Université de Bretagne Occidentale

Faculté des Sciences & Techniques

MASTER INFORMATIQUE

PARCOURS LOGICIELS POUR SYSTÈMES EMBARQUÉS

Semestre 8

S8 4 UEs à choisir parmi

S8 INF Systèmes à objets répartis

Présentation

Etude des applications réparties hétérogènes.

3 crédits ECTS

Volume horaire

Cours Magistral : 12h
Travaux Pratiques : 32h

Objectifs

L'objectif de cette UE est de connaître et savoir utiliser différentes technologies permettant de construire des applications réparties hétérogènes.

Pré-requis nécessaires

système d'exploitation et réseaux, langage et algorithmique, architecture des systèmes distribués, applications réparties

Compétences visées

- Connaître les mécanismes mis en oeuvre dans les systèmes et les services à objets répartis.
- Connaître les modèles d'interaction dans les applications réparties (client/serveur, agents mobiles)
- Savoir programmer en objet/service répartis
- Comprendre les enjeux de l'intégration de service, les problèmes d'interopérabilité (systèmes, langages, protocoles réseaux)
- Connaître les principaux mécanismes et patrons de conception exploités dans les plates-formes offrant un support pour l'intégration de systèmes et l'interopérabilité : patrons service de nommage (naming), service cycle de vie (factory), proxy, service d'événements, service transactionnel ...

Descriptif

1) Introduction aux systèmes répartis

Problèmatiques et notions de transparence. Modèles d'interaction. Mécanismes et patrons usuels. Exemples de systèmes répartis

2) Les approches objets répartis

- Le standard CORBA. L'architecture OMA, notion de bus à objets. Mapping Java et C++. Services standards CORBA (naming et event services)
- Java RMI. Le modèle agent mobile : migration, services web. Architecture et fonctionnement de java RMI
- La plate-forme C#/.NET et les web services . L'architecture .Net, XML et ses outils, C#

3) Les approches services

- L'architecture JEE
- Accès aux bases de données distantes, JDBC
- Gestion des requêtes HTTP, objets Servlet et HttpSession, validation des requêtes
- Conception d'une application MVC basée sur les Servlets
- L'architecture NodeJS
- Programmation asynchrone, gestion des événements, promesses (promises)
- Échanges de données en JSON (Javascript Object Notation)
- Conception d'un serveur Web, le framework express, l'API REST, les routes
- Conception d'une application Web répartie, le framework React

Répartition approximative : 2h introduction, 10h CORBA, 5h RMI, 5h .NET, 12h JEE, 12h NodeJS

Bibliographie

- Learning Node.js, A Hands-On Guide to Building Web Applications in JavaScript. Marc Wandschneider. Addison Wesley editor. 2017.
- Java network programming and distributed computing. David Reilly, Addison Wesley editor. 2003.
- .NET Application Development with C#, ASP.NET, ADO.NET and Web Services. Hanspeter Mossenbock, Dietrich Birngruber, Albrecht Woss, Wolfgang Beer. Addison Wesley editor. 2004.

Modalités de contrôle des connaissances

Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120	1	

Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120		

Langue d'enseignement

Français

S8 INF Administration des systèmes d'exploitation

Présentation

Cette UE aborde le rôle de l'administrateur et ses principales tâches, la méthodologie d'administration système, la procédure d'installation d'un système Linux, la gestion des identités et des annuaires, la gestion de l'espace disque (partition, systèmes de fichiers, organisation, sauvegarde), les services réseaux (NIS, DNS, NFS), les pare-feux et le filtrage réseau, le démarrage des services, l'automatisation des tâches d'administration, les outils de supervision.

3 crédits ECTS

Volume horaire

Travaux Dirigés : 6h
Cours Magistral : 12h
Travaux Pratiques : 26h

Objectifs

Le but de cette UE est de former à l'administration système et réseau sous Linux.

Pré-requis nécessaires

- Connaissance des fonctions d'un système d'exploitation et des principes de mise en œuvre.
- Connaissance de base en programmation shell.
- Connaissance de base sur les réseaux informatique

Compétences visées

- Connaître le rôle et les tâches de l'administrateur système
- Maîtriser la procédure d'installation d'un système Linux
- Maîtriser les bases de l'administration d'un système
- Savoir configurer les services réseaux, les pare-feux et le filtrage réseau
- Savoir gérer les identités et les annuaires

Bibliographie

Linux administration, Jean-François Bouchaudy, Gilles Goubet, Eyrolles, 2007

Modalités de contrôle des connaissances

Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	60	1/3	
	CC	Travaux Pratiques		2/3	

Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	60	100%	

Langue d'enseignement

Français

S8 INF Systèmes d'information

Présentation

Au travers de cette UE, les étudiants considèrent les architectures et technologies utiles à la mise en place d'un système d'information.

3 crédits ECTS

Volume horaire

Travaux Pratiques : 32h
Cours Magistral : 6h
Travaux Dirigés : 6h

Objectifs

le premier objectif est de présenter les architectures à base de services web pour les systèmes d'information.

L'objectif de cette UE est aussi d'illustrer la diversité des systèmes de gestion de bases de données en mettant en œuvre plusieurs types de SGBD qualifiés de NoSQL.

Par ailleurs, les environnements de développement utilisant des outils de travail collaboratif, leur usage est ici illustré via des projets.

Pré-requis nécessaires

Cette UE s'appuie sur les connaissances des langages Java et Javascript.

Les bases du développement web sont supposées connues

Compétences visées

Etre autonome dans la mise en œuvre d'un site web s'appuyant sur une structure à base de services

- Mise en œuvre du développement web à base de services REST en utilisant Java avec le Framework Spring Boot
- Mise en oeuvre du développement web à base de services REST en utilisant NodeJS

Savoir choisir un système de base de données adapté aux besoins entre

- Base de données relationnelle
- Base de données orientée document (MongoDB)
- Base de données graphe (Neo4j)

Être autonome dans l'usage d'un logiciel de versionning et travail collaboratif, utilisation de gitlab.

Modalités de contrôle des connaissances

Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120	2/3	
	CC	Travaux Pratiques		1/3	

Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120		

Langue d'enseignement

Français

S8 INF Programmation Parallèle Haute Performance

Présentation

Cette UE porte sur la programmation parallèle multi-threads et multi-processeurs. Elle aborde les principaux modèles d'exploitation du parallélisme, la mesure de la performance, et se focalise sur la programmation openMP et PVM.

3 crédits ECTS

Volume horaire

Travaux Dirigés : 16h
Travaux Pratiques : 14h
Cours Magistral : 14h

Objectifs

Cette UE est composée de quatre parties :

> Introduction au parallélisme : modèles de parallélisme (données, contrôle, flot), modèles de programmation parallèle, mesures et limites de performance.
> Programmation parallèle multi-threads en mémoire partagée avec openMP : parallélisation de boucles, ordonnancement statique et dynamique, création de tâches dynamiques.
> Programmation parallèle multi-processus à grande échelle en mémoire distribuée avec PVM (Parallel Virtual Machine) : passage de messages, création/destruction de processus, paradigme maître/esclave.
> Algorithmes et techniques pour le parallélisme massif : ordonnancement, analyse de dépendances, réécriture de nids de boucle, calcul systolique.

Pré-requis nécessaires

> Systèmes d'exploitation
> Architecture des ordinateurs

Compétences visées

> Capacité d'identification des différents types de parallélisme exploitables dans une application
> Parallélisation muti-threads de programmes séquentiels en openMP
> Savoir concevoir des programmes multi-processus pour le calcul parallèle
> Amélioration des performances parallèles par réécriture de code

Bibliographie

> Algorithmes et architectures systoliques - P.Quinton et Y. Robert, Masson, 1989
> Algorithmique parallèle – A. Legrand et Y. Robert, Dunod, 2003
> PVM: Parallel Virtual Machine A Users' Guide and Tutorial for Networked Parallel Computing – A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, V. Sunderam, MIT Press, 1994

Modalités de contrôle des connaissances

Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120	2/3	
	CC	Travaux Pratiques		1/3	

Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120		

Langue d'enseignement

Français

S8 INF Architectures et accélérateurs parallèles

Présentation

Cette UE porte sur la connaissance et la programmation des principaux modèles d'architectures et d'accélérateurs parallèles. Elle se focalise plus particulièrement sur la programmation des accélérateurs FPGA et GPU.

3 crédits ECTS

Volume horaire

Travaux Pratiques : 30h
Cours Magistral : 14h

Objectifs

Cette UE présente globalement les différents modèles d'architectures parallèle et leur mise en œuvre matérielle :

- Architectures super-scalaires, VLIW, super-pipeline, vectorielle pour des cœurs de processeurs
- Processeurs multicores et manycores et leur hiérarchie mémoire
- Accélérateurs graphiques GPU
- Architectures reconfigurables FPGA
- Supercalculateurs

D'un point de vue conceptuel et pratique, elle se concentre sur la programmation des deux familles d'accélérateurs parallèles les plus répandus pour le calcul intensif (simulation, traitement d'image, IA, data mining), que sont les GPU et FPGA :

- Dans la partie GPU, on apprend à programmer des kernels de calcul intensif pour GPU. Les langages OpenCL et Compute Shader sont étudiés à l'aide d'environnements de programmation comme MatrixStudio ou ShaderToy. Divers sujets sont alors abordés comme le traitement d'image, la simulation de drones, la simulation de systèmes biologiques et plus généralement le calcul scientifique intensif.
- Dans la partie FPGA, on introduit la synthèse de haut-niveau (HLS), permettant la conception de traitements parallélisés à partir d'une description C ou C++. On apprend comment caractériser et comparer les performances, en termes de fréquence d'horloge, de cycles de fonctionnement et de ressources utilisées. L'environnement utilisé est Vitis, pour la programmation des FPGAs du constructeur Xilinx/AMD.

Pré-requis nécessaires

- Systèmes d'exploitation
- Architecture des ordinateurs

Compétences visées

- Savoir concevoir et programmer des cœurs de traitement pour FPGA par synthèse de haut niveau
- avoir concevoir et programmer des kernels de calcul intensif pour GPU avec OpenCL

Bibliographie

- Computer Organization and Design - David A.Patterson and John L.Hennessy, Morgan Kaufmann Publishers, 1997
- Reconfigurable Computing: The Theory and Practice of Fpga-based Computation - S. Hauck et A. DeHon, Morgan Kaufmann, 2007
- OpenCL Programming by Example. Ravishekhar Banger, Koushik Bhattacharyya. Packt Publishing. ISBN-13: 978-1849692342. 2013.
- OpenCL Programming Guide. Aaftab Munshi. Addison-Wesley. ISBN-13: 978-0-321-74964-2. 2013.
- Using OpenCL – Programming Massively Parallel Computers. Janusz Kowalik et Tadeusz Puzniakowski. IOS Press. ISBN 978-1-61499-029-1. 2012.

Modalités de contrôle des connaissances

Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120	2/3	
	CC	Travaux Pratiques		1/3	

Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120		

Langue d'enseignement

Français

S8 INF Méthodes computationnelles

Présentation

3 crédits ECTS

Volume horaire

Travaux Pratiques : 26h
Travaux Dirigés : 2h
Cours Magistral : 16h

L'objectif de cette UE est de connaître les principales méthodes du soft computing (méthodes computationnelles) de façon à en cerner les intérêts et à en apprécier les limites pratiques. Une définition du soft-computing: "Based on fuzzy logic, Evolutionary Computing, Neural networks, Genetic Algorithms, or their combinations to yield synergetic effects, Soft Computing is aimed to realize intelligent systems for different applications"

Contenu susceptible d'évoluer :

- Logique floue : application au contrôle
- Systèmes Immunitaires Artificiels pour l'optimisation et la détection d'anomalies,
- Méta-heuristiques pour l'optimisation (recuit simulé, Algorithmes Génétiques, etc.),
- Programmation Multi-Objectifs (branch and bound)

Objectifs

L'objectif est de connaître les principales méthodes du soft computing (méthodes computationnelles) de façon à en cerner les intérêts et à en apprécier les limites pratiques.

Pré-requis nécessaires

programmation en C

Compétences visées

Connaissances des principales méthodes théoriques de soft computing. Maîtrise de ces méthodes à travers de nombreux exercices pratiques.

Bibliographie

- Logique floue. Exercices corrigés et exemples d'applications. Bernadette Bouchon-Meunier, Laurent Foulloy, Mohammed Ramdani. Cépaduès, 1998.
- Métaheuristiques pour l'optimisation difficile. Patrick Siarry. Eyrolles, 2005.
- Artificial Immune Systems: A new computational Intelligence Approach. Springer-Verlag, 2002.
- Algorithmes génétiques et réseaux de neurones. Jean-Michel Renders. Hermès, 1995.

Et la vision du soft computing de l'Univ. de Berkeley: http://www.cs.berkeley.edu/~zadeh

Modalités de contrôle des connaissances

Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120	2/3	
	CC	Travaux Pratiques		1/3	

Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120		

Langue d'enseignement

Français