

Rien ne sert de prédire ; il faut servir ancien.

Céline Comte

NOKIA Bell Labs



AlgoTel 2019 – 4 juin 2019

Introduction

Aéroport

www.bostonglobe.com



Introduction

Aéroport



www.bostonglobe.com

Supermarché



www.sellerschaefer.com

Introduction

Aéroport



www.bostonglobe.com

Supermarché



www.sellerschaefer.com



i chokedonmylatte.wordpress.com

Vidéo à la demande

Introduction

Aéroport



www.bostonglobe.com

Supermarché

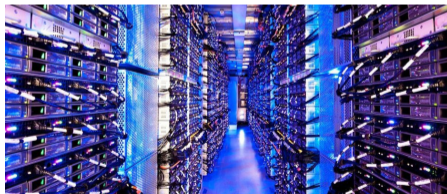


www.sellerschaefer.com



i.chokedonmylatte.wordpress.com

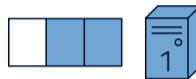
Vidéo à la demande



www.cnet.com

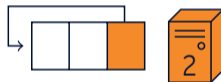
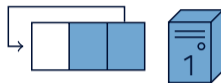
Cluster d'ordinateurs

Abstraction



Abstraction

Round-robin



Abstraction



Abstraction

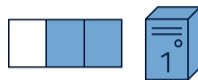
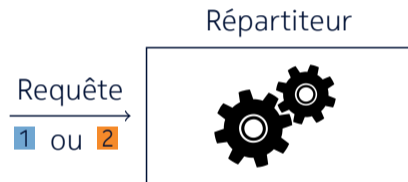
Répartiteur



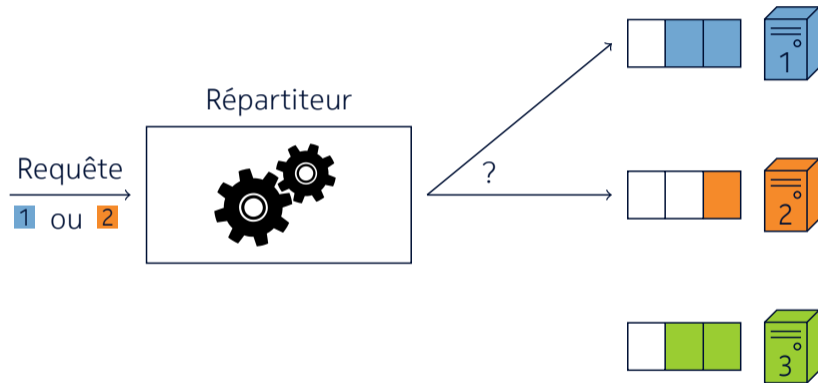
Abstraction



Abstraction



Abstraction



Contributions

Algorithme de répartition de la charge

Simple, insensible et adaptatif

Contributions

Algorithme de répartition de la charge

Simple, insensible et adaptatif

Modèle de file d'attente

En supposant que les requêtes arrivent selon un processus de Poisson

Contributions

Algorithme de répartition de la charge

Simple, insensible et adaptatif

Modèle de file d'attente

En supposant que les requêtes arrivent selon un processus de Poisson

Résultats numériques

Comparaison avec des politiques insensibles statiques

Contributions

Algorithme de répartition de la charge

Simple, insensible et adaptatif

Modèle de file d'attente

En supposant que les requêtes arrivent selon un processus de Poisson

Résultats numériques

Comparaison avec des politiques insensibles statiques

Lien entre plusieurs travaux existants

(discuté dans cette présentation et dans la version longue de l'article)

Algorithme

Modèle de files d'attente

Résultats numériques

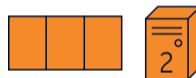
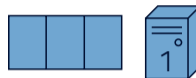
Algorithme

Modèle de files d'attente

Résultats numériques

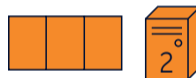
Algorithme

Répartiteur

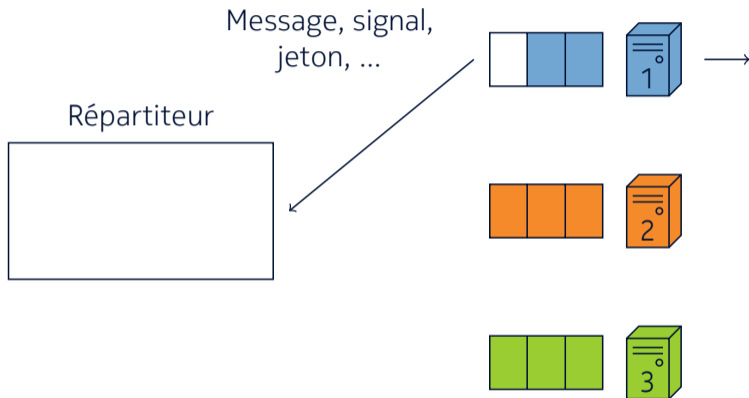


Algorithme

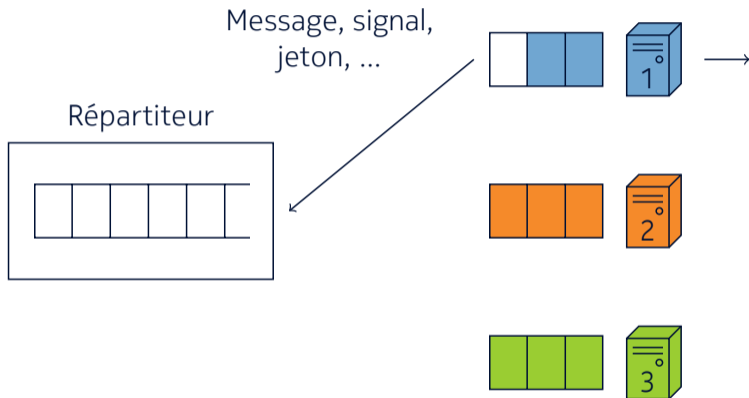
Répartiteur



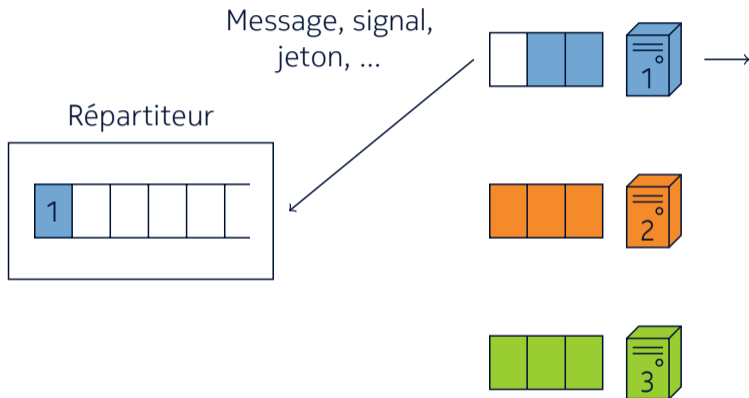
Algorithme



Algorithme

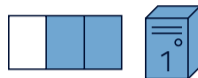
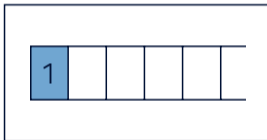


Algorithme



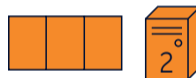
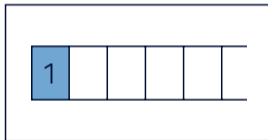
Algorithme

Répartiteur

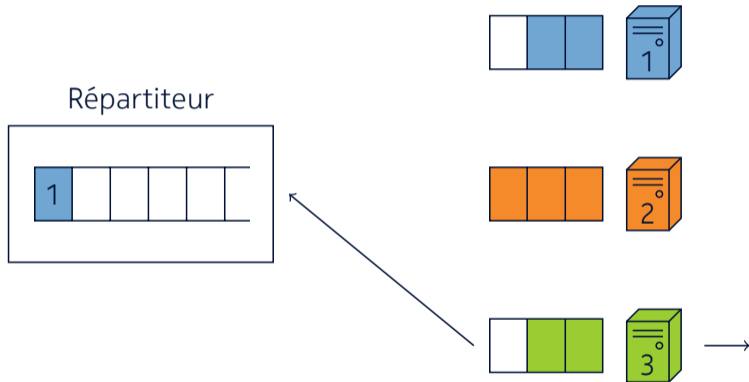


Algorithme

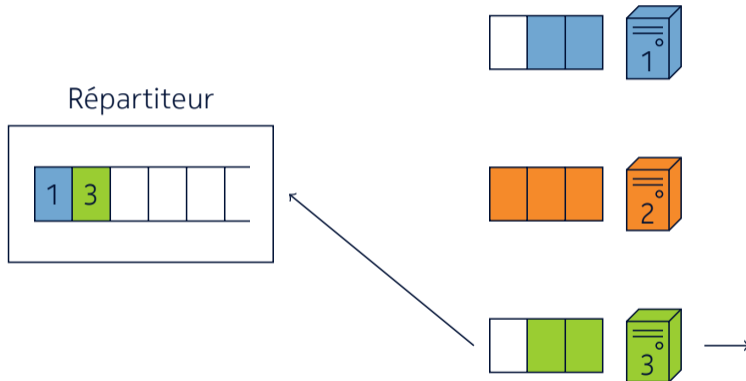
Répartiteur



Algorithme

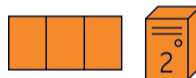
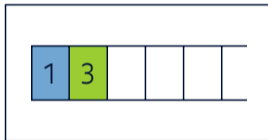


Algorithme



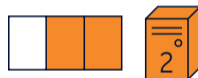
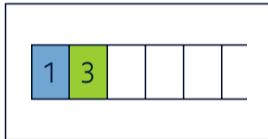
Algorithme

Répartiteur

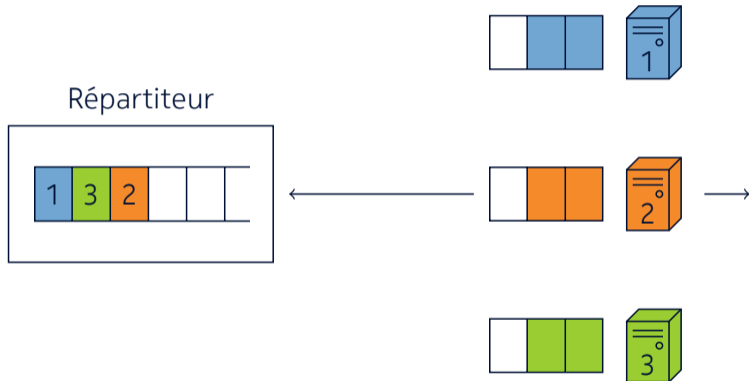


Algorithme

Répartiteur

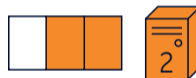
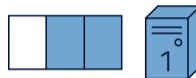
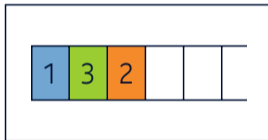


Algorithme



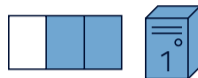
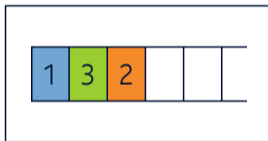
Algorithme

Répartiteur

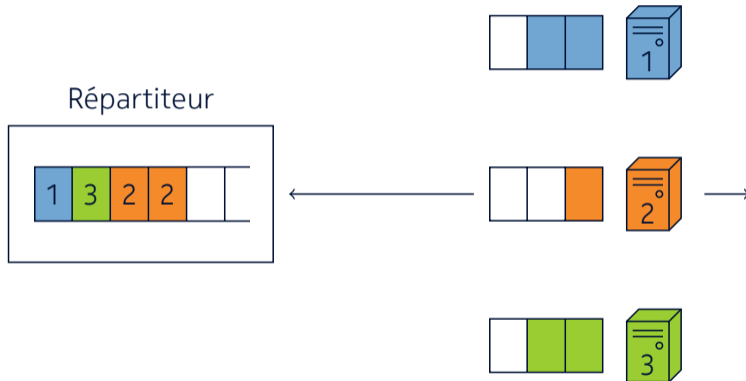


Algorithme

Répartiteur

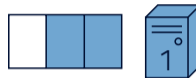


Algorithme

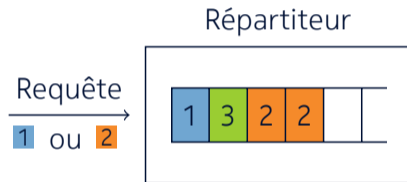


Algorithme

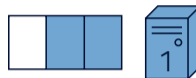
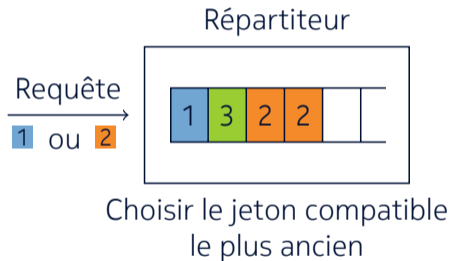
Répartiteur



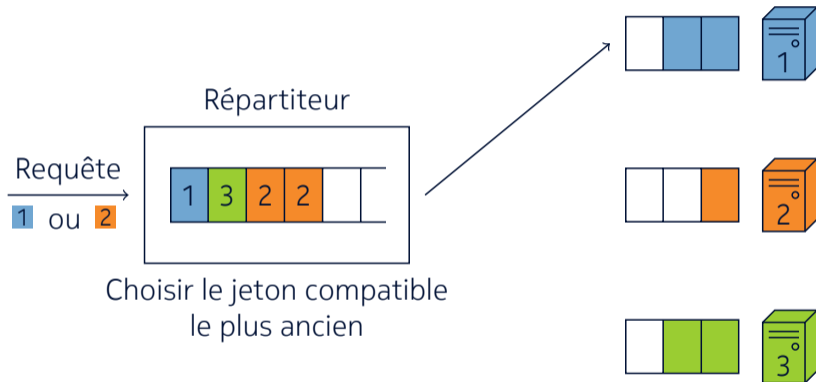
Algorithme



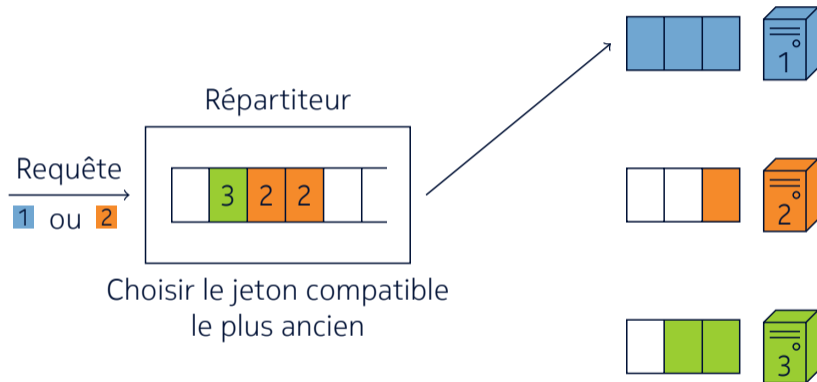
Algorithme



Algorithme

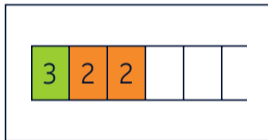


Algorithme

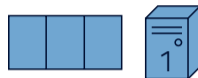


Algorithme

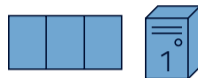
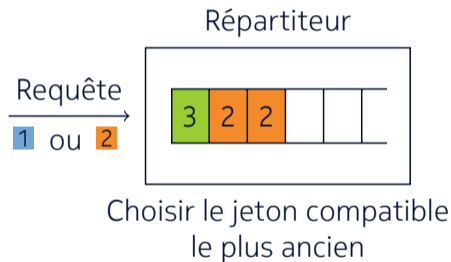
Répartiteur



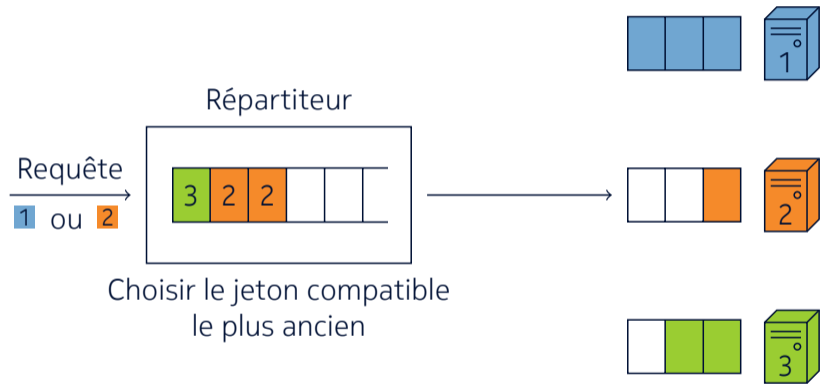
Choisir le jeton compatible
le plus ancien



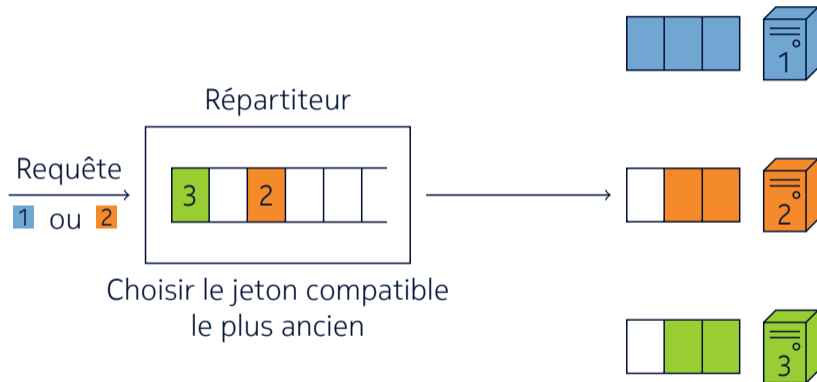
Algorithme



Algorithme

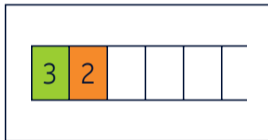


Algorithme

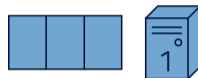


Algorithme

Répartiteur



Choisir le jeton compatible
le plus ancien



Algorithmes reliés

Assign-to-the-longest-idle-server (Adan et Weiss, 2012) :

Choisir la machine compatible inoccupée depuis le plus longtemps

Algorithmes reliés

Assign-to-the-longest-idle-server (Adan et Weiss, 2012) :

Choisir la machine compatible inoccupée depuis le plus longtemps (cette machine étant probablement la moins chargée).

Algorithmes reliés

Assign-to-the-longest-idle-server (Adan et Weiss, 2012) :

Choisir la machine compatible inoccupée depuis le plus longtemps (cette machine étant probablement la moins chargée).

Join-Idle-Queue (Lu et al., 2011) :

Choisir en priorité une machine inoccupée, sinon uniformément au hasard.

Algorithmes reliés

Assign-to-the-longest-idle-server (Adan et Weiss, 2012) :

Choisir la machine compatible inoccupée depuis le plus longtemps (cette machine étant probablement la moins chargée).

Join-Idle-Queue (Lu et al., 2011) :

Choisir en priorité une machine inoccupée, sinon uniformément au hasard.

Round-robin (pour la répartition de charge) :

En l'absence de contraintes, choisir les machines de façon cyclique.

Algorithmes reliés

Assign-to-the-longest-idle-server (Adan et Weiss, 2012) :

Choisir la machine compatible inoccupée depuis le plus longtemps (cette machine étant probablement la moins chargée).

Join-Idle-Queue (Lu et al., 2011) :

Choisir en priorité une machine inoccupée, sinon uniformément au hasard.

Round-robin (pour la répartition de charge) :

En l'absence de contraintes, choisir les machines de façon cyclique.

Join-the-Shortest-Queue :

Choisir la machine qui a le moins de requêtes en cours.

Insensibilité à la loi de la taille des requêtes

Si les requêtes arrivent selon un processus de Poisson

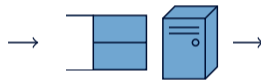
Une seule machine :

Premier arrivé, premier servi



Sensible

Processor-sharing



Insensible

Insensibilité à la loi de la taille des requêtes

Si les requêtes arrivent selon un processus de Poisson

Une seule machine :

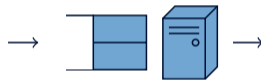
Premier arrivé, premier servi



Sensible

— Round-robin —

Processor-sharing



Insensible

Insensibilité à la loi de la taille des requêtes

Si les requêtes arrivent selon un processus de Poisson

Une seule machine :

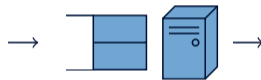
Premier arrivé, premier servi



Sensible

— Round-robin —

Processor-sharing



Insensible

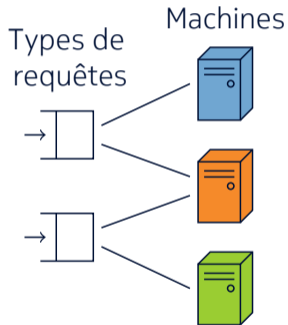
Plusieurs machines : Le modèle de files d'attente va montrer que l'insensibilité de processor-sharing est préservée par notre algorithme.

Algorithme

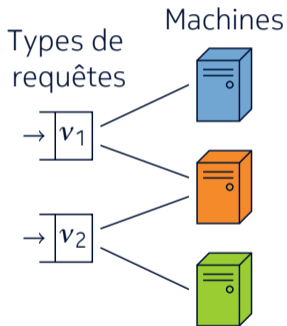
Modèle de files d'attente

Résultats numériques

Grappe de machine



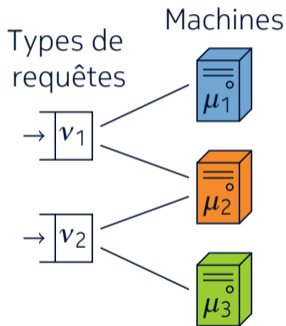
Graphe de machine



Arrivées :

- Les requêtes de type k arrivent au taux ν_k
- Processus d'arrivée Poisson

Grappe de machine



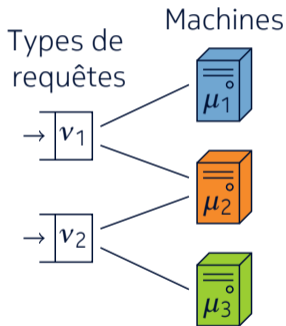
Arrivées :

- Les requêtes de type k arrivent au taux v_k
- Processus d'arrivée Poisson

Service :

- La machine s a une capacité μ_s
- Tailles des requêtes : loi de Cox (denses dans l'ensemble des lois de variables aléatoires à valeurs positives)

Grappe de machine



Arrivées :

- Les requêtes de type k arrivent au taux v_k
- Processus d'arrivée Poisson

Service :

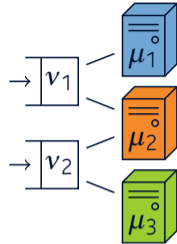
- La machine s a une capacité μ_s
- Tailles des requêtes : loi de Cox (denses dans l'ensemble des lois de variables aléatoires à valeurs positives)

Limite d'admission :

- La machine s possède ℓ_s jetons

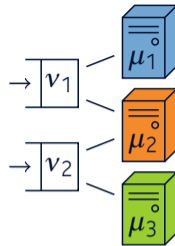
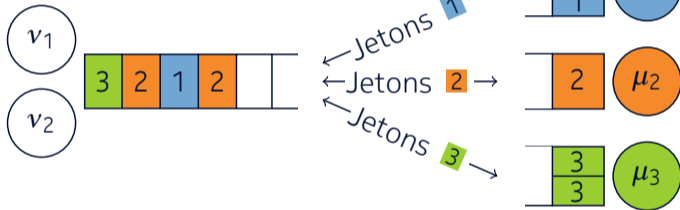
Réseaux fermé de files d'attente

Décrire le mouvement des jetons



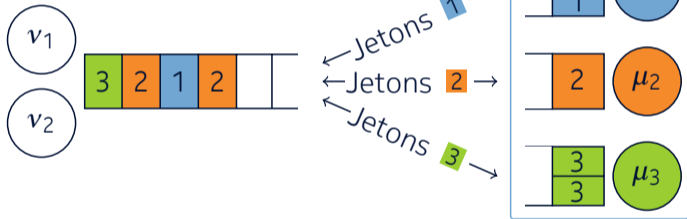
Réseaux fermé de files d'attente

Décrire le mouvement des jetons

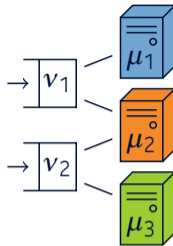


Réseaux fermé de files d'attente

Décrire le mouvement des jetons

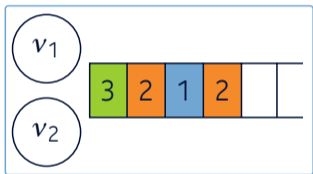


Jetons en service
(Files M/M/1/PS)

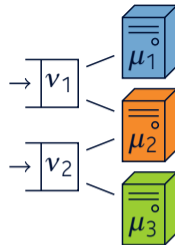
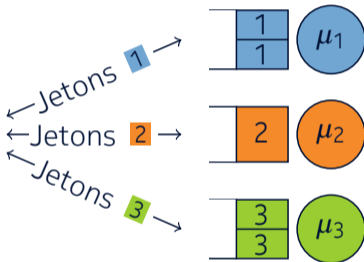


Réseaux fermé de files d'attente

Décrire le mouvement des jetons

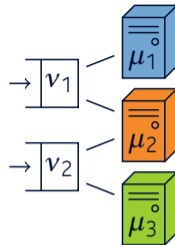
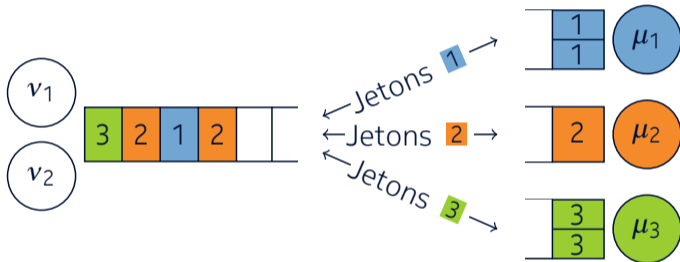


Jetons libres
(File “order-independent”)



Réseaux fermé de files d'attente

Décrire le mouvement des jetons



Aperçu de l'analyse stationnaire

Si la taille des requêtes suit une loi exponentielle :

- Analyse basée sur les files “order-independent” (Berezner et al., 1995).
- Réseau à forme produit → Formule explicite pour la distribution stationnaire.

Aperçu de l'analyse stationnaire

Si la taille des requêtes suit une loi exponentielle :

- Analyse basée sur les files “order-independent” (Berezner et al., 1995).
- Réseau à forme produit → Formule explicite pour la distribution stationnaire.

Si la taille des requêtes suit une loi de Cox :

- Analyse basée sur la méthode des étapes (“method of stages”).
- La distribution de l'état du réseau en régime stationnaire est inchangée.

Aperçu de l'analyse stationnaire

Si la taille des requêtes suit une loi exponentielle :

- Analyse basée sur les files “order-independent” (Berezner et al., 1995).
- Réseau à forme produit → Formule explicite pour la distribution stationnaire.

Si la taille des requêtes suit une loi de Cox :

- Analyse basée sur la méthode des étapes (“method of stages”).
- La distribution de l'état du réseau en régime stationnaire est inchangée.

Répartition moyenne de la charge :

Le taux d'arrivée moyen aux machines, ignorant l'ordre des jetons libres, est comme défini par la politique insensible de (Bonald et al., 2004).

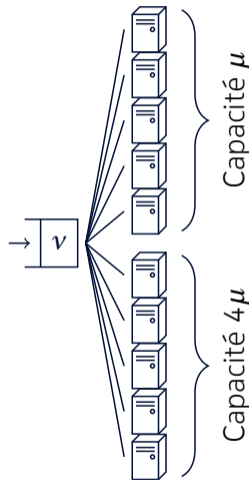
Algorithme

Modèle de files d'attente

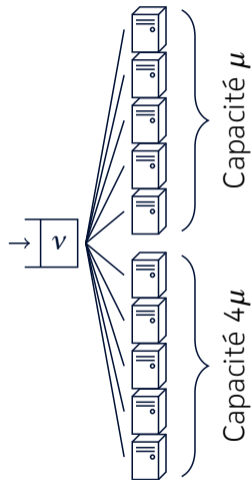
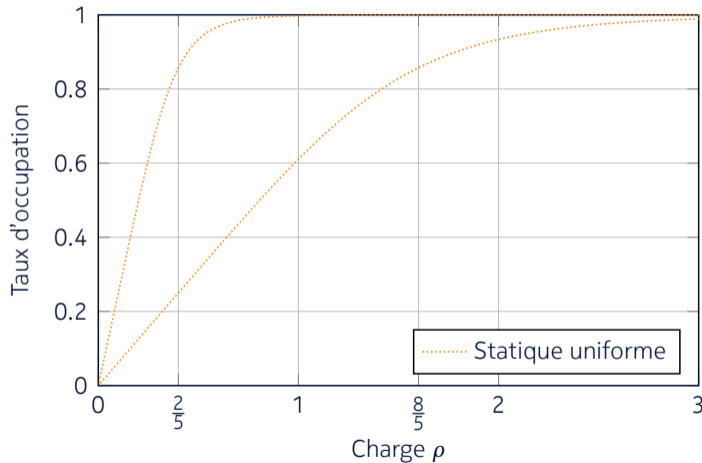
Résultats numériques

Machines avec des capacités inégales

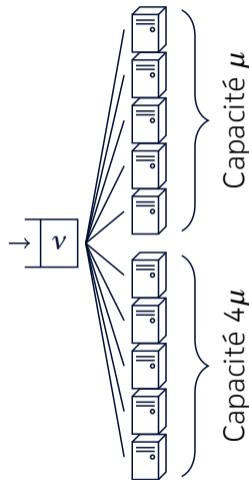
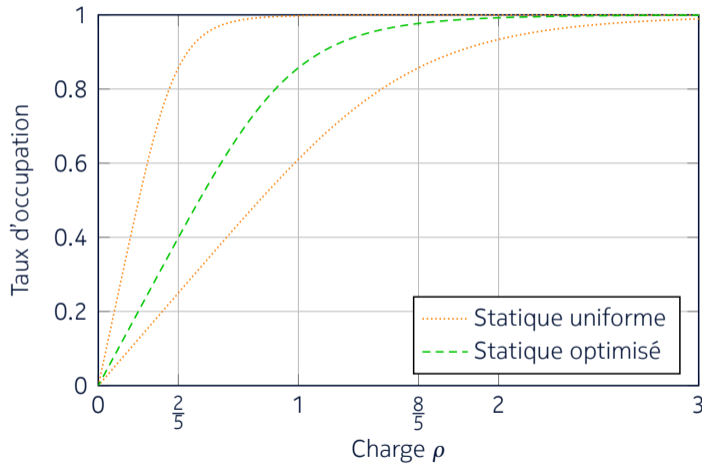
- Pas de contraintes de compatibilités :
Toutes les requêtes peuvent être affectée à toutes les machines
- Arrivées selon un processus de Poisson de taux ν
- La moitié des machines ont une capacité μ ,
l'autre moitié une capacité 4μ
- Chaque machine a 6 jetons
- Charge $\rho = \frac{\nu}{25\mu}$



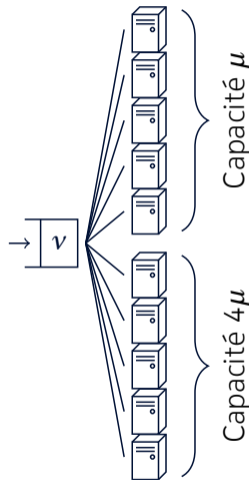
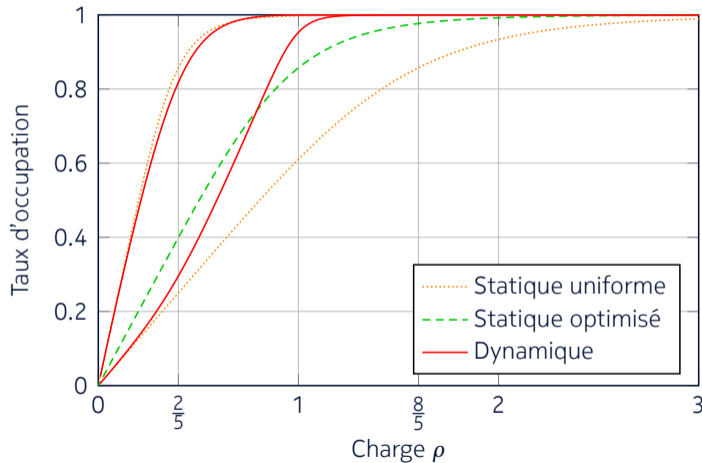
Machines avec des capacités inégales



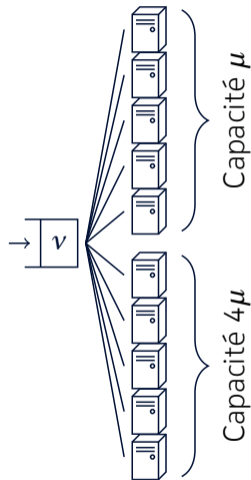
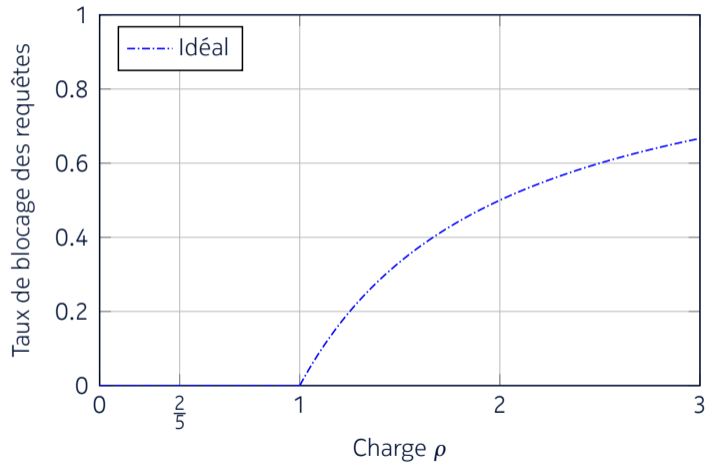
Machines avec des capacités inégales



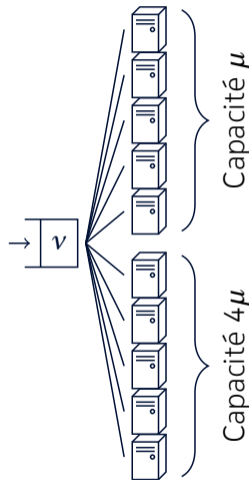
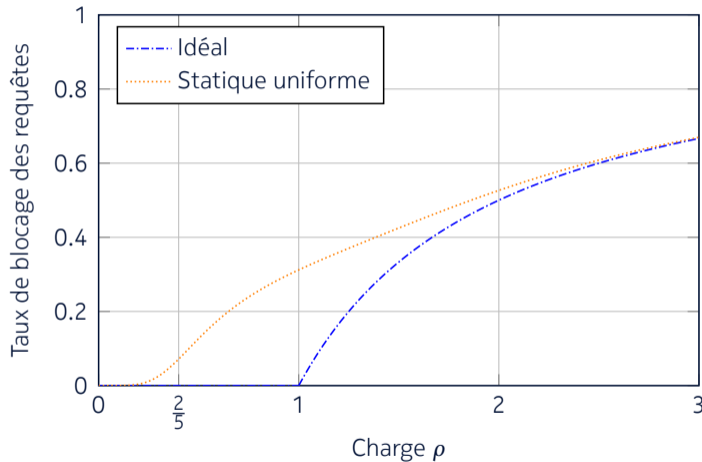
Machines avec des capacités inégales



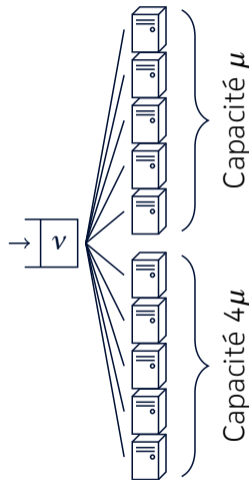
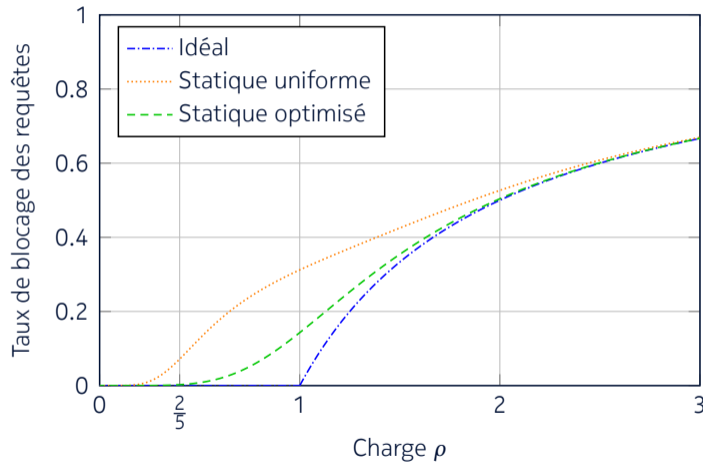
Machines avec des capacités inégales



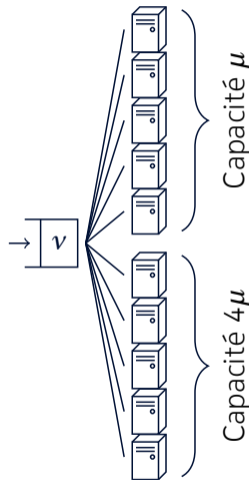
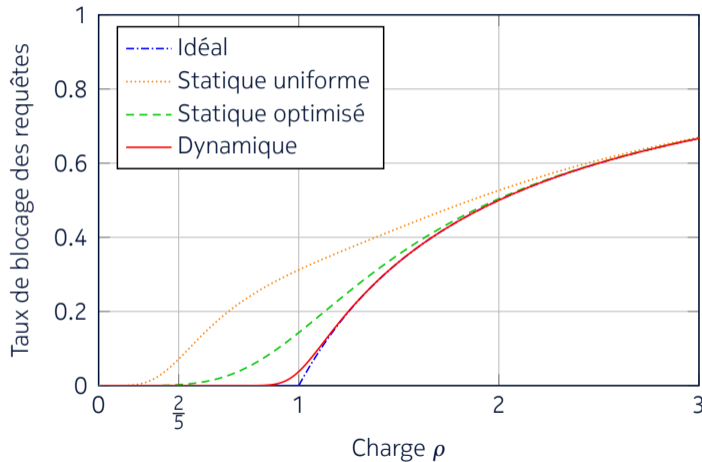
Machines avec des capacités inégales



Machines avec des capacités inégales

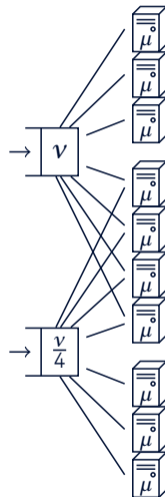


Machines avec des capacités inégales

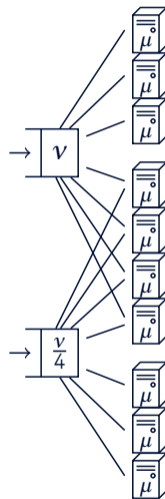
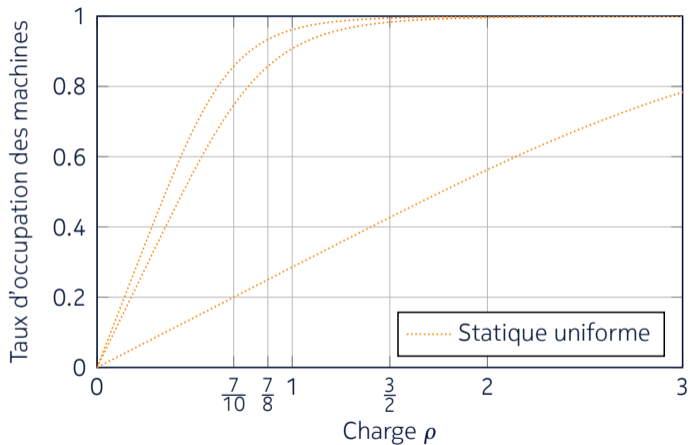


Contraintes d'affectations

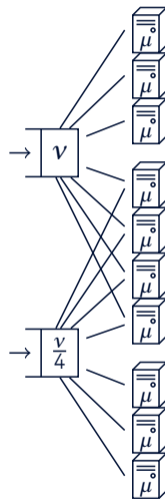
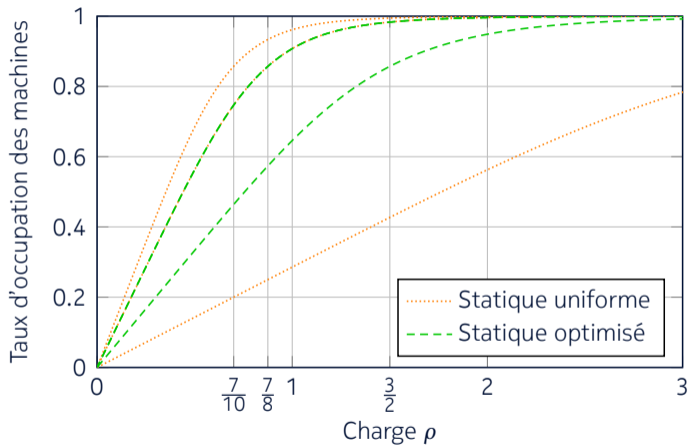
- Deux types de requêtes
- Arrivées selon des processus de Poisson de taux ν et $\frac{\nu}{4}$
- Toutes les machines ont la même capacité μ
- Chaque machine a 6 jetons
- Charge $\rho = \frac{\nu}{8\mu}$



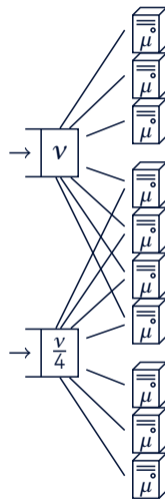
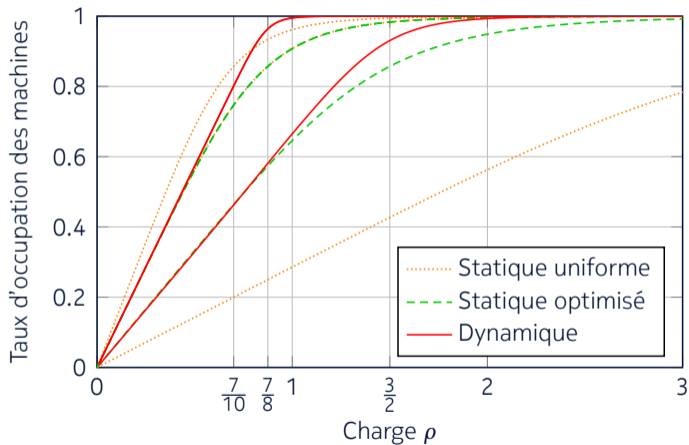
Contraintes d'affectations



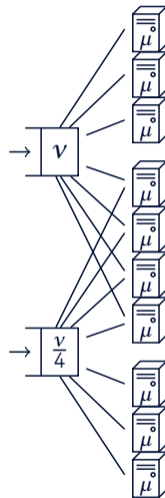
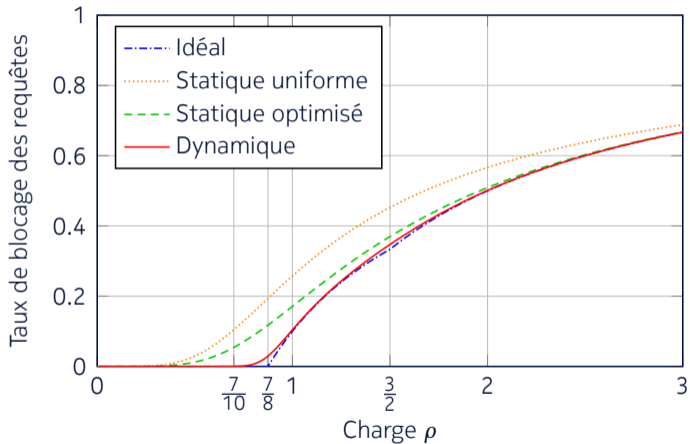
Contraintes d'affectations



Contraintes d'affectations



Contraintes d'affectations



Conclusion

Contributions :

- Algorithme de répartition de charge simple, insensible et dynamique
- Modèle de files d'attente qui atteste de l'insensibilité
- Résultats numériques
- (Des extensions possibles sans perdre l'insensibilité)

Conclusion

Contributions :

- Algorithme de répartition de charge simple, insensible et dynamique
- Modèle de files d'attente qui atteste de l'insensibilité
- Résultats numériques
- (Des extensions possibles sans perdre l'insensibilité)

Travaux futurs :

- Obtenir des formules plus simples prédire la performance
- Comparer la performance avec celle d'algorithmes non-insensibles
- Évaluer les versions non-bloquantes

Questions

