Expectation of LTSI Testing
- What do we need to test? (User’s viewpoint) -

Yoshitake Kobayashi
Advanced Software Technology Group
Corporate Software Engineering Center
Toshiba Corporation
Yoshitake Kobayashi (YOSHI)

- Chief Specialist at Corporate Software Engineering Center, TOSHIBA CORPORATION

- Providing embedded operating system knowledge
  - Linux
  - RTOS
  - Open source software license
Overview

- Basic requirements
- What do we need to test?
  - Case studies
- Expectation of 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
    - Data reliability
    - Heatrun
  - Compatibility
    - ...

Results of LTP on multiple kernels

- Evaluation environment
  - LTP
  - Userland from Debian 4.0

- Results

<table>
<thead>
<tr>
<th>Version</th>
<th>Number of errors</th>
<th>Test case name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.6.18</td>
<td>1</td>
<td>Cron2</td>
</tr>
<tr>
<td>2.6.26</td>
<td>3</td>
<td>getcpu01, stime01, cron02</td>
</tr>
<tr>
<td>2.6.32</td>
<td>7</td>
<td>execve04, getcpu01, swapon03, sched_cli_serv, clock_gettime03, timer_create04</td>
</tr>
</tbody>
</table>

- Reference: Moving Forward: Overcoming from Compatibility issues BoFs, ELC2011
Overview of data reliability test

Test case available at:
https://github.com/ystk/fs-test

Target Host

Writer processes (N procs)

Reset at random timing

write() system call

Each writer process
- writes to text files (ex. 100 files)
- sends progress log to logger

Log Host

Logger
Verifying the data reliability

Verify the following metrics

- file size
- file contents

Estimated file contents

Target file

Estimated file size

OK

AAAAA
BBBBBB
CCCCC
DDDDD
EEEEEE

NG

AAAAA
BBBBBB
CCCCC
DDDDD
AAAAA

OK

AAAAA
BBBBBB
CCCCC
DDDDD
AAAAA

NG

AAAAA
BBBBBB
CCCCC
DDDDD
FFFFF

data mismatch

size mismatch
Results of data reliability test

- Reference: Evaluation of Data Reliability on Linux File Systems, ELC2010
Results of data reliability test

- **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

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)

- Xdalliclock 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](http://www.denx.de/wiki/Know/Linux24vs26)
Requirement for LTSI Testing

- **Test set**
  - OSS test suites like LTP, Iozone, Imbench
  - 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
Expectation of LTSI Testing

- Open all test results for basic test set
- Keep transparent of the test results
  - Open the spec 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
- Merge RT-preempt to LTSI
  - Test results also needed
  - 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
Questions?