CENTRALE
L Y O N

BE n°1-3

Techniques de simulation à base de multitâche

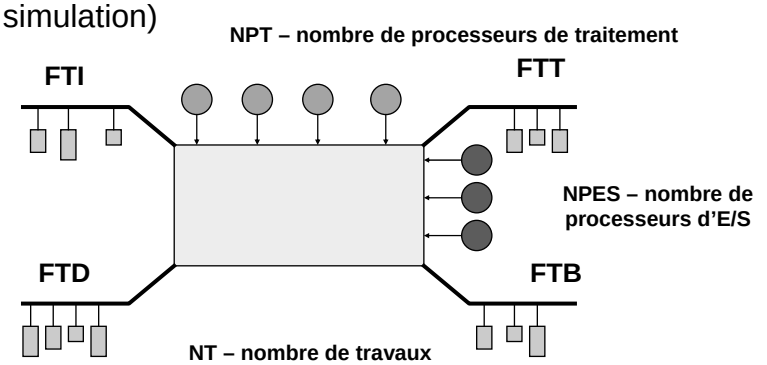
- Exemple de simulation de multiprocesseur
- BE : simulation de fonctionnement d'une application coopérative sur une architecture Client – Serveur par échange de messages

CENTRALE
L Y O N

Bertrand DAVID : Génie Logiciel

BE n°1 : 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)



CENTRALE LYON

Bertrand DAVID : Génie Logiciel

Démarche

- Présentation du problème : principes retenus pour la simulation du multiprocesseur
- Mettre en place un cas (choix du nombre de travaux NT, de processeurs de traitement NPT et de processeurs d'E/S NPES
- Faire à la main la simulation et bâtir des graphiques de bilan manuellement :
  - Processeurs / temps: occupation des processeurs
  - Travaux / temps: traitement des travaux
- Architectures du simulateur : représentation graphique
- Architectures du simulateur : description textuelle (en ADA)
- Prise en compte de la gestion du temps

BTD/GL/BE 3

CENTRALE LYON

Bertrand DAVID : Génie Logiciel

Architecture du multiprocesseur

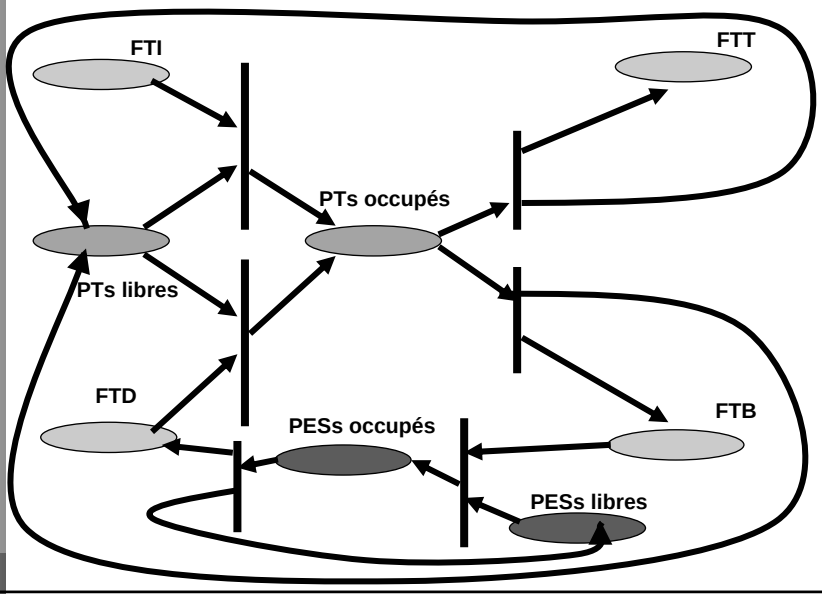
- Structure:
  - Mémoire commune partagée
  - NPT processeurs identiques partageant la mémoire
  - NPES processeurs spécialisés en Entrées-Sorties
  - 1 file de travaux initialisés FTI
  - 1 file de travaux terminés FTT
- Fonctionnement :
  - Un processeur traite un travail tant qu'il peut:
    PAS DE PREEMPTION
  - Les entrées-sorties sont à la charge du processeur d'E/S.
  - La suite du travail peut être effectuée par un autre processeur.
  - Le principe de fonctionnement peut être décrit par le réseau de Petri ci-joint.

BTD/GL/BE 4

CENTRALE LYON

Bertrand DAVID : Génie Logiciel

Modélisation par réseau de PETRI



BTD/GL/BE

5

CENTRALE LYON

Bertrand DAVID : Génie Logiciel

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

BTD/GL/BE

6

CENTRALE LYON

Bertrand DAVID : Génie Logiciel

Modélisation du travail

- 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/BE

7

CENTRALE LYON

Bertrand DAVID : Génie Logiciel

Approche de modélisation par tâches ADA

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

BTD/GL/BE

8

CENTRALE LYON

Bertrand DAVID : Génie Logiciel

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

BTD/GL/BE 9

CENTRALE LYON

Bertrand DAVID : Génie Logiciel

Protection par le Maître

- **Métaphore du banquet :**

  Comment protéger la nourriture et les boissons contre les bousculades

BTD/GL/BE 10

CENTRALE LYON

Bertrand DAVID : Génie Logiciel

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

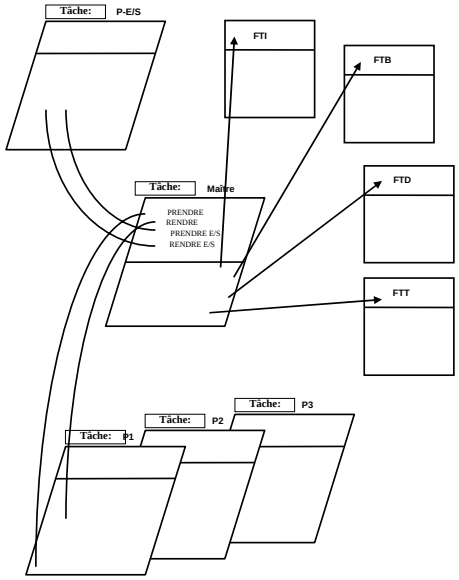
BTD/GL/BE

11

CENTRALE LYON

Bertrand DAVID : Génie Logiciel

Architecture n° 1 : approche centralisée maître-esclaves



BTD/GL/BE

12

CENTRALE LYON

Bertrand DAVID : Génie Logiciel

Approche distribuée

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

BTD/GL/BE

13

CENTRALE LYON

Bertrand DAVID : Génie Logiciel

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)

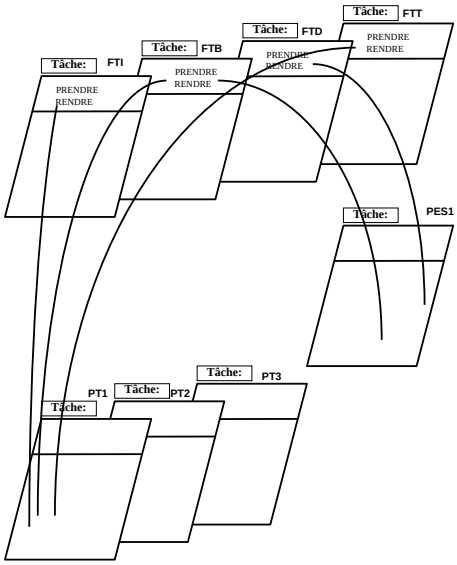
BTD/GL/BE

14

CENTRALE LYON

Bertrand DAVID : Génie Logiciel

Architecture n° 2 : approche décentralisée



BTD/GL/BE

15

CENTRALE LYON

Bertrand DAVID : Génie Logiciel

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		
Nature	Durée	N°Processeur	Temps Début
EXE	10		
E/S	5		
EXE	12		
E/S	2		
EXE	20		

BTD/GL/BE

16

CENTRALE LYON

Bertrand DAVID : Génie Logiciel

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).

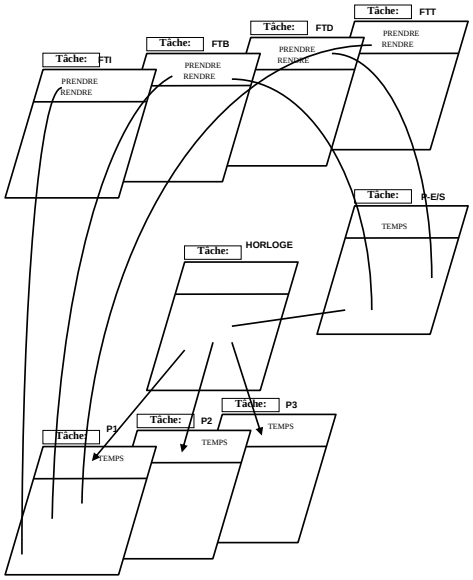
BTD/GL/BE

17

CENTRALE LYON

Bertrand DAVID : Génie Logiciel

Architecture n° 3: approche décentralisée avec gestion du temps



BTD/GL/BE

18

CENTRALE L Y O N

Bertrand DAVID : Génie Logiciel

Ecriture "pseudo-ADA" des tâches HORLOGE et PROCESSEUR

```
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
                    Etat:= FTDVide
            END SELECT
IF Etat= FTDVide then
            SELECT
                    FTI.Prendre (T)
                    Etat:= Travail
            ELSE
                    Etat:=FTIVide
            END SELECT
            END IF;
IF (ETAT= FTIVide)      -- AND (ETAT=FTDVide)
                    THEN        Accept.Temps
            END IF
IF Etat = Travail THEN
                    LOOP
                            FOR I in 1..UT DO
                            Accept.Temps
                            END FOR
                    END LOOP
IF Etat = Travail-fini THEN FTT.Rendre
                    ELSE        FTB.Rendre
                    END IF
            END IF
END LOOP
end processeur
```

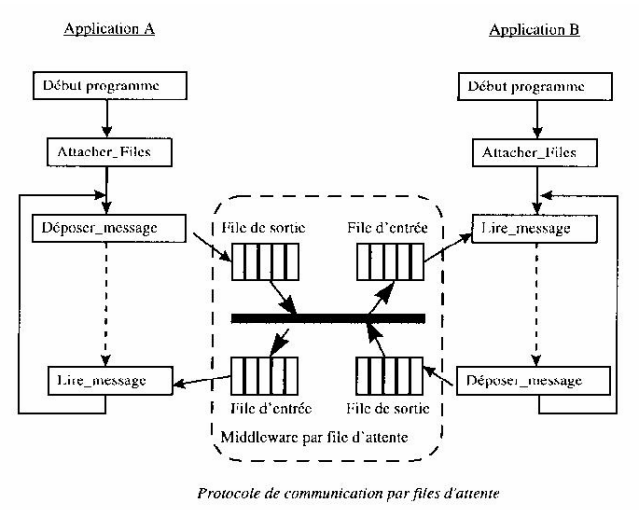
BTD/GL/BE

19

CENTRALE L Y O N

Bertrand DAVID : Génie Logiciel

Client - Serveur



Protocole de communication par files d'attente

BTD/GL/BE

20

CENTRALE LYON

Bertrand DAVID : Génie Logiciel

Différentes configurations possibles

Cas	Clients	Middleware	Serveur
N-1-1	N	1	1
N-L-K	N	Plusieurs identiques	Plusieurs identiques
N-3(Mi-Pi)	N	Différents en amont de différents types de serveurs spécialisés	Différents types spécialisés : Calcul, Impression, Données

BTD/GL/BE

21

CENTRALE LYON

Bertrand DAVID : Génie Logiciel

Travail Coopératif

- **Les outils de travail coopératifs (appelés groupware ou collecticiel) ont pour but de permettre le travail à plusieurs.**
- **Le cas qui nous intéresse est celui de travail coopératif synchrone à distance.**
- **Dans ce cas, les différents utilisateurs travaillent ensemble à distance sur une même application, chacun contribuant à sa façon à l'œuvre collective se mettant en place dans l'application partagée.**
- **Différentes architectures sont possibles, nous allons retenir celle qui épouse l'organisation de type client/serveur :**

  Les différents utilisateurs travaillent à partir de leurs postes respectifs qui sont les clients du serveur de coopération. Le serveur reçoit des clients des demandes d'exécution et répercute les résultats vers tous.

BTD/GL/BE

22

CENTRALE LYON

Bertrand DAVID : Génie Logiciel

Outil de simulation du TC

- **Il s'agit de mettre en place un outil de simulation qui permettra d'observer le fonctionnement de l'application coopérative du point de vue temporel. En effet, pour des applications synchrones, il est primordial que les évolutions effectuées par les différents participants apparaissent dans un délai satisfaisant sur les postes de tous les acteurs.**
  - **Dans le cas de l'approche centralisée (l'application se trouvant sur le serveur), l'exécution des différentes actions doit se faire exclusivement (pour ne pas perturber le fonctionnement).**
  - **Si le grain propagé (commande ou données résultantes) est assez petit pour que le délai soit également suffisamment petit pour ne pas être perceptible par les acteurs, on peut se contenter de l'exclusivité implicite.**
  - **Si le grain est plus grand on doit mettre en place l'exclusivité explicite et indiquer clairement « la prise de parole » pour signifier aux autres qu'ils ne peuvent pas pour l'instant agir sur le même grain.**

BTD/GL/BE

23

CENTRALE LYON

Bertrand DAVID : Génie Logiciel

Coordination implicite ou explicite

- **La coordination peut donc être soit implicite (pour un petit grain), soit explicite (pour un grain plus grand, voire l'espace de travail tout entier).**
- **Dans le cas explicite on doit mettre en place des commandes :**
  - **« Demander la parole »,**
  - **« Attribuer la parole »,**
  - **« Prendre la parole »,**
  - **« Rendre la parole ».**
- **Demander, Prendre et Rendre étant du ressort du client, Attribuer étant du ressort du serveur.**

Exemples de Coopération :
- Implicite de type vote
- Explicite de type écriture

BTD/GL/BE

24

CENTRALE LYON

Bertrand DAVID : Génie Logiciel

Travail demandé

- **On vous demande d'adapter le simulateur étudié en BE à ces deux comportements de systèmes coopératifs. Le but du simulateur est de se faire une idée sur le fonctionnement et déterminer la viabilité ou non viabilité du modèle proposé par rapport aux besoins temporels de chaque acteur :**
  - **Donner le descripteur du travail dans le cas de l'exclusivité implicite et décrire la circulation des informations dans ce cas.**
  - **Donner le descripteur du travail dans le cas de l'exclusivité explicite et décrire la circulation des informations dans ce cas.**
  - **Décrire brièvement le fonctionnement du client et du serveur dans chacun des cas identifiés.**
  - **Donner l'architecture du simulateur supportant l'architecture client/serveur pour le travail coopératif.**

BTD/GL/BE

25

CENTRALE LYON

Bertrand DAVID : Génie Logiciel

Coopération implicite

- **Application de type vote :**
  - **Client : S'identifier**
  - **Serveur : Accepter nouvel acteur**
  - **Serveur : Annoncer le vote**
  - **Client : Demander la question**
  - **Serveur : Poser et diffuser la question**
  - **Client : Voter – envoyer le vote**
  - **Serveur : Recevoir des réponses**
  - **Serveur : Elaborer et diffuser la synthèse de résultat**
  - **Client : Observer les résultats**
  - **Serveur : Passer au vote suivant**
  - **Serveur : Annoncer le vote**
  - **Client : Demander la question**
  - **Serveur : Poser et diffuser la question**
  - **Client : Voter – envoyer le vote**
  - **Serveur : Recevoir des réponses**
  - **Serveur : Elaborer et diffuser la synthèse de résultat**
  - **Client : Observer les résultats**
  - **Client : Partir**

BTD/GL/BE

26

CENTRALE LYON

Bertrand DAVID : Génie Logiciel

Coopération explicite

- **Application de type Dessin collaboratif ou Ecriture collaborative :**
  - **Client : S'identifier**
  - **Serveur : Accepter nouveau acteur**
  - **Serveur : Diffuser l'état du travail**
  - **Client : Demander la parole**
  - **Serveur : Attribuer la parole**
  - **Client : Prendre la parole et Faire**
  - **Serveur : Diffuser l'état du travail**
  - **Client : Rendre la parole**
  - **Serveur : Diffuser l'état du travail**
  - **Client : Demander la parole**
  - **Serveur : Attribuer la parole**
  - **Client : Prendre la Parole et Faire**
  - **Serveur : Diffuser l'état du travail**
  - **Client : Rendre la parole**
  - **Serveur : Diffuser l'état du travail**
  - **Client : Partir**

BTD/GL/BE

27

CENTRALE LYON

Bertrand DAVID : Génie Logiciel

BE à faire

- **Simulation d'une application coopérative fonctionnant sur l'architecture Client-Serveur à base de messages :**
  - **Quoi simuler ?**
    - **Choisir des simplifications**
    - **Trouver des éléments significatifs**
  - **Comment simuler ?**
    - **A l'aide de threads Java**
    - **Quelle(s) architecture(s) ?**
    - **Comment montrer les résultats ?**

BTD/GL/BE

28

CENTRALE
L Y O N

Bertrand DAVID : Génie Logiciel

Comment exporter et importer avec VisualAge :

Pour exporter :

- Il faut versionner le projet puis exporter le Référentiel en se rappelant les noms de projet et de package.

Pour importer :

- Il faut créer un nouveau projet vide. Ce nouveau projet apparaît comme un dossier vide.
- Sélectionner ce projet, cliquer sur le bouton droit et choisir importer...
- Choisir Référentiel, puis Projets et aller chercher le fichier XXX.dat à l'aide du bouton Parcourir...
- Une fois le fichier XXX.dat choisi, cliquer sur Détails et cocher le projet YYY. Vérifiez que la version Z.Z est également cochée.
- Valider 2 fois : VisualAge charge le projet.
- A ce stade, rien n'a changé dans l'interface graphique de VisualAge : le programme a en fait chargé le projet dans une base de données. Il faut donc charger le projet définitivement.
- Sélectionner de nouveau le projet dernièrement créé. Cliquez sur la 4ème icône de la barre de boutons pour créer un nouveau package dans le projet.
- Choisir Ajouter un package à partir d'un référentiel. Rechercher le package correspondant. Valider.
- Normalement, c'est fini et ça marche

BTD/GL/BE

29