

Aceleración y Paralelización de código de MATLAB.

Curso presencial

Descripción del curso:

Este curso de dos días cubre diferentes técnicas para la aceleración de código MATLAB®. Se muestra como identificar y eliminar cuellos de botella utilizando técnicas como reserva de memoria y vectorización. Además, aprenderá a compilar código MATLAB en ficheros MEX usando MATLAB Coder™. También aprenderá a aprovechar los múltiples núcleos de su ordenador mediante bucles for paralelizados con Parallel Computing Toolbox™, y llevarlo a múltiples ordenadores usando MATLAB Distributed Computing Server™. Todos estos conceptos se explorarán durante el curso.

Pre-requisitos:

Fundamentos de MATLAB, o experiencia equivalente utilizando MATLAB.

Imparte: Emmanuel Olivar
Ingeniero en Sistemas Electrónicos Industriales



Emmanuel Olivar Domínguez, egresado de la Universidad Autónoma de la Ciudad de México (UACM), cuenta con amplia experiencia en el mantenimiento preventivo y correctivo de equipo electrónico industrial y comercial, además de utilizar tecnología MathWorks (MATLAB y Simulink) por más de cinco años. Se especializa en las áreas de control e instrumentación con la finalidad de apoyar en la solución de las problemáticas que se puedan presentar en éstas, además de utilizar diversas herramientas con el fin de desarrollar controladores aplicados a la industria.

Actualmente ocupa el cargo de Ingeniero de Aplicación MATLAB en MultiON Consulting S.A. de C.V. en donde se dedica a la resolución de problemas de índole computacional que enfrentan organizaciones tanto públicas como privadas.

Temario

1. Introduction

- a. MathWorks® at a Glance
- b. Worldwide Offices
- c. MathWorks® Product Overview
- d. Diverse Users
- e. Computer Setup
- f. Code Acceleration Workflow
- g. Course Learning Outcomes
- h. Course Outline

2. Improving Performance

- a. Course Example:Monte Carlo Simulation
- b. MATLAB® Code Analyzer
- c. (Cumulative) Sums and Products
- d. MATLAB® Profiler
- e. Vectorization
- f. Rewriting the Algorithm
- g. New Variables
- h. Applying Functions to Matrices
- i. Removing for-Loops
- j. Logical Indexing

3. Generating MEX-Files

- a. Benefits of Using MEX-Files
- b. Supported Language Features and Functions
- c. Workflow for Generating MEX-Files from MATLAB® Code
- d. Preparing MATLAB® Code for Code Generation
- e. Creating a MATLAB® Coder™ Project
- f. Defining Input Types
- g. Checking for Issues
- h. Generating Code
- i. Code Generation Report
- j. Command-Line Interface
- k. Verifying Generated Code
- l. Calling Unsupported Functions
- m. Code Preparation Workflows
- n. MEX-Files and Vectorization
- o. Settings for Generating MEX-Files

4. Parallelizing Computations

- a. Hardware Utilization
- b. Existing Parallelism
- c. Distributing Work
- d. Creating a Pool of Resources
- e. Parallel Computing Support in Other Toolboxes and Products
- f. Parallel for-Loops Using parfor
- g. Evaluating Speedup
- h. Parallelization and Vectorization
- i. Aggregating Iterations
- j. Optimal Combination of Vectorization and Parallelization
- k. Parallelization and MEX-Files
- l. Chapter Example: Analyzing El Niño Temperatures
- m. Parallel Processing of Multiple Files
- n. Tabular Text Datastores
- o. Processing Files with Identical Structure
- p. Additional Considerations

5. Parallel for-Loops

- a. Course Example: Birthday Paradox for Multiple Group Sizes
- b. Requirements for parfor-Loops
- c. Independent parfor Iterations
- d. Parallelizing Nested Loops
- e. Loop Variable Considerations
- f. Obtaining Intermediate Results
- g. Variable Classification
- h. Handling Classification Issues

6. Offloading Execution

- a. Offloading Work
- b. Batch Job Creation and Submission Process
- c. Connecting to the Cluster
- d. Executing a Script in Batch
- e. Executing a Function in Batch
- f. Running the Job
- g. Job Monitor
- h. Gathering Results
- i. Cleaning Up
- j. Retrieving Jobs
- k. Obtaining Intermediate Results
- l. Offloading Work Explicitly
- m. Job Creation and Submission Process

7. Working with Clusters

- a. System Components
- b. Third-Party Schedulers
- c. Dynamic Licensing
- d. Identifying Cluster Resources
- e. Using the Cluster Profile Manager
- f. Attaching Files
- g. File Access and Operating System Considerations
- h. Generating MEX-Files on Clusters

8. GPU Computing

- a. What Is a GPU?
- b. Common Applications That Run on a GPU
- c. Multiprocessor Models
- d. GPU Architecture
- e. GPU Card Considerations
- f. Chapter Example: Julia Sets
- g. Computing with the GPU
- h. Creating Data on the GPU
- i. Using MATLAB[®] Functions
- j. Performing Calculations
- k. Retrieving Data to the CPU
- l. Measuring Performance
- m. Using Preexisting CUDA Code
- n. Speedup Comparisons
- o. Using Multiple GPUs in Parallel



MultiON es líder en México y Latinoamérica en la comercialización, el soporte y la capacitación en el uso de software y hardware especializado para la ciencia, la educación, la industria y los servicios. Esta empresa de cómputo científico y técnico fue fundada en 1989 por el ingeniero y maestro en administración Joaquín Antonio Maury González durante sus estudios de doctorado y es 100% mexicana.

Joel Cervantes

Asesor Comercial LATAM

MultiON Consulting, S.A. de C.V.

Cómputo Científico y Técnico: *software y hardware especializado.*

Tel: +52 (55) 5559-4050 Ext. 119 | cursos@multion.com

www.multion.com