Paris-Saclay / UVSQ

Master 1 AMIS

RAPPORT DE PROJET

Simulation de ferme de serveurs

Étudiants : Nabil Hamoudi, Nicolas Chaumont
Date : Mai 2025

Année universitaire 2024–2025

1 Résumer du projet

Nous devions réaliser un projet consistant à implémenter un système traitant des requêtes, selon le modèle suivant : Les requêtes arrivent selon une loi exponentielle de paramètre λ, et leur catégorie est choisie de façon uniforme. Le routeur, doté d'une file d'attente limitée à 100 requêtes (y compris celle en cours de traitement), applique une politique FIFO et tolère jusqu'à 5 % de perte. Chaque requête y est traitée en un temps constant $\frac{(C-1)}{C}$ unité de temps, puis est dirigée vers un serveur libre de la catégorie correspondante. Si aucun serveur n'est disponible, la file est bloquée jusqu'à qu'un serveur de sa catégorie se libére. Le temps de service suit une loi exponentielle dont le paramètre dépend de C : $\frac{4}{20}$, $\frac{7}{20}$, $\frac{10}{20}$ ou $\frac{14}{20}$ pour $C = \{1, 2, 3, 6\}$ respectivement.

L'objectif est de déterminer la valeur de C qui minimise le temps de réponse en fonction de λ.

2 Présentation des résultats

Une fois le système programmé, nous avons pu tester différents paramètres. Pour chacun des graphiques présentés, nous avons choisi de lancer 100 simulations pour chaque point afin d'obtenir des résultats moins sensibles à des situations improbables, tout en ayant un programme suffisamment rapide. Dans la même logique, nous avons choisi un pas de 0.05 pour le λ afin de maximiser la précision des tests sans trop nuire aux performances de notre code. Nous avons également démarré nos simulations à partir de $\lambda = 0.05$, car nous avons estimé qu'observer les performances du système lorsqu'il n'est soumis à quasiment aucune charge n'est pas pertinent.

2.1 Temps de réponse moyen

Tout d'abord, nous avons mesuré l'impact du taux d'arrivée λ sur le temps de réponse moyen.

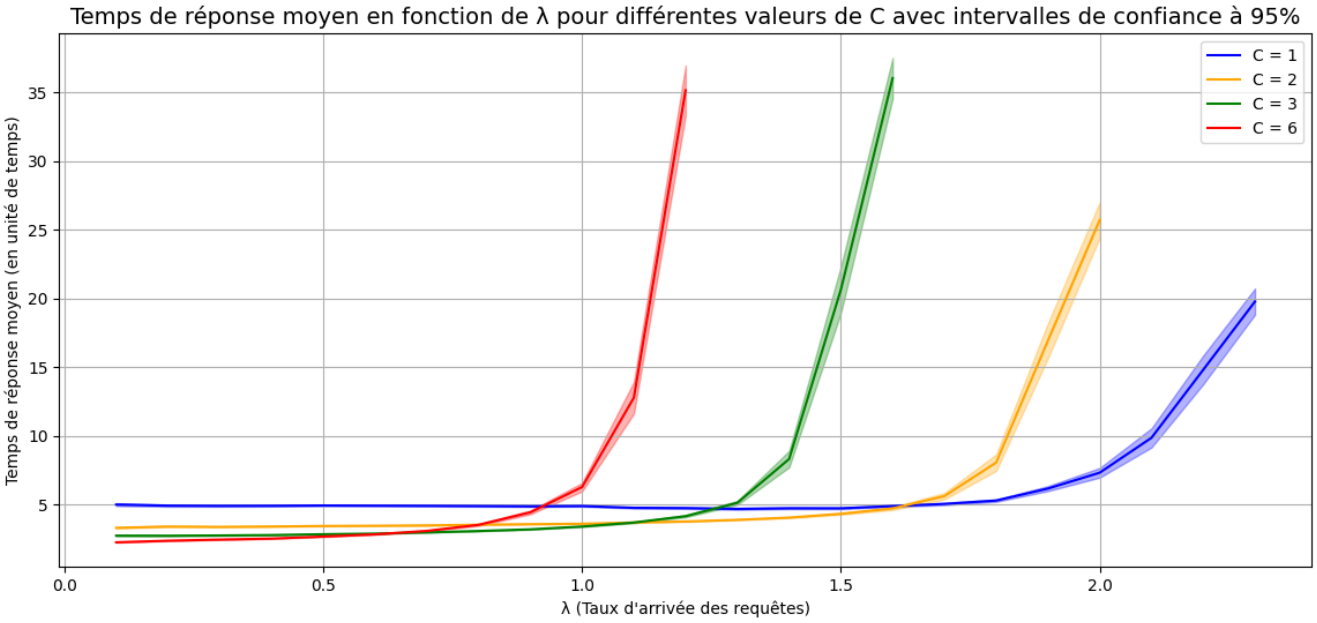

FIGURE 1 – Graphe du temps de réponse moyen en fonction de λ pour chaque C

Sur ce graphique, nous observons que pour $\lambda < 0.5$, la solution $C = 6$ offre aux clients un temps d'attente (2 unité de temps). Cette solution est meilleur que la solution $C = 1$ et $C = 2$ (5 et 3.5 unité de temps respectivement). Elle est aussi marginalement meilleur par rapport aux solutions $C = 3$ (2,25 unité de temps).

À partir de $\lambda = 0.5$, il n'est plus possible de déterminer avec une confiance à 95% laquelle des solutions $C = 6$ ou $C = 3$ est la meilleure. De $\lambda = [0.8, 1[$, la solution $C = 3$ devient la meilleure, tandis que la solution $C = 6$ commence à décrocher en termes de performances, devenant la pire peu avant $\lambda = 1$. Pour cette valeur de λ, les intervalles de confiance pour $C = 3$ et $C = 2$ se chevauchent. Lorsque λ atteint 1.12, les performances de $C = 3$ se sont suffisamment détériorées pour que $C = 2$ prenne le relais, jusqu'à $\lambda = 1.6$, où ses performances deviennent trop proches de celles de $C = 1$ (5 unité de temps) pour les départager. Au-delà de $\lambda = 1.65$, la solution $C = 1$ devient la meilleure.
Pour chacune des quatre courbes, on remarque un phénomène d'effondrement une fois qu'un certain seuil est atteint. Chacune reste à peu près constante, puis s'envole. Cet effondrement s'accompagne d'un élargissement de l'intervalle de confiance.

2.2 Taux de rejet

Sur un second graphique, nous pouvons observer l'impact de λ sur le taux de rejet.

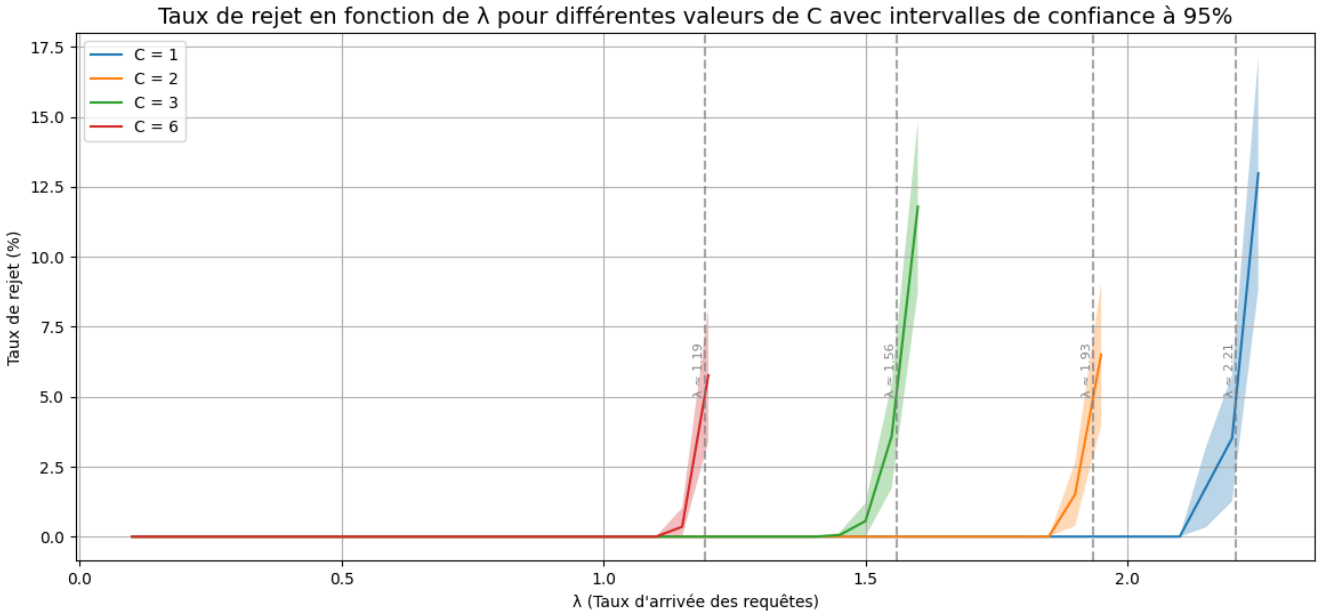

FIGURE 2 – Graphe du taux de rejet en fonction de λ pour chaque C

Ici, nous remarquons des similitudes avec le graphique précédent, à savoir que les solutions s'effondrent dans le même ordre, et que chacune des courbes reste constante avant de s'envoler. On observe néanmoins une différence : la phase d'augmentation du taux de rejet commence un peu après celle de l'augmentation du temps d'attente sur le graphique précédent.
En effet, les limites du taux d'arrivée pour : $C = \{6, 3, 2, 1\}$ sont respectivement $\lambda = \{1.20, 1.56, 1.95, 2.18\}$. Cela s'explique par le fait que le système commence par accumuler jusqu'à 100 clients dans la file du routeur, avant d'être forcé de rejeter un premier client. On peut en déduire que, lorsque le système commence à accumuler trop de clients, les temps d'attente augmentent drastiquement, puis le système finit par s'effondrer.

2.3 Bilan

Considérant toutes ces informations, on peut lister les valeurs optimales de C selon la valeur de λ :
• Pour $0.05 \leq \lambda \leq 0.7$: il faut choisir $C = 6$
• Pour $0.7 < \lambda \leq 1.1$: il faut choisir $C = 3$
• Pour $\lambda = 1$: bien que les intervalles de confiance pour $C = 3$ et $C = 2$ commencent à se chevaucher, la solution $C = 3$ semble légèrement meilleure, mais avec un taux de confiance inférieur à 95 %
• Pour $1.1 < \lambda \leq 1.65$: la valeur optimale de C est 2

• Au-delà de $\lambda = 1.65$: la valeur optimale de C est 1
• Enfin, à partir de $\lambda = 2.18$: aucune solution ne convient, car la contrainte d'un taux de rejet inférieur à 5 % est violée

3 Conclusion

Pour conclure, on peut dire que cette étude nous a permis d'identifier des plages optimales pour la valeur du paramètre C en fonction de la charge λ, afin de minimiser le temps de réponse tout en respectant les contraintes de perte. On observe que des valeurs de C plus élevées (notamment $C = 6$) sont très efficaces sous faible charge, tandis que des valeurs plus faibles deviennent préférables à mesure que la pression exercée par le taux d'arrivée augmente. Cependant, au-delà de ces résultats numériques, notre projet soulève des réflexions plus larges sur deux paramètres clés du système : le taux d'occupation des serveurs et la capacité de la file d'attente.

En effet, un serveur peu sollicité maximise la réactivité mais sous-exploite les ressources, tandis qu'un serveur saturé peut engendrer des délais massifs voire des pertes. Trouver un juste équilibre entre performance et efficacité énergétique (ou coût d'infrastructure) devient alors un enjeu important, notamment dans un contexte de déploiement réel. De même, la taille de la file du routeur joue un rôle tampon crucial : une capacité plus grande retarde le rejet des requêtes mais prolonge potentiellement les temps d'attente et augmente la volatilité du système en phase critique.

Une piste d'approfondissement serait donc d'analyser plus finement l'impact du taux d'occupation moyen des serveurs sur les métriques de performance, en parallèle de tests avec différentes tailles de file. Cela permettrait d'évaluer si l'architecture du système peut être optimisée non seulement par le choix de C, mais aussi par une gestion dynamique de la capacité d'accueil et des politiques de répartition des requêtes.