Toward Contention Analysis for Parallel Executing Real-Time Tasks

Fabrice Guet, Luca Santinelli, Jerome Morio ONERA
Guillaume Phavorin, Eric Jean IRT Saint Exupery

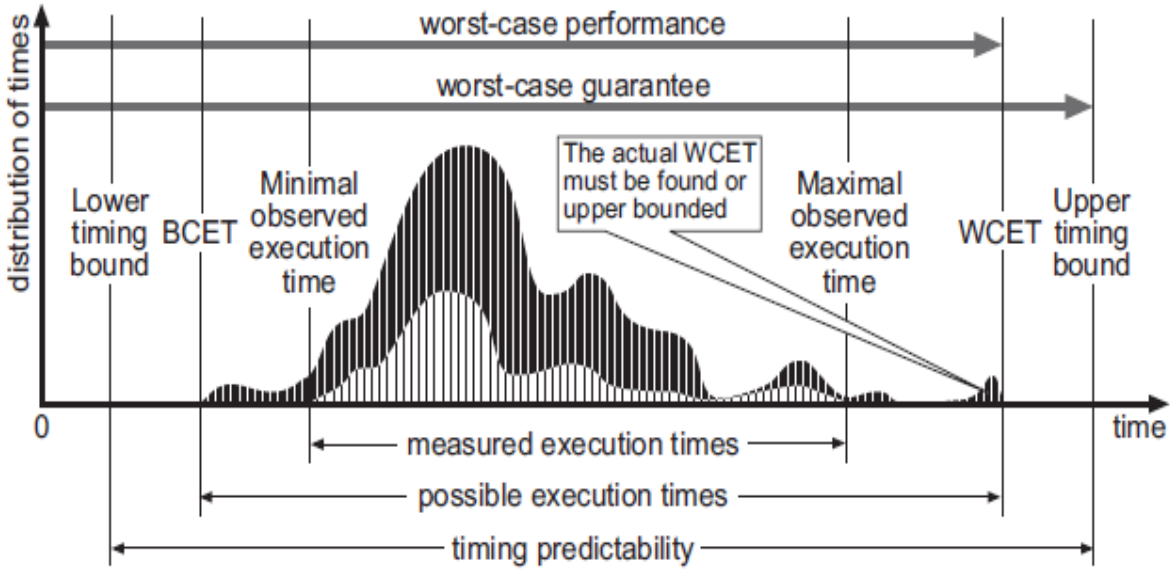
Probabilistic Real-Time System

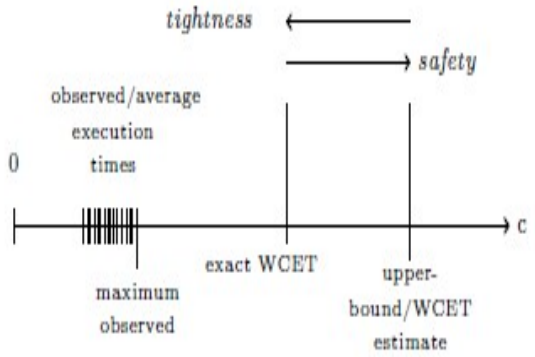
« A probabilistic real-time system is a system where at least one parameter is described with a probability distribution »

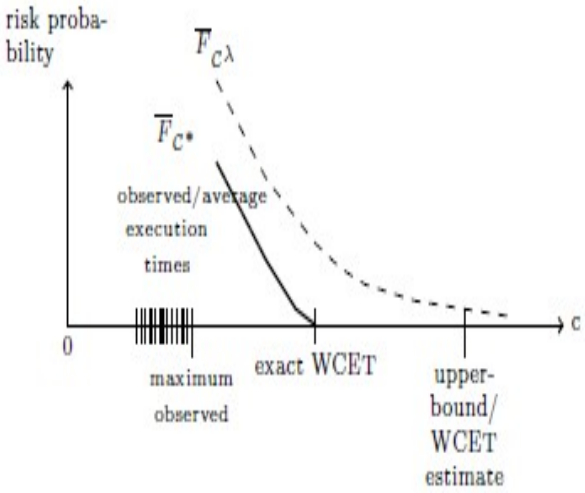
Modeling and schedulability analysis with probabilities

Modeling with Probabilities

Probabilistic Worst-Case Execution Time:

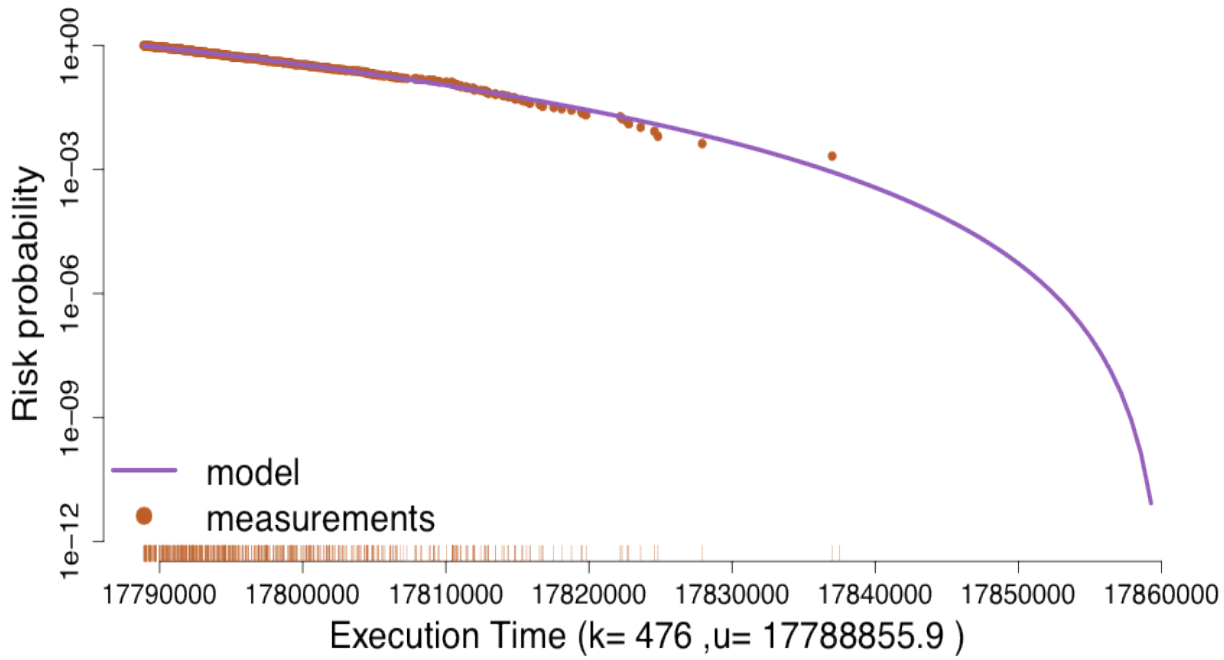




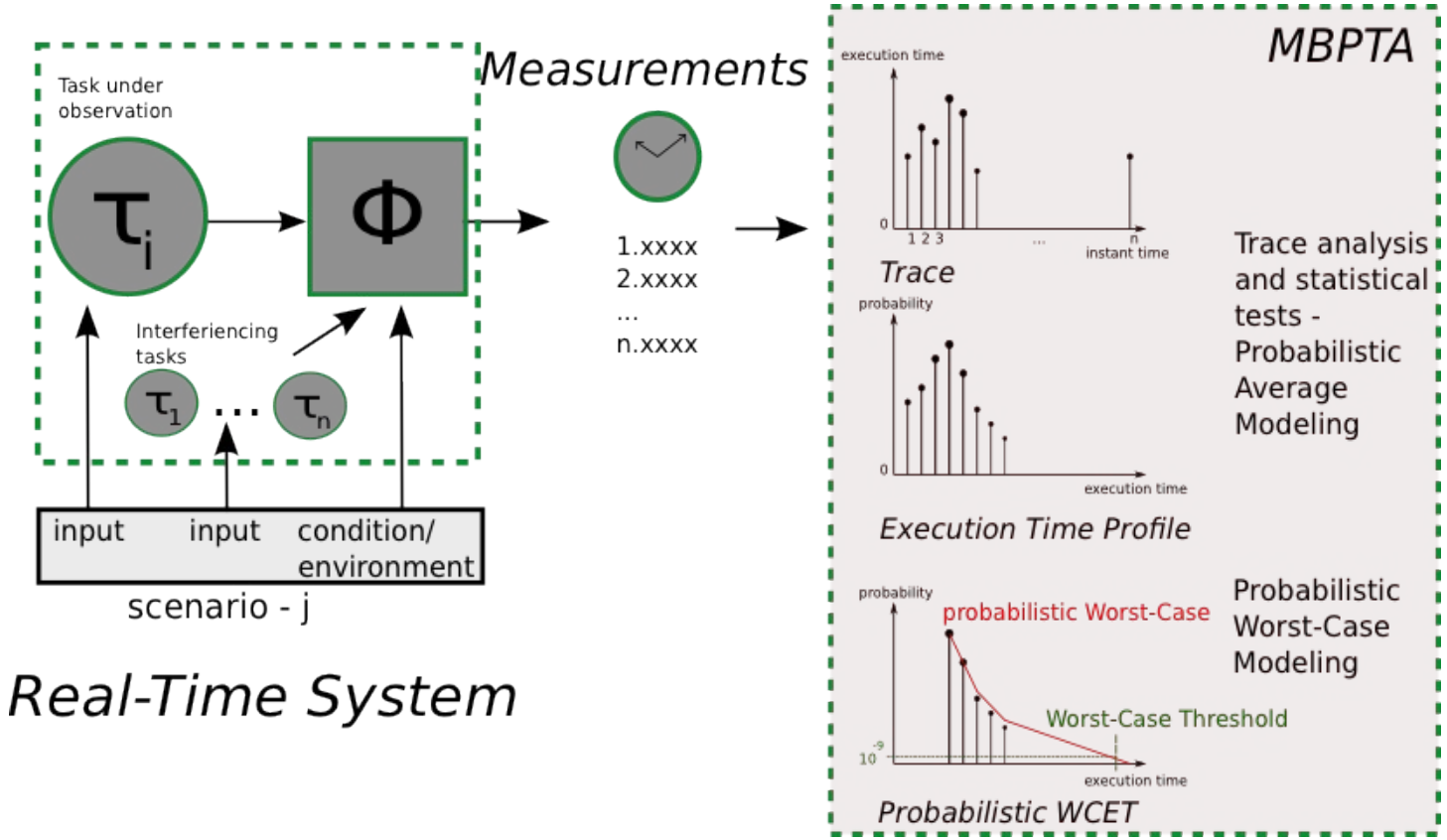


Measurement-Based Probabilistic Timing Analysis

- Measurements
- Statistical analysis + Extreme Value Theory (EVT)
    pWCET estimates

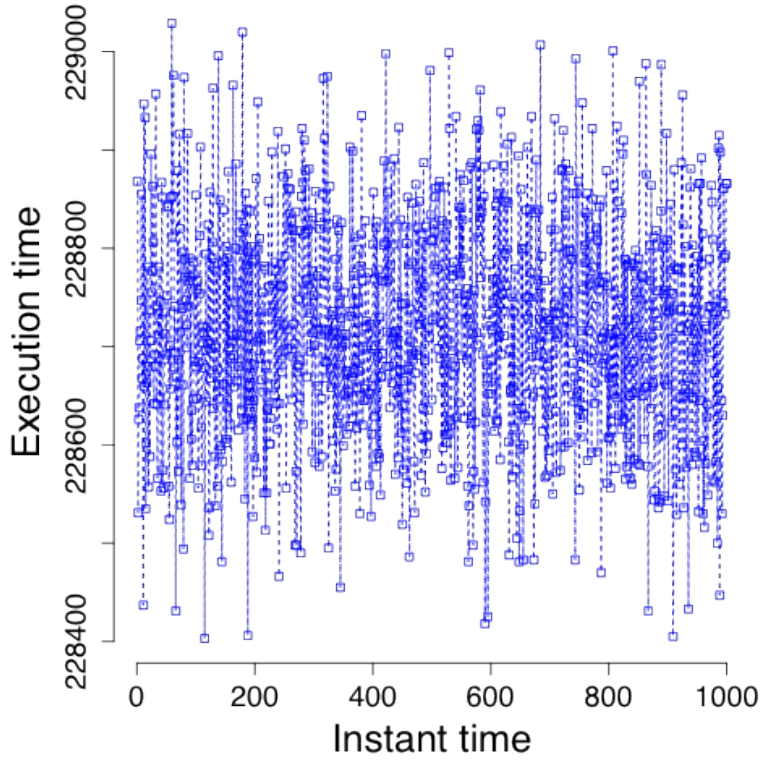


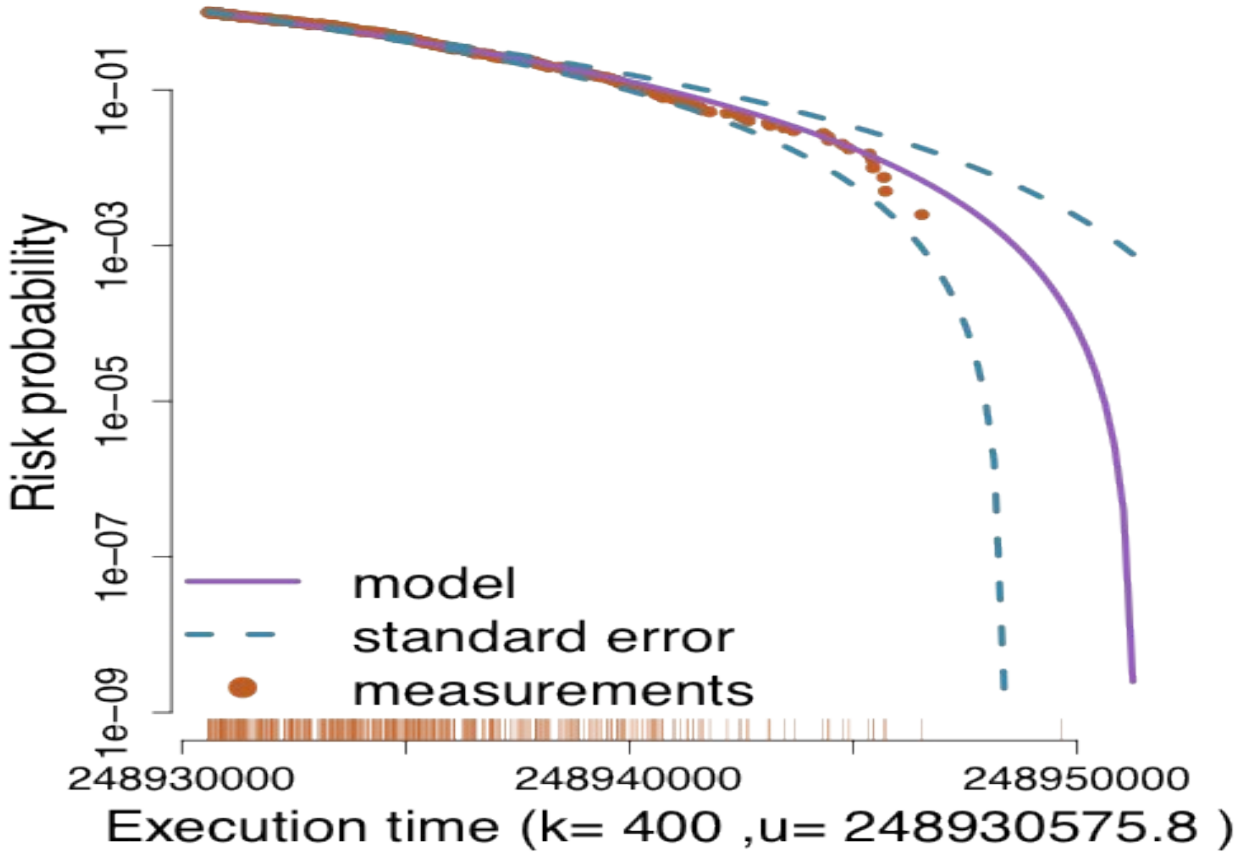
Measurement-Based Probabilistic Timing Analysis



For evaluation only!! Non-time randomized systems!

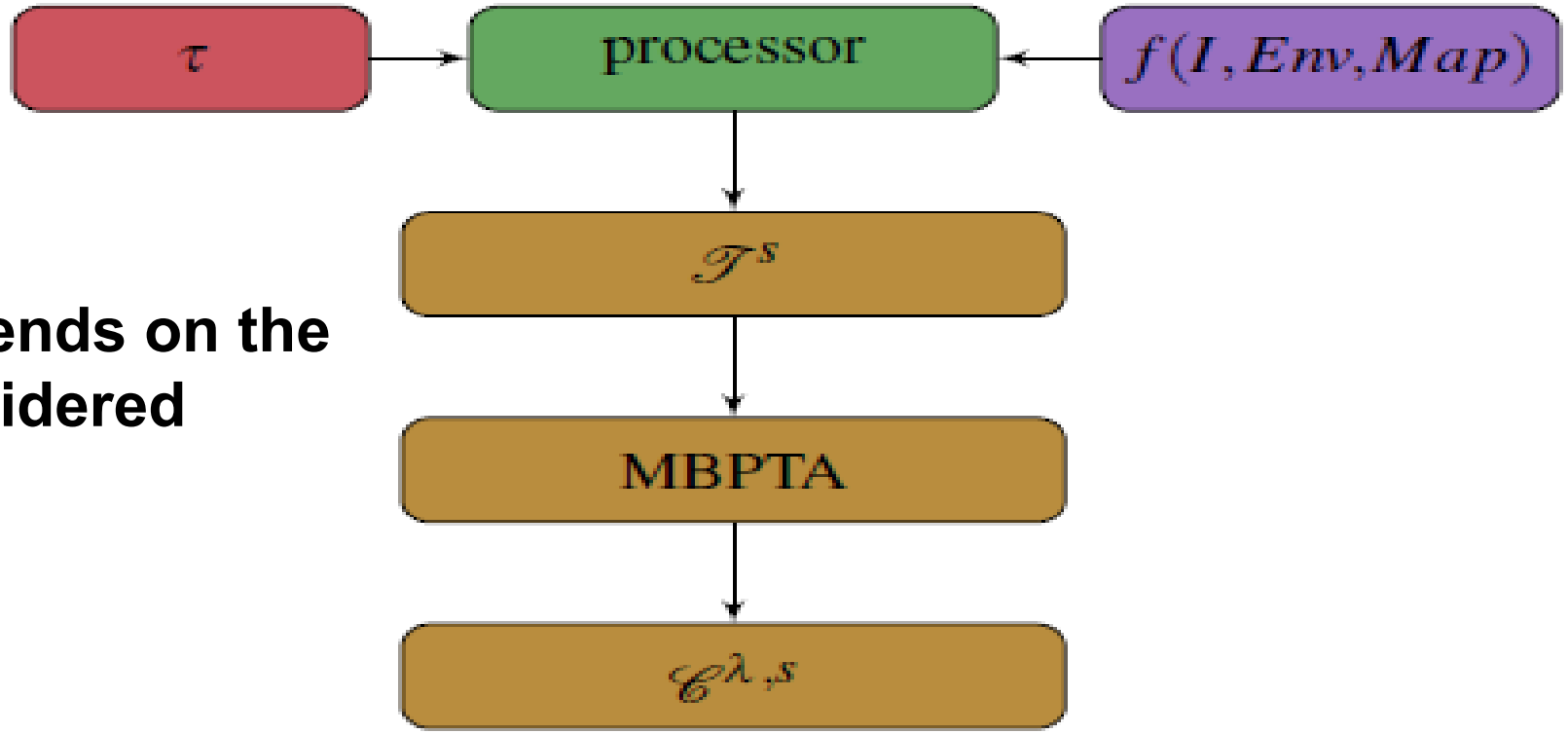
From measurements to pWCET





Scenario dependence

MBPTA and pWCET depends on the execution scenario considered

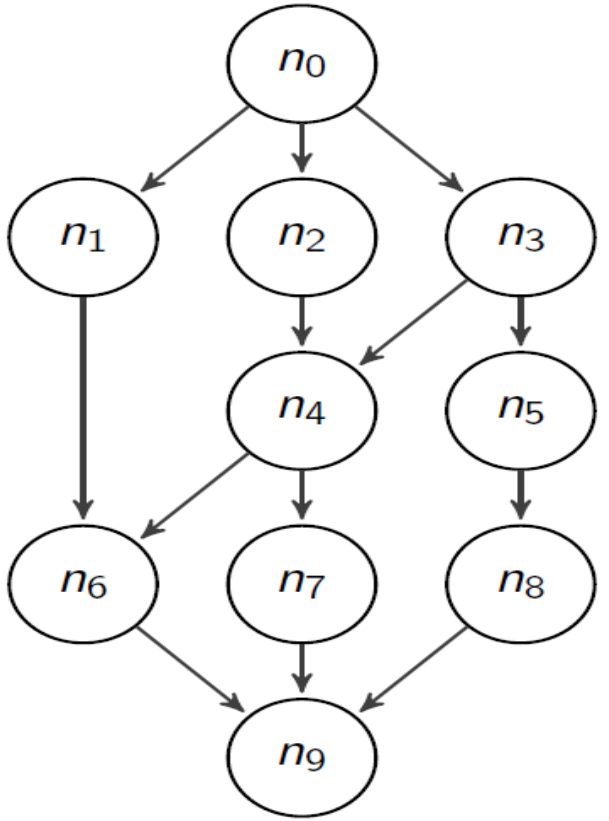


Contention

- Interference from other tasks on shared resources: contention
- Task set with precedence constraints (DAG)
- Multiple possible scenarios

Notions of: independence, parallel execution, and thus contention

Finding contention and worst-case scenario among the possible from parallel executions

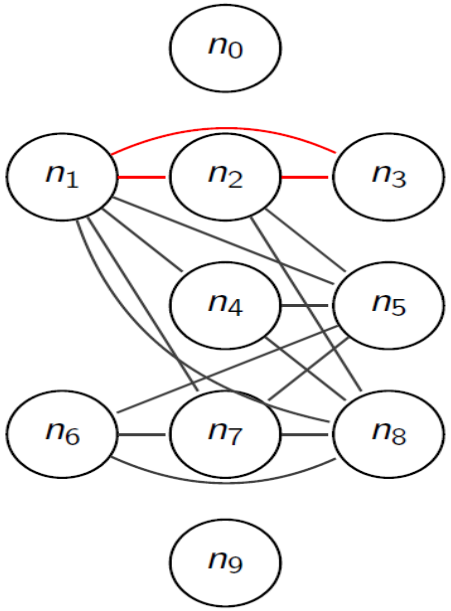


Contention analysis

Step 1: **Independence analysis & Contention list search -**
Potential contenders of a task are tasks that can execute in parallel to it: tasks independent from it

Contender List: seek for the list of possible contenders to a task.
clique - a set of tasks, including the task under observation, in which every couple of tasks are independent

$$\Gamma(n_i) = \{n_j \in G \mid n_i \nabla n_j\}$$



For each task there exists a maximal and minimal size to its cliques ContenderList(task). Thre exists constraints...

Contention analysis

Step 2: **Contentiousness Characterization**
The objective of the worst contention analysis is identifying for a task, its worst contenders within ContenderList(task).
Running potential contender in parallel together with task to seek foster the worst interferences

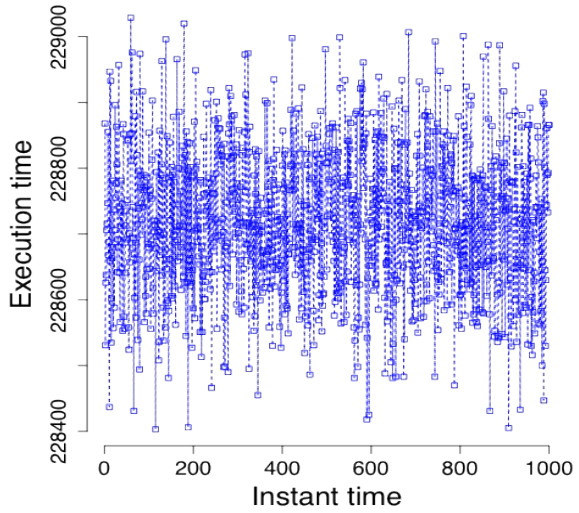
Measure the degree of contentiousness, with mem_band heuristic.

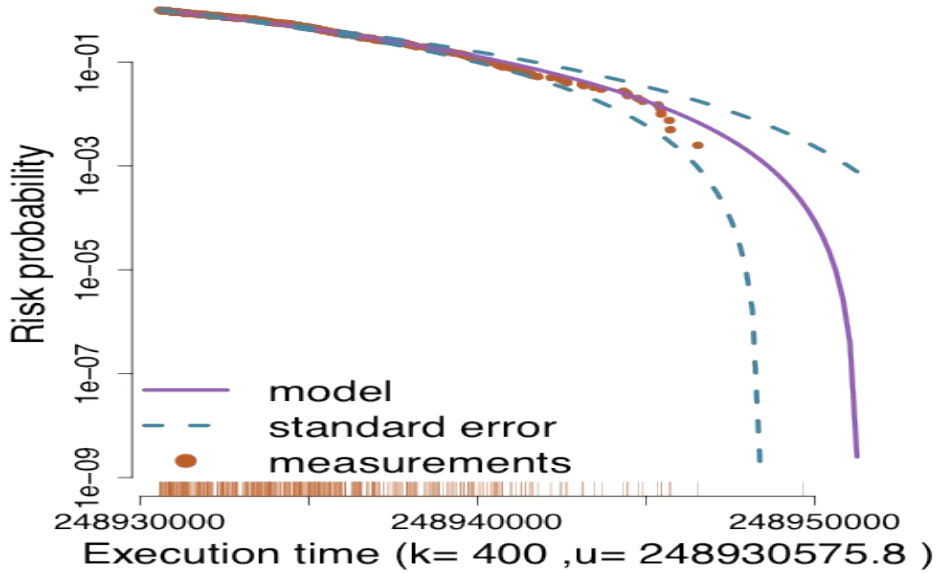
$$mem\_band \text{ as: } mem\_band \stackrel{def}{=} accesses/t$$

Contender Classification. The objective of the worst contention analysis is identifying for n_i, its $min(M-1, |clique^{\max}(n_i)|-1)$ worst contenders within $ContenderList(n_i)$. Running them in parallel together with n_i would foster the worst interferences for n_i.

Contention analysis

Step 3: **Worst contention scenario measurements**:
MBPTA from the worst-scenario. EVT for pWCET estimation: worst pWCET!





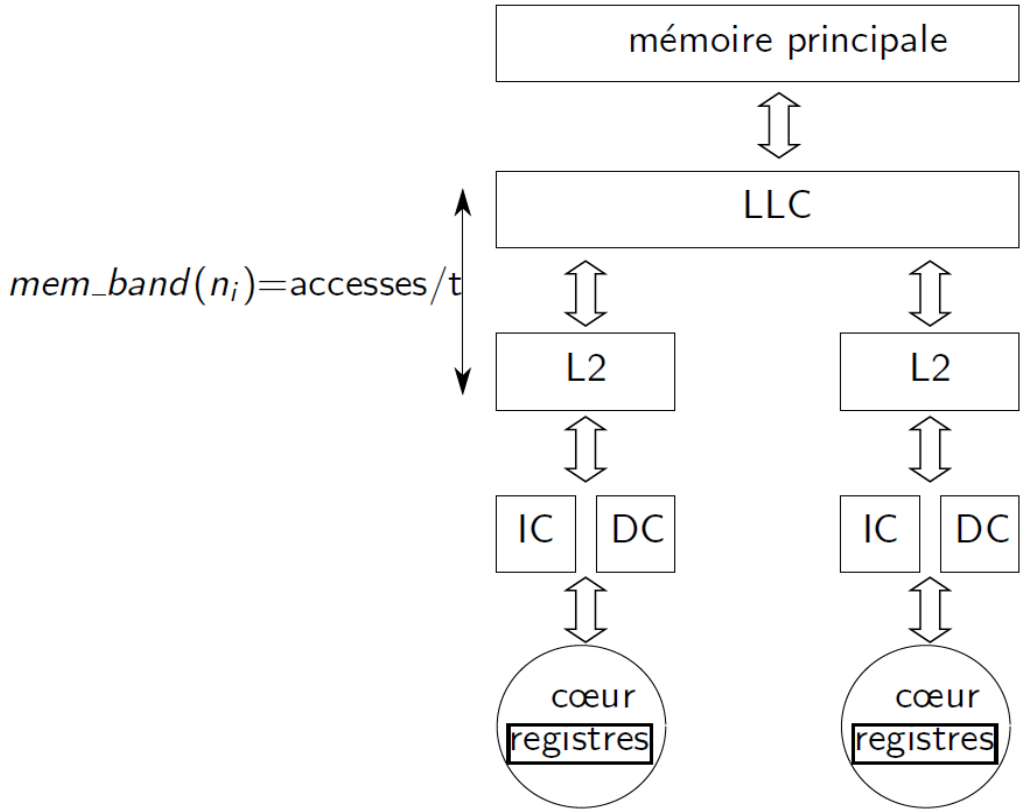
Contention analysis (recap)

1. *Contender List Search* $\forall n_i \in G, ContenderList(n_i)$ is the result of the contender list search and the G transformation into $\overline{G}$. This step includes the independence analysis;
2. *Contentiousness Characterization*: $\forall n_i \in G, mem\_band(n_i)$ is measured using τ_{mon} in case of $\Phi^M(n_i||\tau_{mon}(t))$. $\tau_{mon}(t)$ is an artifact task used to monitor other task effect on shared resources;
3. *Contender Task Sets Classification*: $\forall n_i \in G$, $sort(ContenderList(n_i))$ sorts the task sets from the ones with the greatest sum of memory bandwidth usage values $sort(ContenderList(n_i))[\cdot]$ to those with the least sum of memory bandwidth usage values $sort(ContenderList(n_i))[|ContenderList(n_i)|]$;
4. *Worst Contention Scenario Measurements*: $\forall n_i \in G, \mathcal{T}(n_i)$ is the measurement trace under s^{worst} $\Phi^M(n_i||sort(ContenderList(n_i))[1])$ to which the EVT is applied.

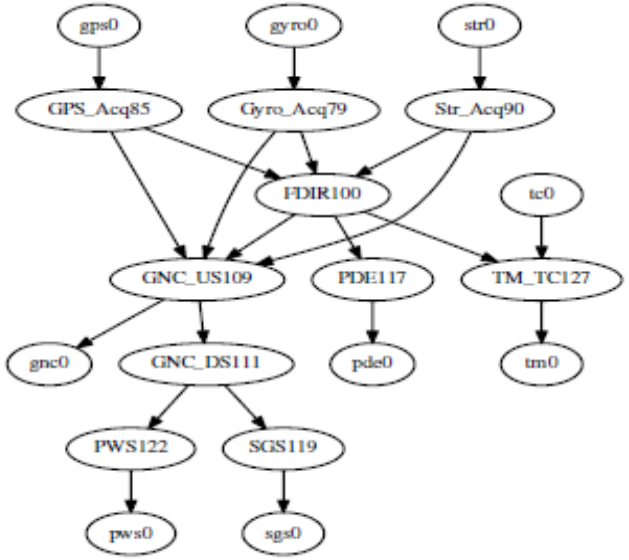
Simulator

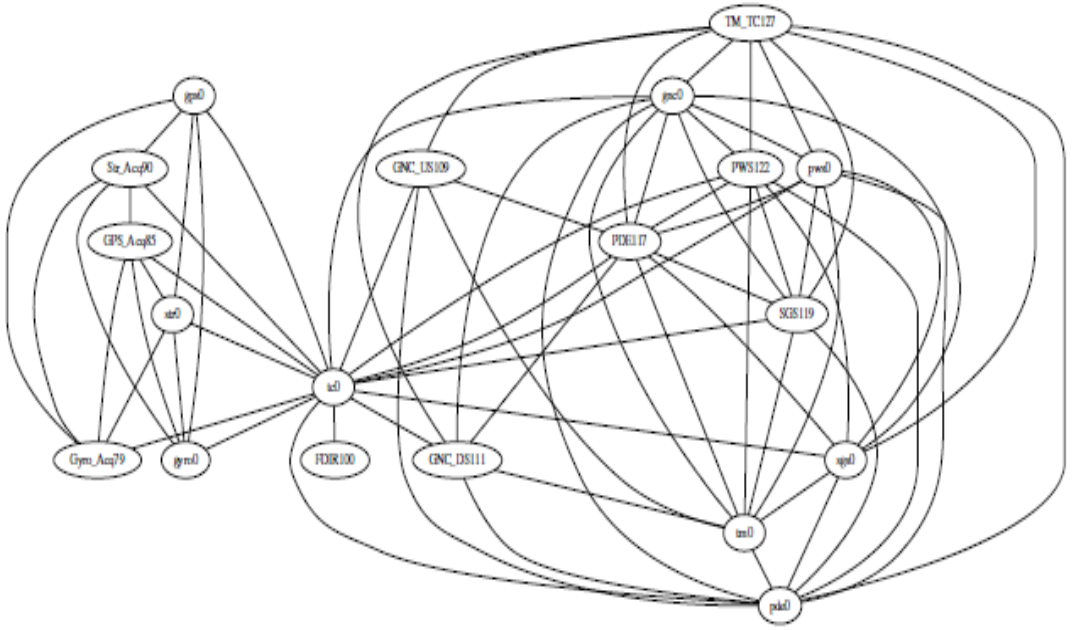
Simulator and random generation:

- Focus on memory shared resources and memory bus
- Random profiles to memory access



Some results





- *ContenderList*(GPS_Acq85) ={{Str_Acq90,tc0,gyro0}, {Str_Acq90,tc0,Gyro_Acq79}, {str0,tc0,gyro0}, {str0,tc0,Gyro_Acq79}}
- *sort*(*ContenderList*(GPS_Acq85)) ={ts1={Str_Acq90,tc0,gyro0}, ts2={str0,tc0,gyro0}, ts3={Str_Acq90,tc0,Gyro_Acq79}, ts4={str0,tc0,Gyro_Acq79}}

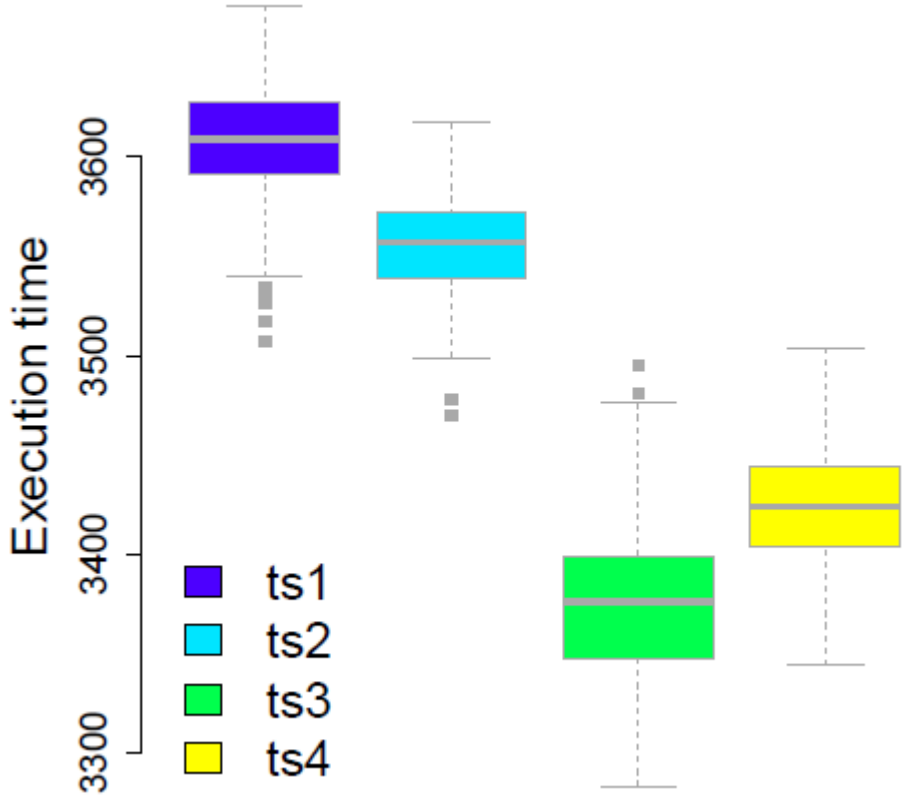
Some results

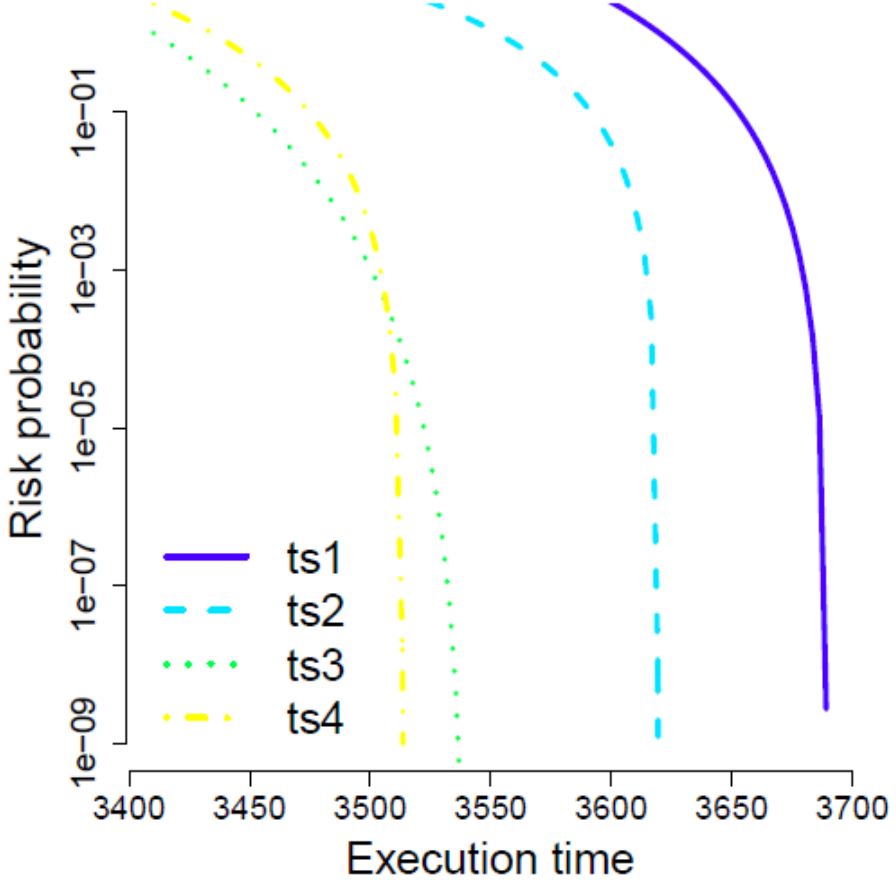
task	$\lvert clique^{\max}(n_i)\rvert$	# $\Gamma(n_i)$
gps0	4	4
gyro0	4	4
str0	4	4
GPS_Acq85	4	4
Gyro_Acq79	4	4
Str_Acq90	4	4
FDIR100	1	1
tc0	4	26
GNC_US109	3	6
PDE117	4	31
TM_TC127	4	22
gnc0	4	50
GNC_DS111	4	6
pde0	4	31
tm0	4	22
PWS122	4	28
SGS119	4	28
pws0	4	28
sgs0	4	28

task	execution time (cycles)	L1 hits/misses	L2 hits/misses	bus accesses	*mem_band*	rank
GPS_Acq85	1025	123/177	93/84	204	0.68	-
Str_Acq90	811	225/75	28/47	118	0.40	2
tc0	271	89/11	0/11	34	0.34	3
gyro0	374	53/47	3/44	57	0.57	1
Gyro_Acq79	641	285/15	0/15	68	0.23	5
str0	261	94/6	0/6	32	0.32	4
gps0	355	55/45	5/40	54	0.54	-

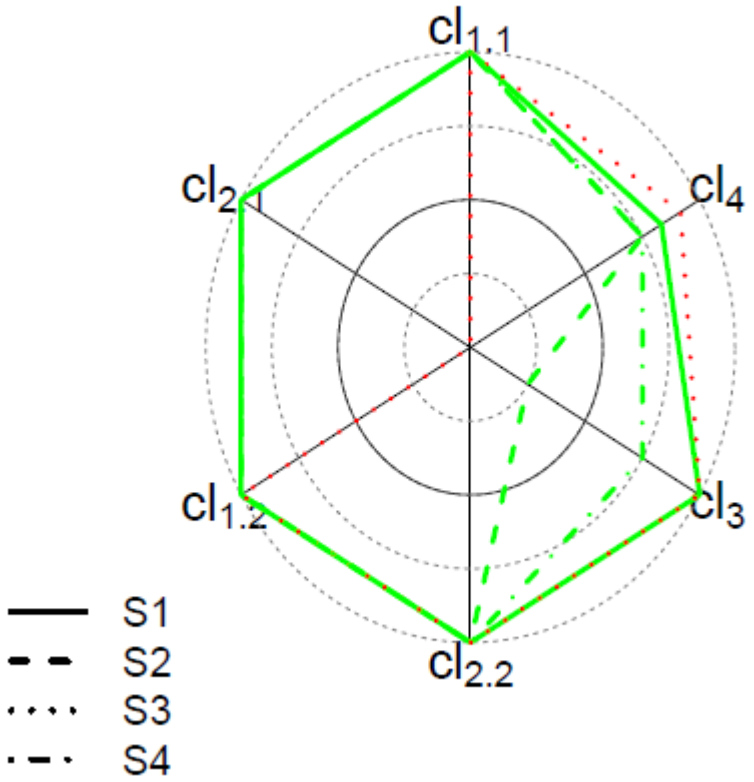
Complex problem: need for less expensive alternatives!

Some results





Some results



$\mathcal{T}$	cl_4	$\langle WCET; 10^{-9} \rangle$	$WCET^{-SE}$	cl_3^{-SE}	$WCET^{+SE}$	cl_3^{+SE}
S1	3.33	3693	3676	4	3790	4
S2	3.00	3618	3617	1	3638	1
S3	3.67	3548	3495	4	5048	4
S4	3.00	3513	3504	3	3568	3

Thank you

luca.santinelli@onera.fr