

# ERO2 : Simulation Moulinette

Promotion : ING3 2026

## Groupe Tu-tu-tu-du

Cédric Brzyski  
Tom Coulliaud-Maisonneuve  
Matthias Laithier  
Thomas Maurer  
Lucas Tilly  
Aziz Zeghal



Repository : ERO2

# Introduction

Dans le cadre de ce projet ERO2, nous avons développé un simulateur interactif d'infrastructure de moulinettage basé sur la théorie des files d'attente. L'objectif principal était de créer un outil permettant d'analyser en temps réel le comportement de systèmes d'attente complexes et de répondre aux questions posées dans le cadre de la SAÉ.

Ce projet se découpe en trois axes principaux :

## **1. Développement d'un simulateur web interactif**

Nous avons implémenté une application React avec un moteur de simulation basé sur le modèle M/M/K. L'interface permet de configurer dynamiquement les paramètres du système (nombre de serveurs, taux d'arrivée, capacités de files, probabilité de backup) et de visualiser en temps réel l'évolution des files d'attente, l'utilisation des serveurs, et les métriques de performance. La simulation utilise des événements discrets et des lois exponentielles pour modéliser fidèlement le comportement théorique.

## **2. Implémentation de deux scénarios d'analyse**

Le simulateur intègre deux modèles distincts : le modèle Waterfall (cascade) avec une population homogène, permettant d'analyser l'impact des files finies et des mécanismes de backup , et le modèle Channels & Dams avec deux populations (ING et PREPA) aux comportements différents, permettant d'étudier les stratégies d'ordonnancement (FIFO, SJF, priorités) et les mécanismes de régulation du flux (Dam).

## **3. Analyse quantitative et validation théorique**

Pour chaque scénario, nous avons conduit des séries de simulations permettant d'extraire des métriques clés (temps de séjour moyen, variance, taux de rejet, utilisation des ressources) et de valider empiriquement les prédictions du modèle M/M/K, notamment la condition de stabilité ( $\rho < 1$ ) et la Loi de Little ( $L = \lambda \cdot W$ ).

Ce rapport présente notre démarche d'implémentation, nos résultats de simulation, et nos recommandations pour le dimensionnement et l'optimisation de ce type de système.

# Contenu

<b>1 Architecture du Simulateur</b>	<b>1</b>
1.1 Vue d'ensemble de l'implémentation . . . . .	1
1.2 Moteur de Simulation . . . . .	1
1.3 Modèle Waterfall : Implémentation . . . . .	2
1.4 Modèle Channels & Dams : Implémentation . . . . .	2
1.5 Collecte et Enregistrement des Métriques . . . . .	3
<b>2 Fondements Théoriques et Validation</b>	<b>4</b>
2.1 Modèle M/M/K Appliqué . . . . .	4
2.2 Condition de Stabilité . . . . .	4
2.3 Loi de Little : Validation des Résultats . . . . .	5
2.4 Métriques Collectées . . . . .	5
<b>3 Résultats : Modèle Waterfall</b>	<b>6</b>
3.1 Configuration des Expériences . . . . .	6
3.2 Résultats avec Files Infinies . . . . .	6
3.3 Impact des Files Finies et Analyse des Rejets . . . . .	7
3.4 Impact du Dimensionnement ( $K_{exec}$ ) . . . . .	7
3.5 Analyse du Mécanisme de Backup . . . . .	8
<b>4 Résultats : Modèle Channels &amp; Dams</b>	<b>9</b>
4.1 Configuration Multi-Classes . . . . .	9
4.2 Comparaison des Stratégies d'Ordonnancement . . . . .	9
4.3 Effet du Mécanisme de Dam . . . . .	10
<b>5 Synthèse et Recommandations Opérationnelles</b>	<b>11</b>
5.1 Synthèse des Résultats . . . . .	11
5.2 Recommandations pour le Dimensionnement . . . . .	11
5.3 Recommandations pour le Scénario Multi-Classes . . . . .	12
5.4 Limites et Perspectives . . . . .	12
<b>Conclusion</b>	<b>13</b>
<b>Annexes</b>	<b>14</b>

## 1. Architecture du Simulateur

### 1.1 Vue d'ensemble de l'implémentation

Nous avons développé un simulateur web permettant une simulation interactive de systèmes d'attente. L'architecture logicielle se décompose en trois composantes principales :

- Moteur de simulation : Logique de simulation à événements discrets
- Interface utilisateur : Configuration, visualisation et analyse
- Visualisation : Animation en temps réel et graphiques dynamiques pour métriques

### 1.2 Moteur de Simulation

#### Principe de fonctionnement

Le moteur implémente une simulation à événements discrets basée sur un pas de temps  $\Delta t$  configurable. À chaque itération, le moteur :

1. Génère les arrivées selon un processus de Poisson (taux  $\lambda$ )
2. Gère les files d'attente (insertion ou rejet si saturation)
3. Assigne les jobs aux serveurs disponibles
4. Simule l'exécution avec des temps de service exponentiels
5. Enregistre les métriques (temps d'attente, rejets, utilisation)

#### Génération des événements stochastiques

Les arrivées suivent un processus de Poisson. Dans un intervalle  $dt$ , la probabilité d'une arrivée est calculée par  $P = 1 - e^{-\lambda \cdot dt}$ .

Les temps de service, eux, suivent une loi exponentielle :  $T_{service} = -\frac{1}{\mu} \ln(U)$  où  $U$  est une variable aléatoire uniforme.

#### Gestion des files d'attente

Le système implémente deux files FIFO avec capacités configurables :

- File d'exécution : Capacité  $k_s$  (infinie ou finie)
- File de résultats : Capacité  $k_f$  (infinie ou finie)

Logique de rejet : Si la longueur de la file est supérieure ou égale à sa capacité, le job est rejeté. Sinon, il est ajouté à la file.

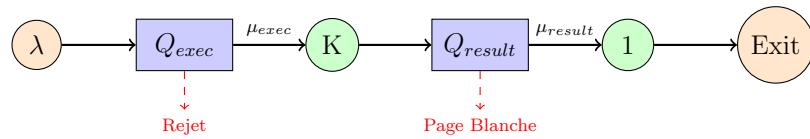
## Mécanisme de backup

Le backup est implémenté avec une probabilité  $p_{backup}$  : lors d'un rejet à la file de résultats, le système effectue un tirage aléatoire. Si le résultat est inférieur à  $p_{backup}$ , le job est sauvegardé, sinon c'est une page blanche.

### 1.3 Modèle Waterfall : Implémentation

## Architecture en cascade

Le modèle Waterfall est représenté par deux stations en série : une station d'exécution ( $M/M/K$  avec  $K$  serveurs parallèles) suivie d'une station de résultat ( $M/M/1$  avec un serveur unique).



## Paramètres configurables

L'interface permet de modifier dynamiquement les paramètres suivants :

Paramètre	Description
Nombre de serveurs (K)	1 à 20
Taux d'arrivée ( $\lambda$ )	0.1 à 10 jobs/s
Temps moyen d'exécution	0.5 à 10 s
Temps moyen de résultat	0.1 à 5 s
Capacité $k_s$	Infinie ou 5-100
Capacité $k_f$	Infinie ou 5-100
Probabilité backup	0% à 100%

TABLE 1.1 – Paramètres configurables du modèle Waterfall

## 1.4 Modèle Channels & Dams : Implémentation

## Architecture multi-classes

Le modèle Channels & Dams implémente deux flux d'arrivées distincts : les étudiants ING (jobs courts, arrivées fréquentes) et les étudiants PREPA (jobs longs, arrivées rares). Chaque arrivée est typée et conserve son type tout au long du traitement.

## Stratégies d'ordonnancement implémentées

Nous avons implémenté 4 stratégies de scheduling :

1. FIFO : Ordre d'arrivée (équitable)
2. SJF (Shortest Job First) : Priorise les jobs à temps court
3. ING First : Priorité absolue aux ING (ici, similaire à SJF)
4. PREPA First : Priorité absolue aux PREPA

Pour SJF, la file d'exécution est triée par temps de service attendu à chaque assignation de serveur.

## Mécanisme de Dam

Le Dam bloque périodiquement l'accès aux serveurs selon un cycle défini, composé d'une phase de blocage, suivie d'une phase d'ouverture. Pendant le blocage, les jobs ING s'accumulent dans la file et les PREPA passent. À l'ouverture, ils sont traités comme d'habitude. Cela permet de réguler le flux des ING et d'offrir des fenêtres plus libres aux PREPA.

## 1.5 Collecte et Enregistrement des Métriques

### Métriques temps réel

Le simulateur collecte en continu :

- Le nombre de jobs complétés, rejetés (exec), rejetés (result)
- Les temps d'attente individuels
- Les longueurs des files à chaque instant
- L'utilisation des serveurs (% de serveurs actifs)
- Les pages blanches et les jobs sauvés par backup

### Calcul de la variance

La variance empirique est calculée en temps réel selon la formule classique :  $\sigma^2 = \frac{1}{n} \sum_{i=1}^n (W_i - \bar{W})^2$  où les  $W_i$  sont les temps d'attente individuels.

### Historique pour graphiques

L'historique temporel est enregistré à intervalles réguliers et contient le temps écoulé, la longueur de chaque file, ainsi que le taux d'utilisation des serveurs. Ces données alimentent les graphiques de l'interface.

## 2. Fondements Théoriques et Validation

### 2.1 Modèle M/M/K Appliqué

Notre implémentation repose sur le modèle M/M/K :

- Premier M : Arrivées poissonniennes (processus de Poisson de taux  $\lambda$ )
- Second M : Temps de service exponentiels (moyenne  $\frac{1}{\mu}$ )
- K : K serveurs en parallèle

#### Génération des événements aléatoires

Les arrivées suivent un processus de Poisson. Dans un intervalle  $dt$ , la probabilité d'une arrivée est :

$$P(\text{arrivée}) = 1 - e^{-\lambda \cdot dt} \quad (2.1)$$

Les temps de service suivent une loi exponentielle de paramètre  $\mu$  :

$$T_{service} = -\frac{1}{\mu} \ln(U), \quad U \sim \text{Uniform}(0, 1) \quad (2.2)$$

### 2.2 Condition de Stabilité

#### Intensité du trafic

L'intensité du trafic caractérise la charge du système :

$$\rho = \frac{\lambda}{K \cdot \mu} \quad (2.3)$$

L'interprétation de cette égalité est que  $\rho$  représente le ratio entre le taux d'arrivée et la capacité maximale de traitement.

#### Stabilité du système

- Si  $\rho < 1$  : Le système est stable
- Si  $\rho \geq 1$  : Le système sature
- En pratique, nous cherchons à maintenir  $\rho < 0.8$  pour absorber les variations

## 2.3 Loi de Little : Validation des Résultats

La Loi de Little établit une relation fondamentale en régime stationnaire :

$$L = \lambda \cdot W \quad (2.4)$$

où :

- $L$  : Nombre moyen de jobs dans le système
- $\lambda$  : Taux d'arrivée effectif (jobs/s)
- $W$  : Temps de séjour moyen (s)

### Utilisation pour la validation

Nous utilisons la Loi de Little pour valider nos simulations en suivant cet algorithme :

1. Mesurer  $\lambda$  (arrivées par seconde)
2. Mesurer  $W$  (moyenne des temps de séjour)
3. Calculer  $L_{théorique} = \lambda \cdot W$
4. Comparer avec  $L_{observé}$  (moyenne de la longueur totale du système)
5. Écart acceptable : < 5%

## 2.4 Métriques Collectées

Pour chaque simulation, nous collectons :

Métrique	Formule/Description
Temps de séjour moyen	$W = \frac{1}{n} \sum_{i=1}^n (t_{sortie,i} - t_{arrivée,i})$
Variance du temps	$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (W_i - \bar{W})^2$
Taux de rejet exec	$\frac{\text{rejets exec}}{\text{total arrivées}} \times 100$
Taux pages blanches	$\frac{\text{rejets result}}{\text{jobs complétés}} \times 100$
Utilisation serveurs	$\frac{\text{serveurs actifs}}{K} \times 100$

TABLE 2.1 – Métriques de performance collectées

### 3. Résultats : Modèle Waterfall

#### 3.1 Configuration des Expériences

Nous avons testé plusieurs configurations pour analyser l'impact des paramètres :

Expérience	Paramètres	Objectif
Exp. 1	$k_s, k_f$	Comportement de base
Exp. 2	$k_s, k_f$	Impact files finies
Exp. 3	$K$	Dimensionnement serveurs
Exp. 4	$p_{backup}$	Efficacité backup

TABLE 3.1 – Configurations expérimentales testées

#### 3.2 Résultats avec Files Infinies

Voici nos résultats avec la configuration de base :

- $K_{exec} = 3$  serveurs d'exécution
- $K_{result} = 1$  serveur de résultat
- $\lambda = 1.0$  job/s
- Temps moyen exécution = 2.5s
- Temps moyen résultat = 0.5s
- $k_s = k_f = \infty$
- Durée totale de l'expérience = 60000 secondes

Métrique	Valeur Mesurée
Jobs complétés	59769
Temps de séjour moyen ( $W$ )	7.02 s
Variance du temps de séjour	$29.15 \text{ s}^2$
Intensité du trafic ( $\rho$ )	0.83

TABLE 3.2 – Résultats avec files infinies

#### Analyse de la Stabilité

Théoriquement, un système M/M/K est stable si  $\rho < 1$ . Ici, avec  $\rho = 0.83$ , nous sommes en dessous du seuil critique (1.0). Le système est donc stable (le temps d'attente converge vers  $\approx 7\text{s}$ ) (cf. Figure 1). Cependant, une charge de 83% est élevée : le système est sensible aux variations et les files peuvent se remplir temporairement.

### 3.3 Impact des Files Finies et Analyse des Rejets

Le tableau ci-dessous montre l'impact de la réduction des capacités  $k_s$  et  $k_f$  sur les taux de rejet.

Les autres paramètres restent identiques à la configuration de base (cf. première expérience).

Capacités ( $k_s/k_f$ )	Rejet Exec (%)	Page Blanche %	Perte Totale (%)
$\infty/\infty$	0.00%	0.00%	0.00%
50/25	0.00%	0.00%	0.00%
20/10	0.23%	0.05%	0.28%
10/5	2.13%	0.77%	2.90%
5/5	6.45%	0.45%	6.90%

TABLE 3.3 – Impact des capacités de files sur les taux de rejet

On constate que la réduction drastique des capacités (5/5) entraîne un taux de rejet combiné critique de 6.90%. Le dimensionnement 20/10 offre un compromis plus acceptable avec un taux de perte global autour de 0.28%.

Ce résultat est confirmé visuellement par la saturation des filles d'attentes (cf. Figure 2 et Figure 3).

### Observations Clés

- Impact différencié des rejets : Il convient de distinguer les rejets d'exécution des rejets de résultats. Ces derniers sont jugés plus critiques car ils provoquent une rupture de service visible (pages blanches).
- Ratio de dimensionnement : Les résultats de simulation convergent vers un équilibre optimal lorsque le paramètre  $k_f$  représente environ la moitié de  $k_s$  ( $k_f \approx 0.5 \times k_s$ ).
- Seuil de tolérance : Afin de maintenir une qualité d'expérience utilisateur acceptable, il faut, selon l'objectif visé, déterminer un seuil de taux de rejet cumulé maximum.

### 3.4 Impact du Dimensionnement ( $K_{exec}$ )

Nous avons fait varier le nombre de serveurs d'exécution  $K_{exec}$  pour observer la transition entre zone stable et instable.

Les autres paramètres restent identiques à la configuration de base (cf. première expérience).

K	Charge ( $\rho$ )	Temps Séjour (s)	Variance	État
2	1.25	Saturé	Énorme	Instable
3	0.83	7.21	28.76	Stable (Tendu)
4	0.63	3.97	8.55	Confortable
5	0.50	3.60	7.53	Surdimensionné

TABLE 3.4 – Impact du nombre de serveurs sur la performance

**Analyse :**

- Pour  $K = 2$  ( $\rho = 1.25$ ), le système est en situation de saturation et ne parvient pas à absorber la charge, ce qui entraîne une divergence du temps d'attente (cf. Figure 4).
- Avec  $K = 3$  ( $\rho = 0.83$ ), le fonctionnement devient stable mais demeure précaire en raison d'une variance excessive (28.76).
- À  $K = 4$  ( $\rho = 0.63$ ), on atteint le dimensionnement optimal où le temps de séjour chute à 3.97s et la variance se stabilise.
- Enfin, pour  $K = 5$  ( $\rho = 0.50$ ), le gain de performance est marginal (3.60s) et rend le système inutilement surdimensionné (cf. Figure 5).

### 3.5 Analyse du Mécanisme de Backup

Nous avons simulé différents taux de backup ( $p_{backup}$ ) sur une configuration contrainte ( $k_s = 10, k_f = 5$ ) pour évaluer la récupération des pages blanches.

Les autres paramètres restent identiques à la configuration de base (cf. première expérience).

Probabilité Backup	Pages Blanches (nb)	Sauvés (nb)	Perte Effective (%)
0%	532	0	0.91%
20%	369	92	0.63%
50%	251	261	0.42%
80%	100	344	0.17%
100%	0	463	0.00%

TABLE 3.5 – Efficacité du mécanisme de backup

### Observations

D'après nos simulations :

- Une probabilité de backup à 100% élimine totalement les pages blanches.
- Une probabilité de backup à 50% réduit déjà la perte de manière significative (0.42% vs 0.91%).

## 4. Résultats : Modèle Channels & Dams

### 4.1 Configuration Multi-Classes

Nous avons configuré le scénario multi-classes comme suit :

- ING :  $\lambda = 0.8$  jobs/s, temps de service=2.0s
- PREPA :  $\lambda = 0.2$  jobs/s, temps de service=5.0s
- Charge totale : 1.0 jobs/s.
- Files d'attente : Infinies.
- Temps moyen résultat : 0.5s.
- Barrage (Dam) : Désactivé.
- Ressources :  $K_{exec} = 3$ ,  $K_{result} = 1$ .
- Durée de simulation : 60000 secondes.

L'intensité du trafic pour chaque groupe est :

- $\rho_{ING} = \frac{0.8 \times 2.0}{3} \approx 0.53$
- $\rho_{PREPA} = \frac{0.2 \times 5.0}{3} \approx 0.33$ .

Ainsi, l'intensité totale est :  $\rho \approx 0.87$  (Système chargé pour accentuer les différences).

### 4.2 Comparaison des Stratégies d'Ordonnancement

Stratégie	Temps Moyen Global (s)	Temps ING (s)	Temps PREPA (s)
FIFO	11.14	10.54	13.49
SJF	7.81	4.85	19.60
ING First	7.83	4.85	19.85
PREPA First	11.78	13.00	6.87

TABLE 4.1 – Comparaison des stratégies d'ordonnancement

### Analyse de l'équité

Nous nous intéressons ici au ratio  $\frac{\text{Temps PREPA}}{\text{Temps ING}}$  :

- FIFO : Le ratio est de 1.3. C'est l'algorithme le plus "neutre".
- SJF : Le ratio est de 4.0. C'est un cas de famine massive pour les PREPA (19.60s d'attente vs 4.85s pour les ING). Ici, l'algorithme SJF sacrifie la minorité "lente" pour optimiser la moyenne globale.

- ING First : Similaire à SJF dans notre cas.
- PREPA First : Le ratio est de 0.5. Cela inverse la tendance mais est inefficace globalement (moyenne de 11.78s d'attente, ce qui est la pire selon nos tests).

Nous pouvons donc voir que l'algorithme SJF est le meilleur au global avec un temps d'attente moyen de 7.81 secondes. Cependant, il crée une inéquité importante entre le temps d'attente des ING et celui des PREPA ( $\approx$  4 fois plus long pour les PREPA).

### 4.3 Effet du Mécanisme de Dam

Configuration (Ouverture/Fermeture)	Temps Moyen Global (s)	Temps ING (s)	Temps PREPA (s)
Sans Dam	10.19	9.59	12.54
Avec Dam (5s/25s)	29.24	28.66	31.56
Avec Dam (5s/20s)	69.53	68.85	72.29

TABLE 4.2 – Impact du Dam sur les temps de séjour

### Observations

Le Dam, même dimensionné au mieux, dégrade massivement la performance globale du système (cf. Figure 6 et Figure 7).

En effet, contrairement à notre intuition initiale, le mécanisme de barrage est contre-productif pour réduire l'attente dans ce système spécifique. Son utilisation ne se justifie que pour des contraintes de protection (si un serveur ne peut pas encaisser trop de requêtes) et non pour l'optimisation des flux.

## 5. Synthèse et Recommandations Opérationnelles

### 5.1 Synthèse des Résultats

Scénario	Meilleure Config	Performance Clé
Waterfall (Charge normale)	Files Infinies, K=3	0% Rejet, Attente $\approx 7\text{s}$
Waterfall (Charge haute)	Backup 100%	0% Page Blanche
Channels (Équité)	FIFO	Ratio $\approx 1.3$
Channels (Efficacité)	SJF	Attente globale min (7.8s)

TABLE 5.1 – Synthèse des recommandations

### Validation du Modèle M/M/K

Nos simulations valident les prédictions théoriques :

- Stabilité observée pour  $\rho < 1$
- Loi de Little vérifiée avec écart  $< 5\%$
- Distribution des temps d'attente conforme à la théorie

### 5.2 Recommandations pour le Dimensionnement

#### Nombre de Serveurs

Pour un taux d'arrivée  $\lambda$  donné :

$$K_{min} = \lceil \frac{\lambda}{\mu \times 0.8} \rceil \quad (5.1)$$

Ce résultat est vérifié au cours de nos expériences sur le modèle Waterfall, et il permet aussi de maintenir  $\rho < 0.8$  pour absorber les variations.

#### Capacités des Files

- $k_s$  : Au moins  $2 \times K$  pour amortir les arrivées en rafale
- $k_f$  : Environ  $0.5 \times k_s$  pour un traitement du résultat plus rapide

#### Stratégie de Backup

- Si aucune perte n'est tolérée :  $p_{backup} = 100\%$  malgré le coût
- Si le budget est limité :  $p_{backup} = 30 - 50\%$  offre un bon compromis
- Métrique à monitorer : Taux de pages blanches en production

## 5.3 Recommandations pour le Scénario Multi-Classes

### Choix de la Stratégie

Nous recommandons de choisir la stratégie en fonction de l'objectif prioritaire de l'administration :

- Objectif Efficacité Maximale (SJF/ING First) : Cette stratégie est la seule permettant de réduire significativement le temps d'attente moyen global (7.8s contre plus de 10s), bien que ce gain se fasse au détriment des étudiants PREPA.
- Objectif Équité (FIFO) : Ce choix offre le ratio d'attente le plus équilibré (1.3) entre les différents usagers.
- Configurations à éviter : Le mécanisme de DAM s'avère contre-productif, tandis que la priorité PREPA First est déconseillée car elle pénalise excessivement la majorité des étudiants (ING).

## 5.4 Limites et Perspectives

### Limites de l'Étude

- Stationnarité du flux : Le modèle repose sur une hypothèse de stationnarité qui ne capture pas les variations réelles, comme les pics de charge observés en fin de période de rendu.
- Loi de service : L'utilisation de temps de service strictement exponentiels constitue une simplification théorique, les distributions réelles étant souvent plus complexes.
- Fiabilité du matériel : L'analyse actuelle ne prend pas en compte les aléas opérationnels tels que les pannes de serveurs (cela pourrait par ailleurs justifier l'utilisation du DAM).

### Perspectives d'Amélioration et Pistes Futures

Des solutions plus avancées pourraient être implémentées pour pallier ces limites :

- SJF avec mécanisme anti-famine : Cette approche permettrait de bénéficier de l'efficacité du SJF tout en augmentant dynamiquement la priorité des tâches dont le temps d'attente devient critique.
- Segmentation des ressources : L'allocation de serveurs dédiés ( $K_{ING}$  et  $K_{PREPA}$ ) proportionnellement au trafic garantirait une meilleure isolation des performances entre les populations.
- Modèle M/G/K : L'extension de l'étude à un modèle M/G/K permettrait d'évaluer le comportement du système avec des distributions de temps de service non exponentielles.
- Optimisation dynamique : Un système d'ajustement automatique du nombre de serveurs  $K$  en fonction de la charge instantanée permettrait d'optimiser la consommation des ressources.

## Conclusion

Cette étude a permis de démontrer la pertinence de la théorie des files d'attente appliquée à une infrastructure réelle de validation (la moulinette). Les simulations réalisées ont offert une analyse fine de la dynamique du système, permettant d'établir des recommandations robustes.

Les principaux enseignements de ces travaux sont les suivants :

1. Gestion de la charge : Un dimensionnement approprié des ressources est critique. Il est impératif de maintenir un taux de charge  $\rho < 0.8$  pour garantir la fluidité du service et éviter la saturation.
2. Robustesse du service : La mise en place d'un mécanisme de secours (le backup) s'avère indispensable pour prévenir les défaillances critiques, telles que l'affichage de pages blanches.
3. Dilemme Efficacité/Équité : Dans les systèmes multi-classes, un arbitrage est nécessaire entre l'algorithme SJF, qui maximise l'efficacité globale, et le FIFO, qui assure une meilleure équité entre les usagers.
4. Validation théorique : Les résultats expérimentaux confirment la validité des modèles théoriques classiques, offrant une vérification empirique de la Loi de Little et du modèle M/M/K.

En définitive, le simulateur développé constitue un outil à la fois pédagogique et analytique, permettant d'explorer diverses configurations et de visualiser concrètement leur impact sur les performances globales du système.

## **A**nnexes

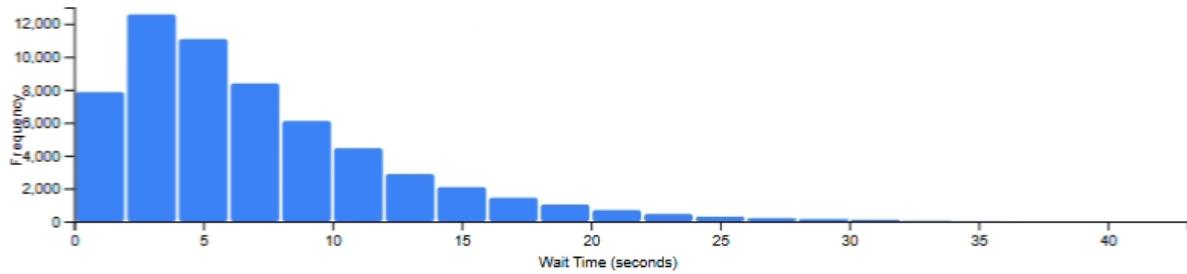
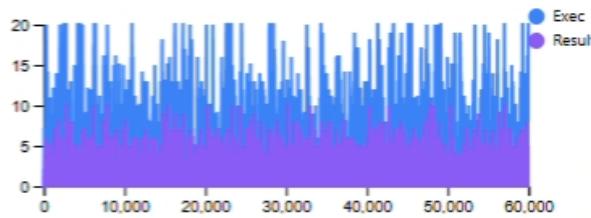
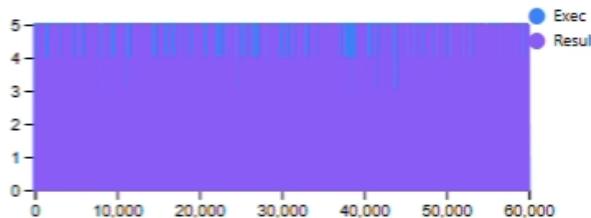
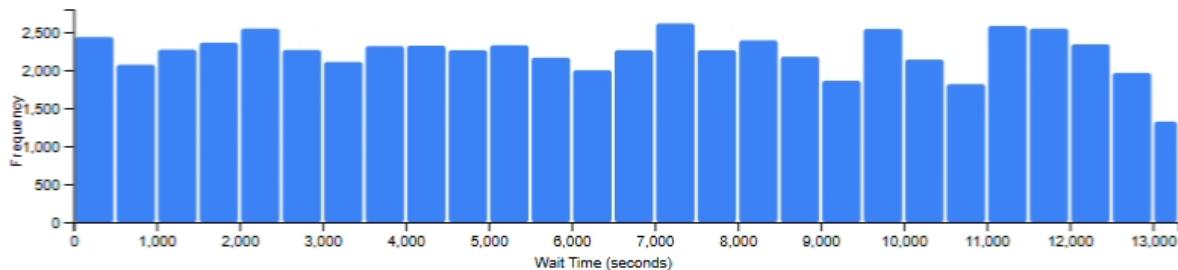
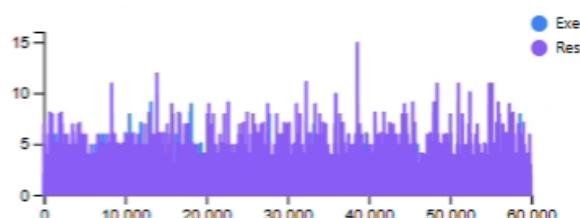


FIGURE 1 – Waterfall - Files infinies - Histogramme des temps d'attente

FIGURE 2 – Waterfall -  $k_s = 20, k_f = 10$  - Longueur des files au cours du tempsFIGURE 3 – Waterfall -  $k_s = 5, k_f = 5$  - Longueur des files au cours du tempsFIGURE 4 – Waterfall -  $K_{serv} = 2$  - Histogramme des temps d'attenteFIGURE 5 – Waterfall -  $K_{serv} = 5$  - Histogramme des temps d'attente

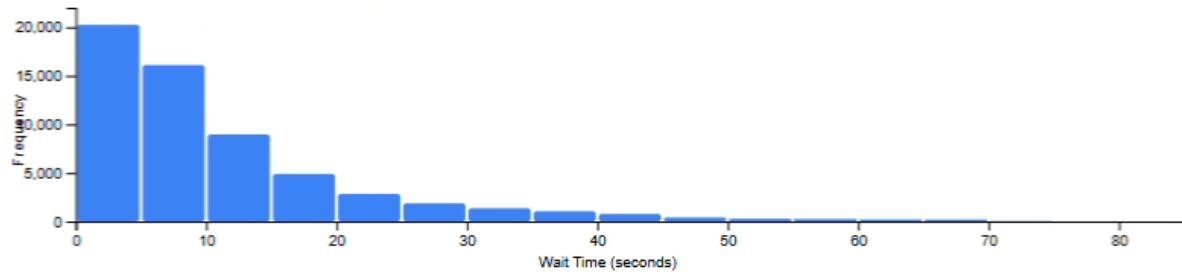


FIGURE 6 – Channels &amp; Dams - Pas de Dam - Histogramme des temps d'attente

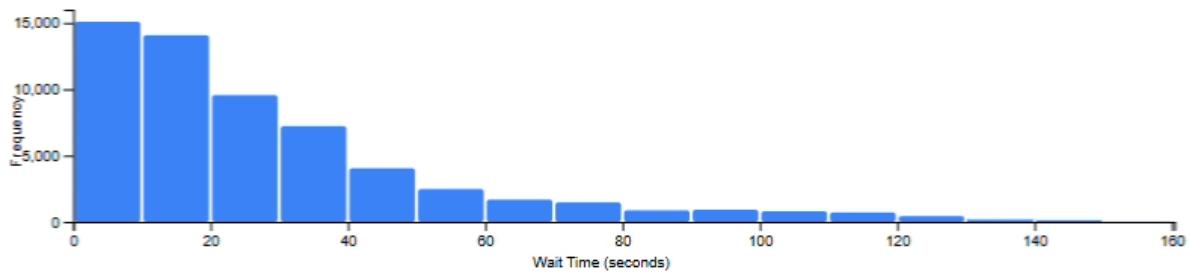


FIGURE 7 – Channels &amp; Dams - Fermeture = 5, Ouverture = 25 - Histogramme des temps d'attente