Chapitre 04: Files d'attente

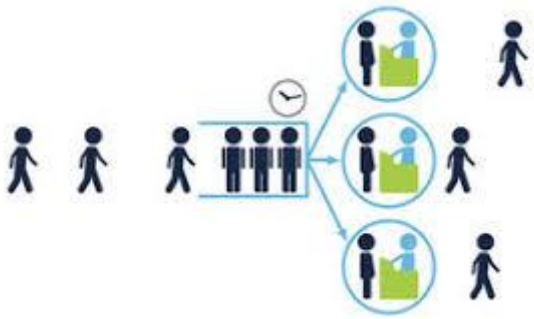
Introduction

Les files d'attente apparaissent lorsqu'un service ne peut traiter immédiatement toutes les demandes reçues. Elles se forment naturellement dans les systèmes informatiques, comme les serveurs, les processeurs ou les réseaux. Étudier leur comportement permet de comprendre les délais, la congestion et l'utilisation des ressources. La théorie des files d'attente offre des outils pour modéliser ces situations et prévoir leurs performances. Ce chapitre présente les concepts essentiels liés aux files d'attente, leurs types et leurs caractéristiques.

1. Notions fondamentales

1.1. File d'attente

Une file d'attente est une structure d'attente organisée dans laquelle des entités (appelées clients, tâches ou requêtes) arrivent pour recevoir un service fourni par un ou plusieurs serveurs. Lorsque tous les serveurs sont occupés, les entités en attente sont mises en file selon une certaine discipline jusqu'à ce qu'un serveur soit disponible.



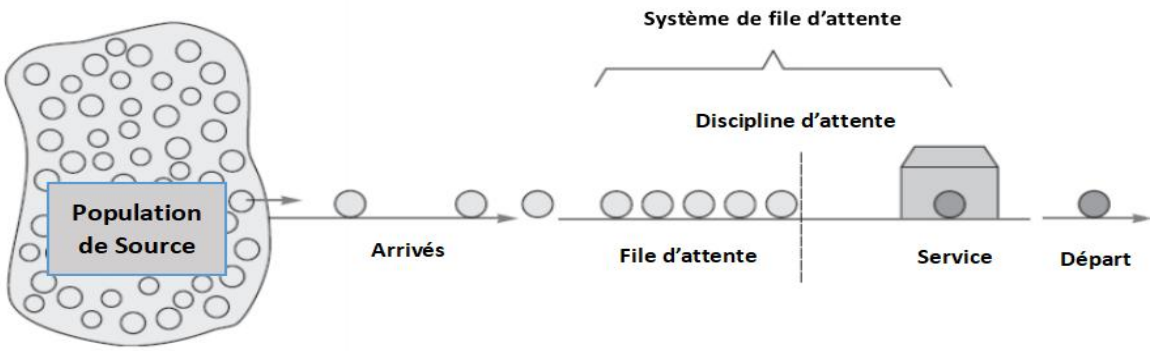
1.2. Théorie des files d'attente

La théorie des files d'attente, ou ***«Queuing Theory»*** en anglais, est une branche des mathématiques appliquées et des probabilités qui s'intéresse à l'étude des phénomènes d'attente. Elle

modélise et analyse les systèmes où des **clients** (personnes, données, tâches, véhicules, etc.) se présentent à un **serveur** (guichet, processeur, machine, routeur, etc.) pour recevoir un **service**.

La théorie des files d'attente cherche à comprendre et à prédire le comportement de ces systèmes, notamment en termes de temps d'attente, longueur de la file, utilisation des serveurs, taux de rejet/perte.

2. Caractéristiques d'une file d'attente



Un système de file d'attente est décrit par plusieurs caractéristiques fondamentales :

1) **Un client** est l'entité qui arrive dans le système de file d'attente avec une demande de service. Le client attend d'être servi si tous les serveurs sont occupés, puis quitte le système une fois le service rendu.
2) **Un serveur** est la ressource qui fournit le service demandé par les clients. Un système de file d'attente peut avoir un ou plusieurs serveurs identiques, travaillant en parallèle. Chaque serveur a une certaine capacité de traitement, définie par son taux de service.
3) **La population source** est l'ensemble des clients potentiels pouvant générer des demandes. Elle peut être:
    - ✓ **Finie**: nombre limité de clients. Dans un modèle à population source finie, le taux d'arrivée dépend directement du nombre de clients actuellement présents dans le système (en service ou en attente).
    - ✓ **Infinie**: le nombre de clients est supposé illimité, ce qui est courant dans les modèles classiques. Dans un modèle à population source infinie, le taux d'arrivée n'est pas affecté par le nombre de clients actuellement présents dans le système.
4) **Le processus d'arrivée:** décrit la façon dont les demandes accèdent au système (*ex.* : taux

d'arrivée, loi de probabilité). Il répond à la question : Comment et à quelle fréquence les clients arrivent dans le système ? (Souvent modélisé par des processus stochastiques comme le processus de Poisson).

5) **Le processus de service:** décrit le temps nécessaire pour traiter une demande (*ex.* : taux de service, distribution des temps de service). Il répond à la question :Combien de temps le service prend-il pour chaque client ? (Souvent modélisé par des distributions de probabilité, comme la distribution exponentielle).
6) **Le nombre de serveurs:** indique combien d'unités de traitement sont disponibles pour répondre aux demandes.
7) **La capacité du système:** correspond au nombre maximal de demandes que le système peut contenir (y compris celles en attente et en service). Elle répond la question: La file peut-elle contenir un nombre infini ou limité de clients ?
8) **La discipline de file:** définit l'ordre dans lequel les demandes sont servies. Exemple:
    - ✓ FIFO / FCFS (First-In, First-Out / First-Come, First-Served) : Le client qui est arrivé le premier dans la file est le premier à être servi.
    - ✓ SJF (Shortest Job First) : Le client dont le temps de service estimé est le plus court est choisi en premier.
    - ✓ Priorité : Les clients sont classés selon des niveaux de priorité prédéfinis. Les clients de haute priorité sont servis avant ceux de basse priorité.
    - ✓ Service in Random Order (SIRO) : Les clients sont sélectionnés au hasard dans la file d'attente.
    - ✓ ...etc.
9) **Le comportement de système:** décrit les réactions possibles des clients lorsqu'ils interagissent avec le système de file d'attente, notamment face à l'attente. On distingue principalement trois types de comportements :
        - ✓ **Attente patiente** : les clients restent dans la file jusqu'à ce qu'un serveur soit disponible, quel que soit le temps d'attente.
        - ✓ **Abandon (Reneging)** : certains clients quittent la file avant d'être servis si l'attente dépasse un certain seuil (temps de patience limité).
        - ✓ **Réticence à entrer (Balking)** : les clients choisissent de ne pas entrer dans la file s'ils la jugent trop longue au moment de leur arrivée.

- ✓ **Changement de file (Jockeying)** : dans un système à files multiples (*ex.* : supermarché), les clients peuvent décider de quitter leur file actuelle pour une autre jugée plus rapide.

Exemples des files d'attentes

Exemple 1 : Files d'attente dans la vie quotidienne

Caractéristique	**Caisse de supermarché**	**Urgences hospitalières**	**Guichet Automatique Bancaire (GAB)**
Client	Client avec chariot	Patient nécessitant des soins	Client souhaitant effectuer une transaction
Serveur(s)	Caissier(s)	Médecin(s), infirmier(s)	Le GAB lui-même
File d'attente	Ligne de clients devant la caisse	Salle d'attente / patients en attente de diagnostic	Personnes en attente devant le GAB
Population	Infinie (nombre de clients potentiels très grand)	Infinie (grand nombre de patients potentiels)	Infinie (grand nombre de clients de la banque)
Discipline	Généralement FIFO (First-In, First-Out)	Priorité (urgences vitales avant tout)	FIFO
Capacité	Infinie (en pratique, limitée par l'espace)	Infinie (en pratique, limitée par l'espace)	Infinie (en pratique, limitée par l'espace)
Comportement spécifique	Changement de file possible si plusieurs files	Réticence possible si le temps d'attente est jugé trop long	Réticence si le GAB est en panne ou trop de personnes

Exemple 2: Files d'attente dans les systèmes informatiques

Caractéristique	**Serveur Web**	**Traitement des tâches dans un système d'exploitation (Ordonnanceur CPU)**	**Requêtes SQL vers une base de données**
Client	Requête HTTP d'un utilisateur / navigateur	Processus / Tâche à exécuter sur le CPU	Requête SQL (SELECT, INSERT, UPDATE, DELETE)
Serveur(s)	Le serveur web (processeur, mémoire, bande passante)	Le(s) cœur(s) du processeur (CPU)	Le moteur de la base de données
File d'attente	File de requêtes HTTP en attente de traitement	File des processus prêts (ready queue)	File des requêtes SQL en attente d'exécution
Population	Infinie (nombre d'utilisateurs d'Internet illimité)	Infinie (nombre de tâches à créer/exécuter potentiellement illimité)	Infinie (nombre de requêtes potentielles très grand)
Discipline	Généralement FIFO ou	Diverses (FIFO, Round-	FIFO ou Priorité

	Round-Robin (selon la configuration)	Robin, Priorité, SJF/SRPT, etc.)	(selon le type de requête ou l'utilisateur)
Capacité	Finie (limité par la mémoire, threads disponibles)	Finie (taille de la "ready queue" peut être limitée)	Finie (nombre max de connexions, buffer pool)
Comportement spécifique	Abondonner si la requête timeout, Balking si le serveur est surchargé	Les tâches n'abandonnent pas le CPU	Abondonner si la requête timeout, Balking si la BD est surchargée

3. Notation de Kendall

La notation de Kendall est une forme standardisée, introduit par David George Kendall en 1953, pour décrire les files d'attente en spécifiant leurs principales caractéristiques sous la forme suivante :

$$A/S/c/K/m/Z$$

N.B : La notation de base est de la forme $A/S/c$, mais elle est souvent étendue pour inclure d'autres paramètres importants.

1) A **: La loi du processus d'arrivée des clients.** Il décrit la loi de probabilité qui régit **les temps entre les arrivées** consécutives des clients (**les "inter-arrivées"**). Il peut être:
    - **M (Markovien / Memoryless / Exponentielle)** : Les temps d'inter-arrivées suivent une loi exponentielle (ou, de manière équivalente, les arrivées forment un processus de Poisson). C'est le cas le plus simple et le plus couramment étudié en raison de la propriété "sans mémoire" de la loi exponentielle.
    - **D (Déterministe)** : Les temps d'inter-arrivées sont constants (fixes).
    - **G (Général)** : Les temps d'inter-arrivées suivent une loi de probabilité générale, c'est-à-dire une distribution quelconque. Il est parfois précisé GI (Général Indépendant) pour indiquer que les temps d'inter-arrivées sont indépendants et identiquement distribués (i.i.d.).
    - **E_k (Erlang d'ordre k)** : Les temps d'inter-arrivées suivent une loi d'Erlang d'ordre k. C'est une généralisation de la loi exponentielle.
    - **H_k (Hyperexponentielle d'ordre k)** : Les temps d'inter-arrivées suivent une loi

hyperexponentielle d'ordre k.

2) S : **La loi du processus de service.** Il décrit la loi de probabilité qui régit **la durée du service** pour chaque client :
    - **M (Markovien / Memoryless / Exponentielle)** : Les temps de service suivent une loi exponentielle.
    - **D (Déterministe)** : Les temps de service sont constants.
    - **E_k (Erlang d'ordre k)** : Les temps de service suivent une loi d'Erlang d'ordre k.
    - **G (Général)** : Les temps de service suivent une loi de probabilité générale, c'est-à-dire une distribution quelconque.
3) c : **Nombre de serveurs.** Ce troisième symbole indique le nombre de serveurs (ou canaux de service) disponibles en parallèle pour traiter les clients. Il s'agit d'un entier positif (1, 2, 3, ...). Un 1 signifie un seul serveur (système à un seul canal). Un nombre supérieur à 1 signifie plusieurs serveurs (système à plusieurs canaux).
4) K : **Capacité du système (optionnel)**. Ce quatrième symbole indique la capacité maximale totale du système, y compris les clients en attente et ceux en cours de service.
    - Si $K = \infty$: Le système peut accueillir un nombre illimité de clients. C'est l'hypothèse par défaut si ce paramètre est omis.
    - Si K est un entier positif : Le système ne peut contenir qu'un nombre limité de clients. Si un client arrive alors que le système est plein, il est bloqué ou perdu.
5) m : **Taille de la population source (optionnel)**. Il indique la taille de la population totale de clients potentiels qui peuvent générer des arrivées. Par défaut, $m = \infty$ (la population de clients est illimitée). Si m est un entier positif, la population est finie.
6) Z : **Discipline de service (optionnel)**. Il spécifie l'ordre dans lequel les clients sont sélectionnés de la file d'attente pour être servis : FIFO, SJF, Priorité, SIRO, ...etc. Il est souvent omis si la discipline est en FIFO.

Exemple des modèles courants de files d'attente:

- ✓ M/M/1 : un seul serveur avec des arrivées et des services suivant une distribution exponentielle.
- ✓ M/M/c : plusieurs serveurs avec des arrivées et des services suivant une distribution exponentielle.

- ✓ M/G/1 : un seul serveur avec une distribution d'arrivée exponentielle et une distribution de service générale.

4. Évaluation de performance d'un système de file d'attente

La théorie des files d'attente travaille sur un ensemble de métriques de performance clés pour évaluer l'efficacité et la qualité de service d'un système en se basant sur les paramètres d'entrée.

4.1. Paramètres d'entrée du système

Ces paramètres décrivent **la configuration du système** et les caractéristiques de son environnement. En plus des paramètres mentionnés dans la notation de Kendall (distribution des temps d'inter-arrivée, distribution de la durée de service, nombre de serveurs, *etc.*), un système de file d'attente est principalement caractérisé par deux paramètres fondamentaux λ et μ:

- λ : **Taux d'arrivés**. Il représente le nombre moyen de clients qui arrivent dans le système par unité de temps.
- μ : **Taux de service.** Il représente le nombre moyen de clients qu'un **seul serveur** peut traiter par unité de temps.

Note : Dans d'autres contextes, λ est remplacé par λ_e le «**taux effectif d'arrivés**». λ_e tient compte des clients perdus (rejet, abandon) et représente ainsi, le nombre moyen de clients qui entrent **réellement** dans le système par unité de temps.

4.2. Paramètre de charge

Noté par ρ et appelé **facteur d'utilisation** ou **intensité de trafic.** Il représente la proportion du temps pendant lequel le serveur est occupé à rendre service. Pour un serveur unique, $\rho = \lambda/\mu$.

4.3. Mesures de performance

Ces mesures permettent d'évaluer **l'efficacité et la qualité de service** du système. Elles résultent de l'analyse du modèle à partir des paramètres d'entrée:

- π_0 : **Probabilité que la file d'attente soit vide**. C'est la probabilité qu'il n'y ait aucun client dans le système (ou bien la probabilité que le serveur ou tous les serveurs soient inactifs).
- $\pi_n(t)$: **Probabilité qu'il y ait n clients dans le système**. C'est la probabilité que, à un instant

donné t, il y ait exactement n clients dans le système (en attente et en service).

- P_w: **Probabilité d'attente.** La probabilité qu'un client arrivant doive attendre avant d'être servi.
- L_Q: **Nombre moyen de clients dans la file d'attente (longueur de la queue)**. C'est le nombre moyen de clients qui sont uniquement en attente..
- L_S: **Nombre moyen de clients** en train d'être servis (nombre moyen de serveurs occupés).
- $L = L_Q + L_S$: **Nombre moyen de clients dans le système**. C'est le nombre moyen de clients présents dans l'ensemble du système, y compris ceux qui sont en attente dans la file et ceux qui sont en cours de service.
- W_Q: **Temps moyen d'attente dans la file**. C'est le temps moyen qu'un client passe à attendre avant que son service ne commence.
- W_S: **Temps moyen de service.** C'est le temps moyen qu'il faut à un serveur pour traiter un client.
- $W = W_Q + W_S$: **Temps moyen de séjour dans le système**. C'est le temps total moyen qu'un client passe dans le système, de son arrivée jusqu'à la fin de son service. Il inclut le temps d'attente et le temps de service.

D'autres mesures spécifiques sont également très importantes dans certains contextes particuliers de files d'attente. Par exemple:

- **Probabilité de rejet**: Dans les systèmes à capacité finie, c'est la probabilité qu'un client arrivant **soit refusé** car la capacité maximale du système est atteinte.
- **Probabilité de défection (abandon)**: Dans les systèmes où les clients peuvent abandonner la file, c'est la probabilité qu'un client quitte la file avant d'être servi. *Exemple d'utilisation*: Systèmes où les clients sont impatients, souvent modélisés avec **un temps de patience**.

5. Applications pratiques

- ***Dimensionnement:*** Déterminer le nombre optimal de serveurs, équilibrer coût de service et coût d'attente, prévoir les besoins en capacité
- ***Optimisation:*** Minimiser les temps d'attente, maximiser l'utilisation des ressources, améliorer la qualité de service
- ***Évaluation de performance:*** Comparer différentes configurations, analyser l'impact des variations de demande, établir des accords de niveau de service.

6. La loi de Little et les files d'attentes

La loi de Little est une relation fondamentale en théorie des files d'attente, qui relie trois métriques clés dans un système stable. Elle s'applique à tout système en régime permanent (avec $\rho < 1$) et ne dépend pas de la distribution des arrivées ou des services. Elle reste valable quelle que soit la discipline de service (FIFO, LIFO, etc.) ou le nombre de serveurs.

$$L = \lambda \cdot W$$

Où :

- L : nombre moyen de clients dans le système,
- λ : taux moyen d'arrivée,
- W : temps moyen passé par un client dans le système.

Note : La loi de Little s'applique aussi bien à la file d'attente seule ($L_q = \lambda \cdot W_q$) qu'au service ($L_S = \lambda \cdot W_S$).

Exemple:

Si un serveur web reçoit en moyenne 20 requêtes par seconde ($\lambda = 20$) et que le temps moyen de traitement est de 0.1 seconde (W = 0.1), alors : L=20×0.1=2 → En moyenne, 2 requêtes sont présentes dans le système à tout moment.

7. Les modèles markoviennes courantes des files d'attentes

7.1. Le modèle $M/M/1$

Le modèle $\boldsymbol{M/M/1}$ (ou encore $M/M/1/\infty/FIFO$) est l'un des modèles de files d'attente les plus fondamentaux et les plus étudiés dans la théorie des files d'attente. Il décrit un système à **un seul serveur**, dans lequel les arrivées et les temps de service suivent **des lois exponentielles** (processus sans mémoire), et les clients sont servis selon la discipline **FIFO** (First-In, First-Out).

7.1.1. Hypothèses du modèle $M/M/1$

1) Les clients arrivent selon un processus de **Poisson** de taux λ ; le temps moyen pour qu'une nouvelle arrivée se produise est $1/\lambda$; le temps entre deux arrivées (inter-arrivés) suit une

distribution exponentielle.

2) Le temps de service suit une distribution **Exponentielle** de taux μ (durée moyenne du service = $1/\mu$).
3) Le système comporte **un seul serveur**.
4) La file d'attente est **infinie** (pas de limite de capacité).
5) Les clients sont servis dans l'ordre d'arrivée (**FIFO**).
6) La population source est **infinie**.
7) Le système est **stationnaire** si $\rho = \lambda/\mu < 1$.

7.1.2. Modélisation du $M/M/1$

Le modèle $M/M/1$ peut être représenté par une **Chaîne de Markov à Temps Continu**, où l'état du système à un instant donné est défini par **le nombre de clients dans le système** (file + service).

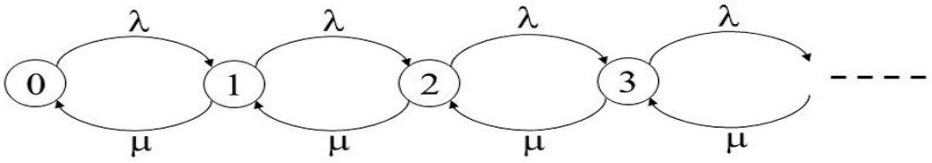
L'espace des états est: $S = \{0, 1, 2, 3, \ldots\}$ (espace infini) où $n = 0$ représente l'état d'un système vide (aucun client n'est présent), et $n >= 1$ représente l'état de système avec n clients présents (en service ou en attente).

Le processus de transition suit la structure d'**un processus de naissance et de mort**, avec :

- **Taux de naissance** $\lambda_n = \lambda$, pour tout $n \geq 0$ $(n = 0, 1, 2, 3, \ldots.)$: le taux d'arrivée est constant quelque soit l'état de système.
- **Taux de mort** $\mu_n = \mu$, pour tout $n \geq 1$ $(n = 1, 2, 3, \ldots.)$, et $\mu_0 = 0$: le taux de service est constant lorsque le serveur est occupé. Aucune service lorsque le système est vide.

Par conséquent, les **transitions** sont :

- De l'état n à l'état $n + 1$ (arrivée d'un client) avec un taux λ, pour tout $n \geq 0$.
- De l'état n à l'état $n - 1$ (fin de service) avec un taux μ, pour tout $n \geq 1$.



La **matrice de taux de transition infinitésimale** Q associée à cette chaîne de Markov a la forme

tridiagonale **infinie** suivante :

$$Q = \begin{pmatrix} -\lambda & \lambda & 0 & 0 & \dots \\ \mu & -(\lambda+\mu) & \lambda & 0 & \dots \\ 0 & \mu & -(\lambda+\mu) & \lambda & \dots \\ 0 & 0 & \mu & -(\lambda+\mu) & \dots \\ : & : & 0 & \mu & \dots \\ : & : & : & 0 & \dots \end{pmatrix}$$

7.1.3. Calcul des probabilités d'états stationnaires

Pour étudier le comportement à long terme du système, on s'intéresse aux probabilités stationnaires, qui décrivent la répartition stable des états lorsque le système atteint un régime **permanent** (ou régime stationnaire).

Soit $\pi_n(t)$ la probabilité qu'il y ait n clients dans le système (dans la file d'attente ou en cours de service) au temps t.

Le processus $M/M/1$ atteint un état stationnaire si $\lambda < \mu$ (c'est-à-dire $\rho < 1$). À ce stade, $\pi_n(t)$ ne dépend plus de t et on le note simplement π_n.

Rappelant que, pour un processus de naissance-mort général avec des taux de naissance : $\lambda_0, \lambda_1, \lambda_2, \dots$; et des taux de mort : $\mu_1, \mu_2, \mu_3, \dots$, la distribution π_n au régime stationnaire est donnée par :

$$\pi_n = \pi_0 * \prod_{k=1}^{n} \frac{\lambda_{k-1}}{\mu_k} \qquad pour\ n \geq 1$$

Et π_0 est obtenu par normalisation:

$$\pi_0 = \left(1 + \sum_{n=1}^{\infty} \prod_{k=1}^{n} \frac{\lambda_{k-1}}{\mu_k}\right)^{-1}$$

Dans un modèle M/M/1, $\lambda_n = \lambda$ et $\mu_n = \mu$, pour tout $n \geq 1$:

$$\pi_n = \pi_0 * \prod_{k=1}^{n} \frac{\lambda_{k-1}}{\mu_k} = \pi_0 * \frac{\lambda_0 * \lambda_1 * \dots * \lambda_{n-1}}{\mu_1 * \mu_2 * \dots * \mu_n} = \pi_0 * (\lambda^n)/(\mu^n) = \pi_0 * (\frac{\lambda}{\mu})^n$$

Soit : $\rho = \lambda/\mu$:

$$\boxed{\boldsymbol{\pi_n = \rho^n * \pi_0}}$$

ρ s'appelle l'**intensité de trafic** ou **facteur d'utilisation**.

π_0 est obtenu par normalisation comme suit:

$$\begin{cases} \pi_n = \rho^n * \pi_0 \\ \sum_{n=0}^{\infty} \pi_n = 1 \end{cases}$$

Ainsi:

$$\sum_{n=0}^{\infty} \pi_n = \sum_{n=0}^{\infty} \rho^n * \pi_0 = \pi_0 * \sum_{n=0}^{\infty} \rho^n = 1$$

$\sum_{n=0}^{\infty} \rho^n$ est une série géométrique qui converge lorsque $\rho < 1$:

$$\sum_{n=0}^{\infty} \rho^n = \frac{1}{1-\rho}$$

Par conséquent :

$$\pi_0 * \frac{1}{1-\rho} = 1$$

Donc :

$$\boxed{\boldsymbol{\pi_0 = 1 - \rho}}$$

$$\boxed{\boldsymbol{\pi_n = \rho^n * \pi_0 = (1-\rho) * \rho^n} \qquad \boldsymbol{pour\ n = 1, 2, 3, \ldots.}}$$

7.1.4. Calcul des mesures de performance

- **Nombre moyen de clients dans le système L:**

Le nombre moyen de clients dans le système, noté L, est **l'espérance mathématique** du nombre de clients présents dans le système (ceux en service et ceux en attente).

$$L = E(N) = \sum_{n=0}^{\infty} n * \pi_n = \sum_{n=0}^{\infty} n * (1-\rho) * \rho^n = (1-\rho) \sum_{n=0}^{\infty} n * \rho^n$$

En remplaçant le deuxième terme de l'expression ci-dessus par la formule de la série géométrique:

$$\sum_{n=0}^{\infty} n * x^n = \frac{x}{(1-x)^2} \qquad pour\ |x| < 1$$

On obtient finalement:

$$L = (1-\rho) * \frac{\rho}{(1-\rho)^2} = \boldsymbol{\frac{\rho}{1-\rho}}$$

- **Nombre moyen de clients entrain d'être servis L_S:**

Dans un modèle M/M/1 avec un seul serveur, le serveur est soit inactif (système vide, probabilité π_0), soit occupé (système non vide, probabilité $1 - \pi_0$). Le nombre moyen de clients en service est

précisément la probabilité que le serveur soit occupé.

$$L_S = 1 - \pi_0 = 1 - (1 - \rho) = \boldsymbol{\rho}$$

- **Nombre moyen de clients dans la file d'attente L_Q:**

$$L_Q = L - L_S = \frac{\rho}{1-\rho} - \rho = \frac{\boldsymbol{\rho^2}}{\boldsymbol{1-\rho}}$$

- **Temps moyen de séjour dans le système W**

Le temps moyen de séjour dans le système est obtenu à partir de la formule de Little : $L = \lambda \cdot W$

$$W = \frac{L}{\lambda} = \frac{\frac{\rho}{1-\rho}}{\lambda} = \frac{\boldsymbol{1}}{\boldsymbol{\mu - \lambda}}$$

- **Temps moyen de service W_S**

Si un serveur peut servir en moyenne μ clients par unité de temps, alors le temps moyen qu'il faut pour servir un seul client est l'inverse de ce taux.

En plus, pour les modèles markoviennes, les temps de service suivent une distribution exponentielle. Une propriété clé de la distribution exponentielle est que sa moyenne est $1/\mu$.

Vérifions cette propriété par la Loi de Little : $L_S = \lambda \cdot W_S$

$$W_S = \frac{L_S}{\lambda} = \frac{\rho}{\lambda} = \frac{\lambda/\mu}{\lambda} = \frac{\boldsymbol{1}}{\boldsymbol{\mu}}$$

- **Temps moyen d'attente dans la file W_Q**

Puisque: $L_Q = \lambda \cdot W_Q$:

$$W_Q = \frac{L_Q}{\lambda} = \frac{\frac{\rho^2}{1-\rho}}{\lambda} = \frac{\boldsymbol{\rho}}{\boldsymbol{\mu(1-\rho)}}$$

7.2. Le modèle $M/M/c$

Le modèle $\boldsymbol{M/M/c}$ (ou encore $M/M/c/\infty/FIFO$) est une extension du modèle $M/M/1$ à plusieurs serveurs. Il représente un système de file d'attente comportant c serveurs **identiques** en parallèle, chacun capable de traiter un client à la fois. Ce modèle est couramment utilisé pour représenter les systèmes où plusieurs unités de traitement (CPU, guichets, files de processus, etc.) partagent une file d'attente commune.

7.2.1. Hypothèses du modèle $M/M/c$

1) Les clients arrivent selon un processus de **Poisson** de taux λ.
2) Le temps de service suit une distribution **Exponentielle** de taux μ.
3) Le système comporte c **serveurs** fonctionnant en parallèle et indépendamment.
4) La capacité du système est illimitée (par défaut).
5) Les clients sont servis dans l'ordre d'arrivée (**FIFO**).
6) La population source est **infinie**.
7) Le système est **stationnaire** si $\rho_s = \lambda/c * \mu < 1$.

7.2.2. Modélisation du $M/M/c$

Le modèle $M/M/c$ peut être représenté par une **Chaîne de Markov à Temps Continu**, où l'état du système à un instant donné est défini par **le nombre de clients dans le système** (file + service).

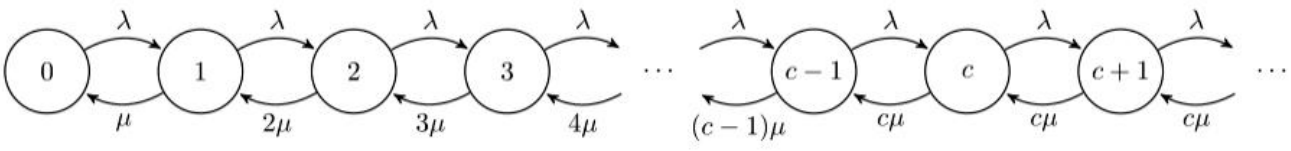
Le processus de transition suit la structure d'**un processus de naissance et de mort généralisé**, avec :

- **Taux de naissance** $\lambda_n = \lambda$, pour tout $n \geq 0$ ($n = 0, 1, 2, 3, \ldots$).
- **Taux de mort** (départ d'un client) : les départs ont un taux croissant jusqu'à $c * \mu$, puis constant:

$$\mu_n = \boldsymbol{Min(n, c)} * \boldsymbol{\mu} = \begin{cases} n * \mu & si \quad 1 \leq n < c \\ c * \mu & si \quad n \geq c \\ 0 & si \quad n = 0 \end{cases}$$

Autrement dit :

- ✓ Si le nombre de clients est inférieur à c, chaque client est servi indépendamment avec un taux μ,
- ✓ Si le nombre de clients est supérieur ou égal à c, tous les serveurs sont occupés et le taux total de service reste constant à $c * \mu$ (c clients en service, $(n - c)$ clients en attente).



La matrice de taux de transition Q a une structure tridiagonale infinie :

$$Q = \begin{pmatrix} -\lambda & \lambda & 0 & 0 & 0 & & & \cdots \\ \mu & -(\lambda+\mu) & \lambda & 0 & 0 & & & \cdots \\ 0 & 2\mu & -(\lambda+2\mu) & \lambda & 0 & & & \cdots \\ : & : & : & : & \lambda & & & \cdots \\ : & : & 0 & : & -(\lambda+c\mu) & \lambda & & \cdots \\ : & : & : & 0 & c\mu & -(\lambda+c\mu) & \lambda & \cdots \\ & & & & & c\mu & -(\lambda+c\mu) & \cdots \\ & & & & & & c\mu & \cdots \end{pmatrix}$$

7.2.3. Calcul des probabilités d'états stationnaires

Le processus du système $M/M/c$ est stationnaire si $\lambda/c * \mu < 1$. Rappelant que, pour un processus de naissance-mort général avec les taux de naissance : $\lambda_0, \lambda_1, \lambda_2$, ...et les taux de mort : μ_1, μ_2, μ_3, ... :

$$\pi_n = \pi_0 * \prod_{k=0}^{n-1} \frac{\lambda_k}{\mu_{k+1}} \qquad pour\ n \geq 1$$

Or, dans un modèle $M/M/c$:

- ✓ $\lambda_n = \lambda$ pour tout $n \geq 0$ (taux d'arrivée constant)
- ✓ $\mu_n = n\mu$, pour tout $n \leq c$ (n serveurs actifs)
- ✓ $\mu_n = c\mu$, pour tout $n > c$ (c serveurs actifs maximum)

Cas 1: Pour $n \leq c$

$$\pi_n = \pi_0 * \prod_{k=0}^{n-1} \frac{\lambda_k}{\mu_{k+1}} = \pi_0 * \prod_{k=0}^{n-1} \frac{\lambda}{(k+1)\mu} = \pi_0 * \prod_{k=0}^{n-1} \frac{\lambda}{\mu} * \frac{1}{(k+1)} = \pi_0 * \left(\frac{\lambda}{\mu}\right)^n * \frac{1}{1*2*3*\ldots*n}$$

$$= \boldsymbol{\pi_0 * \frac{\rho^n}{n!}}$$

Cas 2: Pour $n > c$

$$\pi_n = \pi_0 * \prod_{k=0}^{c-1} \frac{\lambda}{(k+1)\mu} * \prod_{k=c}^{n-1} \frac{\lambda}{c\mu} = \pi_0 * \frac{\rho^c}{c!} * \left(\frac{\lambda}{c\mu}\right)^{n-c} = \pi_0 * \frac{\rho^c}{c!} * \left(\frac{\rho}{c}\right)^{n-c} = \boldsymbol{\pi_0 * \frac{\rho^n}{c! * c^{n-c}}}$$

Les équations de récurrence sont donc (Avec : $\rho = \lambda/\mu$):

$$\boldsymbol{\pi_n = \begin{cases} \frac{\rho^n}{n!} * \pi_0 & si\ 0 \leq n < c \\ \frac{\rho^n}{c! * c^{n-c}} * \pi_0 & si\ n \geq c \end{cases}}$$

Le terme π_0 (probabilité que le système soit vide) est obtenu par normalisation :

$$\begin{cases} \pi_n = \dfrac{\rho^n}{n!} * \pi_0 & si\ 0 \leq n < c \\ \pi_n = \dfrac{\rho^n}{c! * c^{n-c}} * \pi_0 & si\ n \geq c \\ \sum_{n=0}^{\infty} \pi_n = 1 \end{cases}$$

Ainsi:

$$\sum_{n=0}^{\infty} \pi_n = \sum_{n=0}^{c-1} \frac{\rho^n}{n!} * \pi_0 + \sum_{n=c}^{\infty} \frac{\rho^n}{c! * c^{n-c}} * \pi_0 = 1$$

$$\sum_{n=0}^{\infty} \pi_n = \pi_0 * \left[\sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \sum_{n=c}^{\infty} \frac{\rho^c . \rho^{n-c}}{c! * c^{n-c}}\right] = \pi_0 * \left[\sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \frac{\rho^c}{c!} * \sum_{n=c}^{\infty} \frac{\rho^{n-c}}{c^{n-c}}\right]$$

En posant $k = n - c$: (si $n = c$ alors $k = 0$ et quand $n \to \infty,\ k \to \infty$)

$$\pi_0 * \left[\sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \frac{\rho^c}{c!} * \sum_{n=c}^{\infty} \frac{\rho^{n-c}}{c^{n-c}}\right] = \pi_0 * \left[\sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \frac{\rho^c}{c!} * \sum_{k=0}^{\infty} (\frac{\rho}{c})^k\right] = 1$$

La deuxième somme est une série géométrique de la forme:

$$\sum_{k=0}^{\infty} x^k = \frac{1}{1-x}$$

En substituant dans l'expression précédente:

$$\pi_0 * \left[\sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \frac{\rho^c}{c!} * \frac{1}{1-\frac{\rho}{c}}\right] = \pi_0 * \left[\sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \frac{\rho^c}{c!} * \frac{c}{c-\rho}\right] = 1$$

Le résultat final est :

$$\boxed{\boldsymbol{\pi_0 = \left[\sum_{n=0}^{c-1} \frac{\rho^n}{n!} + (\frac{\rho^c}{c!} \cdot \frac{c}{c-\rho})\right]^{-1}}}$$

- **Probabilité d'attente (Formule d'Erlang C)**

La formule d'Erlang C, notée P_w ou $C(c, \rho)$ exprime la probabilité qu'un client arrivant trouve tous les serveurs occupés et **doive donc attendre dans la file**.

Un client attend si et seulement si le système a $n \geq c$ clients. Par conséquent:

$$P_w = P(n \geq c) = \sum_{n=c}^{\infty} P_n = \sum_{n=c}^{\infty} \frac{\rho^n}{c! * c^{n-c}} * \pi_0 = \sum_{n=c}^{\infty} \frac{\rho^c * \rho^{n-c}}{c! * c^{n-c}} * \pi_0 = \frac{\rho^c}{c!} * \pi_0 * \sum_{n=c}^{\infty} (\frac{\rho}{c})^{n-c}$$

En posant $k = n - c$:

$$P_w = \frac{\rho^c}{c!} * \pi_0 * \sum_{k=0}^{\infty} (\frac{\rho}{c})^k$$

La deuxième somme est une série géométrique de la forme:

$$\sum_{k=0}^{\infty} x^k = \frac{1}{1-x}$$

Alors :

$$P_w = \frac{\rho^c}{c!} * \pi_0 * \sum_{k=0}^{\infty} (\frac{\rho}{c})^k = \frac{\rho^c}{c!} * \pi_0 * \frac{1}{1-\frac{\rho}{c}} = \frac{\rho^c}{c!} * \pi_0 * \frac{c}{c-\rho}$$

Le résultat final :

$$\boxed{\boldsymbol{P_w = C(c,\rho) = \frac{\rho^c}{(c-1)! * (c-\rho)} * \pi_0}}$$

7.2.4. Calcul des mesures de performance

- **Nombre moyen de clients dans la file d'attente** L_Q

Un client est dans la file d'attente si le nombre total de clients dans le système, n, est supérieur au nombre de serveurs c. Si $n \leq c$, tous les clients sont en service et il n'y a personne en file d'attente.

Pour un état où n clients sont dans le système ($n > c$), le nombre de clients en file d'attente est $n - c$.

$$L_Q = E(L_Q) = \sum_{n=c}^{\infty} (n-c) * \pi_n = \sum_{n=c}^{\infty} (n-c) * \frac{\rho^n}{c! * c^{n-c}} * \pi_0 = \sum_{n=c}^{\infty} (n-c) * \frac{\rho^c * \rho^{n-c}}{c! * c^{n-c}} * \pi_0$$

$$= \frac{\rho^c}{c!} * \pi_0 * \sum_{n=c}^{\infty} (n-c) * \frac{\rho^{n-c}}{c^{n-c}} = \frac{\rho^c}{c!} * \pi_0 * \sum_{n=c}^{\infty} (n-c) * (\frac{\rho}{c})^{n-c}$$

En posant $k = n - c$: (le terme $k = 0$ contribue par 0 à la somme)

$$L_Q = \frac{\rho^c}{c!} * \pi_0 * \sum_{k=0}^{\infty} k * (\frac{\rho}{c})^k = \frac{\rho^c}{c!} * \pi_0 * \sum_{k=1}^{\infty} k * (\frac{\rho}{c})^k$$

La deuxième somme est une série géométrique de la forme:

$$\sum_{k=1}^{\infty} k * x^k = \frac{x}{(1-x)^2} \qquad pour \ |x| < 1$$

Avec $x = \rho/c < 1$ (condition de stabilité) :

$$\sum_{k=1}^{\infty} k * (\frac{\rho}{c})^k = \frac{\frac{\rho}{c}}{(1-\frac{\rho}{c})^2} = \frac{\rho}{c} * \frac{c^2}{(c-\rho)^2} = \frac{\rho c}{(c-\rho)^2}$$

Le résultat final sera:

$$L_Q = \frac{\rho^c}{c!} * \pi_0 * \frac{\rho c}{(c-\rho)^2}$$

Or :

$$C(c,\rho) = \frac{\rho^c}{c!} * \frac{c}{c-\rho} * \pi_0$$

Donc :

$$L_Q = \boldsymbol{C(c,\rho)} * \frac{\boldsymbol{\rho}}{\boldsymbol{c-\rho}}$$

- **Nombre moyen de clients dans le service** L_S

La valeur de L_S dans un modèle M/M/c peut être calculée de manière très directe en utilisant le taux d'arrivée et le taux de service :

$$L_S = \frac{\lambda}{\mu} = \boldsymbol{\rho}$$

- **Nombre moyen de clients dans le système** L

$$L = \rho + C(c,\rho) * \frac{\rho}{c-\rho} = \boldsymbol{\rho} * \left(\mathbf{1} + \frac{\boldsymbol{C(c,\rho)}}{\boldsymbol{c-\rho}}\right)$$

- **Temps moyen de séjour dans le système** W

De la formule de Little : $L = \lambda \cdot W$ et en substituant ρ par sa valeur λ/μ

$$W = \frac{1}{\lambda} * L = \frac{1}{\lambda} * \frac{\lambda}{\mu} * \left(1 + \frac{C(c,\rho)}{c-\rho}\right) = \frac{1}{\mu} + C(c,\rho) * \frac{1}{\mu(c-\rho)} = \frac{\mathbf{1}}{\boldsymbol{\mu}}\left(\mathbf{1} + \frac{\boldsymbol{C(c,\rho)}}{\boldsymbol{c-\rho}}\right)$$

- **Temps moyen de service** W_S

De la formule de Little : $L_S = \lambda \cdot W_S$

$$W_S = \frac{L_S}{\lambda} = \frac{\rho}{\lambda} = \frac{\lambda/\mu}{\lambda} = \frac{\mathbf{1}}{\boldsymbol{\mu}}$$

- **Temps moyen d'attente (queue)** W_Q

En utilisant la formule de Little: $L_Q = \lambda \cdot W_Q$

$$W_Q = \frac{\boldsymbol{C(c,\rho)}}{\boldsymbol{\mu(c-\rho)}}$$

7.3. Le modèle $M/M/1/K$

Le modèle $M/M/1/K$ représente un système de file d'attente **à capacité limitée**, dans lequel les arrivées suivent un processus de Poisson, les temps de service sont exponentiels, et **un seul serveur** est disponible pour traiter les demandes. Le système ne peut contenir qu'un nombre maximal de K clients, y compris celui en cours de service. Ainsi, lorsqu'un client arrive et que le système est déjà plein (Kclients présents), il est automatiquement **rejeté**.

Ce modèle est particulièrement adapté à la modélisation de systèmes à ressources contraintes, où les arrivées peuvent être perdues en cas de saturation, comme les files d'attente dans les buffers réseau ou les connexions simultanées limitées dans un serveur.

7.3.1. Hypothèses du modèle $M/M/1/K$

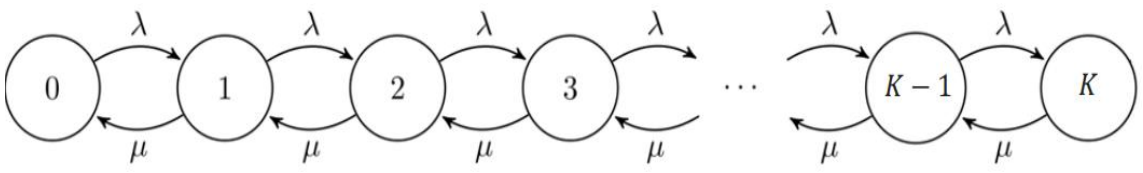
1) Les clients arrivent selon un processus de **Poisson** de taux λ.
2) Le temps de service suit une distribution **Exponentielle** de taux μ.
3) Le système comporte **un seul serveur**.
4) La capacité du système **est limitée**. Le système peut contenir **au plus K clients**.
5) Les clients sont généralement servis dans l'ordre d'arrivée (**FIFO**).
6) Si un client arrive lorsque le système contient déjà K clients, il est rejeté (bloqué).
7) Le régime est stationnaire quelle que soit la valeur de l'intensité de trafic ρ.

7.3.2. Modélisation du $M/M/1/K$

Le modèle $M/M/1/K$ est modélisé par une **Chaîne de Markov à Temps Continu** à $K+1$ états: $n = 0, 1, 2, \ldots, K$, avec :

$$\lambda_n = \begin{cases} \lambda & si \;\; n < K \\ 0 & si \;\; n \geq K \end{cases} \qquad \mu_n = \begin{cases} 0 & si \;\; n = 0 \\ \mu & si \;\; n \geq 1 \end{cases}$$

Le système suit **un processus de naissance-mort tronqué**, avec **rejet des arrivées** à l'état K. Le taux effectif d'arrivée est donc **réduit** à cause du **blocage** dans l'état saturé.



La matrice de taux de transition Q du modèle $M/M/1/K$ a une structure tridiagonale de

dimension $(K+1)\times(K+1)$:

$$Q=\begin{pmatrix} -\lambda & \lambda & 0 & .. & 0 & 0 \\ \mu & -(\lambda+\mu) & \lambda & .. & 0 & 0 \\ 0 & \mu & -(\lambda+\mu) & .. & \lambda & 0 \\ : & 0 & \mu & : & : & : \\ 0 & : & : & .. & -(\lambda+\mu) & \lambda \\ 0 & 0 & 0 & .. & \mu & -\mu \end{pmatrix}$$

7.3.3. **Calcul des probabilités d'états stationnaires**

$$\pi_n=\pi_0*\prod_{k=0}^{n-1}\frac{\lambda_k}{\mu_{k+1}} \qquad pour\ 1\le n\le K$$

$$\pi_n=\pi_0*\prod_{k=1}^{n}\frac{\lambda_{k-1}}{\mu_k}=\pi_0*\frac{\lambda_0*\lambda_1*\ldots.*\lambda_{n-1}}{\mu_1*\mu_2*\ldots.*\mu_n}=\pi_0*(\lambda^n)/(\mu^n)=\pi_0*(\frac{\lambda}{\mu})^n$$

Alors :

$$\boxed{\boldsymbol{\pi_n=\rho^n*\pi_0}} \qquad n=1,2,3,\ldots\ldots,K$$

π_0 est obtenu par normalisation comme suit:

$$\begin{cases}\pi_n=\rho^n*\pi_0\\ \sum_{n=0}^{K}\pi_n=1\end{cases}$$

Ainsi:

$$\sum_{n=0}^{K}\pi_n=\sum_{n=0}^{K}\rho^n*\pi_0=\pi_0*\sum_{n=0}^{K}\rho^n=1$$

***Cas 1:* $\boldsymbol{\rho\neq 1}$**

$$\sum_{n=0}^{K}\rho^n=\frac{1-\rho^{K+1}}{1-\rho}$$

Par conséquent :

$$\pi_0*\frac{1-\rho^{K+1}}{1-\rho}=1$$

Et donc :

$$\boxed{\boldsymbol{\pi_0=\frac{1-\rho}{1-\rho^{K+1}}}}$$

***Cas 2:* $\boldsymbol{\rho=1}$**

$$\sum_{n=0}^{K} \rho^n = \sum_{n=0}^{K} 1^n = 1^0 + 1^1 + 1^2 + \ldots + 1^K = K + 1$$

Donc :

$$\pi_0 * (K+1) = 1 \quad \Leftrightarrow \quad \boxed{\boldsymbol{\pi_0 = \frac{1}{K+1}}}$$

En résumé, les probabilités stationnaires du modèle M/M/1/K sont :

$$\boldsymbol{\pi_0} = \begin{cases} \dfrac{\mathbf{1-\rho}}{\mathbf{1-\rho^{K+1}}} & \boldsymbol{si}\ \rho \neq 1 \\ \dfrac{\mathbf{1}}{\mathbf{K+1}} & \boldsymbol{si}\ \rho = 1 \end{cases} \qquad \boldsymbol{\pi_n} = \begin{cases} \boldsymbol{\rho^n * \pi_0} & si\ 1 \leq n \leq K \\ \mathbf{0} & sinon \end{cases}$$

7.3.4. Calcul des mesures de performance

- **La probabilité de rejet P_K**

La probabilité de rejet (ou probabilité de perte) P_K est la probabilité qu'un client arrivant trouve le système plein (c'est-à-dire qu'il y ait déjà K clients dans le système, incluant celui en service). Dans ce cas, le client est refusé et ne rejoint ni la file ni le service.

$$P_K = P(n = K) = \pi_K = \rho^K * \pi_0$$

Alors :

$$\boldsymbol{P_K} = \begin{cases} \dfrac{\mathbf{1-\rho}}{\mathbf{1-\rho^{K+1}}} * \boldsymbol{\rho^K} & \boldsymbol{si}\ \rho \neq 1 \\ \dfrac{\mathbf{1}}{\mathbf{K+1}} & \boldsymbol{si}\ \rho = 1 \end{cases}$$

- **Taux d'arrivée effectif λ_e**

Le **taux d'arrivée effectif** λ_e est le taux moyen d'entrée réelle dans le système (en excluant les rejets).

Un client est accepté seulement si le système n'est pas plein. La probabilité d'acceptation est :$(1 - P_K)$ et λ_e est donc:

$$\boldsymbol{\lambda_e = \lambda * (1 - P_K)}$$

- **Nombre moyen de clients dans le système L:**

$$L = E(N) = \sum_{n=0}^{K} n * \pi_n$$

Pour $\boldsymbol{\rho \neq 1}$

$$L = \sum_{n=0}^{K} n * \pi_0 * \rho^n = \pi_0 * \sum_{n=1}^{K} n * \rho^n$$

Pour calculer $\sum_{n=1}^{K} n * \rho^n$, utilisons la dérivée de la série géométrique :

$$\frac{d}{d\rho} \sum_{n=0}^{K} n * \rho^n = \sum_{n=1}^{K} n * \rho^{n-1} = \rho * \frac{d}{d\rho}\left(\frac{1-\rho^{K+1}}{1-\rho}\right) = \rho * \frac{-(K+1)\rho^K(1-\rho) - (1-\rho^{K+1})(-1)}{(1-\rho)^2}$$

$$= \rho * \frac{-(K+1)\rho^K + (K+1)\rho^{K+1} + 1 - \rho^{K+1}}{(1-\rho)^2} = \rho * \frac{1-(K+1)\rho^K + K\rho^{K+1}}{(1-\rho)^2}$$

Donc :

$$L = \pi_0 * \rho * \frac{1-(K+1)\rho^K + K\rho^{K+1}}{(1-\rho)^2} = \rho * \frac{1-\rho}{1-\rho^{K+1}} * \frac{1-(K+1)\rho^K + K\rho^{K+1}}{(1-\rho)^2}$$

$$\boldsymbol{L = \frac{\rho(1-(K+1)\rho^K + K\rho^{K+1})}{(1-\rho^{K+1})(1-\rho)} = \frac{\rho}{1-\rho} - \frac{(K+1)\rho^{K+1}}{1-\rho^{K+1}}}$$

Pour $\boldsymbol{\rho = 1}$

$$L = E(N) = \sum_{n=0}^{K} n * \frac{1}{K+1} = \frac{1}{K+1} * \sum_{n=0}^{K} n = \frac{1}{K+1} * (0+1+2+\ldots+K) = \frac{1}{K+1} * \frac{K(K+1)}{2}$$

$$\boldsymbol{L = \frac{K}{2}}$$

Donc :

$$L = \begin{cases} \dfrac{\rho}{1-\rho} - \dfrac{(K+1)\rho^{K+1}}{1-\rho^{K+1}} & si\ \rho \neq 1 \\ \dfrac{K}{2} & si\ \rho = 1 \end{cases}$$

- **Nombre moyen de clients entrain d'être servis L_S:**

$$L_S = \frac{\lambda_e}{\mu} = \frac{\lambda * (1-P_K)}{\mu} = \rho * (1-P_K)$$

Note : On peut la déduire aussi avec la formule : $L_S = 1 - \pi_0$

- **Nombre moyen de clients dans la file d'attente L_Q:**

$$L_Q = L - (1 - \pi_0)$$

$$L_Q = \begin{cases} \dfrac{\boldsymbol{\rho}}{\boldsymbol{1-\rho}} - \dfrac{\boldsymbol{\rho(K\rho^K + 1)}}{\boldsymbol{1-\rho^{K+1}}} & si\ \rho \neq 1 \\ \dfrac{\boldsymbol{K(K-1)}}{\boldsymbol{2(K+1)}} & si\ \rho = 1 \end{cases}$$

- **Temps moyen de séjour dans le système W**:

En utilisant la Loi de Little : $L = \lambda_e \cdot W$

$$W = \frac{L}{\lambda_e}$$

Pour $\boldsymbol{\rho \neq 1}$

$$W = \frac{\dfrac{\rho}{1-\rho} - \dfrac{(K+1)\rho^{K+1}}{1-\rho^{K+1}}}{\lambda * (1 - \dfrac{1-\rho}{1-\rho^{K+1}} * \rho^K)} = \boldsymbol{\frac{1}{\mu} * \left[\frac{1}{1-\rho^K} - \frac{(K+1)\rho^{K+1}}{(1-\rho)(1-\rho^K)}\right]}$$

Pour $\boldsymbol{\rho \neq 1}$

$$W = \frac{\dfrac{K}{2}}{\dfrac{\lambda K}{K+1}} = \boldsymbol{\frac{K+1}{2\mu}}$$

Donc :

$$W = \begin{cases} \boldsymbol{\dfrac{1}{\mu} * \left[\dfrac{1}{1-\rho^K} - \dfrac{(K+1)\rho^{K+1}}{(1-\rho)(1-\rho^K)}\right]} & si\ \rho \neq 1 \\ \boldsymbol{\dfrac{K+1}{2\mu}} & si\ \rho = 1 \end{cases}$$

- **Temps moyen de service W_S**

$$W_S = \frac{1}{\mu}$$

Temps moyen d'attente dans la file W_Q

En utilisant la Loi de Little : $L_Q = \lambda_e \cdot W_Q$

$$W_Q = \frac{L_Q}{\lambda_e} = W - \frac{1}{\mu}$$

7.4. Le modèle $M/M/c/K$

Le modèle $M/M/c/K$ décrit un système de file d'attente avec arrivées de type Poisson, services exponentiels, c serveurs parallèles et une capacité totale limitée à K clients. Si le système est

plein, les nouvelles arrivées sont rejetées. Ce modèle est utilisé pour évaluer les performances des systèmes à ressources et files limitées.

7.4.1. Hypothèses du modèle $M/M/c/K$

1) Les clients arrivent selon un processus de **Poisson** de taux λ.
2) Le temps de service suit une distribution **Exponentielle** de taux μ.
3) Le système comporte c **serveurs**.
4) La capacité du système **est limitée**. Le système peut contenir **au plus K clients** (y compris les clients en attente et en service).
5) Les clients sont généralement servis dans l'ordre d'arrivée (**FIFO**).
6) Si un client arrive lorsque le système contient déjà K clients, il est ignoré ou rejeté (bloqué).
7) Le régime est stationnaire quelle que soit la valeur de l'intensité de trafic.

7.4.2. Modélisation du $M/M/c/K$

Le modèle $M/M/c/K$ est modélisé par une **Chaîne de Markov à Temps Continu** à $K+1$ états: $S = \{0, 1, 2, \ldots, K\}$. Chaque état n représente le nombre total de clients présents dans le système (en attente ou en service) avec les transitions suivantes :

$$\lambda_n = \begin{cases} \lambda & si \quad n < K \\ 0 & si \quad n \geq K \end{cases} \qquad \mu_n = \begin{cases} n * \mu & si \quad 1 \leq n < c \\ c * \mu & si \quad c \leq n \leq K \\ 0 & si \quad n = 0 \end{cases}$$

Le système suit donc **un processus de naissance-mort tronqué**, avec **rejet des arrivées** à l'état K.

La matrice Q est une matrice tridiagonale avec :

$$Q = \begin{pmatrix} -\lambda & \lambda & 0 & .. & 0 & 0 \\ \mu & -(\lambda+\mu) & \lambda & .. & 0 & 0 \\ 0 & 2\mu & -(\lambda+2\mu) & .. & \lambda & 0 \\ : & 0 & 3\mu & : & : & : \\ 0 & : & : & .. & -(\lambda+c\mu) & \lambda \\ 0 & 0 & 0 & .. & c\mu & -c\mu \end{pmatrix}$$

7.4.3. Calcul des probabilités d'états stationnaires

On a :

$$\pi_n = \pi_0 * \prod_{k=0}^{n-1} \frac{\lambda_k}{\mu_{k+1}}$$

Pour $n \leq c$

$$\pi_n = \pi_0 * \prod_{k=0}^{n-1} \frac{\lambda_k}{\mu_{k+1}} = \pi_0 * \prod_{k=0}^{n-1} \frac{\lambda}{(k+1)\mu} = \pi_0 * \prod_{k=0}^{n-1} \frac{\lambda}{\mu} * \frac{1}{(k+1)} = \pi_0 * \left(\frac{\lambda}{\mu}\right)^n * \frac{1}{1 * 2 * 3 * \ldots * n}$$

$$= \boldsymbol{\pi_0 * \frac{\rho^n}{n!}}$$

Pour $n \geq c$

$$\pi_n = \pi_0 * \prod_{k=0}^{c-1} \frac{\lambda}{(k+1)\mu} * \prod_{k=c}^{n-1} \frac{\lambda}{c\mu} = \pi_0 * \frac{\rho^c}{c!} * \left(\frac{\lambda}{c\mu}\right)^{n-c} = \pi_0 * \frac{\rho^c}{c!} * \left(\frac{\rho}{c}\right)^{n-c} = \boldsymbol{\pi_0 * \frac{\rho^n}{c! * c^{n-c}}}$$

Les équations de récurrence sont donc :

$$\boxed{\boldsymbol{\pi_n} = \begin{cases} \boldsymbol{\frac{\rho^n}{n!} * \pi_0} & \boldsymbol{si\ 0 \leq n \leq c} \\ \boldsymbol{\frac{\rho^n}{c! * c^{n-c}} * \pi_0} & \boldsymbol{si\ c \leq n \leq K} \end{cases}}$$

Le terme π_0 (probabilité que le système soit vide) est obtenu par normalisation :

$$\begin{cases} \pi_n = \frac{\rho^n}{n!} * \pi_0 & si\ 0 \leq n \leq c \\ \pi_n = \frac{\rho^n}{c! * c^{n-c}} * \pi_0 & si\ n \geq c \\ \sum_{n=0}^{K} \pi_n = 1 \end{cases}$$

Ainsi:

$$\sum_{n=0}^{K} \pi_n = \sum_{n=0}^{c-1} \frac{\rho^n}{n!} * \pi_0 + \sum_{n=c}^{K} \frac{\rho^n}{c! * c^{n-c}} * \pi_0 = 1$$

$$\pi_0 * \left[\sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \sum_{n=c}^{K} \frac{\rho^c . \rho^{n-c}}{c! * c^{n-c}}\right] = \pi_0 * \left[\sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \frac{\rho^c}{c!} * \sum_{n=c}^{K} \frac{\rho^{n-c}}{c^{n-c}}\right] = \pi_0 * \left[\sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \frac{\rho^c}{c!} * \sum_{n=c}^{K} \left(\frac{\rho}{c}\right)^{n-c}\right] = 1$$

Le résultat est :

$$\pi_0 = \left[\sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \left(\frac{\rho^c}{c!} \cdot \sum_{n=c}^{K} \left(\frac{\rho}{c}\right)^{n-c}\right)\right]^{-1}$$

Si $\rho = c$, la deuxième somme dans l'expression de π_0 est $(K - c + 1)$.

Si $\rho \neq c$, la deuxième somme dans l'expression de π_0 est une série géométrique.

Mettons pour simplification $j = n - c$. Alors :

$$\sum_{n=c}^{K} \left(\frac{\rho}{c}\right)^{n-c} = \sum_{j=0}^{K-c} \left(\frac{\rho}{c}\right)^{j} = \frac{1 - \left(\frac{\rho}{c}\right)^{K-c+1}}{1 - \frac{\rho}{c}}$$

Le résultat final est donc :

$$\boldsymbol{\pi_0} = \begin{cases} \left[\sum_{\boldsymbol{n=0}}^{\boldsymbol{c-1}} \frac{\boldsymbol{\rho^n}}{\boldsymbol{n!}} + \left(\frac{\boldsymbol{\rho^c}}{\boldsymbol{c!}} \cdot \frac{\mathbf{1} - \left(\frac{\rho}{c}\right)^{K-c+1}}{\mathbf{1} - \frac{\rho}{c}}\right)\right]^{-\mathbf{1}} & si\ \rho \neq c \\ \left[\sum_{\boldsymbol{n=0}}^{\boldsymbol{c-1}} \frac{\boldsymbol{\rho^n}}{\boldsymbol{n!}} + \boldsymbol{K - c + 1}\right]^{-\mathbf{1}} & si\ \rho = c \end{cases}$$

7.4.4. Calcul des mesures de performance

- **La probabilité de rejet P_K**

Un client est rejeté si et seulement si le système contient K clients à son arrivée.

$$\pi_n = \frac{\rho^n}{c! * c^{n-c}} * \pi_0 \qquad si\ c \leq n \leq K$$

Alors :

$$P_K = P(n = K) = \boldsymbol{\pi_K} = \frac{\rho^K}{c! * c^{K-c}} * \pi_0$$

- **Taux effectif λ_e**

Sur λ arrivées, seule la fraction $1 - P_K$ entre dans le système.

$$\lambda_e = \lambda * (1 - P_K)$$

- **Probabilité d'attente P_w**

La probabilité qu'un client arrivant doive attendre (c'est-à-dire que tous les serveurs soient occupés) est la somme des probabilités que le nombre de clients dans le système soit supérieur ou égal au nombre de serveurs c.

$$P_w = P(c \leq n \leq K) = \sum_{n=c}^{K} \pi_n = \sum_{n=c}^{K} \frac{\rho^{n+c-c}}{c! * c^{n-c}} * \pi_0 = \pi_0 * \frac{\rho^c}{c!} * \sum_{n=c}^{K} \left(\frac{\rho}{c}\right)^{n-c}$$

Donc :

$$P_w = \begin{cases} \pi_0 * \frac{\rho^c}{c!} * \frac{1 - \left(\frac{\rho}{c}\right)^{K-c+1}}{1 - \frac{\rho}{c}} & si\ \rho \neq c \\ \pi_0 * \frac{\rho^c}{c!} * (K - c + 1) & si\ \rho = c \end{cases}$$

Note : La formule de P_w représente la probabilité qu'un client arrivant **ne puisse pas être servi immédiatement** parce que tous les serveurs sont occupés. Que ce client soit ensuite placé en file d'attente ou rejeté dépend de la capacité restante du système.

- **Nombre moyen de clients dans la file d'attente** L_Q

Pour calculer L_Q, on doit sommer le produit du nombre de clients en file d'attente pour chaque état pertinent $n - c$ par la probabilité de cet état π_n

$$L_Q = E(L_Q) = \sum_{n=c+1}^{K} (n - c) * \pi_n = \sum_{\boldsymbol{n=c+1}}^{\boldsymbol{K}} (\boldsymbol{n - c}) * \frac{\boldsymbol{\rho^n}}{\boldsymbol{c! * c^{n-c}}} * \boldsymbol{\pi_0}$$

Le développement de cette expression nous donne :

$$L_Q = \sum_{n=c+1}^{K} (n - c) * \frac{\rho^c * \rho^{n-c}}{c! * c^{n-c}} * \pi_0 = \frac{\rho^c}{c!} * \pi_0 * \sum_{n=c+1}^{K} (n - c) * \frac{\rho^{n-c}}{c^{n-c}}$$

$$= \frac{\rho^c}{c!} * \pi_0 * \sum_{n=c+1}^{K} (n - c) * (\frac{\rho}{c})^{n-c}$$

Simplifions en posant $j = n - c$. Lorsque $n = c + 1, j = 1$. Lorsque $n = K, j = K - c$.

$$L_Q = \frac{\rho^c}{c!} * \pi_0 * \sum_{j=1}^{K-c} j * (\frac{\rho}{c})^j$$

- ***Pour*** $\boldsymbol{\rho \neq c}$

$$L_Q = \frac{\rho^c}{c!} * \pi_0 * \left(\frac{\rho}{c} * \frac{\boldsymbol{d}}{\boldsymbol{d\rho}} \frac{1 - (\frac{\rho}{c})^{K-c+1}}{1 - \frac{\rho}{c}} \right)$$

$$= \frac{\rho^c}{c!} * \pi_0 * \left(\frac{\rho}{c} * \frac{1 - (K - c + 1)(\frac{\rho}{c})^{K-c} + (K - c)(\frac{\rho}{c})^{K-c+1}}{\left(1 - \frac{\rho}{c}\right)^2} \right)$$

$$= \pi_0 * \frac{\rho^{c+1}}{c * c!} * \left(\frac{1 - (K - c + 1)(\frac{\rho}{c})^{K-c} + (K - c)(\frac{\rho}{c})^{K-c+1}}{\left(1 - \frac{\rho}{c}\right)^2} \right)$$

- ***Pour $\rho = c$***

$$L_Q = \pi_0 * \frac{\rho^c}{c!} * \frac{(K-c)(K-c+1)}{2}$$

- **Nombre moyen de clients dans le service L_S**

En général, pour un système avec des pertes (rejet de clients), le nombre moyen de serveurs occupés est égal au taux d'arrivée effectif divisé par le taux de service d'un serveur:

$$L_S = \frac{\lambda_e}{\mu} = \frac{\lambda(1-P_K)}{\mu} = \boldsymbol{\rho * (1 - P_K)}$$

- **Nombre moyen de clients dans le système L**

$$L = L_Q + L_S$$

- **Temps moyen de séjour dans le système W**

En utilisant la loi de Little : $L = \lambda_e \cdot W$

$$W = \frac{\boldsymbol{L}}{\boldsymbol{\lambda_e}}$$

- **Temps moyen de service W_S**

Le temps de service suit une loi exponentielle de paramètre μ, donc l'espérance est $1/\mu$.

$$W_S = \frac{\mathbf{1}}{\boldsymbol{\mu}}$$

- **Temps moyen d'attente W_Q**

En utilisant la formule de Little: $L_Q = \lambda_e \cdot W_Q$

$$W_Q = \frac{\boldsymbol{L_Q}}{\boldsymbol{\lambda_e}} \quad ou \quad W_Q = \boldsymbol{L} - \frac{\mathbf{1}}{\boldsymbol{\mu}}$$

7.5. Cas particulier : Modèle M/M/c/c et la formule de perte d'Erlang B

Le modèle $M/M/c/c$ est une file d'attente $M/M/c/K$ où la capacité totale du système K est égale au nombre de serveurs c ($K = c$). Cela implique qu'il n'y a pas de file d'attente possible ($K - c = c - c = 0$): si un client arrive et trouve tous les serveurs occupés, il ne peut pas attendre et est immédiatement rejeté (perdu). C'est pourquoi on parle de ***"système avec perte"*** ou "loss system".

Ce modèle est un outil essentiel en ingénierie du trafic pour dimensionner les ressources (comme les lignes téléphoniques ou les serveurs web) afin de garantir un certain niveau de service (faible

probabilité de blocage) face à une charge de trafic donnée.

7.5.1. Formule de probabilité de rejet (Erlang B)

Pour le modèle $M/M/c/c$, la mesure de performance la plus critique est **la probabilité de rejet** (blocage), c'est-à-dire la probabilité qu'un client arrivant soit rejeté parce que tous les serveurs sont occupés. Dans ce cas, P_K est la probabilité que le système soit dans l'état où c serveurs sont occupés, soit P_c.

En utilisant les formules des probabilités d'état stationnaire pour le $M/M/c/K$ et en posant $K = c$:

La probabilité d'avoir n clients dans le système est :

$$\pi_n = \frac{\rho^n}{n!}\pi_0 \qquad pour\ 0 \le n \le c$$

La probabilité que le système soit vide π_0 est donné par la condition de normalisation :

$$\pi_0 = \left[\sum_{n=0}^{c} \frac{\rho^n}{n!}\right]^{-1}$$

Maintenant la probabilité de rejet P_c (Erlang B) est simplement π_c, car un client est rejeté ou bloqué si tous les c serveurs sont occupés :

$$\pi_c = \frac{\rho^c}{c!}\pi_0$$

En substituant l'expression de π_0 dans celle de π_c, on obtient **la formule d'Erlang B**:

$$P_c = B(\rho, c) = \frac{\frac{\rho^c}{c!}}{\sum_{n=0}^{c} \frac{\rho^n}{n!}}$$

Une propriété remarquable de **la formule d'Erlang B** est son insensibilité à la distribution exacte des temps de service, tant que leur moyenne est $1/\mu$. Cela signifie que la formule reste valide pour un modèle $M/G/c/c$ (où 'G' signifie que la distribution du temps de service est générale, pas nécessairement exponentielle) tant que la moyenne est connue. C'est une caractéristique très puissante pour les applications pratiques.

7.5.2. Mesures de performance pour M/M/c/c

✧ Probabilité de rejet (blocage) P_c: C'est la formule d'Erlang B ci-mentionné.

✧ Nombre moyen de clients dans le système $L = \sum_{n=0}^{c} n * \pi_n$

- ✧ Nombre moyen de clients en service $L_s = \lambda_e/\mu = \lambda(1 - P_c)/\mu = \rho * (1 - P_c)$. Notons que dans ce modèle $L = L_s$ car il n'y a pas de file d'attente $L_q = 0$.
- ✧ Temps moyen de séjour dans le système $W = 1/\mu$ car il n'y a ps d'attente, donc $W = W_s$.
- ✧ Temps moyen d'attente en file $W_q = 0$ (pas de file d'attente).
- ✧ Probabilité d'attente $P_w = 0$ car un client qui trouve tous les serveurs occupés est rejeté, il n'attend pas.

Annexe : Calcul des probabilités stationnaire à partir des équations d'équilibre

❖ Modèle M/M/1

On peut calculer les probabilités stationnaires en se reposant sur le principe fondamental de conservation des flux: *«en régime stationnaire, le taux d'entrée dans chaque état doit être égal au taux de sortie de cet état»*.

L'application systématique de ce principe à chaque état du processus de naissance-mort conduit aux **équations de balance global** suivantes :

$$\begin{cases} \lambda\pi_0 = \mu\pi_1 & pour\ l'état\ 0 \\ (\lambda+\mu)\pi_n = \lambda\pi_{n-1} + \mu\pi_{n+1} & pour\ l'état\ n \end{cases}$$

La première équation repose sur le fait que la probabilité qu'un client entre dans le système lorsqu'il n'y a aucun client dans le système est égale à la probabilité qu'un client quitte le système lorsqu'un client est présent dans le système. La seconde équation exprime l'équilibre lorsque un client entre dans le système et qu'un autre client en sort.

En appliquant les deux équations :

Etat	**Équation d'équilibre**	**Équation simplifiée**
0	$\lambda\pi_0 = \mu\pi_1$	$P_1 = {}^{\lambda}/_{\mu}\,\pi_0$
1	$(\lambda+\mu)\pi_1 = \lambda\pi_0 + \mu\pi_2$	$P_2 = ({}^{\lambda}/_{\mu})^2\pi_0$
2	$(\lambda+\mu)\pi_2 = \lambda\pi_1 + \mu\pi_3$	$P_3 = ({}^{\lambda}/_{\mu})^3\pi_0$
3	$(\lambda+\mu)\pi_3 = \lambda\pi_2 + \mu\pi_4$	$P_4 = ({}^{\lambda}/_{\mu})^4\pi_0$
....	.....	.....
n	$(\lambda+\mu)\pi_n = \lambda\pi_{n-1} + \mu\pi_{n+1}$	$\pi_n = ({}^{\lambda}/_{\mu})^n\pi_0$

❖ Modèle M/M/c

Les **équations de balance global** du système M/M/c au régime stationnaire sont :

$$\begin{cases} \lambda\pi_0 = \mu\pi_1 & pour\ n = 0 \\ (\lambda + n\mu)\pi_n = \lambda\pi_{n-1} + (n+1)\mu\pi_{n+1} & pour\ n < c \\ (\lambda + c\mu)\pi_n = \lambda\pi_{n-1} + c\mu\pi_{n+1} & pour\ n \geq c \end{cases}$$

Notons π_n la probabilité d'être dans l'état n à l'équilibre.

Résolution de système d'équations par récurrence:

Etat	**Équation d'équilibre**	**Équation simplifiée**
0	$\lambda\pi_0 = \mu\pi_1$	$\pi_1 = \rho\pi_0$
1	$(\lambda + \mu)\pi_1 = \lambda\pi_0 + 2\pi_2$	$\pi_2 = \frac{\rho^2}{2}\pi_0$
2	$(\lambda + 2\mu)\pi_2 = \lambda\pi_1 + 3\pi_3$	$\pi_3 = \frac{\rho^3}{6}\pi_0$
3	$(\lambda + 3\mu)\pi_3 = \lambda\pi_2 + 4\pi_4$	$P_4 = \frac{\rho^4}{24}\pi_0$
....	.....	.....
$n < c$	$(\lambda + n\mu)\pi_n = \lambda\pi_{n-1} + (n+1)\pi_{n+1}$	$\pi_n = \frac{\rho^n}{n!}\pi_0$
....	.....	.....
$n = c$	$(\lambda + c\mu)\pi_c = \lambda\pi_{c-1} + c\mu\pi_{c+1}$	$\pi_{c+1} = \frac{\rho^{c+1}}{c! * c}\pi_0$
	.....	
$n \geq c$	$(\lambda + c\mu)\pi_n = \lambda\pi_{n-1} + c\mu\pi_{n+1}$	$\pi_n = \frac{\rho^n}{c! * c^{n-c}}\pi_0$