NPRT – version 1.1.19

Imad Alzeer – Pierre Molinaro
IRCCyN
17 janvier 2005

Les algorithmes de calcul sont implémentés sous la forme d'un outil « ligne de commande » : ***nprt*** (***Non Preemptive Response Time***), qui traite aussi bien l'ordonnancement des tâches indépendantes monoprocesseur que les tâches et messages d'un système réparti.

L'outil accepte comme donnée d'entrée la description d'un système de processeurs, de réseaux, de tâches et de messages à ordonnancer. Le compilateur du langage déclaratif utilisé est inclus dans l'outil. Le langage déclaratif est basé sur l'outil GALGAS (http://www.irccyn.ec-nantes.fr/irccyn/d/fr/equipes/TempsReel/logs/software-5-galgas), et il fonctionne sous *Windows*, Linux et *MacOSX*.

Nous présentons les options de ligne de commande disponibles, puis nous détaillons ce langage dédié.

1 Options ligne de commande

1.1 Sysnopsis

```
nprt [-CM] [--create-intermediate-files] [--use-max-durations]
source_files
```

Chaque fichier source contient un texte décrivant un système que nous soumettons à l'analyse. Notre outil traite ces fichiers les uns après les autres, dans leur ordre d'apparition dans la ligne de commande. Les options présentes sur la ligne de commande s'appliquent de la même façon à tous les fichiers source, quelque soit la position à laquelle elles apparaissent.
L'extension que nous avons choisie pour nos fichiers source est ***.nprt***.

1.2 Options

```
-C
--create-intermediate-files
```

Crée pour chaque système traité deux fichiers ***html*** contenant des résultats intermédiaires ; le premier contient la liste des instances déployées et le second les bornes minimum ($R^{\downarrow}$) et maximum ($R^{\uparrow}$) des temps de réponse pour chaque instance.

```
-M
--use-max-durations
```

Force les tâches et messages à leurs durées d'exécution et de transmission maximales.

2 Paramètres du système

Le langage permet la description d'un système par une liste de processeurs, de réseaux (CAN ou VAN), de tâches et de messages.

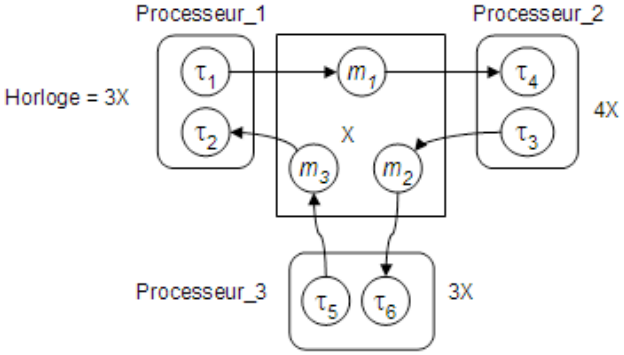
2.1 Bornes du message CAN

Le ***bit stuffing***, a pour conséquence que les durées des messages peuvent varier entre $C^{\downarrow}$ et $C^{\uparrow}$ unités d'horloge réseau. Ces deux bornes sont calculées comme suit, enfonction du nombre b d'octets de données (entre 0 et 8) :

Norme	$C^{\downarrow}$	$C^{\uparrow}$
Standard	$(47 + 8*b)*\tau_{bit}$	$\left(47 + 8*b + \left\lfloor \frac{33 + 8*b}{4} \right\rfloor\right)*\tau_{bit}$
Etendue	$(67 + 8*b)*\tau_{bit}$	$\left(67 + 8*b + \left\lfloor \frac{53 + 8*b}{4} \right\rfloor\right)*\tau_{bit}$

2.2 Exemple

Nous prenons un exemple de trois processeurs connectés par un réseau CAN géré par un protocole standard (identificateur sur 11 bits). Les processeurs dans le système n'ont pas la même puissance du calcul, et leurs horloges sont des multiples de celle du réseau. Les facteurs de ratio d'horloges de trois processeurs par rapport à l'horloge du réseau (X) sont : 3X : 4X : 3X.



Les paramètres des tâches et des messages dans le système sont listés dans la table ci-dessous.

τ/m	π	$C^{\downarrow}$	$C^{\uparrow}$	T
τ_1	1	500	500	2400
τ_2	2	350	360	1200
τ_3	1	440	450	1600
τ_4	2	600	610	3200
τ_5	1	450	455	1200
τ_6	2	350	360	1200

m_1	1	1 oct.=55--65	800
m_2	3	1 oct.=55--65	400
m_3	2	1 oct.=55--65	400

Bien que les tâches et les messages de chaque transaction aient la même période physique, leurs paramètres temporels sont exprimés par des entiers qui sont des multiples de « ticks » de l'horloge de la ressource sur laquelle s'exécute la tâche ou bien circule le message. C'est ce qui explique, par exemple, les valeurs des périodes à 2400 pour $\tau 1$ et 3200 pour $\tau 4$. Les durées des messages (chacun contient un octet de donnée) peuvent varier entre 55 et 65 unités de l'horloge réseau.

2.3 Facteurs d'échelle

Le facteur d'échelle (***scalingfactor***) pour chaque ressource est sont calculé de la manière suivante :

1) On calcule le *PPCM* des horloges dans le système : $PPCM\_hor = PPCM(X, 3X, 4X, 3X)$ =12X.
2) Le facteur d'échelle $Sca_{j\,j}$ pour la ressource res_j est calculé comme : $Sca_j = PPCM\_hor / hor_j$.

2.4 Fichier d'entrée

Donc, le système précédent est décrit par le ficher de texte ***trois_proc.nprt*** suivant :

```
system

processor  p1 scalingfactor  4;
processor  p2 scalingfactor  3;
processor  p3 scalingfactor  4;
can network net  scalingfactor  12;

# independent tasks
task t1 priority 1 duration 500 .. 500 period 2400 processor p1 ;
task t3 priority 1 duration 440 .. 450 period 1600 processor p2 ;
task t5 priority 1 duration 450 .. 455 period 1200 processor p3 ;

# dependent messages
standard message m1 network net length 1 priority 1 on task t1 ;
standard message m2 network net length 1 priority 3 on task t3 ;
standard message m3 network net length 1 priority 2 on task t5 ;

# dependent tasks
task t2 priority 2 duration 350 .. 360 on message m3 processor p1 ;
task t4 priority 2 duration 600 .. 610 on message m1 processor p2 ;
task t6 priority 2 duration 350 .. 360 on message m2 processor p3 ;

end
```

Par commodité, le texte ci-dessus présente la déclaration de chaque entité (processeur, réseau, tâche, message) sur une ligne.

L'ordre de déclaration est quelconque, sous réserve que toute entité soit déclarée avant d'être référencée. Ainsi, la déclaration de la tâche *t1* référence le processeur *p1*, donc la seule contrainte est que la déclaration du processeur *p1* figure avant la déclaration de la tâche *t1*. Cette contrainte n'affecte pas la puissance de description, et plus garantit de facto l'absence de circularité dans les interdépendances entre les tâches et les messages.

Tous les paramètres numériques sont des entiers positifs ou nuls. Des commentaires peuvent être insérés dans le texte : ils sont introduits par le caractère dièse, et s'étendent jusqu'à la fin de la ligne courante. Dans la suite, nous allons détailler chaque type de déclaration.

4 Déclaration

4.1 Déclaration d'un processeur

La déclaration d'un processeur donne un nom à ce processeur et précise son facteur d'échelle. Par exemple :

processor p1 **scalingfactor** 3 ;

Si le facteur d'échelle est égal à 1, sa déclaration peut être omise ; ainsi,

processor Proc1 ;

déclare un processeur nommé Proc1 et dont le facteur d'échelle est 1.

4.2 Déclaration d'un réseau

La déclaration d'un réseau spécifie sa nature (CAN ou VAN) par le mot-clé correspondant, puis nomme ce réseau, et précise son facteur d'échelle. Par exemple :

van network r1 **scalingFactor** 4 ;

Comme pour la déclaration d'un réseau, la déclaration du facteur d'échelle peut être omise si il est égal à 1.

4.3 Déclaration d'une tâche indépendante

La forme générale de déclaration d'une tâche présente de nombreuses variantes, aussi nous les abordons de façon progressive en commençant par la déclaration d'une tâche indépendante.

task t1 **priority** 5 **duration** 100 .. 200 **offset** 30 **deadline** 400 **period** 600 **processor** p2 ;

La déclaration d'une tâche indépendante spécifie, dans cet ordre : son nom, sa priorité, son intervalle de durée, son offset, son échéance, sa période, et le nom du processeur sur lequel elle tourne.

Si l'offset est nul, sa déclaration peut être omise. Si l'échéance ne doit pas être vérifiée, sa déclaration peut être omise.

4.4 Déclaration d'une tâche dépendante d'une autre tâche

task t2 **priority** 5 **duration** 100 .. 200 **deadline** 400 **on task** t1 **every** 2 ;

La déclaration d'une tâche dépendante d'une autre tâche spécifie, dans cet ordre : son nom, sa priorité, son intervalle de durée, l'échéance relative à l'entité indépendante racine, la tâche dont elle dépend directement, et le nombre d'exécutions de la tâche dont elle dépend directement nécessaires pour lancer cette tâche.

Comme pour une tâche indépendante, la déclaration de l'échéance peut être omise si l'échéance n'a pas à être vérifiée. Par contre, comme notre algorithme exige pour ce type de tâche une synchronisation directe sans *idle-time* « Synchronisation directe », la déclaration d'un offset n'est pas autorisée.

Si le nombre d'exécutions de la tâche dont elle dépend directement nécessaires pour lancer cette tâche est égale à 1, la déclaration « **every** …» peut être omise.

4.5 Déclaration d'une tâche dépendante d'un message

task t2 **priority** 5 **duration** 100 .. 200 **deadline** 400 **on message** m **processor** p2 ;
La déclaration d'une tâche dépendante d'un message spécifie, dans cet ordre : son nom, sa priorité, son intervalle de durée, l'échéance relative à l'entité indépendante racine, le nom du message dont elle dépend directement, et le processeur sur lequel elle tourne.

Comme pour une tâche indépendante, la déclaration de l'échéance peut être omise si l'échéance n'a pas à être vérifiée. Par contre, comme notre algorithme exige pour ce type de tâche une synchronisation directe sans « idle-time », la déclaration d'un offset n'est pas autorisée.

4.6 Message CAN standard, CAN étendu, VAN

Le premier mot clé de la déclaration de tout message indique sa nature :

- le mot réservé **standard** introduit un message CAN standard ;
- le mot réservé **extended** introduit un message CAN étendu ;
- si la déclaration commence directement par le mot réservé **message**, il s'agit d'un message VAN.

Comme nous allons le voir, la déclaration du message indique aussi le réseau sur lequel il transite. Bien entendu, la déclaration d'un message CAN (standard ou étendu) n'est autorisée que sur un réseau CAN, et celle d'un message VAN sur un réseau VAN.

4.7 Déclaration d'un message indépendant

Comme pour la déclaration d'une tâche, la forme générale de déclaration d'un message présente de nombreuses variantes, aussi nous les abordons de façon progressive en commençant par la déclaration d'un message indépendant.

standard message m1 **network** net1 **length** 2 **priority** 1 **offset** 50 **deadline** 400 **period** 1000 ;

La déclaration d'un message indépendant spécifie, dans cet ordre : son type (ici un message CAN standard), son nom, le réseau sur lequel il transite, le nombre d'octets de données, sa priorité, son offset, son échéance, et sa période.

Pour un message CAN standard ou étendu, le nombre d'octets de données doit être compris entre 0 et 8. Notre outil calcule à partir de cette valeur et de la valeur de l'identificateur le nombre minimum et maximum de bits de la trame correspondante.

Pour un message VAN, le nombre d'octets de données doit être compris entre 0 et 28. Notre outil en déduit le nombre de bits de la trame correspondante (60+10*nombre d'octets).
Si l'offset est nul, sa déclaration peut être omise. Si l'échéance ne doit pas être vérifiée, sa déclaration peut être omise.

4.8 Déclaration d'un message dépendant d'une tâche

extended message m2 **network** net1 **length** 2 **priority** 12 **deadline** 400 **on task** t1 ;

La déclaration d'un message dépendant d'une tâche spécifie, dans cet ordre : son type (ici un message CAN étendu), son nom, le réseau sur lequel il transite, le nombre d'octets de données, sa priorité, son échéance, et le nom de la tâche de laquelle il dépend.

Comme pour un message indépendant, la déclaration de l'échéance peut être omise si l'échéance n'a pas à être vérifiée. Par contre, comme notre algorithme exige pour ce type de message une synchronisation directe, la déclaration d'un offset n'est pas autorisée.

4.9 Déclaration d'un message dépendant d'un message

message m3 **network** net2 **length** 20 **priority** 44 **deadline** 400 **on message** m2 ;

La déclaration d'un message dépendant d'un autre message spécifie, dans cet ordre : son type (ici un message VAN), son nom, le réseau sur lequel il transite, le nombre d'octets de données, sa priorité, son échéance, et le nom du message du quel il dépend.

Comme pour un message indépendant, la déclaration de l'échéance peut être omise si

l'échéance n'a pas à être vérifiée. Ici aussi la déclaration d'un offset n'est pas autorisée.

5 Fichiers intermédiaires

Lors de l'existence de l'option `-C` dans la ligne de commande, deux fichiers intermédiaires sont créés : le fichier des instances déployées et le fichier contenant des borne des temps de réponse.

5.1 Fichier des instances déployées

Le fichier ***trois_proc_activities.html*** contient les instances déployées.

#	Activity	Resource	Priority	Occurrence	Offset	minDur.	MaxDur.	Predecessor
0	t1	0	1	1	0	2000	2000	-1
1	t3	1	1	1	0	1320	1350	-1
2	t3	1	1	2	4800	1320	1350	-1
3	t5	2	1	1	0	1800	1820	-1
4	t5	2	1	2	4800	1800	1820	-1
5	m1	3	1	1	0	660	780	0
6	m2	3	3	1	0	660	780	1
7	m2	3	3	2	0	660	780	2
8	m3	3	2	1	0	660	780	3
9	m3	3	2	2	0	660	780	4
10	t2	0	2	1	0	1400	1440	8
11	t2	0	2	2	0	1400	1440	9
12	t4	1	2	1	0	1800	1830	5
13	t6	2	2	1	0	1400	1440	6
14	t6	2	2	2	0	1400	1440	7

5.2 Fichiers des temps de réponse

Le fichier ***trois_proc_raw_output.html*** contient les bornes minimum (BRT= $R^{\downarrow}$) et maximum (WRT= $R^{\uparrow}$) des temps de réponse pour chaque instance.

#	Activity	Resource	Occurrence	Offset	BRT	WRT
0	t1	p1	1	0	2000	2000
1	t3	p2	1	0	1320	1350
2	t3	p2	2	4800	6120	6738
3	t5	p3	1	0	1800	1820
4	t5	p3	2	4800	6600	6620
5	m1	net	1	0	2661	3558
6	m2	net	1	0	1980	2130
7	m2	net	2	0	6780	8180
8	m3	net	1	0	2640	3690

9	m3	net	2	0	7260	8178
10	t2	p1	1	0	4040	5130
11	t2	p1	2	0	8660	9618
12	t4	p2	1	0	4461	5388
13	t6	p3	1	0	3380	3570
14	t6	p3	2	0	8180	9620

6 Bilan

Le fichier contenant les résultats finaux ***trois_proc.html*** est créé comme suit :

Processors map

#	Name	ScaFactor
1	p1	4
2	p2	3
3	p3	4

Networks map

#	Name	Type	ScaFactor
1	net	CAN	12

Tasks map

#	Name	Processor	Priority	Offset	Min	Max	Period	Deadline	Dependence	Every
1	t1	1	1	0	500	500	2400	Unknown	---	1
2	t3	2	1	0	440	450	1600	Unknown	---	1
3	t5	3	1	0	450	455	1200	Unknown	---	1
4	t2	1	2	0	350	360	1200	Unknown	message #3	1
5	t4	2	2	0	600	610	3200	Unknown	message #1	1
6	t6	3	2	0	350	360	1200	Unknown	message #2	1

Messages map

#	Name	Network	Type	Priority	Byte	min--max	Offset	Period	Deadline	Dependence
1	m1	1	standard	1	1	55--65	0	800	Unknown	task #1
2	m2	1	standard	3	1	55--65	0	400	Unknown	task #2
3	m3	1	standard	2	1	55--65	0	400	Unknown	task #3

Final results map

#	Element	Resource	BRT	WRT	Period	Deadline	
1	t1	p1	500	500	2400	Unkown	

2	t2	p1	965	1283	1200	Unkown	
3	t3	p2	440	646	1600	Unkown	
4	t4	p2	1487	1796	3200	Unkown	
5	t5	p3	450	455	1200	Unkown	
6	t6	p3	845	1205	1200	Unkown	
7	m1	net	221	297	800	Unkown	
8	m2	net	165	282	400	Unkown	
9	m3	net	205	308	400	Unkown	

Response times are calculated with respect to offsets of independent tasks.