Dependability Benchmarking of Off-the-Shelf OS Kernels

Karama Kanoun

45th Meeting of IFIP Working Group 10.4, Moorea, French Polynesia, March 5-9, 2004

Partially financed by the European Commission
DBench

Objective of DBench

- Conceptual framework & experimental environment for benchmarking the dependability of (C)OTS and COTS-based systems
  - Concepts, specifications and guidelines for dependability benchmarking
  - Dependability benchmark prototypes

Current / final results

- A framework for dependability benchmarking
- A set of benchmark specifications and associated prototypes
  - User point of view: robustness benchmarks wrt external errors
  - Emphasis on representativeness and validation
## DBench Framework

<table>
<thead>
<tr>
<th>Categorization</th>
<th>Measures</th>
<th>Experimentation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Benchmark Target - BT</strong>  &lt;br&gt;(System nature  &lt;br&gt;Application area  &lt;br&gt;Operating environment) &lt;br&gt;Benchmarking context  &lt;br&gt;(Life-cycle phase  &lt;br&gt;Benchmark user  &lt;br&gt;Benchmark performer  &lt;br&gt;Benchmark purpose)</td>
<td><strong>Measure nature</strong>  &lt;br&gt;(qualitative or quantitative) &lt;br&gt;<strong>Measure type</strong>  &lt;br&gt;(dependability- or performance-related) &lt;br&gt;<strong>Measure extent</strong>  &lt;br&gt;(comprehensive or specific) &lt;br&gt;<strong>Assessment method</strong>  &lt;br&gt;(experimentation or modeling &amp; experimentation)</td>
<td><strong>System Under Benchmark - SUB</strong>  &lt;br&gt;Workload  &lt;br&gt;Faultload  &lt;br&gt;Measurements  &lt;br&gt;Procedures &amp; rules</td>
</tr>
</tbody>
</table>

---
Benchmark developed

- General-purpose operating systems
  - Robustness and timing measures, TPC-C Client, faulty application

- Real-Time Kernels in onboard space systems
  - Predictability of the kernel response time, faulty application

- Engine control applications in automotive systems
  - Robustness of the control application, transient hardware faults

- On-line transaction processing (OLTP) systems,
  - TPC-C-based, Operator, software & hardware faults
    - TPC-C like measures
    - Measures based on modeling & experimentation: availability, cost

- Web servers
Properties

- Representativeness
- Repeatability
- Reproducibility
- Portability
- Non-intrusiveness
- Scalability
- Cost effective
  - Set-up
  - Execution duration
OS Benchmarking

- Integrator of a system including an operating system (OS)
  - Information on OS dependability
  - Select the most appropriate OS / system characteristics
  - Publishable results

- Objectives of OS dependability benchmarking
  - Provide generic and reproducible methods
    - Characterize the OS behavior in the presence of faults
  - Compare alternative solutions
OS Benchmarking Context

- Limited knowledge about the OS
- Functional description of the OS
- Non-intrusiveness
  - Faults injected outside the OS
  - Accessibility and observability
Benchmark Target & SUB

System Under Benchmark (SUB)

Workload

API

Operating system

Device drivers

Hardware

Faultload

Benchmark Management System

DBench: Dependability Benchmarking
Benchmark Measures

- Workload level
  - Workload
  - Operating system
    - API
  - Device drivers
- OS level
  - Hardware
OS Level Measures

OS Outcomes

SEr  Error code
SXp  Exception
SPc  Panic
SHg  Hang
SNS  No signaling

Measures

- POS: OS Robustness (outcome distribution)
- Texec: OS reaction time in the presence of faults
- Tres: OS Restart time in the presence of faults
Workload Level Measures

Workload Outcomes

WCC  Correct completion
WEC  Erroneous completion
WAb  Abort
WHg  Hang
## Workload Level Measures

### Combined states

<table>
<thead>
<tr>
<th>Workload</th>
<th>OS</th>
<th>Error code</th>
<th>Exception</th>
<th>Panic</th>
<th>Hang</th>
<th>No signalling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct completion</td>
<td>SEr</td>
<td>WCC</td>
<td>SXp-WCC</td>
<td>—</td>
<td>—</td>
<td>SNS-WCC</td>
</tr>
<tr>
<td>Erroneous completion</td>
<td>SEr</td>
<td>WEC</td>
<td>SXp-WEC</td>
<td>—</td>
<td>—</td>
<td>SNS-WEC</td>
</tr>
<tr>
<td>Abort</td>
<td>SEr</td>
<td>WAb</td>
<td>SXp-WAb</td>
<td>SPc-WAb</td>
<td>—</td>
<td>SNS-WAb</td>
</tr>
<tr>
<td>Hang</td>
<td>SEr</td>
<td>WHg</td>
<td>SXp-WHg</td>
<td>SPc-WHg</td>
<td>SHg-WHg</td>
<td>SNS-WHg</td>
</tr>
</tbody>
</table>

## Workload Measures

- PSNS: WL Robustness (WL outcome distribution)
- TWL: WL completion time in the presence of faults
Measure Summary

❖ OS Measures
  ✦ POS: OS Robustness
  ✦ Texec: reaction time in the presence of fault ($\tau_{\text{exec}}$: in absence of faults)
  ✦ Tres: restart time in the presence of faults ($\tau_{\text{res}}$: in absence of faults)

❖ Workload Measures
  ✦ PSNS: WL Robustness, when OS in SNS
  ✦ TWL: WL correct completion time in the presence of faults
    ($\tau_{\text{WL}}$: in absence of faults)
Execution profile

- **Workload:** TPC-C Client in the current prototype

- **Faultload**
  - Selection of system calls to be corrupted
    - Ideally: all system calls with parameters
    - In practice: most critical OS functional components
      - Processes and Threads, File Input/output,
      - Memory management, Configuration Management
  - 28 system calls, 75 parameters, 502 corrupted values

- Interception of the selected system calls

- Parameter corruption technique: selective substitution
Parameter Corruption technique

Systematic Bit Flip

Selective substitution

Out-of-range Data

Incorrect Data

Incorrect Address
Experimental Set-up
Measurements

Experiments with Workload (WL) completion

- **Workload Completion Time**
- **OS Reaction time**
- **Restart time**

- $t_{ExpStart}$ (n)
- $t_{WStart}$ (n)
- $t_{Resume}$ (n)
- $t_{Response}$ (n)
- $t_{ExpEnd}$ (n)
- $t_{ExpStart}$ (n+1)
Measurements

Experiments without Workload (WL) completion

- **tExpStart** (n)
- **tWStart** (n)
- **tResume** (n)
- **tResponse** (n)
- **OS Reaction time**
- **System Call to intercept**
- **Timeout >> Workload completion duration**
- **tExpEnd** (n)
- **Restart time**
- **tExpStart** (n+1)
- **Experiment End**
- **WL End**
Results: OS Robustness

Pos

Windows NT
- Exception: 12%
- No signaling: 55%
- Error code: 33%

Windows 2000
- Exception: 11.4%
- No signaling: 54.5%
- Error code: 34.1%

Windows XP
- Exception: 11.4%
- No signaling: 57.4%
- Error code: 31.2%

28 system calls intercepted, 552 experiments
# Sensitivity Analysis wrt Faultload

<table>
<thead>
<tr>
<th></th>
<th>Incorrect Data</th>
<th>Incorrect Address</th>
<th>Out-of-range Data</th>
<th>Systematic Bit-Flip</th>
<th># System calls</th>
<th># Experiments</th>
</tr>
</thead>
<tbody>
<tr>
<td>FL0</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td>28</td>
<td>552</td>
</tr>
<tr>
<td>FL1</td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
<td>28</td>
<td>325</td>
</tr>
<tr>
<td>FL2</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td>28</td>
<td>113</td>
</tr>
<tr>
<td>FL3</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>28</td>
<td>2400</td>
</tr>
<tr>
<td>FL4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>All (132)</td>
<td>353</td>
</tr>
</tbody>
</table>
## Workload States

<table>
<thead>
<tr>
<th>Windows NT</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>↓WL</td>
<td></td>
</tr>
<tr>
<td>Completion</td>
<td>(451)</td>
</tr>
<tr>
<td>Abort / Hang</td>
<td>(101)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Windows 2000</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>↓WL</td>
<td></td>
</tr>
<tr>
<td>Completion</td>
<td>(445)</td>
</tr>
<tr>
<td>Abort / Hang</td>
<td>(107)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Windows XP</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>↓WL</td>
<td></td>
</tr>
<tr>
<td>Completion</td>
<td>(424)</td>
</tr>
<tr>
<td>Abort / Hang</td>
<td>(128)</td>
</tr>
</tbody>
</table>
## Refinement of Workload States

<table>
<thead>
<tr>
<th>OS</th>
<th>WL</th>
<th>Error Code (count)</th>
<th>Exception (count)</th>
<th>No Signaling (count)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Windows NT</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Completion</td>
<td>(451)</td>
<td>136</td>
<td>58</td>
<td>257</td>
</tr>
<tr>
<td>Abort / Hang</td>
<td>(101)</td>
<td>46</td>
<td>8</td>
<td>47</td>
</tr>
<tr>
<td><strong>Windows 2000</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Completion</td>
<td>(445)</td>
<td>136</td>
<td>57</td>
<td>252</td>
</tr>
<tr>
<td>Abort / Hang</td>
<td>(107)</td>
<td>52</td>
<td>6</td>
<td>49</td>
</tr>
<tr>
<td><strong>Windows XP</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Completion</td>
<td>(424)</td>
<td>99</td>
<td>57</td>
<td>268</td>
</tr>
<tr>
<td>Abort / Hang</td>
<td>(128)</td>
<td>73</td>
<td>6</td>
<td>49</td>
</tr>
</tbody>
</table>
## OS Reaction Time

<table>
<thead>
<tr>
<th>OS</th>
<th>$\tau_{exec}$</th>
<th>Texec (Std dev.)</th>
<th>Texec Error code</th>
<th>Texec Exception</th>
<th>Texec No-signaling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows NT</td>
<td>344 $\mu$s</td>
<td>128 $\mu$s (230 $\mu$s)</td>
<td>17 $\mu$s (18 $\mu$s)</td>
<td>86 $\mu$s (138 $\mu$s)</td>
<td>203 $\mu$s (281)</td>
</tr>
<tr>
<td>Windows 2000</td>
<td>1782 $\mu$s</td>
<td>1241 $\mu$s (3359 $\mu$s)</td>
<td>22 $\mu$s (28 $\mu$s)</td>
<td>973 $\mu$s (2978 $\mu$s)</td>
<td>2013 $\mu$s (4147)</td>
</tr>
<tr>
<td>Windows XP</td>
<td>111 $\mu$s</td>
<td>114 $\mu$s (176 $\mu$s)</td>
<td>23 $\mu$s (17 $\mu$s)</td>
<td>108 $\mu$s (162 $\mu$s)</td>
<td>165 $\mu$s (204 $\mu$s)</td>
</tr>
</tbody>
</table>
Detailed OS Reaction Time

Error Code return

System Call

Windows NT4
Windows 2000
Windows XP

μs

DBench
Detailed OS Reaction Time

Exception Notification

\[ \mu s \]

<table>
<thead>
<tr>
<th>System Call</th>
<th>NT</th>
<th>2000</th>
<th>XP</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateRemoteThread</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetEnvironmentVariableW</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetExitCodeThread</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetPrivateProfileStringA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetStartupInfoA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GlobalFree</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LocalFree</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ReadFile</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WriteFile</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Windows NT4
Windows 2000
Windows XP
Detailed OS Reaction Time

No-Signaling

<table>
<thead>
<tr>
<th>System call</th>
<th>NT</th>
<th>2000</th>
<th>XP</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateRemoteThread</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CreateThread</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DuplicateHandle</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetExitCodeThread</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetPrivateProfileStringA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetProcessVersion</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GlobalAlloc</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GlobalFree</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GlobalLock</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IsBadReadPtr</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IsBadWritePtr</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LocalAlloc</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LocalFree</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ReadFile</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetThreadPriority</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VirtualAllocEx</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WriteFile</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Windows NT4
- Windows 2000
- Windows XP

DBench
Dependability Benchmarking
### OS Restart Time

<table>
<thead>
<tr>
<th>OS</th>
<th>τres</th>
<th>Tres</th>
<th>Std Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows NT</td>
<td>92 s</td>
<td>96 s</td>
<td>4 s</td>
</tr>
<tr>
<td>Windows 2000</td>
<td>105 s</td>
<td>109 s</td>
<td>8 s</td>
</tr>
<tr>
<td>Windows XP</td>
<td>74 s</td>
<td>80 s</td>
<td>8 s</td>
</tr>
</tbody>
</table>
Detailed OS Restart Time

Windows NT

Windows 2000

Windows XP

WL Abort or Hang
## WL Execution Time

<table>
<thead>
<tr>
<th></th>
<th>$\tau_{WC}$</th>
<th>TWC</th>
<th>Std Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows NT</td>
<td>74 s</td>
<td>80 s</td>
<td>12 s</td>
</tr>
<tr>
<td>Windows 2000</td>
<td>70 s</td>
<td>74 s</td>
<td>13 s</td>
</tr>
<tr>
<td>Windows XP</td>
<td>67 s</td>
<td>69 s</td>
<td>10 s</td>
</tr>
</tbody>
</table>
Conclusion

- OS robustness benchmark wrt application erroneous behavior
- Dependability benchmark prototype for Windows family
- Novelty
  - Structured set of measures
  - Realistic Workload: TPC-C Client
  - Standard experimental procedures and rules
  - Benchmark properties
  - Benchmark execution duration: 2 days
Validation of the benchmark

- Results in conformance with Microsoft claim
- Sensitivity study wrt to parameter corruption technique
- Sensitivity study wrt system calls corrupted
- Benchmark properties

Current work

- Other OS family: Linux
- Other workload