Moving Forward: Overcoming from Compatibility issues BoFs

Yoshitake Kobayashi
Advanced Software Technology Group
Corporate Software Engineering Center
TOSHIBA CORPORATION

13 Apr, 2011

Copyright 2011, Toshiba Corporation.
Problem definition

Product life cycle: more than 10 years

- Test cases
- Application
- Libraries
- Kernel
- Hardware
Problem definition

Product life cycle: more than 10 years

- Test cases
- Application
- Libraries
- Kernel
- Hardware

some years later

- Test cases
- Application
- Libraries
- Kernel
- Hardware

Oh! No!
Problem definition

Product life cycle: more than 10 years

Test cases
Application
Libraries
Kernel

Hardware

Test cases
Application
Libraries
Kernel

Hardware

some years later

Maybe, I’m here

Oh! No!
Problem definition

To overcome the hardware discontinuation:

- Is new hardware introduction enough?
  - No!

- Need to check supported hardware for the current kernel
  - Mostly not….

Test cases
Application
Libraries
Kernel
Hardware

Newer Hardware

approx 5 years

or

Quite new Hardware

Oh! No!
Question 1: How to adapt the new hardware

■ Which approach is the better?
  ■ Upstream kernel driver backports on the old kernel
  ■ Change the current product’s kernel to newer one
Question 1: How to adapt the new hardware

- Which approach is the better?
  - Upstream kernel driver backports on the old kernel
  - Change the current product’s kernel to newer one

- The answer is …
  - I’m not sure

- The correct answer depends on:
  - Hardware specification
  - User’s (or Programmer’s) requirement

- In this BoF, think about use newer kernel version to move forward
Question 2: Required tests

- What kind of test do I need to do to make sure the compatibility?
Question 2: Required tests

- What kind of test do I need to do to make sure the compatibility?

- The following slides describe three aspects:
  - API level
  - Performance verification
  - Service quality verification
Case study 1: API level tests

- Test environment
  - Same libraries and testcases are used on each kernel version
LTP results

- C1: 2.6.18-etch + Etch environment
  - Error count that failed only on new hardware: 1
    - cron02
  - Note: This kernel doesn’t fully support new hardware

- C2: 2.6.26-lenny + Etch environment
  - Error count that failed only on new hardware: 3
    - getcpu01, stime01, cron02
    - needs to run separately: mtest06, cron_deny01
  - Note: This kernel supports almost all devices on new hardware

- C3: 2.6.32-squeeze + Etch environment
  - Error count that failed only on new hardware: 7
    - execve04, getcpu01, swapon03, sched_cli_serv, clock_gettime03, timer_create04
  - Note: This kernel supports almost all devices on new hardware
Determine the reason for the errors on 2.6.26

- getcpu01
  - Only runs >2.6.20
  - Need NUMA support

- stime01
  - time() returns stime()-1
  - A bug fix is available on 2.6.27.13
  - Easy to fix
Case study 2: Performance verification

- CPU performance
  - Has to be better than old one if the application’s CPU usage is high
    - This is easy to fix
- Network throughput
- I/O throughput

- The following aspects are important for real-time systems
  - Scheduling latency
  - Network latency
  - ...(any others?)
Latency test (cycle 300μs / cpu and memory load)

CPU load

0 %

50 %

100 %
Latency test (cycle 300µs / CPU load only)

<table>
<thead>
<tr>
<th>CPU负荷</th>
<th>2.6.31.12</th>
<th>2.6.31.12-RT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 %</td>
<td><img src="image1.png" alt="Graph" /></td>
<td><img src="image2.png" alt="Graph" /></td>
</tr>
<tr>
<td>50 %</td>
<td><img src="image3.png" alt="Graph" /></td>
<td><img src="image4.png" alt="Graph" /></td>
</tr>
<tr>
<td>100 %</td>
<td><img src="image5.png" alt="Graph" /></td>
<td><img src="image6.png" alt="Graph" /></td>
</tr>
</tbody>
</table>
Why this happens?

- **Probably** hardware problem
  - Try to find the bottlenecks by ftrace
    - This latency problem randomly happens in the kernel
    - If same test runs on other machines, nothing happened

- In this case, just throw away the hardware
  - or ask customer service
Case study 3: Quality verification

- Quality verification for:
  - File systems
  - Long-term running
Results of data reliability tests

Point 1:
An filesystem has different characteristics on different kernel

Point 2:
2.6.33 has high error rate on ordered and writeback mode

Point 3:
Ext4-journal and Btrfs has good results

File system types

Error rate [%]

File size mismatch rate
Data mismatch rate

kernel version

kernel 2.6.18

kernel 2.6.31

kernel 2.6.33

Point 1:
An filesystem has different characteristics on different kernel

Point 2:
2.6.33 has high error rate on ordered and writeback mode

Point 3:
Ext4-journal and Btrfs has good results
Conclusion

- This BoF discussed what kind of tests are required to follow newer kernel version
  - The following aspects are only examples
    - API level compatibility
    - Performance compatibility
    - Service quality compatibility

- Of course, application specific tests has to run