



# Quel effet a la priorisation des utilisateurs sur les performances du système lors des rendus de TP prépa et de piscine C ?

Étude des comportements de systèmes d'attente

*Éléments de Recherche Opérationnel*

Etienne Senigout, Paul Pazart, Pierre Braud, Lucas Burgaud,  
Kylian Bozec, Guillaume Hanry

Octobre 2025 - January 2026

---

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Problématique</b>	<b>4</b>
<b>3</b>	<b>Personas</b>	<b>4</b>
<b>4</b>	<b>Recherches préliminaires</b>	<b>5</b>
4.1	Processus de Poisson et variantes . . . . .	5
4.1.1	Définition . . . . .	5
4.1.2	Processus de Poisson non-homogène . . . . .	5
4.1.3	Limites et motivation pour d'autres lois . . . . .	5
4.2	Loi de Student . . . . .	5
4.2.1	Définition . . . . .	5
4.2.2	Propriétés . . . . .	6
4.2.3	Applications . . . . .	6
4.3	Loi exponentielle . . . . .	6
4.3.1	Définition . . . . .	6
4.3.2	Propriétés importantes . . . . .	6
4.3.3	Usage . . . . .	6
4.4	Chapman–Kolmogorov . . . . .	7
4.4.1	Équations générales . . . . .	7
4.4.2	Usage . . . . .	7
4.4.3	Limites . . . . .	7
<b>5</b>	<b>Génération des tags</b>	<b>8</b>
5.1	Génération des tags ING . . . . .	8
5.1.1	Approches testées . . . . .	8
5.1.2	Modélisation retenue . . . . .	8
5.1.3	Nombre de pushs par utilisateur . . . . .	8
5.1.4	Génération finale des tags . . . . .	9
5.2	Génération des tags PREPA . . . . .	10
5.2.1	Construction de la densité horaire . . . . .	10
5.2.2	Nombre de pushs et échantillonnage . . . . .	10
<b>6</b>	<b>Étude de cas</b>	<b>12</b>
6.1	Waterfall . . . . .	12
6.1.1	Visualisation . . . . .	12
6.1.2	Interrogations . . . . .	13
6.2	Channels & Dams . . . . .	14
6.2.1	Visualisation . . . . .	14
6.3	Anti-Priority . . . . .	15
6.3.1	Visualisation . . . . .	15
6.4	Calendar-Priority . . . . .	16
6.4.1	Visualisation . . . . .	16

<b>7 Solutions proposées</b>	<b>17</b>
7.1 Choix des scénarios . . . . .	17
7.2 Choix des critères . . . . .	18
7.3 Tableaux . . . . .	18
7.3.1 PREPA . . . . .	19
7.3.2 ING de Niveau moyen . . . . .	20
7.3.3 ING Fort . . . . .	21
7.4 Réponse à la problématique . . . . .	22
7.4.1 Calendar Priority . . . . .	22
7.4.2 Channels and Dams . . . . .	22
7.5 Conclusion . . . . .	23
<b>8 Sources</b>	<b>24</b>

# 1 Introduction

Dans ce rapport nous allons nous intéresser à l'infrastructure de correction automatique d'EPITA. Nous allons analyser ce système de moulinettage sous différentes angles mathématiques.

Le fonctionnement général du système est le suivant:

- Un étudiant push un tag sur un commit pour déclencher l'exécution de la test-suite associée à l'exercice correspondant afin de recevoir un retour sur sa conformité aux attentes.
- Une vérification est effectuée afin de s'assurer que le tag correspond à un exercice. Selon l'état du système, le tag sera soit exécuté immédiatement, soit mis en attente dans une file d'attente, ou soit rejeté si la file d'attente est pleine.

# 2 Problématique

Notre problématique est la suivante :

**Quel effet a la priorisation des utilisateurs sur les performances du système lors des rendus de TP prépa et de piscine C ?**

Nous nous sommes intéressés au scénario de la **piscine de C des ING** en parallèle des **TPs des Prépas**.

Pour rendre notre analyse pertinente, nous nous sommes appuyés sur des données réelles à l'aide des graphiques disponibles sur Grafana ainsi que nos propres données de piscines que nous avons scrappés directement depuis l'intra (tous les tags de chaque exercice avec leur temps de processing en ms).

# 3 Personas

Dans le cadre des recherches, nous avons besoin de formaliser des utilisateurs. Grâce à ces utilisateurs, nous allons pouvoir établir plusieurs scénarios pour faire varier l'architecture ainsi que la répartition des utilisateurs. Nous avons établi 3 personnes :

- **PREPA** : L'étudiant en cycle préparatoire rend des projets chaque semaine. Son rythme de travail est donc hebdomadaire : intense en début de semaine, plus faible au milieu, puis très soutenu à l'approche de la fin. Il travaille principalement le soir après les cours.
- **ING de Niveau moyen** : L'étudiant en 1ère année de cycle ingénieur est en piscine, il travaille de manière identique chaque jour à partir de 14h après les TPs.
- **ING Fort** : Cet étudiant en 1ère année de cycle ingénieur est en piscine aussi. Il a cependant un meilleur niveau, il utilise très peu de tag par exercice.

## 4 Recherches préliminaires

Cette section présente les recherches préliminaires sur les lois probabilistes utilisées pour modéliser l'arrivée et le traitement des tâches. L'objectif est de décrire leurs propriétés, définitions et applications dans notre contexte.

### 4.1 Processus de Poisson et variantes

#### 4.1.1 Définition

Un processus de Poisson modélise des événements discrets se produisant de manière aléatoire dans le temps, sous les hypothèses suivantes :

- Indépendance des événements,
- Probabilité d'un événement proportionnelle à la durée considérée,
- Absence de mémoire.

Le processus de Poisson homogène possède un taux constant  $\lambda$ . La probabilité de  $k$  événements dans un intervalle  $[0, t]$  est donnée par :

$$P(N(t) = k) = \frac{(\lambda t)^k e^{-\lambda t}}{k!}$$

#### 4.1.2 Processus de Poisson non-homogène

Lorsque le taux d'arrivée dépend du temps, on parle de processus de Poisson non-homogène, avec un taux  $\lambda(t)$ . La probabilité de  $k$  événements sur un intervalle  $[0, t]$  est alors :

$$P(N(t) = k) = \frac{\left(\int_0^t \lambda(s)ds\right)^k}{k!} \exp\left(-\int_0^t \lambda(s)ds\right)$$

Usage : modélisation de phénomènes où le flux d'événements varie dans le temps (pics et creux).

#### 4.1.3 Limites et motivation pour d'autres lois

Bien que le processus de Poisson non-homogène soit pertinent pour représenter un flux variable, il peut être difficile à paramétrier et moins flexible pour des distributions avec queues épaisses. Cela conduit à l'exploration de lois alternatives comme la **loi de Student**.

## 4.2 Loi de Student

#### 4.2.1 Définition

La loi de Student, notée  $t_\nu(\mu, \sigma)$ , est une loi continue définie par trois paramètres :

- $\mu$  : Position (centre de la distribution)
- $\sigma$  : Échelle (dispersion)

- $\nu$  : Degrés de liberté (détermine l'épaisseur des queues).

La densité de probabilité est donnée par :

$$f(x) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi} \Gamma\left(\frac{\nu}{2}\right) \sigma} \left[ 1 + \frac{1}{\nu} \left( \frac{x - \mu}{\sigma} \right)^2 \right]^{-\frac{\nu+1}{2}}$$

#### 4.2.2 Propriétés

- Distribution symétrique autour de  $\mu$ .
- Queues plus épaisses que la loi normale (utile pour capturer des événements extrêmes).
- Pour  $\nu \rightarrow \infty$ , la loi tend vers une loi normale.

#### 4.2.3 Applications

La loi de Student est particulièrement adaptée pour modéliser des phénomènes stochastiques avec pics concentrés mais variabilité importante. Elle permet :

- de représenter des distributions d'événements dans le temps avec des queues épaisses
- d'ajuster des paramètres par sous-groupes (par exemple exercice ou jour)
- de superposer plusieurs distributions pour obtenir des densités globales.

### 4.3 Loi exponentielle

#### 4.3.1 Définition

La loi exponentielle décrit le temps entre deux événements dans un processus de Poisson. Sa densité de probabilité est :

$$f(t) = \lambda e^{-\lambda t}, \quad t \geq 0$$

où  $\lambda$  est le taux moyen d'occurrence.

#### 4.3.2 Propriétés importantes

- **Absence de mémoire** :  $P(T > t + s \mid T > s) = P(T > t)$
- Espérance :  $E[T] = 1/\lambda$
- Variance :  $\text{Var}(T) = 1/\lambda^2$

#### 4.3.3 Usage

La loi exponentielle est utilisée pour modéliser le temps de service ou de traitement, là où les événements se produisent indépendamment et à taux constant.

## 4.4 Chapman–Kolmogorov

### 4.4.1 Équations générales

Pour un processus markovien discret :

$$P(X_{t+h} = j \mid X_0 = i) = \sum_k P(X_{t+h} = j \mid X_t = k)P(X_t = k \mid X_0 = i)$$

Pour un processus continu avec générateur infinitésimal  $\Lambda$  :

$$\frac{d}{dt}P(t) = P(t)\Lambda$$

### 4.4.2 Usage

Ces équations permettent de :

- décrire l'évolution de la distribution d'états dans le temps ;
- calculer les probabilités de transition pour différentes durées ;
- relier l'infime ( $t \rightarrow 0$ ) au global.

### 4.4.3 Limites

Dans des systèmes avec taux non-homogènes ou pics de flux irréguliers, les hypothèses de Markov homogène sont violées. Cependant, elles restent utiles pour :

- comprendre la dynamique théorique du système ;
- servir de base à des simulations numériques.

## 5 Génération des tags

### 5.1 Génération des tags ING

L'objectif est de générer des tags ING réalistes. Pour cela, nous nous basons sur les pushs effectués par les 6 membres de notre groupe lors de la piscine de C et sur les graphiques Grafana de la Forge lors de notre piscine.

Nous récupérons, pour chaque exercice, l'ensemble des dates et heures de pushs. Ces données sont ensuite organisées par couple (exercice, jour). Seules les soumissions effectuées pendant les heures d'ouverture de la moulinette (de 14h à minuit) sont conservées. Les périodes de réouverture ainsi que les phases de ratrapage n'ont pas été prises en compte.

#### 5.1.1 Approches testées

- **Mélange de lois gaussiennes par exercice.** Pour chaque exercice, une loi normale est ajustée à partir des heures de push (en calculant la moyenne et la variance), puis toutes les lois sont sommées. Cette approche produit des distributions trop étalées, générant des pushs en dehors des horaires autorisés.
- **Lois gaussiennes par exercice et par jour.** Cette méthode respecte mieux la contrainte journalière, mais les distributions obtenues sont trop symétriques et ne correspondent pas aux données observées sur le Grafana.

#### 5.1.2 Modélisation retenue

Nous avons finalement retenu la loi de Student, plus flexible et mieux adaptée aux distributions asymétriques observées.

Pour chaque exercice  $e$  et chaque jour  $d$ , on considère la liste des heures de push normalisées

$$\mathcal{H}_{e,d} = \{h_1, \dots, h_n\}, \quad h_i \in [0, T],$$

où  $T$  est la durée d'ouverture de la moulinette. Une loi de Student est ajustée sur ces données :

$$h \sim \mathcal{T}(\nu_{e,d}, \mu_{e,d}, \sigma_{e,d}).$$

La densité est ensuite discrétisée par intervalles de 5 minutes et normalisée afin d'obtenir une loi de probabilité exploitable pour l'échantillonnage.

#### 5.1.3 Nombre de pushs par utilisateur

Les utilisateurs sont séparés en deux catégories : utilisateurs *forts* et utilisateurs *normaux*. Pour chaque exercice  $e$ , nous définissons une difficulté normalisée :

$$d_e = \frac{N_e}{\max_{e'} N_{e'}},$$

où  $N_e$  est le nombre total de pushs observés pour l'exercice  $e$ . Ainsi,  $d_e \in [0, 1]$ .

Le nombre de pushs effectués par un utilisateur dépend de son niveau et de la difficulté de l'exercice. Il est modélisé par une loi Gamma :

$$X \sim \Gamma(k, \theta), \quad \mathbb{E}[X] = \mu = \mu_{\text{base}} + d_e(\mu_{\text{max}} - \mu_{\text{base}}),$$

où les paramètres  $\mu_{\text{base}}$  et  $\mu_{\text{max}}$  dépendent du type d'utilisateur et sont fixés à partir des statistiques globales de la promotion fournies par les ACUs : pour les utilisateurs forts,  $\mu_{\text{base}} = 1.0$  et  $\mu_{\text{max}} = 8.0$ , et pour les utilisateurs normaux,  $\mu_{\text{base}} = 2.0$  et  $\mu_{\text{max}} = 12.0$ . La loi Gamma est utilisée car elle est asymétrique, ce qui permet de représenter une distribution des pushes plus réaliste, et ses paramètres peuvent être ajustés facilement en fonction de la difficulté de l'exercice.

Le nombre final de pushes est donné par :

$$n = \max(1, \lceil X \rceil).$$

#### 5.1.4 Génération finale des tags

Pour chaque utilisateur, chaque exercice et chaque jour :

1. le nombre de pushes est tiré selon la loi Gamma.
2. les heures de push sont échantillonnées depuis la loi de Student discréétisée.
3. un léger bruit aléatoire (inférieur à 5 minutes) est ajouté afin d'atténuer l'effet de la discréétisation.

Voici le résultat des pushes sur les deux populations:

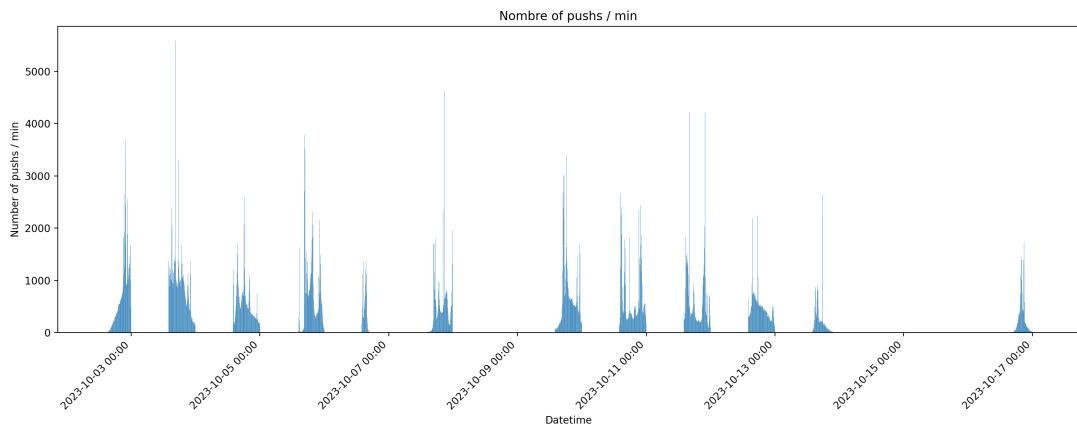


Figure 1: Visualisation de la distribution des tags pour les INGs

## 5.2 Génération des tags PREPA

Pour la prépa, nous n'avons pas pu nous baser sur des données réelles car pour notre promotion il n'y avait pas de système de moulinette pour les TPs. La génération des pushs est donc basée sur notre expérience, des discussions avec d'autres étudiants et les observations des ACDC et ASM. Plusieurs facteurs ont été pris en compte pour modéliser les comportements :

- Un TP par semaine est à rendre le dimanche soir, disponible le lundi après-midi.
- La majorité des pushs se fait après les cours et le soir, avec un pic principal après le dîner (22h-1h) et quasiment aucun push durant la nuit.
- Lors de l'ouverture du sujet, certains étudiants travaillent un peu et font quelques pushs. Durant la semaine, l'activité est faible, un peu plus le vendredi, puis beaucoup le samedi et dimanche, à la dernière minute.

### 5.2.1 Construction de la densité horaire

On définit d'abord une grille horaire pour toute la période considérée (nombre de semaines  $\times$  7 jours), discrétisée avec une résolution  $\Delta t$  (en heures). Pour chaque jour  $d$ , on construit une densité horaire combinant deux composantes :

$$f_d(t) = (1 - w_{\text{exp}}) f_{\text{Student}}(t \mid \nu_d, \mu_d, \sigma_d) + w_{\text{exp}} f_{\text{exp}}(t \mid t \geq \mu_d, \lambda_d),$$

Où :

- $f_{\text{Student}}$  est une loi de Student centrée sur le pic de la journée (fin de cours / soirée),
- $f_{\text{exp}}$  est un décroissant exponentiel modélisant les pushs supplémentaires après le pic,
- $w_{\text{exp}}$  est le poids relatif de la composante exponentielle,
- $\mu_d, \nu_d, \sigma_d$  et  $\lambda_d$  sont ajustés selon le type de jour (semaine ou week-end) et la position dans la semaine.

Chaque densité quotidienne est ensuite pondérée par un facteur  $p_d$  représentant le poids du jour dans la semaine (ex : plus lourd le jour du rendu), puis toutes les densités sont sommées et normalisées pour obtenir une distribution horaire globale  $P(t)$  :

$$\sum_t P(t) = 1.$$

### 5.2.2 Nombre de pushs et échantillonnage

Le nombre total de pushs à générer est calculé selon la taille de la population  $N$  et le nombre moyen de pushs par étudiant  $\bar{X}$  :

$$N_{\text{pushs}} = N \cdot \bar{X} \cdot \sum_d p_d.$$

Les heures de pushs sont ensuite échantillonées aléatoirement selon la densité  $P(t)$ , puis attribuées à un étudiant choisi uniformément parmi la population. Chaque échantillon

devient un tag avec date et heure précises.

Voici le résultat des pushes pour les prépas :

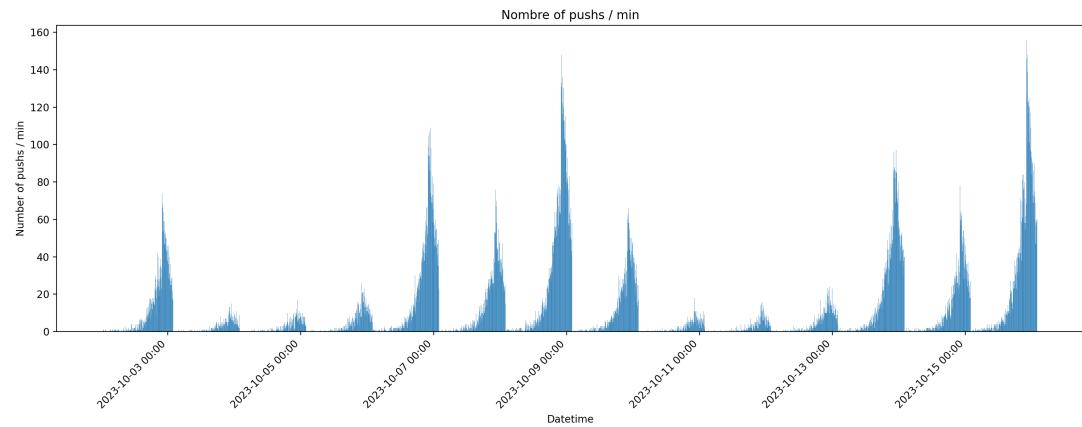


Figure 2: Visualisation de la distribution des tags pour la prépa

## 6 Étude de cas

### 6.1 Waterfall

Ce système est le plus basique : une file d'entrée et une file de sortie toutes les deux infinies.

Aucun tag ne peut être rejeté, dans le pire des cas des élèves peuvent se retrouver à attendre longtemps si le(s) serveur(s) n'arrivent pas à gérer une charge conséquente de tags.

#### 6.1.1 Visualisation

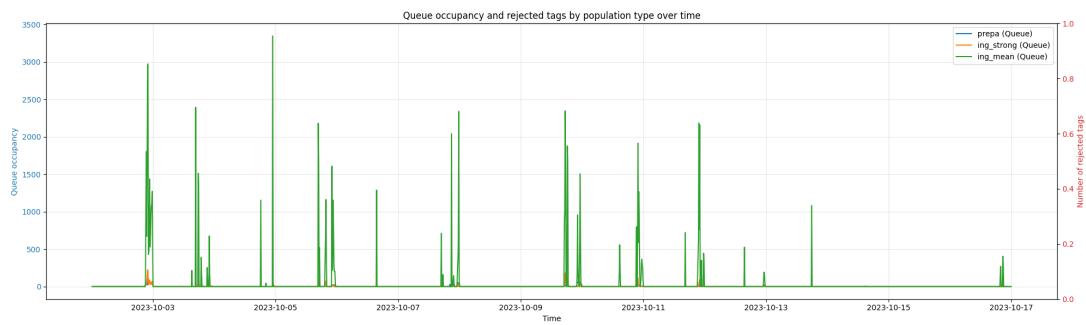


Figure 3: Visualisation de l'occupation

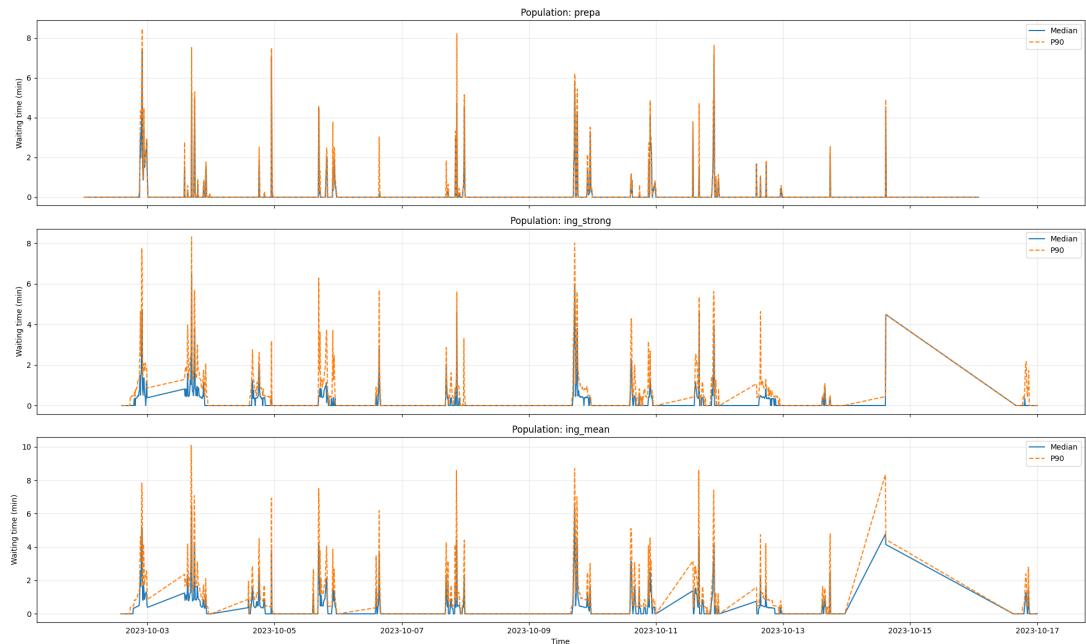


Figure 4: Visualisation des temps d'attentes

### 6.1.2 Interrogations

#### Quel changement cela opère-t-il sur la proportion de pages blanches ?

Concernant la proportion de pages blanches, logiquement, elle va diminuer.

Le fait de mettre en place une backup juste après le moulinettage agit comme une sécurité. Si jamais le transfert vers la file de sortie échoue, on ne perd pas le résultat puisqu'il est sauvegardé. On peut donc récupérer la donnée plutôt que d'afficher une page blanche.

#### Quels problèmes peuvent surgir avec cette solution ?

Les soucis peuvent être les suivants :

1. La latence : Pour chaque tag traité, on le sauvegarde pour la backup. On rajoute donc un délai supplémentaire, ce qui, cumulé, augmente le temps de réponse global.
2. Goulot d'étranglement : Si le système de sauvegarde est plus lent que le "moulinettement", les requêtes vont s'entasser juste avant l'envoi vers la seconde file. On risque de créer une nouvelle file d'attente implicite qui va saturer le système.
3. Coût stockage : La sauvegarde systématique de chaque tag demande beaucoup d'espace disque de parts leurs nombres.

#### Discutez des avantages d'un backup aléatoire plutôt que systématique.

Passer sur un backup aléatoire (par exemple une requête sur 10) a plusieurs avantages :

1. Dans un premier temps, cela nous permet d'augmenter fortement en fluidité. Nous évitons donc de pénaliser tous les utilisateurs via le temps d'écriture du backup. Ainsi, la majorité des requêtes transmises seront traités efficacement et rapidement, ce qui permet un bon maintien du débit global.
2. Enfin, avoir un backup aléatoire permet un monitoring (surveillance) sans surcharger la machine. L'échantillon représentatif permet de vérifier que les calculs sont corrects, sans avoir la nécessité de tout stocker. Nous avons donc ici un bon compromis entre sécurité et vitesse.

#### Calculez le temps de séjour moyen et la variance empirique dans ce modèle.

La variance et le temps moyen vont augmenter (avec un modèle aléatoire) parce qu'on crée un déséquilibre : certains élèves vont passer très vite (sans backup) et d'autres seront ralentis (avec backup). Cet écart de temps va disperser les résultats, rendant le temps de réponse moins prédictible (donc une variance plus élevée).

## 6.2 Channels & Dams

Ce système repose sur l'imitation d'un barrage hydraulique. Le cycle de traitement alterne entre deux phases :

- une de durée  $t$  où les tags de PREPA sont traités tandis que ceux des ING sont bloqués et collectés.
- une autre de durée  $t/2$  où on fait l'inverse : les tags des ING sont traités tandis que ceux des PREPA sont bloqués et collectés.

L'enjeu de ce système repose sur la sélection de ce temps  $t$  pour minimiser le temps d'attente.

### 6.2.1 Visualisation

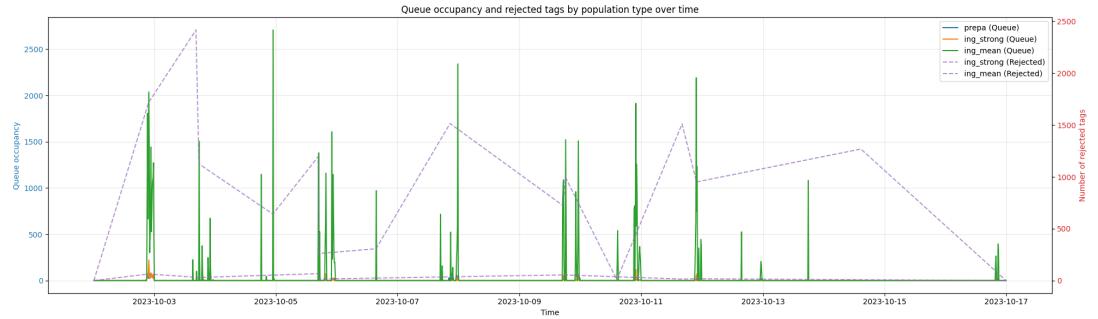


Figure 5: Visualisation de l'occupation

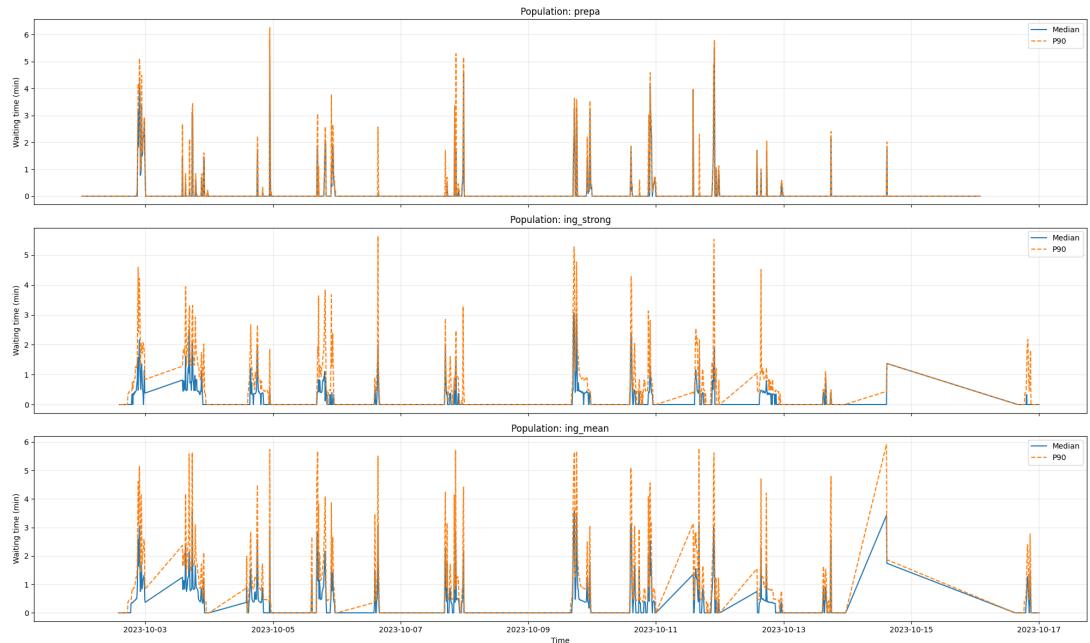


Figure 6: Visualisation des temps d'attentes

## 6.3 Anti-Priority

Ce système se base sur le fait de faire passer en priorité des tags provenant de la file d'attente la moins saturée. Pour assigner cette priorité, on crée un historique des 100 derniers tags traités, et on attribue un poids plus fort à la population la moins représentée dans l'historique.

Cette mécanique permet de ne pas bloquer, ni d'augmenter drastiquement le temps de correction d'un exercice pour une population sous-représentée dans les files d'attentes.

### 6.3.1 Visualisation

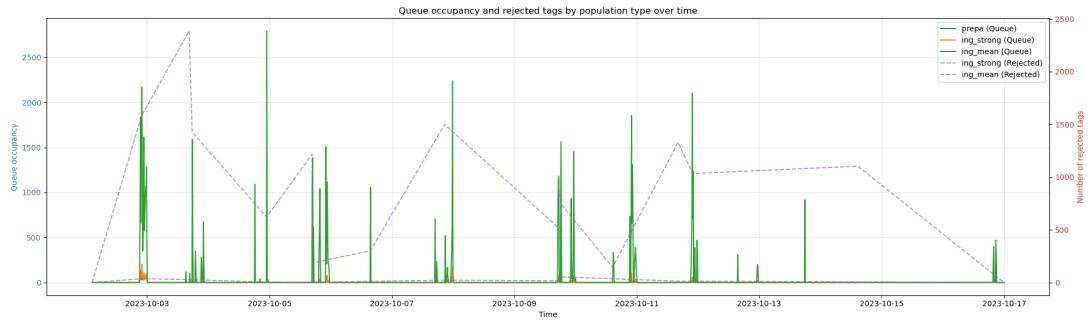


Figure 7: Visualisation de l'occupation

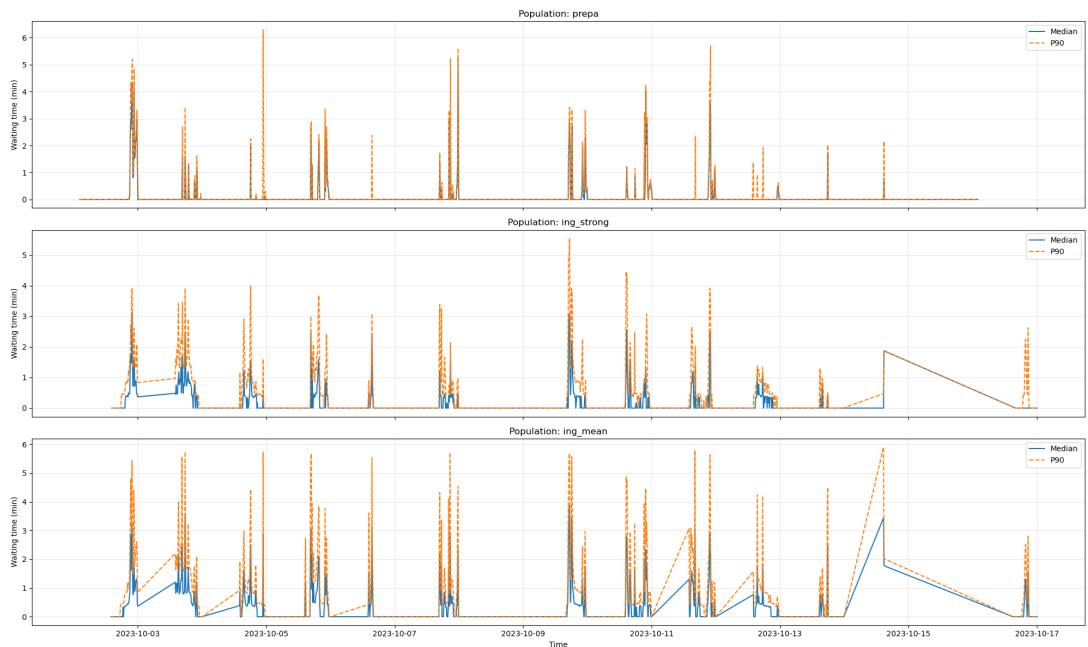


Figure 8: Visualisation des temps d'attentes

## 6.4 Calendar-Priority

Ce système ressemble à Channel & Dams, la différence étant que dans celui-ci nous définissons des plages horaires (ex: 14h à 16h) à laquelle est associée une population et sa priorité sur sa plage horaire.

Nous voulions pouvoir expérimenter en attribuant des priorités à certaines populations sur différents jours (et heures) de la semaine.

Cette mécanique est intéressante dans les cas où nous reconnaissions à l'avance les moments où chaque population va avoir tendance à push en grande quantité. Surtout si les populations ne push pas aux mêmes moments (la priorité prend tout son intérêt)(Oj non).

### 6.4.1 Visualisation

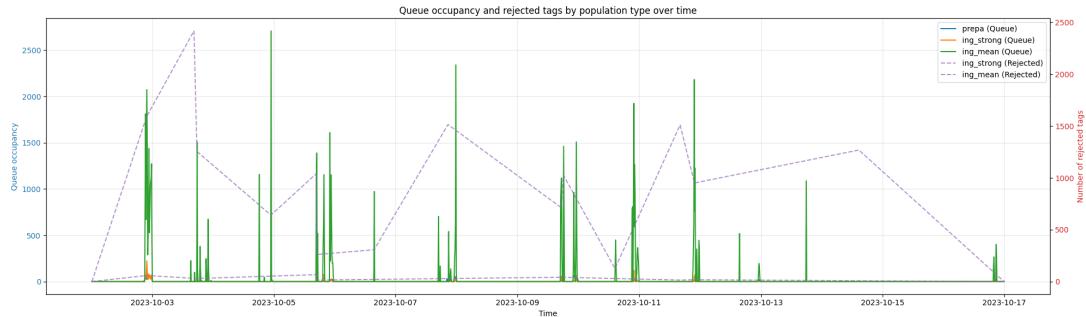


Figure 9: Visualisation de l'occupation

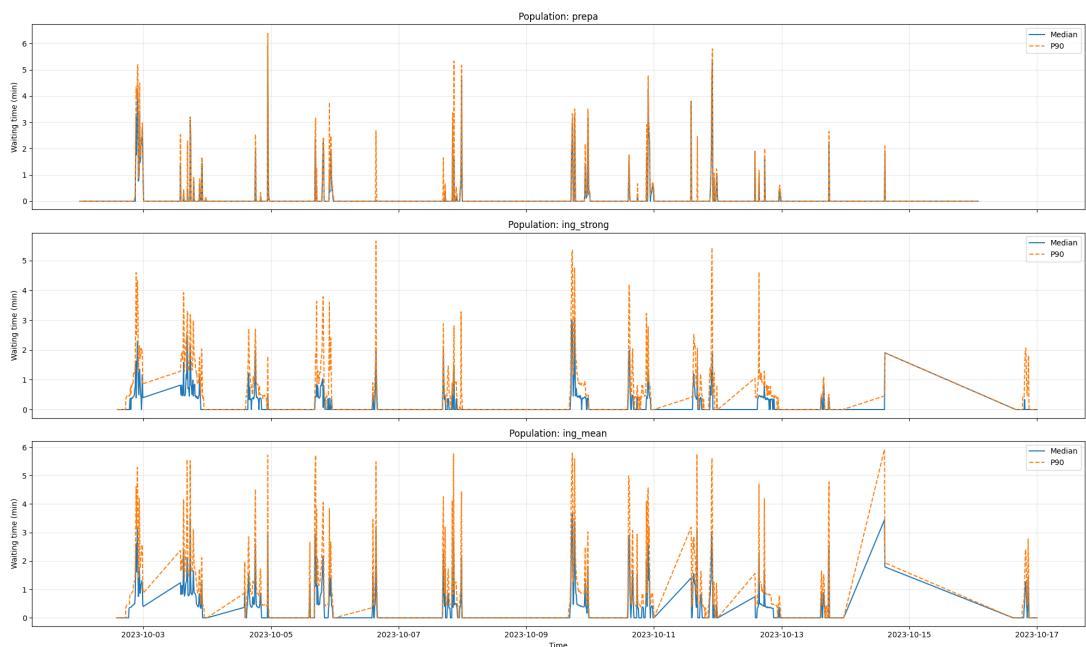


Figure 10: Visualisation des temps d'attentes

## 7 Solutions proposées

### 7.1 Choix des scénarios

Nous avons décidé de tester les quatre systèmes avec plusieurs scénarios. Chaque système sera d'abord testé avec des **critères de base** :

- **Serveur** : 20 serveurs, chacun contenant 10 pods (soit 200 environnements d'exécution).
- **Nombre d'élèves** : 600 élèves du cycle ingénieur et 1500 élèves du cycle préparatoire (première et deuxième année).
- **Proportion d'élèves** : 10% d'élèves forts du cycle ingénieur, ce sont des élèves qui utilisent moins de tags pour faire un exercice.
- **Taille des files** : 3000 pour le cycle ingénieur et 600 pour le cycle préparatoire.

Pour chaque scénario, nous allons **varier un seul critère à la fois** :

- **Serveur** : *critères de base* avec 5 serveurs (soit 50 environnements d'exécution)
- **Nombre d'élèves** : pas de variation réaliste dans le cadre d'EPITA
- **Proportion d'élèves** : *critères de base* avec 70% d'élèves forts
- **Taille des files** : *critères de base* avec 1500 pour le cycle ingénieur et 300 pour le cycle préparatoire

Enfin, pour deux systèmes, nous allons **doubler le nombre de scénarios selon un critère spécifique** :

- **Channel & Dams** : Temps d'ouverture pour la population principale (période de blocage)
  - 5 secondes
  - 30 secondes
- **Anti Priority** : Historique des tags
  - 100 derniers tags
  - 10 derniers tags

Ainsi, nous obtenons :

- 4 scénarios pour **Calendar Priority** et **Waterfall**
- 8 scénarios pour **Channel & Dams** et **Anti Priority**

Soit 24 scénarios au total.

## 7.2 Choix des critères

Nous avons choisi d'évaluer nos systèmes selon 6 critères (pour chacune des 3 populations) :

- Le pourcentage de tags rejetés par rapport au nombre total de pushes. Dans un cadre réaliste (en se basant sur notre expérience), celui-ci devrait tourner autour de 5%.
- Le temps d'attente, divisé en 4 sous-parties : **moyenne**, **maximum**, **médiane (P50)**, **Percentile 90% (P90)**.
- L'occupation moyenne de la file d'entrée, soit le nombre de personnes en moyenne dans la file à n'importe quel moment (bien que ce chiffre soit biaisé par les périodes nocturnes), cela reste une métrique intéressante dans les comparaisons inter-scénarios.

Soit 18 critères au total.

## 7.3 Tableaux

Nous avons récapitulé les données dans 3 tableaux. Chaque tableau contient les 24 scénarios avec les 6 critères d'une population.

### 7.3.1 PREPA

Scénario		% Tags rejetés	Temps d'attente				Occupation moy.
			Mean	Max	P50	P90	
Calendar Priority							
Cas de base		0.00%	0.16	6.51	<b>0.00</b>	0.11	1.12
5 serveurs		0.00%	3.97	28.65	<b>0.00</b>	21.05	11.03
70% d'élèves fort		0.00%	0.10	5.94	<b>0.00</b>	<b>0.00</b>	0.86
1500 ING - 300 PREPA		0.00%	0.12	3.57	<b>0.00</b>	0.06	1.06
Channels Dams							
T5s	Cas de base	0.00%	0.16	6.61	<b>0.00</b>	0.11	1.11
	5 serveurs	0.00%	3.99	28.85	<b>0.00</b>	21.29	11.06
	70% d'élèves fort	0.00%	<b>0.10</b>	5.82	<b>0.00</b>	<b>0.00</b>	0.85
	1500 ING - 300 PREPA	0.00%	0.12	<b>3.50</b>	<b>0.00</b>	0.07	1.06
T30s	Cas de base	0.00%	0.16	6.42	<b>0.00</b>	0.11	1.11
	5 serveurs	0.00%	3.94	28.60	<b>0.00</b>	20.96	10.92
	70% d'élèves fort	0.00%	0.10	5.84	<b>0.00</b>	<b>0.00</b>	<b>0.85</b>
	1500 ING - 300 PREPA	0.00%	0.12	3.55	<b>0.00</b>	0.06	1.06
Anti Priority							
H100	Cas de base	0.00%	0.16	6.37	<b>0.00</b>	0.13	1.16
	5 serveurs	0.00%	4.05	28.63	<b>0.00</b>	21.38	11.30
	70% d'élèves fort	0.00%	0.10	5.99	<b>0.00</b>	<b>0.00</b>	0.88
	1500 ING - 300 PREPA	0.00%	0.12	3.64	<b>0.00</b>	0.09	1.09
H10	Cas de base	0.00%	0.16	6.27	<b>0.00</b>	0.14	1.16
	5 serveurs	0.00%	3.95	29.34	<b>0.00</b>	20.98	11.01
	70% d'élèves fort	0.00%	0.10	5.77	<b>0.00</b>	<b>0.00</b>	0.88
	1500 ING - 300 PREPA	0.00%	0.12	3.65	<b>0.00</b>	0.08	1.09
Waterfall							
Cas de base		N/A	0.18	9.18	<b>0.00</b>	0.14	1.16
5 serveurs		N/A	46.60	441.71	<b>0.00</b>	234.37	100.10
70% d'élèves fort		N/A	0.10	6.42	<b>0.00</b>	<b>0.00</b>	0.86
1500 ING - 300 PREPA		N/A	N/A	N/A	N/A	N/A	N/A

### 7.3.2 ING de Niveau moyen

Scénario		% Tags rejetés	Temps d'attente				Occupation moy.
			Mean	Max	P50	P90	
Calendar Priority							
Cas de base		2.08%	1.13	7.38	0.77	2.98	32.97
5 serveurs		43.87%	13.60	29.22	16.64	22.98	327.08
70% d'élèves fort		0.89%	1.07	6.89	0.68	2.83	<b>10.67</b>
1500 ING - 300 PREPA		9.16%	0.84	<b>4.27</b>	0.50	2.14	21.08
Channels Dams							
T5s	Cas de base	2.09%	1.13	7.46	0.77	2.98	32.90
	5 serveurs	43.81%	13.71	29.73	16.96	23.06	329.90
	70% d'élèves fort	0.91%	1.06	6.84	0.68	2.83	10.70
	1500 ING - 300 PREPA	9.16%	0.84	4.32	0.50	2.14	21.09
T30s	Cas de base	2.10%	1.14	7.26	0.77	2.98	33.04
	5 serveurs	43.90%	13.59	29.41	16.61	22.97	326.92
	70% d'élèves fort	0.89%	1.06	6.88	0.68	2.84	10.68
	1500 ING - 300 PREPA	9.16%	<b>0.84</b>	4.36	0.50	<b>2.14</b>	21.07
Anti Priority							
H100	Cas de base	2.03%	1.14	7.22	0.77	3.00	33.49
	5 serveurs	43.79%	13.87	31.30	17.08	23.22	337.82
	70% d'élèves fort	0.58%	1.08	6.98	0.51	2.94	11.14
	1500 ING - 300 PREPA	9.16%	0.84	4.34	<b>0.50</b>	2.15	21.45
H10	Cas de base	2.07%	1.14	7.22	0.77	2.99	33.35
	5 serveurs	43.87%	13.70	31.63	16.80	22.96	333.85
	70% d'élèves fort	0.65%	1.08	7.02	0.50	2.93	11.13
	1500 ING - 300 PREPA	9.16%	0.84	4.49	<b>0.50</b>	2.15	21.47
Waterfall							
Cas de base		N/A	1.35	11.85	0.80	3.65	40.14
5 serveurs		N/A	116.00	445.64	83.28	268.97	4033.66
70% d'élèves fort		N/A	1.13	9.57	0.71	2.96	11.07
1500 ING - 300 PREPA		N/A	N/A	N/A	N/A	N/A	N/A

### 7.3.3 ING Fort

Scénario		% Tags rejetés	Temps d'attente				Occupation moy.
			Mean	Max	P50	P90	
Calendar Priority							
Cas de base		0.46%	0.72	7.30	0.40	1.99	1.35
5 serveurs		23.93%	13.67	28.77	17.42	22.80	27.45
70% d'élèves fort		0.02%	0.72	6.69	0.42	1.92	9.78
1500 ING - 300 PREPA		2.76%	0.57	4.26	<b>0.38</b>	<b>1.64</b>	<b>0.87</b>
Channels Dams							
T5s	Cas de base	0.49%	0.72	7.21	0.40	2.00	1.34
	5 serveurs	23.79%	13.79	28.65	17.64	22.88	27.70
	70% d'élèves fort	0.03%	0.72	6.75	0.42	1.92	9.81
	1500 ING - 300 PREPA	2.76%	0.57	4.34	<b>0.38</b>	1.64	0.88
T30s	Cas de base	0.40%	0.73	7.05	0.40	2.02	1.36
	5 serveurs	24.11%	13.66	28.52	17.43	22.77	27.38
	70% d'élèves fort	0.02%	0.72	6.89	0.42	1.92	9.79
	1500 ING - 300 PREPA	2.76%	<b>0.57</b>	4.34	<b>0.38</b>	1.64	0.88
Anti Priority							
H100	Cas de base	0.47%	0.73	7.01	0.41	2.03	1.34
	5 serveurs	28.17%	13.55	30.33	17.39	22.84	26.20
	70% d'élèves fort	0.13%	0.71	6.69	0.42	1.95	9.55
	1500 ING - 300 PREPA	2.61%	0.58	<b>4.16</b>	0.38	1.68	1.01
H10	Cas de base	0.48%	0.72	6.42	0.42	2.02	1.32
	5 serveurs	28.43%	13.35	30.21	17.00	22.54	25.73
	70% d'élèves fort	0.13%	0.71	6.75	0.42	1.95	9.53
	1500 ING - 300 PREPA	2.61%	0.58	4.36	0.38	1.68	1.00
Waterfall							
Cas de base		N/A	0.84	9.81	0.41	2.35	1.68
5 serveurs		N/A	113.88	444.16	80.76	267.45	283.55
70% d'élèves fort		N/A	0.72	7.88	0.42	1.95	10.07
1500 ING - 300 PREPA		N/A	N/A	N/A	N/A	N/A	N/A

## 7.4 Réponse à la problématique

Voici les solutions les plus optimales en réponse à notre problématique : "**Quel effet a la priorisation des utilisateurs sur les performances du système lors des rendus de TP prépa et de piscine C ?**"

### 7.4.1 Calendar Priority

C'est la solution la plus robuste pour gérer les rendus réguliers des TPs PREPAs et le flux massif des INGs lors de la piscine de C.

#### Justification par les données

- (Table 7.3.1) Le système garantit une fiabilité parfaite avec **0,00% de rejet**, même en situation de crise. Le temps d'attente moyen reste négligeable (0,16 min en temps normal, et contenu à 3,97 min avec 5 serveurs).
- (Table 7.3.3) L'effet de la priorisation est d'autant plus remarquable lors que peu de ressources sont disponibles (Scénario "5 serveurs") :
  - Waterfall: Le temps d'attente moyen explose à **113,88 minutes** avec un maximum de **444,16 minutes** (plus de 7 heures).
  - Calendar Priority: Le temps moyen est réduit à **13,67 minutes** et le maximum à **28,77 minutes**.
- **Coût de la performance** : Pour maintenir ces temps acceptables, le système rejette **23,93%** des tags en période de surcharge, ce qui est préférable à une paralysie totale.

### 7.4.2 Channels and Dams

Cette solution (notamment la variante  $T = 5s$ ) offre des performances presque identiques à la version *Calendar Priority*, ce qui en fait une excellente alternative basée sur la régulation de flux.

#### Justification par les données

- **Performance équivalente** : Dans le scénario dit "critique" (ING de niveau moyen / 5 serveurs), le temps d'attente moyen est de **13,79 minutes** (comparable aux 13,67 min de la Calendar Priority).
- **Maîtrise du rejet** : Le taux de rejet se stabilise autour de **23,79%**, prouvant ainsi que le mécanisme de gestion de flux est aussi efficace que la priorité calendaire pour égualiser la charge sans rejeter d'utilisateurs inutilement.

Nous avons écartées certaines solutions : **Waterfall** et **Anti-Priority**, qui se sont révélées bien moins intéressantes que les deux autres.

Les données de **Waterfall** démontrent que cette solution est un échec total, cette analyse est d'ailleurs biaisée car des files infinies sont impossibles dans un cadre réaliste. Etant donné qu'aucun tag n'est rejeté, plus le nombre de serveur est réduit, plus les tags s'accumulent et le temps d'attente des étudiants également : **2 à 7 heures** (avec 5 serveurs)!

**Anti Priority**, bien que fonctionnelle, est moins performante que **Calendar** et **Channel & Dams**. Trop de tags sont rejetés en comparaison sans offrir de gain significatif sur les temps d'attentes.

## 7.5 Conclusion

Voici qui conclut notre analyse, nous avons pu filtrer à travers 24 scénarios distincts deux solutions particulièrement adaptées au problème et dont les résultats font écho à notre expérience personnelle, notamment durant la piscine de C.

## 8 Sources

- Wikipedia — *Poisson Process*, section « Definition and properties »
- Wolfram Language — *Inhomogeneous Poisson Process*, section « Extended Probability »
- Wikipedia — *Student's t-distribution*, section « Probability density function »
- Bibmath — *Loi de Student*
- Wikiversity — *Loi de Student*
- Maths-et-tiQues — *Lois de Student et loi normale*
- Wikipedia — *Exponential distribution*, section « Properties »
- Bibmath — *Loi exponentielle*
- Wikipedia — *Chapman–Kolmogorov equation*, section « Applications in Markov processes »
- Moodle EPITA — *Théorie des files d'attente*, cours « Processus stochastiques et files »
- ResearchGate — *Steady-state probability distributions in stochastic processes*