

Programmation Avancée - Introduction

T.DUFAUD

UVSQ - IUT Vélizy - Informatique



INF3 S5 - 8 /09 / 2025

Présentation
du module

Applications et
systèmes
répartis -
généralités

Programmation
répartie

1 Présentation du module

2 Applications et systèmes répartis - généralités

3 Programmation répartie

Présentation du module

Contexte

- PPN BUT Informatique - Parcours A
- **R5.A.05 *Programmation Avancée* avec un focus sur la programmation répartie**
- **Prérequis** : M3101 (Système d'exploitation) M3102 (Service Réseau) M3105 (COOAv)

Objectif

- utiliser un environnement de développement indispensable à une fonction de développement de niveau 6...
- ...S'avoir programmer une application répartie

Contenu

- Modèles et problématiques des applications réparties
- Programmation concurrente (Multiprocessus, multithread, par évènements, etc.)
- Programmation Client/Serveur

Séances et modalités

- Chaque semaine 3h TD/TP FI (bloc de 4h en FA)
- Contrôle continu
 - 1 rapports à faire sur l'ensemble des séances en TP avec un oral
 - 4 contrôles courts (20 minutes)
 - préparation à la SAE

Enseignants

- Thomas Dufaud : MCF UVSQ IUT,
<https://cv.hal.science/thomas-dufaud> LI-PaRAD et Maison de la Simulation (resp. module, CM, TD/TP, SAE)
- contact : thomas.dufaud@uvsq.fr

Du point de vue du logiciel

En terme d'architecture logicielle, comment sont généralement conçues les applications aujourd'hui ?

- les applications web
- les systèmes informatiques pour les systèmes d'informations
- flux de travail pour la simulation numérique

Du point de vue des architectures matérielles

Sur quelles architectures matérielles de votre connaissance sont portées ces applications ?

- processeur multicore/manycore
- smart phone
- cloud
- grille
- système HPC
- carte graphique
- FPGA

Definition 1 (Application Répartie).

Une application répartie peut permettre l'interaction de plusieurs applications séparées. Elle peut également être déployée par nécessité sur d'importantes ressources de traitement.

Definition 2 (Système Répartie).

"Un système réparti est un ensemble de machines autonomes connectées par un réseau et équipées d'un logiciel dédié à la coordination des activités du système ainsi qu'au partage des ses ressources." [Coulouris et al, 1994]

"Un système réparti est un système qui s'exécute sur un ensemble de machines sans mémoire partagée, mais que pourtant l'utilisateur voit comme une seule et unique machine." [Tanenbaum,1994]

Principe de conception et apport

Sources (Singhoff) (Mauran, 2005)

- **Transparence à la localisation** : ignorer la localisation réelle, utiliser une ressource qu'elle soit locale ou distante
- **Transparence d'accès** : utilisation d'interface pour un accès à une ressource de manière identique (exemples: open, lp...)
- **Transparence à l'hétérogénéité** : Interopérabilité - ne pas tenir compte des différences matérielles ou logicielles
- **Transparence aux pannes** (réseaux, machines, logiciels): Les pannes et réincarnations sont cachées à l'utilisateur
- **Transparence à l'extension des ressources** : le système peut être étendu ou réduit, seules les performances peuvent être impactées.

Web

- Transparence à la localisation :
- Transparence d'accès :
- Désignation :
- Interopérabilité :

Web

- Transparence à la localisation : lien hypertexte
- Transparence d'accès : URL
- Désignation : URL, DNS
- Interopérabilité : pages HTML, web service

Difficultés

source (Krakowiak,2005)

- Asynchronisme du système de communication \Rightarrow détection des défaillances difficile
- Dynamisme (la composition du système change en permanence) \Rightarrow définir un état global est difficile
- Grande taille \Rightarrow scalabilité des applications peut être difficile à réaliser

Programmation d'une application distribuée

- **Traitement** : des processus, support d'exécution d'application qui communiquent entre eux
- **Données** : mémoire partagée ou distribuée
- **Interconnection** : mécanisme de communication
- **Méthode de résolution** : algorithme d'élection, comptage, etc. (graphe)

Présentation
du module

Applications et
systèmes
répartis -
généralités

Programmation
répartie

Programmation répartie

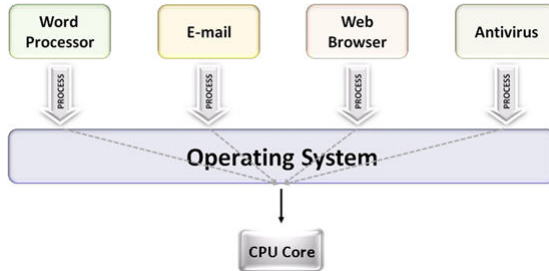


Figure: Mono core : plusieurs tâches sur une unité centrale (Source : National Instrument, 2008)

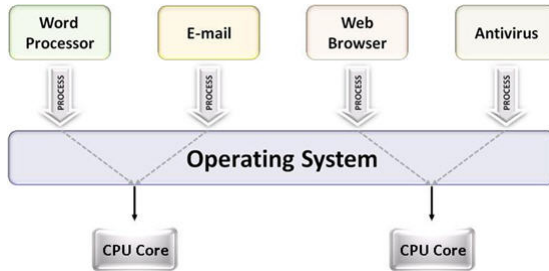


Figure: Multi core : Ici deux tâches peuvent être exécutées simultanément (Source : National Instrument, 2008)

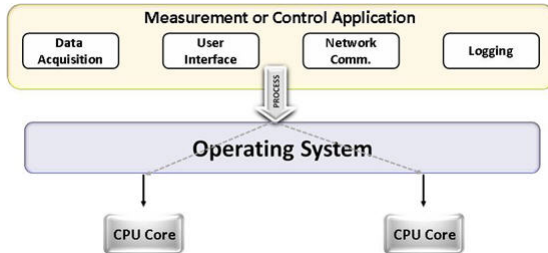


Figure: Multi thread : multi tâche au sein des applications grâce au multi thread
(Source : National Instrument, 2008)

Application séquentielle

L'application est définie par une séquence d'action suivant les structures algorithmiques que sont :

- la séquence
- la sélection (if then else)
- les boucles (for, while)

Contexte mono processeur

Application parallèle

- L'application possède un **potentiel de parallélisme**. Des opérations peuvent s'effectuer indépendamment des autres.
- **Exemple** : la somme de deux vecteurs $z = x + y$ s'écrit comme la somme élément par élément $z_i = x_i + y_i$
Chaque somme i peut être réalisé en parallèle.
- les tâches concourent à la réalisation d'une même application.
Necessite la gestion :
 - des synchronisations
 - des echanges d'informations entre tâches
 - du traitement sélectif
- Contexte multi processeur

Type de mémoire

- Architecture : mémoire distribuée Vs. mémoire partagée
- Algorithme : passage de message Vs. sémaphores, moniteurs, ...

Type de processeur

- Tâche Vs. Données
- Granularité - taille des tâches

Tâche et mémoire partagée

- processus léger : support d'exécution d'une tâche
- collaboration par communication via la mémoire
- mise en oeuvre par les Thread en Java

Remarques

- nous verrons plus tard une architecture distribuée
- la communication entre processus via un réseau

Remarques II

- nous finirons par une ouverture vers le calcul parallèle

- (Coulouris et. al., 1994) G. Coulouris, J. Dollimore, and T. Kindberg. Distributed Systems - Concepts and Design, 2nd Ed. Addison-Wesley Publishers Ltd., 1994.
- (Tanenbaum, 1994) A. Tanenbaum. Systèmes d'exploitation : systèmes centralisés et systèmes distribués. Interéditions, Paris, 1994.
- (F. Singhoff) F. Singhoff, UE systèmes répartis, Université de Brest, http://beru.univ-brest.fr/~singhoff/ENS/UE_systemes_repartis/CM/sd.pdf
- (F. Guinand) F. Guinand, H. Boukachour, Programmation répartie - une introduction, université Le Havre <http://litis.univ-lehavre.fr/~guinand/Enseignement/ProgRep/docs/cours-intro.pdf>
- (Mauran, 2005) Introduction aux systèmes répartis 3ième Année Informatique et Multimédia, 5 octobre 2005, <http://mauran.perso.enseeiht.fr/pages/cours/3AI/introrep.pdf>
- (Krakowiak, 2005) S. Krakowiak Introduction aux systèmes et applications répartis, 2005-2006, Université Joseph Fourier, <http://lig-membres.imag.fr/krakowia/Files/Enseignement/M2P-GI/Flips/1-Intro-objets-lpp.pdf>
- (National Instrument, 2008) Différences entre le multithread et le multitâche pour les programmeurs, National Instrument, 2008, <http://www.ni.com/white-paper/6424/fr/>