Compte rendu TP Trafic de Lyon - C++

Spécification

*I. Choix généraux*

Nous avons choisis une solution intermédiaire entre un stockage des données brutes uniquement et une méthode orientée Big Data où l’on ne stockerait que les résultats des requêtes. En effet ici nous ne nous intéressons qu’à l’intervalle de temps où un capteur est dans un état donné et non à ses états successifs. Ceci signifie que nous ne stockons pas les valeurs envoyées par le capteur mais nous en déduisons plutôt la durée durant laquelle le capteur est dans un état donné. Ce léger calcul est peu couteux en CPU et nous évitera ultérieurement de parcourir un trop grand nombre de données.

De plus, nous stockons et actualisons également le résultat de la requête ‘MAX\_TS’ (le moment avec le plus de bouchons simultanés). Nous avons fait ce choix car cette requête était particulièrement gourmande en temps de calcul par rapport aux autres requêtes. De cette manière nous pensons avoir amélioré les performances de notre application.

*II. Structures de données*

*III. Description technique*

***III.1. Classe ‘Main’***

***III.2. Classe ‘GestionTrafic’***

*Rôle de la classe*

*Les méthodes/constructeurs/destructeurs*

*Ses attributs*

***III.3. Classe ‘Evenements’***

*Rôle de la classe*

*Les méthodes/constructeurs/destructeurs*

*Ses attributs*

***III.4. Classe ‘EvenementsCapteurs’***

*Rôle de la classe*

*Les méthodes/constructeurs/destructeurs*

*Ses attributs*

***III.5. Classe ‘ListeDatesMaxBouchons’***

*Rôle de la classe*

*Les méthodes/constructeurs/destructeurs*

*Ses attributs*

***III.6. Classe ‘Date’***

*Rôle de la classe*

*Les méthodes/constructeurs/destructeurs*

*Ses attributs*

***III.7. Classe ‘ArbreIdentifiants’***

*Rôle de la classe*

*Les méthodes/constructeurs/destructeurs*

*Ses attributs*

***III.8. Classe ‘Constantes’***

*Rôle de la classe*

*Les méthodes/constructeurs/destructeurs*

*Ses attributs*

***III.9. Classe TestsGestionTrafic***

*Voir documents de description des tests fonctionnels.*