DBench

Dependability Benchmarking

IST-2000-25425

Project Presentation

Report Version: Deliverable PP

Report Preparation Date: 20 July 2001

Classification: Public Circulation

Contract Start Date: 1 January 2001

Duration: 36m

Project Co-ordinator: LAAS-CNRS (France)

Partners: Chalmers University of Technology (Sweden), Critical Software (Portugal), University of Coimbra (Portugal), Friedrich Alexander University, Erlangen-Nürnberg (Germany), LAAS-CNRS (France), Polytechnical University of Valencia (Spain).

Sponsor: Microsoft (UK)

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

Contract Number	IST-2000-25425
Project acronym	DBench (http://www.laas.fr/DBench)
Project name	Dependability Benchmarking
Key action	CPA4
Action line	IST-2000-5.1.4

List of participants
LAAS-CNRS, Toulouse, France (Coordinator)
Chalmers University of Technology, Sweden
Critical Software, Coimbra, Portugal
University of Coimbra, Portugal
Friedrich Alexander University, Erlangen-Nürnberg, Germany
Polytechnical University of Valencia, Spain
Microsoft, Cambridge, UK (sponsor)

Advisory Board
Astrium (France), Carnegie Mellon University (USA), INDRA (Spain), Oracle (Portugal), Saab Ericsson Space (Sweden), Thales (France).

Total cost: 2382906 Euros

Commission funding: 1628901 Euros

Coordinator contact details
Karama Kanoun (kanoun@laas.fr)
LAAS-CNRS, 7, Avenue du Colonel Roche
31077 Toulouse Cedex-4, France

Project main goals

DBench aims at defining a conceptual framework and an experimental environment for benchmarking the dependability of Commercial Off-The-Shelf[1] (COTS) components and COTS-based systems. It will allow system developers and end-users to 1) characterize and evaluate the dependability of a component or a system, 2) compare the dependability of alternative or competing solutions, 3) identify malfunctioning or weak parts, requiring more attention and perhaps necessitating improvements by tuning a particular component to enhance its dependability (e.g., by using wrapping), or tuning the system architecture (by adding fault tolerance mechanisms or spare units, for example), to ensure an appropriate dependability level. The two final objectives of the project are a report presenting the concepts, specifications and guidelines for dependability benchmarking and a set of dependability benchmark prototypes.

[1] Indeed our work will address more precisely Off-The-Shelf (including open source software) and commercial off-the-shelf components. However, we will indifferently use the term COTS to designate either of them

Key issues

The pervasive use of COTS components in a wide range of computer systems (e.g., embedded systems or database/web servers based on COTS operating systems and platforms), and the increasing dependence of our society on such systems, make it urgent to address the dependability of COTS components and COTS-based systems. In addition, the use of COTS is extended to systems targeted for high dependability environments (i.e., life-critical or business-critical applications). Although information about functionality and performance of COTS components is well characterized, there are no existing guidelines for characterizing the *behaviour* of these components in the presence of internal and/or external faults. This not only constrains the developers ability to utilize COTS for designing dependable systems, it also comes to the situation that objectively evaluating the behaviour of a COTS-based system is currently either an *ad hoc* process or essentially non-existent.

The goal of a dependability benchmark is to quantify system or component dependability through well-specified measures. Weak parts can thus be identified and emphasis may be put on enhancing system ability to resist to faults. Hence, knowing the behaviour of systems in presence of faults will allow construction of dependable systems.

Another important objective of dependability benchmarking is to help the computer industry to accelerate the improvement of computer's (including hardware and software) resilience to faults, even for general-purpose systems with no special fault tolerance mechanisms. Moreover, dependability benchmarking can constitute useful inputs for the standardization bodies to define guidelines for the certification of safety-critical systems incorporating COTS components.

Technical approach

Although private discussions with large system provider companies and some published work reveals that some relevant work on dependability benchmarks is undertaken in a sparse manner, the current state-of-the-art is still immature, lacks conformity and approval across companies is basically globally non-existent. We utilize the experience from the performance benchmarking community, especially the issues developed in that area on *what* to measure and *how* to measure attributes.

Typically, a performance benchmark is a workload to be run on a system and the output is a performance measure. Analogously, we define a dependability benchmark as a set of workload and faultload to be executed on the system to ascertain dependability measures or indicators on the ability of a system to cope with both internal and external faults. Typical measures that will be directly observable on the target system will be identified during the project. Our starting point is to consider classical measures already well established in the *dependability arena,* (i.e., error detection efficiency, error detection latency, time to diagnosis, failure modes, recovery factors) and metrics adapted from the *performance benchmark arena* such as system response time or number of transactions, in presence of faults. We will also evaluate system level measures**,** such as reliability, availability and safety, obtained from models (incorporating the experimental measures as parameters as well as parameters provided from elsewhere). The results of dependability benchmarks will characterize the property of a system such that reliance can be justifiably placed on the service it delivers.

To facilitate project control and monitoring, the work is structured in four workpackages (denoted respectively WP1 to WP4) in such a way that their sequence reflects the project progress.

WP1: Definition of the conceptual framework for system benchmarking.

WP2: Identification and evaluation of the enabling technologies.

WP3: Benchmark definition, validation and application to pilot experiments.

WP4: Consolidation of the conceptual framework with the experimental results.

The **conceptual framework** addresses the most relevant issues involved in dependability benchmarking. It is a fundamental part of the project: the studies and experiments that follow will abide to common guidelines. Building upon relevant advances in performance and dependability evaluation, we are i) surveying the state-of-the-art in dependability assessment, performance benchmarking and robustness benchmarking, and ii) investigating the basic concepts for dependability benchmarking.

To put into practice the conceptual framework, **enabling technologies** are being investigated and adapted in some respects. One major extension of dependability benchmarking with respect to classical performance benchmarking concerns the characterization of the behaviour of a target system in the presence of a specific faultload (in addition to the workload). Thus, fault injection will play a central role. However, the technologies that were classically used in fault injection experiments are not directly usable for benchmarking purpose. Indeed, to be adopted and so as to provide meaningful quantitative results, benchmarking calls for the proposal of a new set of easy-to-handle, yet reliable, technologies. We will address this fundamental issue by considering: measurements to be performed on the target system, fault representativeness and workload and faultload selection.

Experiments will be performed to achieve a comprehensive analysis: different dependability benchmark prototypes will be defined and developed for two major application-areas (embedded and transactional applications). The benchmark prototypes will actually be implemented in two different families of COTS operating systems (Windows and Linux), allowing a cross evaluation of the concepts and of the enabling technologies in a true dependability benchmark context. The final goal of the experiments is the validation of the dependability benchmark prototypes, in the sense of assuring that the benchmarks outcomes represent a practical and meaningful characterization of the dependability properties of the target systems, both from the end-user and the system developer's point of view.

Finally, the **consolidation** of the whole set of results will allow to push further the enabling technologies and to finalize the benchmark concepts and prototypes. All advances and developments made in the project are summed up, discussed and put into their final format.

Expected achievements and impact

Concepts and guidelines, supported by tools, for benchmarking system dependability developed in DBench are innovative paradigms. They will allow people involved in and interacting with systems to characterise their behaviour in the presence of faults. Thus allowing for dependability attributes such as reliability, safety and availability, to be quantified. This evaluation will allow to: make trade-offs between the evaluated attributes, and identify weak parts so as emphasis may be put on developing means for enhancing their dependability. Indeed, the knowledge of the behaviour of systems in presence of faults constitutes a major support for constructing dependable systems. The selected target systems are general COTS operating systems, operating systems for embedded applications and database applications (standard and widely accepted performance transaction-processing benchmarks implemented over an Oracle database management system). All these COTS-based systems are widely used in largely distributed and open systems.