



**RAPPORT D'ÉTUDE DE POLITIQUE A SEUILS POUR L'OPTIMISATION  
ENERGETIQUE DANS UN CLOUD**

---

**Derra Abdoul Rachid :21918210**

**Maiga Abdalah Mohamed :21916398**

**Ouedraogo Jean Jacques :21921901**

**Simporé Naimatou :21921358**

Encadré par **Youssef Ait El Mahjoub.**

1. INTRODUCTION .....	2
2. DEFINITION DE QUELQUES CONCEPTS IMPORTANTS .....	2
2.1. DEFINITION ET NOTION DE CHAINE DE MARKOV.....	2
2.2. DEFINITION DU MODELE MITRANI ET RAPPEL DE QUELQUES LOIS.....	4
2.3. PRESENTATION DES DEUX MODELES.....	4
3. ANALYSES NUMERIQUES.....	4
4. RESULTATS NUMERIQUES.....	6
5. COMPARAISON DES DEUX MODELES .....	19
6. COMPARAISON DE L'EXPERIENCE 1 DE CHAQUE MODELE .....	20
6.1. COMPARAISON DES HEATMAP DE PERTES.....	20
6.2. COMPARAISON HEATMAP ENERGIE.....	21
7. COMPARAISON DE L'EXPERIENCE 2 DE CHAQUE MODELE .....	22
7.1. HEATMAP DES PERTES .....	22
7.2. HEATMAP DE L'ENERGIE.....	22
8. CONCLUSION .....	24
9. LISTE DES REFERENCES BIBLIOGRAPHIQUES.....	25

### Liste des Tableaux

Tableau 1:récapitulatif du taux de pertes experience1 modèle erlang .....	7
Tableau 2: Récapitulatif énergie expérience 1 modèle erlang .....	9
Tableau 3:Récapitulatif du taux de pertes expérience 2 modèle erlang.....	10
Tableau 4: Récapitulatif de la consommation d'énergie sur l'expérience 2 modèle erlang.....	12
Tableau 5: Récapitulatif du taux de pertes expérience 1 du modèle exponentielle .....	14
Tableau 6: Récapitulatif de la consommation énergétique de l'expérience.....	15
Tableau 7:Recapitulatif du taux de perte expérience 2 du modèle exponentielle.....	17
Tableau 8: Récapitulatif de la consommation d'énergie de l'expérience 2 du modèle exponentielle ....	18
Tableau 9: Liste des compromis des deux modèles .....	18
Tableau 10: Récapitulatif comparaison du modèle erlang et du modèle exponentielle expérience 1 ...	22
Tableau 11: Récapitulatif comparaison du modèle erlang et du modèle exponentielle expérience 2....	23

### Liste des Figures

Figure 1:Heatmap des pertes expérience 1 modèle erlang.....	6
Figure 2: Heatmap consommation d'énergie expérience 1 du modèle erlang.....	8
Figure 3 : Heatmap du taux de perte expérience 2 modèle erlang.....	9
Figure 4: heatmap consommation d'énergie de l'expérience 2 modèle erlang .....	11
Figure 5: Heatmap du taux de perte modèle exponentielle de l'expérience 1 .....	13
Figure 6: Heatmap de la consommation d'énergie du modèle exponentielle de l'expérience 1 .....	14
Figure 7: Heatmap du taux de perte expérience 2 modèle exponentielle.....	16
Figure 8: Heatmap de la consommation d'énergie expérience 2 du modèle exponentielle.....	17
Figure 9: Heatmap des expérience1 de chaque modèle .....	21
Figure 10: Heatmaps expérience 2 de chaque modèle .....	23

## Introduction

La consommation énergétique des data center est un problème crucial. En réalité, cette difficulté s'explique par le fait qu'on doit trouver un mécanisme permettant d'arrêter des serveurs pour qu'ils consomment moins et à les réveiller quand le nombre de jobs augmente. En effet, on devrait pouvoir mettre en œuvre un modèle à seuil où l'on choisit une valeur D pour arrêter des serveurs et U pour les allumer. On compare le nombre de jobs dans le data center à ces deux seuils pour allumer et éteindre. Cependant les seuils ont une influence sur la consommation d'énergie, d'où la nécessité de proposer une solution adéquate notamment en étudiant le compromis consommation énergétique des serveurs et quelques indices de performances tels que le nombre moyen de clients et le taux de pertes. Une des solutions est d'adopter un modèle de politique à seuil Mitrani avec quelques restrictions. Ce modèle est particulier puisqu'il utilise une file d'attente finie, il ne considère pas l'impatience d'un client, aussi les serveurs peuvent prendre 3 états ON (allumé), WARMING (en allumage), OFF (éteint). A l'aide du logiciel Xborne (développé au laboratoire DAVID) nous avons modélisé le système pour un couple de seuils donnés. Ce logiciel permet également de résoudre très vite le modèle. En ce qui concerne l'allumage des serveurs on distingue deux types d'allumages à savoir le modèle avec allumage Erlang et le modèle avec allumage Exponentielle. Dans le but de rendre compréhensible notre étude nous expliquerons la définition des différents concepts, puis nous préciserons le processus de mise en place des différents modèles (Modèle 1 et 2), également on exposera le résultat de nos analyses suivi d'une interprétation avant de proposer une solution pour l'optimisation des ressources utilisées.

## Définition de quelques concepts importants

### Définition et notion de chaîne de Markov

Une chaîne de Markov, est une suite de variables aléatoires ( $X_n$  avec  $n$  strictement positif) qui permet de modéliser l'évolution dynamique d'un système aléatoire, où  $X_n$  représente l'état du système à l'instant  $n$ . Sa propriété fondamentale est que son évolution future ne dépend du passé qu'au travers de sa valeur actuelle c'est-à-dire que dans son évolution dans le temps, l'état du processus à un instant futur ne dépend que de celui à l'instant présent et non de ses états antérieurs. Nous avons deux types de Chaîne de Markov à savoir :

**Les chaînes de Markov en temps discret** : on parle de probabilité de transition d'un état a un autre état, la loi des arrivés des clients dans la file suit une certaine probabilité, c'est à dire que l'ensemble des valeurs que peut prendre la variable aléatoire qui suit cette loi est dans un ensemble discret (ex :  $\mathbb{N}$  l'ensemble des entiers naturel)

**Les chaînes de Markov en temps continue** : la loi des arrivés des clients dans la file suit un certains taux, par exemple on peut avoir 4 clients par unité de temps.

La loi des arrivés et des services nous renseigne sur l'état de notre chaîne en temps discret, et en temps continue.

**XBORNE**, logiciel (en langage C) pour la résolution de chaînes de Markov en temps discret (**CMTD**), or dans les différents modèles nous utilisons les **CMTC**. Ainsi, pour la résolution de notre chaîne de Markov avec XBORNE, il nous faut faire une **uniformisation** qui a pour objectif de transformer une chaîne de Markov en temps continue (CMTD) en chaîne de Markov en temps discret (CMTD). Pour cela, nous utilisons plusieurs fichiers pour la génération et résolution de notre matrice stochastique. Nous avons les fichiers suivant qui servent à :

**const.h** : définir les constantes pour chaque modèle à définir

**var.h** : définir les variables axillaires d'un modèle

**fun.c** : calculer les probabilités pour les évènements (état) en fonction du codage des évènements (déterminant l'état des serveurs).

**generMarkov.c** : générer la matrice de transition qui est définie à travers ces trois fichiers ci-dessous

**model.sz** : décrire la taille de la matrice

**model.cd** : décrire les différents états générés

**model.rii** : déterminer les transitions (dans les différents états)

**gthLD.c** : calculer la distribution stationnaire de la chaîne de Markov en prenant en entrée un fichier de type Rxx (transition des matrices) et qui produira un fichier contenant la distribution stationnaire.

**Rewards** : calculer les récompenses à partir de la matrice, comme récompenses nous avons, la consommation énergétique, le nombre moyen de client dans la file et les pertes de client dans la file.

**La distribution stationnaire d'une chaîne de Markov ( $\pi$ )**, s'interprète comme étant la fraction du temps passé en chaque état  $i$  de l'espace d'états  $E$  d'une chaîne de Markov, elle représente en moyenne combien on peut rester dans un état donné.

## Définition du modèle Mitrani et rappel de quelques lois

- **Modèle Mitrani** : est un modèle de file d'attente dont les arrivées sont **Poissoniennes**, dans ce modèle, un client ne peut pas attendre indéfiniment, c'est à dire à la fin d'un délai le client quitte la file d'attente sans être servi. Le délai d'attente d'un client suit la distribution exponentielle. Le temps d'allumage des serveurs et la durée d'un service est **Exponentiel**, et la file d'attente est infinie.
- **La loi Exponentielle** : modélise la durée de vie d'un phénomène sans mémoire, ou sans vieillissement, ou sans usure : la probabilité que le phénomène dure au moins  $s + t$  heures sachant qu'il a déjà duré  $t$  heures sera la même que la probabilité de durer  $s$  heures à partir de sa mise en fonction initiale. En d'autres termes, le fait que le phénomène ait duré pendant  $t$  heures ne change rien à son espérance de vie à partir du temps  $t$ .
- **La loi d'Erlang** : La **distribution d'Erlang** est une loi de probabilité continue, dont l'intérêt est dû à sa relation avec les distributions exponentielle et Gamma. C'est la loi d'une somme variable exponentielle.

## Présentation des deux modèles

- **Le modèle avec allumage Erlang des serveurs**: la distribution Erlang est une distribution de la somme de  $k$  variables aléatoires indépendantes et identiquement distribuées selon une loi exponentielle de paramètre  $\beta$ . Dans ce modèle, le temps d'allumage suit la loi Erlang  $(k, \beta)$  qui signifie qu'en moyenne le temps d'allumage est de  $k/\beta$ .
- **Le modèle à allumage Exponentielle des serveurs**: La loi exponentielle est une loi de probabilité continue mélangeant plusieurs lois exponentielles. Elle dépend de trois paramètres à savoir :  $k$  qui représente le nombre de lois exponentielles indépendantes, les paramètres de ces lois exponentielles et une pondération de ces lois.

## Analyses Numériques

Pour l'analyse des deux modèles et pour chaque expérience (1 et 2) nous avons généré respectivement 1225 valeurs et 2416 valeurs pour les couples  $U$  et  $D$ .

Dans la suite de cette partie, nous expliquerons les parties de codes :

**Le calcul des pertes** : Le calcul des pertes se fait grâce au fichier funRewardMit.c, celui-ci grâce à la fonction proba\_perte, qui parcourt les états atteignables (c'est à dire on parcourt toutes les composantes du vecteur d'état), puis vérifie si on a atteint valeur du premier composant a atteint la fin de la file c'est à dire le buffersize, puis on vérifie quel est l'état de du contrôle de puissance, c'est à dire 1 pour signifier que les serveurs sont allumés mais pas encore opérationnel, et 2 pour signifier que les serveurs sont allumés et opérationnel.

Si les deux conditions sont vérifiées, alors on récupère l'indice pour chaque composant et on retourne la distribution stationnaire de la somme des deux indices des composants.

**Le calcul de l'énergie** : Le calcul se fait grâce au fichier funRewardMit.c. La fonction énergie fait un parcours des états atteignables puis vérifie si le codage de évènement futur (tab [i+1] ) correspond au codage prédéfinie pour l'état des serveurs de réserves, c'est à dire : 0 = OFF (serveur éteint), 1 = allumé mais pas opérationnel (Serveur en allumage) et 2 = allumé, opérationnel (Serveur allumé).

- **Inservice** : Correspond aux nombres de serveur en service
- **Outservice** : Correspond aux nombres de serveurs non en service
- **Si l'état futur du serveur est 0** : alors les serveurs sont éteints, donc pour calculer la consommation, on multiplie Inservice par **Energie1** qui représente la consommation des serveurs allumés actifs qu'on additionne avec Outservice par **Energie** qui représente la consommation des serveurs allumés non actifs.
- **Si l'état futur du serveur est 1** : alors les serveurs sont en allumage mais pas encore opérationnel, pour calculer la consommation d'énergie, alors on multiplie Inservice par **Energie1** qui représente la consommation énergétique des serveurs a l'état actif, qu'on additionne avec Outservice multiplié par **Energie2** qui représente la consommation des serveurs allumés mais non actif (c'est à dire après avoir fourni les services ils restent actifs, donc consomment de l'énergie), qu'on additionne encore avec OnAndOff (les serveurs de réserves qui doivent s'allumer) multiplié par **EnergieOn** qui représente la consommation d'allumage des serveurs.
- **Si l'état futur du serveur est 2** : alors les serveurs sont allumés et opérationnel, pour calculer sa consommation alors, on multiplie Inservice par **Energie1** (consommation des serveurs actifs), qu'on additionne a Outservice multiplier par **Energie2** (consommation des serveurs

allumé mais non actifs, c'est à dire après avoir fourni le service ils restent actifs et consomment de l'énergie.

## Résultats Numériques

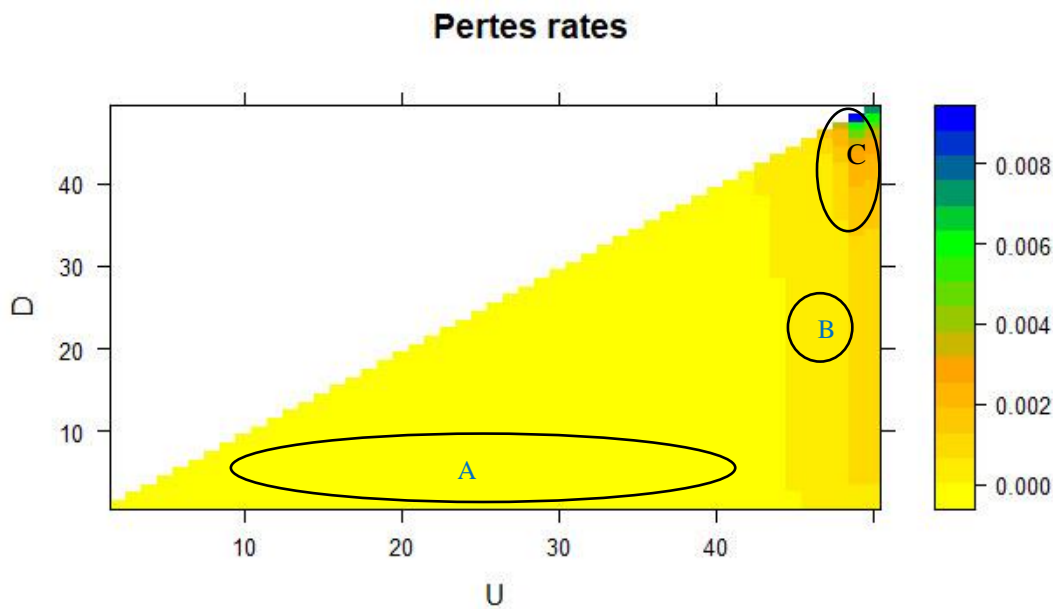
Notre analyse s'est faite sur deux expériences, c'est à dire pour chacun des modèles **Erlang** et **Exponentielle** nous avons effectué deux expériences définies comme suit :

- **Expérience 1** : Pour cette expérience nous avons fixé les valeurs des variables suivantes comme suit, Bufferize = 50 (taille de la file), AlwaysOn = 10 (Les serveurs principaux qui sont toujours allumés), OnandOff = 5 (les serveurs de réserves qui peuvent être soit en état d'allumage, état éteint ou en allumage) et Lambda = 10.
- **Expérience 2** : Pour cette expérience nous avons fixé les valeurs des variables suivantes comme suit, Bufferize = 70 (taille de la file), AlwaysOn = 15 (Les serveurs principaux qui sont toujours allumés), OnandOff = 5 (les serveurs de réserves qui peuvent être soit en état d'allumage, état éteint ou en allumage) et Lambda = 12 (le taux d'arrivée par unité de temps)

Dans la suite nous présenterons pour chacun des deux modèles les résultats obtenus :

### Modèle avec allumage Erlang :

- **Expérience 1** :



**Figure 1 :** Heatmap des pertes expérience 1 modèle erlang

Le graphique ci-après décrit le taux de pertes dans la file d'attente pour le modèle Erlang lors de l'expérience 1. Cette figure présente les valeurs U (up) en abscisse, D (down) en ordonnée puis une troisième valeur qui représente le taux de perte de client en fonction de la couleur.

Par conséquent une observation des différentes couleurs du graphique permet de connaître la perte associée à un ou plusieurs ensembles de couple de valeur de U et D. Nous avons noté principalement 3 grandes zones à coloration différentes identifiées de A à C.

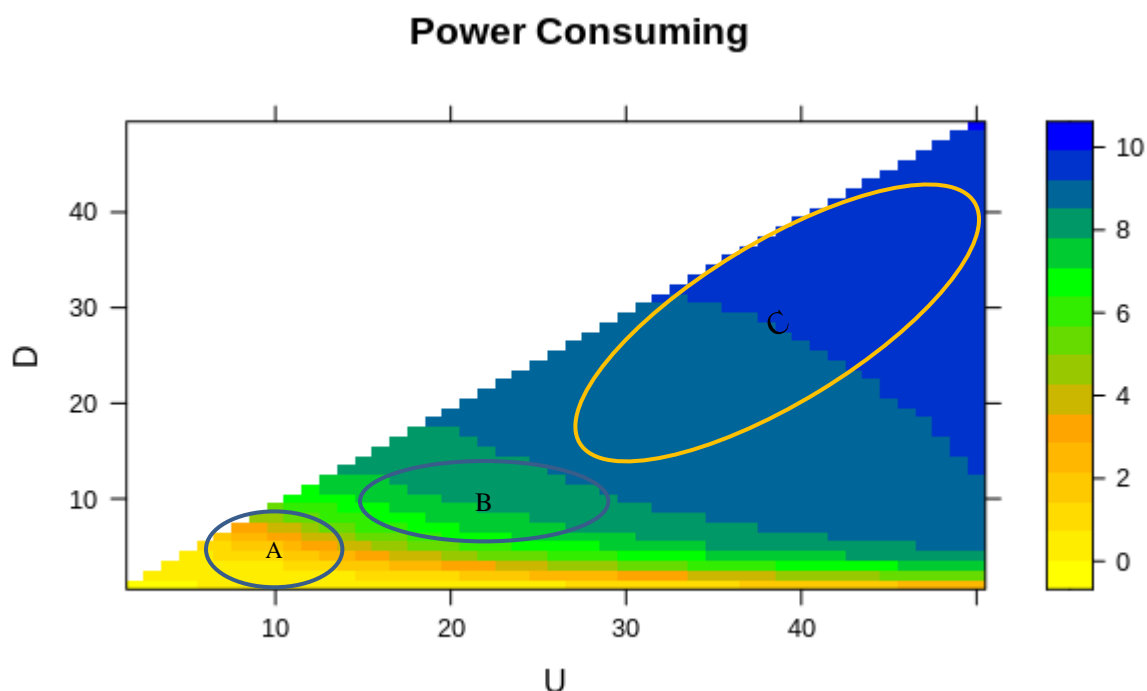
La zone A en forte coloration jaune indique pour les couples de valeur U comprises entre 1 et 45 un taux de perte quasiment nulle. Pour ce qui est de la zone B on remarque un changement de couleur caractéristique de l'augmentation du taux de pertes compris entre 0 et 0.0002. Les valeurs de couples concernés sont de 46 à 50. La zone C est une zone à risque, en effet elle contient des valeurs dont le taux de perte est compris entre 0.002 et 1, les valeurs associées à cette zone doivent être évitées dans le choix des couples U et D.

**Tableau 1:** récapitulatif du taux de pertes expérience 1 modèle erlang

Expérience 1 :Heatmap pertes  Buffersize =50; AlwaysOn=10; OnandOff=5; Lamba=10 Moyenne pertes=0.004.			
Zone	U	D	Conclusion
A	2-45	1-42	Taux de perte nulle.
B	46-50	43-50	Taux de perte acceptable compris entre 0.002 et 0.004.
C	48-50	35-50	Taux de pertes très élevé et varié compris entre 0.002 et 0.01.

### Heatmap des énergies





**Figure 2:** *Heatmap consommation d'énergie expérience 1 du modèle erlang*

Le graphique ci-après décrit la consommation d'énergie pour le modèle Erlang lors de l'expérience 1. Cette figure présente les valeurs U (up) en abscisse, D (down) en ordonnée puis une troisième valeur qui représente l'énergie en fonction de la couleur.

Par conséquent une observation des différentes couleurs du graphique permet de comprendre l'énergie associée à un ou plusieurs ensembles de couple de valeur de U et D. Nous avons noté principalement 3 grandes zones à coloration différentes identifiées de A à C.

La zone A à coloration jaune indique des couples pour lesquelles la consommation d'énergie est comprise entre 0 et 4. Cette zone est délimitée par des couples de valeur U compris de 1 à 50 et D compris de 1 à 9. C'est une zone dont le compromis en perte est bon, en effet pour la même plage de couple U et D on a un taux de perte nul.

La zone B est une zone intermédiaire entre les hautes valeurs et les valeurs basses. En effet, à ce niveau du graphique la consommation d'énergie varie entre 4 et 8 pour des couples de valeurs U compris entre 8 et 50 et D compris entre 2 et 12.

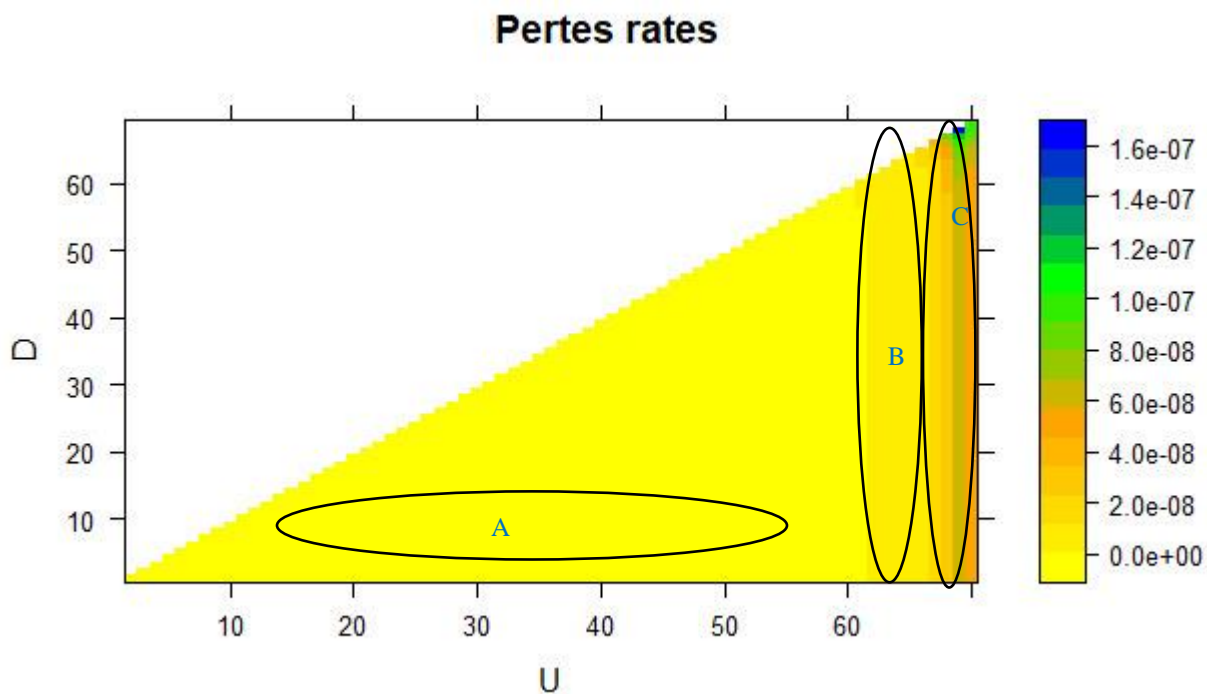
La zone C est la zone à haute valeur d'énergie, en effet les valeurs de l'énergie sont comprises entre 8 et 10 pour des couples de valeurs U comprises entre 13 et 50 et D compris entre 4 et 50.

C'est une zone au mauvais compromis énergie perte car l'énergie consommée est conséquente même si pour certaines valeurs de couple le taux de perte est faible.

**Tableau 2:** Récapitulatif énergie expérience 1 modèle erlang

Expérience 1 :Heatmap Energie			
Buffersize =50; AlwaysOn=10; OnandOff=5; Lamba=10 Moyenne énergie=5.			
Zone	U	D	Conclusion
A	1-50	1-9	Energie situé entre 0 et 4. Bon compromis énergie perte.
B	8-50	2-12	Energie comprise entre 4 et 8.
C	13-50	4-50	Energie comprise entre 8 et 10. Mauvais compromis énergie perte.

- **Expérience 2 :**



**Figure 3 :** Heatmap du taux de perte expérience 2 modèle erlang.

Le graphique ci-après décrit le taux de pertes relatif au nombre de clients dans la file pour le modèle Erlang lors de l'expérience 2. Contrairement à l'expérience 1 les couples de valeurs sont pris entre 1 et 70. Cette figure présente les valeurs U (up) en abscisse, D (down) en ordonnée puis une troisième valeur qui représente le taux de perte en fonction de la couleur. On peut observer principalement trois zones à coloration différentes.

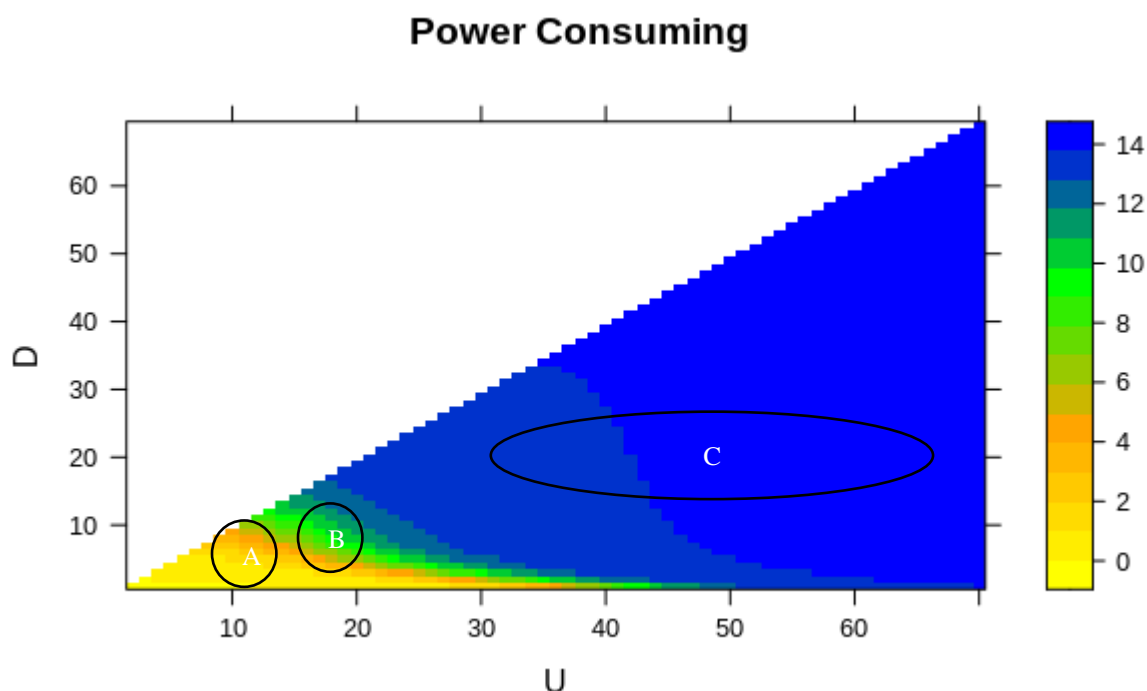
La première zone (A) de couleur jaune vive est caractéristique d'une perte quasiment nulle, elle prend en compte les couples de valeurs U comprises entre 1 et 62 et D comprises entre 1 et 60.

Une deuxième zone (B) intermédiaire de couleur jaune foncée caractéristique d'un taux de perte égale à  $10^{-8}$ . Elle regroupe les couples de valeurs U compris entre 63 et 67 et D compris de 1 à 68.

Une troisième zone (C) avec une multitude de couleur caractéristique d'une forte variation du taux de perte. En effet, le taux de perte est compris entre  $2 \cdot 10^{-8}$  et  $6 \cdot 10^{-7}$ . Elle regroupe les couples de valeurs U compris entre 67 et 70 et D compris de 1 à 70. C'est une zone à éviter afin d'optimiser le taux de perte.

**Tableau 3:** Récapitulatif du taux de pertes expérience 2 modèle erlang

Expérience 2 :Heatmap pertes			
Buffersize =70; AlwaysOn=15; OnandOff=5; Lamba=12, Moyenne pertes=0.8e-7.			
Zone	U	D	Conclusion
A	1-62	1-60	Taux de perte nulle.
B	63-67	1-68	Taux de perte égale $10^{-8}$ .
C	68-70	1-70	Taux de pertes très varié compris entre $2 \cdot 10^{-8}$ et $6 \cdot 10^{-7}$ . Couple de valeur à éviter pour optimiser le taux de perte.



**Figure 4:** *heatmap consommation d'énergie de l'expérience 2 modèle erlang*

Le graphique ci-après décrit la consommation d'énergie pour le modèle Erlang lors de l'expérience 2. Cette figure présente les valeurs U (up) en abscisse, D (down) en ordonnée puis une troisième valeur qui représente l'énergie en fonction de la couleur. Contrairement à l'expérience 1 les couples de valeurs sont pris entre 1 et 70.

Par conséquent une observation des différentes couleurs du graphique permet de comprendre l'énergie associée à un ou plusieurs ensembles de couple de valeur de U et D. Nous avons noté principalement 3 grandes zones à coloration différentes identifiées de A à C.

La zone A à coloration jaune indique des couples pour lesquelles la consommation d'énergie est comprise entre 0 et 4. Cette zone est délimitée par des couples de valeur U comprises de 1 à 38 et D compris de 1 à 9. C'est une zone dont le compromis en perte est bon, en effet pour la même plage de couple U et D on a un taux de perte nul.

La zone B est une zone intermédiaire entre les hautes valeurs et les valeurs basses. En effet, à ce niveau du graphique la consommation d'énergie varie entre 4 et 12 pour des couples de valeurs U compris entre 10 et 48 et D compris entre 2 et 15.

La zone C est la zone à haute valeur d'énergie, en effet les valeurs de l'énergie sont comprises entre 12 et 14 pour des couples de valeurs U compris entre 15 et 70 et D compris entre 1 et 70. C'est une zone

au mauvais compromis énergie perte car l'énergie consommée est conséquente même si pour certaines valeurs de couple le taux de perte est faible.

**Tableau 4:** *Récapitulatif de la consommation d'énergie sur l'expérience 2 modèle erlang*

Expérience 2 :Heatmap Energie			
Buffersize =70; AlwaysOn=5; OnandOff=5;Lamba=12 Moyenne Energie=7.			
Zone	U	D	Conclusion
A	1-38	1-9	Energie situé entre 0 et 4. Bon compromis énergie perte.
B	10-48	2-15	Energie comprise entre 4 et 12.
C	15-70	1-70	Energie comprise entre 12 et 14. Mauvais compromis énergie perte.

### **Comparaison de l'expérience 1 et l'expérience 2 du modèle avec allumage Erlang :**

- **Comparaison paramètres utilisés**

Pour l'expérience 1, nous avons la valeur des variables suivantes qui valent :

**Buffersize = 50**

**Lambda = 10**

**AlwaysOn = 10**

**OnandOff = 5**

Pour l'expérience 2 nous avons :

**Buffersize = 70**

**Lambda = 12**

**AlwaysOn = 15**

**OnandOff = 5**

- **Comparaison des deux heatmaps pour le taux de perte**

Contrairement à l'expérience 1 le taux de perte pour l'expérience 2 reste nul pour des valeurs de couple compris de 1 à 62 malgré l'augmentation du nombre de client qui passe de 50 à 70, du taux d'arrivé par unité de temps qui passe de 10 à 12. Cependant nous observons que le fait d'avoir renforcé la capacité

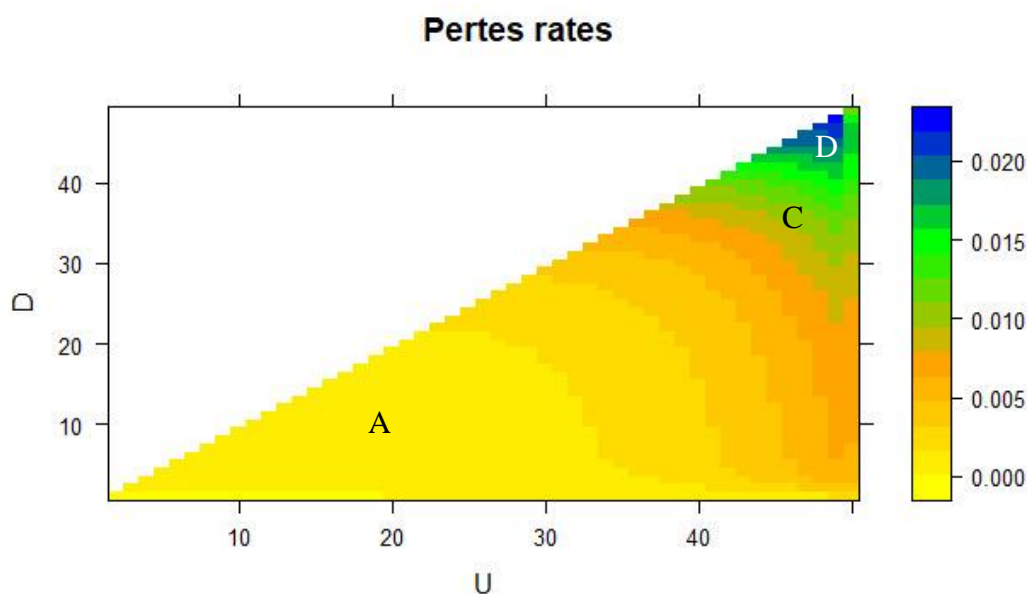
des serveurs actifs participe à réduire conséquemment le taux de pertes, de ce fait, l'idée d'augmenter des serveurs actifs pour diminuer le taux de perte est valable. Par contre en augmentant la capacité de ces serveurs cela à une grande répercussion sur la consommation énergétique.

- **Comparaison des deux heatmaps pour la consommation d'énergie**

Dans la première expérience nous observons que pour des couples avec de grandes valeurs U l'énergie est toujours basse en revanche plus le facteur U croit, plus il va falloir avoir un D petit pour rester dans une zone où la consommation en énergie est petite. Cependant dans la deuxième expérience pour des couples avec U supérieur à 40 la consommation en énergie est déjà très grande. Dans les deux cas le compromis énergie pertes se situe pour des valeurs de D comprises entre 1 et 9.

### **Interprétation modèle Exponentielle**

#### **Expérience 1 :**



**Figure 5:** *Heatmap du taux de perte modèle exponentielle de l'expérience 1*

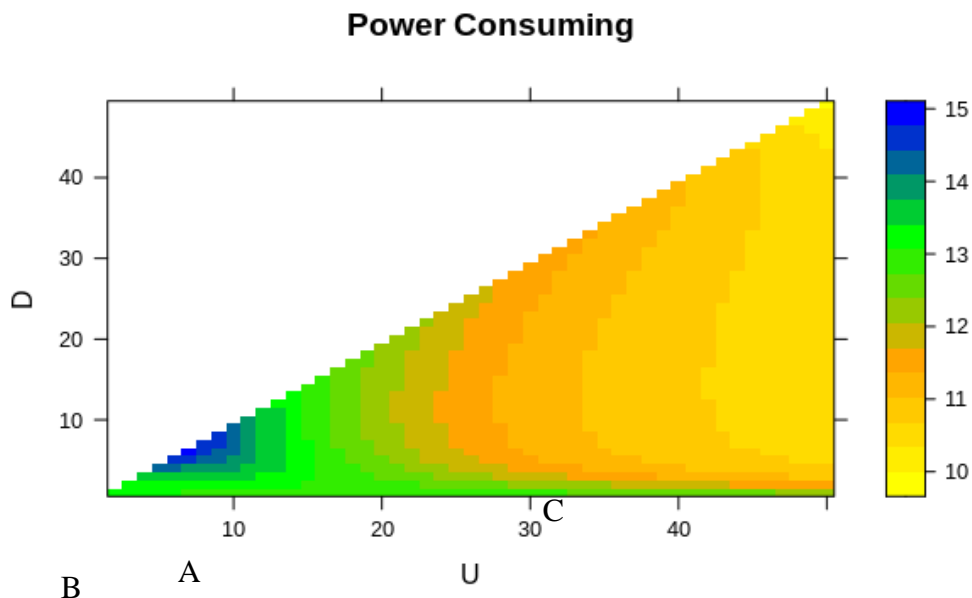
Le graphique ci-dessus décrit le taux de pertes des clients pour le modèle Exponentielle. Nous avons les valeurs de U (en abscisse) et D (en ordonnées) puis une troisième valeur représentant les pertes liées aux différents couples de valeur correspondants. Nous avons noté principalement 3 grandes zones correspondantes à une consommation pour ces différents couples de valeur, c'est à dire la zone A, C et D.

La zone A, en forte coloration jaune, nous indique que pour des couples de valeur U comprises entre 2 et 33, et D compris entre 1 et 23 le taux de pertes est quasiment nul. Pour ce qui est de la zone C, nous

remarquons un changement de couleur virant à l'orangée indiquant une augmentation du taux de pertes avec des valeurs de U comprises entre 40 et 44, avec D compris entre 36 et 42. Enfin la zone **D**, zone à risque pour des valeurs de U supérieures 44 et D supérieures 43, donnant un taux de pertes supérieur à 0.010, ainsi les valeurs associées à cette zone doivent être évitées.

**Tableau 5:** Récapitulatif du taux de pertes expérience 1 du modèle exponentielle

Expérience 1 :Heatmap Pertes			
Buffersize =50; AlwaysOn=10; OnandOff=5;Lamba=10 Moyenne Pertes=0.010.			
ZONE	U	D	CONCLUSION
A	2 à 33	1 à 23	Taux de perte quasiment nul, perte minime
C	40 à 44	36 à 42	Taux de pertes acceptables compris entre 0.002 et 0.010
D	> 44	> 43	Taux de pertes assez élevé c'est à dire supérieur à 0.020



**Figure 6:** Heatmap de la consommation d'énergie du modèle exponentielle de l'expérience 1

Le graphique ci-dessus décrit la consommation d'énergie pour le modèle Exponentielle. Nous avons les valeurs de U (en abscisse) et D (en ordonnées) puis une troisième valeur représentant les pertes liées aux différents couples de valeur correspondantes. Nous avons noté principalement 3 grandes zones correspondantes à une consommation pour ces différents couples de valeur, c'est à dire la zone A, B et C.

La zone **B**, en forte coloration bleu, nous indique que pour des couples de valeur U comprises entre 2 et 10, et D comprises entre 3 et 10 nous avons la consommation d'énergie qui est très élevée c'est à dire supérieur à 14. Pour ce qui est de la zone **A**, nous remarquons un changement de couleur virant au vert indiquant que la consommation d'énergie est assez élevée par rapport à la moyenne (12.5 ). Enfin, la zone **C**, zone a non risque pour des valeurs de U supérieures à 30 et D supérieures à 30, donnant une consommation d'énergie comprise entre [10, 11], ainsi dans cette zone les valeurs de U et D permettent de minimiser la consommation de l'énergie.

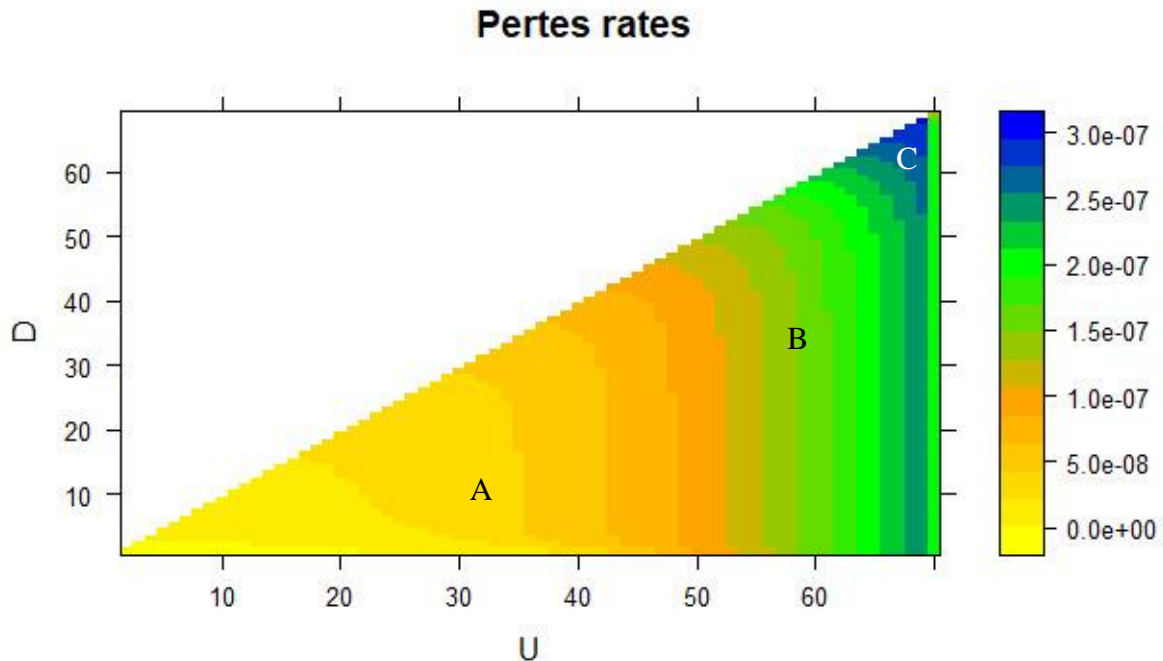
On pourrait prendre le compromis énergie- perte situé de 30 à 40. En effet dans ces deux cas l'énergie et la perte sont acceptables.

**Tableau 6:** *Récapitulatif de la consommation énergétique de l'expérience 1 du modèle exponentielle*

Expérience 1 :Heatmap Energie			
Buffersize =50; AlwaysOn=10; OnandOff=5;Lamba=10 Moyenne Energie=12.5			
ZONE	U	D	CONCLUSION
A	2 à 32	1 à 20	Consommation d'énergie assez élevée comprise compris entre 12.5 et 13.5
B	1 à 10	3 à 10	Consommation d'énergie très élevée supérieure à 14
C	> 30	> 30	Consommation d'énergie assez faible, comprise entre [10, 11]



## Expérience 2 :



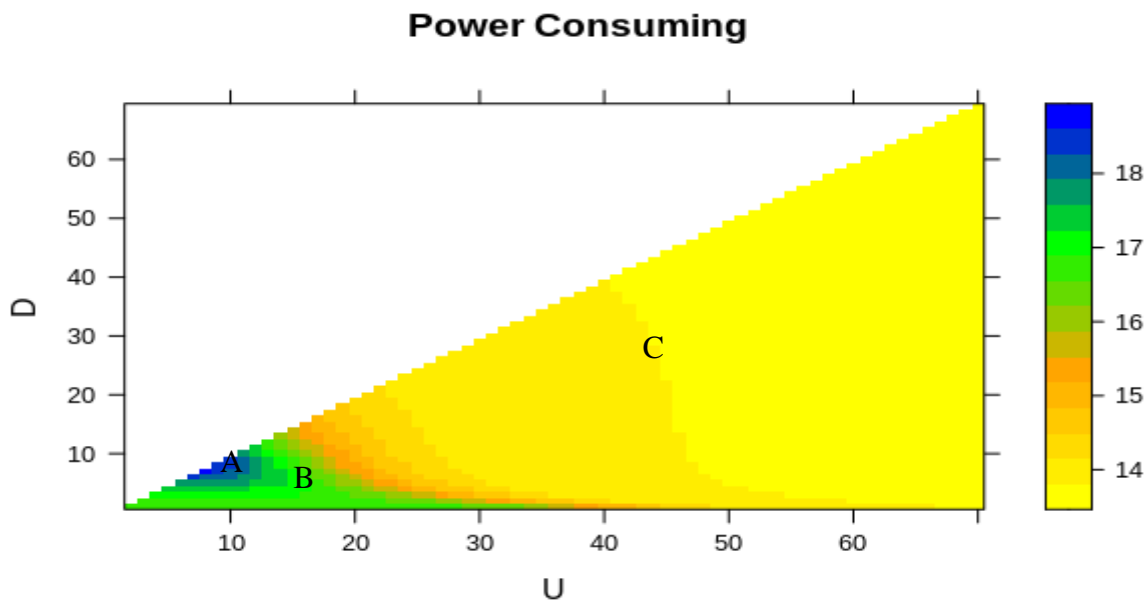
**Figure 7:** *Heatmap du taux de perte expérience 2 modèle exponentielle*

Le graphique ci-dessus décrit le taux de pertes des clients pour le modèle Exponentielle concernant l'expérience 2. Nous avons les valeurs de U (en abscisse) et D (en ordonnées) puis une troisième valeur représentant les pertes liées aux différents couples de valeur correspondantes. Nous avons noté principalement 3 grandes zones correspondant à une consommation pour ces différents couples de valeur, c'est à dire la zone A, B et C.

La zone **A**, en forte coloration jaune, nous indique que pour des couples de valeur U comprises entre 2 et 50, et D comprises entre 0 et 45 nous avons le taux des pertes qui sont quasiment nul (compris dans l'intervalle  $[0.0e+00, 1.0e-07]$ ). Pour ce qui est de la zone **B**, nous remarquons un changement de couleur virant à l'orangée indiquant une augmentation du taux des pertes avec des valeurs de U comprises entre 53 et 63, avec D comprises entre 0 et 55. Enfin la zone **C**, zone à risque pour des valeurs de U supérieures à 64 et D supérieures à 56, donnant un taux de pertes supérieur à  $2.5e-07$ , ainsi les valeurs associées à cette zone doivent être évité.

**Tableau 7:**Recapitulatif du taux de perte expérience 2 du modèle exponentielle

Expérience 2 :Heatmap Pertes			
Buffersize =70; AlwaysOn=15; OnandOff=5;Lamba=12 Moyenne Pertes=1.5 e-07.			
ZONE	U	D	CONCLUSION
A	2 à 50	0 à 45	Taux de pertes quasiment nul, minime
B	53 à 63	0 à 55	Taux de perte acceptable compris entre 0.0e+00 et 1.0e-07
C	> 64	> 56	Taux de pertes assez élevée c'est à dire > 2.5e-07



**Figure 8:** Heatmap de la consommation d'énergie expérience 2 du modèle exponentielle

Le graphique ci-dessus décrit la consommation d'énergie pour le modèle Exponentielle. Nous avons les valeurs de U (en abscisse) et D (en ordonnées) puis une troisième valeur représentant les pertes liées aux différents couples de valeur correspondantes. Nous avons noté principalement 3 grandes zones correspondant à une consommation pour ces différents couples de valeur, c'est à dire la zone A, B et C. Pour ce qui est de la zone **A**, en forte coloration bleu, nous indique que pour des couples de valeur U comprises entre 8 et 12, et D comprises entre 8 et 10 nous avons la consommation d'énergie qui est très élevée c'est à dire supérieur 18.

La zone **B**, nous remarquons un changement de couleur virant au vert indiquant un bon compromis de la consommation d'énergie par les serveurs. Enfin la zone **C**, zone a non risque pour des valeurs de U

supérieures à 40 et D supérieures à 15, donnant une consommation d'énergie comprise entre [14, 15], ainsi dans cette zone les valeurs de U et D permettent de minimiser la consommation de l'énergie.

**Tableau 8:** Récapitulatif de la consommation d'énergie de l'expérience 2 du modèle exponentielle

Expérience 2 :Heatmap Energie			
Buffersize =70; AlwaysOn=15; OnandOff=5;Lamba=12 Moyenne Energie=16.			
ZONE	U	D	CONCLUSION
A	8 à 12	0 à 13	Consommation d'énergie assez élevée c'est à dire supérieure à 18
B	0 à 39	0 à 10	Consommation d'énergie acceptable comprise entre 16 et 17
C	> 40	> 15	Consommation d'énergie quasiment nulle, minime

**Tableau 9:** Liste des valeurs de U et D pour un compromis énergie pertes.

Modèle Erlang				Modèle exponentielle			
Expérience 1		Expérience 2		Expérience 1		Expérience 2	
U	D	U	D	U	D	U	D
[2 ; 12]	[1 ; 8]	[2 ; 12]	[1 ; 10]	[2 ; 40]	[1 ; 20]	[2 ; 47]	[1 ; 30]
[13,15]	[1 ; 6]	13	[1 ; 9]	[24 ; 50]	2, 8, 10	[48 ; 52]	[30 ; 45]
16, 17, 21	[1 ; 5]	14,15	[1 ; 8]	[24 ; 50]	[1 ; 50]	[38; 70]	[1 ; 70]
[18 ; 20]	[1 ; 4]	16	[1 ; 7]			[15 ; 20]	[3 ; 10]
[22 ; 28]	[1 ; 3]	17,18	[1 ; 6]			[20 ; 38]	[3 ; 20]
[29 ; 50]	[1 ; 2]	19,20	[1 ; 5]				
		21 ,22	[1 ; 4]				
		[23 ; 26]	[1 ; 3]				
		[27 ; 31]	[1 ; 2]				
		[32 ; 39]	1				

Le tableau ci-dessus illustre un aperçu de certains couple de valeurs U et D qui offre un bon compromis énergie perte dans les deux modèles d'allumage.

## Comparaison des deux modèles

Avant toute comparaison, il est important de rappeler certaines propriétés en relation avec les deux lois d'allumages. En effet, elles ont la même moyenne mais pas le même coefficient de variation.

De ce fait, le coefficient de variation de la loi Erlang vaut  $r = (1/k)$  avec  $k$  étant le paramètre  $k$  d'une Erlang  $(k,B)$ , représentant la somme de  $k$  variables aléatoires indépendantes et identiquement distribuées alors que le coefficient de variation de la loi Exponentielle vaut  $r = 1$ .

Par conséquent, le coefficient de variation de l'erlang est inférieur ou égale à celui de l'Exponentielle.

Donc on compare l'allumage des serveurs de réserves en utilisant des lois ayant des coefficients de variation différents mais de même moyenne.

### Comparaison de l'expérience 1 et 2 du modèle Exponentielle :

Pour l'expérience 1, nous avons la valeur des variables suivantes qui valent :

- **Buffersize = 50**
- **Lambda = 10**
- **AlwaysOn = 10**
- **OnandOff = 5**

Pour l'expérience 2 nous avons :

- **Buffersize = 70**
- **Lambda = 12**
- **AlwaysOn = 15**
- **OnandOff = 5.**

### Heatmap des pertes

Avec l'expérience 1, nous avons observé que, le taux de pertes est quasiment nul pour des valeurs de  $U$  comprises dans  $[1, 33]$  et  $D$  dans  $[0, 35]$ , des pertes acceptables et comprises entre 0.002 et 0.010 pour des valeurs de  $U$  comprises dans  $[40, 44]$  et  $D$  comprise dans  $[36, 42]$ .

Enfin, nous avons remarqué que nous avons des taux de pertes très élevé c'est à dire supérieur à 0.01 pour des valeurs de  $U$  supérieures à 44 et  $D$  supérieures à 43.

Au vu de l'expérience 2, nous avons constaté qu'avec l'augmentation du Buffersize a 70, du AlwaysOn (le nombre de serveurs principaux toujours allumés) à 15, le Lambda (le taux d'arrivée par unité de temps) à 12 une très grande amélioration des pertes au niveau de l'expérience, c'est à dire :

Le taux de pertes maximal dans l'expérience 2 est inférieur au taux de perte minimal dans l'expérience 1, de ce fait si nous augmentons la capacité de la file, ainsi que le nombre de serveurs principaux et lambda, alors nous pouvons réduire considérablement le taux de pertes des clients.

## Heatmap des énergies

Avec l'expérience 1, nous avons observé que, la consommation d'énergie est quasiment nulle pour des valeurs de  $U$  supérieures à 30 et  $D$  supérieures à 30, une consommation acceptables pour des valeurs de  $U$  comprises dans  $[1, 32]$  et  $D$  dans  $[0, 20]$ .

Enfin, nous remarquons que la consommation est très élevée c'est à dire supérieure à 14 pour des valeurs de  $U$  comprises dans  $[1, 10]$  et  $D$  comprises dans  $[3, 10]$ .

Au vu de l'expérience 2, nous avons constaté qu'avec l'augmentation du Buffersize à 70, du AlwaysOn (le nombre de serveurs principaux toujours allumés) à 15, le Lambda à 12, nous avons remarqué une nette amélioration des pertes au niveau de l'expérience, c'est à dire :

Pour les couples de valeurs  $U$  comprises dans  $[5, 12]$  et  $D$  comprise dans  $[5, 10]$  la consommation d'énergie a augmenté considérablement, c'est à dire compris de  $[14, 15]$  dans l'expérience 1 et supérieur à 18 dans l'expérience 2. De même, la consommation d'énergie avec les différents couples possibles a augmentée considérablement, cela nous indique que malgré l'augmentation de la capacité de la file, le nombre de serveurs principaux et le taux d'arrivé des clients, cela n'améliore pas la consommation d'énergie mais, plutôt augmente considérablement la consommation.

## Comparaison de l'expérience 1 de chaque modèle

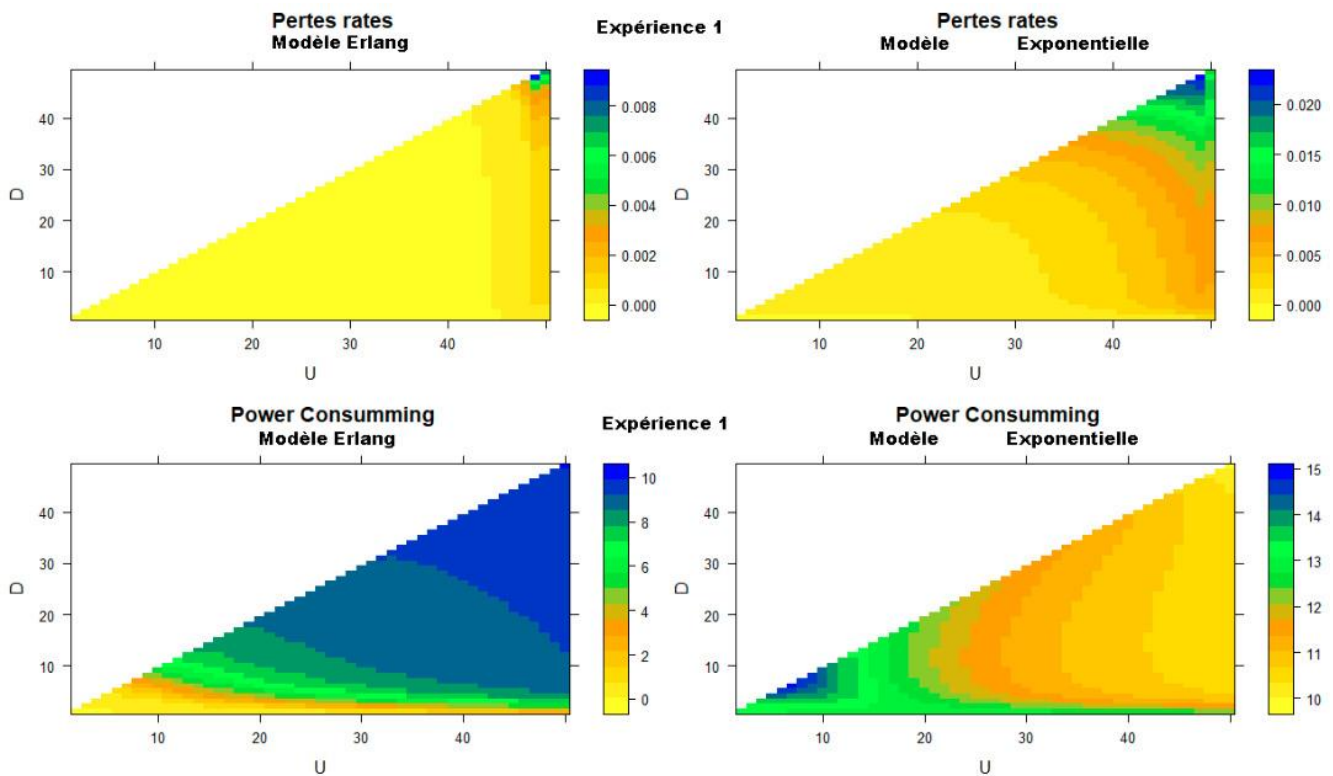
### Comparaison des heatmap de pertes

En observant le résultat des deux expériences on note que le modèle exponentielle à un taux de perte qui croit plus vite que celui du modèle erlang. En effet, l'amplitude maximale dans le modèle exponentielle est de **0.020** pendant que celui dans le modèle erlang est de **0.008**. Egalement, pour des valeurs de couples  $U$  compris entre 1 et 25 le taux de perte reste pareil, cependant pour  $U$  compris entre 26 et 50 pour  $D$  variant entre 22 et 36 nous notons une nette augmentation du taux de perte pour le modèle exponentielle qui est maintenant compris entre 0 et 0.010. Aussi, pour de grandes valeurs de  $U$  et  $D$  le modèle exponentielle produit un taux de perte plus considérable, cela est visible en observant les zones C de chaque graphiques, c'est-à-dire pour  $U$  compris entre 40 et 50 et  $D$  variant entre 20 et 50 pour le modèle exponentielle ou encore  $U$  compris entre 48 et 50 et  $D$  variant entre 40 et 50.

De ce fait, pour obtenir un compromis pertes énergie rentable il est obligatoire d'éviter d'utiliser des couples de valeur élevée pour le modèle exponentielle. Le modèle erlang pour les pertes dispose d'un taux de perte maximale faible par rapport à celle de l'exponentielle. En résumé, le modèle erlang est un choix optimal pour l'optimisation du taux de perte.

## Comparaison heatmap énergie

Dans l'expérience 1 du modèle Erlang on constate que pour les couples de valeurs  $U$  compris entre  $[20 ; 50]$  et  $D$  entre  $[4 ; 49]$  la consommation en énergie augmente cependant dans le modèle exponentielle pour les même intervalles de  $U$  et  $D$  la consommation en énergie diminue. Nous pouvons aussi constater que le modèle Erlang a une consommation d'énergie maximale de 10 et une consommation minimale de 0, par contre celui du modèle exponentielle a une consommation minimale égale à 10. Par conséquent, le modèle Erlang est un meilleur choix afin d'assurer une consommation minimale en énergie des serveurs.



**Figure 9:** Heatmap des expérience 1 de chaque modèle

**Tableau 10:** Récapitulatif de la comparaison du modèle erlang et du modèle exponentielle expérience 1

Récapitulatif de la comparaison des expériences 1 de chaque modèle	
Expérience1 modèle exponentielle	Expérience1 modèle erlang
<ul style="list-style-type: none"><li>• Le taux de perte croît vite ;</li><li>• Amplitude maximale de <b>0.02</b> ;</li><li>• Les couples de valeur U et D élevés sont à éviter afin d'assurer un bon compromis pertes énergie.</li><li>• Taux de perte maximale élevé par rapport à celui du modèle erlang.</li><li>• Consommation d'énergie minimale est de 10.</li><li>• Consommation d'énergie décroît pour des couples de valeurs U et D élevés. En effet, plus les couples de valeurs U et D sont élevés plus la consommation d'énergie est faible.</li></ul>	<ul style="list-style-type: none"><li>• Le taux de perte croît moins vite ;</li><li>• Amplitude maximale de <b>0.008</b> ;</li><li>• Un taux de perte acceptable pour un très grand nombre de valeur U et D de ce fait il est facile de trouver un bon compromis perte énergie.</li><li>• Taux de perte maximale faible par rapport à celle de l'exponentielle.</li><li>• Consommation d'énergie maximale est de 10.</li><li>• Consommation d'énergie croît pour des couples de valeurs U et D élevés. En effet, plus les couples de valeurs U et D sont élevés plus la consommation d'énergie augmente.</li></ul>

## Comparaison de l'expérience 2 de chaque modèle

### Heatmap des pertes

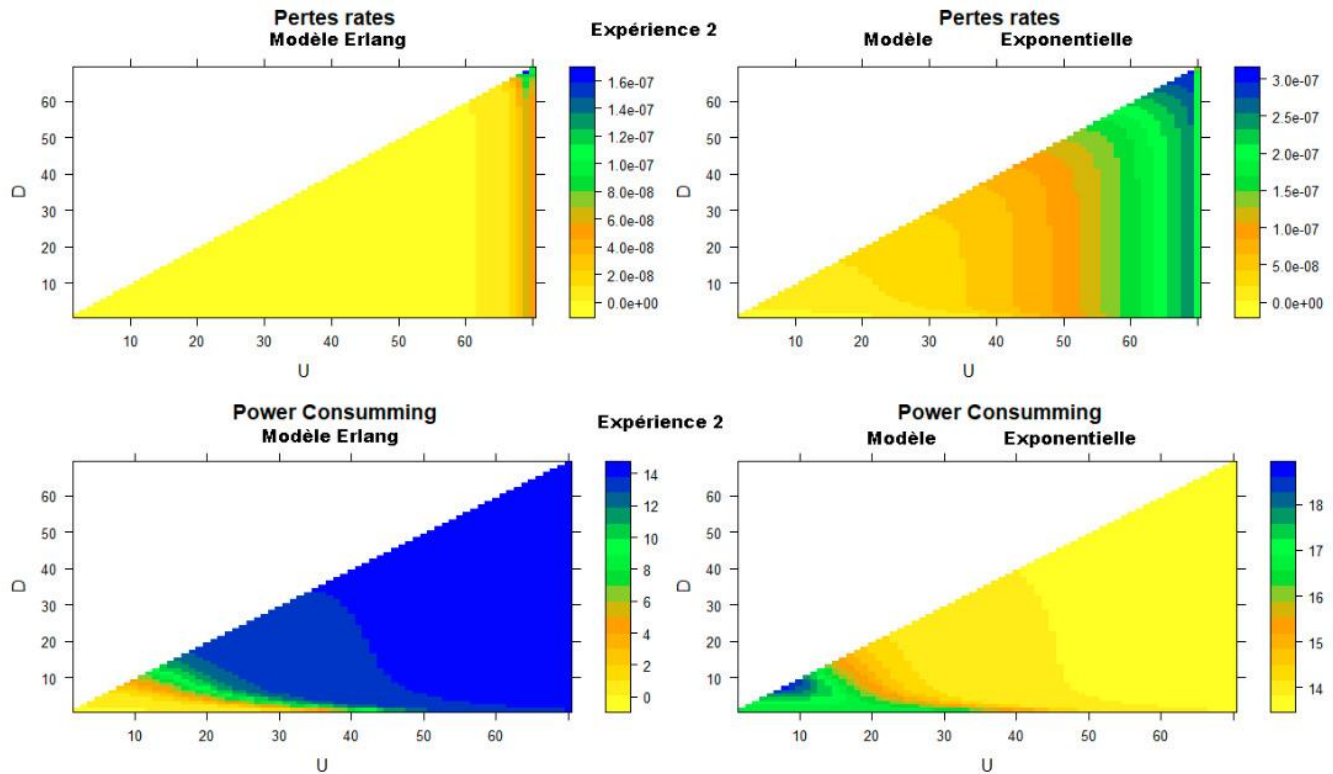
Dans l'expérience 2 du modèle Erlang on constate que pour les couples de valeurs U et D compris entre 1 et 65, les pertes sont quasiment nulles, ce qui n'est pas le cas pour le modèle Exponentielle. En effet, c'est seulement pour les couples de valeurs U compris entre 1 et 35 du modèle exponentielle que les pertes sont nulles, de ce fait on peut dire qu'avec le choix du modèle Erlang on a un large panel dans l'utilisation des couples de valeur U et D. Egalement, le taux de perte dans le modèle Exponentielle croît considérablement pour les valeurs de U compris 35 et 70. Au regard des différentes observations le choix du modèle erlang est celui qui minimise le plus le taux de perte.

De ce fait, pour obtenir un bon compromis pertes énergie il est conseillé d'éviter d'utiliser des couples de valeur élevée pour les deux modèles.

### Heatmap de l'énergie

Nous pouvons constater que dans l'expérience 2 du modèle Exponentielle la consommation d'énergie maximale vaut 18 et la consommation minimale vaut 14 pour les différents couples de valeur U et D ce qui est largement supérieur à celui du modèle Erlang, en effet sa consommation maximale est égale à

14. Par conséquent le choix du modèle Erlang est meilleur afin d’assurer une consommation d’énergie minimale des serveurs.



**Figure 10:** Heatmaps expérience 2 de chaque modèle

**Tableau 11:** Récapitulatif comparaison du modèle erlang et du modèle exponentielle expérience 2

Récapitulatif comparaison des expériences 2 de chaque modèle	
Expérience2 modèle exponentielle	Expérience2 modèle Erlang
<ul style="list-style-type: none"> <li>Le taux de perte croit assez vite ;</li> <li>Le taux de perte maximale est de 3.0e-07 ;</li> <li>Le choix des couples de valeur U et D élevées sont à éviter afin d’assurer un bon compromis entre les pertes et la consommation énergie ;</li> <li>Les pertes maximales sont élevées par rapport à celui du modèle Erlang ;</li> <li>Consommation d’énergie minimale est de 14 ce qui correspond à la consommation maximale d’énergie au niveau du modèle Erlang.</li> </ul>	<ul style="list-style-type: none"> <li>Le taux de perte croit moins vite ;</li> <li>Le taux de perte maximale est de 1.6e-07 ;</li> <li>Un taux de perte acceptable pour un très grand nombre de valeur U et D c’est à dire entre 0 et 65, de ce fait il est facile de trouver un bon compromis perte énergie ;</li> <li>Les pertes maximales sont faibles par rapport à celle de l’exponentielle ;</li> <li>Taux de perte optimal par rapport au modèle exponentielle ;</li> <li>Consommation d’énergie maximale est de 14.</li> </ul>



## Conclusion

Dans le cadre de notre projet de travaux d'étude et de recherche il nous a été confié une étude sur la politique à seuil pour l'optimisation énergétique des serveurs. Cette étude nous a permis d'aborder plusieurs notions techniques comme le modèle de politique à seuil Mitrani, les différents états des serveurs, les lois d'allumages, l'utilisation du langage de programmation C, R et la réalisation de cartes à chaleur (Heatmap). Dans le souci de suivre une méthode scientifique nous avons effectué plusieurs expériences dans le but d'observer, analyser et interpréter les résultats obtenus. A l'issue de ces différentes expériences nous pouvons affirmer avec certitude que l'étude des deux modèles respectifs exponentiel et erlang ont des caractéristiques différentes sur le plan de l'évolution des pertes, de l'énergie, mais aussi des différents intervalles dans le choix d'un compromis visant à minimiser la consommation d'énergie et la perte de clients. Par conséquent, après étude il est à noter que le modèle erlang s'avère être un meilleur choix dans la mesure où elle réduit considérablement le taux de pertes mais également la consommation en énergie des serveurs.

## LISTE DES REFERENCES BIBLIOGRAPHIQUES

- [1] **Isi Mitrani**. Service center trade-offs between customer impatience and power consumption. Elsevier Performance Evaluation (2011) 68 : 1222-1231, July 2011.
- [2] [Lien github vers la documentation du projet](#).