ENERGY-AWARE STRATEGIES FOR RELIABILITY-ORIENTED REAL-TIME TASK ALLOCATION ON HETEROGENEOUS PLATFORMS

ICPP2020, 18TH AUGUST, 2020

Li HAN[1,2,3], **Yiqin GAO[2,3]**, Jing LIU[1], Yves ROBERT[2,3,4], Frédéric VIVIEN[2,3]

[1]East China Normal University, Shanghai, China

[2]Univ Lyon, ENS Lyon, UCBL, CNRS

[3]Inria, LIP, Lyon, France

[4]University of Tennessee Knoxville

Content

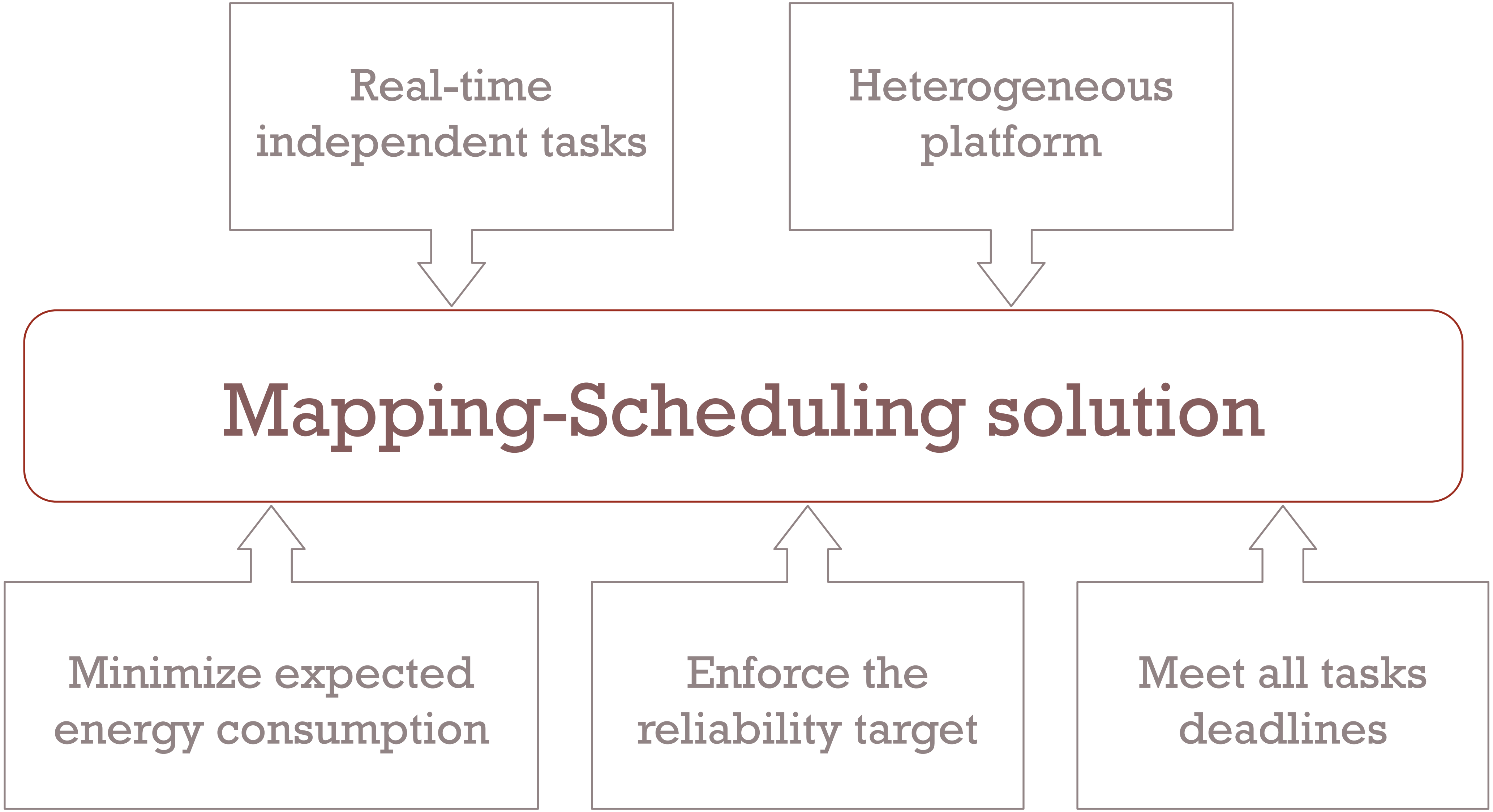
ENERGY-AWARE STRATEGIES FOR RELIABILITY-ORIENTED REAL-TIME TASK ALLOCATION ON HETEROGENEOUS PLATFORMS

Problem definition

Model and algorithm

Experiments

Conclusion


Real-time independent tasks
Heterogeneous platform
Mapping-Scheduling solution
Minimize expected energy consumption
Enforce the reliability target
Meet all tasks deadlines

ENERGY-AWARE STRATEGIES FOR RELIABILITY-ORIENTED REAL-TIME TASK ALLOCATION ON HETEROGENEOUS PLATFORMS

Model

Platform	• M heterogeneous processors • N periodic atomic independent tasks • Worst case execution time of task τ_i on processor m_k : $c_{i,k}$
Deadline	• Tasks period: p
Energy	• Static energy consumption of processor m_k used: $E_{k,s} = P_{k,s} \times p$ • Dynamic energy consumption of task τ_i on processor m_k: $E_d(\tau_i, m_k) = P_{k,d} \times c_{i,k}$
Reliability	• Transient fault rate of processor m_k: λ_k • Reliability of task τ_i on processor m_k: $R(\tau_i, m_k) = e^{-\lambda_k \times c_{i,k}}$ • Task τ_i needs to reach a reliability target R_i using replicas: $R(\tau_i) = 1 - \Pi_{k \in alloc(i)}(1 - R(\tau_i, m_k)) \geq R_i$

Problem description

Objective	
	• Determine for each task τ_i a set of replicas which are executed on a set of processors • Build a schedule for each processor m_k • Minimize expected energy consumption, while matching the deadline p and reliability threshold R_i for each task

This is an **NP-Hard** problem!

Mapping strategies

Objective	Define the number of replicas for each task, as well as the execution processor for every replica

Algorithm	• Order all the tasks with one of the following criteria and renumber them $\tau_1, \tau_2, \ldots, \tau_N$
	- **deW/inW**: decreasing/increasing average WCET on all processors
	- **deMinW/inMinW**: decreasing/increasing minimum WCET on all processors
	- **deMaxW/inMaxW**: decreasing/increasing maximum WCET on all processors
	- **random**: random ordering
	• For each task τ_i in the list,
	  • Order all processors with one of the following criteria and renumber them $m_1, m_2, \ldots, m_M$
	  - **inE**: increasing energy cost
	  - **deR**: decreasing reliability
	  - **deP**: decreasing ratio of $-\frac{\log_{10}(1-R(\tau_i,m_k))}{E(\tau_i,m_k)}$
	  - **random**: random ordering
	  • While the reliability target R_i is not reached, add replicas for task τ_i in the order of $m_1, m_2, \ldots, m_M$, skip the processor if already full

Scheduling strategies

Objective	Order the replicas mapped on each processor, to minimize the energy consumption during execution
Algorithm	• Order all the tasks with one of the following criteria and renumber them $\tau_1, \tau_2, \ldots, \tau_N$ - **deNR/inNR**: decreasing/increasing number of replicas - **deU/inU**: decreasing/increasing total utilization (sum up the ratio of $c_{i,k}/p$ for all replicas of task τ_i) - **random**: random ordering • For each task τ_i in the list, identify a primary replica according to one of following criteria and schedule it as soon as possible on its assigned processor - **time**: choose the replica that can complete the earliest on its processor - **energy**: choose the replica that can be executed with the smallest dynamic energy • With a round-robin fashion in the reversed list $\tau_N, \tau_{N-1}, \ldots, \tau_1$, choose a secondary replica of task τ_i (if exists) and schedule it as late as possible on its assigned processor - **time**: choose the replica whose start time can be the latest - **energy**: choose the replica that can be executed with the largest dynamic energy

ENERGY-AWARE STRATEGIES FOR RELIABILITY-ORIENTED REAL-TIME TASK ALLOCATION ON HETEROGENEOUS PLATFORMS

Problem definition

Model and algorithm

Experiments

Conclusion

Parameters

Parameter	Description	Values
M	Number of processors	= 10
N	Number of tasks	= 20
p	Period and deadline of tasks	= 100
cor_{task} / cor_{proc}	Correlation of WCET values between the different tasks and processors	= {0, 0.25, 0.5, 0.75, 1}
$c_{i,k}$	Worst case execution time of task τ_i on processor m_k	Randomly generated with correlation
BasicWork	Estimation of the fraction of time that the platform is used if each task has a single replica	$= \frac{\sum_{i,k} c_{i\,k}}{M^2 p} = \frac{\sum_{i,k} c_{i\,k}}{M} \times \frac{1}{Mp}$ ={0.1, 0.2, 0.3}
$\beta_{b/w}$	Ratio between best case execution time and worst case execution time	= {0.2, 0.4, 0.6, 0.8, 1}
$w_{i,k}$	Actual execution time of task τ_i on processor m_k	$[\beta_{b/w}\, c_{i,k}, c_{i,k}]$
$P_{k,s}$	Static power of processor m_k	= 0.001 (for all k)
$P_{k,d}$	Dynamic power of processor m_k	[0.8, 1.2] (small failure rate case) [0.08, 0.12] (big failure rate case)
$\lambda_{k,d}$	Failure rate of processor m_k	[0.0001, 0.00023] (small failure rate case) [0.01, 0.023] (big failure rate case)
R_i	Reliability target of task τ_i	{0.9, 0.92, 0.94, 0.96, 0.98} (small failure rate case) {0.8, 0.85, 0.9, 0.95} (big failure rate case)

Results

- Large failure rate case and only representative results presented
- Comparison of our strategies and a theoretical unreachable lower bound established
- Result represented as a ratio to the random baseline method

  Mapping: For each task, add replicas randomly on available processors until reaching its reliability target

  Scheduling: Randomly order replicas mapped on each processor and execute them in sequence and as soon as possible
- 1000 experiments randomly generated for each point on the graph

Results

Ratio of energy consumption to the baseline of different mapping and scheduling ordering tasks criteria

map \ sch	*deNR*	*inNR*	*deU*	*inU*	*random*
deW	0.5655	0.5662	0.5655	0.5660	0.5663
inW	0.5631	0.5635	0.5630	0.5635	0.5635
deMinW	0.5658	0.5662	0.5657	0.5661	0.5665
inMinW	0.5637	0.5642	0.5637	0.5642	0.5641
deMaxW	0.5658	0.5664	0.5657	0.5663	0.5665
inMaxW	0.5629	0.5633	0.5629	0.5633	0.5633
random	0.5633	0.5639	0.5633	0.5638	0.5639

Reminder

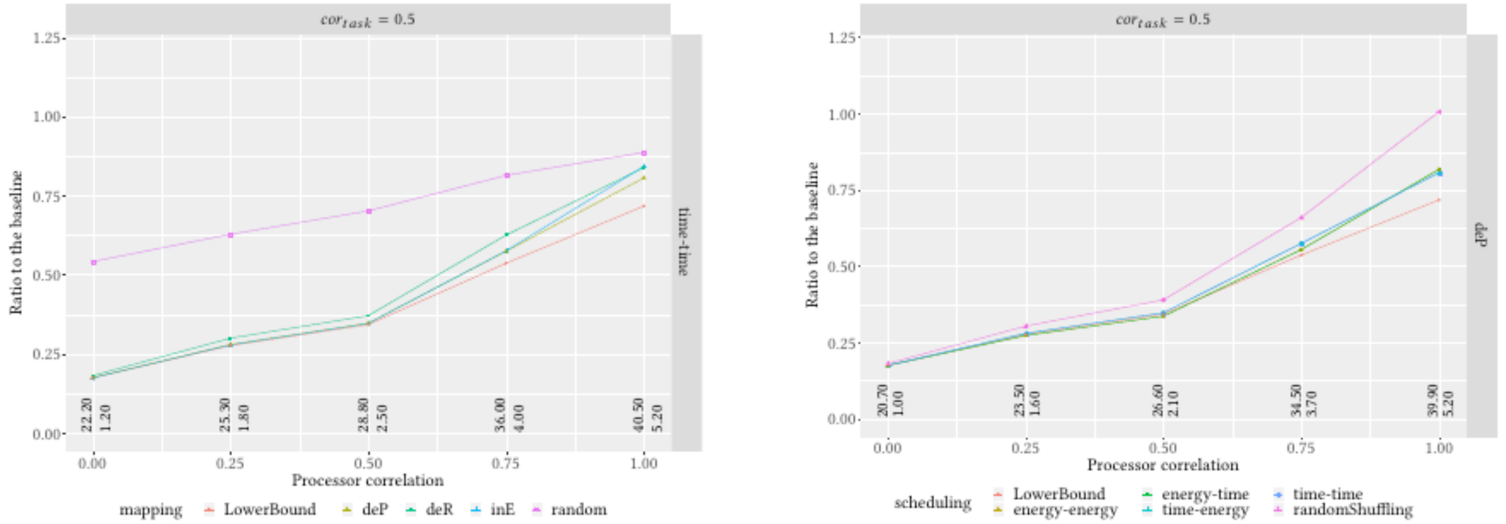
Mapping ordering tasks criteria
- **deW/inW**: decreasing/increasing average WCET on all processors
- **deMinW/inMinW**: decreasing/increasing minimum WCET on all processors
- **deMaxW/inMaxW**: decreasing/increasing maximum WCET on all processors
- **random**: random ordering

Scheduling ordering tasks criteria
- **deNR/inNR**: decreasing/increasing number of replicas
- **deU/inU**: decreasing/increasing total utilization (sum up the values of $c_{i,k}/p$ for all replicas)
- **random**: random ordering

◆ Ordering task criteria do not critically influence energy consumption

◆ Focus on selecting processors strategies during the mapping phase, and on choosing primary and secondary replicas during the scheduling phase

Results



(a) Comparing mapping strategies when using *time-time* as scheduling strategy

(b) Comparing scheduling strategies when using *deP* as mapping strategy

Ratio of energy consumption using different mapping and scheduling strategies under big failure rate, when varying cor_{proc}, with $basicWork = 0.3$, $\beta_{\text{b/w}} = 1$, $\mathcal{R}_i = 0.95$ and $cor_{\text{task}} = 0.5$.

Reminder

Mapping ordering processors criteria

- **inE**: increasing energy cost
- **deR**: decreasing reliability
- **deP**: decreasing ratio of $-\frac{\log_{10}(1-R(\tau_i, m_k))}{E(\tau_i, m_k)}$
- **random**: random ordering

Scheduling primary replica choosing criteria

- **time**: choose the processor that can complete the replica the earliest
- **energy**: choose the processor that can execute the replica with the smallest dynamic energy

Scheduling secondary replica choosing criteria

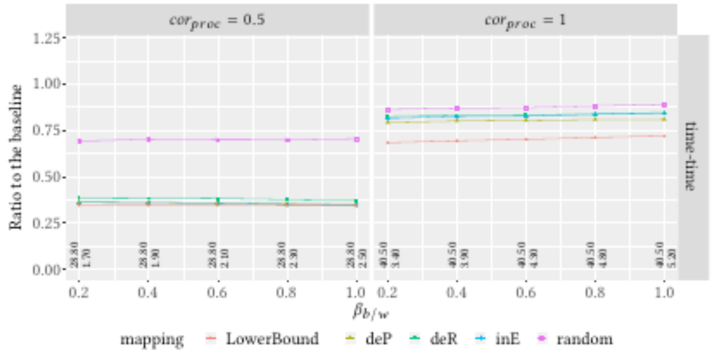
- **time**: choose the processor whose start time can be the latest
- **energy**: choose the processor that can execute the replica with the largest dynamic energy

◆ Result less than 0.25 and close to lower bound when $\text{cor}_{\text{proc}} = 0$, ratio increases with cor_{proc}

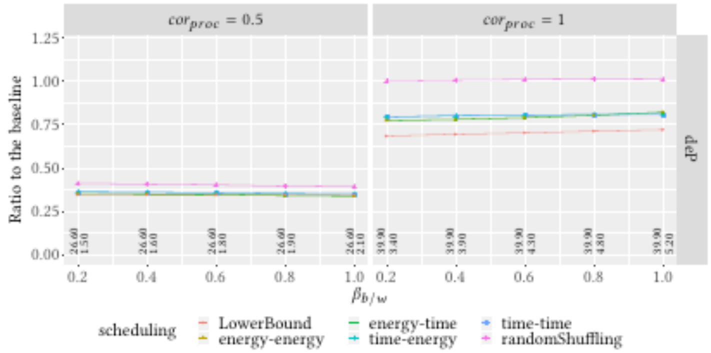
◆ Mapping: deP performs better than, or similarly to the best strategy between deR and inE

◆ Scheduling: Little difference between our different scheduling criteria, time better when $\text{cor}_{\text{proc}} \neq 1$ and energy better when $\text{cor}_{\text{proc}} = 1$

Results



(a) Comparing mapping strategies when using *time-time* as scheduling strategy



(b) Comparing scheduling strategies when using *deP* as mapping strategy

Ratio of energy consumption using different mapping and scheduling strategies under big failure rate when varying $\beta_{b/w}$, with *basicWork* = 0.3, $\mathcal{R}_i$ = 0.95 and *cor*$_{\text{task}}$ = 0.5.

Reminder

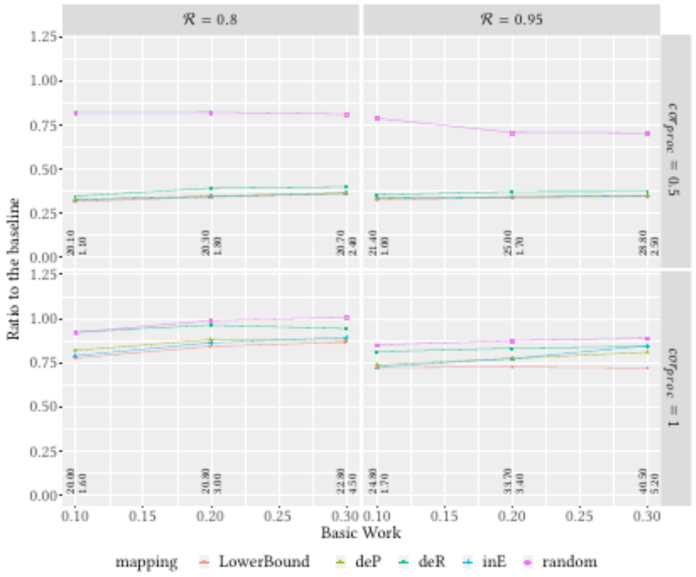
Mapping ordering processors criteria
- **inE**: increasing energy cost
- **deR**: decreasing reliability
- **deP**: decreasing ratio of $-\frac{\log_{10}(1-R(\tau_i,m_k))}{E(\tau_i,m_k)}$
- **random**: random ordering

Scheduling primary replica choosing criteria
- **time**: choose the processor that can complete the replica the earliest
- **energy**: choose the processor that can execute the replica with the smallest dynamic energy

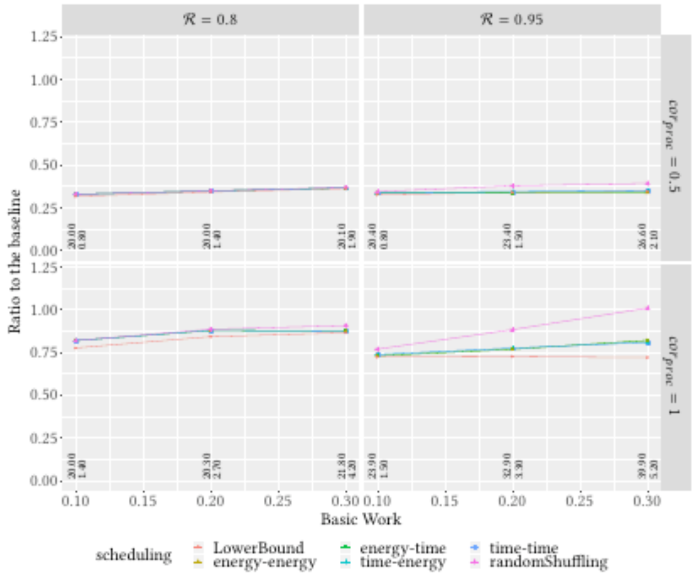
Scheduling secondary replica choosing criteria
- **time**: choose the processor whose start time can be the latest
- **energy**: choose the processor that can execute the replica with the largest dynamic energy

- Results are similar whatever the value of $\beta_{b/w}$
- Actual tasks execution time in $[\beta_{b/w} c_{i,k},\ c_{i,k}]$, thus robustness of our heuristics

Results



(a) Comparing mapping strategies when using *time-time* as scheduling strategy



(b) Comparing scheduling strategies when using *deP* as mapping strategy

Ratio of energy consumption using different mapping and scheduling strategies when varying *basicWork* and $\mathcal{R}_i$, under big failure rate, with $\beta_{b/w} = 1$ and $cor_{task} = 0.5$.

Reminder

Mapping ordering processors criteria

- **inE**: increasing energy cost
- **deR**: decreasing reliability
- **deP**: decreasing ratio of $-\frac{\log_{10}(1-R(\tau_i,m_k))}{E(\tau_i,m_k)}$
- **random**: random ordering

Scheduling primary replica choosing criteria

- **time**: choose the processor that can complete the replica the earliest
- **energy**: choose the processor that can execute the replica with the smallest dynamic energy

Scheduling secondary replica choosing criteria

- **time**: choose the processor whose start time can be the latest
- **energy**: choose the processor that can execute the replica with the largest dynamic energy

- ◆Mapping: deP has always better or similar performance than the best of deR and inE
- ◆Scheduling: similar performance on all criteria except random
- ◆Little difference between our strategies and the lower bound, difference of 10% in the worst case

Results

- Save more than 40% of energy compared to baseline, except in the high cor_{proc} case, ratio can be as low as 20% in the best case
- deP method is the best processor ordering during the mapping phase
- All primary-secondary replica choosing heuristic performs well, "time" good primary replica choosing criteria when $cor_{proc} \approx 1$, "energy" other cases
- Performance of the best heuristics is only 17.0% higher than the lower bound in the worst case, the median value is only 3.5% and the average value is only 4.3% higher
- We can confidently conclude that our best strategies perform remarkably well over the whole experimental setting

ENERGY-AWARE STRATEGIES FOR RELIABILITY-ORIENTED REAL-TIME TASK ALLOCATION ON HETEROGENEOUS PLATFORMS

Problem definition

Model and algorithm

Experiments

Conclusion

Conclusion

Problem	Execute periodic real-time tasks on an heterogeneous platform, with objectives to minimizing the energy consumption, guaranteeing some reliability thresholds, and meeting all deadlines
Solution	A multi-criteria mapping-scheduling heuristic
Experiments	Our best heuristics always achieve good performance, which is very close to the lower bound
Future work	Extend to periodic graphs of tasks instead of periodic set of independent tasks

THANK YOU FOR LISTENING !