

oL9 - OpenCL

Programmation parallèle avec OpenCL

Objectifs

- Apprendre la programmation parallèle avec OpenCL
- Savoir ce qu'il (ne) faut (pas) attendre de la programmation parallèle
- Comprendre le multithreading lourd et comment il est mis en correspondance avec le matériel
- Mesurer les performances du code OpenCL, localiser et résoudre les blocages
- Écrire un code OpenCL efficace

Les exercices seront exécutés sur des CPU multi-core avec GPU nVidia fonctionnant sous Linux

Pré-requis

- Bonne connaissance du langage C

Environnement du cours

- Cours théorique
 - Support de cours au format PDF (en anglais)
 - Cours dispensé via le système de visioconférence Teams
 - Le formateur répond aux questions des stagiaires en direct pendant la formation et fournit une assistance technique et pédagogique
- Activités pratiques
 - Les activités pratiques représentent de 40% à 50% de la durée du cours
 - Elles permettent de valider ou compléter les connaissances acquises pendant le cours théorique.
 - Un PC Linux en ligne par stagiaire pour les activités pratiques
 - Exemples de code, exercices et solutions
 - Le formateur a accès aux PC en ligne des stagiaires pour l'assistance technique et pédagogique
 - Certains travaux pratiques peuvent être réalisés entre les sessions et sont vérifiés par le formateur lors de la session suivante
- Une machine virtuelle préconfigurée téléchargeable pour les activités pratiques après le cours
- Au début de chaque demi-journée une période est réservée à une interaction avec les stagiaires pour s'assurer que le cours répond à leurs attentes et l'adapter si nécessaire

Audience visée

- Tout ingénieur ou technicien en systèmes embarqués possédant les prérequis ci-dessus

Durée

- Totale : 24 heures
- 4 sessions de 6 heures chacune (hors temps de pause)
- De 40% à 50% du temps de formation est consacré aux activités pratiques
- Certains travaux pratiques peuvent être réalisés entre les sessions et sont vérifiés par le formateur lors de la session suivante

Course Outline

Première session

Introduction to OpenCL

- History
 - OpenCL 1.2
 - OpenCL 2.2
 - OpenCP/EP (Embedded Profile)
- Design goals of OpenCL
 - CPUs, GPUs and GPGPUs
 - Data-parallel and Task-parallel
 - Hardware related and portable
- Terminology
 - Host / Device
 - Memory model
 - Execution Model

The OpenCL Architecture

- The OpenCL Architecture
 - Platform Model
 - Execution Model
 - Memory Model
 - Programming Model
- The OpenCL Software Stack
- Example

Exercise : Installation and test of the OpenCL SDK

Deuxième session

The OpenCL Host API

- Platform layer
 - Querying and selecting devices
 - Managing compute devices
 - Managing computing contexts and queues
 - The host objects: program, kernel, buffer, image

Exercise : Write a platform discovery and analysis program (displaying CPUs, GPUs, versions...)

- Runtime
 - Managing resources
 - Managing memory domains
 - Executing compute kernels

Exercise : Write an image loader program, transferring image to/from compute devices

- Compiler
 - The OpenCL C programming language
 - Online compilation
 - Offline compilation

The Basic OpenCL Execution Model

- How code is executed on hardware
 - Compute kernel

- Compute program
- Application queues
- OpenCL Data-parallel execution
 - N-dimensional computation domains
 - Work-items and work-groups
 - Synchronization and communication in a work-group
 - Mapping global work size to work-groups
 - Parallel execution of work-groups

Exercise : Compile and execute a program to square an array on the platform computing nodes

Troisième session

The OpenCL Programming Language

- Restrictions from C99
- Data types
 - Scalar
 - Vector
 - Structs and pointers
 - Type-conversion functions
 - Image types

Exercise : Rewrite the square program to use vector operations

- Required built-in functions
 - Work-item functions
 - Math and relational
 - Input/output
 - Geometric functions
 - Synchronization
- Optional features
 - Atomics
 - Rounding modes

Exercise : Write and execute an image manipulation program (Blur filter)

Quatrième session

Advanced OpenCL Execution modes

- Profiling

Exercise : Enhance the image manipulation program to measure kernel computation time

- The OpenCL Memory Model
 - Global Memory
 - Local Memory
 - Private Memory
- OpenCL Task-parallel execution
 - Optional OpenCL feature
 - Native work-items

Exercise : Simulate the N-Body problem, displaying data using OpenGL

Efficient OpenCL

- When (not) to use OpenCL
- Code design guidelines
- Explicit vectorization

Exercise : Explore vectorisation on an image rotation kernel

- Memory latency and access patterns
 - ALU latency
 - Using local memory

Exercise : Enhance the Blur filter program to investigate memory optimisations

- Synchronizing threads
- Warps/Wavefronts, work groups, and GPU cores