Energy/Reliability Trade-offs in Fault-Tolerant Event-Triggered Distributed Embedded Systems

Junhe Gan[1], Flavius Gruian[2], Paul Pop[1], Jan Madsen[1]

[1] Technical University of Denmark, Denmark

[2] Lund University, Sweden

DTU Informatics

Department of Informatics and Mathematical Modeling

Motivation

- Newer technologies are leading to:
  - Higher power density
  - Increased likelihood of transient faults

- A lot of research on **energy/performance trade-offs** using DVFS:
  - Scaling voltage and frequency can reduce the energy consumption, but it prolongs the tasks execution time.

- We focus on **energy/reliability trade-offs** in this paper:
  - Scaling voltage and frequency can reduce the energy consumption, but it increases the number of transient faults exponentially.
  - This is orthogonal to peak temperature minimization used to increase the life-time of a system.

Outline

- **System Models and Schedulability Analysis**
  - Architecture model
  - Reliability model
  - Application model
  - Schedulability analysis
  - Power model
  - Energy/reliability trade-off model
  - **Problem Formulation**
  - Motivational Example
- **Optimization Strategy**
  - TABU search-based algorithm
- **Experimental Results**
- **Conclusion and Contributions**

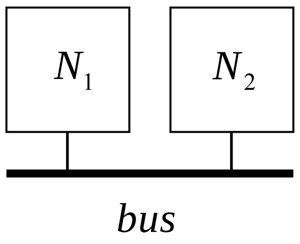
Architecture Model

- A set of heterogeneous processing elements interconnected by a communication channel

- Each processing element might have a set of operating modes

- For each operating mode we know

  (frequency: $f_i^{N_j}$, voltage: $v_i^{N_j}$, power dissipation: $p_i^{N_j}$)

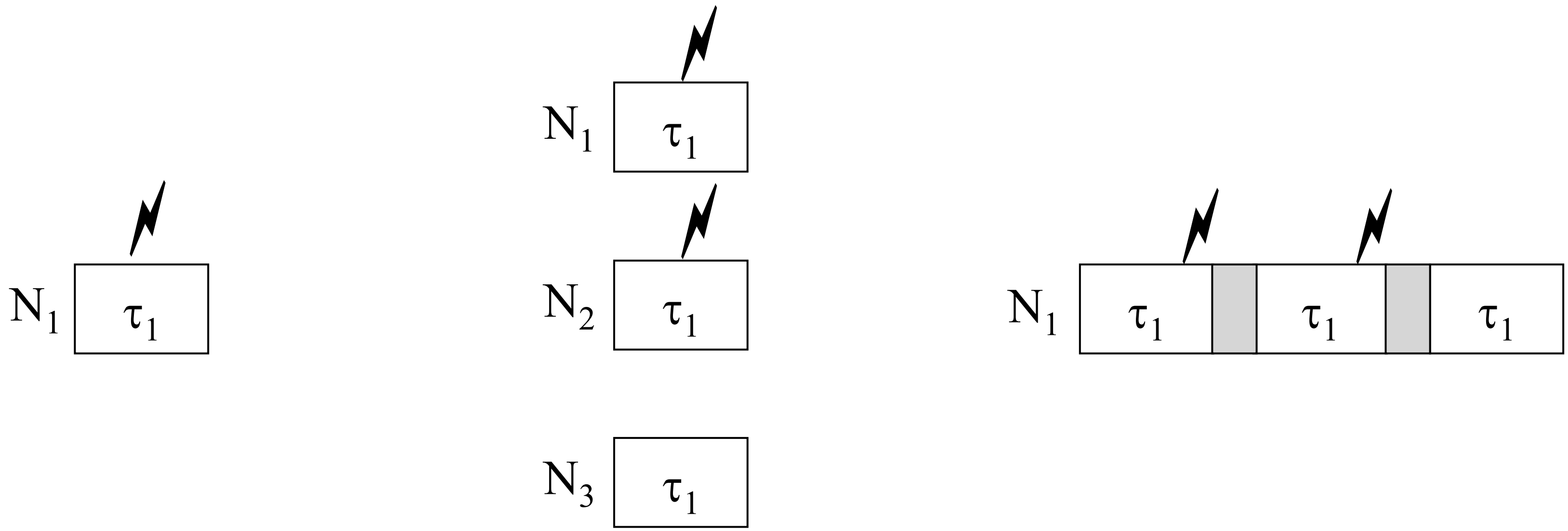


	N_1			N_2		
Operating Modes	Freq. [MHz]	Volt. [V]	Power [W]	Freq. [MHz]	Volt. [V]	Power [W]
1	333	1.2	4	166	1.1	2
2	666	1.4	12	333	1.25	4.5
3	1000	1.6	25	500	1.5	11

Reliability Model

Fault-tolerance: $k_1 = 2$

No fault-tolerance | **Replication** | Re-execution



$$R_i = e^{-\lambda c}$$

$$R_i^{rep} = 1 - \prod_{i=1}^{k} (1 - R_i)$$

$$R_i^{reex} = 1 - (1 - R_i)^{1+k}$$

- In this paper, we use $R_s = \prod_{i=1}^{|\Gamma|} R_i^{rep}$

Application Model

- **Application Model:**
  - A set of periodic tasks
  - For each task τ_i , we know
    - $(C_i^{N_j}, T_i, D_i)$
    - Unique priority
    - Number of replicas k_i (critical task: $k_i > 0$, non-critical task: $k_i = 0$)
  - Reliability goal R_g.
    - If reliability is lower, the no. of replicas is not enough to tolerate the faults.

Γ'	$C_i^{N_1}$	$C_i^{N_2}$	$T_i = D_i$	Priority
τ_1	7	14	50	1
τ_2	6	12	100	4
τ_3	5	10	50	2
τ_4	8	16	100	5
τ_1'	7	14	50	3
τ_2'	6	12	100	6

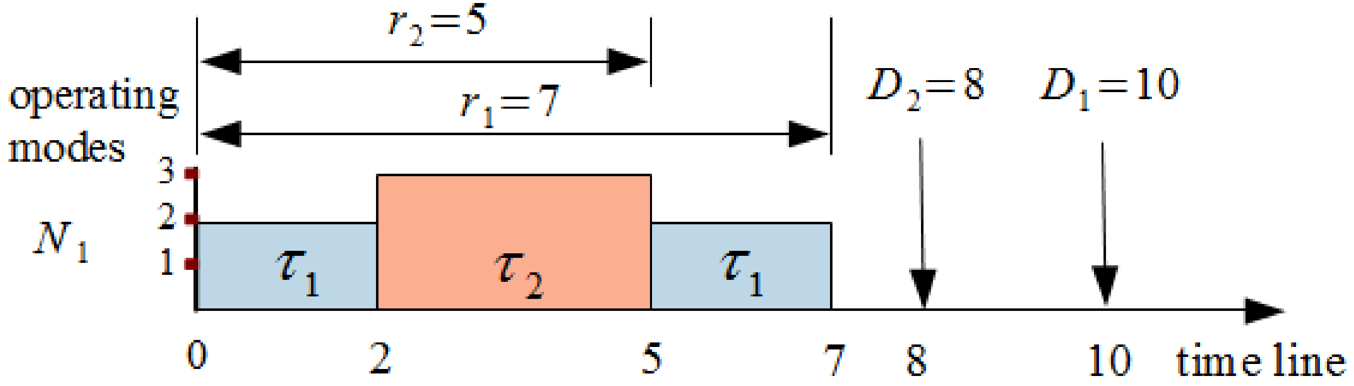
Schedulability Analysis

- Tasks are scheduled by **fixed-priority preemptive scheduling**.

- We use **response time analysis** to calculate the worst-case response time r_i for each task.

- We use the **degree of schedulability** r_S to measure which design alternative is "more schedulable".

$$c_1^{l=2} = 4, c_2^{l=3} = 3$$



Energy Model

Energy consumption in **power-aware processing elements**

$$E_s = \sum_{\tau_i \in \Gamma} \left\lceil \frac{T_\Gamma}{T_i} \right\rceil \times p_l^{N_j} \times c_i^l + O$$

$\left\lceil \frac{T_\Gamma}{T_i} \right\rceil$: the number of τ_i's jobs within the application period T_Γ

$p_l^{N_j} \times c_i^l$: the energy consumption of τ_i

O : the sum of mode switching overheads

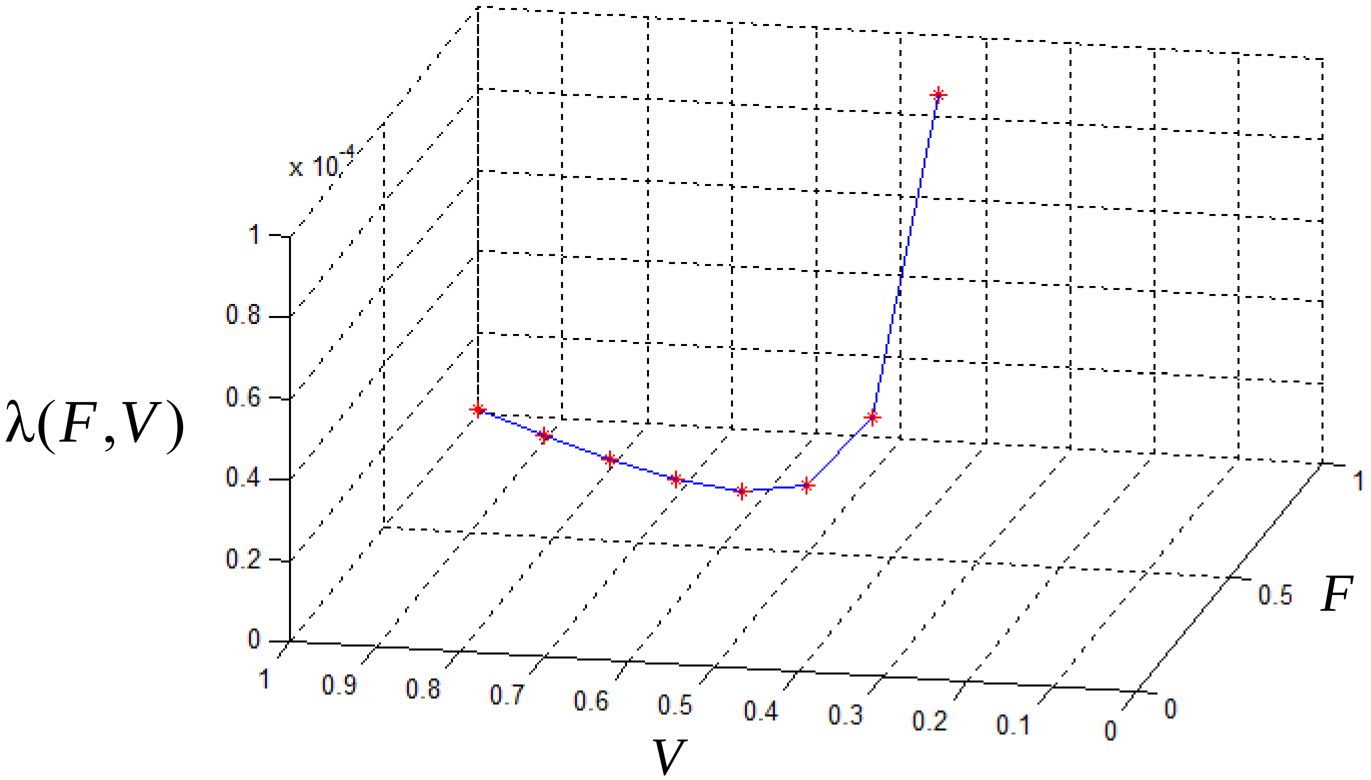
Energy model is from: E. Bini, G. Buttazzo and G. Lipari, "Minimizing CPU energy in realtime systems with discrete speed management", Embedded Computing Systems, 31(8), 2009.

Energy/reliability Trade-off Model

The fault rate λ increases exponentially when

the normalized voltage V and the normalized frequency F decreases

$$\lambda(F,V) = \lambda_0 \cdot F^{\alpha} \cdot 10^{-\beta V}$$



The equation is adapted from: D. Zhu and H. Aydin, "Reliability-Aware Energy Management for Periodic Real-Time Tasks", IEEE Transactions on Computers, 58(10), pp. 1382 - 1397, 2009.

Problem Formulation

- **Given:**
  - Application and architecture models
  - Reliability goal and corresponding number of replicas for each task

- **Determine offline:**
  - the mapping of each task to processing element
  - the operating mode for executing each task

- **Such that:**
  - all tasks **meet** their **timing requirements**
  - the application reliability **meets** the given **reliability goal**
  - the **energy consumption** of the system **is minimized**

Motivational Example

- Application and architecture

Γ'	$C_i^{N_1}$	$C_i^{N_2}$	$T_i = D_i$	Priority
τ_1	7	14	50	1
τ_2	6	12	100	4
τ_3	5	10	50	2
τ_4	8	16	100	5
τ'_1	7	14	50	3
τ'_2	6	12	100	6

	N_1			N_2		
Operating Modes	Freq. [MHz]	Volt. [V]	Power [W]	Freq. [MHz]	Volt. [V]	Power [W]
1	333	1.2	4	166	1.1	2
2	666	1.4	12	333	1.25	4.5
3	1000	1.6	25	500	1.5	11
	$\lambda_0 = 10^{-6}, \alpha = -4, \beta = -0.04$					

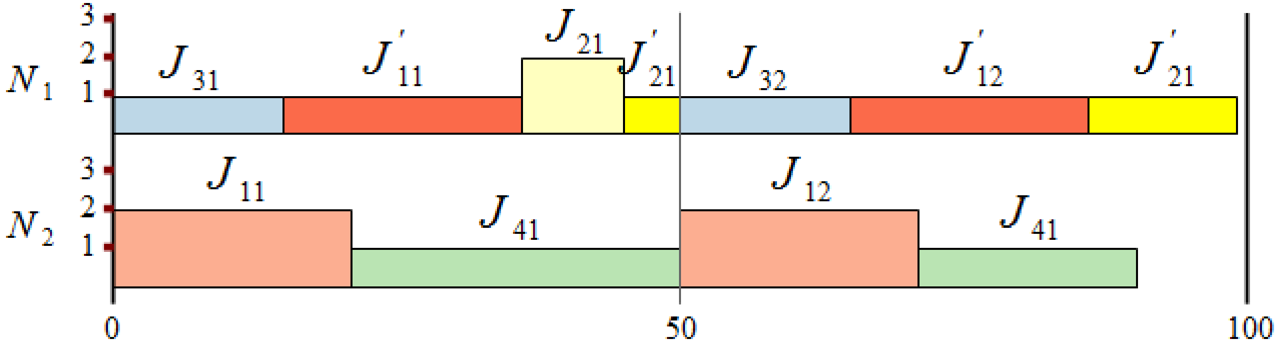
- Initial solution: no voltage and frequency scaling
  - Runs all the tasks in the maximum speed operating mode and maps the tasks on the low power PEs.
  - E_0=1312, R_s^0=0.999996.

- The given reliability goal: $R_g = 1 - 10(1 - R_s^0) = 0.99996$

  which means that we accept at most a 10 times decrease in reliability.

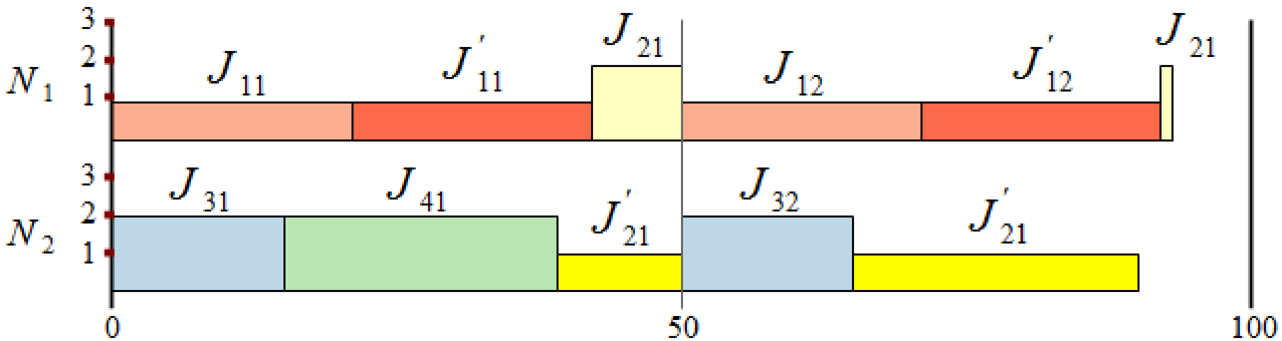
Motivational Example

Energy minimization without concern for reliability



Reduce energy by 42.58%. Prob. of failure increased by 160 times (>> 10 times).

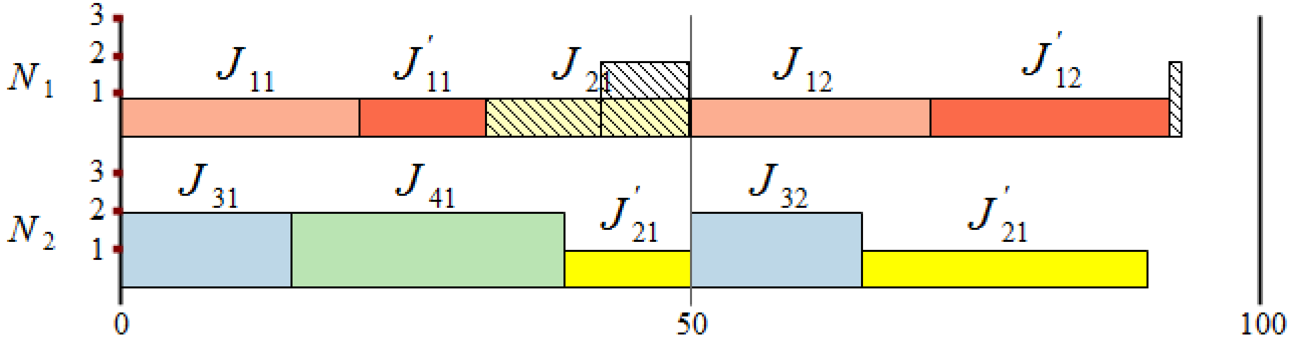
Energy/reliability trade-off optimization



Reduce energy by 42.13%. Prob. of failure increased by 7 times (< 10 times).

Energy/Reliability Trade-off at Runtime

Further energy saving in online optimization without impacting the reliability goal



Reduce energy by 50.66%. Prob. of failure increased by 10 times (= 10 times).

Optimization Strategy

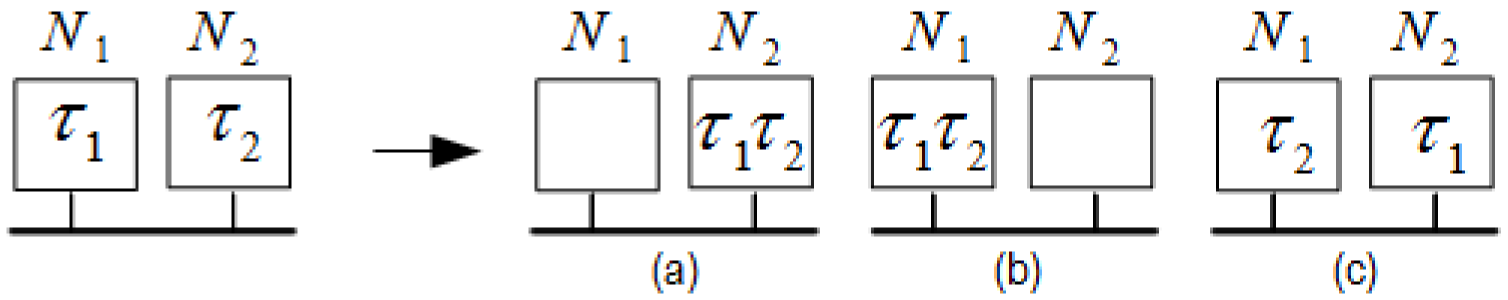
- **Optimization Problem**
  - NP-hard
  - Minimize the cost function:

$$Cost(S) = \underbrace{E_S}_{\text{Energy}} + \underbrace{W_R \cdot \max(0, R_g - R_s)}_{\text{Reliability}} + \underbrace{W_r \cdot \max(0, r_s)}_{\text{Schedulability}}$$

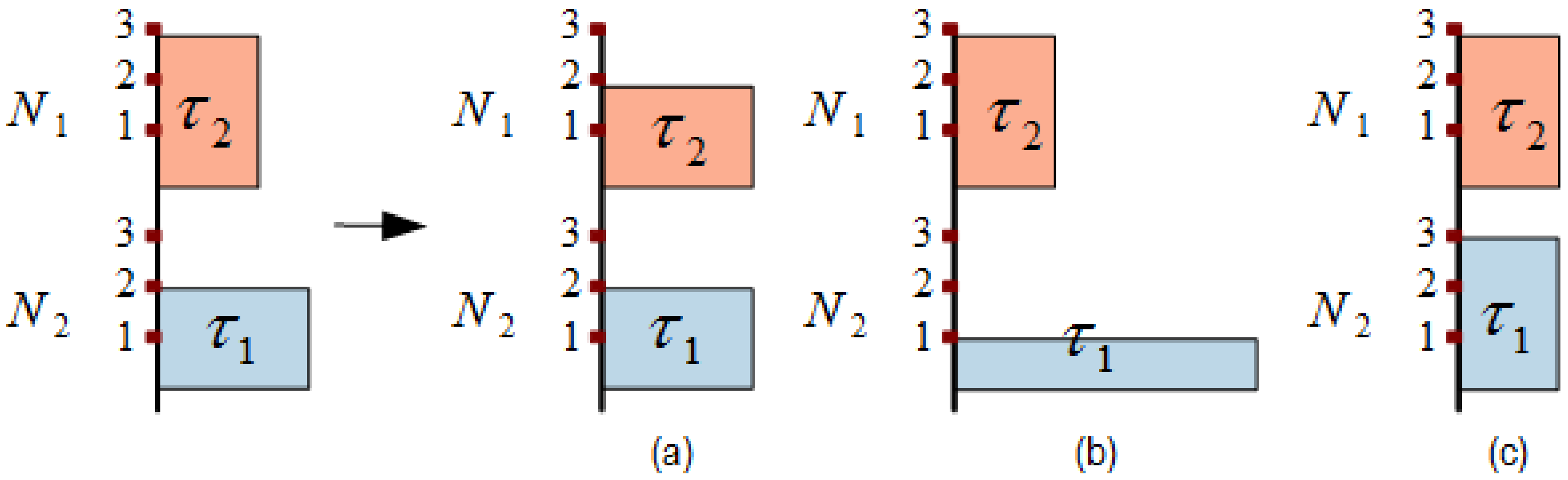
- Use a **TABU search-based algorithm** to explore the design space
  - Iteratively explores neighborhood solutions by
    - mapping moves
    - operating mode moves
  - Avoid being stuck in local optimum
  - Prevents cycling back to previously visited solutions

TABU Search-based Algorithm

Mapping moves



Operating mode moves



TABU Search-based Algorithm

Mapping moves

↓

Operating mode moves

↓

Select a new solution

↓

Save best solution

- **First attempt**
  An improved solution
- **Otherwise**
  Randomly select a non-improving and not-tabu solution
- Maintenance of **TABU-list**

Experimental Results

Reliability improvement for different size of systems

Test Set	Numbers of			$MVFS^{-}$		MVFS	
	PEs	Orig. Tasks	Repl. Tasks	θ [times]	Saved E [%]	θ [times]	Saved E [%]
1	2	8	2	166	28.23	10	24.64
2	4	31	8	112	28.56	10	25.28
3	4	42	11	137	30.47	10	26.04
4	6	63	16	104	25.92	10	21.92
5	6	84	21	57	22.78	10	20.57

$$\theta = \frac{1 - R_s}{1 - R_s^0}$$: measures the reliability degeneration

(i.e. how many times the prob. of failure increases)

$MVFS^{-}$: optimization without concern for reliability

MVFS: energy/reliability trade-off optimization

Experimental Results

Reliability improvement as system utilization increases

Test Set	Numbers of			Initial Util. [%]	MVFS⁻		MVFS	
	PEs	Orig. Tasks	Repl. Tasks		θ [times]	Saved E [%]	θ [times]	Saved E [%]
1	3	20	5	27.21	198	29.57	10	25.00
2	3	20	5	41.00	121	26.55	8	23.02
3	3	20	5	52.73	101	25.26	9	21.05
4	3	20	5	61.56	72	22.94	10	20.09
5	3	20	5	71.98	7	12.76	7	12.76

Experimental Results

Reliability improvement for real-life case studies

Benchmarks	Numbers of			$MVFS^-$		MVFS	
	PEs	Orig. Tasks	Repl. Tasks	θ [times]	Saved E [%]	θ [times]	Saved E [%]
networking-cords	2	13	3	141	28.01	10	20.49
auto-indust-cords	4	24	6	77	22.68	10	17.87
telecom-cords	4	30	8	129	28.16	9	19.57
3 Apps together	6	67	17	64	15.26	10	13.86
Smart-phone	2	61	16	60	18.71	9	15.23

Conclusion and Contributions

- **Conclusion:**

  We are able to reduce the negative impact of energy minimization on reliability with minimal decrease in energy savings.

- **Contributions:**
- Considered energy/reliability trade-offs
  - Proposed an optimization algorithm for the energy/reliability trade-off problem