Fiche Technique de Projet – Programmation Système & Réseau

Programme de *monitoring* : Surveillance de calcul

Formation : ING2 – GSI

Nature : Conception, réalisation

Année : 2020/2021

1. Préambule

Il vous est demandé de faire une analyse du système avant de vous lancer dans le code. Une analyse bien faite est un prérequis indispensable avant de commencer à coder. Vous devez donc rendre en plus des différents codes demandés, un rapport expliquant clairement votre analyse des différentes parties dans un fichier séparé.

Par ailleurs, il vous est demandé de produire un code correct. C'est à dire qui compile, et sans *warning* (en utilisant l'option `-Wall`).

2. Monitoring de calcul

La programmation multiprocessus permet d'accélérer les traitements. Dans ce projet, nous nous proposons de résoudre un cas pratique de la programmation multiprocessus.

Avec l'avènement de la programmation multiprocessus, l'utilisation de programme de *monitoring* est devenue indispensable pour visualiser l'état d'un système. Le moniteur est le point d'entrée de l'application, il permet à l'utilisateur souhaitant visualiser le système de fournir des informations nécessaires à la visualisation. Traditionnellement, ce moniteur possède une interface web, mais dans notre cas des affichages consoles suffiront. Le moniteur collecte les données envoyées par les différents (sous)processus, réalise une synthèse, puis présente les informations à l'utilisateur. Les moniteurs peuvent être, ou non, dédiés à un système. Pour plus de simplicité, nous nous concentrerons sur un moniteur dédié à la surveillance de calcul. Le calcul n'étant pas l'objectif du projet, nous nous contenterons d'implémenter une "bête" somme d'entier distribuée sur *m* processus. Toutes les 2 secondes, chaque processus calculateur enverra un rapport. Ce rapport sera constitué de deux parties : la somme partielle calculée ; le temps d'activité du processus. Pour simuler le "dur" calcul, vous mettrez un temps de pause entre chaque addition de 1 seconde.

Le moniteur doit dans un premier temps connaître les processus qu'il doit "*monitorer*". Une fois l'ensemble des processus répertoriés, il devra faire une synthèse de l'état des processus. Pour des raisons de "performance" cette synthèse n'est effectuée que toutes les 3 secondes.

Cependant, il doit fournir un état du système à chaque fois que l'utilisateur lui demande. Le rapport que le moniteur doit fournir contient les informations suivantes :

- Nombre de processus calculateurs
- Somme totale partielle calculée
- Pour chaque processus
    - Somme partielle calculée
    - Temps d'exécution

3. Stratégie de l'échec

Dans un monde totalement distribué, chaque processus est localisé sur des machines différentes.

Ces machines pouvant tomber en panne, ou n'être plus accessible, il faut prévoir le cas où les processus puissent tomber en panne. Nous ne vous demandons pas de vous occuper de la partie distribution, mais de vous occuper de mettre en place, la gestion de l'échec.

4. Evil monkey

Pour tester ce type de système, il existe une solution. Il s'agit d'implémenter un processus de type *evil monkey*. C'est à dire un processus qui toutes les *x* secondes (*x* étant un nombre aléatoire), tue un processus parmi tous les processus concernés (sauf bien sûr le processus *evil monkey*).

5. Travail à réaliser

Le projet sera réalisé par groupe de 3/4 étudiants. Le travail à réaliser consiste à concevoir et mettre en œuvre une architecture de monitoring.

1ère partie : Moniteur & Calculateurs

1. Faire l'analyse du moniteur et des processus calculateurs :
    a. Conception de l'architecture
    b. Structure de données et moyens de communications utilisés
2. Implémenter le moniteur et les calculateurs
3. Créez un programme lanceur capable de lancer le moniteur et les *m* processus de calcul, ainsi que les bornes de départ et de fin des nombres à sommer. Vous veillerez à ce que la répartition entre chaque processus de calcul soit la plus équitable possible.

2ème partie : Stratégie d'échec

4. Quels sont les impacts sur le moniteur et les calculateurs ? proposez une solution ?
5. Implémenter les modifications proposées dans votre solution.

3ème partie : *Evil monkey*

6. Faire l'analyse du processus *evil monkey*.
7. Implémenter ce processus.

6. Livrables

1. Rapport d'analyse. Le rapport d'analyse doit comprendre obligatoirement :
   - Un diagramme de cas d'utilisation. Il n'est pas nécessaire de décrire complètement les cas d'utilisation (partie textuelle) mais il faut dessiner quelques diagrammes de séquences pour montrer les points clés.
   - Les ambiguïtés identifiées dans le sujet et les choix faits pour les lever.
   - L'architecture adoptée.
   - La politique de gestion de l'échec.
   - Les tests envisagés et leur planification.
   - Un manuel d'utilisation succinct.
   - La répartition du travail entre les membres du projet.
2. Une archive contenant tous les codes sources de votre projet.

7. Dates de remise des documents

Mardi 2 mars, Distribution du sujet du projet.

Vendredi 5 mars, envoie de la liste des groupes.

Dimanche 28 mars, 20 heures Remise du rapport du projet et des sources.