



DBench

Dependability Benchmarking

IST-2000-25425

DBench Dependability Benchmarks

Report Version: Deliverables BDEV3, BDEV4 and CD1

Report Preparation Date: 27 February 2004 — Revision: 27 May 2004

Classification: Public Circulation

Contract Start Date: 1 January 2001

Duration: 36m

Project Co-ordinator: LAAS-CNRS (France)

Partners: Critical Software (Portugal)

University of Coimbra (Portugal)

Friedrich Alexander University, Erlangen-Nürnberg (Germany)

LAAS-CNRS (France)

Polytechnic University of Valencia (Spain)



**Project funded by the European Community
under the “Information Society Technology”
Programme (1998-2002)**

Editors

Karama Kanoun (LAAS-CNRS)
Henrique Madeira (University of Coimbra)
Yves Crouzet (LAAS-CNRS)
Mario Dal Cin (Friedrich Alexander University)
Francisco Moreira (Critical Software)
Juan-Carlos Ruiz García (Polytechnic University of Valencia)

Contributors

Jean Arlat (LAAS-CNRS)
Kerstin Buchacker (Friedrich Alexander University)
Diamantino Costa (Critical Software)
Yves Crouzet (LAAS-CNRS)
Mario Dal Cin (Friedrich Alexander University)
João Durães (University of Coimbra)
Pedro Gil (Polytechnic University of Valencia)
Francisco González (Polytechnic University of Valencia)
Hans-Jörg Höxer (Friedrich Alexander University)
Tahar Jarboui (LAAS-CNRS)
Ali Kalakech (LAAS-CNRS)
Karama Kanoun (LAAS-CNRS)
Szymon Kolelis (Critical Software)
Rui Leitão (Critical Software)
Lenin Lemus (Polytechnic University of Valencia)
Henrique Madeira (University of Coimbra)
Francisco Moreira (Critical Software)
Manuel Rodríguez (Critical Software)
Juan-Carlos Ruiz García (Polytechnic University of Valencia)
Volkmar Sieh (Friedrich Alexander University)
Oliver Tschäche (Friedrich Alexander University)
Marco Vieira (University of Coimbra)
Martin Waitz (Friedrich Alexander University)
Pedro Yuste (Polytechnic University of Valencia)

Contents

Foreword

Chapter 1

Dependability Benchmarking Concepts

1.1 Introduction.....	1-2
1.2 Related Work.....	1-3
1.3 Benchmark Specifications	1-5
1.3.1 Dependability Benchmarking Dimensions.....	1-5
1.3.1.1 Categorization Dimensions	1-6
1.3.1.2 Measure Dimensions	1-7
1.3.1.3 Experimentation Dimensions	1-8
1.3.2 Benchmark Conduct	1-12
1.3.2.1 Benchmarking Scenarios.....	1-12
1.3.2.2 Benchmark Management System	1-13
1.3.2.3 Procedures and Rules	1-13
1.4 Benchmark Validation.....	1-14
1.4.1 Representativeness	1-15
1.4.2 Repeatability and reproducibility	1-16
1.4.3 Portability	1-16
1.4.4 Non-intrusiveness.....	1-16
1.4.5 Scalability	1-17
1.4.6 Benchmarking Time and Cost	1-17
1.5 On-Line Analytical Processing (OLAP) of Experiment Results.....	1-18
1.6 Benchmarks developed within DBench	1-19
1.6.1 General purpose Operating Systems	1-20
1.6.2 Real Time Kernels in Onboard Space Systems	1-20
1.6.3 Engine Control Applications in Automotive Systems.....	1-21
1.6.4 On Line Transaction Processing Systems	1-21

Chapter 2

Dependability Benchmark for General Purpose Operating Systems

2.1. Introduction.....	2-2
2.2. Dependability Benchmark Overview	2-3
2.3. Benchmark Measures.....	2-4
2.3.1. OS Measures.....	2-4

2.3.2. Workload Measures and Comprehensive Combined Measures.....	2-6
2.3.3. Summary of Measures.....	2-7
2.3.4. Basic and Complementary Measures.....	2-8
2.4. Experimental Dimensions.....	2-9
2.4.1. Parameter Corruption Technique.....	2-9
2.4.2. System Calls to Be Corrupted.....	2-10
2.4.3. Measurements.....	2-10
2.5. Benchmark Prototype.....	2-11
2.5.1. Systems Under Benchmark.....	2-11
2.5.2. Execution Profile.....	2-12
2.5.3. Experimental Set-up and Benchmark Conduct.....	2-12
2.6. Results.....	2-14
2.6.1 OS Robustness.....	2-14
2.6.2 OS Reaction Time.....	2-15
2.6.3 System Restart Time.....	2-15
2.6.4 Result Summary.....	2-16
2.6.5. Time Needed to Develop and to Run the Benchmark.....	2-16
2.7. Benchmark Validation.....	2-17
2.7.1. Representativeness.....	2-17
2.7.2. Repeatability and Reproducibility.....	2-18
2.7.3. Portability.....	2-19
2.7.4. Non-intrusiveness.....	2-19
2.7.5. Scalability.....	2-19
2.8. Conclusion.....	2-19
Annex 2-A Faultload Validation.....	2-21
2-A.1 Impact of Parameter Corruption Technique.....	2-21
2-A.2 Bit-Flip Technique and Selective Substitution Technique.....	2-22
2-A.3 Impact of System Calls Considered.....	2-23
Annex 2-B Benchmark Measure Refinement.....	2-25
2-B.1 Robustness.....	2-25
2-B.2 OS Reaction Time.....	2-26
2-B.3 System Restart Time.....	2-28
2-B.4 Workload execution time.....	2-29

Chapter 3

Dependability Benchmark for Real Time Kernels in Onboard Space Systems

3.1 Introduction.....	3-2
3.2 Basics on Space systems.....	3-2
3.2.1 Components of Space systems.....	3-3
3.3 Benchmark Specification.....	3-4
3.3.1 Benchmark Overview.....	3-4
3.3.2 Dependability measures.....	3-5
3.3.3 System Under Benchmarking.....	3-6
3.3.4 Experimental dimensions.....	3-7

3.4 Benchmark Prototype	3-15
3.4.1 Benchmark Configuration	3-15
3.4.2 Workload	3-18
3.4.3 Faultload	3-21
3.4.4 Experiments and results	3-21
3.5 Benchmark Validation	3-22
3.5.1 Representativeness	3-23
3.5.2 Reproducibility and Repeatability	3-23
3.5.3 Portability	3-23
3.5.4 Non-intrusiveness	3-23
3.5.5 Scalability	3-24
3.6 Conclusions	3-24
Annex 3-A Detailed Measures definition	3-26
3-A.1 Notation	3-26
3-A.2 Special cases	3-26
3-A.3 Measures for individual system calls	3-26
3-A.4 Benchmark measures	3-28
Annex 3-B Detailed results obtained	3-30
3-B.1 Timer Manager	3-31
3-B.2 Task Manager	3-31
3-B.3 Partition Manager	3-31
3-B.4 Message Manager	3-32
3-B.5 Semaphore Manager	3-32
3-B.6 Clock Manager	3-32
3.7 References	3-33
3.7.1 Applicable Documents	3-33
3.7.2 Reference Documents	3-33
3.8 Definitions and acronyms	3-35

Chapter 4

Dependability Benchmark for Engine Control Applications in Automotive Systems

4.1. Introduction	4-2
4.2. Basics on Automotive Engine Control Systems	4-3
4.2.1. How Engines Work	4-3
4.2.2. Engine Control System Model	4-5
4.3. Benchmark Specification	4-6
4.3.1. Benchmark Overview	4-6
4.3.2. Dependability Measures	4-7
4.3.3. Execution Profile	4-9
4.3.4. Benchmark Procedure	4-12
4.4. Benchmark Prototype	4-18
4.4.1. Specification of the case study DBT	4-18
4.4.2. Prototype description	4-19

4.4.3. Benchmark experiments	4-23
4.4.4. Implementation Cost and Effort	4-25
4.5. Benchmark Properties and Their Validation	4-27
4.5.1. Portability	4-27
4.5.2. Non-intrusiveness.....	4-27
4.5.3. Scalability	4-28
4.5.4. Repeatability of the experiments.....	4-28
4.5.5. Representativeness	4-29
4.6. Conclusions	4-29

Chapter 5

Dependability Benchmark for OLTP Environments

5.1. Introduction.....	5-2
5.2. Benchmark Specification.....	5-3
5.2.1. Experimental Set-up.....	5-4
5.2.2. Benchmark Procedure	5-5
5.2.3. Measures	5-6
5.2.4. Workload.....	5-8
5.2.5. Faultload.....	5-8
5.3. Benchmark Validation.....	5-10
5.3.1. Systems Used in the Benchmark Validation Experiments.....	5-12
5.3.2. Faultload Based on Operator Faults.....	5-13
5.3.3. Faultload Based on Software Faults	5-15
5.3.4. Faultload Based on High-level Hardware Failures.....	5-18
5.4. Benchmark Prototype	5-20
5.4.1. Systems Under Benchmarking	5-20
5.4.2. Different Operating Systems and DBMS	5-21
5.4.3. Different DBMS Configurations.....	5-23
5.4.4. Different Hardware Platforms	5-23
5.4.5. Results Summary.....	5-24
5.4.6. Benchmark Execution Effort.....	5-25
5.5. Conclusion.....	5-26
Annex 5-A DBench-OLTP Dependability Benchmark Specification	5-28
Clause 0: PREAMBLE.....	5-28
Clause 1: BENCHMARK SETUP	5-29
Clause 2: BENCHMARKING PROCEDURE	5-31
Clause 3: MEASURES	5-34
Clause 4: FAULTLOAD.....	5-36
Clause 5: FULL DISCLOSURE.....	5-40

Chapter 6**TPC-C based Dependability Benchmark Focusing on Hardware Faults**

6.1. Introduction.....	6-2
6.2. Building Dependability Benchmarks for Transactional Systems.....	6-2
6.2.1. Benchmarking Approach	6-3
6.2.2. Dependability Measures	6-5
6.2.3. Cost Measure	6-9
6.2.4. Summary – Full Disclosure Report	6-10
6.3. Specification of TPC-C-Depend	6-11
6.3.1. Introduction.....	6-11
6.3.2. System Under Benchmarking	6-12
6.3.3. Workload	6-14
6.3.4. Failure Modes	6-15
6.3.5. Measures and Measurements	6-18
6.3.6. Benchmark Conduct, Procedures and Rules	6-19
6.3.7. Formal Full Disclosure Reports to Ensure Reproducibility.....	6-19
6.4. Implementation Example.....	6-21
6.4.1. Experiment Set-up.....	6-21
6.4.2. Faultload.....	6-25
6.4.3. Results.....	6-26
6.4.4. Automation.....	6-28
6.4.5. Effort Needed.....	6-29
6.4.6. Benchmark Execution Duration.....	6-30
6.5. Benchmark Validation.....	6-30
6.5.1. Representativeness	6-30
6.5.2. Portability	6-31
6.5.3. Repeatability and Reproducibility	6-31
6.5.4. Scalability	6-32
6.5.5. Non-Intrusiveness.....	6-32
Annex 6-A Full Disclosure Reports.....	6-33
6-A.1 Describing the Static Set-up	6-33
6-A.2 Describing the Dynamic Set-up	6-37

Chapter 7**Cross Exploitation of Results,****Lessons Learned and Recommendations for Future Work**

7.1. Introduction.....	7-2
7.2. Cross exploitation of results.....	7-2
7.2.1. Benchmarks for Operating Systems.....	7-3
7.2.2. Benchmarks for Embedded Systems.....	7-3
7.2.3. Benchmarks for Transactional Systems	7-5
7.3. Lessons learned	7-5
7.3.1. General Lessons learned.....	7-6
7.3.2. Lessons learned for Operating Systems	7-8

7.3.3. Lessons learned for Embedded Systems 7-9
7.3.4. Lessons learned for Transactional Systems 7-9
7.4. Recommendations for future work 7-10
7.4.1. Recommendations for Operating Systems..... 7-10
7.4.2. Recommendations for Embedded Systems..... 7-11
7.4.3. Recommendations for Transactional Systems 7-12

References

Foreword

This deliverable gathers the three following deliverables described in Annex 1 of the contract:

- BDEV3: Benchmark experiments and cross exploitation of results
- BDEV4: Benchmark validation
- CD1: Benchmark concepts, specifications and guidelines

The project scientific officer and the project experts agreed on our proposal to replace the three deliverables by a single one containing all the information that should be included in the three above deliverables, but organized in a different manner. The new deliverable is structured in a way that all the information related to a given class of systems (operating systems, embedded systems or transactional systems) is grouped in the same chapter. It also includes an introductory chapter giving an overview of the DBench concepts and framework, as well as a concluding chapter.

The deliverable is thus composed of seven chapters. Chapter 1 presents the dependability benchmarking concepts defined in DBench. Chapters 2 and 3 present two benchmarks dedicated respectively to general-purpose operating systems, and to real time kernels for onboard space systems. Chapter 4 concentrates on a benchmark for engine control applications in automotive embedded systems. Chapters 5 and 6 are dedicated to two examples of benchmarks, for transactional systems, based on the performance benchmark TPC-C for on-line transaction systems. The first one provides results based on experimentation only, while the second one provides results based on experimentation and modelling. Chapter 7 concludes the deliverable, putting emphasis on lessons learned and recommendations for future work.

Chapter 2 to 6 are written in such a way that each of them can be read after reading Chapter 1 without reading the other chapters. Our aim is to make easy the understanding for readers interested only on one category of benchmarks (for embedded systems for example) and at the same time provide a full range of benchmarks addressing various categories of systems in various manners, when possible. Needless to say that reading all the chapters provides a thorough idea about the varieties of dependability benchmarks that could be developed.

