DISTRIBUTED RESOURCE MANAGEMENT IN NOC-BASED MPSOCS WITH DYNAMIC CLUSTER SIZES

***Guilherme M. Castilhos**, Marcelo Mandelli, Guilherme A. Madalozzo, Fernando G. Moraes*

guilherme.castilhos@acad.pucrs.br

Hardware Design Support Group
www.inf.pucrs.br/~gaph

PUCRS

Outline

- Introduction
- Architectural Assumptions
- Distributed Resource Management
- Results
- Conclusion

Introduction

- It is expected that the number of cores continue to increase in MPSoCs: thousand of cores within a decade
- The constant growth in the number of cores implies an important issue: scalability
- Despite the scalability offered by NoCs and distributed processing, the MPSoC resources must be managed to deliver the expected performance

Introduction

An alternative to ensure scalability is to decentralize or distribute the management functions of the system.

Goals

This paper has two main goals:

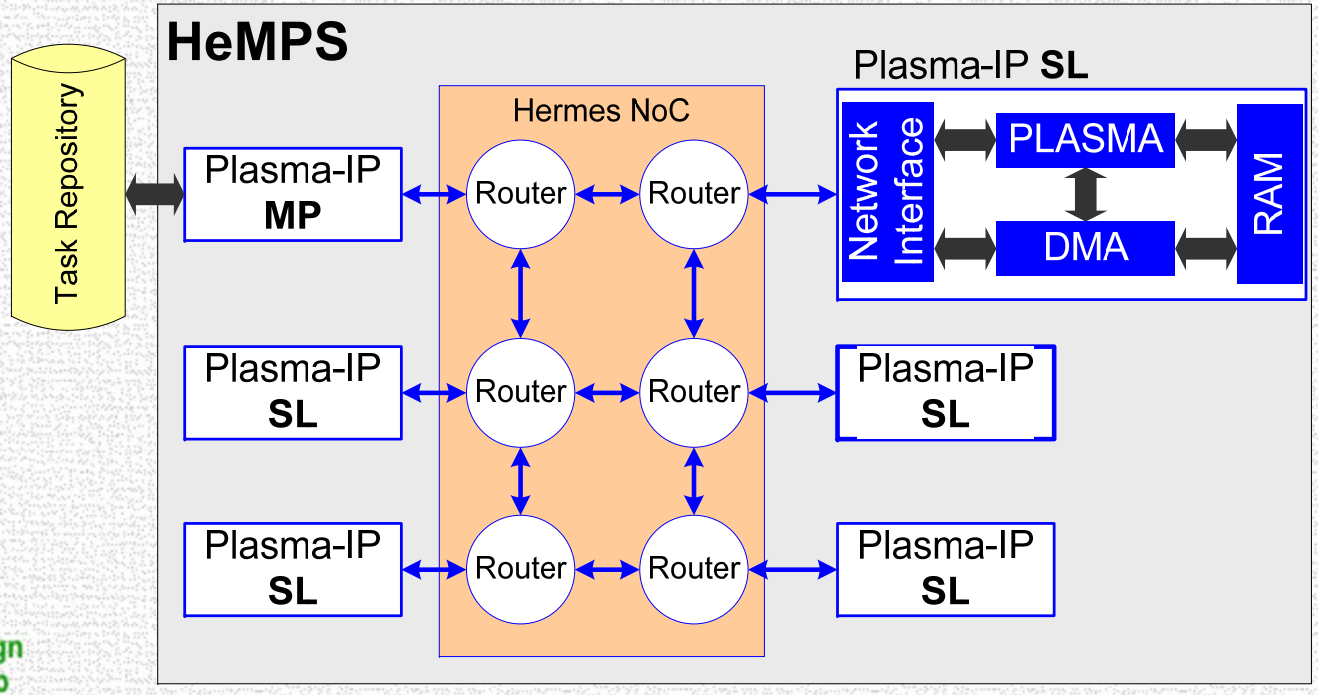
- Deploy a distributed resource management architecture for NoC-based MPSoCs with dynamic cluster sizes, using a cycle accurate SystemC model.
- Compare the performance of the centralized versus distributed approaches, using as cost function the total execution time, number of hops between taks and reclustering method.

Outline

- Introduction
- Architectural Assumptions
- Distributed Resource Management
- Results
- Conclusio`n

Architectural Assumptions

- Homogeneous NoC-Based MPSoC
- 2D-mesh NoC.
- Each PE contains a Plasma-IP.
- An external memory, named *task repository*.

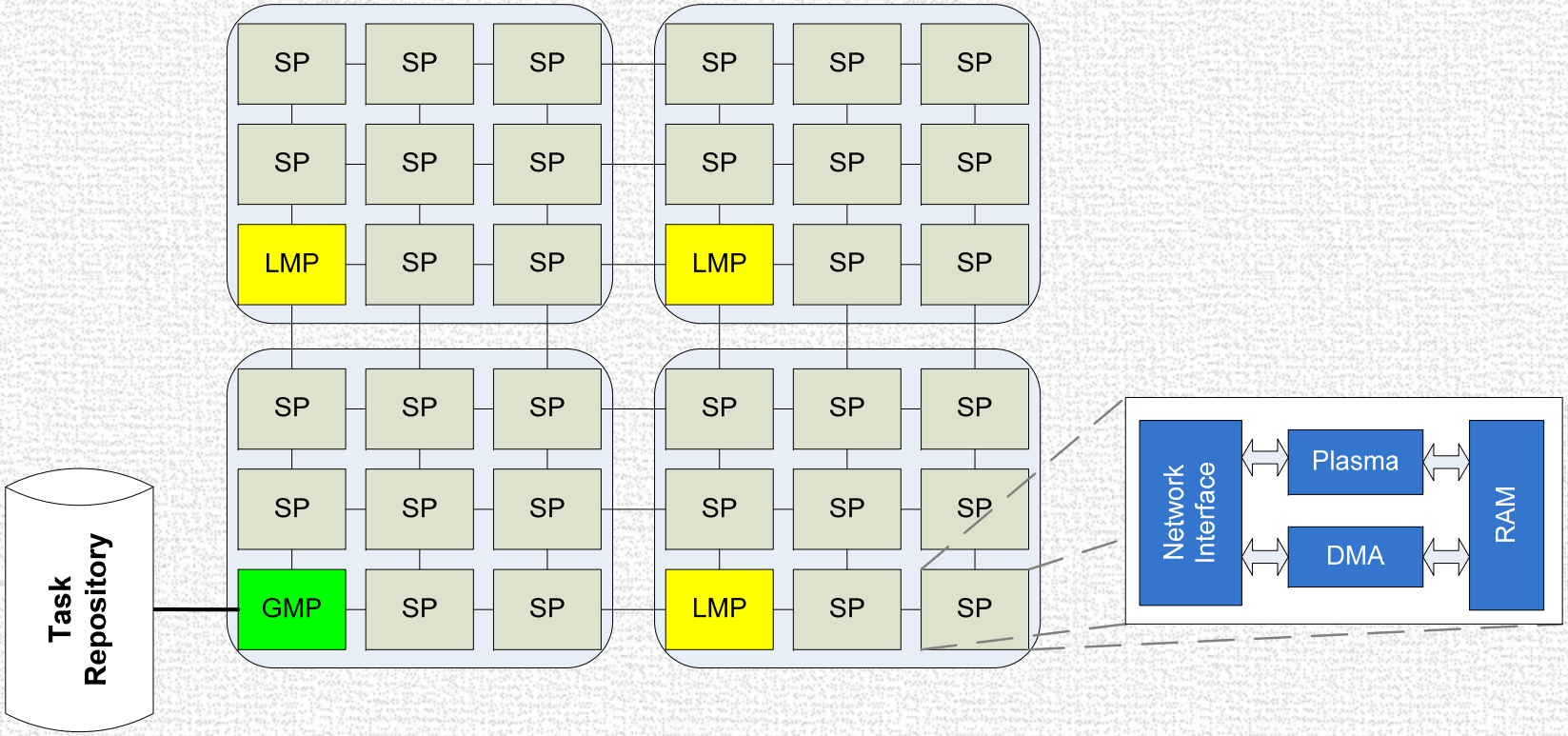


Outline

- Introduction
- Architectural Assumptions
- Distributed Resource Management
- Results
- Conclusion

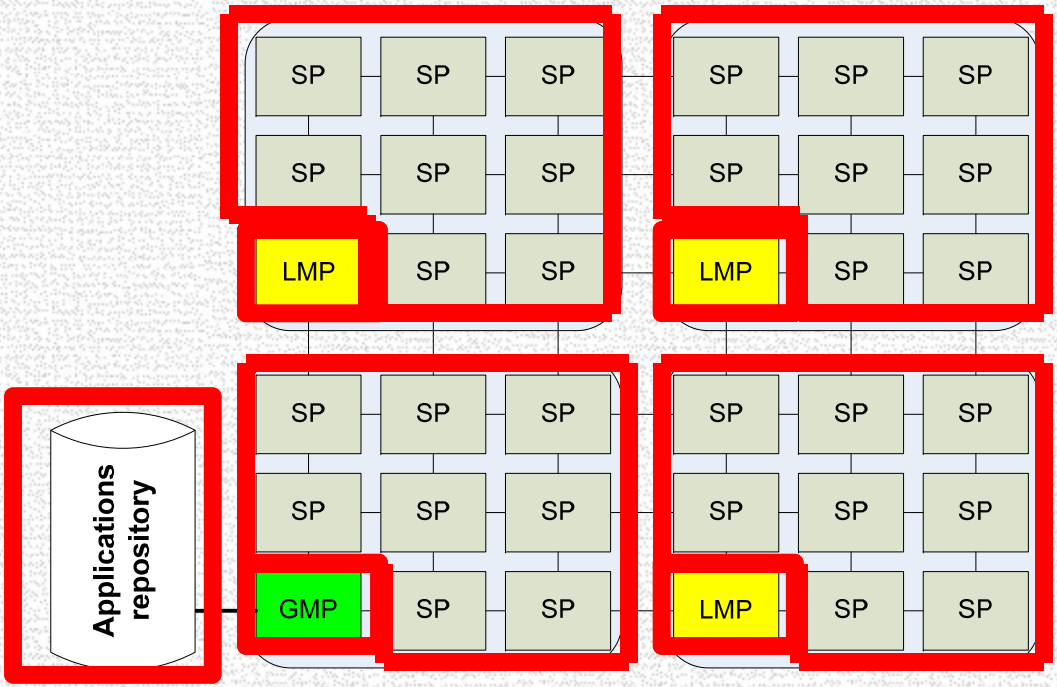
Distributed Resource Management

- The distributed resource management assumes an MPSoC divided in *n* regions, named clusters.



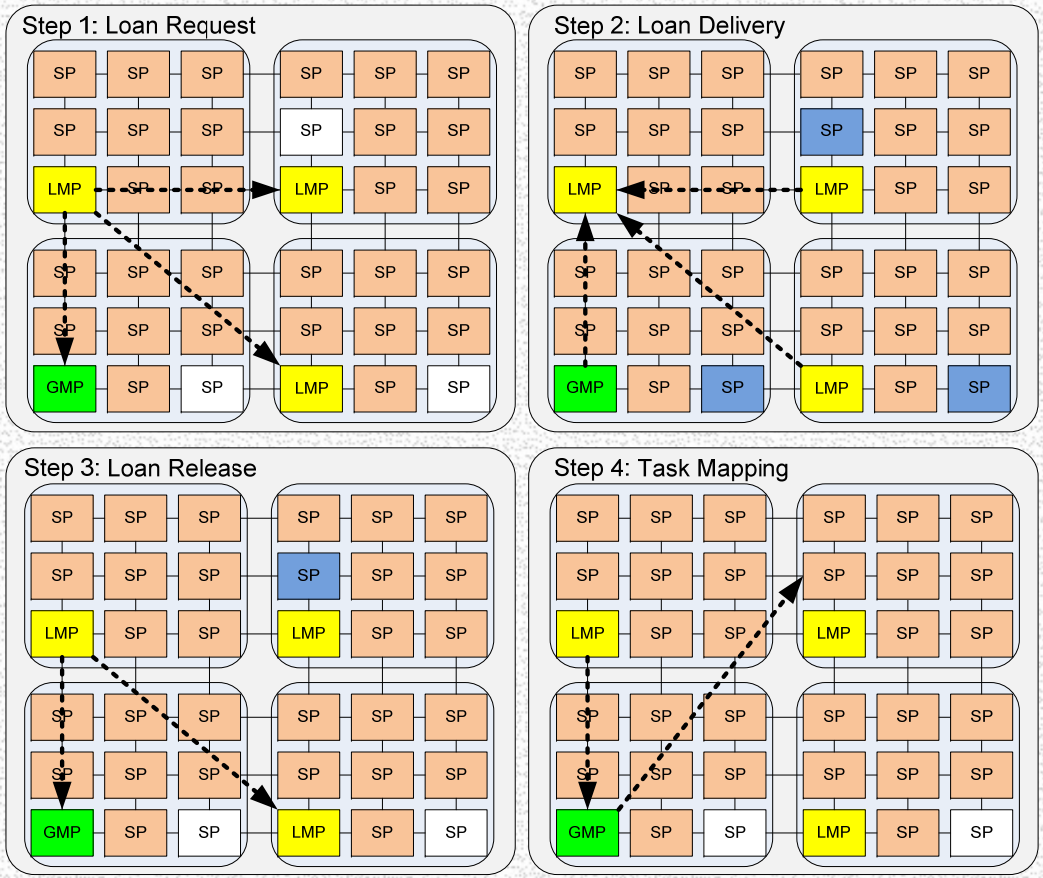
Distributed Resource Management

- The MPSoC contains three types of PEs
  - Global Master PE (GMP) :
    - contains all functions of the LMP, and functions related to the overall system management
    - only PE with access to the external devices (e.g the application repository)
  - Local Master PEs (LMP) : responsible to control the cluster
  - Slave PEs (SP) : responsible for task execution



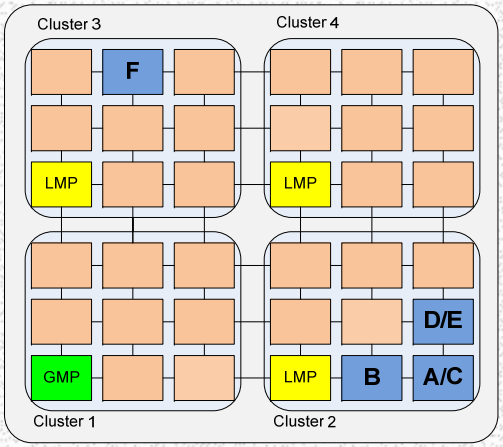
Distributed Mapping

- According to user requests, new applications can be loaded at runtime.
- If an application does not fit in a given cluster, the LMP of the cluster may request resources to neighbor clusters

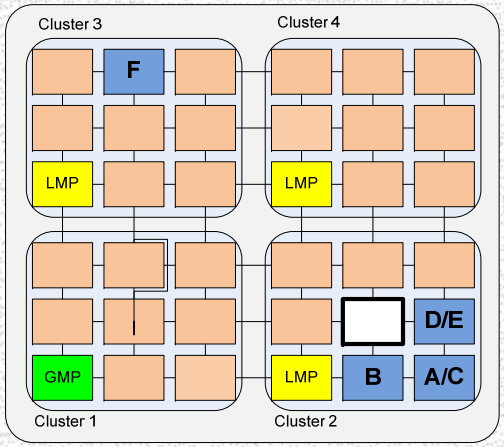


Task Migration

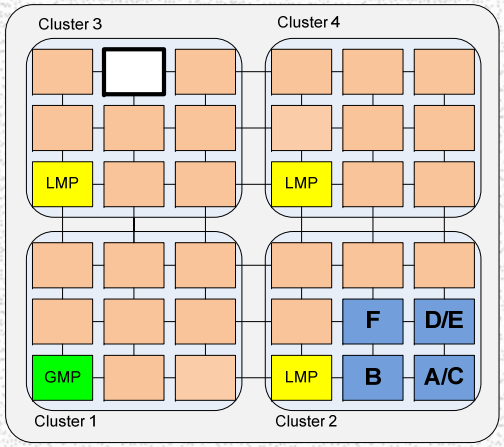
- Important features of the task migration:
  - Tasks may be migrated at any moment;
  - Complete task migration (context, code and data);
  - In-order message delivery.



(a)



(b)



(c)

Task Migration

- Drawbacks by increasing the hop number between tasks:
  - Performance degradation of the application, due to its fragmentation;
  - Increased data traffic volume in the NoC;
  - Increased communication energy, since it is proportional to the number of traversed hops.

Outline

- Introduction
- Architectural Assumptions
- Distributed Resource Management
- Results
- Conclusion

Results

- Results were obtained using three benchmarks:
  - *MPEG*;
  - *Multispec* image analysis, evaluate the similarity between two images using different frequencies;
  - *Synthetic*.

- The experiments use the HeMPS MPSoC, described in RTL cycle accurate modeling (SystemC).

Results

Total Execution Time

- This table presents the execution time normalized w.r.t the centralized management in a 12x12 MPSoC, with an MPSoC load equal to 75%.
- As can be observed, the distributed management leads to a total execution time reduction. The smaller reduction observed in the MPEG benchmark is due to its periodicity feature.
- The reduction in the total execution time reduction comes from:
  - Several PEs execute the task mapping in parallel;
  - Each manager treats a smaller number of control packets compared to the centralized approach.

Cluster Size	Nb of Clusters	Benchmark		
		MPEG	Synthetic	Multispec
12x12	1	1,00	1,00	1,00
12x6	2	0,94	0,78	0,77
6x6	4	0,90	0,67	0,63
6x4	6	0,88	0,58	0,71
6x3	8	0,86	0,57	0,56
4x4	9	0,88	0,58	0,52
3x3	16	0,87	0,54	0,49

Results

Hop Number

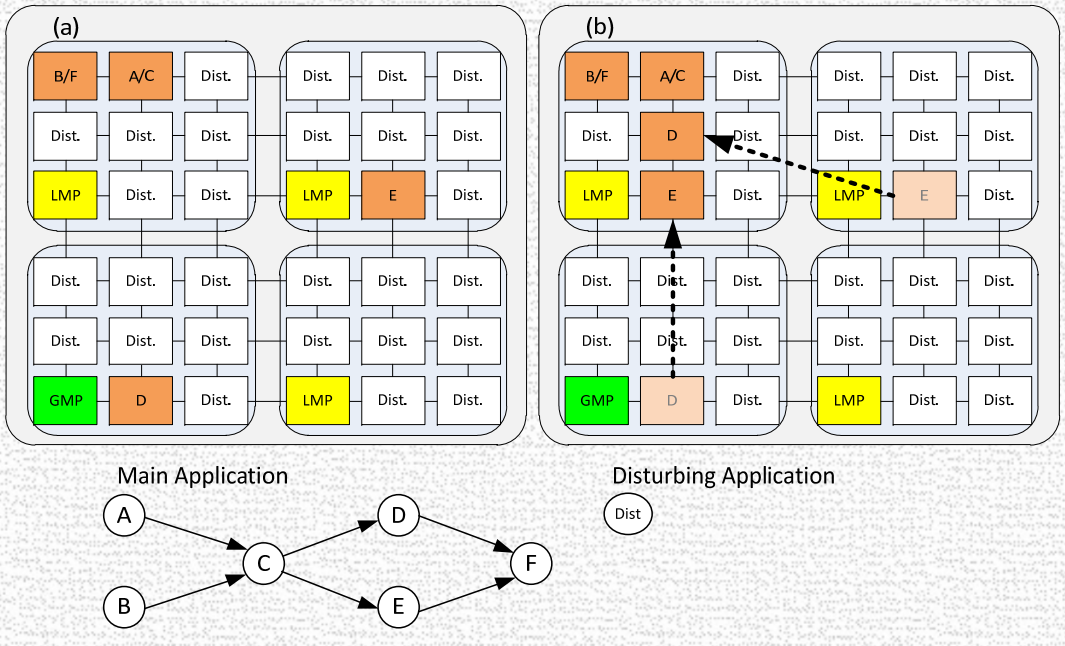
- The evaluation of the average hop number is a key parameter to evaluate the mapping quality.
- Higher values of hop number on the other side penalize the performance of applications, since disturbing traffic may interfere in the communication.

Cluster Size	Nb of Clusters	MPEG (5 tasks)			
		min	avg	max	Std dev
12x12	1	4	5.14	14	2.17
6x3	8	4	4.05	6	0.31
4x4	9	4	4.45	8	1.15
Cluster Size	**Nb of Clusters**	**Synthetic (6 tasks)**			
		min	avg	max	Std dev
12x12	1	2	9.66	38	7.98
6x3	8	2	5.65	10	1.51
4x4	9	2	6.00	10	1.65

Results

Reclustering

- Evaluates monitoring with task migration. Two scenarios were evaluated:
  - *main application* with disturbing applications, without task migration;
  - *main application* with disturbing applications and two task migrations (tasks E and D).
- The total execution time presents a reduction of 2.67%, considering two task migrations. Therefore, even if task migration momently increases the execution time, the final result is an improvement in the overall performance.



Outline

- Introduction
- Architectural Assumptions
- Distributed Resource Management
- Results
- Conclusion

Conclusion

- The proposed management technique reduced the distance among tasks, resulting in an important reduction in the total execution time.

- In addition, it was shown that monitoring coupled to task migration is an effective adaptive method to improve the system performance.

Thank You

Questions?

{ guilherme.castilhos, marcelo.mandelli, guilherme.madalozzo }@acad.pucrs.br

www.inf.pucrs.br/gaph