Simulation de serveurs en parallèle et en série

ISIMA |1 rue de la Chebarde, 63170 Aubière

TP1 - Simulation

Jael Vavasseur – ClÉment Mesnil

2019

Compte rendu

TP 1

Simulation à flux discrets

Jaël Vavasseur – Clément Mesnil Philippe Lacomme

Table des matières

[Introduction 3](#_Toc28642994)

[Partie 1 – Structures utilisées 5](#_Toc28642995)

[Partie 2 – Serveurs en série 6](#_Toc28642996)

[Partie 3 – Serveurs en parallèle avec probabilité 8](#_Toc28642997)

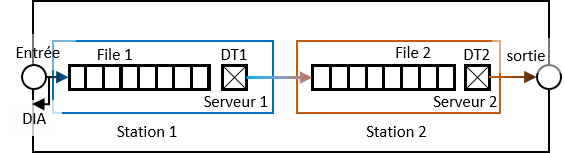
[Partie 4 - Serveurs en parallèle et réunion de pièces 9](#_Toc28642998)

# Introduction

Le but de ce TP est de modéliser et simuler en C++ des serveurs avec des files d’attentes en parallèle ou en série. Ce TP comporte plusieurs parties.

Tout d’abord, nous allons modéliser deux serveurs avec des files de capacités limitées, en série. Les durées de traitement des machines diffèrent et leurs capacités seront identiques. Évidemment, si la durée de traitement de la deuxième machine est trop importante par rapport à celle de la machine n°1, les pièces vont s’entasser dans la première file voire même bloquer l’arrivée des pièces.

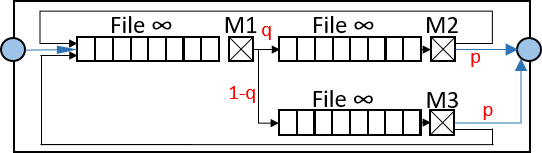
Nous allons donc étudier ce phénomène. De plus, nous modéliserons une péremption des pièces, c’est-à-dire une durée maximale d’attente dans la première file qui, si dépassée, rendra la pièce invalide.



*Figure 1. Schéma représentant la première modélisation :  
Serveurs en série*

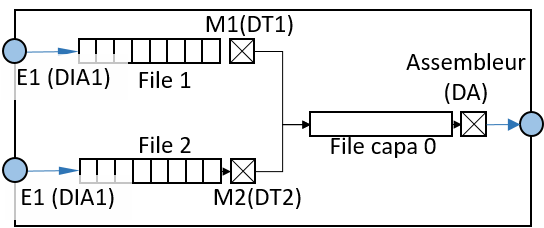
Dans un second temps, la modélisation effectuée sera composée de deux étapes : une première composée d’une seule machine sur laquelle toutes les pièces doivent passer avant d’aller vers, et une deuxième composée de deux machines en parallèle.

Après être passées sur la première machine, les pièces sont redirigées vers la machine M2 **ou bien** la machine M3, selon une probabilité **q**. Enfin, après être passées sur les machines 2 ou 3, les pièces retournent soit dans la 1ère file, soit vers la sortie et ce, selon une probabilité **p**. Les files considérées dans cette partie sont supposées de capacités infinies.



*Figure 2. Schéma représentant la deuxième modélisation :  
Serveurs en série probabilisés*

Pour finir, nous avons modélisé un assemblage de pièces, c’est-à-dire que la modélisation est composée de deux groupes entrée / file / machine en parallèle et qu’une fois que deux pièces sont prêtes (1 pour chaque machine), elles sont mises sur une autre machine, appelée assembleur. Dans le cas où une pièce est prête mais pas la deuxième, la pièce prête est mise en attente sur sa machine et bloque donc celles qui la suivent.

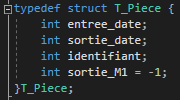


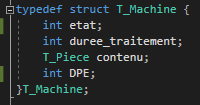
*Figure 3. Schéma représentant la troisième modélisation :  
Assembleur*

Pour réaliser ces différentes modélisations, nous avons dû mettre en place des structures en C++ nous permettant de gérer des pièces, des files, des machines et des entrées/sorties. C’est ce que nous allons détailler maintenant.

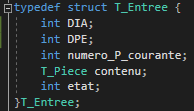
# Partie 1 – Structures utilisées

Nous avons d’abord mis en place la structure **T\_Piece** : les objets qui passeront plus tard dans les machines.

* entree\_date : date d’entrée dans le système global
* sortie\_date : date de sortie
* identifiant : entier permettant d’identifier la pièce
* sortie\_M1 : entier représentant la date à laquelle la pièce est sortie de la machine M1 (utilisé pour la deuxième partie du TP).

La structure **T\_Machine** peut contenir une pièce et est représentée par son état et la durée de son traitement.

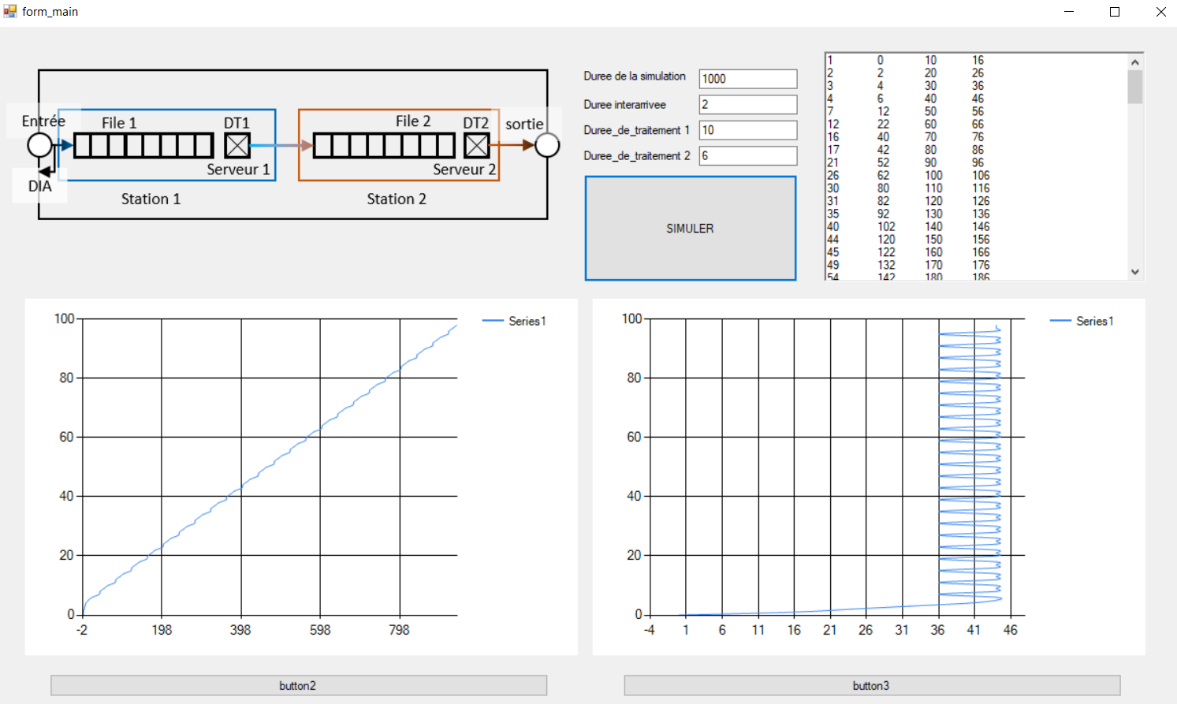
* etat : entier permettant de connaître l’état de la machine : 0 = vide, 1 = occupée et 2 = bloquée
* duree\_traitement : durée de traitement d’une pièce
* contenu : pièce sur laquelle la machine travaille
* DPE : date du prochain événement

La structure **T\_Entree** est celle qui dicte l’arrivée des pièces dans le système.

* DIA : durée inter-arrivée des pièces
* DPE : date du prochain événement
* numero\_P\_courante : indice permettant de générer les pièces
* contenu : pièce qui arrive dans l’entrée
* etat : état de l’entrée

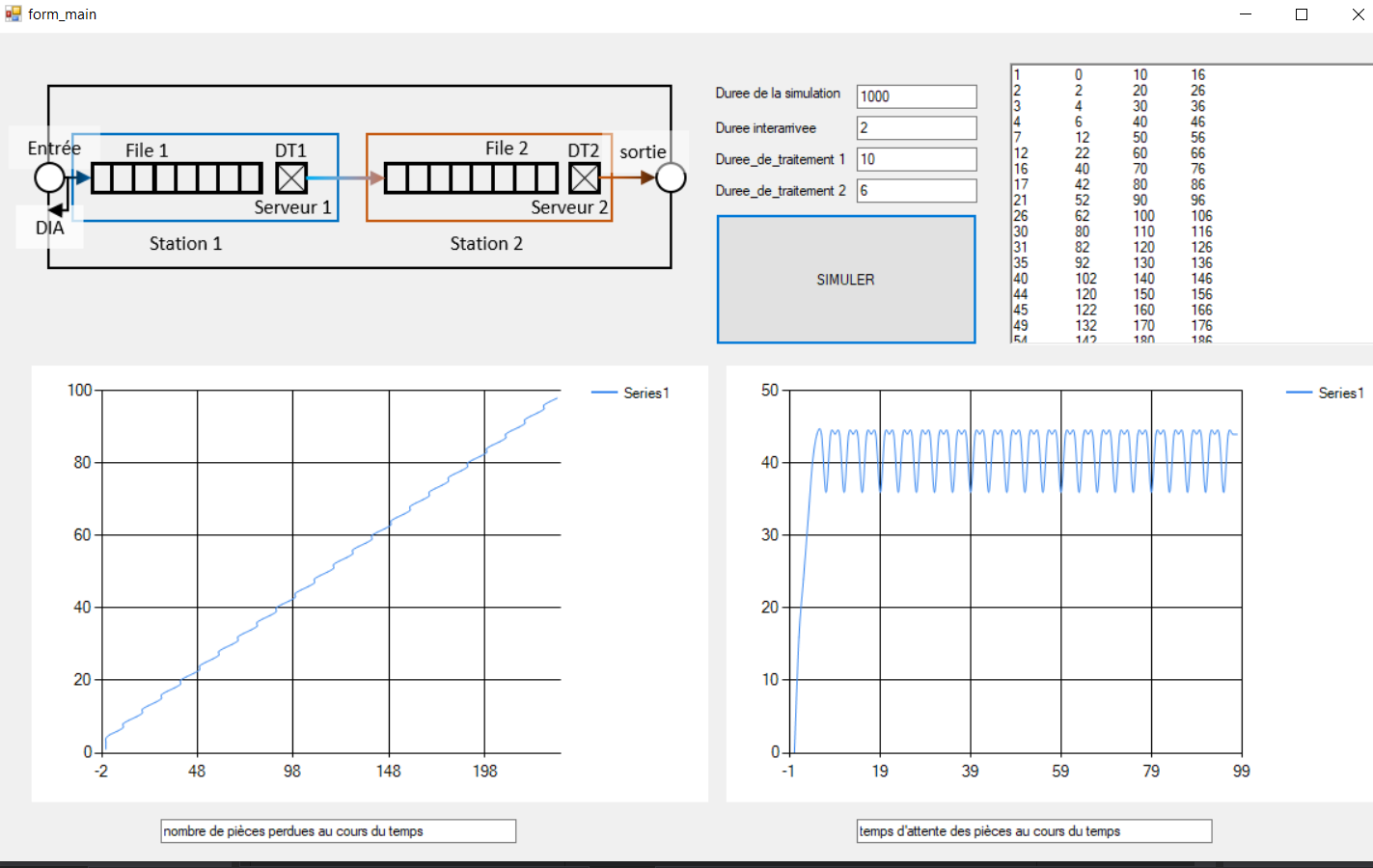
# Partie 2 – Serveurs en série

Nous avons utilisé les mêmes structures ainsi que les mêmes fonctions pour réaliser toutes les parties. Notamment, nous avons mis en place une un module de gestion de nos files, que l’on utilise pour stocker les clients qui attendent un serveur. Nous utilisons une structure T\_file pour gérer ces files.



Ci-dessus une capture d’écran du notre application. On a en haut à gauche le schéma du système modélisé, on peut régler les paramètres de la simulation au milieu et lorsqu’on lance la simulation (bouton SIMULER) les détails de la simulation apparaissent dans la fenêtre à droite. La première colonne correspond au numéro de la pièce, puis on a la date d’entrée de la pièce dans le système, la date de sortie de la machine 1, et enfin la date de sortie du système.

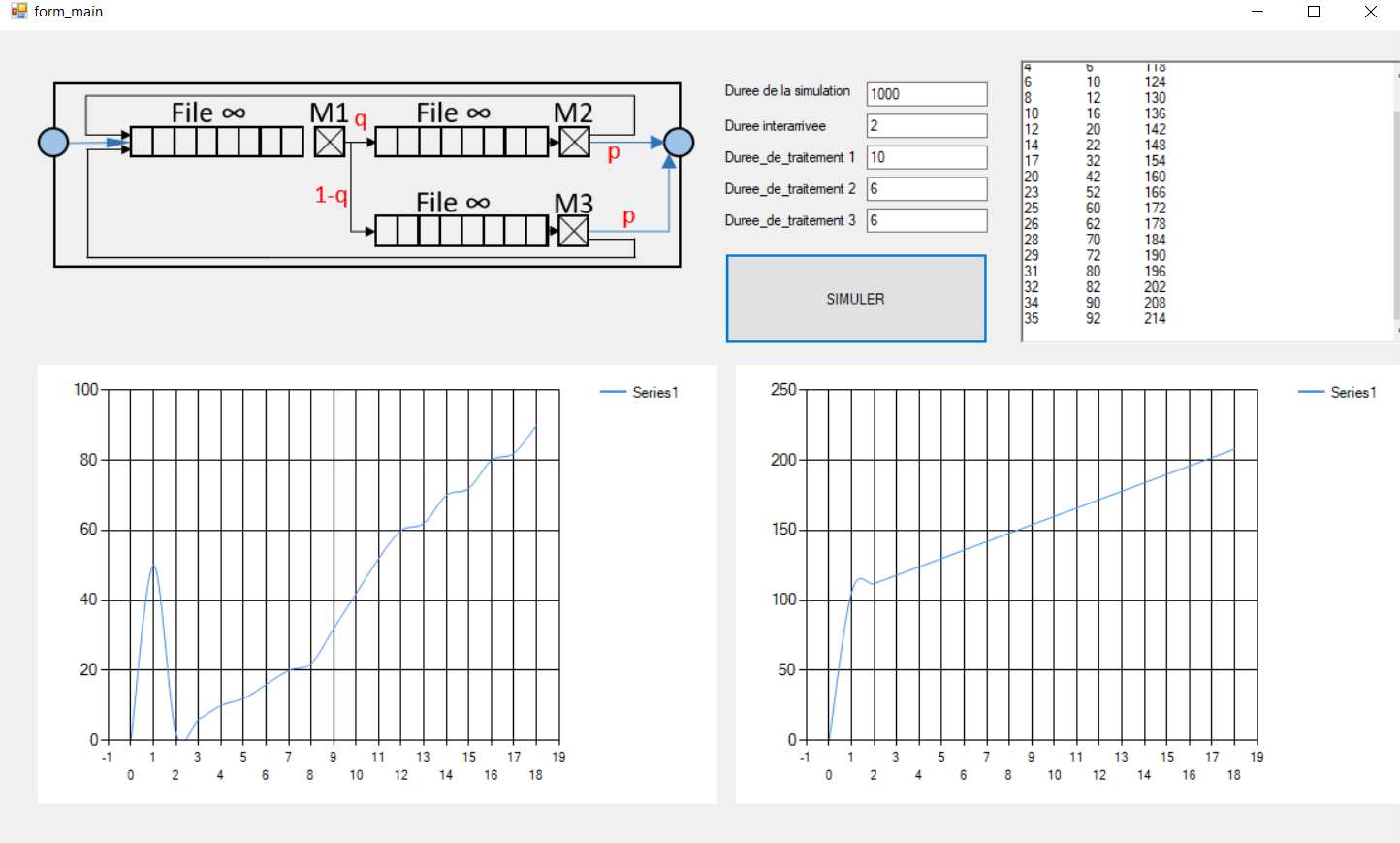
Ci-dessous une simulation du système avec départ du client s’il attend plus de 30 unités de temps dans la première file.

Le premier graphique nous indique le nombre de pièces perdues au cours du temps, on voit que l’on perd beaucoup de pièces, cependant, on remarque un schéma régulier de perte qui se reproduit au cours de la simulation. Le second graphique représente le temps total passé par les pièces dans le système. On remarque une augmentation exponentielle dès le début, mais rapidement, on atteint une limite vers 44 unités de temps et on oscille régulièrement entre 44 et 36. Cela vient corroborer notre premier graphique et son schéma régulier.

On remarque donc que quels que soient les paramètres utilisés pour la simulation, on atteint toujours une limite de temps d’attente, et contrairement à ce que l’on pourrait croire lorsque l’on fait la queue à La Poste, le temps d’attente ne tend pas vers l’infini !

# Partie 3 – Serveurs en parallèle avec probabilité

Ce système est constitué de 3 files mais à chaque passage dans le système, une pièce va choisir aléatoirement une file parmi les 2 proposées et passera seulement dans celle-là. Puis chaque pièce a une chance de repasser dans le système aléatoirement à l’envi (cf. figure 2).



On ne s’intéresse pas ici au blocage des files ou au nombre de pièces perdues puisqu’on suppose les files comme étant infinies. Le premier graphique n’est donc pas utilisé.

On retient toutefois la liste des machines par lesquelles les pièces sorties sont passées.

# Partie 4 - Serveurs en parallèle et réunion de pièces

Ce système (cf. figure 3) consiste en la réunion de 2 éléments pour former la pièce lors d’une troisième manipulation. La fabrication des 2 éléments doit être terminée avant défaire sortir n’importe quelle pièce des machines précédents. C’est-à-dire que si la première machine a terminé avant la deuxième, la première doit attendre la seconde et passe donc dans un état d’attente jusqu’à ce que la deuxième machine ait fini. Quand les 2 machines ont fini, on peut vider les 2 machines et on lance l’assemblage. La machine d’assemblage ne possède pas de file.

J’ai pas mis de capture parce que je sais pas quoi mettre comme graphe …