CISTER
Research Center in
Real-Time & Embedded
Computing Systems

Technical Report

CPMD-mindful task assignment for NPS-F

Geoffrey Nelissen*
Konstantinos Bletsas*
Joel Goossens

*CISTER Research Center

CISTER-TR-140711

2014/11

CPMD-mindful task assignment for NPS-F

Geoffrey Nelissen*, Konstantinos Bletsas*, Joel Goossens

*CISTER Research Center

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: grrpn@isep.ipp.pt, ksbs@isep.ipp.pt, joel.goossens@ulb.ac.be

http://www.cister.isep.ipp.pt

Abstract

The multiprocessor scheduling scheme NPS-F for sporadic tasks has a highutilisation bound and an overall number of preemptions bounded at design time. NPS-Fbinpacks tasks offline to as many servers as needed. At runtime, the scheduler ensuresthat each server is mapped to at most one of the m processors, at any instant. Whenscheduled, servers use EDF to select which of their tasks to run. Yet, unlike the overallnumber of preemptions, the migrations per se are not tightly bounded. Moreover, wecannot know a priori which task a server will be currently executing at the instant whenit migrates. This uncertainty complicates the estimation of cache-related preemptionand migration costs (CPMD), potentially resulting in their overestimation. Therefore,to simplify the CPMD estimation, we propose an amended bin-packing scheme forNPS-F allowing us (i) to identify at design time, which task migrates at which instantand (ii) bound a priori the number of migrating tasks, while preserving the utilisationbound of NPS-F.

Real-Time Syst (2014) 50:585–591
DOI 10.1007/s11241-014-9206-8

CPMD-mindful task assignment for NPS-F

Geoffrey Nelissen · Konstantinos Bletsas ·
Joël Goossens

Published online: 19 July 2014

Abstract The multiprocessor scheduling scheme NPS-F for sporadic tasks has a high utilisation bound and an overall number of preemptions bounded at design time. NPS-F binpacks tasks offline to as many servers as needed. At runtime, the scheduler ensures that each server is mapped to at most one of the m processors, at any instant. When scheduled, servers use EDF to select which of their tasks to run. Yet, unlike the overall number of preemptions, the migrations per se are not tightly bounded. Moreover, we cannot know a priori which task a server will be currently executing at the instant when it migrates. This uncertainty complicates the estimation of cache-related preemption and migration costs (CPMD), potentially resulting in their overestimation. Therefore, to simplify the CPMD estimation, we propose an amended bin-packing scheme for NPS-F allowing us (i) to identify at design time, which task migrates at which instant and (ii) bound a priori the number of migrating tasks, while preserving the utilisation bound of NPS-F.

Keywords Real-time scheduling · Multiprocessor · Semi-partitioned · NPS-F · Bin packing · CPMD estimation

G. Nelissen (✉) · K. Bletsas
CISTER/INESC-TEC, ISEP, Polytechnic Institute of Porto, Porto, Portugal
e-mail: grrpn@isep.ipp.pt

K. Bletsas
e-mail: ksbs@isep.ipp.pt

J. Goossens
PARTS Research Center, Université Libre de Bruxelles (ULB), Brussels, Belgium
e-mail: joel.goossens@ulb.ac.be

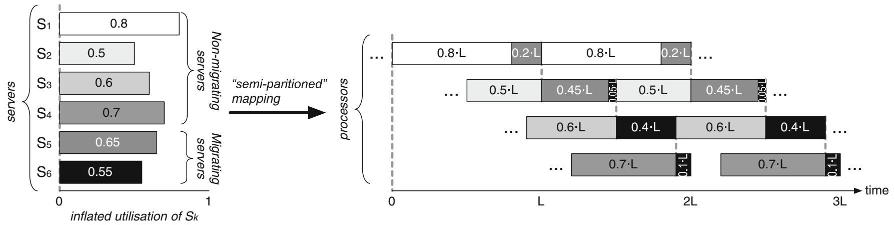

Fig. 1 Server schedule with the semi-partitioned mapping (with $L \overset{\text{def}}{=} \frac{\min_i\{T_i\}}{\delta}$)

1 Introduction and motivation

The multiprocessor scheduling scheme NPS-F Bletsas and Andersson (2011) for sporadic tasks has a high utilisation bound and an overall number of preemptions[1] bounded at design time. NPS-F bin-packs tasks *offline* to as many servers as needed. *At runtime*, the scheduler ensures that each server is mapped to at most one of the m processors, at any instant. When scheduled, servers use EDF to select which of their tasks to run.

Yet, unlike the overall number of preemptions, the migrations *per se* are not tightly bounded. Moreover, we cannot know *a priori* which task a server will be currently executing at the instant when it migrates. This uncertainty complicates the estimation of cache-related preemption and migration costs (CPMD), potentially resulting in their overestimation Bastoni and Brandenburg (2011). Therefore, to simplify the CPMD estimation, we propose an amended bin-packing scheme for NPS-F allowing us (i) to identify *at design time*, which task migrates at which instant and (ii) bound *a priori* the number of migrating tasks, while preserving the utilisation bound of NPS-F. The proposed solution is based on the fact that CPMD estimation is simplified if the migrating server serves only one task. This paper assumes the so-called *semi-partitioned mapping* of servers to processors (see Fig. 1). Namely, m servers use just one (respective) processor each, while the rest migrate.

2 A CPMD-mindful task assignment

NPS-F packs tasks into unit-capacity bins, corresponding to servers. Let us categorize those bins as either *migrating* or *non-migrating*. There are at most m non-migrating bins. A non-migrating bin is denoted by N_k with $1 \leq k \leq m$, and a migrating bin is denoted by M_ℓ with $1 \leq \ell \leq m'' - m$, where m'' is the total number of bins. Each sporadic implicit deadline task τ_i is characterized by an execution time C_i, a minimum inter-arrival time $T_i \geq C_i$ and a utilization $u_i \overset{\text{def}}{=} \frac{C_i}{T_i}$.

As the main contribution of this work, we propose a new family of bin-packing heuristics ensuring that each migrating bin may hold *only one task*. Hence, there are as many migrating bins as migrating tasks. Mapping-wise, the server corresponding to the non-migrating bin N_k is mapped to processor P_k, whereas servers corresponding to migrating bins switch between multiple processors (see Fig. 1).

[1] In the overall number of preemptions, we also count the preemptions ending in a migration.

```
1.  $\ell$:=0;
2.  for each task $\tau_i$ do //in any order
3.   assigned:=false;
4.   for each server N$_k$ with 1$\leq$k$\leq$m do //non-empty first
                                              //in any order
5.    if (U(N$_k$)+u$_i$$\leq$1) then
6.     N$_k$:=N$_k$$\cup\tau_i$;
7.     assigned:=true;
8.     break;
9.    end if
10.  end for
11.   if (not assigned) then
12.     M$_{\ell+1}$:={$\tau_i$};
13.     $\ell$:=$\ell$+1;
14.   end if
15.  end for
16.
17.  discard_empty_bins();
18.  m"=number_of_bins();
```

Fig. 2 New family of bin-packing heuristics for NPS-F

The schedulability condition for NPS-F (Eq. (5) in Bletsas and Andersson (2011)) can then be written as

$$\sum_{k=1}^{\min(m,m'')} \frac{(\delta+1)\times U(N_k)}{U(N_k)+\delta} + \sum_{1\leq\ell\leq m''-m} \frac{(\delta+1)\times U(M_\ell)}{U(M_\ell)+\delta} \leq m \tag{1}$$

where $U(S)$ is the sum of the utilizations of the component tasks of S and parameter $\delta \in N^+$ controls the migration frequency as explained in Bletsas and Andersson (2011) and shown on Fig. 1.

The new bin packing scheme (Fig. 2) proposed for NPS-F works as follows. Tasks are considered in any conceivable order. For each task τ_i of utilization u_i, an attempt is made to assign τ_i to any already populated *non-migrating* bin or, if this is not possible, to an empty *non-migrating* bin (lines 4–10). Note that First-Fit, Best-Fit and Worst-Fist are all examples of bin-packing heuristics complying with that definition. If there are already m populated non-migrating bins and τ_i cannot fit in any of them, τ_i is added to a new *migrating* bin (lines 11–14).

2.1 Utilization bound

To prove that the utilization bound of NPS-F is preserved, we rely on the property (proven as Lemma 1) that the average bin utilization exceeds 0.5. In our reasoning, we assume the bins obtained by application of the algorithm of Fig. 2:

Property 1 *For any non-migrating bin N_k and migrating bin M_ℓ: $U(N_k)+U(M_\ell) > 1$.*

Proof A task τ_i is tagged as migrating if and only if it does not fit in any non-migrating bin, subject to existing assignments. Hence $U(N_k)+u_i > 1$, $\forall 1 \leq k < m$ (line 5). Since τ_i is then assigned to M_ℓ (line 16), it holds that $U(M_\ell) = u_i$ and $U(N_k)+U(M_\ell) > 1$ for all k, before even assigning any remaining tasks. Applying this argument to every migrating task, we get $U(N_k)+U(M_\ell) > 1$ for any k and ℓ.

Corollary 1 *If for the total utilization U it holds that $U \leq m$, then $m'' < 2m$.*

Proof The total utilization U is equal to the sum of the task utilizations. Thus:

$$U = \sum_{k=1}^{\min(m,m'')} U(N_k) + \sum_{1 \le \ell \le m''-m} U(M_\ell)$$

The rest of the proof is by contradiction. Assume that $m'' \ge 2m$. Then,

$$U = \sum_{k=1}^{m} (U(N_k) + U(M_k)) + \sum_{\ell=m+1}^{(m''-m)} U(M_\ell) \ge \sum_{k=1}^{m} (U(N_k) + U(M_k))$$

and using Property 1 we get $U > m$, which contradicts the assumption that $U \le m$.

Property 2 *For any two non-migrating bins N_k and N_ℓ ($k \ne \ell$): $U(N_k)+U(N_\ell) > 1$.*

Proof The algorithm creates a new non-migrating bin N_ℓ if and only if the current task τ_i cannot fit in any of the existing bins. That is, $U(N_k) + u_i > 1$ for all $1 \le k < \ell$ (line 5). Since τ_i is then assigned to N_ℓ (line 14), it holds that $U(N_\ell) = u_i$ and $U(N_k) + U(N_\ell) > 1$ before even assigning any remaining tasks. Applying this argument to every newly created non-migrating bin, we get $U(N_k) + U(N_\ell) > 1$ for any two non-migrating bins N_k and N_ℓ with $k \ne \ell$.

Corollary 2 *There is at most one non-migrating bin N_k with $U(N_k) \le 0.5$.*

Proof By contradiction, assume that we have two or more non-migrating bins with a utilization smaller than or equal to 0.5. Then, for any two such bins N_k and N_ℓ, we have $U(N_k) + U(N_\ell) \le 1$, thereby implying a contradiction with Property 2.

Lemma 1 *If $0.5 < U \le m$, then $\frac{1}{m''}\left(\sum_{k=1}^{\min(m,m'')} U(N_k) + \sum_{1 \le \ell \le m''-m} U(M_\ell)\right) > \frac{1}{2}$.*

Proof By the algorithm of Fig. 2, migrating servers are created only if there are already m populated non-migrating servers and since by Corollary 1, the total number of servers is smaller than $2m$, there cannot be one non-migrating server and two or more migrating ones. Therefore, we analyze the three remaining cases:

1. If there is only one non-migrating and no migrating bin, then $U(N_1) = U > 0.5$.
2. If there are at least two non-migrating bins, then from Corollary 2, at most one of those bins has utilization smaller than 0.5. Assume, without loss of generality, that, if it exists, this particular bin is N_1. Two situations may arise:

(a) If there are no migrating bins, then there are m'' non-migrating bins and

$$\sum_{k=1}^{\min(m,m'')} U(N_k) + \sum_{1 \le \ell \le m''-m} U(M_\ell) = \sum_{k=1}^{m''} U(N_k)$$

From Property 2, we have that $U(N_1) + U(N_2) > 1$, thereby implying that

$$\sum_{k=1}^{\min(m,m'')} U(N_k) + \sum_{1\leq\ell\leq m''-m} U(M_\ell) > 1 + \sum_{k=3}^{m''} U(N_k)$$

Because by assumption $U(N_k) > 0.5$ for all $k > 1$, this expression yields

$$\sum_{k=1}^{\min(m,m'')} U(N_k) + \sum_{1\leq\ell\leq m''-m} U(M_\ell) > 1 + (m'' - 2) \times 0.5 = m'' \times 0.5$$

(b) If there is at least one migrating bin, then the algorithm implies that there are m non-migrating bins. Moreover, we know that $m'' < 2m$ (Corollary 1). Thus,

$$\sum_{k=1}^{\min(m,m'')} U(N_k) + \sum_{1\leq\ell\leq m''-m} U(M_\ell) = \sum_{p=1}^{m''-m} \left(U\left(N_p\right) + U\left(M_p\right)\right) + \sum_{q=m''-m+1}^{m} U\left(N_q\right)$$

From Property 1, we know that $U(N_k) + U(M_\ell) > 1$ $\forall k$ and $\forall \ell$, implying that

$$\sum_{k=1}^{\min(m,m'')} U(N_k) + \sum_{1\leq\ell\leq m''-m} U(M_\ell) > (m'' - m) + \sum_{q=m''-m+1}^{m} U\left(N_q\right)$$

and because by assumption $U\left(N_q\right) > 0.5$ for all $q > 1$, we have

$$\sum_{k=1}^{\min(m,m'')} U(N_k) + \sum_{1\leq\ell\leq m''-m} U(M_\ell) > (m'' - m) + (2m - m'') \times \frac{1}{2} = m'' \times \frac{1}{2}$$

Hence, in all three cases, the claim holds.

Theorem 1 *The utilization bound* UB *of NPS-F under the new bin-packing remains*

$$\text{UB} = \frac{2\delta + 1}{2\delta + 2} \times m$$

Proof Let S_1 to $S_{m''}$ be the servers output by the bin-packing. The key property for proving the utilization bound for the original NPS-F was that $\bar{U} > 0.5$ where $\bar{U} \overset{\text{def}}{=} \frac{1}{m''} \sum_{q=1}^{m''} U(S_q)$. This property is still true with the new bin-packing (Lemma 1). Hence, the utilization bound remains unchanged. A proof sketch is provided hereafter. The schedulability condition (Eq. 1) can be written as $\sum_{q=1}^{m''} I(U(S_q)) \leq m$, where $I\left(U\left(S_q\right)\right) \overset{\text{def}}{=} \frac{(\delta+1)U(S_q)}{U(S_q)+\delta}$. From Th. 2 in Bletsas and Andersson (2011),

$\sum_{q=1}^{m''} I\left(U\left(S_q\right)\right) \leq m'' I\left(\bar{U}\right)$. Hence, for schedulability, it suffices that $m'' I\left(\bar{U}\right) \leq m$ or equivalently $m''\bar{U} \leq m\frac{\bar{U}}{I\left(\bar{U}\right)}$. Yet, $\frac{\bar{U}}{I\left(\bar{U}\right)}$ is strictly increasing over [0.5, 1] and from Lemma 1, $\bar{U} > 0.5$. Thus, substituting in the schedulability condition, we get $m''\bar{U} \leq m\frac{0.5}{I(0.5)} = \frac{2\delta+1}{2\delta+2}m$. Hence, a sufficient schedulability condition is that $\left(m''\bar{U}\right)$, which corresponds to the task set utilization, is not greater than $\frac{2\delta+1}{2\delta+2}m$.

2.2 Number of migrating tasks

Theorem 2 *If $U \leq m$ then at most* $\max(0, \lceil 2 \times U \rceil - m - 1)$ *tasks migrate.*

Proof If $m'' \leq m$, then the algorithm implies that no migrating tasks exist. Else if $(m'' > m)$, the bin-packing of Fig. 2 ensures that there are m non-migrating servers and $(m'' - m)$ migrating servers with one task each. Because the total utilization U is given by the sum of the utilization of all servers, it holds by Lemma 1 that $U > 0.5 \times m'' \Rightarrow m'' < 2 \times U$. The number of migrating tasks n_{migr} is therefore $(m'' - m) < 2 \times U - m$. And because the number of migrating tasks is a non-negative integer but U is a real number, we get $n_{\text{migr}} \leq \max(0, \lceil 2 \times U \rceil - m - 1)$.

3 Conclusion

The alternative bin-packing eliminates the non-determinism in task migrations, while preserving all other aspects and useful theoretical properties of NPS-F. It also upper-bounds the number of migrating tasks *a priori* as a function of the task set utilization. Hence, the predictability and practical efficiency of NPS-F improve.

Acknowledgments This work was partially supported by National Funds through FCT (Portuguese Foundation for Science and Technology) and by ERDF (European Regional Development Fund) through COMPETE (Operational Programme 'Thematic Factors of Competitiveness'), within projects FCOMP-01-0124-FEDER-037281 (CISTER) and FCOMP-01-0124-FEDER-020447 (REGAIN); by National Funds through FCT and the EU ARTEMIS JU funding, within project ref. ARTEMIS/0003/2012, JU grant nr. 333053 (CONCERTO); and by the North Portugal Regional Operational Programme (ON.2 - O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by National Funds through FCT, within project ref. NORTE-07-0124-FEDER-000063 (BEST-CASE, New Frontiers).

References

Bletsas K, Andersson B (2011) Preemption-light multiprocessor scheduling of sporadic tasks with high utilisation bound. Real-Time Syst 47:319–355

Bastoni A, Brandenburg BB, Anderson JH (2011) Is semi-partitioned scheduling practical? In: ECRTS'11, Proceedings of the 23rd Euromicro Conference on Real-Time Systems, pp. 125–135

Geoffrey Nelissen earned his MSc degree in Electrical Engineering at the Université Libre de Bruxelles in 2008. After 4 years spent in the "Parallel Architectures for Real-Time Systems" research group at the Université Libre de Bruxelles, he received his Ph.D. in 2012. Since then, he has been working at the CISTER research unit in Porto, Portugal as a researcher associate in the area of multiprocessor real-time scheduling theory. His research interests include real-time scheduling theory, real-time operating systems, component based design and multi-processors/multi-cores architectures.

Konstantinos Bletsas (born in 1978 in Greece) has a Degree in Electronic and Computer Engineering (2002) from the Technical University of Crete (Chania, Greece) and a PhD in Computer Science (2007) from the University of York (UK). Since joining the CISTER/INESC-TEC, ISEP research unit (Porto, Portugal) in 2007, his main focus has been the design of multiprocessor hard-real time scheduling algorithms.

Joël Goossens is Professor at the Université Libre de Bruxelles (U.L.B.), since October 2006. He founded and chairs the "Parallel Architectures for Real-Time Systems" research group, U.L.B. Joël Goossens received his M.Sc. degree in computer science in 1992, his M.Sc. degree in network and management in 1993 and his Ph.D. degree in computer science in 1999, all from the Université Libre de Bruxelles, Belgium. He teaches algorithms and programming, operating systems and real-time scheduling. His main research interests are presently in real-time scheduling theory, real-time operating systems and embedded systems..