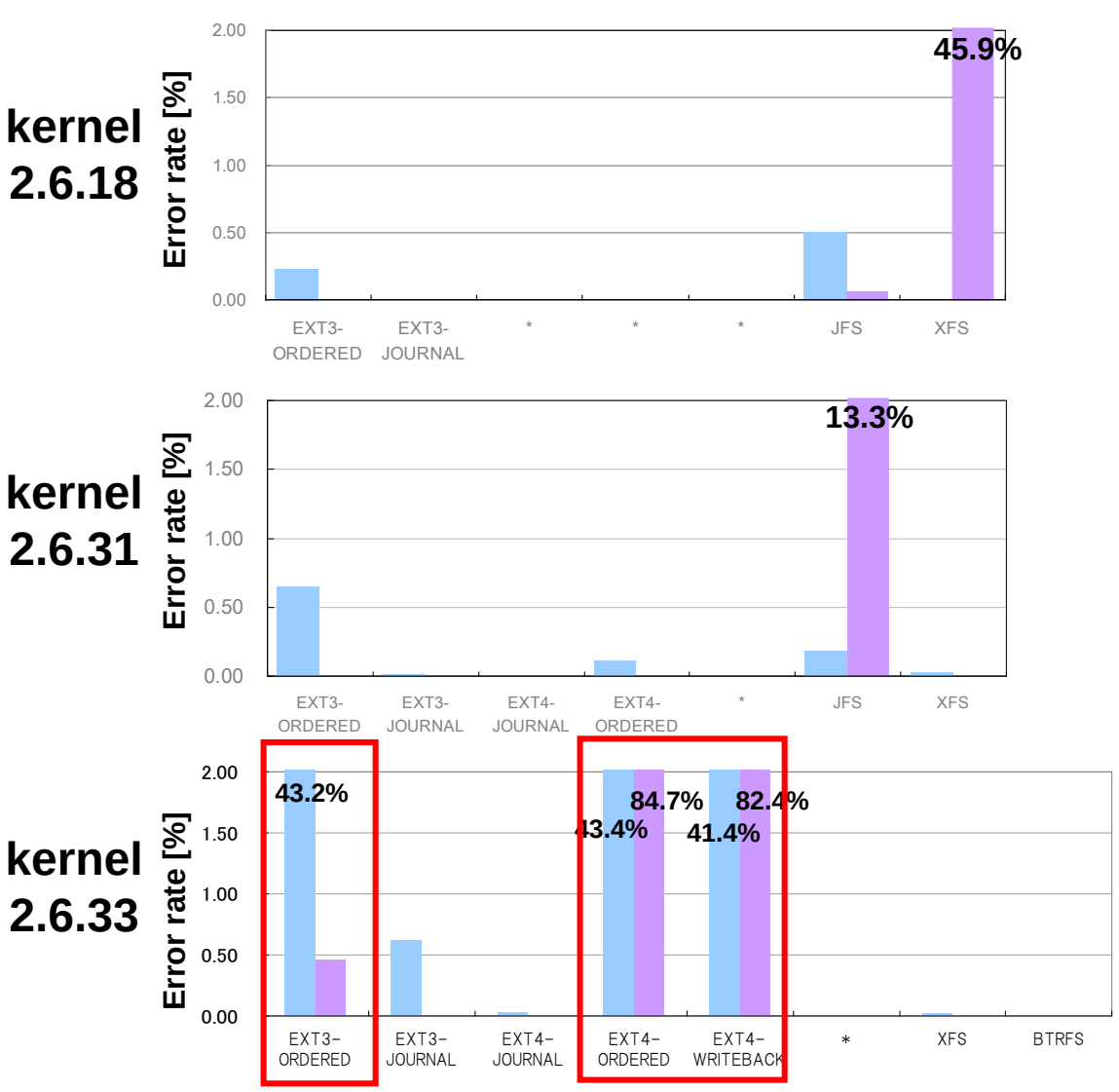


File size mismatch
Data mismatch

Point 1:
A file system has different characteristics of data reliability

- Reference: Evaluation of Data Reliability on Linux File Systems, ELC2010

Results of data reliability test



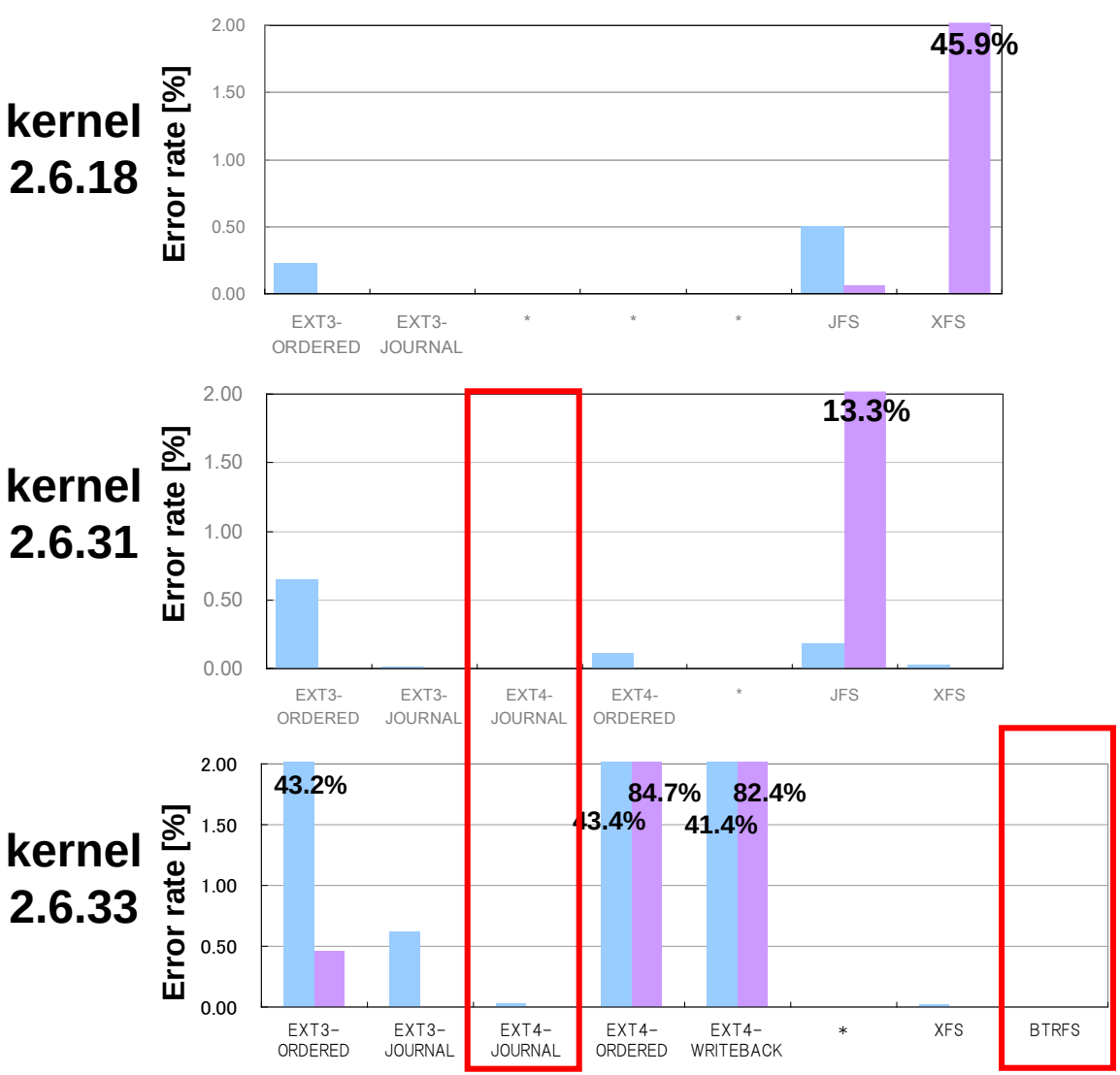
File size mismatch
Data mismatch

Point 1:
A file system has different characteristics of data reliability

Point 2:
Some Results depends on kernel version

- Reference: Evaluation of Data Reliability on Linux File Systems, ELC2010

Results of data reliability test



File size mismatch
Data mismatch

Point 1:
A file system has different characteristics of data reliability

Point 2:
Some Results depends on kernel version

Point 3:
EXT4-Journal and BTRFS has a nice result

- Reference: Evaluation of Data Reliability on Linux File Systems, ELC2010

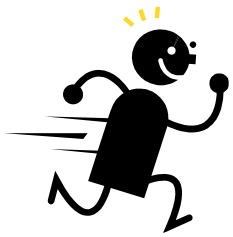
Linux Kernel Acceleration for Long-term Testing

Issues

- Long-term testing takes really long time
  → We want results as fast as possible

Accelerate

START



GOAL

Things that cannot be accelerated

- CPU clock
- I/O access speed (ex. SSD)
- Network bandwidth
- etc.

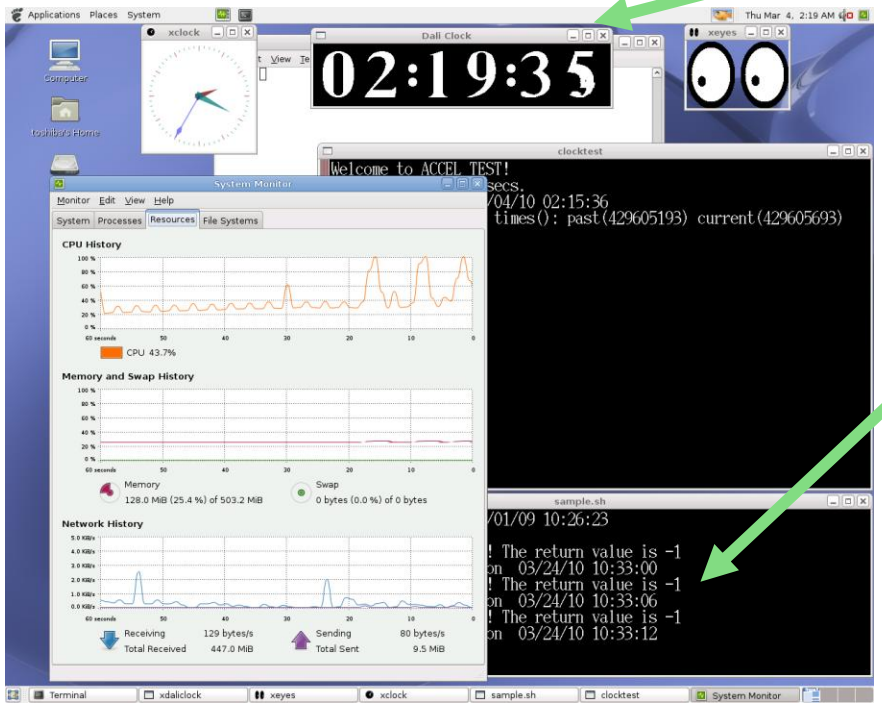
Focus to accelerate clock

Try to detect errors that caused by clock

- Reference: Linux Kernel Acceleration for Long-term Testing, ELC2010

Example of acceleration (A screenshot)

Xdaliclock works as a stopwatch



Returned an incorrect value after about 450 days.
(It takes about 6 hours in 1000 times acceleration)

- **Reference**: Linux Kernel Acceleration for Long-term Testing, ELC2010

Performance compatibility issues between 2.4 and 2.6

- Slow to run
  - context switches up to 96% slower
  - local communication latencies up to 80% slower
  - file system latencies up to 76% slower
  - local communication bandwidth less than 50% in some cases.

- Reference: http://www.denx.de/wiki/Know/Linux24vs26

Requirement for LTSI Testing

- Test set
  - OSS test suites like LTP, Iozone, lmbench
  - Deta reliability test
    - Runs on multiple file systems
    - Compare the results
  - Compatibility test
    - Aspects
      - API
      - Performance (I/O, Network and more)
      - Service quality

- How to test?
  - User land
    - Same user land for all kernel version
    - The latest version
  - Multiple CPU architectures
    - ARM, PowerPC, X86_32, X86_64

Issues

- Test result sharing
  - Each test result has deferent output format
  - Need to have the following features
    - Test result upload
    - Regression problem detection
- Device driver test cases
  - How to make common test sets

Collaboration with LTSI kernel/LTSI Testing

- Open all test results for basic test set
- Keep transparency of the test results
  - Open the specs of testing environment
- Run same tests on multiple environment
  - User can be refer the nearest setup to choose a hardware
- Give some aspects for long term support
  - Super long term support (ex. 20 years)
  - kernel migration to newer version
    - Compatibility test

- Merge RT-preempt to LTSI
  - LTSI-RT
    - https://github.com/ystk/linux-ltsi/tree/ltsi-3.0.y-rt
    - https://github.com/ystk/linux-ltsi/tree/ltsi-3.4.y-rt
    - https://github.com/ystk/linux-ltsi/tree/ltsi-3.10.y-rt (available soon)

Questions?