Recovering Device Drivers

Michael M Swift, Muthukaruppan Annamalai, Brian N. Bershad, and Henry M. Levy
Department of Computer Science and Engineering, University of Washington
USENIX OSDI 2004

Presented by Shian-Tai Chiou
08/22/07
Outline

- Introduction
- Device Drivers
- Shadow Driver Design
- Shadow Driver Implementation
- Evaluation
- Conclusions
Introduction

- Improve OS reliability
- Shadow driver
  - Conceal a driver’s failure from its client
  - Recovery from failure
- Four principles
  - Failure should be concealed from clients
  - Recovery logic should be centralized
  - Recovery logic should be generic
  - Low overhead when not needed
Device Drivers (1/2)

- Kernel-mode software component
- Two interfaces
  - Drivers export to kernel
  - Drivers import kernel interface
- Driver class
  - Defined by its interface
- Configuration request
Device Drivers (2/2)

- **Deterministic failures**
  - Sequence of configuration or I/O requests

- **Transient failures**
  - Additional inputs
  - Occur infrequently

- **Fail-stop**
  - Can be detected and stopped

- **Insidious failures**
Shadow Drivers Design (1/4)

- Passive mode
  - Normal operation
  - Monitor all communication between drivers and kernel
  - Transparent to both of them
  - Track the state of the driver as necessary for recovery
Shadow Drivers Design (2/4)

- **Active mode**
  - During recovery
  - Impersonates the failed driver
    - Intercept and respond
  - Impersonates the kernel
    - Restart failed driver
    - Intercept and respond

- **Recovery**
  - Re-establish application configurations
  - Resume pending requests
Taps
- A T-junction placed between kernel and driver
- Replicate calls – passive mode, redirect call – active mode

The communication between kernel and driver must be explicit
- Message or procedure call
Shadow Drivers Design (4/4)

- Shadow manager
  - Interfaces and controls all shadow drivers
  - Receives notification from the fault-isolation subsystem
Implementation (1/5)

- General infrastructure
  - Isolation service
  - Redirection mechanism
  - Object tracking service

- Nooks
  - Separate kernel protection domains
  - Interpose proxy procedures on all communication between kernel and driver
  - Tracks kernel objects used by drivers
Implementation (2/5)

- Adds a shadow manager to OS
  - Initial installation of shadow drivers
Implementation (3/5)

- Passive-mode monitoring
  - Tracks requests
    - Connection-oriented
      - Save the state of each active connection
    - Request-oriented
      - Logs pending commands and arguments
  - Configuration and driver parameters
  - Tracks kernel objects
  - In many cases, calls do no work
Implementation (4/5)

- Active-mode recovery
  - Stop failed driver
    - Garbage collect resources
  - Reinitializing the driver a clean state
  - Transferring relevant shadow driver state into new driver
Implementation (5/5)

- Active-mode proxying of kernel request
  - Respond with information that it has record
  - Drop the request
  - Queue the request
  - Block the request
  - Report that the driver is busy
### Hardware
- P4 3.0GHz, 1GB RAM, 80G 7200RPM IDE

### Software

<table>
<thead>
<tr>
<th>Device Driver</th>
<th>Application Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sound</strong></td>
<td>• mp3 player (<em>zinf</em>) playing 128kb/s audio</td>
</tr>
<tr>
<td><em>(audigy driver)</em></td>
<td>• audio recorder (<em>audacity</em>) recording from microphone</td>
</tr>
<tr>
<td></td>
<td>• speech synthesizer (<em>festival</em>) reading a text file</td>
</tr>
<tr>
<td></td>
<td>• strategy game (<em>Battle of Wesnoth</em>)</td>
</tr>
<tr>
<td><strong>Network</strong></td>
<td>• network send (<em>netperf</em>) over TCP/IP</td>
</tr>
<tr>
<td><em>(e1000 driver)</em></td>
<td>• network receive (<em>netperf</em>) over TCP/IP</td>
</tr>
<tr>
<td></td>
<td>• network file transfer (<em>scp</em>) of a 1GB file</td>
</tr>
<tr>
<td></td>
<td>• remote window manager (<em>vnc</em>)</td>
</tr>
<tr>
<td></td>
<td>• network analyzer (<em>ethereal</em>) sniffing packets</td>
</tr>
<tr>
<td><strong>Storage</strong></td>
<td>• compiler (<em>make/gcc</em>) compiling 788 C files</td>
</tr>
<tr>
<td><em>(ide-disk driver)</em></td>
<td>• encoder (<em>LAME</em>) converting 90 MB file .wav to .mp3</td>
</tr>
<tr>
<td></td>
<td>• database (<em>MySQL</em>) processing the <em>Wisconsin Benchmark</em></td>
</tr>
</tbody>
</table>
Evaluation (2/4)

- Performance
**Evaluation(3/4)**

- Fault-tolerance

<table>
<thead>
<tr>
<th>Device Driver</th>
<th>Application Activity</th>
<th>Application Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sound</strong> <strong>(audigy driver)</strong></td>
<td>mp3 player, audio recorder, speech synthesizer, strategy game</td>
<td>Linux-Native: CRASH, Linux-Nooks: CRASH, Linux-SD: CRASH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Linux-Native: CRASH, Linux-Nooks: MALFUNCTION, Linux-SD: ✓</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Linux-Native: CRASH, Linux-Nooks: ✓, Linux-SD: ✓</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Linux-Native: CRASH, Linux-Nooks: ✓, Linux-SD: ✓</td>
</tr>
<tr>
<td><strong>Network</strong> <strong>(e1000 driver)</strong></td>
<td>network file transfer, remote window manager, network analyzer</td>
<td>Linux-Native: CRASH, Linux-Nooks: ✓, Linux-SD: ✓</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Linux-Native: ✓, Linux-Nooks: ✓, Linux-SD: ✓</td>
</tr>
<tr>
<td><strong>IDE</strong> <strong>(ide-disk driver)</strong></td>
<td>compiler, encoder, database</td>
<td>Linux-Native: CRASH, Linux-Nooks: CRASH, Linux-SD: ✓</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Linux-Native: CRASH, Linux-Nooks: CRASH, Linux-SD: ✓</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Linux-Native: CRASH, Linux-Nooks: CRASH, Linux-SD: ✓</td>
</tr>
</tbody>
</table>
Evaluation (4/4)

- Code size

<table>
<thead>
<tr>
<th>Driver Class</th>
<th>Shadow Driver Lines of Code</th>
<th>Device Driver Shadowed Lines of Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sound</td>
<td>666</td>
<td>7,381 (audigy)</td>
</tr>
<tr>
<td>Network</td>
<td>198</td>
<td>13,577 (e1000)</td>
</tr>
<tr>
<td>Storage</td>
<td>321</td>
<td>5,358 (ide-disk)</td>
</tr>
</tbody>
</table>
Conclusion

- Mask device driver failures from both OS and applications