**Project : The Energy Market**

Raphaël Monin Hind Amhache

1. **Le design du projet**

Homes

- Process qui communique avec les autres homes via des « message queues »

- Vend de l’énergie à Market ou la donne à une autre maison

Interprocessing : Message queues

Communication enfant-parent

Envoi de signaux

Facteurs externes

- Evénements aléatoires d’ordre politique, social…

- Processus enfant de Market

Variation de la température

- Mémoire partagée avec les données de la température

- Processus enfant de Market

Schéma représentant la relation entre les 4 processus du programme

**3.1. Market**

Chaque thread dans Market va calculer une partie de l’équation en appelant le processus correspondant :

: Calculé par les processus facteurs internes (fonctions weather, homes…)

 : Calculé par les processus facteurs externes (fonction extFactors)

NB : Il faut au moins 2 facteurs externes et 2 facteurs internes

**3.2. Homes :**

Une maison possède 2 paramètres : Son identifiant et son type. Elle retourne le facteur.

* Il existe 3 types de maisons :
  + Type 1 : Donne toujours son excès d’énergie à une autre maison ;
  + Type 2 : Vend toujours son excès d’énergie dans le marché ;
  + Type 3 : Vend son excès d’énergie dans le marché **si** aucune maison n’a besoin d’énergie ;

**Comment savoir si une maison a un surplus d’énergie ?**

On le définit par le **rendement,** qu’on appellera R dans notre programme. Il sera compris dans le calcul du facteur.

Nous aurons donc 3 cas de figure :

* Si R est inférieur à 1, augmente et le prix de l’énergie subit donc une hausse, car la maison consomme plus qu’elle produit. Elle est **demandeuse** d’énergie et sera placée dans une file d’attente (queue) ;
* Si R est entre 1 et 1.1, elle ne fait rien (on laisse une marge de 10% d’énergie à la maison) ;
* Si R est supérieur à 1,1, elle produit plus qu’elle consomme, le surplus d’énergie X sera géré suivant le type de la maison (exemple : Si R=1.4, X=0.3 🡪 Surplus de 30%) :
* Si type 1 🡪 X sera donné à une autre maison demandeuse ;
* Si type 2 🡪 X sera vendu dans le marché, et le prix de l’énergie baissent ;
* Si type 3 🡪 Si la file d’attente qui représente les maisons demandeuses d’énergie est vide, la maison vend X dans le marché ;

**3.3. La température**

La variation de la température:

* Est un processus enfant de Market qui s’exécute à une certaine fréquence, le processus Market exécute toujours le processus température et attend qu’il termine avant de calculer l’énergie ;
* A une mémoire partagée avec les données de la température (sous forme d’une fonction retournant un entier aléatoire entre 0 et 30) ;

**3.4. Facteurs externes**

C’est un processus exécutant une fonction qui génère au moins 2 événements aléatoires (sociaux, politiques...). Cette fonction retourne, jusqu’à , étant le nombre d’évènements. Ces retours de fonction peuvent prendre 2 valeurs :

* + 1 si un événement a eu lieu ;
  + 0 si l’événement dont il est question n’a plus lieu ;

Dans les deux cas, une communication interprocessus doit se faire : Le processus facteurs externes envoie un signal à Market avec comme donnée 1 ou 0, pour que ce dernier recalcule l’énergie.

**Comment s’établira la communication entre ces 4 processus ?**

**Maisons entre elles**: Création d’une *message queue* de maisons demandeuses d’énergie. Si R<1, sera un élément de la file d’attente et représentera la quantité d’énergie que demande la maison. Une maison de types 1 ou 3 consulte cette queue et donne son énergie à la première maison de la file d’attente.

**Les maisons et le marché :**

Création de 2 queues :

* Une queue dont les éléments sont des flottants représentants la quantité d’énergie que la maison souhaite vendre ;
* Une queue dont les éléments sont des flottants représentants la quantité d’énergie que la maison souhaite acheter ;

Le marché consulte toujours ces 2 queues avant de finir son exécution, pour recalculer l’énergie.

**La maison et la température :**

Pour calculer le taux de consommation d’une maison, puisque plus la température augmente moins on consomme de l’énergie, il faut utiliser une formule exponentielle qui calcule la nouvelle quantité d’énergie consommée à t en fonction de la nouvelle température après avoir appelé la fonction weather(), la quantité d’énergie consommée à t-1 et de la température à t-1 qui sera accessible à travers une mémoire partagée. Elle sera synchronisée et protégée à travers des mutex et des semaphore.

**La température et le marché** : le marché exécute weather(), qui est son processus fils, et attend qu’il termine avant de recalculer le prix de l’énergie.

**Les facteurs externes et le marché** : Communications avec des signaux.

**Initialisation des processus : Prix initial de l’énergie, valeur de la température, nombre de maisons… (On doit fixer le nombre max de maisons, le nombre max de transactions entre market et maisons, les valeurs aléatoires sont comprise entre quels nombres ? définir les coef )**

**Check list de notre rapport**

● relationships between processes (parent-child or unrelated) 🡪 fait dans le schema, le seul process enfant est température

● messages exchanged between processes along with their types 🡪 en cours

● data structures stored in shared memory and how they are accessed 🡪 en cours, expliquer ce qui est mis en mémoire partagée comme les valeurs de la température

● synchronization primitives to protect access to shared resources or to count resources 🡪 en cours

● signals exchanged between processes, if any, and their types 🡪 fait (les seuls signaux à transmettre sont ceux des facteurs externes)

● tubes involved in pairwise process communication, if any, and their types 🡪 en cours

+ Pseudo code des 4 processus