Power, Performance and Reliability Management

Kishor S. Trivedi and Paulo Maciel

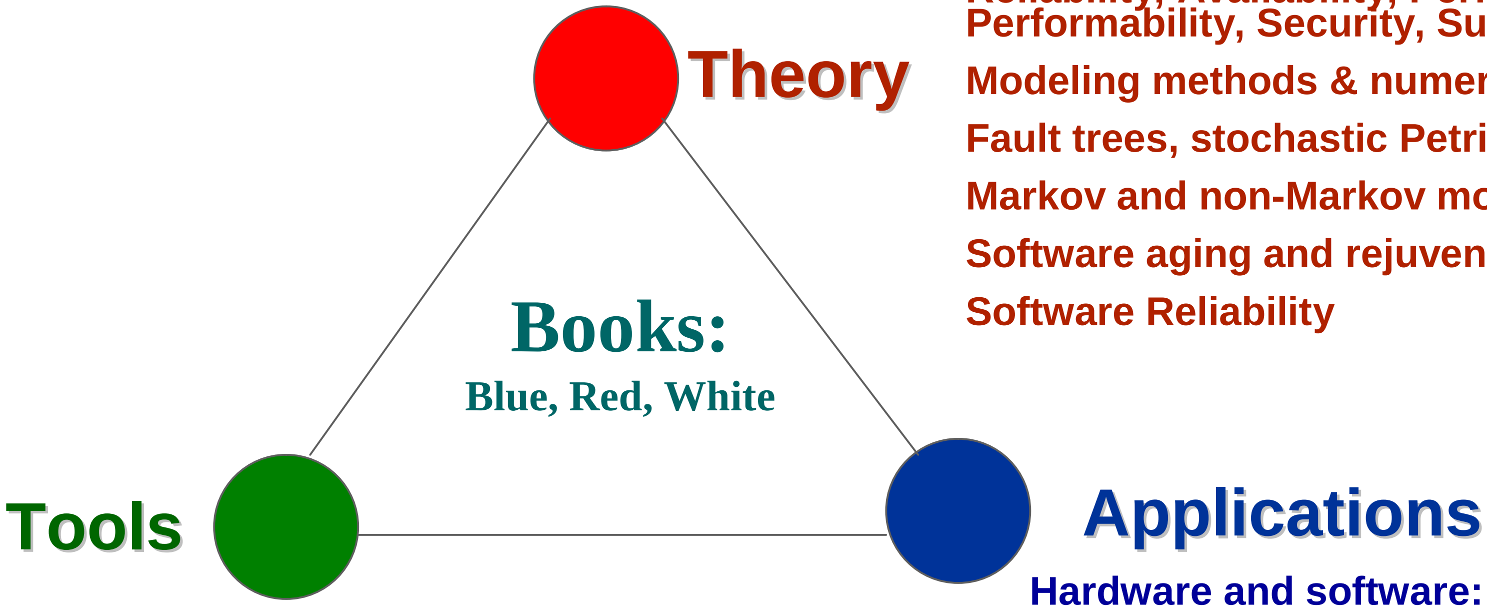
Duke University
Research Triangle Park, NC

Universidade Federal de Pernambuco
Recife, Brasil

Outline

- Research at Duke
- Research at UFPE
- Problem definition
- Overview of the proposed approach
- Energy evaluation applied to embedded systems
- Potential use of data mining and machine learning techniques

Trivedi's Research Triangle



Theory

Reliability, Availability, Performance, Performability, Security, Survivability

Modeling methods & numerical solution:

Fault trees, stochastic Petri Nets,

Markov and non-Markov models

Software aging and rejuvenation

Software Reliability

Books: Blue, Red, White

Tools

HARP,

SAVE,

SHARPE,

SPNP,

SREPT

Applications

Hardware and software:

Boeing, EMC, HP (Peter Piet), Ericsson, IBM, Sun, Cisco, GE, 3Com, Lucent, Motorola,…

Research at Duke

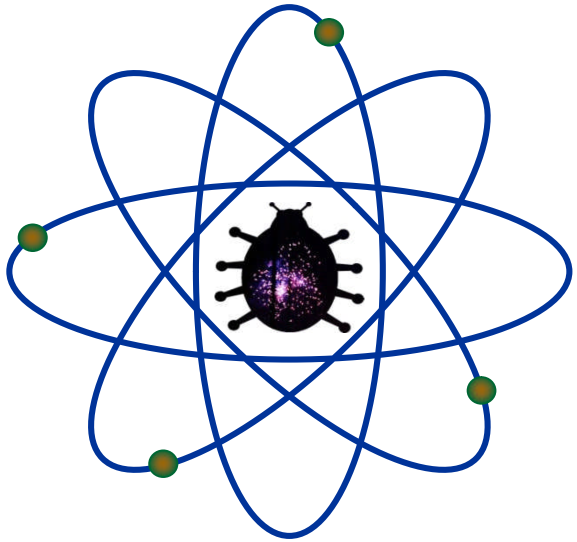
- We have aided many companies in computer, telecommunications and aerospace industries, e.g.:
  - Boeing
    - Reliability Analysis of CRN subsystem in Boeing 787 using our algorithm and our software package, SHARPE; for FAA certification
    - Boeing Integrated Reliability Analysis package built with our help and contains our tools, HARP, SHARPE, SPNP
  - IBM
    - Helped implementation of software rejuvenation in IBM X-series
    - Reliability and Availability analysis of SIP on HA WebSphere (was responsible for the sale to a major Telco customer)
  - NASA/JPL Software fault classification from Satellite problem reports
  - HP: Worked with Peter Piet, Rudy Gomez and Linda Peckham LaMarca in availability modeling and use of SHARPE, SPNP packages
- 2008 IEEE Technical Achievement Award for our work on Software Aging and Rejuvenation; first workshop on this theme on Nov 11 in Seattle, part of ISSRE

Current Research at Duke

- Software Reliability/Availability/Performance
  - Fault Classification and Mitigation Techniques
  - Software Aging and Software Rejuvenation
  - Architecture-Based Software Performance, Reliability and Availability Assessment and Optimization
  - Holistic Approach using
    - Measurements: controlled (fault injection experiments; software aging data) and operational (problem reports from the field; software aging data)
    - Structural and Empirical Stochastic Models
    - Optimization of test resource scheduling, rejuvenation scheduling, recovery sequencing

Fault Classification: Bohrbug

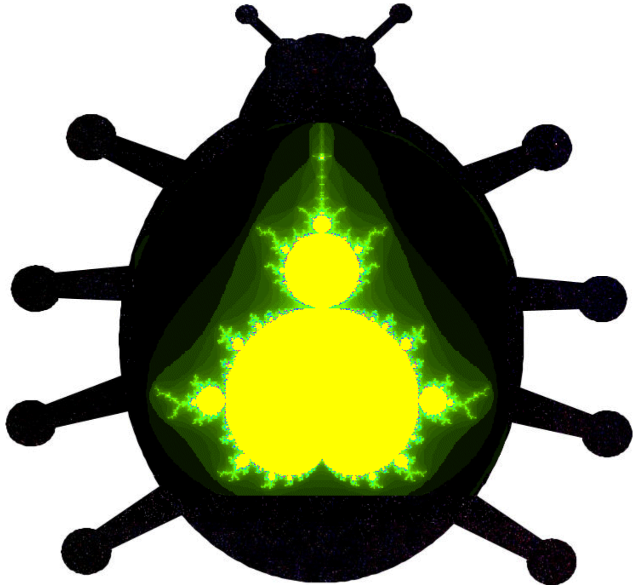
- Fighting bugs: Remove, retry, replicate, and rejuvenate. IEEE Computer 40(2). Michael Grottke & Kishor Trivedi.
- **Bohrbug :=** A fault that is easily isolated and that manifests consistently under a well-defined set of conditions, because its activation and error propagation lack complexity.



- Example: A bug causing a failure whenever the user enters a negative date of birth
- Since they are easily found, Bohrbugs tend to be detected and fixed during the software testing phase.
- The term alludes to the physicist Niels Bohr and his rather simple atomic model.

Fault Classification: Mandelbug

- **Mandelbug :=** A fault whose activation and/or error propagation are complex. Typically, a Mandelbug is difficult to isolate, and/or the failures caused by it are not systematically reproducible.
- Example: A bug whose activation is scheduling-dependent



- The residual faults in a thoroughly-tested piece of software are mainly Mandelbugs.
- The term alludes to the mathematician Benoît Mandelbrot and his research in fractal geometry.

Fault Classification: Aging-related Bug

- **Aging-related bug :=** A fault that leads to the accumulation of errors either inside the running application or in its system-internal environment, resulting in an increased failure rate and/or degraded performance.

- Example:
  - A bug causing memory leaks in the application
- The activation rate of the fault is influenced by the total time for which the system has been continuously running or even the workload variation.
- Note that the aging phenomenon requires a delay between fault activation and failure occurrence.
- Note also that the software appears to age due to such a bug; there is no physical deterioration

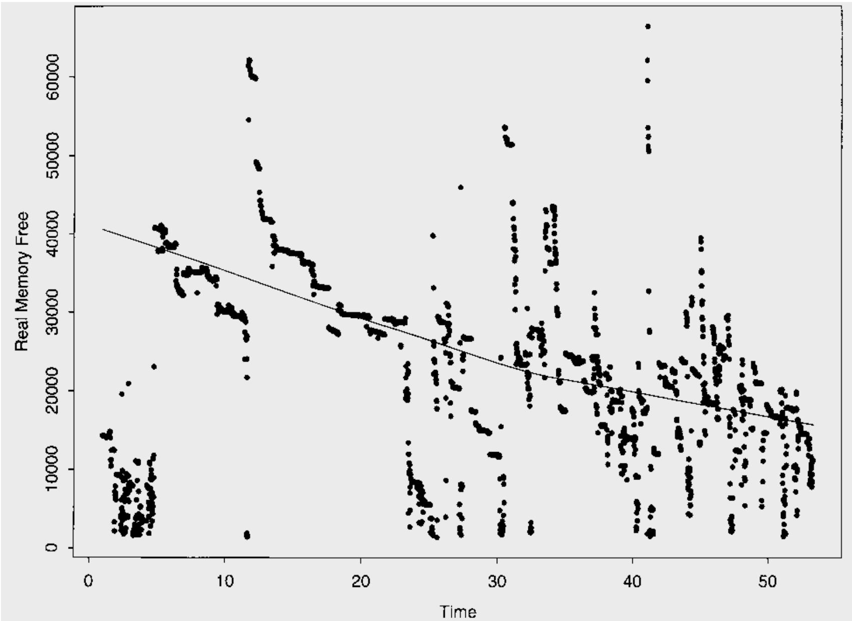
Fault Classification for Space Mission System Software; Michael Grottke, Allen Nikora & Kishor Trivedi

ID	L order	duration	Fault type proportions			
			BOH	NAM	ARB	UNK
1	10	0.705	0.000	1.000	0.000	0.000
2	4	0.911	0.571	0.379	0.043	0.007
3	17	0.226	0.815	0.130	0.019	0.037
4	14	0.388	0.429	0.429	0.143	0.000
5	1	1.318	0.500	0.000	0.000	0.500
6	12	0.292	0.810	0.143	0.048	0.000
7	13	0.519	0.231	0.538	0.231	0.000
8	5	0.074	0.000	0.500	0.500	0.000
9	15	0.376	0.522	0.435	0.000	0.043
10	3	1.000	0.595	0.270	0.135	0.000
11	11	0.582	0.554	0.369	0.062	0.015
12	6	0.087	0.857	0.143	0.000	0.000
13	9	0.706	0.667	0.333	0.000	0.000
14	18	0.171	0.643	0.343	0.014	0.000
15	8	0.753	0.000	0.500	0.500	0.000
16	7	0.657	0.481	0.481	0.000	0.037
17	16	0.272	-	-	-	-
18	2	1.246	1.000	0.000	0.000	0.000
Average proportions			**0.510**	**0.382**	**0.070**	**0.038**

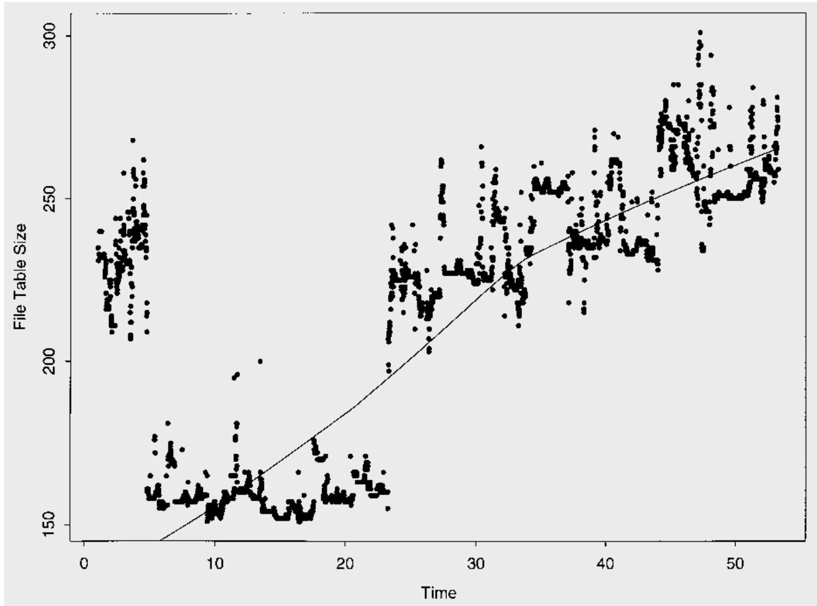
BOH: Bohrbugs; NAM: non-aging-related Mandelbugs; ARB: aging-related bugs; UNK: unknown
This was done manually; in the future we plan to use data mining techniques

Prediction of Time to Resource Exhaustion or Time to Software Aging Related Failure

Real Memory Free

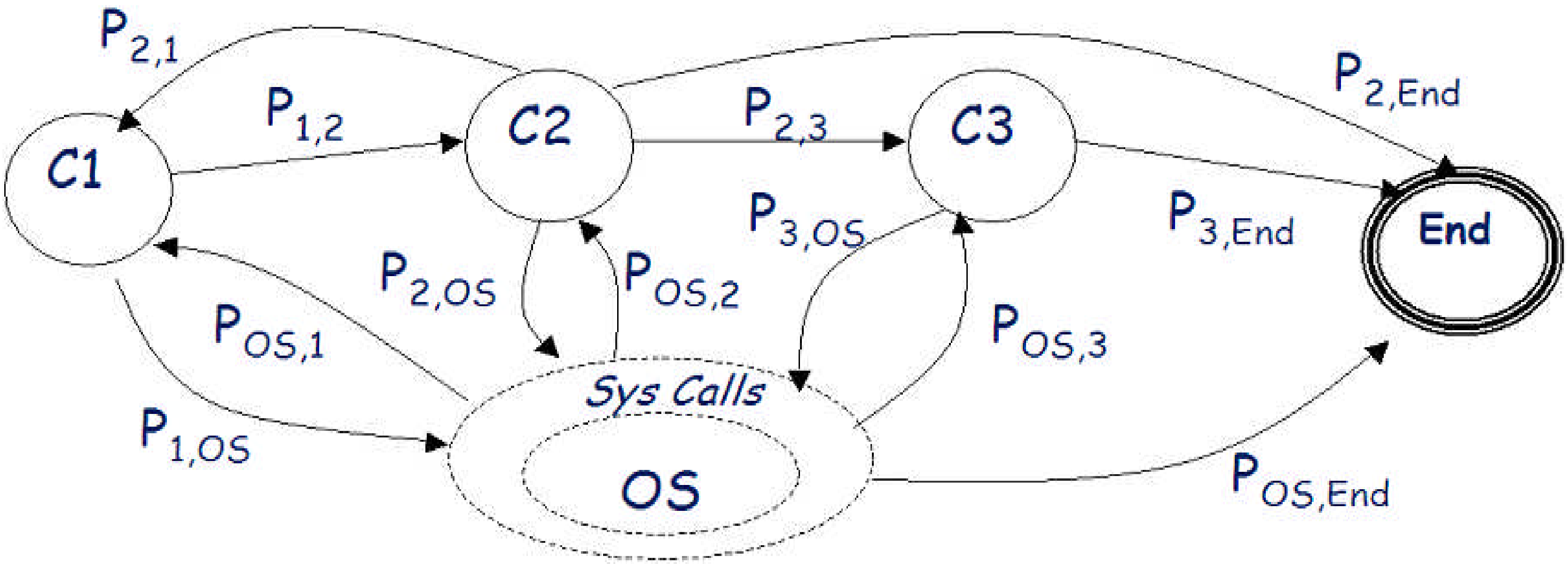


File Table Size



Prediction methods for Rejuvenation scheduling: **Linear & non-linear statistical methods, Pattern recognition, Neural nets, Optimization (off-line and on-line)**

Architecture-Based Software Reliability and Optimization: European Space Agency Example



Minimize the total test effort $T = \sum f(\lambda_i)$

subject to a reliability constraint

Software Reliability and Testing Time Allocation: An Architecture-Based Approach, Roberto Pietrantuono, Stefano Russo & Kishor Trivedi,

European Space Agency Example

TABLE III
ESTIMATED PARAMTER VALUES

Minimum Required Reliability					0.99
Reliability of the Previous Version					0.93583
Mean User Process Execution Time					0.01389
Mean OS Execution Time					0.10589
Mean # User Calls					407.4
Mean # System Calls					5442
Transition Probabilities					
TO / FROM	1	2	3	OS	End
1	0.237	2.71E-4	0	0.7689	7.79E-5
2	0	0.625	0.029	0.3439	4.96E-4
3	0	0	7.72E/4	0.9902	2.10E-4
OS	0.4049	0.0141	0.5810	0	0
END	0	0	0	0	1
Visit Counts	2865.8	222.8	3165.0	5442.3	-
Exec. Time	0.01128	0.00248	1.251E-4	0.10589	-
SRGM: Exponential SRGM. $\lambda(t) = age^{-gt}$					
1		2		3	
a	g	a	g	a	g
13	5.46E-2	11	9.46E-2	5	5.34E-2

European Space Agency Example

TABLE IV
TESTING OF THE SYSTEM ACCORDING TO THE MODEL RESULTS

Componenent Componenent	#Fault Removed	Testing Time	#Test Cases
1	11	40.638	348
2	5	6.315	58
3	3	24.682	215
Fault Number		Detection/Removing Time	Test Case Number
Component 1			
21		1.213	10
22		3.112	25
23		6.452	53
24		6.992	57
25		8.346	69
26		12.240	102
27		13.332	111
28		15.341	128
29		22.021	183
30		30.041	250
31		38.098 of 40.638	318 of 348
Component 2			
51		0.933	7
52		2.729	22
53		2.981	24
54		4.065	33
55		6.209 of 6.315	51 of 58
Component 3			
69		5.034	42
70		7.355	61
71		20.442 of 24.682	170 of 215

The reliability predicted by the model is 0.990289. Measured reliability (as 1 – $\lim_n n_f/n$), is 0.989722, with n_f = 37 observed failures over n = 3600 executions. The relative error is 0.057 %.

Research at UFPE

- Real-Time Power-Aware Embedded Systems Evaluation and Design
  - real-time power-aware automatic code generation.
  - power management methods.
  - and power and performance estimation:
    - Measurement based and
    - stochastic model based evaluation for supporting system design and tuning.

Research at UFPE

- Performance Evaluation:
  - Study of process, methods, models for performance evaluation, capacity planning and process optimization.
  - Projects:
    - modeling and performance evaluation of manufacturing process.
    - modeling and performance evaluation for tailoring of software processes.
    - performance evaluation of Electronic Funds Transfer (EFT) Systems.
    - database server tuning, distributed web-services evaluation.
    - synthetic workload generation

Research at UFPE

- Performance Evaluation:
  - **Projects GCAP**
    - synthetic workload generator for **HP Capacity Advisor**.
    - aims to provide reliable and more flexible means for generation of significant **workloads scenarios based on real traces or even on statistical summaries of real traces** .
    - the project aim is to expand the Capacity Advisor functionalities **to allow capacity planning even when reduced information is available**.
  - **Contact person at HP: José Paulo Pires, Porto Alegre, Brazil,**
    e-mail: jpaulo.pires@hp.com

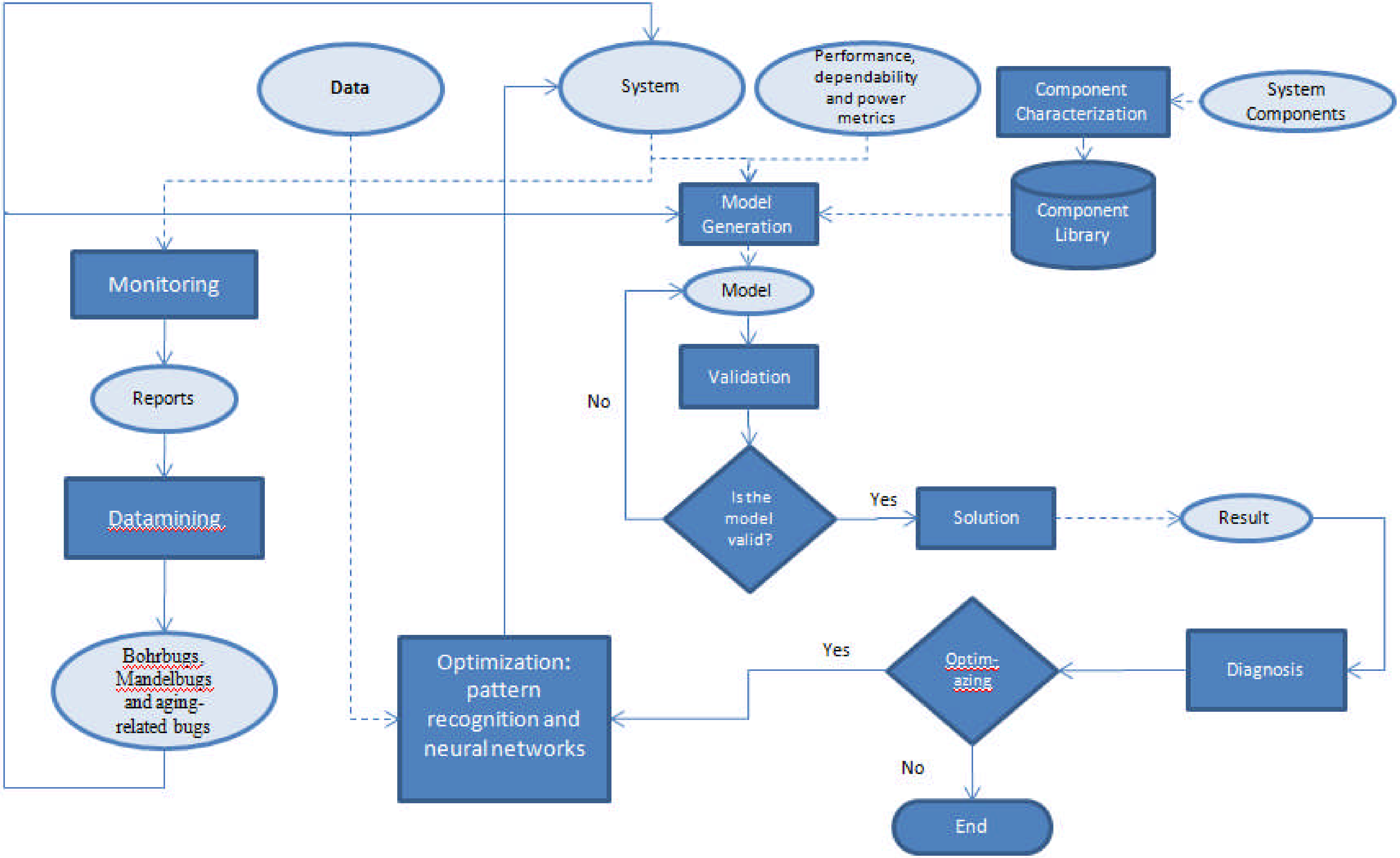
Research at UFPE

- Performance, Reliability and Availability Evaluation:
  - Study of process, methods, models for performance evaluation and capacity planning considering dependability, and repairing issues
  - Projects:
    - availability and reliability evaluation of services in electrical generation and transmission companies.
    - performance modeling and evaluation of automation system considering availability service level agreements.

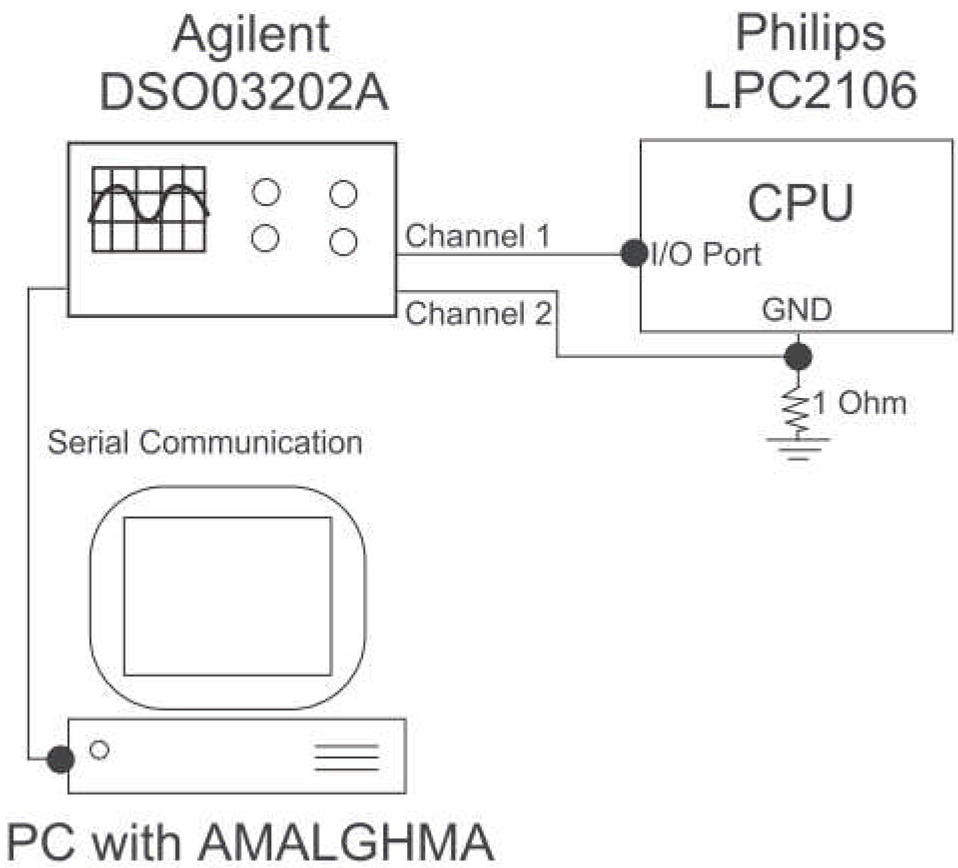
Proposal

- Combine the efforts at UFPE and Duke
- Measurements, Models and Optimization in Architecture-based reliability, performance and power management
- Data mining (possible tools: WEKA and TnT; these tools are being used by our co-investigator, Allen Nikora at JPL) techniques to examine problem reports for the classification of software faults into Bohrbugs, non-aging-related Mandelbugs and aging-related bugs
- Use of pattern recognition (being used at SUN Microsystems by former student Kalyan Vaidyanathan), hidden Markov models (being used by Felix Salfner at Humbolt University) and neural networks (Hisham El-Shishiney at IBM Cairo has used this on the data that we supplied to him)

Overview of the proposed approach

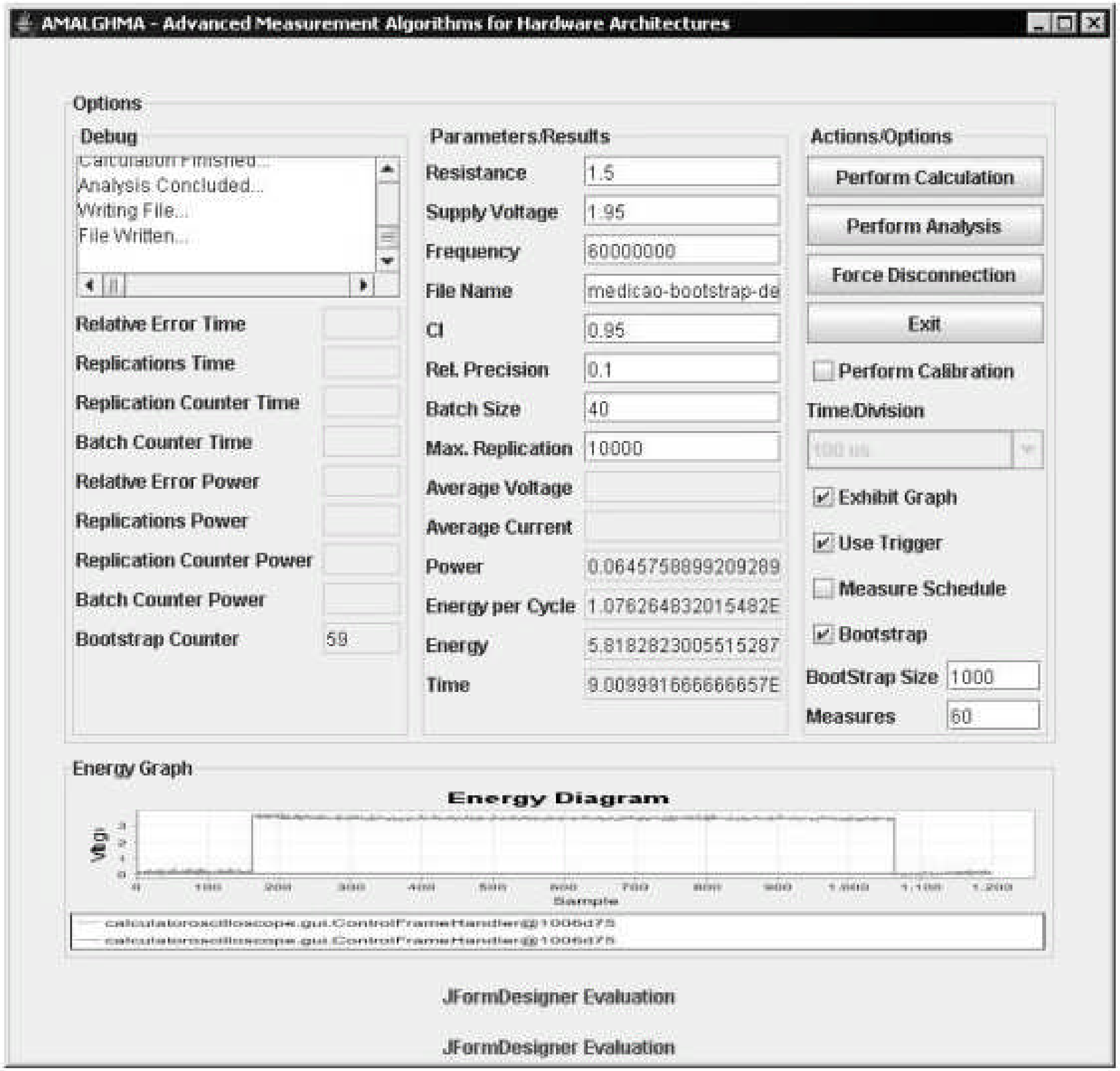


Energy and performance measurements of embedded systems



Energy and performance measurements of embedded systems

AMALGHMA Tool



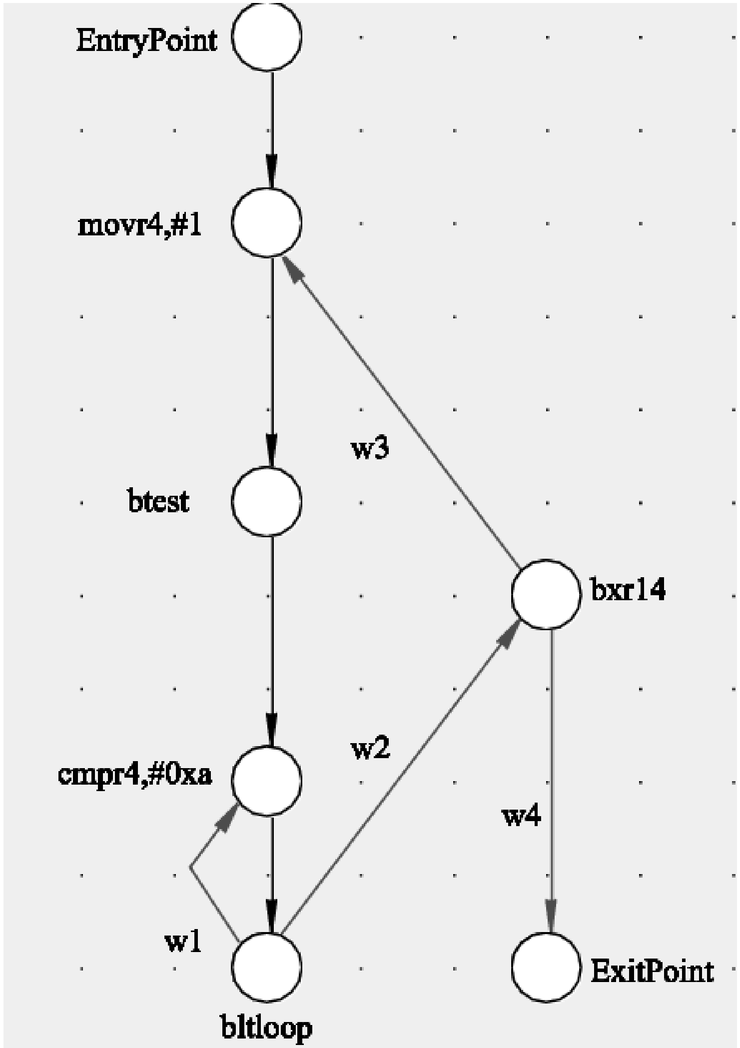
Instruction set characterization

	ARM7 Instructions (MAM off)	TIME(μS)	ENERGY(ηJ)
Ordinay Instructions	ADD , EOR, ORR, AND, SUB, MOV, MVN, CMP, RSB	0.0673	3.78
	...		
Load / Store Instructions	STR3	0.085	6.11
	STR1	0.085	5.95
	LDR1	0.102	5.95
	LDR3	0.152	9.01
	LDRB	0.102	6.273
	...		
Multip. Instructions	MLA	0.1	6.21
	MUL	0.083	5.05
	UMUL	0.05	4.16
	...		
Branch Instructions	B	0.201	12.2
	Bcond(FALSE)	0.0673	3.78
	BX	0.201	11.4

	8051 Instructions	CYCLES	ENERGY
Ordinary Instructions	XCHD, swap, sub, setb, rrc_a, rr_a, orl, nop, MOV,	12	47.479
	INC, DEC, DA, CPL, ANL, ADDC, ADD		
Branch Instructions	SJMP, RET, PUSH, POP, ljmp, LCALL, JNZ, JNB,	24	105.45
	JC, CJNE, ACALL		
Multip. Instructions	MUL_AB	48	189.916

A glance at a model

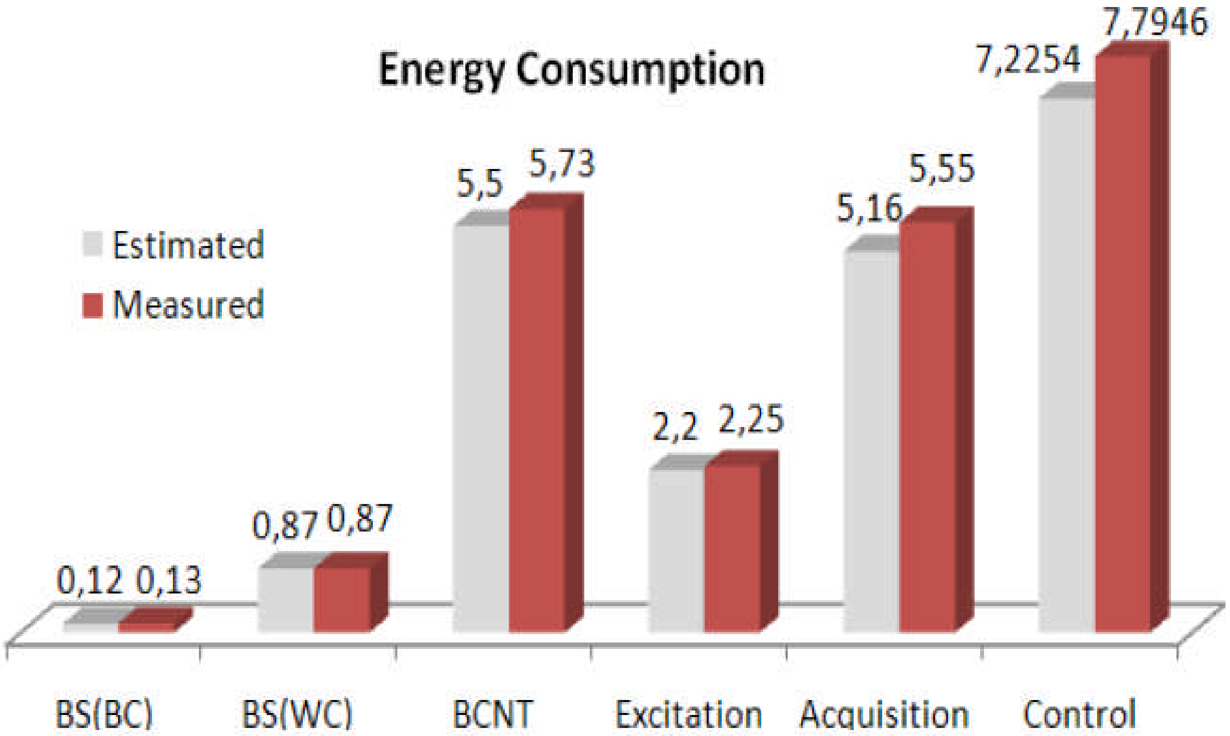
```
.
.
.

   for mov r4,#1
          b test
   loop add r4,r4,#1
   test cmp r4,#0xa
          blt loop ;<@0.9@>
          bx r14
```



- Annotated Assembly or C code
- Basic blocks

IE1

Summary of experiments



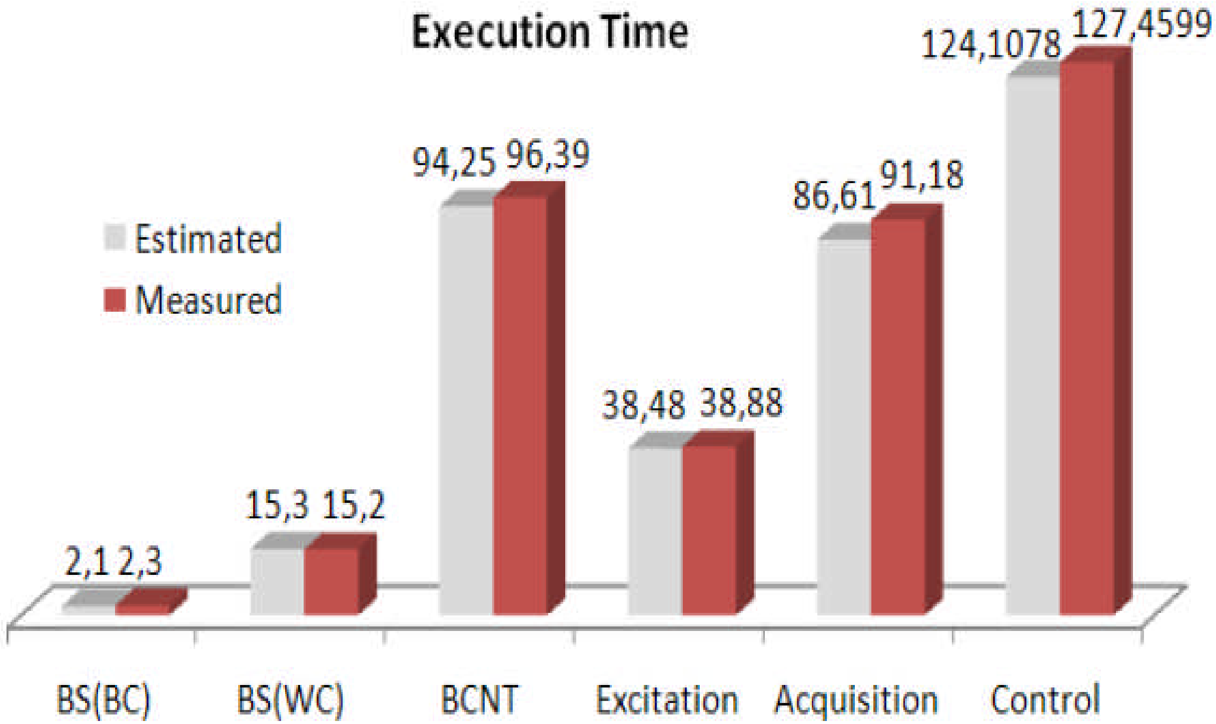
Case Study	Estimated Time(μs)	Estimated Energy(μJ)	Hardware Time(μs)	Hardware Energy(μJ)
1. Binary Search(BC)	2,1	0,12	2,3	0,13
2. Binary Search(WC)	15,3	0,87	15,2	0,87
3. BCNT	94,25	5,50	96,39	5,73
4. Excitation	38,48	2,20	38,88	2,25
5. Acquisition	86,61	5,16	91,18	5,55
6. Control	12410,78	722,54	12745,99	779,46

Slide 24

IE1 Paulo
Please replace commas by periods in the table and the figures
IBM Employee, 10/16/2008

IE3

Summary of experiments



Case Study	Estimated Time(μs)	Estimated Energy(μJ)	Hardware Time(μs)	Hardware Energy(μJ)
1. Binary Search(BC)	2,1	0,12	2,3	0,13
2. Binary Search(WC)	15,3	0,87	15,2	0,87
3. BCNT	94,25	5,50	96,39	5,73
4. Excitation	38,48	2,20	38,88	2,25
5. Acquisition	86,61	5,16	91,18	5,55
6. Control	12410,78	722,54	12745,99	779,46

Slide 25

IE3 Paulo
Please replace commas by periods in the table and the figures
IBM Employee, 10/16/2008

Expected results

- provide means for trading-off:
  - infrastructure costs.
  - energy consumption.
  - availability, reliability, survivability and
  - performance.

More Research Opportunities

- Sustainability-related areas
  - Software aging increases power consumption
    - e.g., memory fragmentation and memory leak cause more power consumption per physical memory surface usage.
    - software rejuvenation through process restart or warm-reboot use less energy than non-planned system crash followed by cold reboots.
  - Power-aware workload balance protocol
    - Distribute the workload across cluster nodes taking into account minimization of the total energy consumption.
  - Power-aware demonstration tests for selecting software COTS
    - DOE applied to evaluate the influence of COTS in the system energy consumption.

Question and Comments

- Thanks