Dependability Benchmarking of Operating Systems

Karama KANOUN
LAAS-CNRS
(karama.kanoun@laas.fr)

PhD seminars - University of Roma, Italy
September 13 - 17, 2004

Benchmarking Context

Selection Criteria: cost, performance, Dependability
**Benchmarking Context**

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

- Limited knowledge about the OS
  - Functional description of the OS
  - Accessibility and observability

**Benchmark Components**

- System Under Benchmark (SUB)
  - Workload
  - API
  - Operating system
  - Device drivers
  - Hardware

- Faultload
- Benchmark controller
Benchmark Measures

Workload level

OS level

OS Outcomes
- SEr: Error code
- SXp: Exception
- SPc: Panic
- SHg: Hang
- SNS: No signaling

Basic Measures
- POS: OS Robustness (outcome distribution)
- Texec: OS reaction time in the presence of faults
  ($\tau_{exec}$: in absence of fault)
- Tres: OS Restart time in the presence of faults
  ($\tau_{res}$: in absence of fault)
**Execution Profile**

- **Workload**
  - TPC-C Client

- **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, 552 corrupted values
    - Interception of the selected system calls
    - Parameter corruption technique: selective substitution

**Faultload**

Parameter corruption technique

- Systematic Bit Flip
- Selective substitution

- Out-of-range Data
- Incorrect Data
- Incorrect Address

Software Reliability Evaluation (Dependability Benchmarking) - Karama KANOUN - LAAS-CNRS
Experimental Set-up

Experiments with Workload completion

Measurements
Measurements

Experiments without Workload completion

Timeout >> Workload completion duration

OS Reaction time

System Call to intercept

Experiment End

Restart time

Experiment Start
(n)

Workload End

Experiment Start
(n+1)

OS Robustness

POS

28 system calls intercepted, 552 experiments
**OS Reaction Time**

<table>
<thead>
<tr>
<th></th>
<th>$\tau_{\text{exec}}$</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>

**OS Restart Time**

<table>
<thead>
<tr>
<th></th>
<th>$\tau_{\text{res}}$</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>
Benchmark Validation

- Results in conformance with Microsoft claims
- Benchmark properties
  - Repeatability
  - Reproducibility
  - Portability
  - Cost effectiveness
- Sensitivity analysis wrt to 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>28</td>
<td>552</td>
<td></td>
</tr>
<tr>
<td>FL1</td>
<td>x</td>
<td>x</td>
<td></td>
<td>28</td>
<td>325</td>
<td></td>
</tr>
<tr>
<td>FL2</td>
<td></td>
<td>x</td>
<td></td>
<td>28</td>
<td>113</td>
<td></td>
</tr>
<tr>
<td>FL3</td>
<td></td>
<td></td>
<td></td>
<td>All (132)</td>
<td>468</td>
<td></td>
</tr>
<tr>
<td>FL4</td>
<td></td>
<td></td>
<td></td>
<td>28</td>
<td>2400</td>
<td></td>
</tr>
</tbody>
</table>

Sensitivity Analyses wrt Faultload

Software Reliability Evaluation (Dependability Benchmarking) - Karama KANOUN - LAAS-CNRS
Conclusions

- OS robustness benchmark wrt application erroneous behavior
- Dependability benchmark prototype for Windows family
- Characteristics
  - Structured set of measures
  - Realistic Workload: TPC-C Client
  - Standard experimental procedures and rules
  - Validation of the benchmark properties
- Lesson learned
  - Workload state difficult to identify → workload selection

Future Work

- Other workloads: Postmark, Java Virtual Machine
- Other OS family: Linux (Postmark, Java Virtual Machine)
- Other measure: error propagation
- Other faultload:
  - Hardware faults
  - Erroneous behaviour of device drivers