DBench Project
(Dependability Benchmarking)

European IST Program IST-2000-25425

Karama Kanoun

39th Meeting IFIP Working Group 10.4 — Paraty, Brazil — 28 February- 3 March, 2001
Consortium

**Partners**

- Chalmers University of Technology (S)
- Critical Software (P)
- Faculdade de Ciencias e Technologia da Universidade de Coimbra (P)
- Friedrich-Alexander Universität, Erlangen-Nürnberg (D)
- LAAS-CNRS (F)
- Universidad Politechnica de Valencia (E)

**Sponsor**

- Microsoft (UK)

**Advisory Board**

- Astrium (F), CMU (USA), INDRA (E), Oracle (P), Saab Ericsson Space (S), Thales (F)
DBench Objectives

Conceptual framework & Experimental environment

→ benchmarking the dependability of COTS and COTS-based systems

Provide means for:

✦ Characterising & evaluating the dependability of a component / system
✦ Identifying malfunctioning or weakness
✦ Comparing the dependability of alternative solutions

Final outputs

✦ Concepts, specifications and guidelines for dependability benchmarking
✦ Set of dependability benchmark prototype tools
Steps

Preliminary definition of the conceptual framework for system dependability benchmarking
  → Relevant issues

Identification and evaluation of enabling technologies
  → To put into practice the conceptual framework

Experimentation and validation
  → Pilot experiments on various target systems

Consolidation
  → Recommendations
Conceptual Framework

**Aim:** characterize the target system behavior in presence of workload + faultload

Definition of meaningful benchmark measures, based on:

→ measurements on the target system

→ measurements + modeling

**Benchmark objectives & utilization**

Benchmark users: system end-users & system providers

Assess dependability, identify weakness, tune component/architecture, compare

**Benchmark properties**

Portability, adaptability, non-interference, repeatability, reproducibility, …

**Guidelines for conducting a dependability benchmark**

Unified bases to carry out experiments & interpret results
Enabling technologies

**Aim:** adapt, extend fault injection techniques

Relevant measurements ← meaningful measures

What, where and when?

Fault representativeness

Distinct causes ⇒ similar error patterns ⇒

Validate the nature of erroneous behavior

Workload & Faultload

Selection

Synchronization?
Target Systems

- Transactional Applications (ORACLE)
- General purpose OS
- Linux
- Windows 2000
Target Systems

Transaction Applications (ORACLE)

General purpose OS

Linux
Windows 2000

Embedded Applications (on-board)

OS for embedded applications

Linux EB
Windows EB
<table>
<thead>
<tr>
<th>Linux</th>
<th>Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>General purpose for embedded applications</td>
<td>General purpose for embedded applications</td>
</tr>
<tr>
<td>ORACLE on top of Linux</td>
<td>ORACLE on top of Windows</td>
</tr>
<tr>
<td>On-board guidance and control appli.</td>
<td></td>
</tr>
</tbody>
</table>
Cross exploitation of results

Linux

- General purpose
- for embedded applications
- ORACLE on top of Linux

Windows

- General purpose
- for embedded applications
- ORACLE on top of Windows

On-board guidance & control appli.
Cross exploitation of results

**Linux**
- General purpose
- for embedded applications
- ORACLE on top of Linux

**Windows**
- General purpose
- for embedded applications
- ORACLE on top of Windows

On-board guidance & control appli.
Cross exploitation of results

Linux

General purpose

for embedded applications

ORACLE on top of Linux

On-board guidance & control appli.

Windows

General purpose

for embedded applications

ORACLE on top of Windows
Practical Outcomes

A set of dependability measures (meaningful to system end-users and system providers)

A strategy for characterizing & quantifying system dependability based on modeling and experimentation, depending on the target system nature

Methods for system dependability measurements: system solicitation, observation, analysis and processing

Framework for dependability benchmarking according to various dimensions

Recommendations and guidelines for system dependability benchmarking (methods + supporting tools)
<table>
<thead>
<tr>
<th>Workpackages</th>
<th>Year 1</th>
<th>Year 2</th>
<th>Year 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>WP1 Conceptual Framework</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WP2 Enabling Technologies Identification &amp; Evaluation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WP3 Benchmark Definition, Experimentation &amp; Validation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WP4 Consolidation</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Workpackages & Tasks

<table>
<thead>
<tr>
<th>WP1</th>
<th>Conceptual Framework (CF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WP2</td>
<td>Enabling Technologies Identification and Evaluation (ETIE)</td>
</tr>
<tr>
<td>T21</td>
<td>Measurements</td>
</tr>
<tr>
<td>T22</td>
<td>Fault Representativeness</td>
</tr>
<tr>
<td>T23</td>
<td>Workload and Faultload Selection</td>
</tr>
<tr>
<td>WP3</td>
<td>Benchmark Definition, Experimentation and Validation (BDEV)</td>
</tr>
<tr>
<td>T31</td>
<td>Benchmark Definition</td>
</tr>
<tr>
<td>T32</td>
<td>Benchmark Experimentation</td>
</tr>
<tr>
<td>T33</td>
<td>Benchmark Validation</td>
</tr>
<tr>
<td>WP4</td>
<td>Consolidation (CD)</td>
</tr>
</tbody>
</table>