



UNIVERSITÉ DU HAVRE

Rapport:Master 2 MATIS

Simulation d'un SGBD temps réel

Auteurs :
Sidi LEKHALIFA

Responsables :
M. Duvallet C

Mars 2016

Table des matières

1	Génération de la base de données temps réel	3
2	Génération des transactions temps réel	5
2.1	Génération des transactions des mises à jours	5
2.2	Génération des transactions utilisateurs	6
3	Exécution des transactions	6
	Conclusion	8
	Références	9

Rappel du sujet

Une bases de données est un ensemble des informations(données) stockées sur un support informatique, elle est facilement structurée et organisée de manières à pouvoir facilement le consulté ou modifier son contenu.

Pour interagir avec la base de donnée on est confronter à utiliser des logiciels qui permettent de gérer les base de données, c'est qu'on appelle **Système de gestion de bases de données**.

Depuis dernière temps on voit l'utilisation de SGBD temps réel qui permettent de répondre au besoin des applications telques :

- Pilotage d'usines automatisées ;
- Applications embarquées ;
- Applications mobile... ;

Qui demande les respect des contraintes temporelles, de maintient de la cohérence des données ainsi que la répartition des données.

Dans ce rapport nous allons faire un simulateur d'un SGBD temps réel. Pour arriver à notre objectif nous allons procéder comme suit :

- Dans un première temps on va générer les données temps réel qui vont re-présenter notre Base de données temps réel ;
- deuxiément on va générer les transactions temps réel ;
- Ensuite nous allons exécuter les transactions temps réel et mesurer leur taux de réussite en termes de respect des échéances ;

Ce simulation va être effectuer en java.

1 Génération de la base de données temps réel

Pour concevoir un simulateur de base de données temps réel on va se répartir sur trois branches importantes qui sont :

- **La base de donnée ;**
- **L'ordonnaceur** qui va correspondre à l'exécution des transactions qui vont être mis sur une file d'attente ;
- **Les transactions** vont posséder chacune une échéance et une durée d'exécution et vont posséder également des ressources soit en lecture soit en écriture ;

Pour réaliser le simulateur de base de données temps réel et classique , je vais créer trois classes qui sont les suivantes :

- **SGBD** : qui va représenter la base de donnée classique.
- **SGBDTR** : qui va représenter la base de donnée temps réel. Cette classe va hériter de la classe précédente et en plus on rajoute trois champs(à savoir la Valeur, Estampille et la durée de validité).
- **saisi** : qui va correspondre au formulaire de saisie.

La figure suivante illustre le format général d'une donnée :



FIGURE 1 – Représentation générale d'une données classique et temps réel.

La classe de saisi va permettre d'interagir avec l'utilisateur il va être comme suit dans un première temps :

The image shows a Java Swing window with a standard title bar (minimize, maximize, close buttons). The window has a light gray background. It contains three labels on the left side, each followed by an input field on the right:

- Nombre Réel à Simuler**: followed by a single-line text input field.
- Intervale de Validité**: followed by two single-line text input fields side-by-side.
- Nombre Classique à Simuler**: followed by a single-line text input field.

At the bottom right of the window, there is a dark gray button with the word **Valider** in blue text.

FIGURE 2 – Formulaire de saisie

2 Génération des transactions temps réel

Chaque transactions temps réel doit finir son exécutions avant son échéance sinon elle va être abandonner. Une fois qu'elle à fini sont exécution elle va être acheminée vers une file d'attente.

la génération va être de deux types :

- Génération des transactions de mise à jour ;
- Génération des transactions utilisateurs ;

2.1 Génération des transactions des mises à jours

Dans cette partie nous allons associer au données temps réel dans la base une mise à jour périodiquement, avec une période de $\frac{2}{3}$ de la durée de validité de données.

On va créer ici une classe au quel je l'appelle **GenerateTransactionMS** qui hérite de la classe **Runnable**. Dans cette classe on va ajouter des attributs tels que identifiant de transactions, la période de la mise à jour d'une données temps réels, une échéance et une données temps réels. Ensuite je vais déclarer une file d'attente pour mettre les différentes transactions générer.

Pour pouvoir interagir avec l'utilisateur on modifie notre classe de saisie pour que l'utilisateur puisse générer des transactions de mise à jours et d'utilisateurs qu'on verra dans la prochaine partie.

2.2 Génération des transactions utilisateurs

Dans cette partie nous allons générer des transactions qui va comporter des opérations de lectures de données temps réel et de lecture, écriture des données classiques. les différentes transactions vont être générées suivant en processus de poisson.

On va créer ici une nouvelle classe que je vais l'appeler **GenerateTransactionsUSER** qui hérite aussi de la classe **Runnable**, est elle admet comme attribut le nombre de transactions (qui va correspondre à une valeur dans un intervalle), une durée de lectures temps réels et classiques, une durée d'écriture pour une données classiques, durée de simulation, une échéance, paramètre lambda de loi de poisson, une données temps réel et une données classiques ainsi qu'un file d'attente de données.

3 Exécution des transactions

Pour exécuter des transactions, on ouvre la classe de **saisie** et on l'exécute il va nous apparaître un formulaire comme suit :

A screenshot of a software window titled "Formulaire de saisie" (Input Form). The window contains several input fields for simulation parameters. The fields are arranged in a list on the left, with corresponding input boxes on the right. The parameters are: "Nombre Réel à Simuler", "Nombre Classique à Simuler", "Intervale de Validité" (with two sub-fields), "Verrouillage(0 ou 1)", "nombre opération" (with two sub-fields), "Durée Lecture TR", "Durée Lecture CL", "Durée Ecriture CL", "Durée Simulation", and "lambda". A "Valider" button is located at the bottom right of the form.

Paramètre	Champ de saisie
Nombre Réel à Simuler	[Champ de saisie]
Nombre Classique à Simuler	[Champ de saisie]
Intervale de Validité	[Champ de saisie] [Champ de saisie]
Verrouillage(0 ou 1)	[Champ de saisie]
nombre opération	[Champ de saisie] [Champ de saisie]
Durée Lecture TR	[Champ de saisie]
Durée Lecture CL	[Champ de saisie]
Durée Ecriture CL	[Champ de saisie]
Durée Simulation	[Champ de saisie]
lambda	[Champ de saisie]

Valider

FIGURE 3 – Formulaire de saisie

on remplit les champs et on clique sur **valider**.

Conclusion

Pour conclure, dans ce travail je peux dire que j'étais content de faire ce projet qui m'a permis d'approfondir mes connaissances en développement java particulièrement l'utilisation de plugin **WindowsBuilder** que je les télécharger et installer. Ainsi que mon approfondissement dans le domaine de base de données temps réels et la gestion des transactions ainsi que les notions de la cohérence temporelles.

Références

- C. Duvallet et Al.** *LES SGBD TEMPS REEL* 1999 page 479 - 516
- C. Duvallet** Cours. *LES SYSTEMES DE GESTION DE BASES DE DONNEES TEMPS REEL*
- B. Dadeg** *CONTRIBUTIONS A LA GESTION DES TRANSACTIONS DANS LES SGBD TEMPS REEL*
- J. Haubert et Al.** *J-RADEX : UN SIMULATEUR DE SGBDTR CONVIVIAL*