From Experimental Assessment of Fault-Tolerant Systems to Dependability Benchmarking

Jean Arlat

Work partially supported by project IST 2000-25425
Dependability Assessment

- **Objectives**
  - Evaluation of Dependability Measures (Reliability, Availability, etc.)
  - Verification of Properties
    - Nominal Service
    - Service in presence of Faults
  - Characterization of Behavior in Presence of Faults
    - Failure modes
    - Efficiency of fault tolerance

- **Methods and Techniques**
  - Axiomatic
    - Stochastic processes
    - Model checking
  - Empirical
    - Field measurement
    - Robustness testing
    - Fault injection
### Fault Tolerance Validation

<table>
<thead>
<tr>
<th><strong>Dependability</strong></th>
<th><strong>Fault Tolerance (FT)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>- FT mechanisms = human artefacts (not perfect)</td>
<td>- Impact on dependability measures</td>
</tr>
<tr>
<td>- Calibration of models</td>
<td>- Estimation of FT efficiency</td>
</tr>
<tr>
<td>- Formal approaches limits</td>
<td>- Experimental approaches</td>
</tr>
<tr>
<td>- Threats = rare event</td>
<td>- Controlled experiments</td>
</tr>
</tbody>
</table>

### Fault Injection

Validation of fault tolerance wrt specific inputs it is designed to deal with: **the faults**
Fault Injection

- Test and evaluation of fault-tolerant systems & FT mechanisms
- Explicit characterization of faulty behaviors
Analytical & Experimental Evaluations

1. Target System

2. Construction of Axiomatic Models
   - Calibration & Validation

3. Processing by Stochastic Processes

4. Evaluation of Dependability Measures

5. Construction of Empirical & Physical Models

6. Fault Injection Experiments

7. Estimation of FTMs Coverage Parameters

8. Feedback to Construction of Axiomatic Models
Fault Injection as A Design Aid

[ESPRIT Project Delta-4]

Target Architecture

Model

Coverage factors

Target system | $C_T$ | $\bar{C}_{T,1}$ | $\bar{C}_{T,2}$ | $\bar{C}_{T,3}$
--- | --- | --- | --- | ---
NAC Std - Amp V1 | 79.08% | 2.32% | 11.77% | 6.83%
NAC Std - Amp V2.5 | 99.55% | 0.32% | 0.00% | 0.12%
NAC Std - Amp V2 | 85.02% | 8.73% | 2.80% | 3.45%
NAC Duplex - Amp V2.5 | 90.32% | 7.79% | 1.05% | 0.84%
NAC Duplex - Amp V1 | 99.55% | 0.32% | 0.00% | 0.12%
(Software) Component-based Development

- Integration of previously developed components (commercial or not)
  \[\text{OTS} = \text{either COTS or OSS}\]

- Main advantages
  - productivity and time to market
  - incorporation of technology advances
  - compatibility with industry standards
  - quality (widely deployed components)

Applications Requiring High-level of Dependability

- Lack of observability and controlability: ? Dependability

- Global cost (development, validation, usage, etc.): ?
Fault Injection-based Dependability Characterization of COTS SW

15 « COTS » OSs
[Koopman & DeVale 99 (FTCS-29)]

Normalized failure rate (%)

0 10 20 30 40 50

Abort. Silent Restart Catastrophic

Invalid parameters in system calls at POSIX Interface

Bit flips
- on parameters of kernel calls (ext.)
- in kernel memory space (int.)

Chorus microkernel
[Arlat et al. 2002 (ToC Feb.)]
Faul Injection
Well-Accepted by Industry as a Whole

- **Provider**
  - IBM, Intel, Sun MicroSystems,

- **Integrator**
  - Ansaldo Segnalamento Ferroviario, Astrium, DaimlerChrysler, Saab Ericsson Space, Siemens, Technicatome, THALES, Volvo,

- **Stakeholder**
  - Electricité de France, ESA, NASA (JPL),

- **Consultant**
  - Critical Software, Cigital,
Dependability Benchmarking

Naive View ... :-)

Dependability Benchmarking

≈

Performance Benchmarking

+
More Realistic View

- Dependability Assessment
- Performance Benchmarking

Desired Properties:
- Agreement/Acceptance
- Usefulness
- Fairness
- Usability
- Portability
- etc.
Dependability Benchmarking Framework

Analysis

Modeling

Experimental Features & Measures

Faultload Workload Target System

Comprehensive Measures

Experimental Features & Measures

A

B

Experimentation

Features & Measures
Challenges

Properties
• Agreement/Acceptance
• Representativeness
• Usefulness,
• Portability,…

Dimensions
• Faultload
• Workload
• Measurements
• Measures

Real Faults
Fault Injection Techniques

Faultload(s) for Dependability Benchmarking
The Fault Injection Techniques

**Logical & Information**
- Simulation Model
  - Compile-time software mutation
    - SESAME, ...
- Software-Implemented
  - Communication
    - FIMD debugger
  - Task
    - FIESTA
  - Executive
    - Ballista, (DE)FINE, MAFALDA
  - Memory
    - DEF.I, SOFIT, ...
  - Instruction set
    - FERRARI
  - Processor
    - Xception, ...

**Physical**
- Prototype/Real System
  - Built-in test devices
    - (SCIFI) FIMBUL

- Target System
  - Physical

**MEAN**
- Logical & Information
  - System
    - DEPEND, REACT, ...
  - RT Level
    - ASPHALT, ...
  - Logical Gate
    - Zycad, Technost, ...
  - Switch
    - FOCUS, ...
- Physical
  - Wide Range
    - MEFISTO, VERIFY, ...

Heavy-ions
- FIST, ...
EM perturbations
- TU Vienna
Pins
- MESSALINE, Scorpion, DEFOR, RIFLE, AFIT, ...
Target System Levels & Fault Pathology

X: Fault locations  O: Observation locations
Target System Levels & Fault Pathology

- Fault locations: X
- Observation locations: O
Target System Levels & Fault Pathology

X: Fault locations
O: Observation locations
Target System Levels — Ref. & Obs. dist.

Dist. to fault ref.   Dist. to observation
Mutation vs. Real Software Faults

- Critical software from civil nuclear field - 12 programming faults
- Sets of Errors Provoked => 395 distinct errors

Impact of the Mutation Experiments (wrt Real Faults)

Not found in error set created by real faults
SWIFI vs. Software Faults

- **SW Fault Classification (ODC)**
  - Assignment
  - Checking
  - Interface
  - Timing
  - Algorithm
  - Function

  Can be (easily) emulated by SWIFI

-> Main open issues are related to fault-trigerring conditions?
SWIFI Bit-flips vs. SEUs

- Computerized system (80C51 µcontroller)
- Activity: 6x6 matrix multiplication

SWIFI Bit-flips

- 3% Tolerated
- 46% Erroneous result
- 51% Sequence loss

SEUs Radiation

- 1.5% Tolerated
- 44% Erroneous result
- 54.5% Sequence loss

Several ΦFI techniques

[Karlsson et al. 1998 — DCCA-5]

- **MARS fault-tolerant distributed system** (prior version of TTP architecture)

  - Heavy-ion radiation
  - Electromagnetic interferences
  - Pin-level (forcing)
Scan Chain-Implemented Fault Injection vs. Simulation

[Folkesson et al. 1998 — FTCS-28]

- 32-bit Processor (Saab Ericsson Space)
- Control program

**SCIFI**

- Overwritten 42%
- Exceptions 23%
- Control Flow 1%
- Other 0%
- Failure 6%

**Simulation (VHDL)**

- Overwritten 61%
- Exceptions 16%
- Control Flow 3%
- Other 1%
- Failure 4%
- Latent 15%
Comprehensive & Coordinated Study

- **Physical Faults** *(VHDL Simulation and SWIFI)*

- **Software Faults**
  - Application *(Mutation, Controlled-SWIFI)*
  - OS *(Bit-flips on system call parameters, Invalid system call parameters, Bit-flips on internal function calls, real faults)*

- **Operator & Maintenance Administrator Faults** *(emulation scripts, real faults from field data and interviews)*
Software Faults in OSs

- **Target:**
  - Linux OS
  - Scheduling component

---

**Invalid API parameter**
- No signaling: 30%
- Error code: 66%
- App. hang: 4%

**Bit-flipped API parameter**
- No signaling: 27%
- App. hang: 8%
- Error code: 65%

**Bit-flipped internal function parameter**
- No signaling: 53%
- Exception: 13%
- App. hang: 18%
- Error code: 4%
- Kernel hang: 12%

**Counts:**
- Invalid API parameter: 507
- Bit-flipped API parameter: 1890
- Bit-flipped internal function parameter: 552
More information

- **DBench - Dependability Benchmarking**
  [IST Project 2000-25425]
  -> [http://www.laas.fr/dbench/](http://www.laas.fr/dbench/)

- **IFIP WG 10.4 - SIGDeB**
  Special Interest Group on Dependability Benchmarking
  -> [http://www.dependability.org/wg10.4/SIGDeB](http://www.dependability.org/wg10.4/SIGDeB)