Dependability Benchmarking of Operating Systems

Karama KANOUN
LAAS-CNRS

Workshop on Dependability Benchmarking
In conjunction with the 16th IEEE International Symposium on Software Reliability Engineering
Chicago, IL, USA — November 8, 2005

Context

DBench (Dependability Benchmarking)

European IST Program — Project IST-2000-25425

January 2001- March 2004

Consortium

Critical Software (P), University of Coimbra (P), Friedrich-Alexander University, (D), LAAS-CNRS (F), Polytechnic University of Valencia (E)

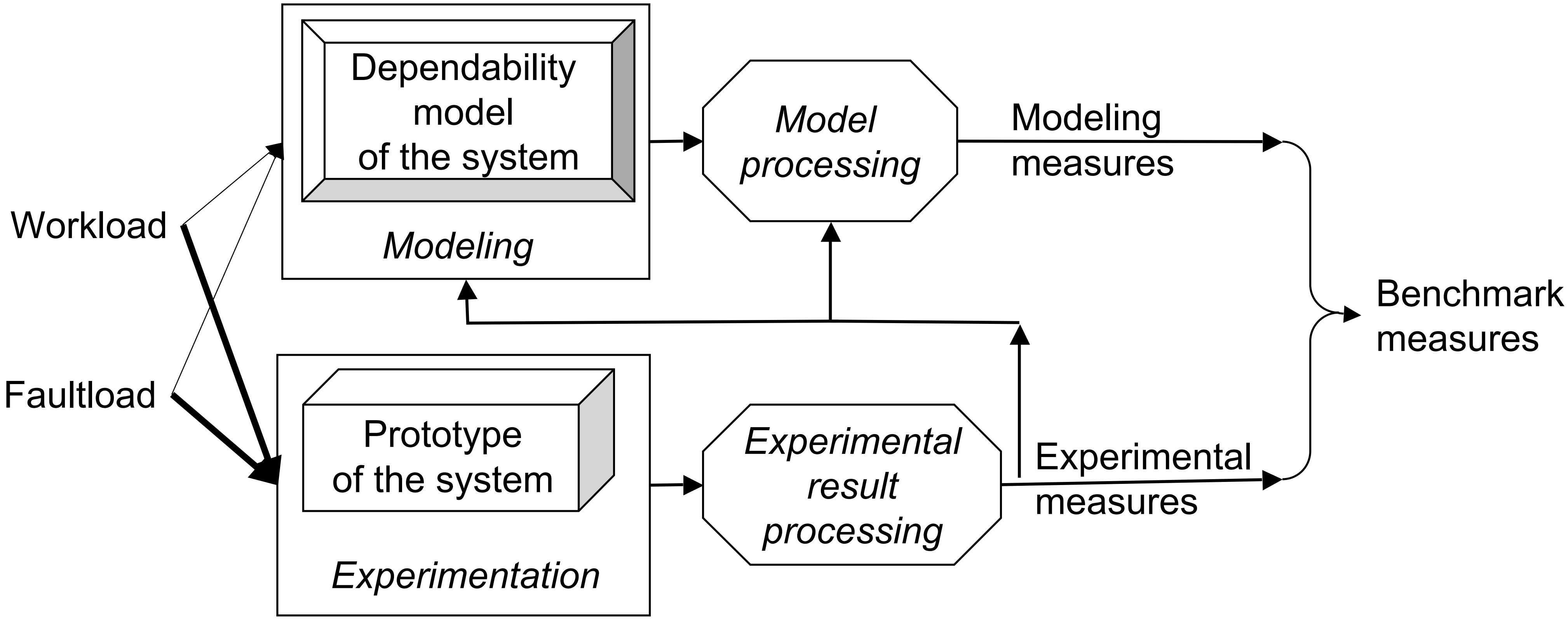
Main results

Benchmarking framework

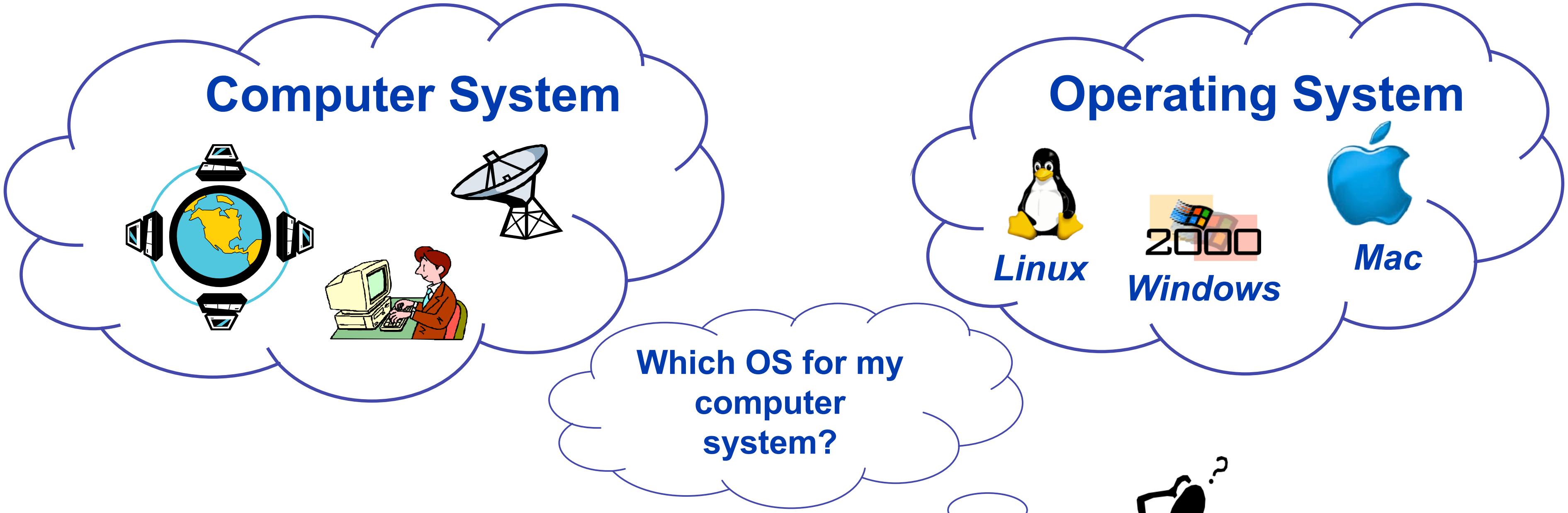
A set of benchmarking principles common to all DBench benchmarks

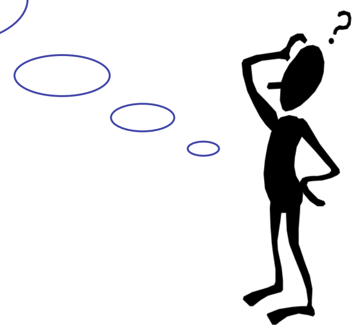
A set of benchmarks

DBench Benchmarking Framework



OS Benchmarking: User Point of View



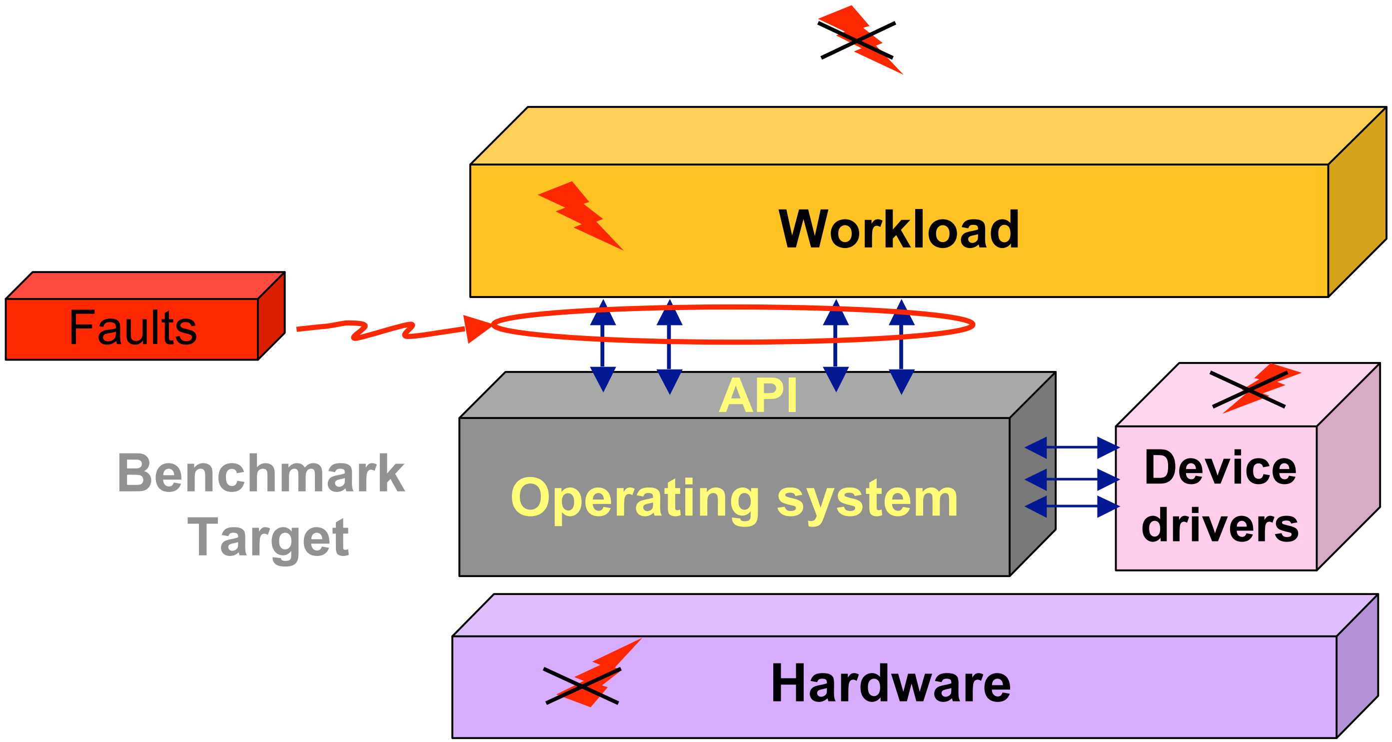


- Limited knowledge: functional description
- Limited accessibility and observability
- Limited intrusiveness and interference

⇒ **Black-box approach ⇒ robustness benchmark**

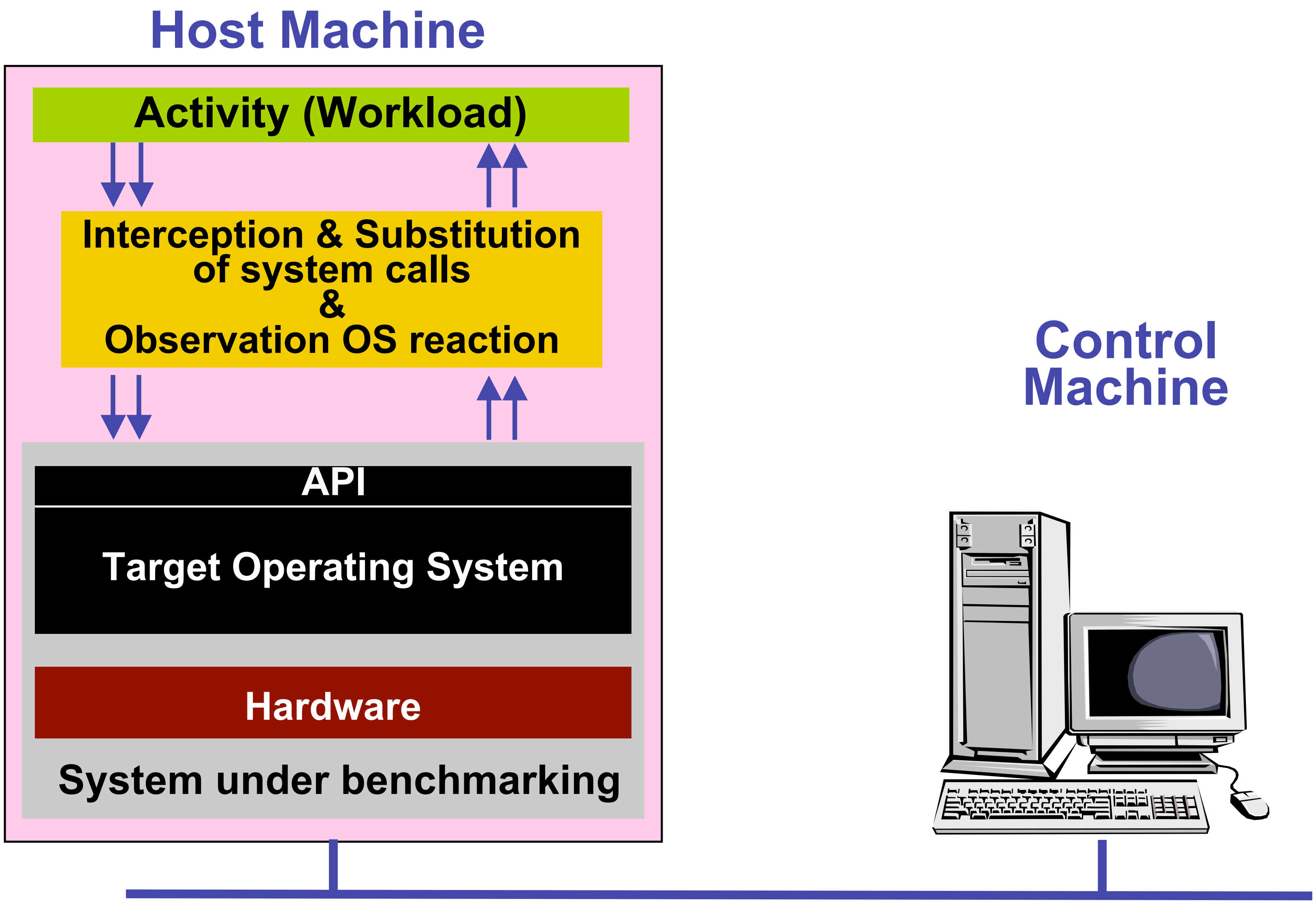
⇒ **Properties: reproducibility, repeatability, portability, representativeness, acceptable cost/effort**

Benchmarking wrt Class of Faults?

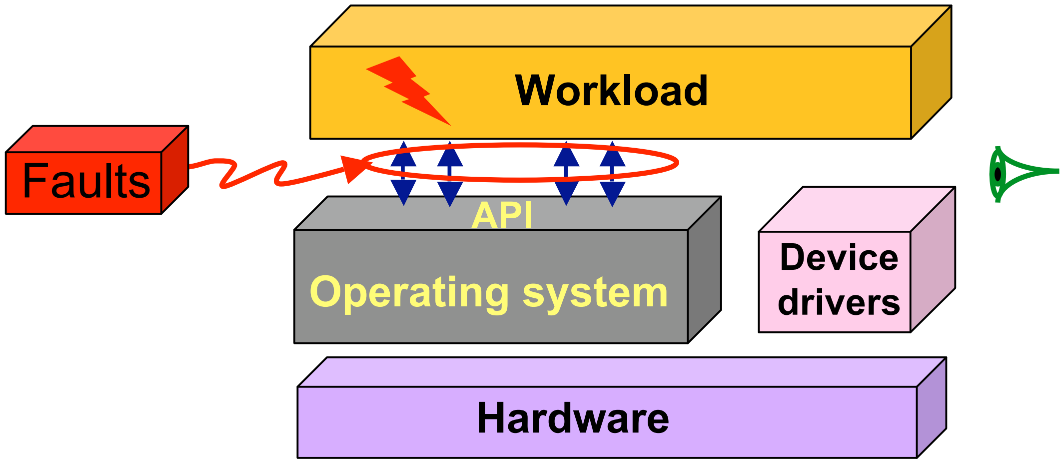


Wrt application erroneous behavior

Experimental Set-up



OS Benchmark & Measures



OS Outcomes

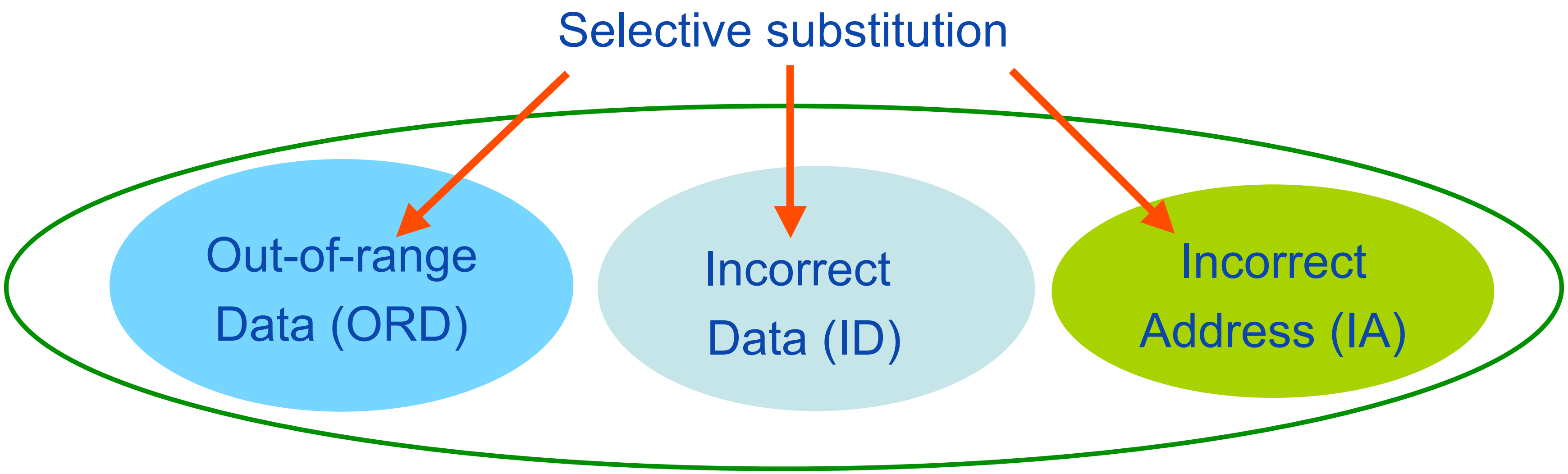
SEr	Error code
SXp	Exception
SPc	Panic
SHg	Hang
SNS	No signaling

Measures

- POS: OS Robustness [%SEr %SXP %SPc %SHg %SNS]
- Texec: OS reaction time in the presence of faults
- Tres: OS Restart time after fault insertion

Execution Profile

- Workload
  - TPC-C Client, Java Virtual Machine, PostMark
- Faultload
  - Corruption of parameters of all system calls



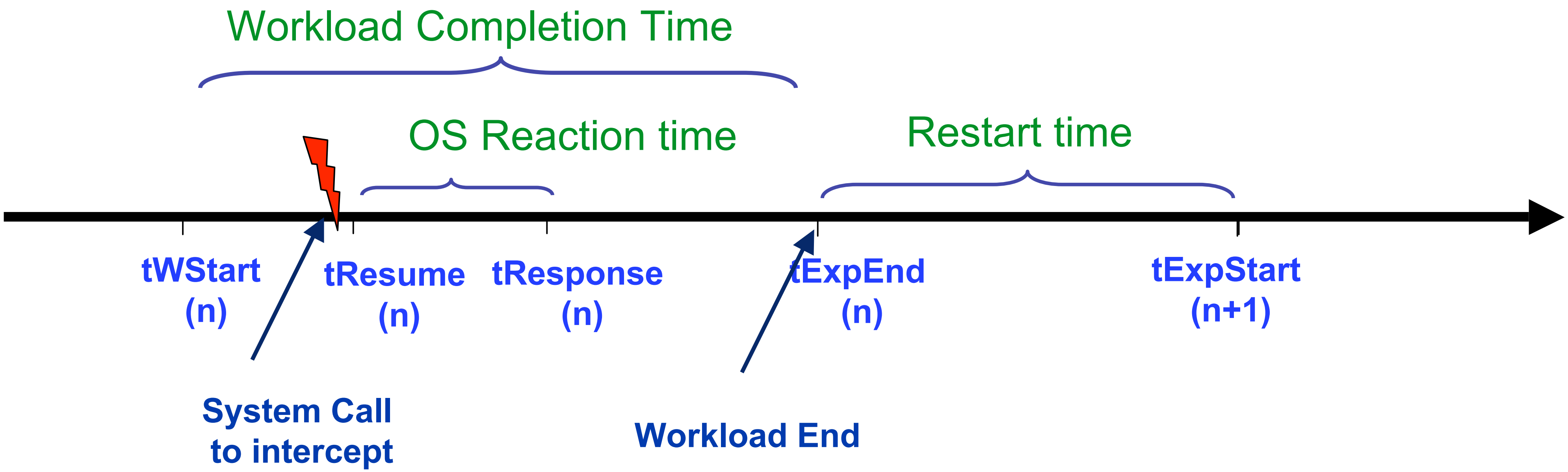
Dependability Benchmarks with PostMark WL

	# system calls	# experiments
Windows NT 4	25	418
Windows 2000	25 + 1 + 1	433
Windows XP	25 + 1	424
Windows NT 4 Server	25	418
Windows 2000 Server	25 + 1 + 1	433
Windows 2003 Server	25 + 1 + 1	433

Linux 2.2.26	15 + 1	206
Linux 2.4.5	15 + 1	206
Linux 2.4.26	15 + 1	206
Linux 2.6.6	15 + 2	228

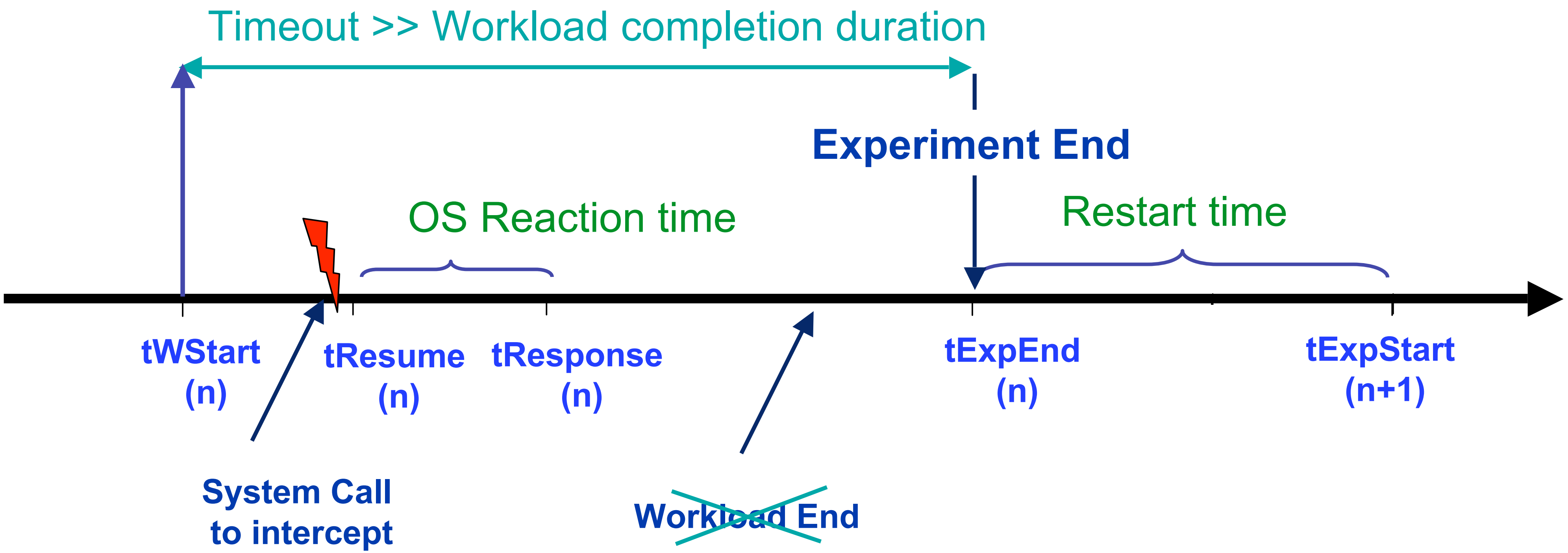
Measurements

Experiments with Workload completion



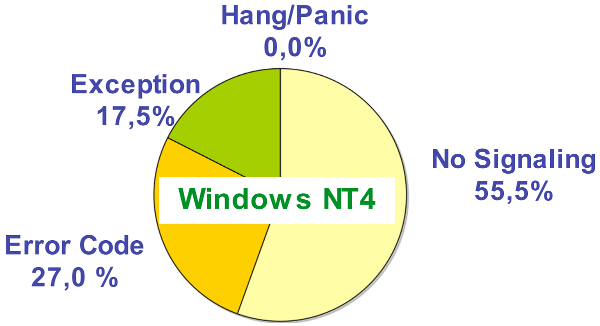
Measurements

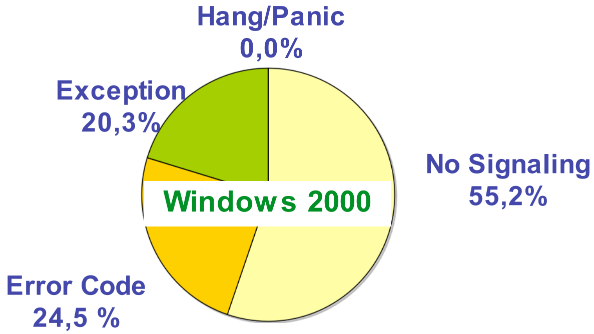
Experiments without Workload completion

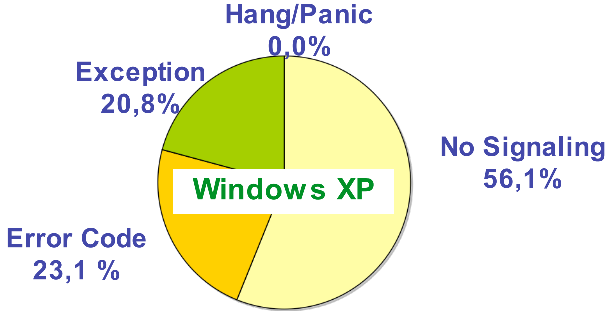


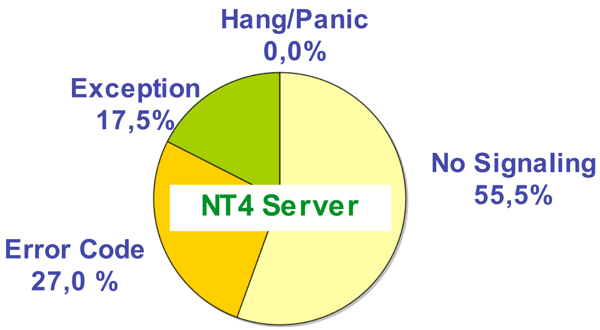
Robustness (WL = PostMark)

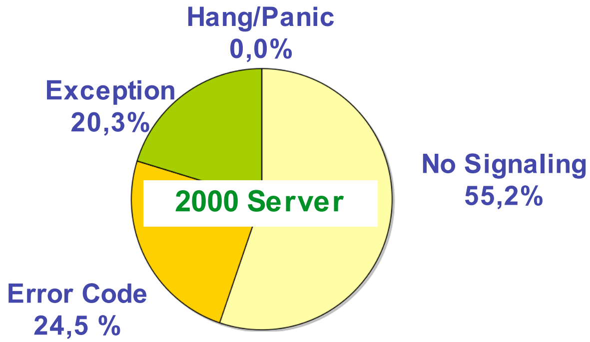
Windows

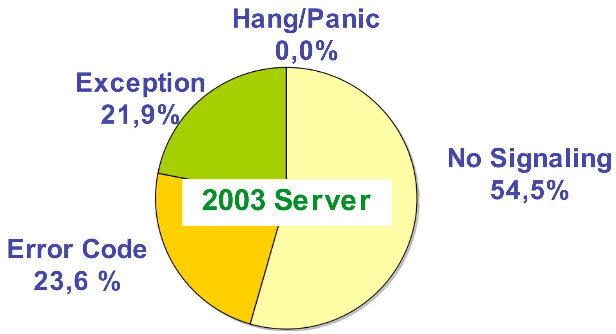




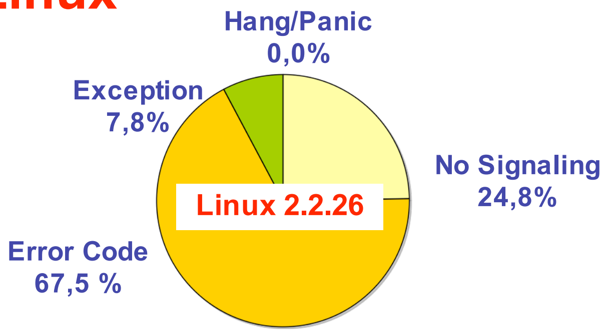


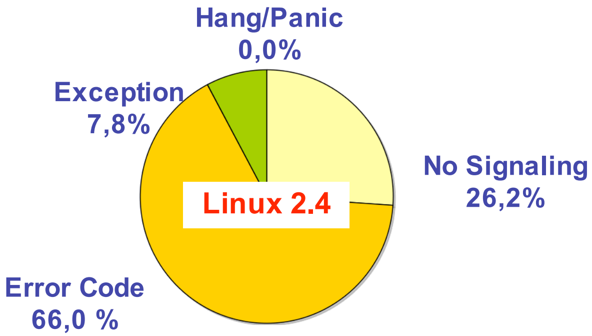


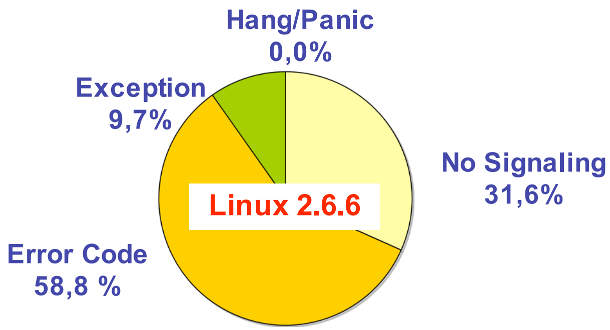




Linux

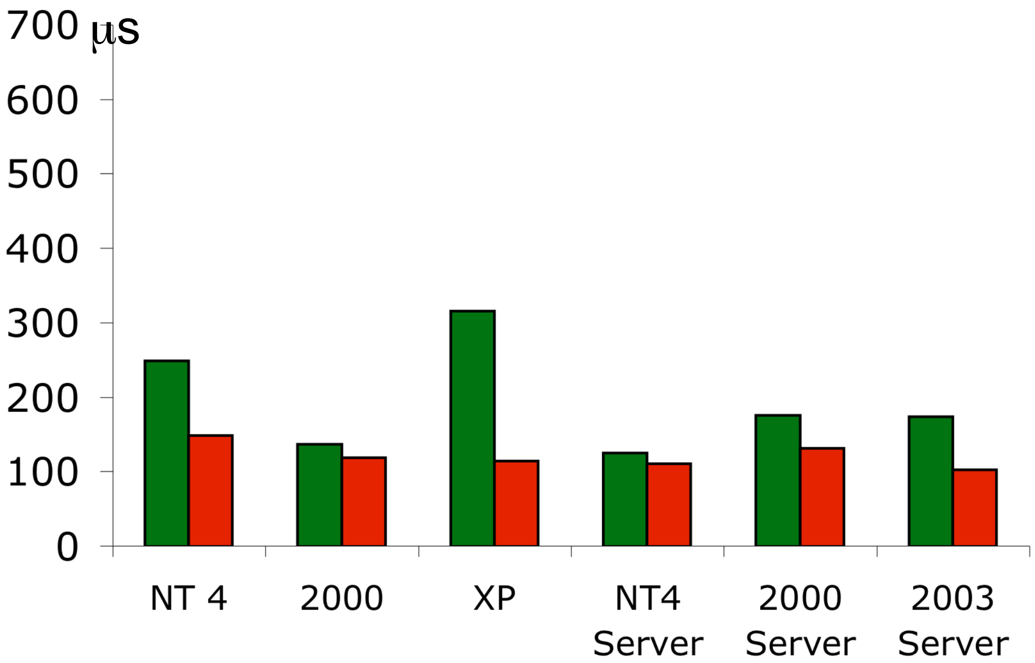




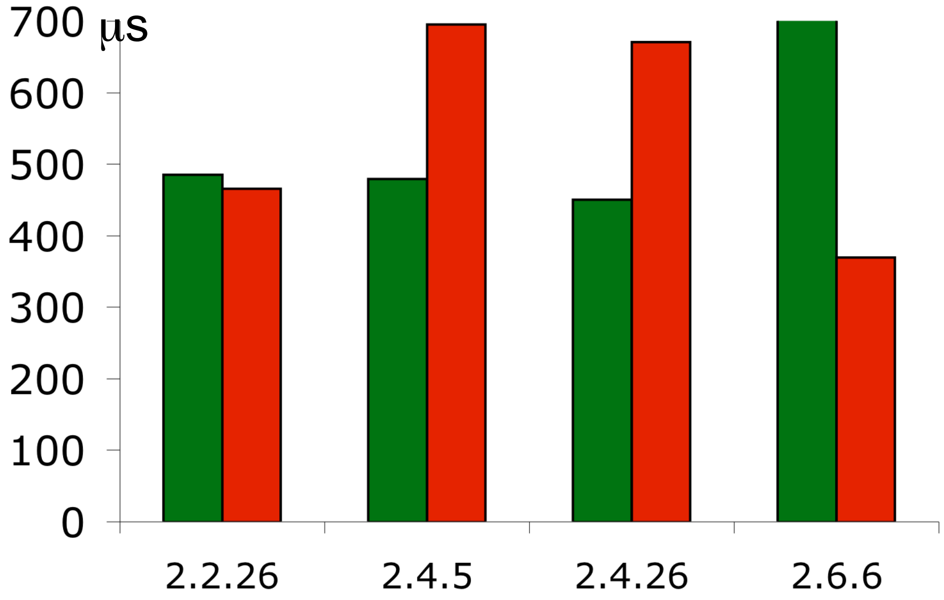


OS Reaction Time (WL = PostMark)

Windows



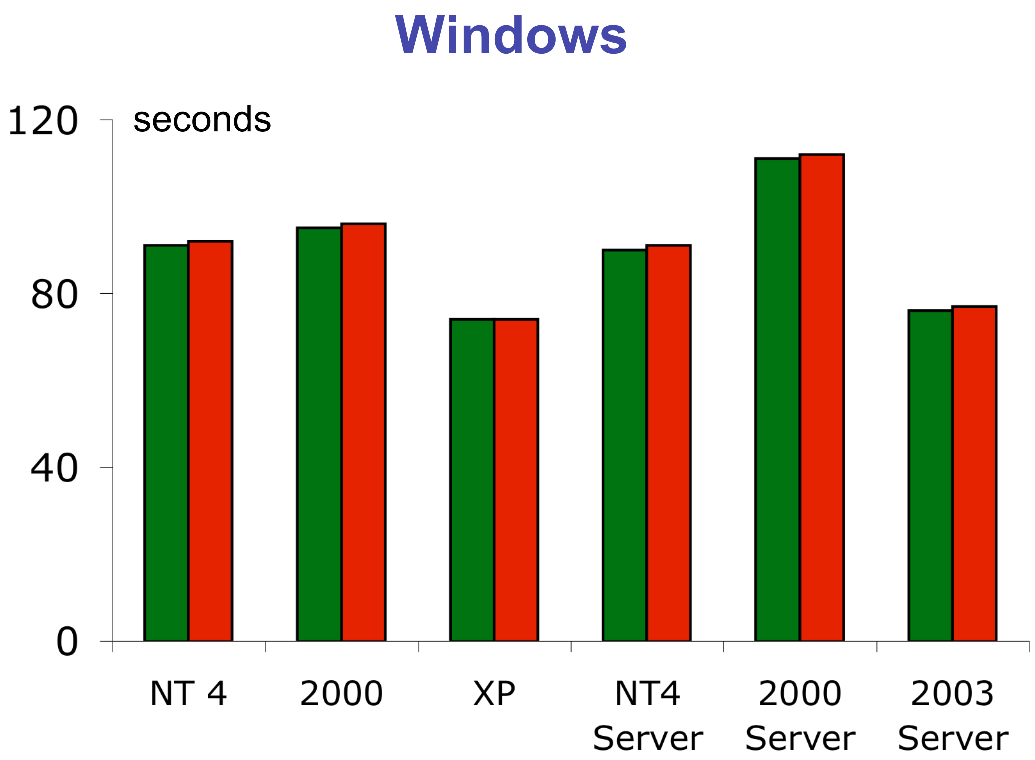
Linux

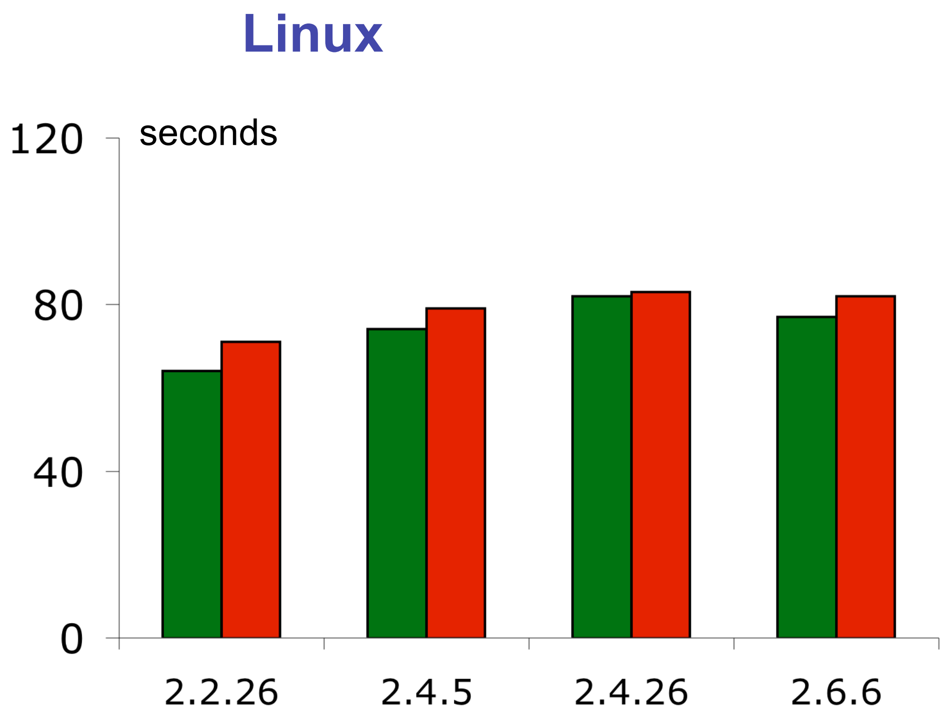


In the presence of faults

Without parameter corruption

Restart Time (WL = PostMark)

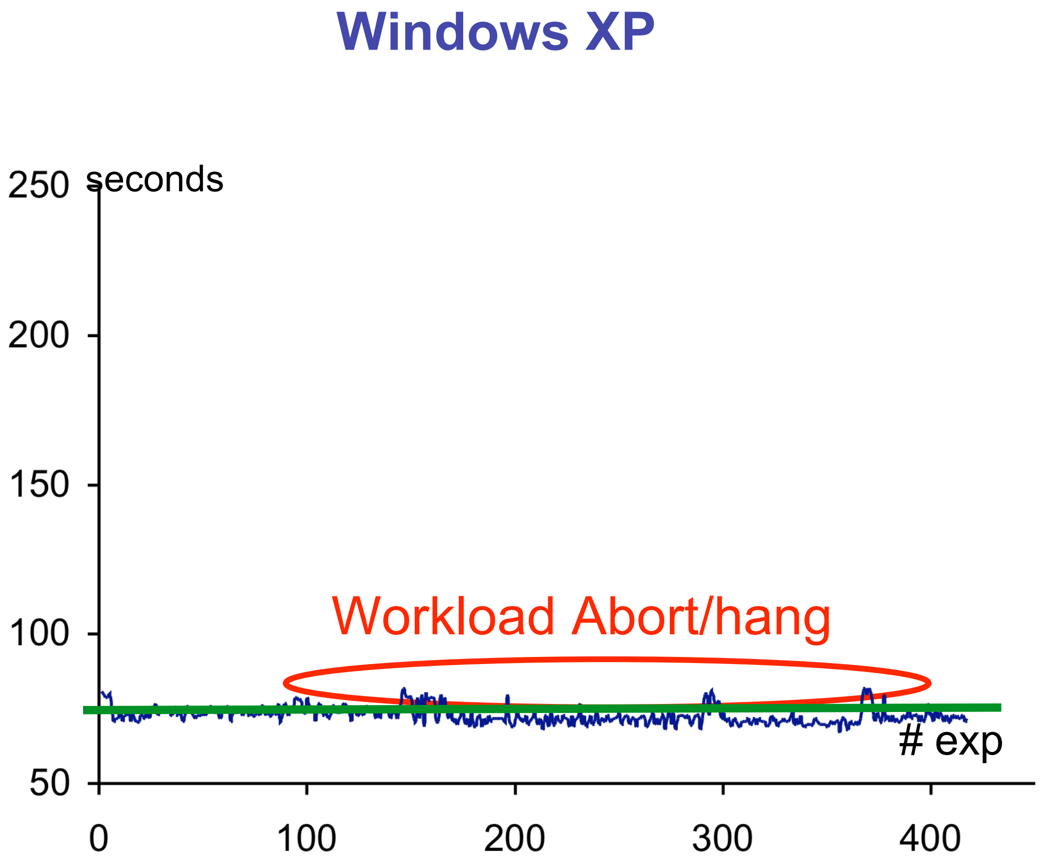


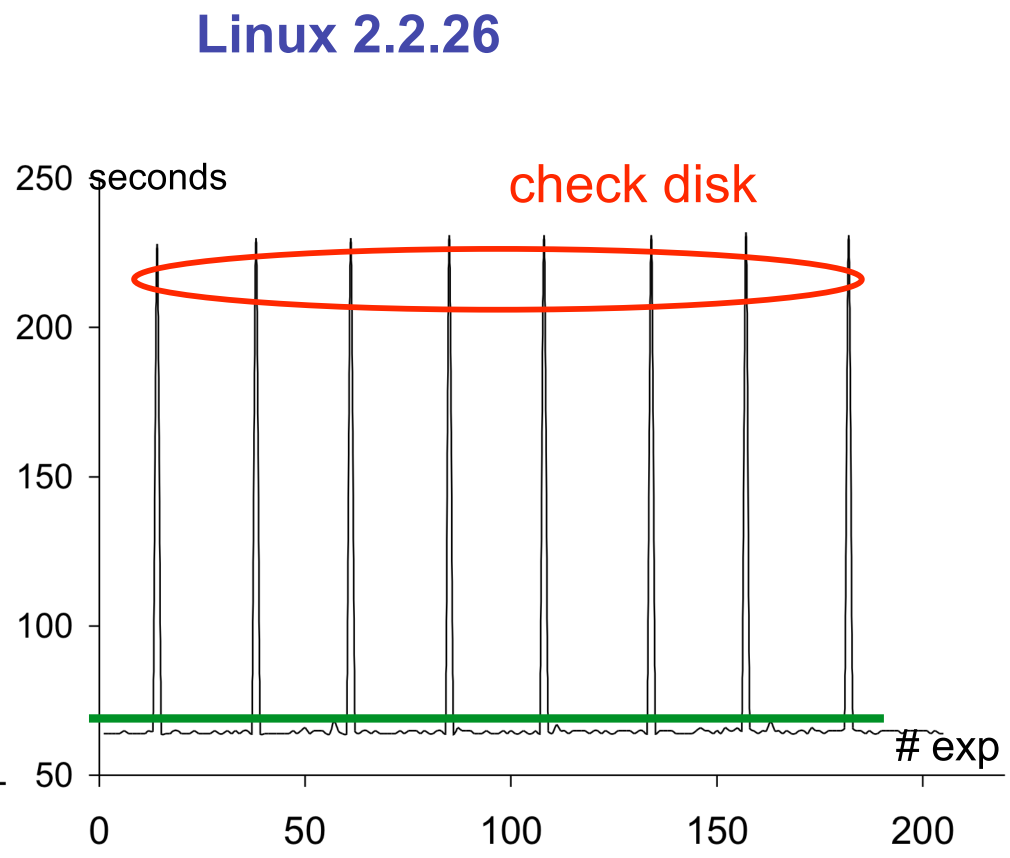


In the presence of faults

Without parameter corruption

Restart Time (WL = PostMark)





Validation of Properties

- Reproducibility
  - ✓ By construction
  - ✓ Set of faults
    - System Calls to be corrupted
    - Substitution values

- Repeatability
  - ✓ Each benchmark has been executed three times
    - Same robustness
    - Variation of the reaction time (< 4% for TPC-C client)
    - Variation of the restart time (< 3% for TPC-C client)

Validation - Sensitivity Analyses wrt Faultload

	Parameter corruption type			# experiments	
	Incorrect Data	Incorrect Address	Out-of-range Data	Windows NT4	Linux
Faultload 0	x	x	x	418	206
Faultload 1		x	x	331	135
Faultload 2			x	77	55

- Equivalence of versions of the same family
- Same comparison results between the two families

Additional analysis: incorrect data = out-of-range data in the context

Validation - Cost/effort

- Benchmark implementation duration

  For each OS family

  ✓ Implementation of the workload: 1 - 3 days

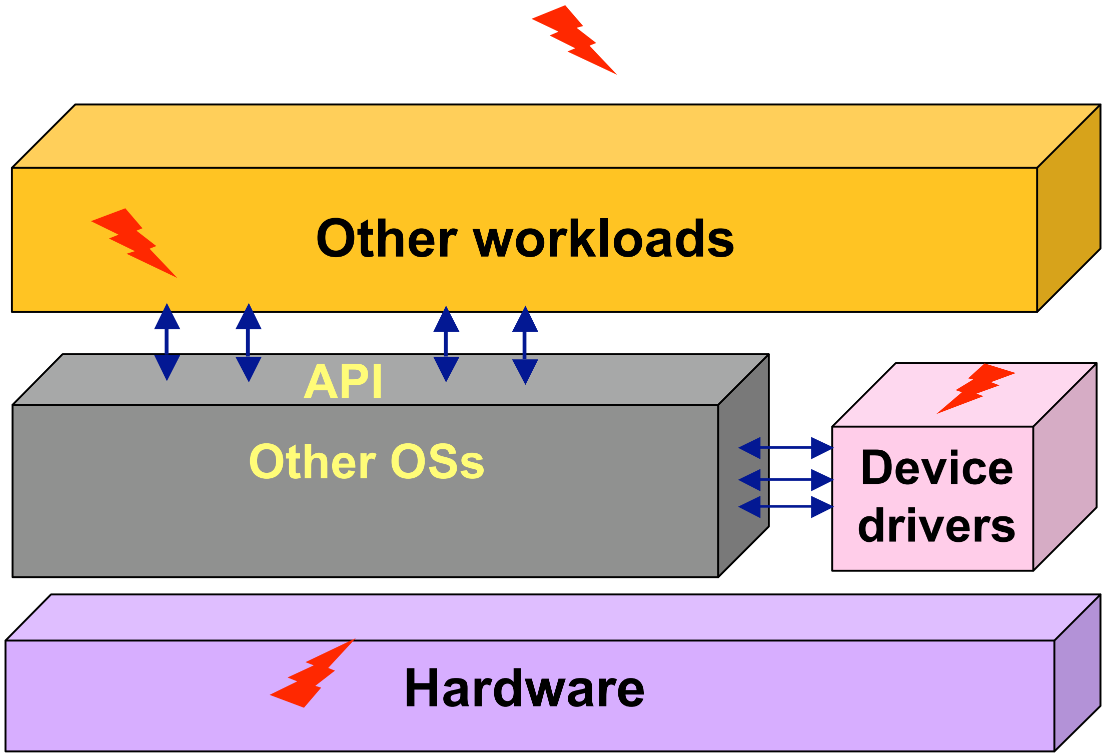
  ✓ Controller, parameter corruption and observation: 2 weeks

  ✓ Definition and implementation of fault set: 1 week

- Experiment duration

	Windows	Linux
TPC-C client	2 days	1 day
Postmark	2 days	1 day
JVM	4 days	2 days

Extensions



Extensions

- Application level:
  - Other workloads: **FreeBSD, OpenBSD**
  - Other faultloads stressful conditions/disturbances
  - Error propagation between applications
- Malicious faults
- Device drivers
  - Interface between OS and drivers
  - Impact of not well-debugged drivers
    - On the application
    - On other drivers
- Hardware
  - Simulation of exceptions due to the hardware fault occurrence
  - Transient faults
- Vender point of view?

References

- T. Jarboui, J. Arlat, Y. Crouzet, K. Kanoun and T. Marteau, “Impact of Internal and External Software Faults on the Linux Kernel”, IEICE Trans. on Information and Systems, vol. E86-D, no. 12, pp. 2571-2578, December 2003 — A preliminary version appeared in Proc. PRDC-2002, Tsukuba, Japan, 2002, pp. 51-58.
- A. Kalakech, T. Jarboui, J. Arlat, Y. Crouzet and K. Kanoun. “Benchmarking Operating Systems Dependability: Windows as a Case Study,” in Proc. 2004 Pacific Rim International Symposium on Dependable Computing (PRDC 2004), pp. 261-270, Papeete, Polynesia , 2-4 March 2004.
- A. Kalakech, K. Kanoun, Y. Crouzet and J. Arlat. “Benchmarking the Dependability of Windows NT, 2000 and XP,” in Proc. International Conference on Dependable Systems and Networks (DSN 2004), pp. 681-686, Florence, Italy, 2004.
- K. Kanoun, Y. Crouzet, A. Kalakech, A.-E. Rugina and Ph. Rumeau, “Benchmarking the Dependability of Windows and Linux using PostMark Workloads”, in Proc. 16th IEEE International Symposium on Software Reliability Engineering (ISSRE 2005), pp. 11-20, Chicago, IL, USA — November 8, 2005