Projet de simulation: ferme de serveurs

2024-2025

- *Ce projet doit être réalisé en binôme (demander l'accord de l'enseignant si vous voulez le réaliser seul).*
- *Le langage de programmation utilisé est Python.*
- *Vous devrez déposer sur e-campus un fichier tar ou zip contenant votre programme ainsi qu'un rapport au format PDF (détails à la fin du document).*
- *La date limite de dépôt sera communiquée ultérieurement.*
- *Une brève soutenance sera organisée à l'issue du rendu du projet.*

Système étudié et modélisation

Nous considérons une ferme de serveurs composée de 12 serveurs identiques. Cette ferme est capable de traiter des requêtes. Pour optimiser le temps de réponse, il peut être avantageux de spécialiser les serveurs. Un serveur spécialisé ne traite qu'une catégorie spécifique de requêtes, mais le fait plus rapidement. En regroupant les 12 serveurs en C groupes de K serveurs ($12 = K \times C$), chaque groupe peut se spécialiser dans une catégorie de requêtes. Cette spécialisation nécessite un routeur en amont des serveurs, qui reçoit toutes les requêtes, détermine leur catégorie, et les dirige vers le groupe de serveurs spécialisé correspondant. Le paramètre C prendra des valeurs dans l'ensemble $\{1, 2, 3, 6\}$.

- Les durées inter-arrivées des requêtes suivent une loi exponentielle de paramètre λ et la catégorie est tirée selon la loi uniforme.
- Le routeur a une file d'attente de capacité finie : il peut stocker jusqu'à 100 requêtes (y compris celle en cours de traitement). Si une requête arrive alors que la file est pleine, celle-ci est perdu. On accepte jusqu'à 5% de taux de perte sur les requêtes.
- Le routeur traite les requêtes en ordre FIFO (premier arrivé, premier servi).
- Le temps de traitement d'une requête par le routeur est constant et vaut $\frac{C-1}{C}$.
- Les serveurs n'ont pas de file d'attente.
- À la sortie du routeur, la requête est instantanément dirigée vers un serveur libre appartenant au groupe de serveurs correspondant au type de requête. S'il n'y en a pas de disponible, la file est bloquée en attente qu'un serveur de la catégorie se libère.
- Le temps de service suit une loi exponentielle de paramètre 4/20 ou 7/20 ou 10/20 ou 14/20 selon que C vaut 1, 2, 3 ou 6 respectivement.

Le but est de trouver le regroupement de serveur, c'est à dire la valeur de C qui minimise le temps de réponse pour différentes valeurs de λ.

Travail demandé

Sur un premier graphique, vous afficherez le temps de réponse moyen pour chaque valeur de C quand le paramètre λ varie (une courbe pour chaque valeur de C, avec λ en abscisse, et le temps de réponse moyen en ordonnée). Vous choisirez les valeurs de λ de manière pertinente. Vous inclurez également les intervalles de confiance à 95% dans le graphique.

Sur un deuxième graphique, vous ferez le même travail en étudiant cette fois le taux de perte des requêtes. Vous déterminerez ainsi de la manière la plus fine possible, pour chaque valeur du paramètre C, la valeur limite de λ à partir de laquelle le taux de perte dépasse 5%.

Vous détaillerez ensuite votre méthodologie pour la détermination du choix optimal de C dans le cas où $\lambda = 1$.

Enfin, vous déterminerez le choix optimal pour le paramètre C pour différentes valeurs de λ. L'optimalité du choix sera considérée avec un niveau de confiance 95%.

Ce qu'il faut rendre

Vous devez fournir un dossier nommé `NOM1_NOM2` où `NOM1` et `NOM2` sont les noms des deux membres du binôme. Si vous travaillez seul, le dossier devra s'appeler `NOM`. Ce dossier doit contenir les éléments suivants :

- Votre programme, dans un ou plusieurs fichiers `.py`.
- Un fichier `README` expliquant la structure générale de votre programme et son utilisation.
- Un rapport au format PDF répondant aux questions posées.
- Vous présenterez votre projet lors d'une soutenance (aucun support n'est demandé pour la soutenance).