

CENTRALE  
L Y O N

## BE 1&2 de GL

**Techniques de simulation à base de multitâche**

- Exemple de simulation de multiprocesseur
- TD de simulation de fonctionnement d'e-Livraison :  
Gestion informatisée de livraisons en ville

BTD/GL/TD2 1

CENTRALE  
L Y O N

## BE 1&2 : Simulateur d'un multiprocesseur

- Principes de simulation
- Expression du parallélisme en langage ADA : Tâches
- Etudier des architectures à base de tâches
- Etudier comment gérer le temps virtuel (de simulation)

NPT – nombre de processeurs de traitement

FTI

FTT

NPES – nombre de processeurs d'E/S

FTD

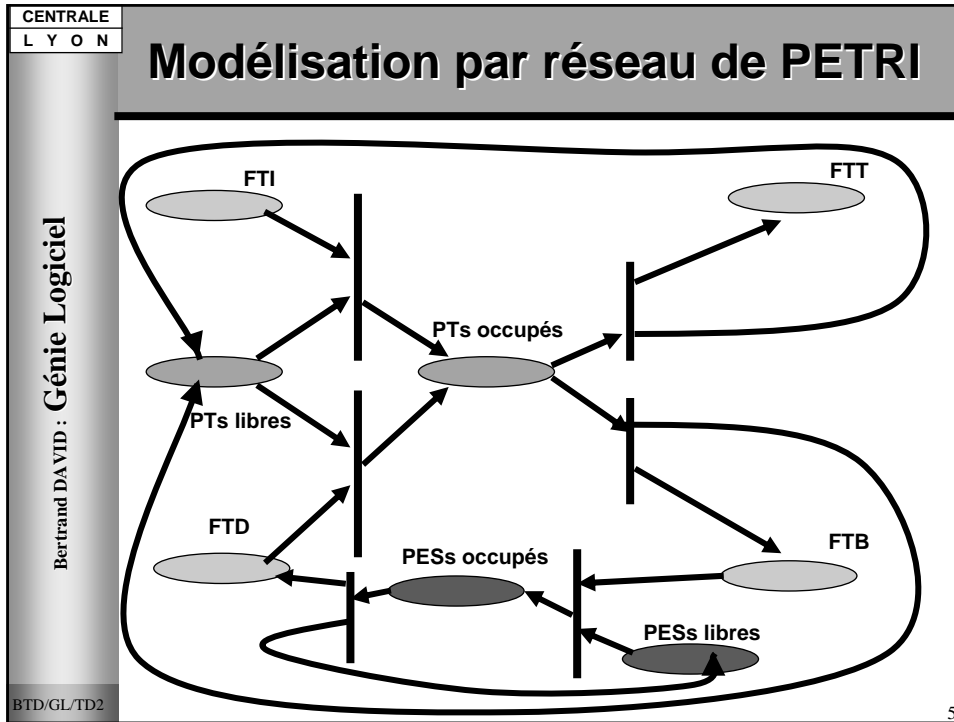
FTB

NT – nombre de travaux

BTD/GL/TD2 2

CENTRALE L Y O N	<h2>Démarche</h2>
Bertrand DAVID : Génie Logiciel	<ul style="list-style-type: none"><li>● Présentation du problème : principes retenus pour la simulation du multiprocesseur</li><li>● Mettre en place un cas (choix du nombre de travaux NT, de processeurs de traitement NPT et de processeurs d'E/S NPES)</li><li>● Faire à la main la simulation et bâtir des graphiques de bilan manuellement :<ul style="list-style-type: none"><li>– Processeurs / temps: occupation des processeurs</li><li>– Travaux / temps: traitement des travaux</li></ul></li><li>● Architectures du simulateur : représentation graphique</li><li>● Architectures du simulateur : description textuelle (en ADA)</li><li>● Prise en compte de la gestion du temps</li></ul>
BTD/GL/TD2	3

CENTRALE L Y O N	<h2>Architecture du multiprocesseur</h2>
Bertrand DAVID : Génie Logiciel	<ul style="list-style-type: none"><li>● Structure:<ul style="list-style-type: none"><li>– Mémoire commune partagée</li><li>– NPT processeurs identiques partageant la mémoire</li><li>– NPES processeurs spécialisés en Entrées-Sorties</li><li>– 1 file de travaux initialisés FTI</li><li>– 1 file de travaux terminés FTT</li></ul></li><li>● Fonctionnement :<ul style="list-style-type: none"><li>– Un processeur traite un travail tant qu'il peut: PAS DE PREEMPTION</li><li>– Les entrées-sorties sont à la charge du processeur d'E/S.</li><li>– La suite du travail peut être effectuée par un autre processeur.</li><li>– Le principe de fonctionnement peut être décrit par le réseau de Petri ci-joint.</li></ul></li></ul>
BTD/GL/TD2	4



CENTRALE  
L Y O N

## Multiprocesseur

- Ressources critiques : files d'attente des travaux
  - FTI: file de travaux initialisés
  - FTB: file des travaux bloqués (sur E/S)
  - FTD: file des travaux débloqués
  - FTT: file des travaux terminés
- Simulation d'un multiprocesseur sur un mono ou multi-processeur :
 

raisonner de façon abstraite en considérant d'avoir le nombre suffisant de processeurs

6

CENTRALE  
L Y O N

## Modélisation du travail

Bertrand DAVID : Génie Logiciel

- Une suite alternée de traitements et d'E/S
- Un vecteur donne successivement les temps des séquences
- Travail : Item courant; NB d'Items
  - EXEC 10
  - E/S 5
  - EXEC 12
  - E/S 2
  - EXEC 20
  - E/S 4
  - EXEC 4
- Bilans à fournir :
  - Occupation des processeurs/ temps
  - Traitement des travaux par le processeurs/ temps

BTD/GL/TD2 7

CENTRALE  
L Y O N

## Approche de modélisation par tâches ADA

Bertrand DAVID : Génie Logiciel

- Rappels du cours :
  - tâches
  - Synchronisation : rendez-vous
  - protection de ressources sensibles : tampon actif
  - select et attente limitée

BTD/GL/TD2 8

CENTRALE  
L Y O N

## Propositions de l'architecture du simulateur à base de tâches

- Processeur modélisé sous forme de tâche ADA ?
- Travail modélisé sous forme de tâche ADA ?
- File modélisée sous forme de tâche ADA ?
- Synchronisations:
  - Qui sollicite le rendez-vous ?
  - Qui propose et accepte le rendez-vous ?
  - Quelles ressources critiques à protéger ?
  - Comment protéger ?
  - Exclusivité d'accès

Bertrand DAVID : Génie Logiciel

BTD/GL/TD2

9

CENTRALE  
L Y O N

## Protection par le Maître

- **Métaphore du banquet :**  
  
**Comment protéger la nourriture et les boissons contre les bousculades**

Bertrand DAVID : Génie Logiciel

BTD/GL/TD2

10

CENTRALE  
L Y O N

## Explications du modèle

- Maître accède aux files
- Esclaves sollicitent le rendez-vous avec le maître pour
  - PRENDRE un travail à faire
  - RENDRE un travail fait
- Esclaves : processeurs de traitement et processeurs d'E/S

Bertrand DAVID : Génie Logiciel

BTD/GL/TD2

11

CENTRALE  
L Y O N

## Architecture n° 1 : approche centralisée maître-esclaves

Bertrand DAVID : Génie Logiciel

BTD/GL/TD2

12

CENTRALE  
L Y O N

## Approche distribuée

- Comment éviter la fatigue du Maître ?
- Comment se protéger soi-même ?

Bertrand DAVID : Génie Logiciel

BTD/GL/TD2

13

CENTRALE  
L Y O N

## Explications du modèle

- Chacun est maître à son tour
- Pour accéder aux files il faut être maître (mais un seul maître à la fois) :
- Tâches : processeurs et files
- Qui propose et accepte (les files)
- Qui sollicite (les processeurs)

Bertrand DAVID : Génie Logiciel

BTD/GL/TD2

14

CENTRALE  
L Y O N

## Architecture n° 2 : approche décentralisée

15

CENTRALE  
L Y O N

## Gestion du temps

**Trouver le modèle du temps le plus juste : attention on a**

- le **temps réel** d'exécution sur le simulateur qui nous n'intéresse pas particulièrement,
- le **temps virtuel de chaque processeur** qui nous intéresse

Comment faire:

- approche non tâche: une boucle sur les processeurs faisant avancer le temps virtuel d'une unité de temps à chaque tour.
- approche tâche :  
Rendez-vous avec une tâche Horloge

- Travail : Item courant; NB d'Items

EXEC 10  
E/S 5  
EXEC 12  
E/S 2  
EXEC 20

Item courant	Nb d'Items	N°Processeur	Temps Début
EXE	10		
E/S	5		
EXE	12		
E/S	2		
EXE	20		

16

CENTRALE  
L Y O N

## Deux solutions

==> **modèle à progression indépendante de chaque processeur** : le processeur avance directement du temps demandé par le travail.  
 Approche très séduisante, mais dans ce cas on trouve dans les files (FTB, FTD, FTT) des travaux dans le désordre. On peut essayer d'ordonner, mais cela ne marche pas.

==> **modèle avec l'horloge virtuelle** : chaque processeur doit recevoir un top d'horloge pour pouvoir progresser dans son exécution. Il faut donc une tâche HORLOGE et choisir un bon modèle de rendez-vous: qui sollicite, qui propose et accepte. Attention à la famine et aux blocages.

- Etudier la bonne terminaison de la simulation (sur nombre de travaux terminés).

Bertrand DAVID : Génie Logiciel

BTD/GL/TD2

17

CENTRALE  
L Y O N

## Architecture n° 3: approche décentralisée avec gestion du temps

The diagram illustrates a decentralized architecture with time management. It features several task objects represented as boxes with labels: 'Tâche: P1', 'Tâche: P2', 'Tâche: P3', 'Tâche: FTB', 'Tâche: FTD', 'Tâche: FTT', 'Tâche: HORLOGE', and 'Tâche: P/E/S'. Each task object has a 'PRENDRE' (take) and 'RENDRE' (return) message. Arrows labeled 'TEMPS' (time) show the flow of time between tasks, indicating dependencies and synchronization points. The 'HORLOGE' task is central, and the 'P/E/S' task is at the bottom right. The tasks are arranged in a roughly parallel, staggered fashion, suggesting a decentralized execution model.

Bertrand DAVID : Génie Logiciel

BTD/GL/TD2

18

CENTRALE  
L Y O N

## Ecriture "pseudo-ADA" des tâches HORLOGE et PROCESSEUR

Bertrand DAVID : Génie Logiciel

```

task HORLOGE
task body is
LOOP
  FOR I in 1..N DO
    P(I).Temps
  END FOR

  PES.Temps
END LOOP
end horloge

task type PROCESSEUR
entry Temps
end

task body is
LOOP
  SELECT
    FTD.Prendre (T)
    Etat:= Travail
  ELSE
    FTD.Vide
    Etat:= FTDVide
  END SELECT
  IF Etat= FTDVide then
    SELECT
      FTI.Prendre (T)
      Etat:= Travail
    ELSE
      FTI.Vide
      Etat:=FTIVide
    END SELECT
  END IF;

  IF (ETAT= FTIVide) THEN
    -- AND (ETAT=FTDVide)
    THEN Accept.Temps
  END IF
  IF Etat = Travail THEN
    FOR I in 1..UT DO
      Accept.Temps
    END FOR
  END IF
  IF Etat = Travail-fini THEN
    FTI.Rendre
    ELSE FTB.Rendre
  END IF
  END IF
END LOOP
end processeur

```

BTD/GL/TD2

19

CENTRALE  
L Y O N

## TD à faire

Bertrand DAVID : Génie Logiciel

- **Simulation d'un de fonctionnement d'un environnement de e-Livraison :**
  - **Quoi simuler ?**
    - Choisir des simplifications
    - Trouver des éléments significatifs
  - **Comment simuler ?**
    - A l'aide de tâches ADA
    - Quelle(s) architecture(s) ?
    - Comment montrer les résultats ?

BTD/GL/TD2

20

CENTRALE  
L Y O N

## e-Livraison

- L'objectif de ce TD est de simuler la distribution dans une ville en s'appuyant sur le concept de e-Livraison, c'est-à-dire la distribution avec possibilité de réserver au préalable des aires de livraison et éventuellement changer cette réservation dynamiquement si un empêchement se produit (retard, prise en compte de livraison urgente, changement de trajet, ...)

Bertrand DAVID : Génie Logiciel

BTD/GL/TD2

21

CENTRALE  
L Y O N

## Notions importantes :

- Trajet : succession de Segments de Transport à parcourir et d'Aires de Livraisons à visiter
- Segment de Transport (ST) : parcours élémentaire de déplacement
- Aire de Livraison (AL) : lieu d'arrêt pour livraison
- Trajet : succession de ST et d'AL avec durée correspondante
- Ville : graphe composé de ST et pouvant avoir aux extrémités des ST des AL.
- Camion = Chauffeur : effectuant une livraison

Bertrand DAVID : Génie Logiciel

BTD/GL/TD2

22

CENTRALE  
L Y O N

## Une simulation est définie par :

- NT nombre de trajets
- NC nombre de camions (chauffeurs)
- NST nombre de segments de transport
- NAL nombre d'aires de livraison

Bertrand DAVID : Génie Logiciel

BTD/GL/TD2 23

CENTRALE  
L Y O N

## Trajet

- **Trajet** : Etape courante; NB d'Items

ST1	1
AL2	5
ST3	1
ST5	2
AL3	20
ST7	1

Etape courante	Nb d'Items		
Nature	Durée	Temps Début	Temps Fin
ST1	1		
AL2	5		
ST3	1		
ST5	2		
AL3	20		
ST7	1		

BTD/GL/TD2 24

CENTRALE  
L Y O N

Bertrand DAVID : Génie Logiciel

## Deux approches :

- **Grphe explicite de la ville comme support d'architecture du simulateur :**
  - Le simulateur reflète la topologie du réseau de la ville et l'emplacement des Aires de Livraison
- **Grphe implicite comme support de l'architecture du simulateur :**
  - On interprète les trajets, considérés comme justes par rapport à la topologie du réseau
- **Avantage / inconvénients de chaque cas :**
  - **Explicite :** on contrôle la validité des trajets proposés, mais l'approche est rigide et toute modification lourde
  - **Implicite :** on ne contrôle pas la validité, mais l'approche est souple, sans modification de l'architecture lors d'un changement de topologie et d'emplacement des Aires de Livraison

BTD/GL/TD2 25

CENTRALE  
L Y O N

Bertrand DAVID : Génie Logiciel

## Composants de la structure et principes de l'architecture

- **Composant de base : File d'entrée et Processeur (STi ou ALj).**

```

graph LR
    File[File] --> STiALi[STi/ALi]
  
```

- **Dans le cas implicite le chauffeur envoie le trajet dans le prochain élément à exécuter (ST ou AL), c'est-à-dire dans la file du processeur correspondant STi, ALj, ....**
- **A la fin de l'exécution de l'élément du trajet, le processeur retourne le Trajet dans la file d'entrée du Chauffeur, qui l'oriente vers le prochain processeur (STi, ALj).**

BTD/GL/TD2 26

CENTRALE  
L Y O N

Bertrand DAVID : Génie Logiciel

## Trois cas à considérer :

1. **Simulation FIFO**, exécution dans l'ordre d'apparition
2. **Simulation avec respect des Réservations** : files à priorité, pour permettre à ceux qui ont des réservations de passer devant.
3. **Simulation avec la modification dynamique des Réservations**, le chauffeur change le trajet pour mieux s'en sortir

BTD/GL/TD2 27

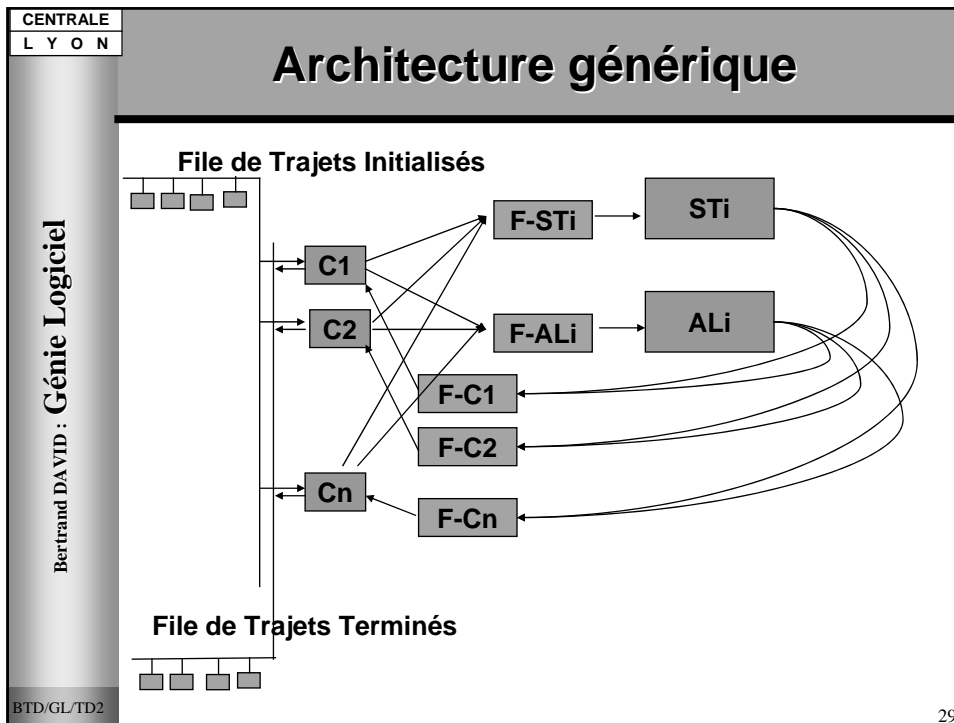
CENTRALE  
L Y O N

Bertrand DAVID : Génie Logiciel

## Architecture (début)

- FTI : file des trajets à effectuer
- FTT : file des trajets effectués
- Chauffeur accède à la FTI pour prendre un Trajet.
- Quand un trajet est terminé, le chauffeur l'enfile dans FTT

BTD/GL/TD2 28



CENTRALE  
L Y O N

## Dossier à rendre

- Décrire l'approche : principes, éléments significatifs, modélisation des travaux, comment ça fonctionne.
- Faire une simulation à la main d'un cas simple
- Elaborer une architecture (composants et relations, notamment de type Rendez-vous ou appels de services) pour le cas FIFO et donner le code des tâches ADA correspondantes
- Montrer les évolutions (architecture et code) pour gérer les priorités liées aux réservations
- Discuter comment arriver à mettre en place le 3° cas. Donner des explications sous forme d'une rédaction et ajouter éventuellement des modifications de l'architecture et du fonctionnement (code).

Bertrand DAVID : Génie Logiciel

30