

CATALOGUE DE FORMATION 2021-2022



Calcul Québec

À PROPOS DE NOS ATELIERS

- Nos ateliers peuvent être donnés en français et en anglais;
- Nos formateur.trice.s sont des experts et des expertes en calcul informatique de pointe
- Vous pouvez consulter notre calendrier de formations sur notre page Eventbrite;
- Pour plus d'information, merci d'écrire à formation@calculquebec.ca.



FORMATIONS

LES FONDAMENTAUX

Ligne de commande Unix	4
Gestion de révisions avec Git	5
Premiers pas sur les serveurs de calcul	6

ANALYSE ET MANIPULATION DES DONNÉES

Introduction à la programmation avec Python	8
Introduction à la programmation en R	9
Programmation performante en R	10
Nettoyage des données avec OpenRefine	11
Analyse de données et visualisation avec Python	12
Analyse de données massives avec Spark	13

PROGRAMMATION PARALLÈLE

Introduction à la programmation parallèle avec OpenMP	15
Introduction à la programmation parallèle avec MPI	16

PROGRAMMATION GPU

Introduction à la programmation CUDA	18
Programmation GPU avec OpenACC	19
Bibliothèques de programmation GPU	20
Programmation GPU avec Python	21

LES FONDAMENTAUX



LIGNE DE COMMANDE UNIX

Apprenez les bases d'un système de fichiers et d'une interface en ligne de commande Unix.



Résumé

La ligne de commande Unix (Unix Shell) est un outil puissant qui permet d'effectuer des opérations complexes en quelques commandes, de combiner des programmes existants et d'automatiser des tâches répétitives.

Cette ligne de commande est indispensable à toute personne désirant utiliser des ressources de calcul tels que des supercalculateurs.

Prérequis

Comprendre les concepts de fichiers et répertoires.

Plan de cours

1. Introduction à la ligne de commande;
2. Arborescence de fichiers et répertoires;
3. Création, suppression et gestion de fichiers et répertoires;
4. Combinaisons de commandes, redirections et filtres;
5. Boucles;
6. Scripts bash;
7. Recherche de fichiers.

GESTION DE RÉVISIONS AVEC GIT

Apprenez à utiliser un système de gestion de révisions pour suivre les changements que vous apportez à vos fichiers.

**3h30****Débutant**

Résumé

La gestion des révisions permet aux professionnels de garder une trace de leur travail et de collaborer avec d'autres personnes plus facilement. Tous les projets de développement majeurs utilisent un système de gestion de révisions.

Cette formation s'intéresse particulièrement aux bases de la gestion de code source avec Git, le logiciel de suivi des révisions le plus populaire au monde. Vous apprendrez également comment Git peut être utilisé avec d'autres formats de fichiers (livres, articles, jeux de données etc).

Prérequis

Comprendre les concepts de fichiers et répertoires, et avoir une connaissance de base de la ligne de commande Unix. Référez-vous à la formation UNX101 Ligne de commande Unix.

Plan de cours

1. Introduction à la gestion des révisions;
2. Configuration de Git;
3. Créer un dépôt;
4. Suivre les révisions;
5. Explorer l'historique;
6. Ignorer certains fichiers;
7. Dépôts distants;
8. Collaborer;
9. Gestion des conflits;
10. Science ouverte, licences et hébergement.

PREMIERS PAS SUR LES SERVEURS DE CALCUL

Apprenez à lancer vos premiers calculs sur les serveurs de Calcul Canada.

**3h30****Débutant**

Résumé

Utiliser un superordinateur peut être une expérience plutôt déroutante pour un débutant. Plusieurs défis doivent être relevés avant de pouvoir bénéficier pleinement de la puissance d'une telle installation.

Cette formation vous permettra de comprendre le fonctionnement d'un serveur de calcul en plus de vous fournir les outils nécessaires pour lancer correctement et efficacement vos premiers calculs.

Prérequis

Comprendre les concepts de fichiers et répertoires, et avoir une connaissance de base de la ligne de commande Unix. Référez-vous à la formation UNX101 Ligne de commande Unix.

Plan de cours

1. Introduction à Calcul Québec et Calcul Canada;
2. Introduction au calcul informatique de pointe;
3. Transfert de fichiers entre la grappe de calcul et un autre système avec scp/sftp et Globus;
4. Utilisation de modules et logiciels variés;
5. Utilisation de l'ordonnanceur et soumission de tâches;
6. Erreurs courantes, bonnes pratiques et recommandations.

A thick yellow L-shaped graphic element is positioned on the left side of the page. It consists of a vertical bar extending from the top to the bottom, and a horizontal bar extending from the vertical bar towards the right, intersecting it at approximately one-third of the way down.

ANALYSE ET MANIPULATION DES DONNÉES

A thick yellow horizontal bar is positioned below the title, extending from the left edge of the page to the right edge. It is slightly offset to the right, starting from the vertical bar of the L-shaped graphic.

INTRODUCTION À LA PROGRAMMATION AVEC PYTHON

Maîtrisez les concepts de base de la programmation avec le langage Python.

**3h30****Débutant**

Résumé

La meilleure façon d'apprendre à programmer est de réaliser une tâche utile. Cette introduction à la programmation en Python est donc construite autour d'une tâche scientifique courante : l'analyse de données.

Le but n'est pas de vous enseigner Python, mais de vous enseigner les concepts de base de la programmation.

Prérequis

Comprendre les concepts de fichiers et répertoires, et avoir une connaissance de base de la ligne de commande Unix.

Référez-vous à la formation UNX101 Ligne de commande Unix.

Plan de cours

1. Concepts de programmation : variables, types de données, répétitions d'actions en boucles;
2. Analyser des données avec des bibliothèques externes;
3. Listes et autres conteneurs de données;
4. Analyses des données issues de plusieurs fichiers;
5. Programmation conditionnelle et prise de décision;
6. Programmation réutilisable en utilisant des fonctions;
7. Programmation défensive et autres bonnes pratiques de développement.

INTRODUCTION À LA PROGRAMMATION EN R

Apprenez les concepts de base sur lesquels reposent toute la programmation avec R.

**3h30****Débutant**

Résumé

La meilleure façon d'apprendre à programmer est de réaliser une tâche utile. Cette introduction à la programmation avec R est donc construite autour d'une tâche scientifique courante : l'analyse de données.

Le but n'est pas de vous enseigner R, mais de vous enseigner les concepts de base de la programmation.

Prérequis

Comprendre les concepts de fichiers et répertoires et avoir une connaissance de base de la ligne de commande Unix. Référez-vous à la formation UNX101 Ligne de commande Unix.

Plan de cours

1. L'environnement R;
2. Variables;
3. Structures de données;
4. Boucles;
5. Conditions;
6. Fonctions;
7. Usage de R sur les grappes de Calcul Canada.

PROGRAMMATION PERFORMANTE EN R

Vous maîtrisez le langage de programmation R et vous souhaitez aller plus loin? Suivez cette formation pour découvrir des concepts et des structures de programmation plus avancés en R.

**3h30****Débutant**

Résumé

Cette formation vous donne les outils nécessaires pour améliorer la performance et la portabilité de votre code en R.

Nous traitons des aspects plus avancés de la programmation en R : les structures de programmation, la philosophie de programmation (impérative vs. fonctionnelle), la meilleure méthode pour améliorer la performance de son code et comment rendre son code plus souple, portable et propre. Cette formation s'adresse donc aux personnes ayant déjà une expérience de programmation en R.

Prérequis

Avoir de l'expérience de programmation en R ainsi qu'une bonne connaissance de la ligne de commande en Unix.

Référez-vous aux formations UNX101 Ligne de commande Unix et RRR101 Introduction à la programmation en R.

Plan de cours

1. Création des diagrammes;
2. Interaction avec l'environnement Unix;
3. Programmation fonctionnelle;
4. Parallélisation et vectorisation;
5. Programmation mixte avec C;
6. Débogage, profilage et optimisation;
7. Autres sujets.

NETTOYAGE DES DONNÉES AVEC OPENREFINE

Apprenez à utiliser OpenRefine pour nettoyer et formater efficacement vos données et pour automatiquement suivre tous les changements que vous faites.



Sommaire

OpenRefine est un projet ouvert permettant le nettoyage et la bonification des données en recherche. Outil incontournable du chercheur, il permet de conserver le fichier original intact, de garder une trace des manipulations sur les données, d'annuler une modification facilement, de sauvegarder des routines et de les appliquer à d'autres fichiers.

Dans cette formation, vous apprendrez à utiliser le logiciel et à créer des routines réutilisables sur d'autres fichiers. Vous vous familiariserez aussi avec l'exploration des données ainsi que certaines bonnes pratiques en manipulation de données.

Prérequis

Avoir des connaissances de base en statistiques (moyenne, écart-type, distribution normale).

Plan de cours

1. Filtres et "facetting";
2. Transformations communes;
3. Identification et correction de mauvaises entrées et erreurs de frappe;
4. Utilisation du langage GREL;
5. Automatisation des modifications;
6. Réconciliation des données;
7. Bonification de données (API).

**Ce cours peut également être donné en 3h30. Nous consulter pour plus de détails.*

ANALYSE DE DONNÉES ET VISUALISATION AVEC PYTHON

Si vous avez suivi notre formation d'introduction à la programmation en Python et que vous vous sentez prêt à aller plus loin, cette formation est faite pour vous.

**Intermédiaire**

Résumé

Cette formation vous permettra d'effectuer une analyse de données de base, du chargement des données dans Python à une visualisation finale. Pour ce faire, vous découvrirez notamment les bibliothèques Pandas et une bibliothèque de visualisation telle que Matplotlib ou Plotnine.

L'analyse des données se limite à la sélection de certaines entrées, à la transformation temporaire des données, au calcul de statistiques simples et à l'automatisation de tout ce processus d'analyse.

Prérequis

Une connaissance de base en programmation, préférablement en Python ou un langage similaire est fortement suggérée. Référez-vous à la formation PYT101 Introduction à la programmation en Python.

Plan de cours

1. Pandas et les DataFrames (jeux de données) en Python;
2. Indexation, découpage et sélection de jeux de données;
3. Type et format des données;
4. Combiner des jeux de données;
5. Automatisation d'analyses;
6. Visualiser vos données avec Matplotlib ou Plotnine;
7. Accéder à des bases de données SQLite avec Python et Pandas.

**Ce cours peut également être donné en 3h30. Nous consulter pour plus de détails.*

ANALYSE DE DONNÉES MASSIVES AVEC SPARK

Pour analyser efficacement une importante quantité de données, apprenez à utiliser Apache Spark avec Python.



3h



Intermédiaire

Résumé

Apache Spark est l'un des logiciels libres les plus importants pour le traitement et l'analyse des données massives. Durant cet atelier, nous vous enseignerons l'utilisation d'Apache Spark avec Python (PySpark) pour analyser des jeux de données trop volumineux pour être traités par un seul ordinateur.

Avec PySpark, vous apprendrez à importer vos données, à utiliser les fonctions pour transformer, réduire et compiler vos données, et à produire des algorithmes parallèles pouvant s'exécuter sur des grappes de calcul telles que celles offertes par Calcul Québec et Calcul Canada.

Prérequis

Avoir une bonne connaissance de la ligne de commande Unix (référez-vous à la formation UNX101 Ligne de commande Unix) et savoir écrire des fonctions en Python.

Plan de cours

1. Introduction aux données massives et à Map-Reduce;
2. Présentation d'Apache Spark;
3. Importation de données avec PySpark;
4. Organisation des données en clé-valeur;
5. Travailler avec des données structurées (PySpark SQL);
6. Développement d'algorithmes parallèles.

PROGRAMMATION PARALLÈLE



INTRODUCTION À LA PROGRAMMATION PARALLÈLE AVEC OPENMP

Afin d'améliorer les performances de votre code, vous apprendrez à maîtriser les bases de l'interface de programmation OpenMP grâce à des exemples d'application concrets.



Intermédiaire à avancé

Sommaire

Dans cet atelier, vous apprendrez à utiliser OpenMP, une interface de programmation standard pour le calcul parallèle sur architecture à mémoire partagée. Vous explorerez également les techniques de synchronisation des fils d'exécution (threads) pour éviter les situations de compétition (race conditions) et les techniques d'amélioration des performances.

Prérequis

Maîtriser les bases de la programmation en C, C++ ou Fortran.

Plan de cours

1. Introduction;
2. Modèle de programmation à mémoire partagée;
3. Compilation et exécution de programmes OpenMP;
4. Éléments de syntaxe : directives, clauses, routines de la bibliothèque, variables d'environnement;
5. Parallélisation automatique;
6. Parallélisation manuelle;
7. Situations de compétition.

**Ce cours peut également être donné en 3h30. Nous consulter pour plus de détails.*

INTRODUCTION À LA PROGRAMMATION PARALLÈLE AVEC MPI

Cette formation a pour objectif de vous faire maîtriser les fondamentaux du calcul parallèle distribué sur plusieurs noeuds grâce à MPI.

**7:00*****Intermédiaire à avancé**

Sommaire

Exploiter la pleine puissance d'une grappe de calcul nécessite d'utiliser plusieurs noeuds de manière coordonnée. Pour ce faire, en plus de programmer l'algorithme de calcul, il est nécessaire de programmer la communication entre les processus s'exécutant sur chacun des noeuds. MPI (Message Passing Interface) est la façon la plus courante de gérer cette communication.

À l'issue de cette formation, vous maîtriserez les notions de base de la norme MPI et vous serez ainsi capable de coordonner les processus sur les différents noeuds.

Prérequis

Maîtriser les bases de la programmation en C, C++ ou Fortran.

Plan de cours

1. Concepts génériques;
2. Un premier programme, les rangs et le nombre de processus dans un communicateur;
3. Communications point-à-point;
4. Synchronisation entre processus;
5. Communications collectives : réduction, collection, diffusion et distribution;
6. Types de données MPI;
7. Communications non bloquantes.

**Ce cours peut également être donné en 3h30. Nous consulter pour plus de détails.*

PROGRAMMATION GPU



INTRODUCTION À LA PROGRAMMATION CUDA

Vous apprendrez à écrire des algorithmes simple sur des GPU en utilisant CUDA avec le langage de programmation C.



7h



Intermédiaire à avancé

Résumé

CUDA (Compute Unified Device Architecture) est le langage de programmation le plus utilisé pour exploiter la puissance de calcul des processeurs graphiques (GPU) dans le domaine du calcul de haute performance. Ce langage est supporté par la plupart des accélérateurs graphiques NVIDIA.

Dans ce cours, vous apprendrez comment à utiliser CUDA avec le langage de programmation C dans un environnement Unix pour écrire des algorithmes simples sur des GPU.

Prérequis

- Maîtriser la ligne de commande Unix. Référez-vous à la formation UNX101 Ligne de commande Unix; et
- Avoir une connaissance de base du langage C (incluant les pointeurs et l'allocation de mémoire).

Plan de cours

1. Architecture d'un GPU;
2. Flot d'exécution d'un programme CUDA;
3. Les fondamentaux pour utiliser un GPU efficacement;
4. Trouver de l'information sur le web et à l'intérieur du kit de développement de CUDA;
5. Écrire et compiler un code minimal et les exemples CUDA;
6. Gérer la mémoire du GPU et comprendre les types de mémoire disponibles;
7. Utiliser des fils (thread) et blocs pour écrire des algorithmes parallèles;
8. Vérifier les erreurs;
9. Autres façons d'utiliser un GPU.

Note : La connaissance du calcul parallèle et du calcul sur GPU n'est pas un prérequis pour ce cours.

PROGRAMMATION GPU AVEC OPENACC

Grâce à ce cours, vous serez capable de porter un code existant vers les accélérateurs NVIDIA grâce à OpenACC.

**3h30****Avancé**

Résumé

OpenACC est un ensemble de directives, semblable à OpenMP, qui permet de paralléliser un code existant et de le porter pour une exécution sur des accélérateurs (GPU ou autres). Comparativement à CUDA, ce langage permet d'écrire facilement et rapidement un code portable qui s'exécutera aussi bien sur un accélérateur que sur un processeur conventionnel.

Dans ce cours, vous apprendrez à utiliser OpenACC pour porter un code existant vers les accélérateurs NVIDIA. Ce cours utilise le langage C ou Fortran dans un environnement Linux.

Prérequis

Être capable de vous connecter à une grappe de calcul Linux et d'interagir avec celle-ci (éditer des fichiers, naviguer dans les répertoires) en ligne de commande et avoir une connaissance de base du langage C ou de Fortran.

Plan de cours

1. Introduction aux architectures d'accélérateurs;
2. Profiler le code existant et extraire des informations du compilateur;
3. Exprimer le parallélisme du code avec des directives OpenACC;
4. Exprimer les transferts de données;
5. Optimiser les boucles.

Note : La connaissance du calcul parallèle et du calcul sur GPU n'est pas un prérequis pour ce cours.

BIBLIOTHÈQUES DE PROGRAMMATION GPU

Vous avez besoin de transférer des tâches de calcul intensives des CPU vers les GPU? Gagnez du temps en apprenant à utiliser des bibliothèques GPU pour vos codes.



3h30



Avancé

Résumé

Réécrire un code scientifique en utilisant un langage spécialisé pour les GPU tel que CUDA ou OpenCL peut s'avérer compliqué et long. Toutefois, les bibliothèques supportées par les GPU permettent aux chercheurs de transférer facilement des tâches de calcul intensives (par exemple, la multiplication matricielle, l'inversion, la diagonalisation etc) des CPU vers les GPU.

À l'issue de cette formation, vous maîtriserez les concepts de base des GPU et vous serez capable d'utiliser des bibliothèques GPU pour vos codes.

Prérequisites

Être capable de vous connecter à une grappe de calcul Unix et d'interagir avec celle-ci (éditer des fichiers, naviguer dans les répertoires) en ligne de commande et avoir une connaissance de base du langage C, Fortran ou Python.

Plan de cours

1. Introduction;
2. Les bibliothèques GPU comme solution de rechange à CUDA;
3. Convertir du code CPU en code GPU;
4. Bibliothèque NVIDIA cuBLAS;
5. Bibliothèque NVIDIA cuFFT;
6. Bibliothèque CULA;
7. Bibliothèque MAGMA;
8. NVIDIA CUDA Thrust;
9. Session pratique.

Note : La connaissance du calcul parallèle et du calcul sur GPU n'est pas un prérequis pour ce cours.

PROGRAMMATION GPU AVEC PYTHON

Venez apprendre à développer des programmes GPU simples en Python.



Résumé

Cet atelier se concentre sur le calcul accéléré par GPU avec Python. Bien qu'il s'agisse d'un langage très populaire, Python n'est pas optimisé pour le calcul haute performance (CHP) et ne contient aucune fonctionnalité permettant de l'utiliser sur les GPU. C'est pourquoi, il est nécessaire d'utiliser des bibliothèques basées sur Python, telles que CUDA Python (PyCUDA) et Numba. En effet, ces outils permettent de combiner la rapidité de développement qu'offre Python et la rapidité d'un langage compilé ciblant à la fois les CPU et les GPU.

À la fin de la formation, les participants et participantes maîtriseront les concepts clés du GPU et seront capables de développer des programmes GPU simples en Python.

Prérequis

Une bonne connaissance du langage de programmation Python est nécessaire. Référez-vous à la formation PYT101 Introduction à la programmation avec Python.

Plan de cours

1. Pourquoi écrire sur des GPU?
2. Comprendre l'architecture d'un GPU;
3. Qu'est ce que CUDA?
4. Python + CUDA: votre premier code PyCUDA;
5. GPUArray: une bibliothèque PyCUDA très pratique;
6. Numba: compilation à la volée;
7. Numba: votre premier code;
8. Fonctions Numba et modes de compilation;
9. Numba + CUDA;
10. Session pratique: apprendre à programmer le GPU avec PyCUDA et Numba.

Note : La connaissance du calcul parallèle et du calcul sur GPU n'est pas un prérequis pour ce cours.



Calcul Québec

Adresse postale :

Calcul Québec
Université de Montréal
Complexe des sciences
1375 Avenue Thérèse-Lavoie-Roux
Montréal (Québec) H2V 0B3

Autres sites :

Université McGill | Université Laval |
Université de Sherbrooke

Pour plus d'information :

www.calculquebec.ca

CONTACT :

formation@calculquebec.ca