

Analyse de la moulinette sous l'angle des systèmes d'attente

*Lucas Collemare, Gabriel Cellier, Robin de Bastos
Thomas Polo, Mathys Popoff-Morin, Erwin Rodrigues
Nom du groupe : groupe*

12 janvier 2026

1 Contexte et objectifs

Ce document propose une modélisation et une analyse de l'infrastructure de correction automatique (moulinette) sous l'angle des systèmes d'attente. Une moulinette exécute des suites de tests unitaires sur du code soumis par des étudiants (push tag) puis renvoie le résultat via une interface front : les tags génèrent des jobs, les ressources de calcul et d'envoi jouent le rôle de serveurs, et la latence perçue correspond au temps de séjour.

Nous étudions deux familles de scénarios :

- Un modèle Waterfall en deux étages (exécution puis envoi), décliné avec files infinies ou finies et un mécanisme de back-up
- Un modèle Channels & Dams où deux populations (ING et PREPA) suivent des dynamiques différentes, avec régulation périodique de la population ING.

Pour chaque cas, le travail demandé comprend la modélisation, la simulation et l'analyse (paramètres, conditions de stabilité), le calcul de métriques (temps de séjour, taux de blocage, etc.), ainsi qu'une synthèse argumentée accompagnée des résultats bruts.

2 Hypothèses et modélisation

Notation	Signification
λ	taux d'arrivée des tags (jobs)
μ_s	taux de service d'exécution (stage 1), avec K serveurs en parallèle
μ_f	taux de service d'envoi des résultats (stage 2), avec un serveur unique
k_s	capacité finie du premier étage (exécution)
k_f	capacité finie du second étage (envoi front)

Le workflow nominal est modélisé comme un système à deux étages en série :

- Exécution : K serveurs, FIFO, capacité infinie ou finie.
- Envoi : 1 serveur, FIFO, capacité infinie ou finie.

Files infinies : modélisation $\rightarrow M/M/K$ et $M/M/1$ | condition de stabilité $\rightarrow \lambda < K\mu_s$ et $\lambda < \mu_f$.

Capacités finies : le système est borné, mais peut rejeter des jobs lorsque les capacités k_s ou k_f sont atteintes.

Pour éviter des résultats perdu lorsquie la file d'envoi est pleine, on introduit un back-up : tout résultat refusé à l'étage 2 est stocké, puis réinséré dès qu'une place se libère. Cela supprime la perte de résultat (en théorie), au prix d'un risque d'accumulation (stockage, retard).

On considère deux populations :

- ING : arrivées fréquentes, jobs courts
- PREPA : arrivées plus rares, jobs plus longs.

Le but est d'évaluer l'impact de cette hétérogénéité sur les métriques (temps de séjour, blocage, débit) et de comparer une régulation de type "barrage" : blocage de ING pendant t_b , puis ouverture pendant $t_b/2$, etc.

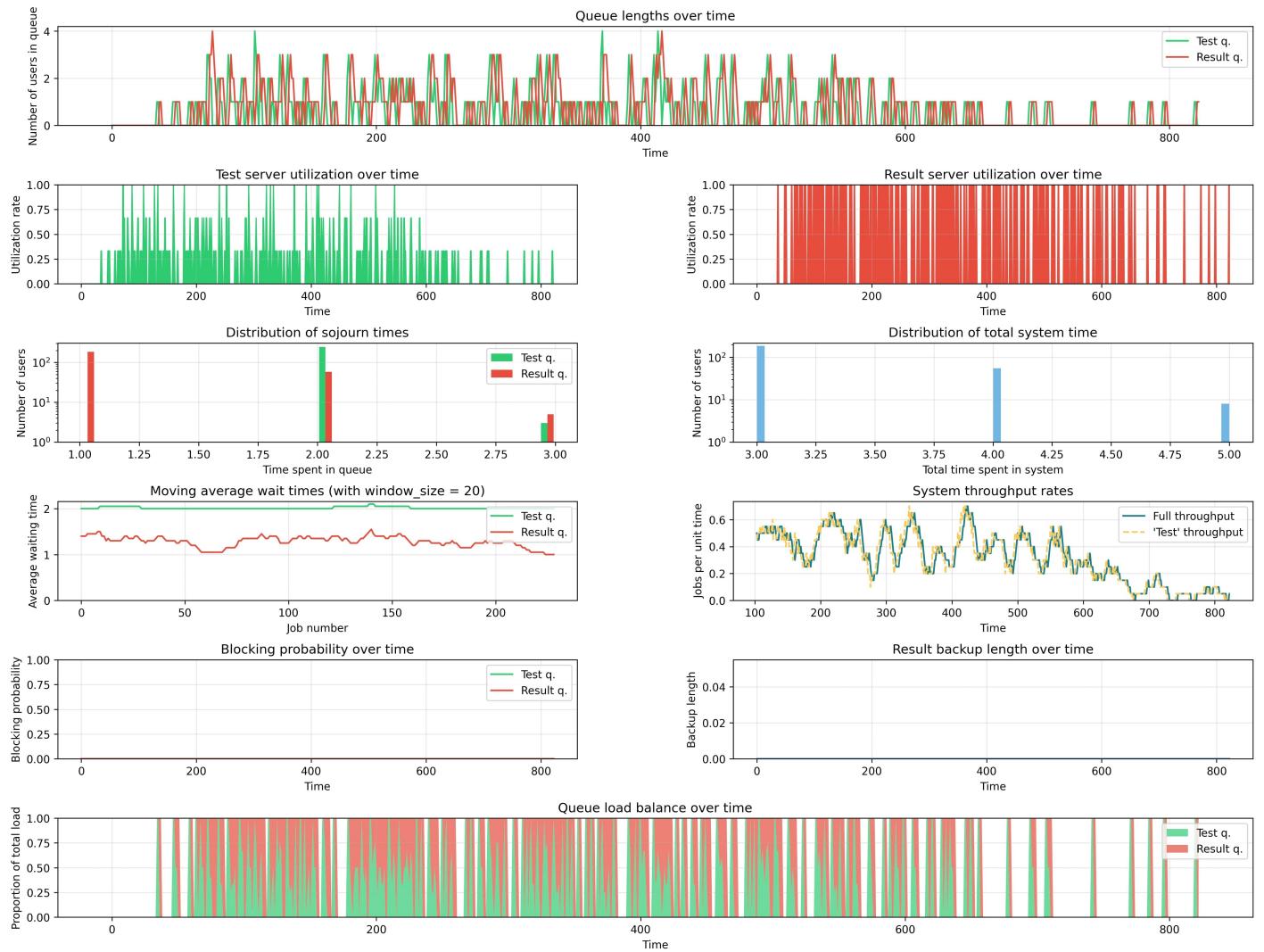
3 Méthode de simulation

Les simulations sont implémentées en Python avec SimPy. Chaque étudiant réalise une séquence de soumissions (nb_exos) avec des temps d'attente entre tags ; les durées de service sont paramétrées (exécution et envoi). Les métriques collectées sont la longueurs de file et utilisation serveur (moyenne, variance, max), le taux de blocage (blocking rate) par étage, le temps de séjour moyen et variance (par étage et total) et débit (throughput).

4 Étude de cas 1 : Waterfall

4.1 Files infinies

Configuration simulée : $K = 3$, temps de service exécution = 2, envoi = 1, avec 30 utilisateurs.



Stabilité Dans ce modèle, une surcharge (à λ fixé) se traduit par une croissance des files et une dégradation forte des temps de séjour. En pratique, le goulot d'étranglement peut être l'étage 1 (si $K\mu_s$ est trop faible) ou l'étage 2 (si μ_f est trop faible).

Temps de séjour empirique Sur cette configuration, on mesure :

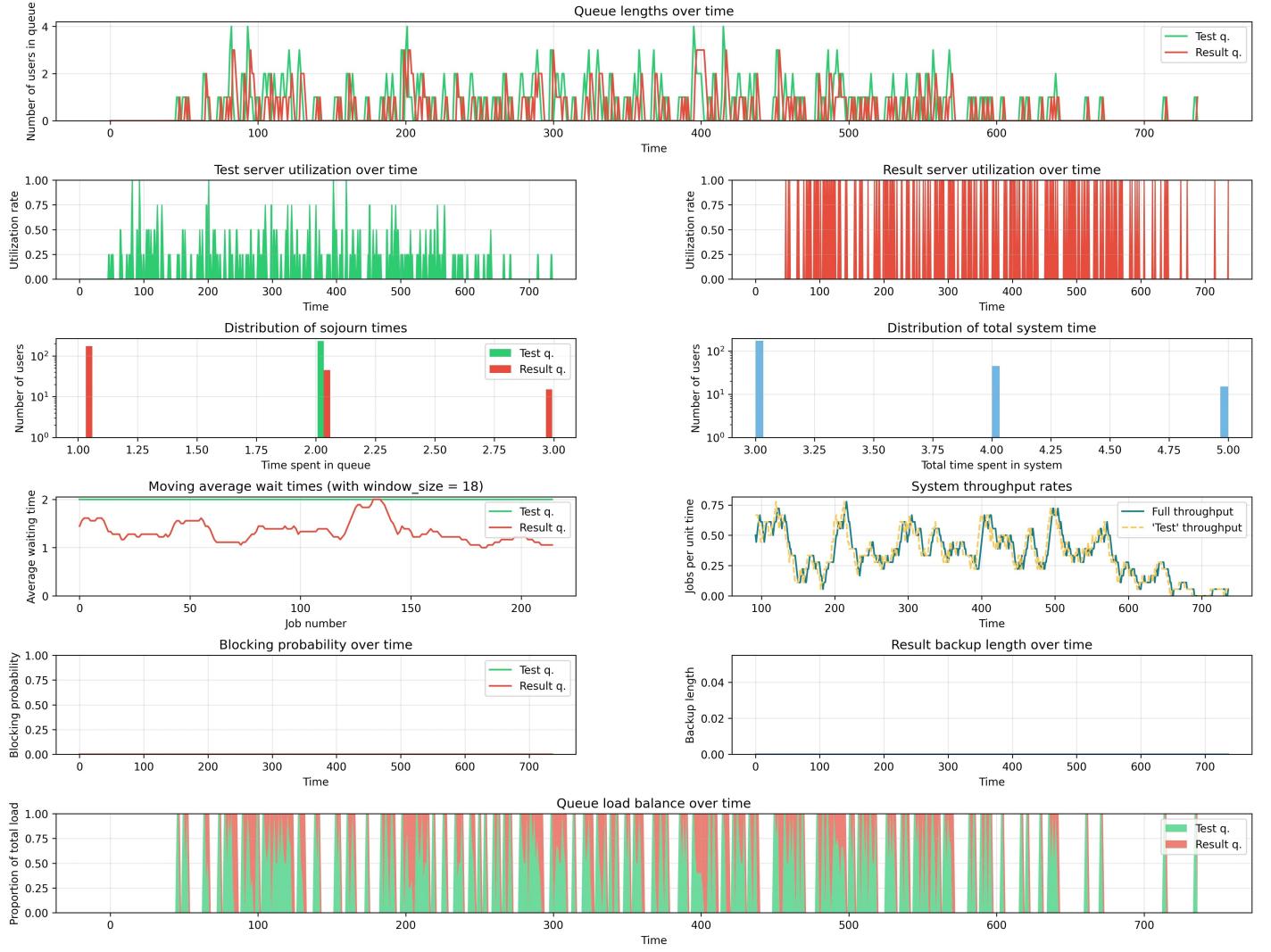
$$\mathbb{E}[T] \approx 3.286 \quad , \quad \text{Var}(T) \approx 0.269$$

4.2 Files finies : proportions de refus

Avec capacités finies, on observe deux types de refus :

- Refus à l'étage 1 : le tag est refusé si la capacité k_s est atteinte
- Refus à l'étage 2 : le résultat est perdu si k_f est atteint (page blanche).

Configuration simulée : $K = 4$, $k_s = 20$, $k_f = 10$, temps de service exécution = 2, envoi = 1.



Métriques observées Sur cette configuration, les taux de blocage observés sont nuls (`blocking_rate = 0.0` sur les deux étages), et le temps de séjour total empirique est :

$$\mathbb{E}[T] \approx 3.323 , \quad \text{Var}(T) \approx 0.348$$

Discussion paramétrique diminuer k_s augmente la proportion de tags refusés ; diminuer k_f augmente la proportion de pages blanches ; augmenter K réduit la congestion à l'étage 1, mais peut déplacer la congestion vers l'étage 2.

4.3 Back-up : pages blanches et risques

Configuration simulée : $K = 4$, $k_s = 20$, $k_f = 5$ (envoi plus constraint), back-up activé.



Effet sur les pages blanches Le back-up permet de sauvegarder puis renvoyer ultérieurement les résultats refusés à l'étage 2. Ainsi, la proportion de pages blanches tend vers 0 tant que le back-up n'est pas saturé et que les résultats sont effectivement réémis.

Risques et effets de bord Cette solution peut engendrer : une accumulation en back-up et consommation mémoire/stockage, une augmentation de la latence, un réordonnancement des retours et gestion des doublons, un besoin de politiques de purge, de reprise et d'idempotence.

Back-up aléatoire vs systématique Un back-up systématique minimise la perte d'information, mais coûte cher à pleine charge. Un back-up aléatoire permet de contrôler le coût : le taux de pages blanches devient alors approximativement $(1 - p) B_f$ où B_f est la probabilité de blocage à l'étage 2.

Temps de séjour empirique Dans la configuration back-up, on observe :

$$\mathbb{E}[T] \approx 6.166 \quad , \quad \text{Var}(T) \approx 40.811$$

La variance est nettement plus élevée, ce qui est cohérent avec des attentes prolongées lors de congestion et de vidage progressif du back-up.

5 Étude de cas 2 : Channels & Dams

5.1 Sans régulation (baseline)

Configuration simulée : $K = 3$, $k_s = 15$, $k_f = 8$ et deux populations (ING plus fréquente, PREPA avec exécution plus longue).

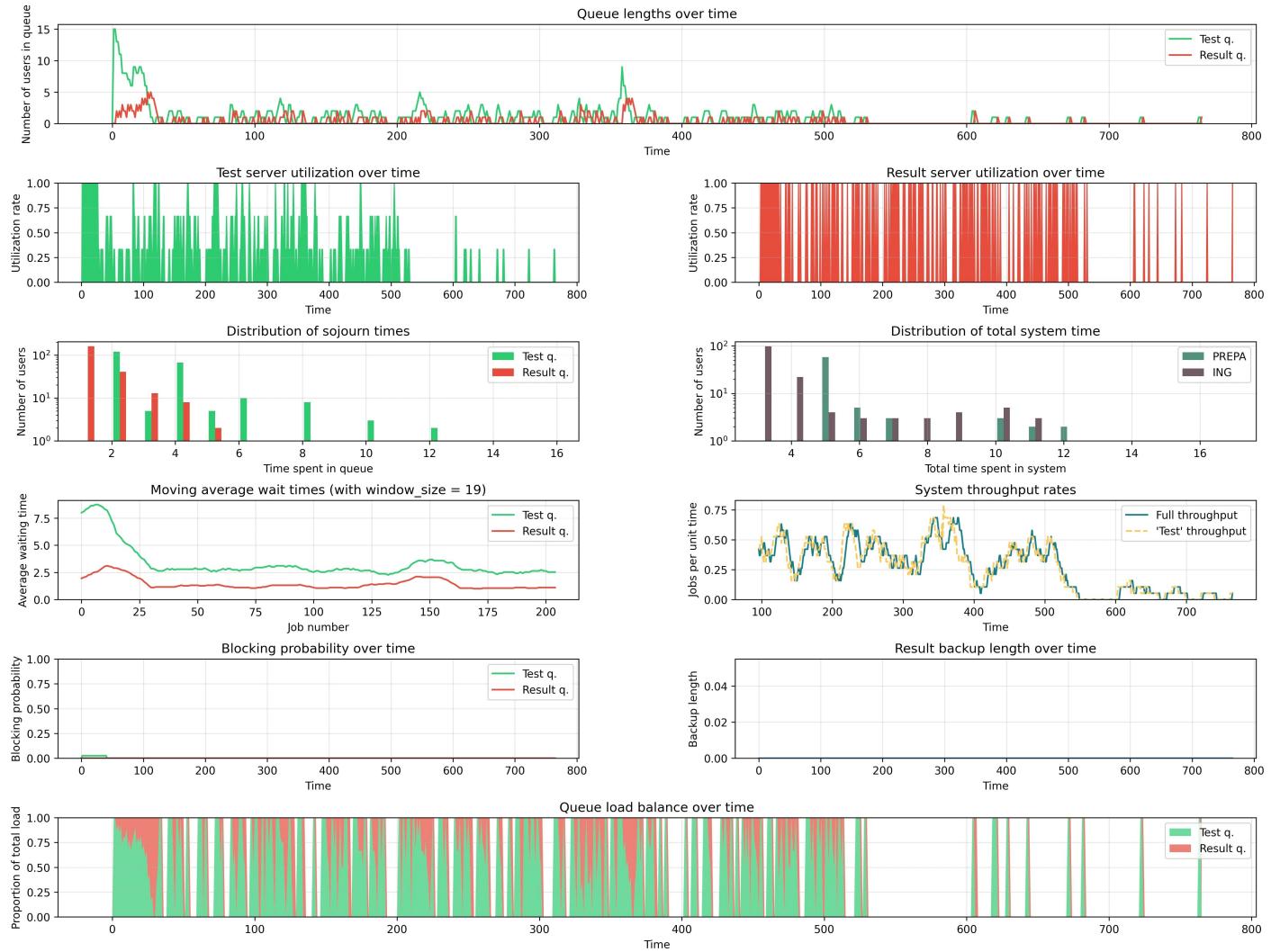


Figure 1: Channels & Dams sans régulation effective.

Métrique	Moyenne	Variance
Temps de séjour total T	4.865	7.022
Temps de séjour étage test T_s	3.422	4.809
Temps de séjour étage envoi T_f	1.444	0.686
Taux de blocage étage test	0.0673	—
Taux de blocage étage envoi	0.0	—
Débit (throughput)	0.2915	—

Table 1: Métriques globales observées sans régulation.

5.2 Avec barrage sur ING

Le barrage est caractérisé par un cycle de blocage (t_b) puis ouverture ($t_b/2$).

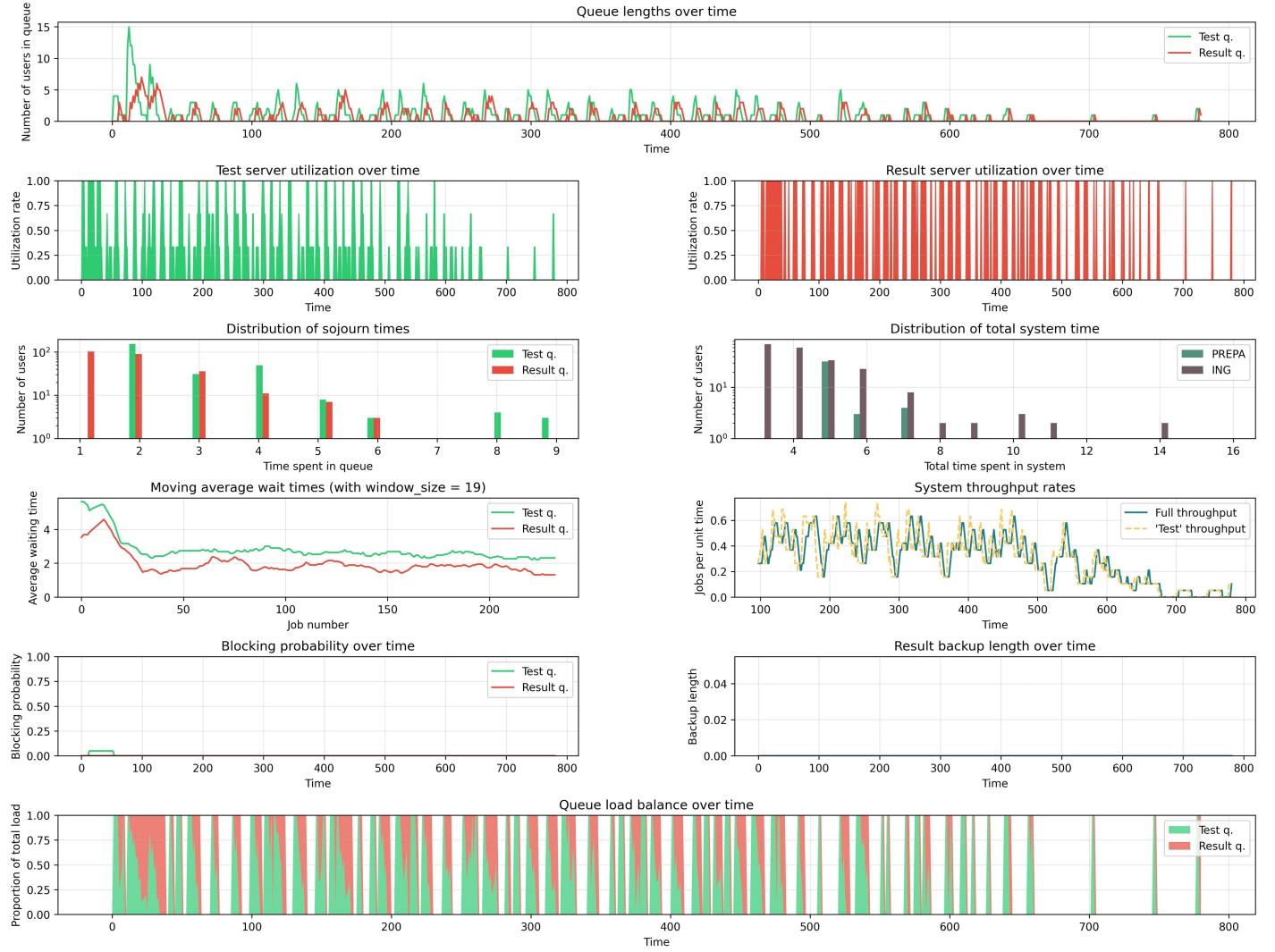


Figure 2: Channels & Dams avec régulation par barrage.

Métrique	Moyenne	Variance
Temps de séjour total T	4.809	4.473
Temps de séjour étage test T_s	2.837	1.842
Temps de séjour étage envoi T_f	1.972	1.270
Taux de blocage étage test	0.0438	—
Taux de blocage étage envoi	0.0	—
Débit (throughput)	0.3218	—

Table 2: Métriques globales observées avec barrage.

5.3 Comparaison et recommandations

	Sans régulation	Avec barrage
Blocking rate (test)	0.0673	0.0438
$\mathbb{E}[T]$	4.865	4.809
$\text{Var}(T)$	7.022	4.473
Throughput	0.2915	0.3218

Table 3: Comparaison globale

Le barrage réduit la variance du temps de séjour et diminue le taux de blocage à l'étage test. Le gain sur la moyenne est plus modeste, mais la diminution de variance peut fortement améliorer l'expérience utilisateur. Le barrage améliore aussi la régularité côté test (baisse de T_s et de sa variance), mais peut augmenter le temps moyen à l'étage envoi (T_f) si davantage de résultats arrivent en rafales lors des phases d'ouverture.

Alternative pour minimiser le temps de séjour des deux populations Une approche classique est d'introduire une file à priorité (par exemple PREPA prioritaire) ou des ressources dédiées afin d'assurer de l'équité sans bloquer par fenêtres temporelles.

6 Validation par modèles théoriques

En complément, nous comparons la simulation à des résultats théoriques de base ($M/M/1$, $M/M/k$, $M/G/1$ avec service déterministe), afin de valider la cohérence globale du moteur de simulation.

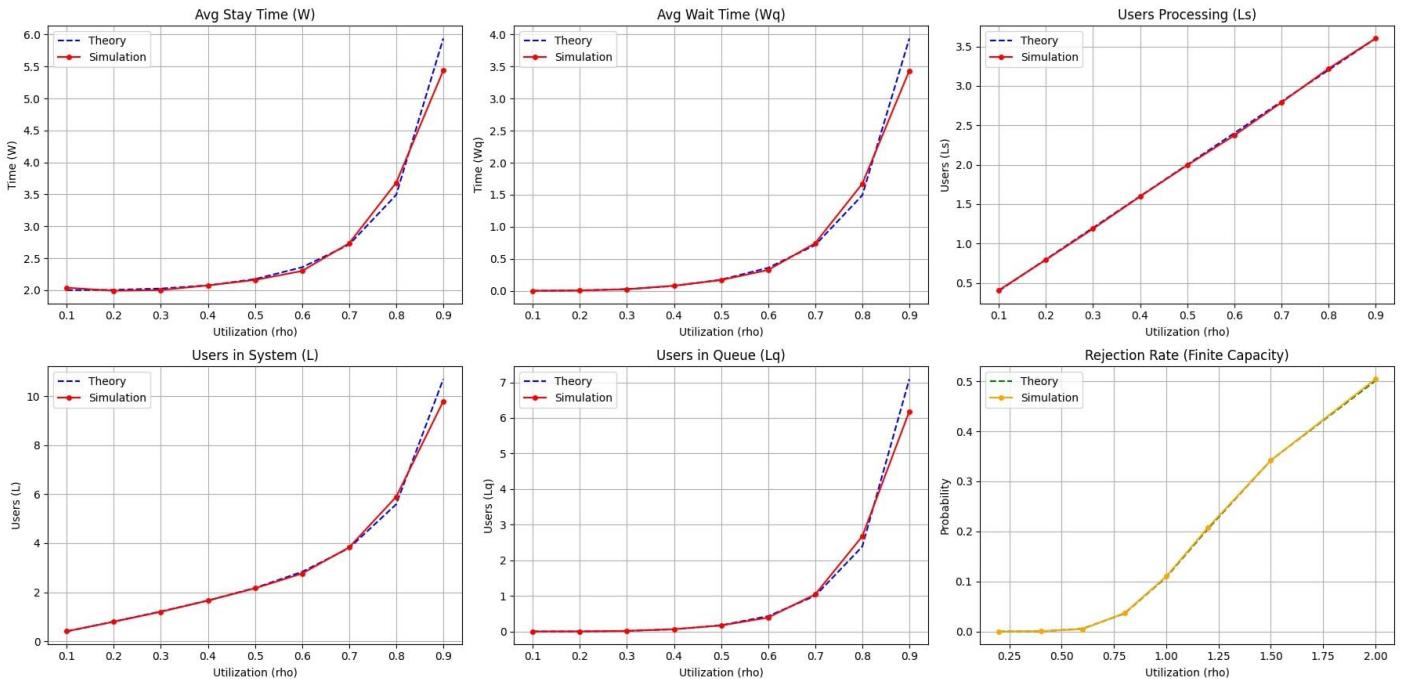


Figure 3: Comparaison théorie vs simulation sur des modèles de référence.

7 Analyse des coûts

En plus des métriques de performance (temps de séjour, taux de blocage, débit), nous proposons une lecture économique du dimensionnement et du choix d'architecture : un coût d'infrastructure (serveurs), un coût de qualité de service (rejets, attente excessive, pertes) et un coût opérationnel (bande passante, stockage).

7.1 Modèle de coût et paramètres

Le coût total est décomposé sous la forme :

$$C_{\text{total}} = C_{\text{infra}} + C_{\text{qualité}} + C_{\text{op}}$$

Infrastructure Dans les scripts, l'infrastructure est modélisée comme un coût horaire sur une durée de simulation fixée (ici 1h) :

$$C_{\text{infra}} = D \cdot (K \cdot c_{\text{test}} + 1 \cdot c_{\text{résultat}})$$

où K est le nombre de serveurs de test, et un serveur unique est utilisé pour l'envoi des résultats.

Qualité de service À partir des métriques issues de la simulation (taux de blocage par étage, temps de séjour moyen par étage), on valorise : la proportion de requêtes rejetées, l'attente au-delà d'un seuil acceptable (ici 5 minutes), et la perte de résultat (rejet à l'étage envoi lorsque le résultat est perdu).

Opérationnel Le coût opérationnel prend en compte la bande passante proportionnelle au nombre de résultats renvoyés avec succès et à leur taille moyenne, et optionnellement un stockage de back-up.

Les paramètres utilisés (profil AWS small) sont :

- $c_{\text{test}} = 0.04 \text{ EUR/h}$
- $c_{\text{résultat}} = 0.02 \text{ EUR/h}$
- Taille moyenne d'un résultat = 0.5MB, bande passante = 0.09 EUR/GB
- Coût par rejet = 0.10 EUR
- Coût par minute d'attente excessive = 0.02 EUR
- Coût par résultat perdu = 0.15 EUR.

7.2 Comparaison des architectures à configuration fixée

On compare plusieurs architectures dans un cadre homogène (mêmes coûts unitaires et même charge) : Waterfall files infinies, Waterfall files finies, Waterfall avec back-up, Channels & Dams régulé et non régulé. Les configurations utilisées sont $K = 4$ pour Waterfall, et $K = 3$ pour Channels & Dams.

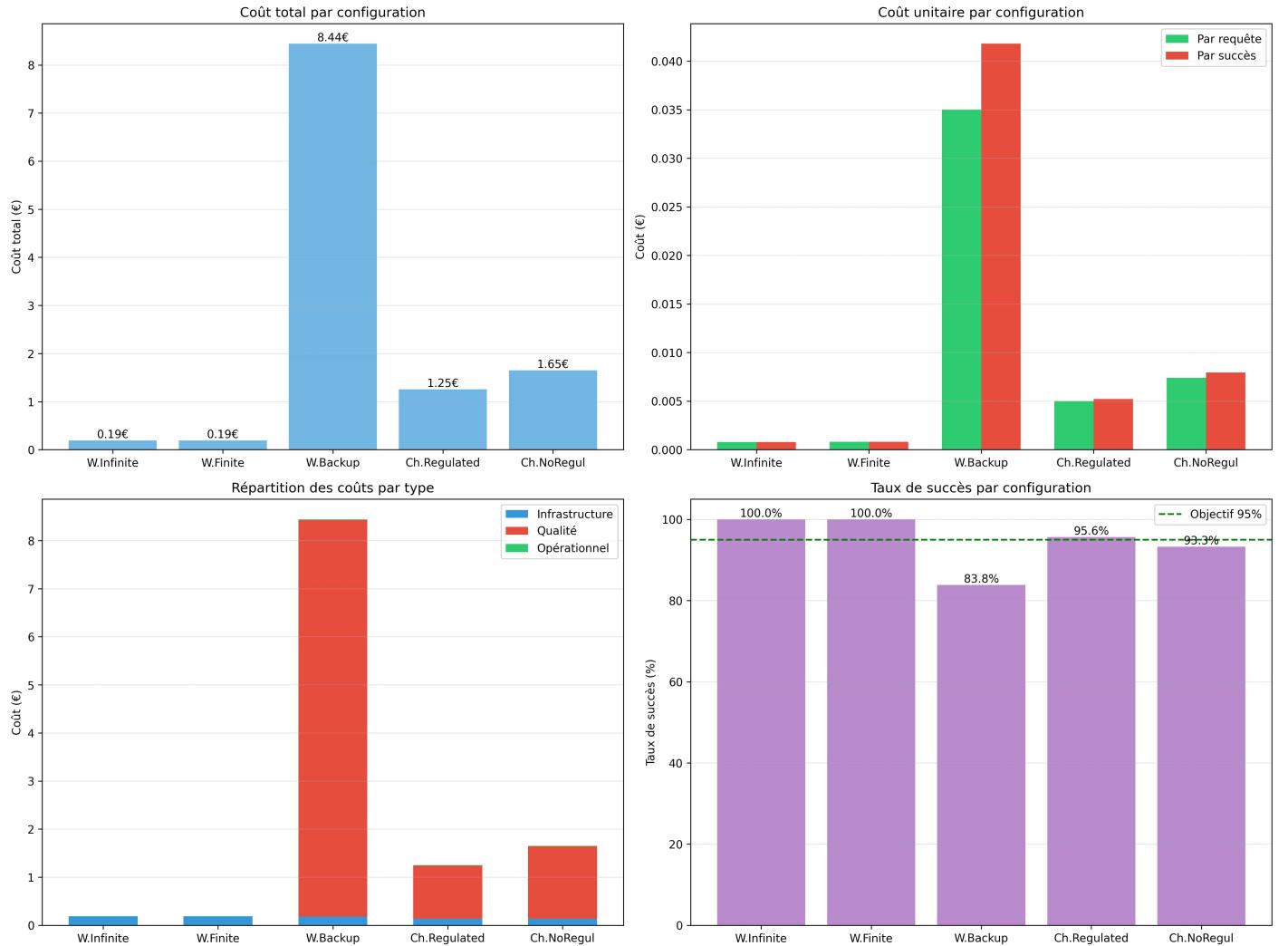


Figure 4: Comparaison des coûts (toutes architectures).

Architecture	Coût total (EUR)	Succès (%)	Coût / succès (EUR)
W.Infinite ($K = 4$)	0.19	100.0	0.0008
W.Finite ($K = 4$)	0.19	100.0	0.0008
W.Backup ($K = 4$)	8.44	83.8	0.0418
Ch.Regulated ($K = 3$)	1.25	95.6	0.0052
Ch.NoRegul ($K = 3$)	1.65	93.3	0.0079

Table 5: Synthèse numérique

Lecture Sous cette charge, Waterfall (infini ou fini) atteint un taux de succès de 100% avec un coût faible. Le back-up, ici, dégrade fortement le coût par succès : le taux de rejet plus élevé combiné à l'augmentation des temps de séjours se traduit en coût de qualité important, ce qui illustre que le back-up n'est pertinent que si l'on valorise fortement l'élimination des pertes et que l'on contrôle son effet sur la latence.

7.3 Passage à l'échelle et comparaison multi-architectures

Enfin, on fait varier K pour chaque architecture (mêmes valeurs de K que précédemment) et on observe l'évolution du coût total, du taux de succès, du coût par succès et de l'efficacité (requêtes réussies par euro).

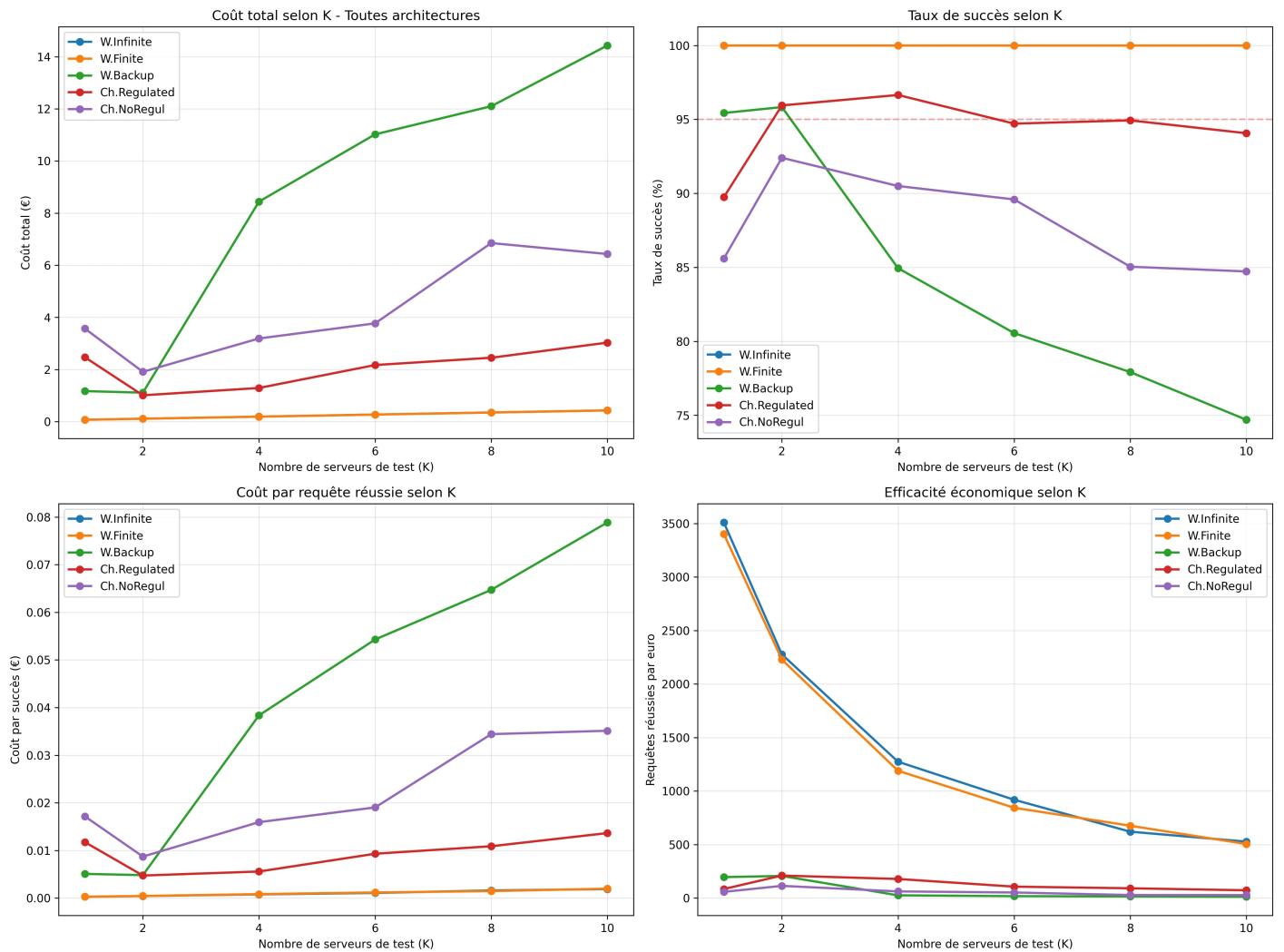


Figure 5: Comparaison multi-architecture : coût et efficacité selon K .

7.4 Dimensionnement de K (Waterfall fini)

Les méthodes qui proposent le meilleur coût sont waterfall fini et infini. Il faut donc déterminer le cas optimal. On étudie alors l'impact du nombre de serveur $K \in \{1, 2, 4, 6, 8, 10\}$ sur le coût pour l'architecture Waterfall à capacités finies ($k_s = 20, k_f = 10$), avec 30 utilisateurs. Les scripts génèrent deux figures : une comparaison agrégée des coûts et une analyse d'échelle.

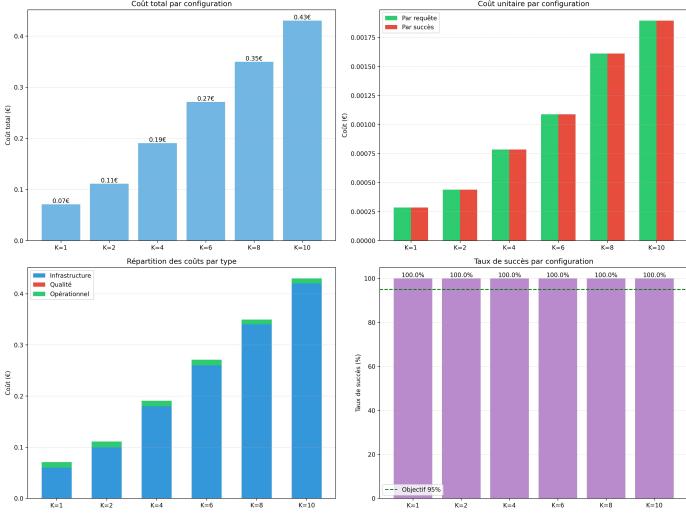


Figure 6: comparaison des configurations K .

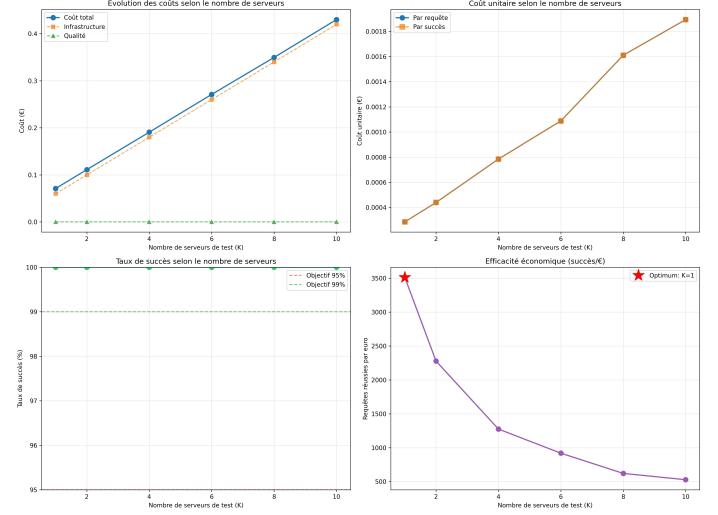


Figure 7: Évolution selon K .

Lecture Dans le régime simulé, le taux de succès est de 100% pour tous les K testés ; le coût total est alors majoritairement porté par C_{infra} et croît avec K . Dans ce cas, minimiser le coût conduit mécaniquement à un K minimal, mais un choix réaliste doit intégrer les différentes contraintes évoquées plus haut et ne peut pas être guidé uniquement par la minimisation du coût.

8 Conclusion

Le modèle Waterfall met en évidence un compromis entre dimensionnement (serveurs K et capacités k_s, k_f), qualité de service (temps de séjour) et robustesse (taux de blocage). Le back-up est efficace contre les pages blanches, mais peut augmenter fortement la variance des temps de séjour en charge. Le modèle waterfall fini et infini présente le meilleur ratio succès/coût. Avec un nombre de serveur minimal il permet d'atteindre le coût minimal sur tous les scénarios étudiés.

Dans le scénario Channels & Dams, la coexistence de populations hétérogènes peut dégrader l'équité. Un barrage réduit certains effets pathologiques, mais des solutions plus fines (priorités, quotas, files séparées) sont souvent préférables pour minimiser les temps de séjour de chaque population tout en contrôlant le risque côté expérience utilisateur, cela s'accompagne néanmoins par des hausses de coûts.