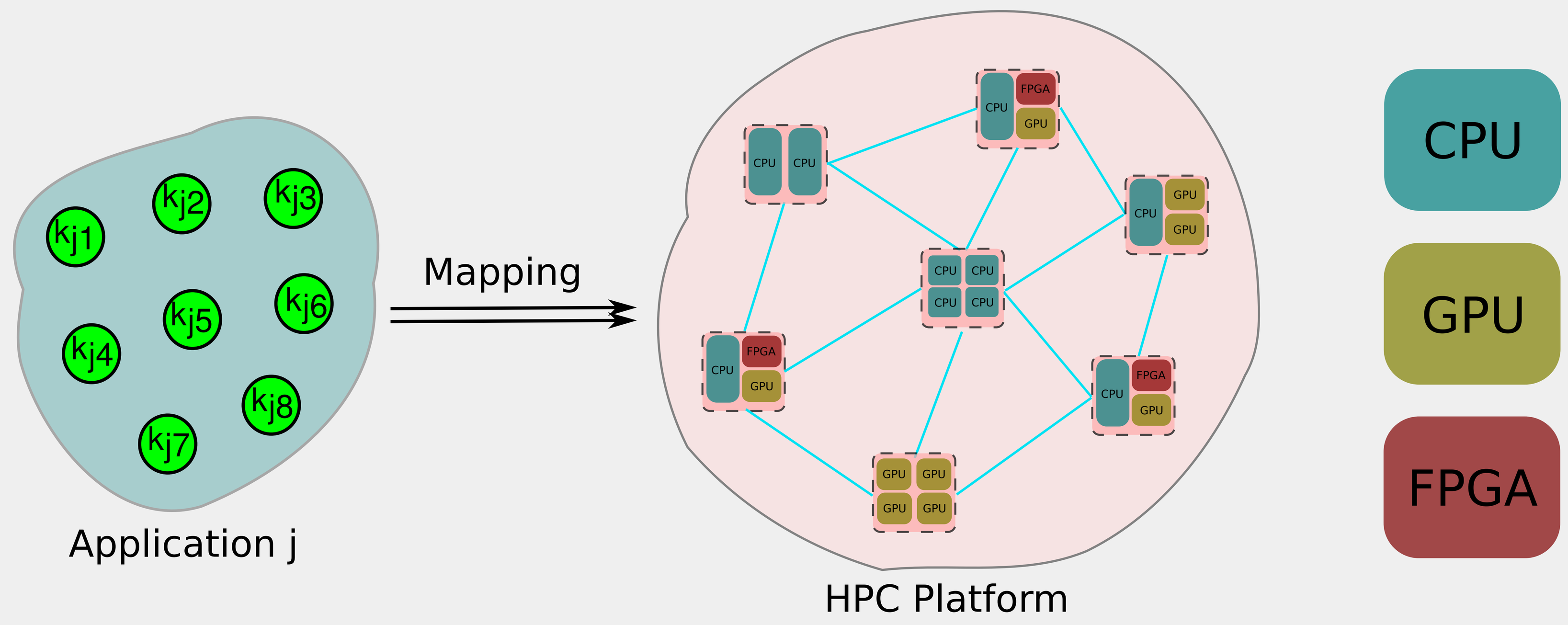
Techniques for Dynamic Workload Partitioning in High-Performance Heterogeneous Computing Platforms

INFIERI Summer School 2017
23/01 to 03/02 2017
University of São Paulo, Brazil

THALES
Wilder Lopes
wilderlopes@gmail.com

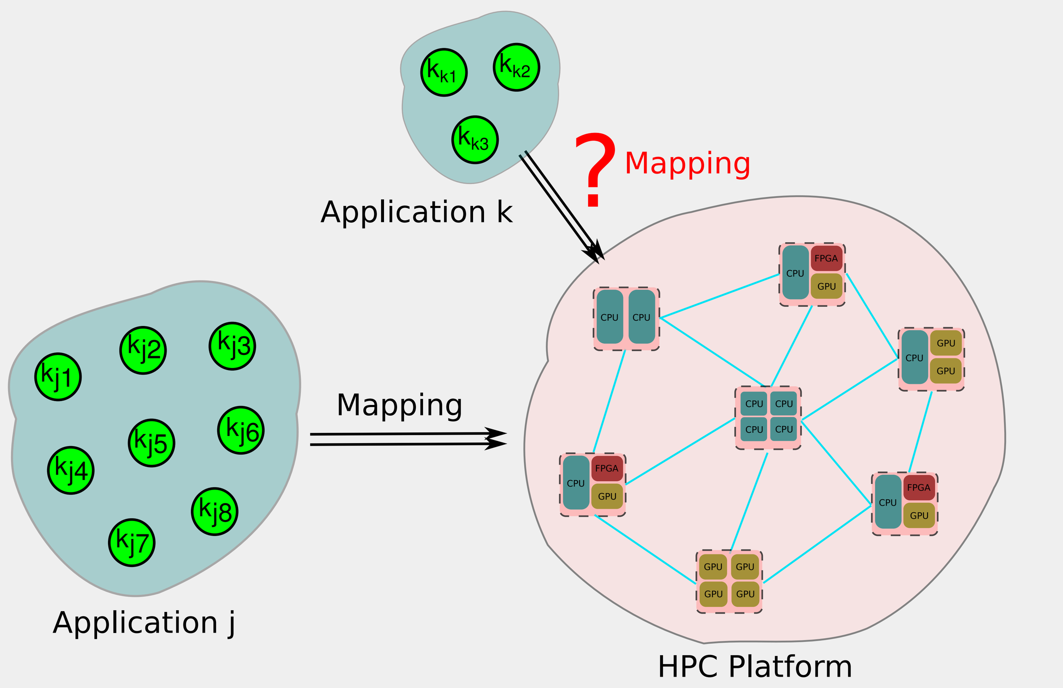
Introduction

- The proper allocation/mapping of data-parallel processes (kernels) in a High-Performance Computing (HPC) platform is crucial to exploit the system full potential
- Dynamic Heterogeneous Platforms: type and amount of available devices **may change at runtime**
- Applications requirements (Quality of Service – Qos): **also may change at runtime**

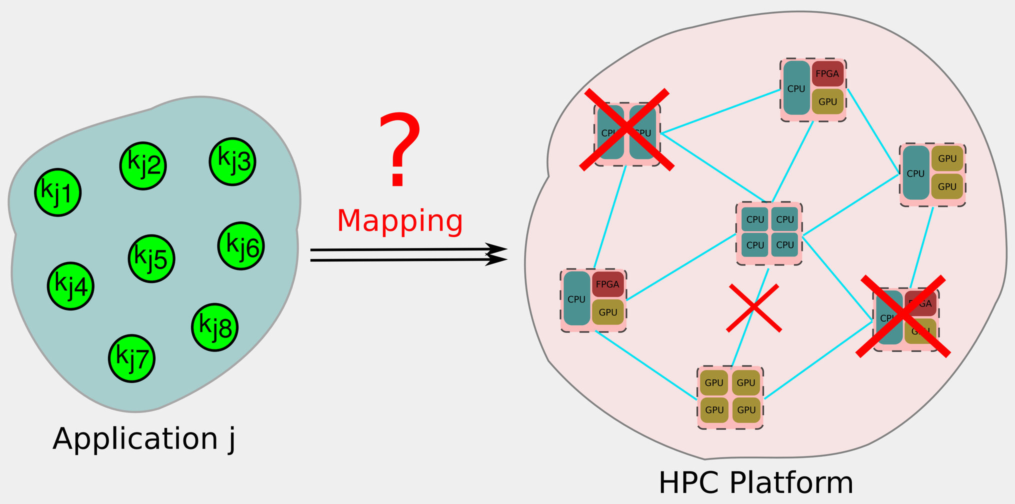


GOAL

Develop a system manager able to sense and react at runtime to variations in the High-Performance Heterogeneous Computing platform as well as in the QoS requirements



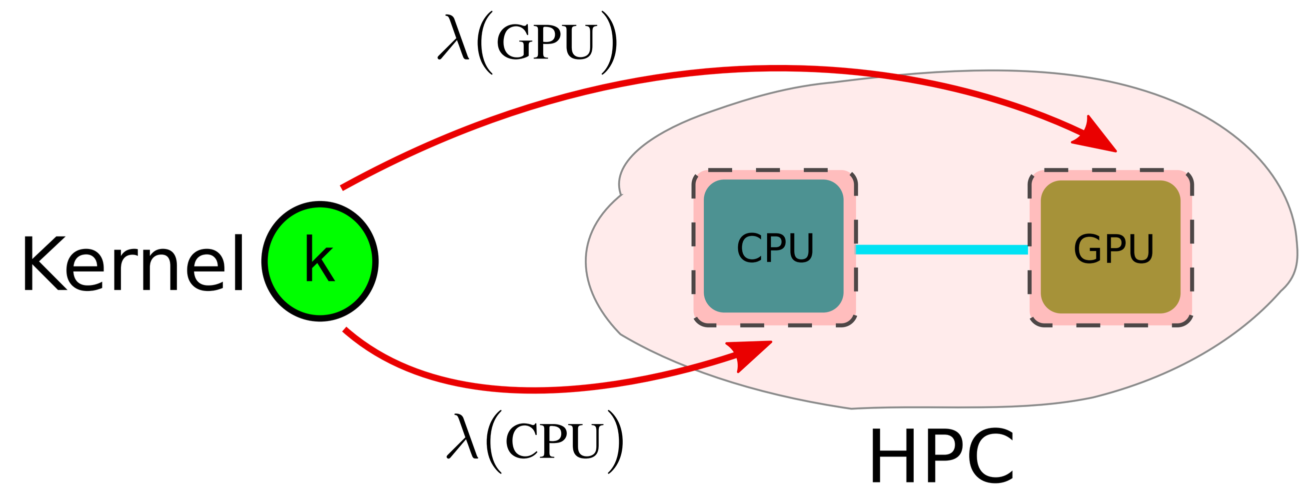
Scenario 1: new application needs to be mapped



Scenario 2: resource/network failures

Problem Formulation

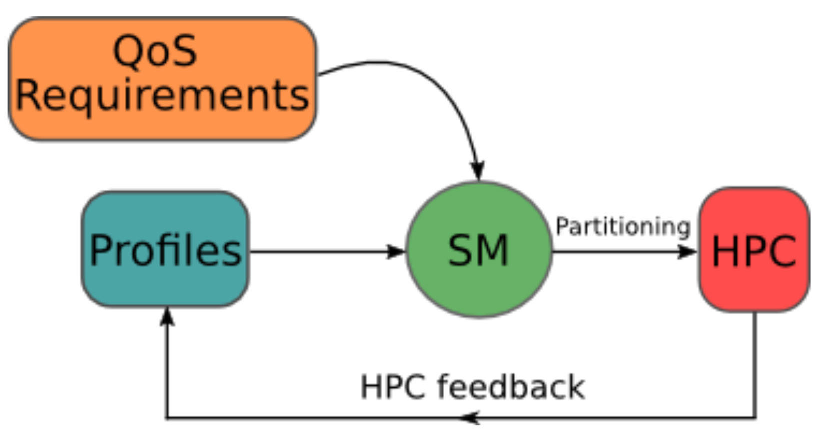
- **Simple scenario**: single-kernel data-parallel application → preliminary study
- Only two devices: **CPU and GPU**
- Workload partitioning ⇔ Data partitioning
- Estimate the **amount of data** to be allocated to each device
- **Definition:** **Best partitioning** *is the one that enforces minimization of execution time*



$\lambda(\mathrm{CPU})$ → amount of data in CPU
$\lambda(\mathrm{GPU})$ → amount of data in GPU

$$\lambda_k = \begin{bmatrix} \lambda(\mathrm{CPU}) \\ \lambda(\mathrm{GPU}) \end{bmatrix}$$

Design of the System Manager (SM)



- It is assumed a **performance profile** of the kernel in each device is available → Matrix R
- **HPC feedback:** measurement of devices performances in order to update profiles in R
- **QoS requirements:** application dependent
- SM uses those to estimate the workload partitioning that minimizes kernel execution time

Mathematical Representation

error e_k: **Measure of how well the resources can be combined to achieve the QoS requirements**

$$e_k = \left(d - \sum_{j}^{N} r_j \lambda(j) \right) = (d - R\lambda_k)$$

d → QoS requirements vector
R → resources matrix
r_j → column of R

The Intelligence embedded in the SM

Workload Partitioning → Minimization Problem

$$\min_{\lambda} J(\lambda) = \mathrm{E}|e_k|^2 = \mathrm{E}\left|\boldsymbol{d} - \boldsymbol{R}\lambda_k\right|^2$$

$$\text{subject to} \quad \lambda(j) \geq 0, j \in [1, N], \ \mathbb{1}^T \lambda_k = \sum_{j=1}^{N} \lambda(j) = 1$$

Adaptive Filter - Reweighted LMS

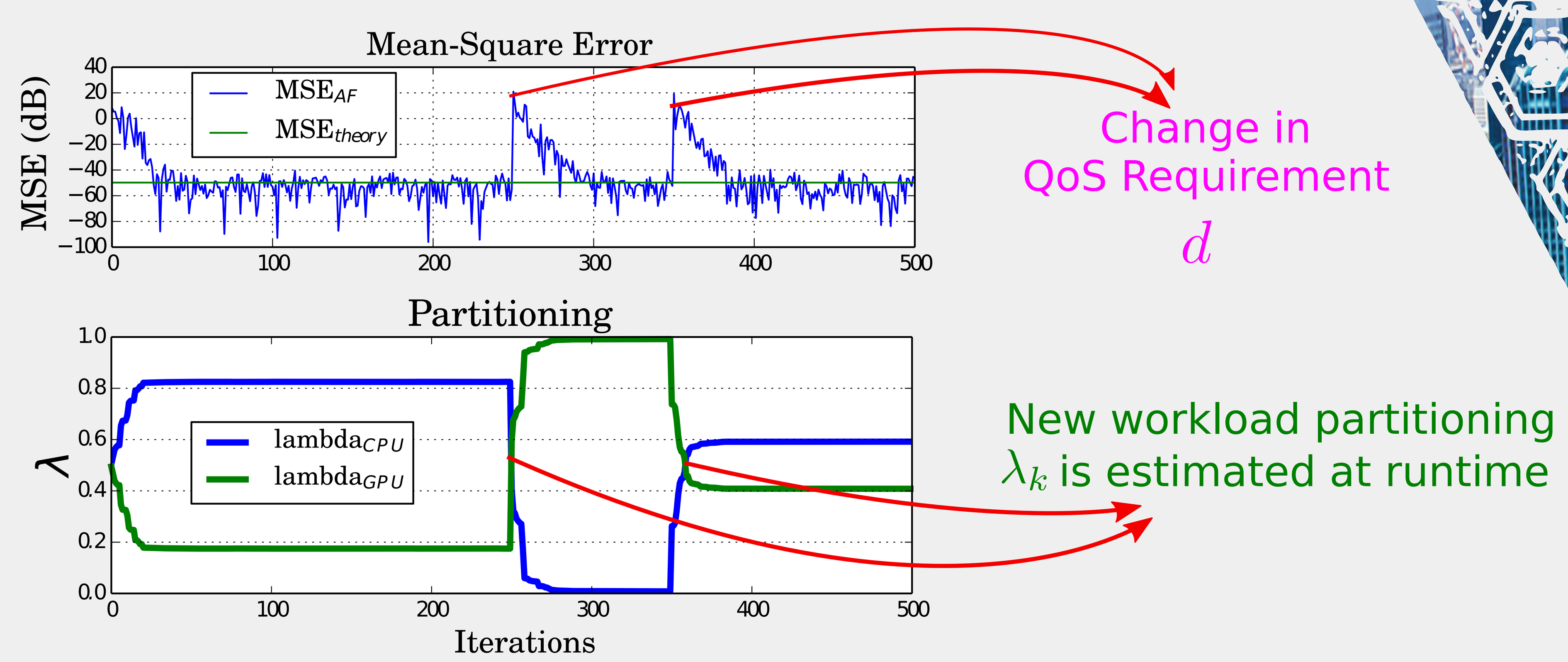
- **Naturally fit for real-time estimation**: no need for previous training
- Able to **track variations** in the HPC resources (matrix R) and in the Qos requirements (vector d)
- **Easy to be scaled up** towards several applications (kernels) and computing devices

$$\lambda_{k,i} = P[\lambda_{k,i-1} + \mu D_\lambda R_i^T e_k(i)] + F$$

$P = I - \frac{1}{N}\mathbb{1}\mathbb{1}^T$ $F = \frac{1}{N}\mathbb{1}$ D_λ= diag{entries of λ }

Results

- **Test: variation in the QoS requirements**
  d changes while R remains fix
- Adaptive Filter is able to provide a **new workload partitioning at runtime → Fast reaction**



The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007-2013/ under REA grant agreement No. [317446] INFIERI "INtelligent Fast Interconnected and Efficient Devices for Frontier Exploitation in Research and Industry"