Faultload Representativeness for Dependability Benchmarking

Jean Arlat and Yves Crouzet
LAAS-CNRS, 7 Avenue du Colonel Roche, 31077 Toulouse Cedex 4, France

1. Introduction

In spite of several pioneering efforts (e.g., see [1-4]), and the related initiatives currently being developed — e.g., the IFIP WG 10.4 SIGDeB[1] and the European IST project DBench [5], there is still a significant gap between i) the level of recognition attached to robustness benchmarks and fault injection-based dependability benchmarking, and ii) the wide offer and broad agreement that characterize performance benchmarks (e.g., see [6]). Much effort is needed before the same standing can eventually be achieved.

In practice, basic attributes such as workload, faultload, measurements and measures precisely characterize a dependability benchmark. Clearly, the determination of a *representative faultload* is one of the key issues for specifying a dependability benchmark. In particular, one important question is to figure out whether a focused set of techniques (ideally, a single one) could be identified as sufficient to generate a faultload for many classes of faults, or whether a distinct technique is needed for each class.

2. Fault Injection-based Validation

For the past 30 years, many efforts were reported on the use of fault injection for contributing to the validation of fault-tolerant systems, sometimes in cooperation with other dependability validation techniques (e.g., formal verification or analytical modeling).

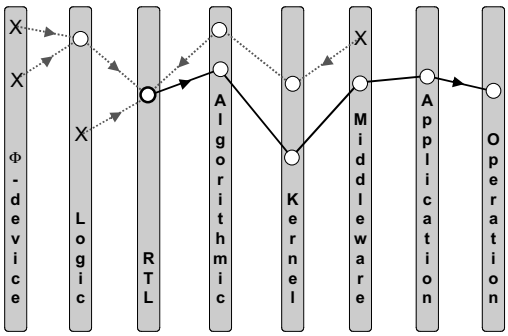
Numerous injection techniques were proposed [7], ranging from i) simulation-based techniques at various levels of representation of the target system (physical, logical, RTL, PMS, etc.), ii) hardware techniques (e.g., pin-level injection, heavy-ion radiation, EMI, power supply alteration, etc.), and iii) software-implemented techniques that support the bit-flip model in memory elements. Many tools were developed to facilitate the conduct of experiments based on these various techniques.

Building on these advances, fault injection made progressively its way to industry, where it is actually part of the development process of many manufacturers, integrators or stakeholders of dependable computer systems. This confirms the pertinence of the approach. Nevertheless, the investigations carried out concerning the comparison of the impact of i) some specific injection technique with respect to real faults (e.g., see [8, 9]) and ii) several injection techniques (e.g., see [10, 11]), have shown mixed results. Some techniques were found to be quite equivalent, while others were rather complementary. Thus, it is necessary to provide a general framework to better analyze the variability of these results and help develop a comprehensive research effort to better address the question raised at the end of Section 1.

3. Faultload Representativeness

In the case of dependability benchmarking, the main question is to identify the technology that is both *necessary* and *sufficient* to generate the faultload to be included into a dependability benchmark. Several important issues have to be accounted for in this effort:

1) As shown by Figure 1, several relevant levels of a computer system can be identified where faults can occur and errors can be observed (e.g., physical-device, logic, RTL, algorithmic, kernel, middleware, application, operation). Concerning faults, these levels may correspond to levels where real faults are considered and (artificial) faults can be injected. Concerning errors, the fault tolerance mechanisms (especially, the error detection mechanisms) provide convenient built-in monitors.
2) For characterizing the behavior of a computer system in presence of faults, it is not necessary *a priori* that the injected faults be "close" to the target faults (reference), it is sufficient that they induce similar behaviors. Indeed, similar errors can be induced by different types of faults (e.g., a bit-flip in a register or memory cell can be provoked by an heavy-ion or as the result of a glitch provoked by a software fault). What is important is not to establish an equivalence in the fault domain, but rather in the error domain.
3) What matters is that the respective error propagation paths *converge* before the level where the behaviors are observed. Two important parameters can be defined on these various levels (Figure 2):
   - the *distance* d_r that separates the level where faults are injected from the ***reference*** fault level(s);
   - the *distance* d_o that separates the level where the faults are injected from the levels their effects are ***observed***.



X: reference fault locations — O: Observation locations

Figure 1: Target system levels and fault pathology

[1] Special Interest Group on Dependability Benchmarking — See: http://www.dependability.org/wg10.4/SIGDeB.

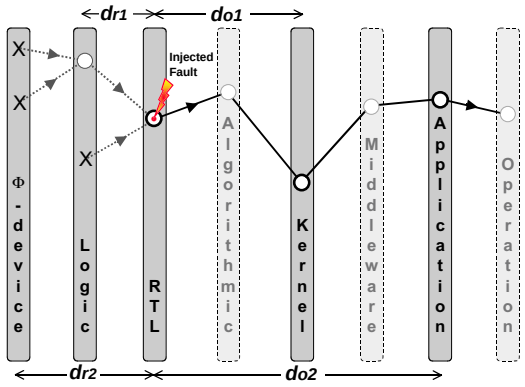


Figure 2: Reference fault and observation distances

The shorter d_r and the longer d_o, the more it is likely that the injected faults will exhibit behaviors similar to those provoked by the targeted reference faults.

4) In practice, it may be the case that the presence of a specific fault tolerance mechanism (FTM) on one target system (and not on the other one(s)) will alter the error propagation paths. This has a significant impact on the scope of (real) faults actually covered by the injected faults, whenever the FTM is implemented at level located between the level of the targeted faults and the level where the faults are injected and thus intercept the error propagation paths. Indeed, assuming a perfect (100%) coverage for the FTM, then representativeness (with respect to the targeted faults) of the benchmark using the faultload characterized by the injected faults would then be zero. This could be simply accounted for by introducing another distance parameter: the *distance* d_m separating the level where the faults are injected from the level where the fault tolerance ***mechanism*** is acting.
5) From a dependability benchmarking point of view, it might not always be possible nor cost-effective to have access to the actual structure of the target system to identify *a priori* a faultload complying with the representativeness property. Accordingly, an alternative could be to favor a standard fault injection technique that is less than perfect, but that is easy to implement and that induces a large set of errors, and then to establish a dialogue with the target system provider in order to derive a fair interpretation or post processing of the benchmark measurements.

4. Concluding Remarks

The paper proposed a framework for addressing and illustrating explicitly — with respect to the various *levels* were faults may affect a computer system — the problems attached to the definition of a *faultload* to be used for dependability benchmarking, Thus, the presentation has focused on the related *distances* that are useful to precisely characterize the links between the target faults, the injected fault and the measurements carried out.

For sake of brevity, other related issues (e.g., *"distance"* between targeted and injected faults with respect to *frequency/distribution* of occurrence) were not considered. Nevertheless, we advocate that a possible extension to this framework aimed at accommodating such a concern can be derived from the results presented in [12].

Finally, it is worth pointing out that, even in such a preliminary form, this framework was found helpful within the DBench project [5] for defining and coordinating the experiments aimed at assessing the *representativeness* and *equivalence* of various fault injection techniques.

Acknowledgement. This work is partially supported by the DBench project (IST 2000-25425). Our thanks go to Jean-Charles Fabre, Karama Kanoun and Jean-Claude Laprie at LAAS, as well as Henrique Madeira (Coimbra Univ., Portugal) and Philip Koopman (CMU, USA), for the insightful discussions on the issues addressed herein.

References

[1] T. K. Tsai, R. K. Iyer, D. Jewitt, "An Approach Towards Benchmarking of Fault-Tolerant Commercial Systems", *Proc. FTCS-26,* Sendai, Japan, 1996, pp. 314-323 (IEEE CS Press).

[2] A. Mukherjee, D. P. Siewiorek, "Measuring Software Dependability by Robustness Benchmarking", *IEEE Trans. on Software Engineering*, vol. 23, no. 6, pp. 366-323, 1997.

[3] P. Koopman, J. DeVale, "Comparing the Robustness of POSIX Operating Systems", *Proc. FTCS-29,* Madison, WI, USA, 1999, pp. 30-37 (IEEE CS Press).

[4] J. Arlat, J.-C. Fabre, M. Rodríguez, F. Salles, "Dependability of COTS Microkernel-Based Systems", *IEEE Trans. on Computers*, vol. 51, no. 2, pp. 138-163, 2002.

[5] K. Kanoun, J. Arlat, D. J. G. Costa, M. Dal Cin, P. Gil, J.-C. Laprie, H. Madeira and N. Suri, "DBench – Dependability Benchmarking", *Supplement of Proc. DSN-2001,* Göteborg, Sweden, 2001, pp. D.12-D.15. (http://www.laas.fr/DBench).

[6] J. Gray (Ed.), *The Benchmark Handbook for Database and Transaction Processing Systems, S*an Francisco, CA, USA: Morgan Kaufmann Publishers, 1993.

[7] J. V. Carreira, D. Costa, J. G. Silva, "Fault Injection Spot-checks Computer System Dependability", *IEEE Spectrum*, vol. 36, pp. 50-55, August 1999.

[8] M. Daran, P. Thévenod-Fosse, "Software Error Analysis: A Real Case Study Involving Real Faults and Mutations", *Proc. ISSTA'96,* San Diego, CA, USA, 1996, pp. 158-171 (ACM Press).

[9] H. Madeira, D. Costa, M. Vieira, "On the Emulation of Software Faults by Software Fault Injection", *Proc. DSN-2000,* New York, NY, USA, 2000, pp. 417-426 (IEEE CS Press).

[10] D. T. Stott, G. Ries, M.-C. Hsueh, R. K. Iyer, "Dependability Analysis of a High-Speed Network Using Software-Implemented Fault Injection and Simulated Fault Injection", *IEEE Trans. on Computers*, vol. 47, no. 1, pp. 108-119, 1998.

[11] P. Folkesson, S. Svensson, J. Karlsson, "A Comparison of Simulation Based and Scan Chain Implemented Fault Injection", in *Proc. FTCS-28,* Munich, Germany, 1998, pp. 284-293 (IEEE CS Press).

[12] D. Powell, E. Martins, J. Arlat, Y. Crouzet, "Estimators for Fault Tolerance Coverage Evaluation", *IEEE Trans. on Computers*, vol. 44, no. 2, pp. 261-274, 1995.