



## DBench \* (Dependability Benchmarking)

Karama Kanoun, Henrique Madeira, Mario Dalcin, Francisco Moreira, Juan Carlos Ruiz Garcia

### ► To cite this version:

Karama Kanoun, Henrique Madeira, Mario Dalcin, Francisco Moreira, Juan Carlos Ruiz Garcia.  
DBench \* (Dependability Benchmarking). 5th European Dependable Computing Conference, Apr 2005, Budapest, Hungary. hal-01982631

HAL Id: hal-01982631

<https://hal.science/hal-01982631>

Submitted on 15 Jan 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## **DBench<sup>\*</sup> (Dependability Benchmarking)**

Karama Kanoun (LAAS-CNRS, France),  
Henrique Madeira (University of Coimbra, Portugal),  
Mario Dalcin (Friedrich Alexander University, Germany),  
Francisco Moreira (Critical Software, Portugal),  
and Juan Carlos Ruiz García (Polytechnic University of Valencia, Spain)

### **1 Overview**

The goal of benchmarking the dependability of computer systems is to provide generic and reproducible ways for characterizing system 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 characterise the dependability of computers.

The success of well-established performance benchmarks in comparing performance of components and computer systems probably accounts for the generalised 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 specification of 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 i) implementation of the specification in order to benchmark the dependability of a given target system or component and ii) full understanding and interpretation of the benchmark results.

---

\* Project IST-2000-25425, funded by the European Commission, <http://www.laas.fr/DBench/>  
Project coordinator: Karama Kanoun (Karama.Kanoun@laas.fr)  
LAAS-CNRS, 7 Avenue du Colonel Roche, 31077 Toulouse Cedex 4 — France

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 accomplished (in our proposal) through the implementation of that benchmark in a set of representative systems for a given application domain. These properties have been 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.

## 2 Benchmarks Developed within DBench

The benchmark concepts and the implementation prototypes have been validated through the benchmarking of specific systems. The results illustrate concretely the kind of outcomes that could be obtained from the prototypes. Table 1 summarizes the benchmarks and associated prototypes.

**Table 1.** Benchmarks and prototypes developed within DBench

Application area	Target system	Type of measures	Workload	Faultload
General purpose	Operating system	<ul style="list-style-type: none"> <li>▪ OS robustness</li> <li>▪ OS reaction time</li> <li>▪ OS restart time</li> </ul>	TPC-C Client	<ul style="list-style-type: none"> <li>▪ Application erroneous behaviour (erroneous parameters in selected system calls)</li> </ul>
Onboard space control	Real time kernel	<ul style="list-style-type: none"> <li>▪ Predictability of response time</li> </ul>	Onboard telecommand scheduling	<ul style="list-style-type: none"> <li>▪ idem</li> </ul>
Automotive control system	Embedded control application	<ul style="list-style-type: none"> <li>▪ Dependability (safety-related) measures</li> </ul>	Driving cycles defined in the directive ECC-90/C81/01-1999	<ul style="list-style-type: none"> <li>▪ Stressful workload</li> <li>▪ Memory single bit-flips</li> </ul>
On-Line Transaction Processing systems (OLTP)	Transactional system	<ul style="list-style-type: none"> <li>▪ Transaction throughput</li> <li>▪ Experience Availability</li> <li>▪ DBMS-level specific measures</li> <li>▪ DBMS-level specific dependability measures</li> </ul>	TPC-C based workload	<ul style="list-style-type: none"> <li>▪ Scripts simulating real operator faults</li> <li>▪ Low-level educated mutations</li> <li>▪ Operating system simulated faults</li> <li>▪ Hardware faults emulated by scripts</li> </ul>
On-Line Transaction Processing systems (OLTP)	Transactional system	<ul style="list-style-type: none"> <li>• Failure modes</li> <li>• Steady state availability</li> <li>• Cost of failures</li> </ul>	TPC-C based workload (simulated environment)	<ul style="list-style-type: none"> <li>▪ Mixed level hardware simulation</li> <li>▪ VHDL controlled fault injection</li> <li>▪ Component level hardware faults: <ul style="list-style-type: none"> <li>- Disk fault</li> <li>- Network outages</li> <li>- Power failure</li> </ul> </li> </ul>

Although the benchmark targets are (purposely) quite distinct, the benchmark prototypes considered share some common characterisation dimensions as follows.

**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.

## Conclusion

The work carried out in DBench proved to be very successful, as witnessed i) by its selection in *IST Results*, and ii) by the 35 papers published in conferences and journals, the 47 presentations given in workshops and meetings, and the 6 PhD theses and 10 Master theses prepared within DBench.

## Acknowledgement

Several people contributed to DBench: J. Arlat, K. Buchacker, D. Costa, Y. Crouzet, J. Durães, P. Gil, F. González, H-J. Höxer, T. Jarboui, A. Kalakech, S. Kolelis, J.-C. Laprie, R. Leitão, L. Lemus, M. Rodríguez, V. Sieh, O Tschäche, M. Vieira, M. Waitz, P. Yuste

## Short list of publications

- A. Kalakech, K. Kanoun, Y. Crouzet, J. Arlat. "Benchmarking the Dependability of Windows NT, 2000 and XP", in Proc. *Int. Conf. on Dependable Systems and Networks* (DSN 2004), pp. 681-686, Florence, Italy, 2004.
- F. Moreira, D. Costa, M. Rodríguez, "Dependability Benchmarking of Real-Time Kernels for Onboard Space Systems", Proc. *15th International Symposium on Software Reliability Engineering* (ISSRE 2004), Saint-Malo, France, November 2-5, 2004.
- J. C. Ruiz, P. Yuste, L. Lemus, P. Gil, "On Benchmarking the Dependability of Automotive Engine Control Applications", Proc. *Int. Conf. on Dependable Systems and Networks* (DSN-2004), pp. 857-866, Florence, Italy, 2004.
- O. Tschäche, "Dependability Benchmarking of Linux based Systems", in Proc. *Informatik 2003 - Beiträge des Schwerpunkts Sicherheit - Schutz und Zuverlässigkeit*, Frankfurt a.M., Germany, 29 Sept.- 2 Oct. 2003, pp. 237-248.
- M. Vieira, H. Madeira, "A Dependability Benchmark for OLTP Application Environments", in Proc. *29th International Conference on Very Large Data Bases* (VLDB 2003), pp. 742-753, Berlin, Germany, 2003.