

PARALLEL APPLICATION PROGRAMMING FOR EMBEDDED SYSTEMS :  
PERFORMANCE CHARACTERIZATION

**Context and Proposed Work:**

The use of parallel programming models is very common in the field of high-performance computing (HPC), and usually leverages supercomputers, *i.e.*, a set of machines (often called *cluster*) dedicated to intensive/scientific computation, with a fast interconnection network, that is, with high throughput and very low latency.

The advent of chip multi-processors, *a.k.a.*, multicore chips (CMPs), from servers, to workstations, down to high-end embedded systems, has made parallel and concurrent programming more interesting to a wider audience. However, parallel and distributed programming which ties multiple workstations or embedded devices has been less explored, in part because of the assumptions which are generally made in a parallel application: homogeneous latency and throughput, identical compute nodes, *etc.*

This internship aims at leveraging high-end embedded systems, for example Raspberry Pi 4 boards (RPi4), and after having connected them to the same network, distribute computations on the resulting parallel embedded computer. Applications of interest include linear algebra kernels, which are the basis for many tasks tied to computer vision, image processing, machine learning and neural network processing, *etc.*

The intern will have several tasks to perform, from performing multiple configurations of the network (wired-only, wireless-only, hybrid), and compare the results to what can be obtained on a single high-end workstation, in terms of pure performance (FLOP/s) as well as energy/power efficiency (FLOPs/W).

**Technical skills required:** C or C++ programming; some prior knowledge of MPI and/or OpenMP would be greatly appreciated. A basic understanding of UNIX/Linux would go a long way.

**Advisement team:** J.Lorandel, S.Zuckerman

PROGRAMMATION D'APPLICATIONS PARALLÈLES POUR SYSTÈMES EMBARQUÉS :  
CARACTÉRISATION DES PERFORMANCES

**Contexte et travail proposé :**

L'utilisation de modèles de programmation parallèle est quelque chose de commun dans le domaine du calcul haute-performance, qui utilise généralement des supercalculateurs, c'est-à-dire un ensemble de machines dédiées au calcul, reliées par un réseau d'interconnexion ultra-rapide (fournissant une latence très basse et un débit très élevé).

L'avènement des puces multi-cœurs tant pour les stations de travail que pour les systèmes embarqués de haute-performance a accru l'intérêt d'un public plus large envers la programmation parallèle et concurrente. Cependant, la programmation parallèle et distribuée reliant plusieurs stations de travail ou objets embarqués, est moins explorée, notamment à cause des hypothèses qui sont faites pour faire du calcul parallèle : latences et débits homogènes, nœuds de calcul identiques, etc.

Ce stage consiste à utiliser des systèmes embarqués haut de gamme, par exemple des Raspberry Pi 4 (RPi4), et après les avoir connectés en réseau, distribuer certains calculs dessus, comme par exemple des noyaux de calcul liés à l'algèbre linéaire qui sont particulièrement prisés dans les environnements d'apprentissage (*machine learning*), de traitement d'image, etc.

Le ou la stagiaire aura aussi pour tâche de comparer la performance obtenue sur un système reliant plusieurs cartes RPi4 dans différentes configurations (purement filaire, purement sans-fil, ou hybride), et comparer celle-ci aux résultats obtenus sur une ou plusieurs stations de travail haut-de-gamme, tant en termes de performances brutes (FLOP/s) qu'en termes de performance énergétique (FLOPs/W).

**Compétences requises :** programmation C ou C++, une connaissance de MPI et OpenMP sont un plus. Utilisation basique d'Unix/Linux.

**Référents :** J.Lorandel, S.Zuckerman