DBench

Dependability Benchmarking

IST-2000-25425

DBench Project Final Report

Report Version: V1

Report Preparation Date: 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)

Polytechnical University of Valencia (Spain)

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

Table of Contents

1. Introduction

The goal of benchmarking the dependability of computer systems is to provide generic and reproducible ways for characterizing their behaviour in the presence of faults. The key aspect that distinguishes benchmarking from existing evaluation and validation techniques is that a benchmark represents an agreement that is widely accepted both by the computer industry and/or by the user community. This technical agreement should state the system that is benchmarked, the measures, the way and conditions under which these measures are obtained, and their domain of validity. The objective of such a benchmark is to provide practical ways to characterize the dependability of computers.

The success of well-established performance benchmarks in comparing performance of components and computer systems probably accounts for the generalized idea that the main goal of benchmarks is to compare systems on the basis of benchmark results.

DBench has developed a framework for defining dependability benchmarks for computer systems, with emphasis on *Off-the-Shelf component* (OTS), commercial or not, and also OTS-based systems, via experimentation and modelling. The ultimate objective of our work is to provide a framework and guidelines for defining dependability benchmarks for computer systems, and provide means for implementing them.

From a practical point of view, a dependability benchmark is a specification of a procedure to assess measures related to the behaviour of a computer system or computer component in the presence of faults. Obviously, the benchmark specification may include source code samples or even tools to facilitate the benchmark implementation. However, what is relevant is that one must be able to implement the dependability benchmark from that specification (i.e., perform all the steps required to obtain the measures for a given system or component under benchmarking). In other words, the specification should be unambiguous and clear enough to allow:

- Implementation of the specification in order to benchmark the dependability of a given target system or component.
- Full understanding and interpretation of the benchmark results.

In DBench we have identified the main dimensions that are decisive for defining dependability benchmarks and the way experimentation can be conducted in practice. These dimensions describe: i) the target system and the benchmarking context, ii) the measures to be evaluated, as well as iii) the experimental conditions.

The *DBench framework* defines not only the above dimensions but also the different guidelines that should be followed in order to develop useful benchmarks. These guidelines should include procedures and rules for i) implementing the specifications, ii) performing the experiments to ensure uniform conditions for measurement and iii) exploiting results.

In addition, to be *meaningful* under *economically acceptable conditions*, a dependability benchmark should satisfy a set of properties. For example, a benchmark must be repeatable (in statistical terms), representative, portable, cost effective, etc. These properties represent goals that must be achieved when defining dependability benchmarks. The relevance of the benchmark properties is quite clear, as they take into account all the relevant problems that must be solved to define and validate actual dependability benchmarks. These properties should be taken into consideration from the earliest phases of the benchmark definition as they have a deep impact on the experimental dimensions and, consequently, on the benchmark specification. Also, these properties should be checked after specifying a benchmark, which can be accomplish (in our proposal) through the implementation of that benchmark in a set of representative systems for a given application domain. These properties are explicitly addressed in DBench.

In order to make the analysis and sharing of dependability benchmark results possible, we have proposed an approach, based on multidimensional analysis and data warehousing, and On-Line Analytical Processing (OLAP) technology, to solve the problem of analysing, sharing, and cross-exploiting results from dependability benchmarking experiments. OLAP allows the analysis of raw data of single experiments, analysis and comparison of benchmark results obtained in different systems, and sharing of results among project partners.

To exemplify how the benchmarking issues can actually be handled in different application domains, five examples of benchmarks and their associated prototypes (i.e., actual implementations of the benchmarks) have been developed in DBench. They concern general-purpose operating systems, embedded and transactional systems. It is expected that these benchmarks and the results obtained will help understanding the various concepts developed in DBench, at least for the considered classes of systems.

More details about the benchmarks developed are given in Section 2. Section 3 gives the list of papers published by the DBench partners throughout the project duration as well as papers and presentation related to DBench. Section 4 lists the events and fairs attended by the project partners throughout the project duration.

2. Benchmarks Developed within DBench

Table 1 summarizes the benchmark and the associated prototypes developed within DBench. The benchmark concepts and the prototypes have been validated

through the benchmarking of specific systems. The results illustrate concretely the kind of results that could be obtained from the prototypes.

Table 1: Benchmarks and prototypes developed within DBench

Application area	**Target system**	**Type of measures** (in presence of faultload)	**Workload**	**Faultload**
General purpose	Operating system	▪ OS robustness ▪ OS reaction time ▪ OS restart time	TPC-C Client	▪ Application erroneous behaviour (erroneous parameters in selected system calls)
Onboard space control	Real time kernel	▪ Predictability of response time	Onboard telecommand scheduling	▪ Erroneous parameters in selected system calls
Automotive control system	Embedded control application	▪ Dependability (safety-related) measures	Driving cycles defined in the directive ECC-90/C81/01-1999	▪ Stressful workload ▪ Memory single bit-flips
On-Line Transaction Processing systems (OLTP)	Transactional system	▪ Transaction throughput ▪ Experience Availability ▪ DBMS-level specific measures ▪ DBMS-level specific dependability measures	TPC-C based workload	▪ Scripts simulating real operator faults ▪ Low-level educated mutations ▪ Operating system simulated faults ▪ Hardware faults emulated by scripts
On-Line Transaction Processing systems (OLTP)	Transactional system	• Failure modes • Steady state availability • Cost of failures	TPC-C based workload (simulated environment)	▪ Mixed level hardware simulation ▪ VHDL controlled fault injection ▪ Component level hardware faults: - Disk fault - Network outages - Power failure

Although the benchmark targets are (purposely) quite distinct, the benchmark prototypes we are considering share some common aspects among the characterization dimensions:

- Benchmark Target: In all cases the benchmark target is always either an *Off-the-Shelf component*, either commercial (COTS) or Open Software System or a *system including at least one such component*.

- Life cycle phase: It is assumed that the benchmark is being performed during the *integration phase* of a system including the COTS benchmark target or when the system is available for *operational phase.*

- Benchmark user: The primary users are the *integrators of the system including the benchmark target* or the *end-users* of the benchmark target, as it is assumed that the benchmark results are to be standardized so that they can be made publicly available. However, some results may be of interest to the developer(s) of the BT component, for improving its dependability, should the benchmark reveals some deficiencies.

- Benchmark purpose: For all target systems, the following possible purposes are identified: i) assess some dependability features, ii) assess dependability (and performance) related measures and iii) compare alternative systems.

- Benchmark performer: We consider that the benchmark performer is someone (or an entity) who has no in depth knowledge about the benchmark target and who is aiming at i) improving significantly her/his knowledge about its dependability features, and ii) publicizing information on the BT dependability in a standardized way.

In the rest of this section, we briefly describe the measures and the main experimentation dimensions for each benchmark prototype.

2.1 General Purpose Operating Systems

The benchmark developed in DBench for general purpose OSs addresses mainly the robustness of the OS (and more precisely its kernel) with respect to faulty applications. The measures evaluated are: i) the distribution of the OS outcomes following activation of faulty system calls, ii) reaction time of the OS for faulty system calls and iii) system restart time after activation of faulty system calls. These basic measures are complemented by additional measures intended to refine them.

The workload we have considered is a realistic workload, TPC-C client. The workload and the faultload are implemented separately. A subset of system calls used by the workload is selected a priori (according to OS function criticality). When a system call belonging to this subset is invoked it is intercepted and substituted by the same system call with a corrupted parameter value. Three different parameter corruption techniques are used and their effects are compared.

The prototype developed is used to compare the dependability of three operating systems: Windows NT4, Windows 2000 and Windows XP.

2.2 Real Time Kernels in Onboard Space Systems

In real-time systems, correctness of operation depends not only on the right results being generated but also on the results being produced within time constraints. With the increase use of COTS Real-Time Kernels (RTK) in embedded systems the need

for assuring a high-level of dependability level of in such kernels also arose. Among several dependability attributes, the determinism of the response time of RTK services, even in presence of faults, is of paramount importance for hard real-time systems. This is particularly true for onboard space systems that are more exposed to external disturbances such as radiation.

DBench-RTK is a benchmark for assessing the predictability of response time of a Real-Time Kernel (RTK) service calls. This benchmark aims to allow integrators/developers to assess and compare the determinism of response time of the service calls of RTKs. The benchmark is targeted at space domain systems and addresses mainly the robustness of a RTK interface with respect to faulty applications providing wrong parameters to it. The measurements collected are combined into one single metric: predictability of response time.

2.3 Engine Control Applications in Automotive Systems

The core of modern vehicle engines is managed by the control algorithms running inside Electronic Control Units (ECUs). Due to the high scales of integration used in these electronic components, engine control systems are subject to a number of transient faults that may impact their hardware and lead their software to the production of unsafe outputs for the vehicle engine. In this context, DBench has developed a benchmark specification that addresses the robustness of the control applications running inside the ECUs with respect to transient hardware faults.

The proposed workload is inspired by the standards currently used in Europe for the emission certification of light duty vehicles. On the other hand, the faultload is defined in terms of hardware faults that affect the cells of the memory allocating the engine software control. The high scale of integration used in most modern engine ECUs induces many controllability and observability problems that have a deep impact over the definition of a suitable benchmark procedure. In order to overcome these problems, the benchmark exploits the tracing and on-the-fly memory access features existing in the debugging interfaces of current automotive embedded microprocessors.

The benchmark prototype has been specialized to the case of diesel engine control units. This prototype shows the feasibility of the approach and the various steps in which the benchmark procedure can be divided in practice. The prototype is also used as a support for the experiments conducted for validation purposes.

2.4 On Line Transaction Processing Systems (OLTP)

Large transactional systems are usually at the very centre of the IT infrastructure of companies. Even short downtimes of such systems are very expensive. To be able to evaluate the dependability of transactional systems is, therefore, of great importance. We have developed two complementary benchmarks for OLTP systems, respectively DBench-OLTP and TPC-C-Depend. Both benchmarks are extensions of TPC-C and closely follow the form and structure of the latter and use TPC-C

workload. Both benchmarks are used to characterize the Data Base Management System (DBMS) and to compare DBMSs.

The measures of **DBench-OLTP** include the TPC-C measures in the presence of faults (i.e., the number of transactions executed per minute and the price per transactions in the presence of faults), system availability during the benchmarking (for both the server and the clients point of view), and the number of data integrity errors detected during the benchmark runs. The measures are derived directly from experimentation.

The faultload includes the three fault classes considered in DBench: hardware faults, operator faults and software faults.

The DBench-OLTP prototypes implemented along the DBench project have been used to benchmark many OLTP systems and configurations, including large database management systems (DBMS) such as Oracle 9i, small DBMS as PostgreSQL, running on top of Windows (several versions) and Linux operating systems, and including several database/server configurations.

The two final measures provided by **TPC-C-Depend** are the stationary system availability and the total cost of failures. The measures area evaluated by combining measures obtained from experimentation on the target system (e.g., the percentages of the various *failure modes*) and information from outside the benchmark experimentation (e.g., the failure rate, the repair rate and the cost of each failure mode).

The faultload used in the prototype developed includes exclusively hardware faults, but operator faults have also been considered for validation purpose.

The TPC-C-Depend prototype developed has been used to illustrate the benchmark on Oracle and PostgreSQL.

Main difference between the two OLTP benchmarks

Having two different benchmarks for OLTP systems may raise one important question from the potential benchmark users: which benchmark to use under given circumstances?

The answer to that question lies in the difference between the two benchmarks developed. In addition to the difference in the measures and the way to obtain them presented in the previous paragraph, the main difference between the two benchmarks is mostly related to the approach followed in i) the faultload definition and, as a consequence, to ii) the benchmark potential users. The following paragraphs detail the differences between the two benchmarks and can be used as a guideline for the benchmark user on how to select the appropriate benchmark for a given OLTP system.

Faultload Definition

In DBench-OLTP, the faultload specification is part of the benchmark specification. The faultload specification results from research concerning which faults are possible and representative in different system under benchmarks (SUBs). This includes the database management system (DBMS), the OS and hardware. When applying DBench-OLTP to a new SUB, the faultload has to be ported, i.e., the injection tools and emulation techniques may have to be adapted to the new DBMS/OS/Hardware. However, from a conceptual viewpoint, the faultload is the same. This is especially useful to compare the results across different SUBs.

Users that choose the DBench-OLTP benchmark accept that the faultload defined in the benchmark specification is representative of real scenarios. In this case, as the faultload used is conceptually the same, the results obtained by different users are directly comparable.

In TPC-C-Depend benchmark the faultload is considered dependent of each particular configuration used in the SUB, in particular hardware faults. Fault rates and costs associated to each fault are taken into account. These figures are not part of the benchmark specification and must be provided by the end-user of the benchmark. A formal language to describe the hardware is specifically recommended to emulate the hardware and its faults. Because the hardware can be emulated, several configurations may be evaluated without actually incurring in the costs of buying each one. This is especially useful for developers or IT administrators that are tied to a particular DBMS and wish to select the “best” hardware that should be used with the DBMS.

Users that chose the TPC-C-Depend benchmark must be able to provide fault rates for the systems being considered. In this case, results obtained by different users are not directly comparable because different users may consider different fault rates for the same system.

Potential Benchmark Users

From what precedes, it could be seen that:

• DBench-OLTP benchmark potential users are:

- End-users (i.e., system/IT administrators) when choosing among different (similar) DBMS.
- End-users (i.e., system/IT administrators) with a well defined (already decided) SUB, experimenting different optimisation settings to decide the best trade-off between performance and stability.
- DBMS manufacturers when assessing the quality of the DBMS before releasing it to the market. This is especially useful during development phases where compromises must be made when deciding to improve stability or performance or time-to-market, etc.

• TPC-C-Depend benchmark potential users are:

- DMBS developers intending to recommend a specific hardware for their DBMS.
- Vendors of Information Systems Package (a complete solution including software and hardware).
- Legacy DBMS that must be used in new hardware.

3. Publications and Presentations

The following papers have been published or accepted for publication during the whole project. We first give the list of full papers issued directly from the work performed. Short papers such as "Fast Abstracts", papers and presentations related to DBench are then presented.

DBench Papers

K. Buchacker, M. Dal Cin, H. Höxer, R. Karch, V. Sieh and O. Tschäche, “Reproducible Dependability Benchmarking Experiments Based on Unambiguous Benchmark Setup Descriptions”, International Conference on Dependable Systems and Networks (DSN 2003), San Francisco, Ca, USA, June 22-25, 2003, pp. 469-478.

K. Buchacker and V. Sieh, “UMLinux — A Versatile SWIFI Tool”, Fourth European Dependable Computing Conference (EDCC-4), Toulouse, France, October 23-25, 2002, pp. 159-171.

P. Costa, M. Vieira, H. Madeira and J. Gabriel Silva, “Plug and Play Fault Injector for Dependability Benchmarking”, First Latin-American Symposium on Dependable Computing (LADC 2003), São Paulo, Brazil, October 21-24, 2003.

J. Durães, and H. Madeira, “Emulation of Software Faults by Selective Mutations at Machine-Code Level”, 13th International Symposium on Software Reliability Engineering. (ISSRE 2002) Annapolis, MD, USA, November 12-15, 2002, pp. 329-340.

J. Durães and H. Madeira, “Characterization of Operating Systems Behavior in the Presence of Faulty Drivers through Software Fault Emulation”, Pacific Rim International Symposium on Dependable Computing (PRDC-2002), Tsukuba, Japan, December 16-18, 2002, pp. 201-209.

J. Durães and H. Madeira, “Definition of Software Fault Emulation Operators: a Field Data Study”, IEEE/IFIP International Conference on Dependable Systems and Networks, Dependable Computing and Communications(DSN-2003), San Francisco, CA, USA, June 22-25, 2003 (William Carter award for the best paper).

J. Durães and H. Madeira, “Multidimensional Characterization of the Impact of Faulty Drivers on the Operating Systems Behavior”, Special Issue on Dependable Computing of the journal Transactions of IEICE (Institute of the Electronics, Information and Communication Engineers), vol. E86-D, no 12, December 2003.

J. Durães and H. Madeira, “Generic Faultloads Based on Software Faults for Dependability Benchmarking”, paper accepted for presentation at the IEEE/IFIP International Conference on Dependable Systems and Networks, Dependable Computing and Communications (DSN-2004), Florence, Italy, June 2004.

J. Durães and H. Madeira, "Web-server availability from the end-user viewpoint: a comparative study" (Fast Abstract), accepted for presentation at the IEEE/IFIP International Conference on Dependable Systems and Networks, Dependable Computing and Communications (DSN-2004), Florence, Italy, June 2004.

J. Durães, M. Vieira, and H. Madeira, “Dependability Benchmarking of Web-Servers”, paper accepted for presentation at the International Conference on Computer Safety, Reliability and Security (SAFECOMP-2004), Potsdam, Germany, September 2004.

J. Gracia, D.Gil, L.G. Lemus, P.J. Gil, “Studying Hardware Fault Representativeness with VHDL Models”, in Proc. XVII Conference on Design of Circuits and Integrated System (DCIS 2002), pp. 33-39, Santander, Spain, November 2002

H.-J. Höxer, K. Buchacker, V. Sieh, “Implementing a User Mode Linux with Minimal Changes from Original Kernel”, 9th International Linux System Technology Conference, Köln, Germany, September 4-6, 2002, pp. 71-82.

T. Jarboui, J. Arlat, Y. Crouzet and K. Kanoun, “Experimental Analysis of the Errors Induced into Linux by Three Fault-injection Techniques”, International Conference on Dependable Systems and Networks (DSN-2002), Bethesda, Maryland, USA, pp. 331-336, June 23-26, 2002.

T. Jarboui, J Arlat, Y. Crouzet, K. Kanoun and T. Marteau, “Analysis of the Effects of Real and Injected Software Faults: Linux as a Case Study”, Pacific Rim International Symposium on Dependable Computing (PRDC-2002), Tsukuba, Japan, December 16-18, 2002, pp. 51-58.

T. Jarboui, J. Arlat, Y. Crouzet, K. Kanoun and T. Marteau, “Impact of Internal and External Software Faults on the Linux Kernel,” Special Issue on Dependable Computing of the Journal Transactions of IEICE (Institute of the Electronics, Information and Communication Engineers), vol. E86-D, no 12, pp. 2571-2578, December 2003.

A. Kalakech, T. Jarboui, J. Arlat, Y. Crouzet and K. Kanoun, “Benchmarking Operating System Dependability: Windows 2000 as a Case Study,” 10th Pacific Rim Int. Symp. on Dependable Computing (PRDC-2004), Papeete, French Polynesia, 2-4 March 2004, pp. 261-270.

A. Kalakech, K. Kanoun, Y. Crouzet, J. Arlat, “Benchmarking the Dependability of Windows NT4, 2000 and XP”, International Conference on Dependable Systems and Networks (DSN-2004), Florence, Italy, June 28-July 1, 2004 (accepted).

H. Madeira, J. Costa and M. Vieira, “The OLAP and Data Warehousing Approaches for Analysis and Sharing of Results from Dependability Evaluation Experiments”, IEEE/IFIP International Conference on Dependable Systems and Networks, Dependable Computing and Communications (DSN-2003), San Francisco, CA, USA, June 22-25, 2003.

H. Madeira, J. Durães, and M. Viera, “Emulation of Software Faults: Representativeness and Usefulness”, First Latin-American Symposium on Dependable Computing (LADC 2003), São Paulo, Brazil, October 21-24,2003 (invited paper).

R. Maia, F. Moreira, R. Barbosa, D. Costa, P. Rodriguez, K. Hjortnaes, L. M. Pinho, “Verifying, Validating and Monitoring the Open Ravenscar Real Time Kernel”, 12th International Real Time Ada Workshop (IRTAW12), Viana do Castelo, Portugal, September 15-19, 2003.

M. Rodriguez, N. Silva, J. Esteves, L. Henriques, D. Costa, “Challenges in Calculating the WCET of a Complex Onboard Satellite Application”, in Proc. of the 3rd Euromicro Workshop on WCET analysis, pp. 3-6, Porto, Portugal, July 1, 2003.

J. C. Ruiz, P. Yuste, L. Lemus, P. Gil, “On Benchmarking the Dependability of Automotive Engine Control Applications”, International Conference on Dependable Systems and Networks (DSN-2004), Florence, Italy, June 28 - July 1, 2004.

V. Sieh and K. Buchacker, “Testing the Fault-Tolerance of Networked Systems”,: International Conference on Architecture of Computing Systems (ARCS 2002), Workshop Proceedings. VDE, Karlsruhe, Germany, April 8-11, 2002, pp. 37-46.

O. Tschäche, “Dependability Benchmarking of Linux based Systems”, Proceedings Informatik 2003 - Beiträge des Schwerpunkts Sicherheit - Schutz und Zuverlässigkeit, Frankfurt a.M., Germany, 29 Sept.- 2 Oct. 2003, pp. 237-248.

O. Tschäche, “Deriving Dependability Measures of Measurements Recorded in a Matrix”, International Conference on Architecture of Computing Systems (ARCS 2004), Augsburg, Germany, 23 – 26 March 2004.

M. Vieira and H. Madeira, “Recovery and Performance Balance of a COTS DBMS in the Presence of Operator Faults”, International Performance and Dependability Symposium (jointly organized with DSN-2002), Bethesda, Maryland, USA, June 23-26, 2002.

M. Vieira and H. Madeira, “Definition of Faultloads Based on Operator Faults for DMBS Recovery Benchmarking”, Pacific Rim International Symposium on

Dependable Computing (PRDC-2002), Tsukuba, Japan, December 16-18, 2002, PP. 265-272.

M. Vieira and H. Madeira, "Benchmarking the Dependability of Different OLTP Systems", IEEE/IFIP International Conference on Dependable Systems and Networks, Dependable Computing and Communications (DSN-2003), San Francisco, CA, USA, June 22-25, 2003.

M. Vieira and H. Madeira, “A Dependability Benchmark for OLTP Application Environments”, 29th International Conference on Very Large Databases (VLDB 2003), Berlin, Germany, September 9-12, 2003.

M. Vieira and H. Madeira, “Joint Evaluation of Recovery and Performance of a COTS DBMS in the Presence of Operator”, Special Issue of the Performance Evaluation Journal, Elsevier Science Publication, December 2003.

M. Vieira, A. Casimiro, H. Madeira, “Timely ACID Transactions on DBMS” (Fast Abstract), accepted for presentation at the IEEE/IFIP International Conference on Dependable Systems and Networks, Dependable Computing and Communications (DSN-2004), Florence, Italy, June 2004.

P. Yuste, J. C. Ruiz, L. Lemus, P. Gil, “Non-Intrusive Software-Implemented Fault Injection in Embedded Systems”, First Latin American Symposium on Dependable Computing (LADC 2003), Sao Paulo, Brasil, October 2003. Lecture Notes in Computer Science, No 2847, pp. 23-38.

Papers and Presentations Related to DBench

Several papers devoted to work related to DBench have been published by the partners throughout the project duration. Also several talks related to DBench have been presented by the partners, either in conferences (as invited talks) or workshops without publicly available proceedings. A non-exhaustive list of such papers and talks is given in the following.

A. Albinet, J. Arlat and J.-C. Fabre. “Characterization of the Impact of Faulty Drivers on the Robustness of the Linux Kernel,” in Proc. Int. Conf. on Dependable Systems and Networks (DSN-2004), Florence, Italy, IEEE CS Press, 2004.

J. Arlat, “From Fault Injection Experiments to Dependability Benchmarking,“ position paper, Workshop on Challenges and Directions for Dependable Computing, IFIP WG 10.4, St. John, Virgin Islands, USA, January 2002.

J. Arlat, “From Experimental Assessment of Fault-Tolerant Systems to Dependability Benchmarking“, Invited Talk, IPDPS 2002, April 15-19, Fort Lauderdale, FL, USA.

J. Arlat and Y. Crouzet, “Faultload Representativeness for Dependability Benchmarking”, Workshop on Dependability Benchmarking (jointly organized with DSN-2002), Bethesda, Maryland, USA, pp. F-29-F-30, June 23-26, 2002.

J. Arlat, J.-C. Fabre, M. Rodriguez and F. Salles, “MAFALDA: a Series of Prototype Tools for the Assessment of Real Time COTS Microkernel-based Systems”, in Fault Injection Techniques and Tools for Embedded Systems Reliability Evaluation, Kluwer Academic Publishers, N°ISBN 1-4020-7589-8, 2003

J. Arlat, J. Boue, Y. Crouzet, E. Jenn, J. Aidemark, P. Folkesson, J. Karlsson, J. Ohlsson and M. Rimen, “MEFISTO: a Series of Prototype Tools for Fault Injection into VHDL Models”, in Fault Injection Techniques and Tools for Embedded Systems Reliability Evaluation, Kluwer Academic Publishers, N°ISBN 1-4020-7589-8, 2003

J. Arlat, “RoCADE: Robustness Characterization and Assessment wrt Driver Errors”, 45th IFIP Working Group 10.4 Meeting, Moorea, French Polynesia, March 5-9, 2004.

O. Askerdal, M. Galvert, M. Hiller, N. Suri, “A Control Theory Approach for Analysing the Effects of Data Errors in Safety Critical Systems”, Pacific Rim Dependable Computing Conf., December 2002

J. C. Baraza, J. Gracia, D. Gil, P. J. Gil, “A Prototype of a VHDL-Based Fault Injection Tool. Description and Application”. Journal of Systems Architecture, vol. 47, Issue 10, April 2002, pp. 847-867.

S. Blanc, J.C. Campelo, P.J. Gil, J.J. Serrano, “Stratified Fault Injection using Hardware and Software-Implemented Tools”, 4th. IEEE Design and Diagnostic of Electronic Circuits and Systems,(IEEE DDECS 2001), Györ, Hungary, April 18-20, 2001, pp. 259-266.

S. Blanc, P. Gil, A. Ademaj and H. Sivencrona, J. Torin, “Three Different Fault Injection Techniques combined to improve the Detection Efficiency for Time-Triggered Systems”, IEEE Design and Diagnostics of Electronic Circuits and Systems. Brno, Czeck Republic. April 2002.

S. Blanc, J. Gracia and P. J. Gil, “A Fault Hypothesis Study on the TTP/C using VHDL-based and Pin-level Fault Injection Techniques” IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT 2002), Vancouver, Canada, November 2002.

K. Buchacker and V. Sieh, “Framework for Testing the Fault-Tolerance of Systems Including OS and Network Aspects”, Proc. IEEE High-Assurance System Engineering Symposium, HASE 2001, October 22-24, 2001, pp. 95-105.

K. Buchacker, Höxer, H.-J. and V. Sieh, “Presentation of UMLinux”, High-Tech-Kärwa at IGZ (Innovations- und Gründerzentrum), Erlangen, Germany, July 5, 2002.

K. Buchacker, H.-J Höxer, V.Sieh, “Presentation of UMLinux Systems”, International trade fair for Information Technology, Telecommunications and New Media, Munich, Germany, October 14-18, 2002.

K. Buchacker, “Presentation of FAUmachine” CeBit 2003, Hannover, Germany, March 12-19, 2003

K. Buchacker, M. Dal Cin, H. Höxer, V. Sieh, O. Tschäche and M. Waitz, “Hardware Fault Injection with UMLinux”, International Conference on Dependable Systems and Networks (DSN-2003), San Francisco, Ca, USA, June 22-25, 2003, pp. 670.

K. Buchacker, H.-J. Höxer and V. Sieh, “UMLinux als Sandbox”, IT-Sicherheit im verteilten Chaos, 2003, ISBN 3-922746-49-7, pp. 409-423.

D. Costa, H. Madeira, J. Carreira, J. Gabriel, “Software Implemented Fault Injection: the Xception™ Approach”, Alfredo Benso and Paolo Prineto Eds., “Fault Injection Technique and Tools for VLSI Reliability Evaluation”, Kluwer Academic Publisher, October 2003.

D. Costa, T. Rilho, M.Vieira and H. Madeira, “ESFFI: A Novel Technique for the Emulation of Software Faults in COTS Components”, in Proc. of the Int. Conference on Engineering of Computer-Based Systems (ECBS-2001), Washington DC, April 18, 2001.

M. Dal Cin, “Dependability Benchmarking”, Lecture series for PhD students at K.U.Leuven, September 2003.

J. Durães, D. Costa and H. Madeira, “Accuracy of the Emulation of Software Faults by Machine-Code Level Errors”, in Supplement of the Int. Conference on Dependable Systems and Networks (DSN-2001), (Göteborg, Sweden), pp. B.92-B.93, (FastAbstract), Chalmers University of Technology, Göteborg, Sweden, 2001.

N. Duro, D. Costa, and H. Shaebe, “Criticality Analysis of SCOS-2000 for a SPEC based certification”, 2nd ESA’s Spacecraft Operations System (SCOS2K) Workshop, April 24-25 2002, Darmstadt, Germany.

Th. Glanzmann, H. Höxer and K. Buchacker, “UMlinux – Details”, LinuxDay 2002, Dornbirn, Austria, Nov. 16-17, 2002.

J. Gracia, J.C. Baraza, D. Gil and P.J. Gil, “Comparison and Application of Different VHDL-Based Fault Injection Techniques”, IEEE International Symposium on Defect and Fault Tolerance in VLSI systems (DFT 2001), October 2001, San Francisco, USA.

J. Gracia, J. C. Baraza, D. Gil and P. J. Gil, “Using VHDL-Based Fault Injection to Exercise Error Detection Mechanisms in the Time-Triggered Architecture”, 2002 Pacific Rim International Symposium on Dependable Computing (PRDC-2002). Tsukuba, Japan, December 2002.

D. Gil, J. Gracia, J. C. Baraza and P. J. Gil: Study, “Comparison and Application of Different VHDL-Based Fault Injection Techniques for the Experimental Validation of a Fault-Tolerant System”,. Microelectronics Journal, Special Section on Defect and Fault Tolerance in VLSI Systems (DFT).

M. Hiller, A. Jhumka, N. Suri, “PROPANE: An Environment for Examining the Propagation of Errors in Software”, International Symposium on Software Testing and Analysis (ISSTA-2002), ACM Press Software Engineering Notes, Vol. 27, No.4, pp. 81-85, 2002

H.-J. Höxer, V. Sieh, and K. Buchacker, “UMLinux - A Tool for Testing a Linux System's Fault Tolerance”, In: LinuxTag 2002, Karlsruhe, Germany, June 6-9, 2002.

H.-J Höxer, “Presentation of UMlinux”, BSI, Bonn, Germany, Nov. 28, 2002.

H.-J Höxer, “System im System”, Linux-Magazin 12/2003, 2003, pp. 88-91.

T. Jarboui, “Fault Model Consolidation: Linux as a Case Study”, in Supplement of the IEEE Int. Conference on Dependable Systems and Networks (DSN-2001), (Göteborg, Sweden), pp. A.13-15 (Student paper), Chalmers University of Technology, Göteborg, Sweden, 2001.

T. Jarboui, A. Kalakech and O. Guitton, “Assessment of the Robustness of Windows 2000 via the Injection of Select Bit-flips”, Fourth European Dependable Computing Conference (EDCC-4), Toulouse, France, October, 2002 (Fast Abstract).

A. Johansson, R. Lindström and N. Suri, “A Look at Benchmarking the Dependability of SW and OS's”, in Supplement of the ISSRE'2001 (FastAbstract), November 27-30, Hong Kong.

A. Johansson, R. Lindström and N. Suri, “A Service Based Outlook on Dependability Benchmarking”, Supplement of IEEE DSN, Bethesda, Maryland, USA, June 23-26, pp. B-18-19, 2002 (Fastabstract)

K. Kanoun, “DBench Project (Dependability Benchmarking) ”, IFIP WG 10.4 SIG on Dependability Benchmarking, Pittsburgh, USA, 9-10 November 2000

K. Kanoun, Panel: “Software Testing & Evaluation and Dependability Benchmarking”, 39th Meeting IFIP Working Group 10.4 — Paraty, Brazil — 28 February-3 March, 2001.

K. Kanoun, “DBench Project (Dependability Benchmarking) ”, 39th Meeting IFIP Working Group 10.4 — Paraty, Brazil — 28 February- 3 March, 2001

K. Kanoun, J. Arlat, D. J.G. Costa, M. DalCin, P. Gil, J.-C. Laprie, H. Madeira and N. Suri, “DBench (Dependability Benchmarking)”, in Supplement of the Int. Conference on Dependable Systems and Networks (DSN-2001), (Göteborg, Sweden), DEPPY Workshop, pp. D.12-15, Chalmers University of Technology, Göteborg, Sweden, 2001.

K. Kanoun, H. Madeira and J. Arlat, “A Framework for Dependability Benchmarking”, Workshop on Dependability Benchmarking (jointly organized with DSN-2002), Bethesda, Maryland, USA, pp. F-7-F-8 June 23-26, 2002.

K. Kanoun, “The DBench Framework for Dependability Benchmark Design”, SIGDeB Meeting, Washington DC, USA, February 11-12, 2003

K. Kanoun, “Dependability Benchmarking: How Far are We?”, First Latin-American Symposium on Dependable Computing (LADC 2003), São Paulo, Brazil, 21 - 24 October 2003 (invited keynote).

K. Kanoun, “Dependability Benchmarking of Off-The-Shelf OS Kernels”, 45th IFIP Working Group 10.4 Meeting, Workshop on Open Source and Dependability, Moorea, French Polynesia, 5-9 March 2004.

H. Madeira, Raphael R. Some, F. Moreira, D. Costa, David Rennels, “Experimental Evaluation of a COTS System for Space Applications”, International Conference on Dependable Systems and Networks (DSN-2002), Bethesda, Maryland, USA, June 23-26, 2002.

H. Madeira, J. Costa, M. Vieira, M. Costa, and D. Costa, “Data Warehousing Approach for the Analysis of Dependability Benchmarking Experiments”, Fourth European Dependable Computing Conference (EDCC-4), Toulouse, France, October, 2002 (Fast Abstract).

E. Marsden, J.-C. Fabre and J. Arlat, “Dependability of CORBA Systems: Service Characterization by Fault Injection”, 21st IEEE Symposium on Reliable Distributed Systems (SRDS-2002), Osaka (Japan), October 13-16, 2002, pp.276-285

F. Moreira, R. Maia, D. Costa, N. Duro, P. Rodríguez-Dapena and K. Hjortnaes, “Static and Dynamic Verification of Critical Software for Space Applications”, Data Systems in Aerospace 2003 (DASIA-03), Prague, Czech Republic, June 2-6, 2003.

M. Rodriguez, A. Albinet and J. Arlat, “MAFALDA-RT: a Tool for Dependability Assessment of Real-Time Systems”, 2002 International Conference on Dependable Systems & Networks (DSN-2002), Washington (USA), June 23-26, 2002, pp.267-272

M. Rodriguez, J.-C. Fabre and J. Arlat, “Building SWIFI Tools from Temporal Logic Specifications”, 2003 International Conference on Dependable Systems and Networks (DSN-2003), Dependable Computing and Communications Symposium, San Francisco (USA), June 22-25, 2003, pp.95-104

V. Sieh, “UMLinux”, 5th Linux Setup Party, Nuernberg, Germany, November 23-24, 2002.

M. Vieira, D. Costa and H. Madeira, “Optimization of Performance and Recovery of Database Systems in The Presence of Operation Faults”, in Supplement of the

IEEE Int. Conference on Dependable Systems and Networks (DSN-2001), (Göteborg, Sweden), pp. B102-B103 (FastAbstract), Chalmers University of Technology, Göteborg, Sweden, 2001.

M. Vieira, “DBench - Dependability Benchmarking”, Invited Presentation, ISAT Workshop on Benchmarks for Cognitive Systems, Information Science and Technology (ISAT) study group (funded by DARPA), IBM T. J. Watson Research Center, Yorktown Heights, NY, USA, May 04-05, 2004.

P. Yuste, D. d. Andrés, L. Lemus, J. J. Serrano and P. Gil, “INERTE: Integrated NExus-Based Real-Time Fault Injection Tool for Embedded Systems”, in Proc. of the IEEE/IFIP International Conference on Dependable Systems and Networks (DSN-2003), Tool Demonstrations Session, pp. 669, 2003

P. Yuste, L. Lemus, J. J. Serrano and P. Gil, “A Methodology for Software Implemented Fault Injection Using Nexus", in Proc. of the IEEE/IFIP International Conference on Dependable Systems and Networks (DSN-2003), Supplemental Volume, Fast Abstracts Session, pp. B14-B15, 2003

4. DBench in Conferences, Workshops and Trade Fairs

DBench project members attended more than forty conferences and events, throughout the project duration. They are listed in a chronological order

- 39th IFIP WG 10.4 Meeting. Paraty, Brazil, February 28 - March 3, 2001.
- 9th Brazilian Fault-Tolerant Computing Symposium (SCTF-9). Florianopolis, Brazil, March 5-7, 2001.
- 4th. IEEE Design and Diagnostic of Electronic Circuits and Systems - IEEE DDECS 2001, Györ, Hungary, April 18-20, 2001.
- 2nd HDCC (High Dependability Computing Consortium) Workshop, held in May 6-8, 2001Santa Cruz, California, USA.
- Industry Space Days 2001 (ISD2001) held in May 9-10, 2001 in Noordwijk, The Netherlands, in the ESTEC (European Space Technology Centre).
- IEEE International Conference on Dependable Systems and Networks (DSN-2001), Göteborg, Sweden, July 2001. Attended by all partners.
- On-line Testing Workshop. July 9-11, 2001, Giardini Naxos - Taormina, Italy.
- SAFECOMP'2001, Budapest, Hungary, September 26-28, 2001

- 10th International Conference on Computer Communications and Networks (ICCCN-2001), October 15-17, 2001, Scottsdale, AZ, USA
- 6th IEEE International High-Assurance Systems Engineering Symposium (HASE 2001), October 22-24, 2001, Boca Raton, Florida, USA
- Diskussionskreis "Fehlertoleranz", November 23, 2001, St. Augustin, Germany
- Arbeitskreis Softwarequalitaet Franken (ASQF), November 29, 2001, Erlangen, Germany
- 12th International Symposium on Software Reliability Engineering (ISSRE 2001) November 27-30, 2001, Hong Kong,
- Pan-dependability Workshop, Toulouse December 10-12, 2001, Participation of all partners
- 2001 Pacific Rim International Symposium on Dependable Computing (PRDC 2001), Seoul
- 41st Meeting of IFIP Working Group 10.4 — Workshop on Challenges and Directions for Dependable Computing, Saint John, Virgin Islands, USA, January 4-8, 2002.
- ARCS 2002 - International Conference on Architecture of Computing Systems, Karlsruhe, Germany, April 8-11, 2002
- 2nd ESA's Spacecraft Operations System (SCOS2K) Workshop with a presentation on the deployment of software certification schemes, April 2002.
- 16th IEEE Int. Parallel and Distributed Processing Symposium — Workshop on Fault-Tolerant Parallel and Distributed Systems, Fort Lauderdale, FL, USA, April 15-19, 2002.
- LinuxTag 2002, Karlsruhe, Germany, June 6-9,2002.
- CNES Workshop "Commercial Components for Embedded Computer Systems", June 12, 2002, Toulouse, France
- International Conference on Dependable Systems and Networks (DSN), Bethesda, Maryland, USA, June 2002.
- High-Tech-Kärwa at IGZ (Innovations- und Gründerzentrum), Erlangen, Germany, July 5, 2002.
- 9th International Linux System Technology Conference, Köln, Germany, September 4-6, 2002, pp. 71-82.
- The International Trade Fair for Information Technology, Telecommunications and New Media, Munich, Germany, October 14-18, 2002.

- EDCC-4, Fourth European Dependable Computing Conference, Toulouse, October 23-25, 2002.
- 13th International Symposium on Software Reliability Engineering. ISSRE 2002 Annapolis, MD, USA, November 12-15, 2002.
- 5th Linux Setup Party, Nuernberg, Germany, Nov. 23-24, 2002.M.
- BSI, Bonn, Germany, Nov. 28, 2002.
- Pacific Rim International Symposium on Dependable Computing (PRDC2002), Tsukuba, Japan, December 16-18, 2002.
- SCOS-2000 Users Workshop, ESA/ESOC, Darmstadt, Germany, May 5, 2003.
- Onboard Software Crisis ESA/ESTEC, Noordwijk, The Netherlands, May 7, 2003.
- Data Systems in Aerospace, DASIA 2003, Prague, Czech Republic, June 2-6, 2003.
- The International Conference on Dependable Systems and Networks (DNS 2003), San Francisco, Ca, USA, June 22-25, 2003.
- 12th International Real Time Ada Workshop (IRTAW12), Viana do Castelo, Portugal, September 15-19, 2003.
- First Latin-American Symposium on Dependable Computing, LADC 2003, São Paulo, Brazil, October 21 - 24, 2003
- Galileo Software Engineering Workshop (GSOFT 2003), ESTEC, Noordwijk, Netherlands, November, 2003.
- 27th Computer Software and Applications Conference (COMPSAC 2003), Dallas, Texas, USA, November 3-6, 2003
- 14th IEEE International Symposium on Software Reliability Engineering (ISSRE 2003), Denver, Colorado, USA, November 17-20, 2003.
- 10th IEEE International Pacific Rim Dependable Computing (PRDC-2004), Papeete, Tahiti, French Polynesia, March 3-5, 2004.
- 45th IFIP Working Group 10.4 Meeting, Workshop on Open Source and Dependability, Moorea, French Polynesia, March 5-9, 2004.