

Projet

But du travail pratique

Le but de ce travail pratique est de vous familiariser avec :

- La simulation à événements discrets
- La gestion de listes d'événements
- Les résultats statistiques
- La conception expérimentale

Note : Ce travail peut être réalisé individuellement ou en équipe de 2 étudiants.

Description

Le pavillon de l'humanité contient $F = 7$ étages et a normalement $N = 2$ ascenseurs fonctionnels. On aimerait connaître l'impact de l'ajout d'un troisième ascenseur sur le service aux utilisateurs.

Les gens arrivent au bâtiment à des moments aléatoires et entrent au rez-de-chaussée (étage 1) selon un processus d'arrivée de Poisson avec un taux moyen d'arrivée global de λ (par exemple, $\lambda = 0,5$ personne par minute). Ils demandent un ascenseur, y pénètrent, montent et descendent à un étage uniformément choisi au hasard parmi les étages restants ($F - 1$). Ils restent au travail sur cet étage choisi pour un temps donné aléatoirement (distribution exponentielle, avec une moyenne de 60 minutes), avant de retourner aux ascenseurs, d'en demander un, de descendre au premier étage et de quitter le bâtiment.

*La mesure de la performance d'intérêt est le **temps de réponse perçu par l'utilisateur**, qui est le temps écoulé entre demander un ascenseur et se rendre l'étage désiré (c.-à-d., une fois en montée et une fois en descente).*

Il existe de nombreuses variantes possibles à la configuration et au fonctionnement des ascenseurs: nombre, vitesse, capacité, ordonnancement. Votre but est d'explorer un petit sous-ensemble de ces possibilités et de comprendre l'impact sur le temps de réponse perçu par l'utilisateur.

La « performance » d'un ascenseur dépend grandement de son algorithme d'ordonnancement. En particulier, il existe plusieurs choix pour ce qu'un ascenseur peut faire lorsque plusieurs demandes sont en attente. Un choix consiste à traiter les demandes strictement dans l'ordre d'arrivée, appelé First-Come-First-Serve (premier arrivé, premier servi). Une autre consiste à répondre à la demande « la plus proche », quelle que soit la direction dans laquelle on se déplace. C'est ce qu'on appelle le « Shortest-Seek-Time-First » (SSTF). Un autre choix consiste à ne traiter que les demandes qui se trouvent dans la même direction verticale que celle en cours de déplacement. C'est ce qu'on appelle le

balayage linéaire ou Linear Scan. *Le choix parmi ces méthodes est particulièrement important lorsque la charge est élevée.*

Un autre facteur influençant la performance des ascenseurs est la politique de marche au ralenti/idle. C'est-à-dire, il y a plusieurs choix pour ce qu'un ascenseur devrait faire lorsqu'il est vide: rester où il est, aller à l'étage inférieur, aller à l'étage supérieur, ou aller à l'étage « du milieu ». *Cette politique est particulièrement importante pour les charges légères, mais moins importante pour les charges plus élevées, lorsque l'ascenseur se vide rarement.*

Vos expériences de simulation initiales n'auront qu'un seul ascenseur ($N = 1$). En termes d'autres paramètres, vous pouvez supposer que $F = 7$ et que le temps de mouvement entre les étages adjacents du bâtiment est toujours exactement de 10 secondes, indépendamment de l'occupation, de la distance ou de la direction parcourue par l'ascenseur. Vous pouvez supposer que l'ascenseur peut contenir un nombre illimité de personnes.

Exigences

Écrire une simulation à événements discrets qui modélise le fonctionnement des ascenseurs dans le pavillon des humanités de l'UQAC. Plus précisément, procédez comme suit:

- Concevoir et mettre en œuvre un modèle de simulation de l'ascenseur(s). Paramétrez raisonnablement votre simulation et documentez les hypothèses supplémentaires que vous faites. Instrumentez le simulateur de manière adéquate afin de pouvoir collecter les données/informations de temps nécessaires à l'analyse des résultats.
- Pour le scénario à un seul ascenseur, implémentez **deux** des algorithmes d'ordonnancement possibles: FCFS, SSTF ou Linear Scan. Mener une expérience de simulation pour voir lequel de ces algorithmes d'ordonnancement est le meilleur, en termes de temps de réponse moyen perçu par l'utilisateur. Utilisez uniquement le meilleur algorithme d'ordonnancement obtenu dans vos expériences de simulation.
- Pour $N = 1$, effectuez des expériences de simulation qui évaluent **deux** des politiques de ralenti suggérées pour voir l'impact, le cas échéant, sur le temps de réponse perçu par l'utilisateur. Choisissez la meilleure politique de ralenti pour vos expériences de simulation restantes.
- Augmentez le nombre d'ascenseurs de $N = 1$ à $N = 2$, et ensuite de $N = 2$ à $N = 3$. Indiquez l'impact, le cas échéant, sur le temps de réponse perçu par l'utilisateur. Justifiez toutes les hypothèses supplémentaires ou les décisions de conception que vous prenez.

Documents à remettre

- Le code source de votre programme de simulation,
- Un bref manuel d'utilisation décrivant comment compiler et utiliser votre simulateur,

- Une description des résultats générés à l'aide de votre programme ainsi que toute autre information pertinente (hypothèses effectuées, expériences, etc.)

Barème de correction (20% total)

- (10%) conception et mise en œuvre d'un simulateur d'événements discrets approprié (c.-à-d. Boucle de simulation principale, variables d'état, structures de données de liste d'événements, génération de nombres aléatoires, résultats statistiques, etc.)
- (3%) pour les résultats de simulation fournissant des réponses à la tâche de planification de l'ascenseur
- (2%) pour les résultats de simulation fournissant des réponses à la tâche de gestion de l'ascenseur au ralenti
- (4%) pour les résultats de simulation fournissant des réponses dans le contexte de deux et trois ascenseurs
- (1%) pour un manuel d'utilisation clair et concis (au plus 1 page) qui décrit comment compiler, configurer et utiliser votre programme de simulation. Le cas échéant, précisez ce qui fonctionne et ce qui ne fonctionne pas.

Date de remise

Le **lundi 4 mai 2020** via le Moodle du cours.

Ce travail peut être réalisé en équipe de deux étudiants.

Conseils

- Réfléchissez bien aux éléments d'états importants dans votre système. Assurez-vous de disposer de structures de données et/ou de variables appropriées pour représenter le ou les états dont vous avez besoin.
- Concevez, déboguez et testez votre simulation avec un seul ascenseur ($N = 1$) au départ. Plus tard, vous pourriez être capable de l'étendre à N ascenseurs, peut-être en changeant les variables d'état scalaires (par exemple, l'occupation, l'étage, la direction) en variables de type tableaux indexés ou autre.
- Pour le débogage initial, il peut être utile de paramétrer votre modèle pour avoir seulement $F = 2$ ou $F = 3$ étages, plutôt que $F = 7$. Juste une suggestion.
- Pour le débogage, n'hésitez pas à varier λ , de sorte que vous puissiez tester les scénarios de charge légère, de charge moyenne et de charge lourde. Vos dernières exécutions devraient utiliser un niveau de charge raisonnable (ou une gamme de niveaux de charge) qui fournit des résultats intéressants.
- Les variantes multi-serveurs du modèle d'ascenseur de base sont difficiles. Assurez-vous que le modèle à un seul ascenseur fonctionne bien et qu'il est bien compris avant de vous attaquer à plusieurs ascenseurs. Bonne chance!