République Algérienne Démocratique et populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE AMAR TELIDJI
LAGHOUAT

FACULTE DES SCIENCES ET D'INGENIERIE
DEPARTEMENT DE GENIE INFORMATIQUE

PROJET DE FIN D'ETUDES
Pour l'Obtention Du Diplôme

D'INGENIEUR D'ETAT EN INFORMATIQUE

Option : intelligence artificielle

Thème

Simulation du fonctionnement d'un serveur web par l'approche des Trois Phases

Présenté par :

- Zitouni Amina
- Mahtal Amina

Encadré par :

Dr : Ouinten Youcef

N° d'ordre : 06 …….. **/2005-PFE/DGI**

Dédicaces

Je dédie ce modeste travail :

A mes chers parents, en qui j'ai puisé tout le courage, la volonté et la confiance

A mes frères, Mohamed, Abdelkhader, Abderrezak,

A ma grand-mère B.Fatma,

A mes chères tantes, et plus particulièrement à Meriem pour son soutien et ses encouragements qu'elle m'a apportée,

A mes oncles, A mes chères cousines et cousins,

A toute ma famille,

A tous mes Amis plus particulièrement A Amina et A la promotion du 2005.

Zitouni Amina

Je dédie ce modeste travail

À mes très chers parents, à qui je dois ma réussite

Grâce à leur soutien morale et matériel,

je leurs serai éternellement reconnaissante

À mes très chères frères, Abd Eldjallil et Ismail

À mes adorables soeures : Wafa, Fella et Yasmine, Qui m'ont encouragés

A mes oncles, A mes tantes, A mes cousins et cousines

Et A toute ma famille

A ma très cher Amina (Binôme) pour son amitié sacré et sa confiance

A tous mes amis, et A la promotion du 2005 et plus particulièrement A :

Toufik, Lakhdar, Kamal, Cherifa, Nassima, Zoulikha,…

Mahtal Amina

Remerciements

Nous tenons tout d'abord à remercier Monsieur le Docteur Y.OUINTEN, d'avoir accepté de nous encadrer et consacré énormément de temps à la correction de ce mémoire. Ses précieux conseils nous ont permis une bonne orientation.

Nous adressons nos remerciements à tous nos enseignants pour leurs aides et soutiens durant les quatre années d'études. Nous remercions surtout :
Monsieur le chef du département d'informatique D.Derradji Ben Hssen, Monsieur N.Lagraa, Monsieur Y.Belabbassi, Monsieur M.Djoudi, Monsieur K.Boukhalfa, Monsieur Yagoubi, Monsieur A.Tahari, Monsieur R.Ben Zide, Madame B.Kerrouche, Madame H.Bouzouad, Madame A.Chetih et Madame F.Gibadge.

Nos remerciements les plus vifs vont tous particulièrement A Monsieur A.Mokhtari et Monsieur A.Badache pour leurs aides accordés à notre travail.

Nous remercions les membres de jury, pour l'honneur qu'il nous fait en acceptant de présider et d'examiner notre mémoire.

Enfin, merci à tout ceux qui ont contribué de près ou de loin à ce mémoire, du point de vue scientifique ou administratif.

Table des matières

Résumé

La simulation est un outil très puissant dans l'étude et la résolution d'un problème réel complexe dont la solution théorique est souvent trop compliquée ou impossible. Dans ce travail, nous présentons une conception et une réalisation d'un simulateur d'un serveur web en utilisant la méthode des trois phases pour sa simplicité et son efficacité d'exécution par rapport aux autres approches.

Vu l'importance des files d'attentes dans les réseaux informatiques, nous avons commencé par l'étude des réseaux de files d'attentes et de leurs performances. Le serveur web a été représenté par un réseau de quatre files d'attentes interconnectées entre elles. Le simulateur réalisé a été programmé à l'aide du langage de programmation Pascal sous l'environnement Delphi. Il permet d'étudier les performances de notre système telles que le temps de séjour des requêtes dans le système, la longueur des files, le nombre de requêtes servies, les taux d'occupation des serveurs etc. Nous avons étudié et comparé, à l'aide de ce simulateur, cinq configurations différentes du serveur web et nous avons montré que le cache contribue beaucoup à l'amélioration de la performance du serveur web en terme de temps de séjour de la requête dans le système.

Introduction

Le thème réseaux et performances s'intéressent depuis plusieurs années à la conception et l'analyse des réseaux, ainsi qu'aux techniques de modélisation, des systèmes informatiques et des systèmes de productions.

Les réseaux comme beaucoup d'objets du monde réel, peuvent être étudiés au moyen d'un simulateur à évènements discrets. Il s'agit d'un programme dans lequel le temps est discrétisé, et une horloge qui gère l'ensemble des évènements pouvant se produire dans le système réel qui est modélisé dans le simulateur. A chaque pas (cycle), c'est-à-dire l'avancement d'une unité de temps virtuel l'ensemble des effets nécessaires à la simulation des composants du système est appliqué.

La simulation est un outil très puissant dans la résolution d'un problème réel. Son importance réside surtout le fait qu'elle permet d'étudier des problèmes assez complexes, dont la solution théorique est souvent trop compliquée ou impossible.

Concevoir un modèle de simulation d'un système c'est en faire produire une représentation qui permet de saisir le fonctionnement de ce système, et d'anticiper le comportement sous divers modes d'utilisation. Il existe plusieurs approches de simulation et on a choisi parmi ces approches, l'approche des trois phases.

Vu l'importance et la complexité des réseaux de files d'attentes, nous avons choisi pour notre projet de fin d'études, un thème qui a pour but l'étude des performances des réseaux de files d'attentes. On a modélisé un serveur web par un réseaux de files d'attente qui a quatre files interconnectées entre elles ; une représente le cache, les deux autres représentent chacune un disque et la dernière représente la file des requêtes servies. Les arrivées sont supposées suivre une loi de Poisson et les sorties une loi exponentielle.

L'objectif de ce travail est avant tous de nous familiariser avec la modélisation et notamment avec la simulation.

Structure du mémoire

Ce mémoire comporte une introduction, trois chapitres, une annexe et une liste de références bibliographiques.

Le premier chapitre, débute par une présentation détaillée des composantes d'une file d'attente, suivie d'une étude de quelques modèles markovien avec les calculs des performances d'un système en régime stationnaire. Nous terminons ce chapitre par une présentation d'un réseau de file d'attente.

Le deuxième chapitre est consacré à la simulation. On introduit les principales notions, ainsi que les différentes approches, les avantages et les inconvénients de la modélisation et de la simulation.

Le dernier chapitre sera consacré à la conception et à l'implémentation d'un simulateur qui permet d'étudier le comportement d'un serveur web en utilisant l'approche des trois phases. La présentation du simulateur qu'on a réalisé est suivie d'une interprétation des résultats obtenus après l'étude d'un cas.

En Annexe, nous présentons tous les résultats bruts obtenus par le simulateur et qui nous ont servi pour la synthèse des résultats présentés pour l'interprétation.

Chapitre I

LES FILES D'ATTENTE

Nous présentons dans ce chapitre, les notions de bases de la théorie des files d'attente et des réseaux de files d'attente. Nous supposons que les clients arrivent selon un processus de Poisson et que le temps de service suit une loi exponentielle. Nous calculons explicitement les caractéristiques de l'état stationnaire, ensuite nous présentons une étude approfondie de quelques modèles Markovien.

---

1. Généralités sur les phénomènes d'attente

Les phénomènes d'attentes sont d'observation courante dans la vie quotidienne. Quand nous nous rendons à la poste, à la gare, à la banque, bien souvent nous devons 'faire la queue' pour obtenir des timbres, un billet, de l'argent. On a l'habitude d'appeler **clients** les individus qui constituent la file d'attente et **station** le guichet où un **serveur** leur procure un service déterminé.

La file d'attente ne se manifeste pas toujours d'une manière physique : par exemple, si, dans un atelier, des machines tombées en panne 'attendent' la visite et les soins du mécanicien réparateur, elles ne se constituent pas en file.

Considérons le cas très simple (figure I.1) où il n'existe qu'une station, et qu'une file d'attente. Chaque client arrive pour recevoir un service. Lorsque le serveur est occupé, le client attend dans le système pour être servi. Dès la fin du traitement du client en service, le client suivant est choisi de la file d'attente selon la discipline utilisée **[S2][5]**.

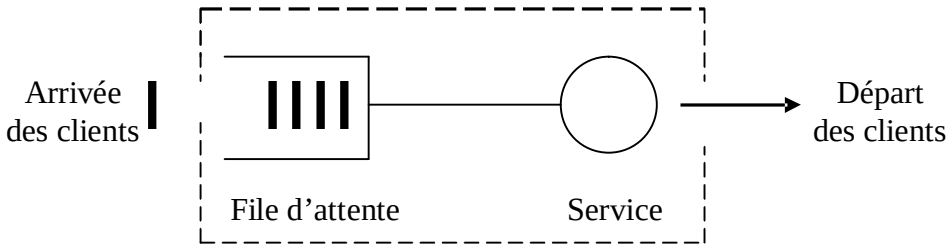


Figure I.1 : Système de file d'attente à canal unique.

Est-ce qu'un système d'attente est la même chose qu'un phénomène d'attente ?
On a en effet coutume d'appeler **file d'attente** l'ensemble des clients qui attendent d'être servis, à l'exclusion de celui qui est entrain de se faire servir.
On nomme **système d'attente** l'ensemble des clients qui sont dans le système, c'est-à-dire ceux qui font la queue, et celui qui se fait servir. Le **phénomène d'attente** s'étend à tous les clients possibles (dans le cas de systèmes bouclés, où les mêmes clients reviennent plus tard à l'entrée, le nombre de client est, en général, fini).
Ces appellations se généralisent et prennent surtout leur intérêt dans les situations où coexistent plusieurs stations et plusieurs files d'attente.

Nous distinguerons donc deux types principaux de systèmes d'attente : le système **ouvert**, dans lequel le nombre de clients potentiel est très élevé (cas des guichets publiques, des grands magasins, etc.), au point qu'on peut considérer qu'il est illimité ; le système d'attente **fermé** dans lequel le nombre de clients est limité (cas d'un atelier dans lequel existe un service de réparation de machines) ; dans les deux cas il peut y avoir une ou plusieurs files d'attentes.

La théorie des files d'attentes remonte aux premiers travaux de K.Erlang, en 1917, sur le calcul du nombre d'organes de chaque type à installer dans un centre de téléphone automatique. Développée aussi par Engset (1918), cette théorie s'est amplifiée sous l'impulsion de nombreux chercheurs, parmi lesquels il faut citer E.Borel, A.Khintchine, F.Pollaczek,W.Feller, etc. Après l'avoir longtemps boudée, les informaticiens y viennent par nécessité, notamment pour établir des systèmes en temps réel ou des réseaux d'ordinateurs **[5]**.

2. Structure d'un phénomène d'attente

La figure suivante (figureI.2) schématise un phénomène d'attente. Les clients entrent dans le système par des portes tambours. Ils prennent place dans les files d'attente. Dès qu'un client a obtenu le service désiré auprès d'une station, il sort du système et il est remplacé par l'un des premiers clients attendant dans les files.

m est le nombre d'unités pouvant se trouver dans l'ensemble du phénomène,

n le nombre d'unités dans le système proprement dit, limité sur la figure à l'intérieur du rectangle tracé en traits tirés,

v le nombre d'unités dans les files,

j le nombre d'unités en cours de service.

Il peut y avoir pour les clients une possibilité de choix entre les stations ou encore une discipline à respecter, par exemple des règles de priorité.

Les phénomènes d'attente sont régis par le hasard, la durée d'occupation d'une station par un client n'étant pas constante et les arrivées de la clientèle aléatoires **[2]**.

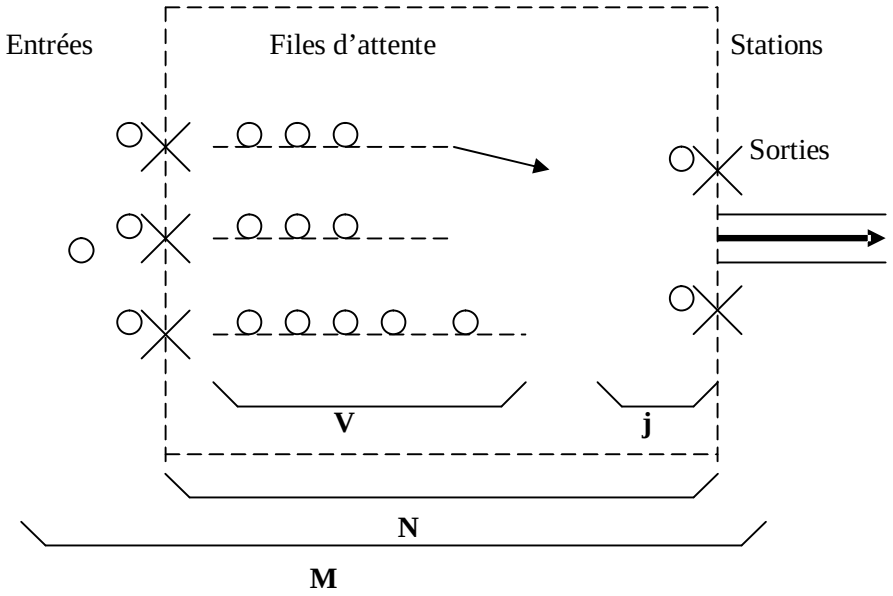


Figure I.2 : Structure d'un phénomène d'attente.

3. Définition d'un système de file d'attente

On considère ici un système destiné à offrir un certain service- celui (personne ou objet) qui vient pour bénéficier du service est appelé client- les postes de service (lieu où s'effectue le service) sont appelés guichets (serveurs), où officient des serveurs.

Un client qui rejoint le système ne sera pas toujours immédiatement servi : si tous les guichets sont occupés lors de son arrivée, il rejoint la file d'attente, ou file des clients en instance de service **[4]**.

Un système de file est file d'attente est donc constitue de :

- Service.
- Client.
- Discipline de service.

3.1. Le service

La prestation que vient chercher un client auprès d'un système est appelée service. Le service dure généralement un certain temps qui correspond au temps que le client passe au guichet par exemple. Le temps de séjour total dans le système est la somme du temps d'attente et de la durée du service effectif.

3.2. Clients

On appelle ainsi les objets ou les personnes qui arrivent dans le système, attendent s'il y a lieu, sont traitées (service) et repartent.
On suppose que les clients arrivent un par un, et sont servis selon la discipline de service choisie.

3.3. La discipline de service

Détermine l'ordre dans lequel les clients sont rangés dans la file et y sont retirés et par conséquent, servis. Les disciplines les plus connues sont :

- FIFO (First in, First out) ou PAPS (premier arrivé, premier servi).
- LIFO (Last in, First out) ou DAPS (dernier arrivé, premier servi).
- PS Processor Sharing. (processeur partagé).
- RANDOM.

4. Processus d'arrivée

Les clients arrivent au sein du système en décrivant un processus déterminé. Ils peuvent par exemple être réguliers et leurs arrivées sont espacées par un temps égal à τ (c'est-à-dire chaque τ unités de temps, on a une arrivée). Le modèle le plus simple et le plus courant est celui des arrivées complètement aléatoires, ce qui est caractérisé par le processus de Poisson **[3]**.

$$P_n(t) = \frac{(\alpha t)^n}{n!} e^{-\alpha t} \qquad \alpha > 0$$

Où α représente le taux des arrivés ou encore $1/\alpha$ représente le temps qui sépare deux arrivées successives.

5. Processus de service

La deuxième composante d'un système de files d'attente est la quantité de service demandée par un client. Dans la majorité des cas, on suppose que la population des clients est homogène, ce qui entraîne que les services demandés sont identiquement distribués, ou ont une distribution commune dite distribution de service. Dans des cas plus compliqués, les clients sont classés dans de différents types suivant la nature de distribution de leurs services.

En pratique, on rencontre la distribution exponentielle qui est la plus simple à manipuler mathématiquement **[3]**.

Dans notre étude le temps de service a une distribution exponentielle. Si on appelle *T* la variable aléatoire qui représente ce temps, *T* aura pour densité de probabilité :

$$f(t) = \alpha e^{-\alpha t} \qquad \alpha > 0$$

6. Système de notation

Le système de notation le plus répondu actuellement est introduit par Kendall en 1953 et complété par Lee en 1966 **[S5]**.

Kendall a introduit La notation suivante :

Forme abrégée : A / B / S.

Forme complète : A / B / S / N / P / D.

où chaque symbole correspond à une caractéristique de la file d'attente **[S3] [3]:**

- ***A*** désigne **la loi des inter-arrivées**. Ce paramètre peut prendre la valeur
  - *M* pour la distribution exponentielle
  - E_n pour la distribution d'Erlang avec *n* phases.
- ***B*** désigne **la loi de service**. Ce paramètre peut prendre la valeur
  - *M* pour la distribution exponentielle (M comme Markov).
  - E_n pour la distribution d'Erlang avec *n* phases.
- ***S*** désigne le **nombre de serveurs (guichets)**. Par défaut, c'est 1.
- ***N*** désigne la **capacité du système**, c'est-à-dire le nombre maximum de clients qui peuvent être présents dans le système simultanément.
- ***P*** désigne la taille de la population des clients susceptible d'utiliser le système.
- ***D*** désigne la **discipline de service**, qui détermine comment sont servis les clients. Ce paramètre peut prendre les valeurs suivantes.
  - FIFO (First-In-First-Out)
  - LIFO (Last-In-First-Out)
  - PS (Processor Sharing): le serveur est partagé entre tous les clients
  - Priorités spécifiques : chaque client a une priorité et celui qui a la plus grande priorité sera servi en premier.

7. Les mesures de performances

L'étude d'une file d'attente ou d'un réseau de files d'attente a pour but de calculer ou d'estimer les performances d'un système dans des conditions de fonctionnement données. Ce calcul se fait le plus souvent pour le régime stationnaire uniquement. Les mesures les plus fréquemment utilisées sont **[S3] [S5]** :

- Le nombre moyen de clients dans le système.
- Le temps moyen de séjour dans le système ou temps de réponse (attente et service).
- Le temps de service.
- Le taux d'utilisation des serveurs.
- La longueur de la file.

Nombre moyen des clients dans le système

Le nombre moyen des clients dans le système représente le nombre moyen de clients en attente et en service.

Temps de réponse du système

Le temps de réponse du système est le temps moyen de séjour d'un client dans le système. Il est égal à la différence entre sa date de départ du système et sa date d'arrivée dans le système.

Temps de service

C'est la durée du service délivré par le serveur dans la plupart des cas.

Taux d'utilisation des serveurs

Le taux d'utilisation d'un serveur représente la proportion du temps pendant laquelle le serveur est effectivement occupé à délivrer un service pendant toute la durée de fonctionnement du système.

La Longueur de la file

C'est la taille de la file c'est-à-dire le nombre de clients qui attendent dans la file d'attente pour être servis.

7.1. Calcul des caractéristiques d'un système d'attente

- $\overline{N}$: le nombre moyen de clients présents dans le système (en attente et en service);
- $\overline{Q}$: le nombre moyen de clients en attente;
- $\overline{T}$: le temps moyen de séjour dans le système ou de réponse (attente et service);

- $\overline{W}$: le temps moyen d'attente;
- ρ : le coefficient d'utilisation du système ;
- U_j : le taux d'utilisation du serveur j;
- U : le taux d'utilisation du système;

Ces valeurs permettent de juger le comportement d'un système d'attente. Elles sont liées entre elles par les relations suivantes.

$$\overline{N} = \lambda \overline{T} \quad (7.1)$$

$$\overline{Q} = \lambda \overline{W} \quad (7.2)$$

$$\overline{T} = \overline{W} + 1/\mu \quad (7.3)$$

$$\overline{N} = \overline{Q} + \lambda/\mu \quad (7.4)$$

Avec :

λ : Le taux d'entrées des clients dans le système,

$1/\lambda$: L'intervalle de temps moyen séparant deux arrivées consécutives,

μ : Le taux de service,

$1/\mu$: La durée moyenne de service

Les formules (7.1) et (7.2) sont appelées formules de Little **[S7]**.

L'un des aspects importants auquel on s'intéresse souvent lorsqu'on analyse un système est la stabilité du système. Lorsque le système est stable, on peut alors utiliser les relations suivantes:

$\overline{T} = \overline{W} + \overline{S}$ où $\overline{S}$ est le temps moyen de service

et

$$\overline{N} = \overline{Q} + mU_j$$

Où *m* est le nombre de serveurs.

8. Système de files d'attentes élémentaires

8.1. Système M / M / 1

On considère un système de files d'attentes M/M/1. Ce système est formé d'une file d'attente et d'un serveur. Le temps qui sépare deux arrivées successives est une variable aléatoire qui suit une loi exponentielle de paramètre λ et la durée de service est une variable aléatoire exponentielle de paramètre μ . La capacité d'attente est illimitée.

On notera $\rho = \left(\frac{\lambda}{\mu}\right)$ **[S1][1]**.

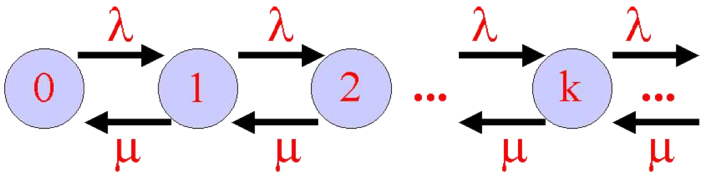


Figure I.3 : Evolution de l'état de la file M/M/1.

La probabilité d'avoir n clients dans le système à l'état d'équilibre est :

$$P_n = (1-\rho)\rho^n \quad \text{et} \quad P_0 = 1-\rho \quad .$$

Ø **Les mesures de performances : Nombres moyens**

Le nombre moyen des clients présents dans le système est :

$$\overline{N} = \mathrm{E}[X] = \sum n(1-\rho)\rho^n = \rho(1-\rho)\sum_{n=1}^{\infty} n\rho^{n-1}$$

$$= \rho(1-\rho)\frac{d}{d_\rho}\left(\sum_{n=1}^{\infty}\rho^n\right) = \rho(1-\rho)\frac{d}{d_\rho}\left(\frac{\rho}{1-\rho}\right) = \frac{\rho}{1-\rho}$$

Et le nombre moyen de clients en attente dans la file est :

$$\overline{Q} = \overline{N} - U_J = \frac{\rho}{1-\rho} - \rho = \frac{\rho^2}{1-\rho}$$

Ø **Les mesures de performances : Temps moyen**

En utilisant la formule de Little, on obtient le temps moyen de réponse,

$$\overline{T} = \frac{\overline{N}}{\lambda} = \frac{\rho}{\lambda(1-\rho)} = \frac{1}{\mu(1-\rho)} = \frac{\overline{S}}{1-\rho}$$

Et pour le temps moyen d'attente, on a :

$$\overline{W} = \frac{\overline{Q}}{\lambda} = \frac{\rho^2}{\lambda(1-\rho)} = \frac{\rho}{\mu(1-\rho)} = \frac{\rho\overline{S}}{(1-\rho)}$$

Pour que ce système soit en équilibre, ou stable, il faut que la condition $\rho < 1$ soit vérifiée. Cette condition est appelée condition d'ergodicité.

Remarque : On a bien $\overline{T} = \overline{W} + \overline{S}$ où $\overline{S} = 1/\mu$.

Pour une file stable le taux d'utilisation du serveur j est : $U_j = \rho$

8.2. Système M / M / S

On considère un système de files d'attente M/M/S, les clients arrivent suivant un processus de Poisson de paramètre λ et vont se faire servir dans un des *S* guichets. Chaque serveur ne sert qu'un client à la fois. Les temps de services sont indépendants entre eux et suivent la loi exponentielle de paramètre μ. Dés qu'un guichet se libère, le premier client de la file va immédiatement s'y faire servir **[1]**.

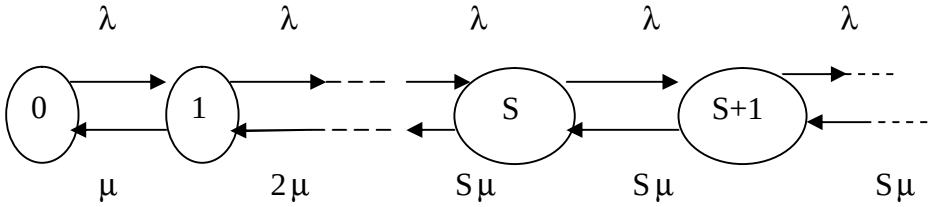


Figure I.4 : Evolution de l'état de la file M / M / S

La condition d'ergodicité du système est : $\rho = (\lambda/(m\mu)) < 1$

D'où on déduit :

$$P_0 = \left[\sum_{n=0}^{S-1} \frac{(\lambda/\mu)^n}{n!} + \frac{(\lambda/\mu)^S}{S!\left(1 - \frac{\lambda}{\mu S}\right)} \right]^{-1}$$

Ø **Les mesures de performances : Nombres moyens**

Le nombre moyen de clients en attente est :

$$\overline{Q} = \left(\frac{\lambda}{\mu}\right)^s \cdot \frac{1}{S!} \cdot \frac{\rho}{(1-\rho)^2} P_0$$

Et le nombre moyen des clients présents dans le système est : $\overline{N} = \overline{Q} + \rho$

Les expressions $\overline{T}$ et $\overline{W}$ peuvent être calculées par les formules de Little.

9. Les réseaux de files d'attente

9.1. Définition

Un réseau de files d'attente est composé d'un ensemble de stations de service et d'un ensemble de clients. Le réseau de file d'attente est caractérisé par les processus représentant l'arrivée des clients au réseau, les temps de service des clients aux stations, les disciplines de service des clients aux stations et le routage des clients d'une station à une autre suivant les probabilités de routage P_{ij} **[S5]**.

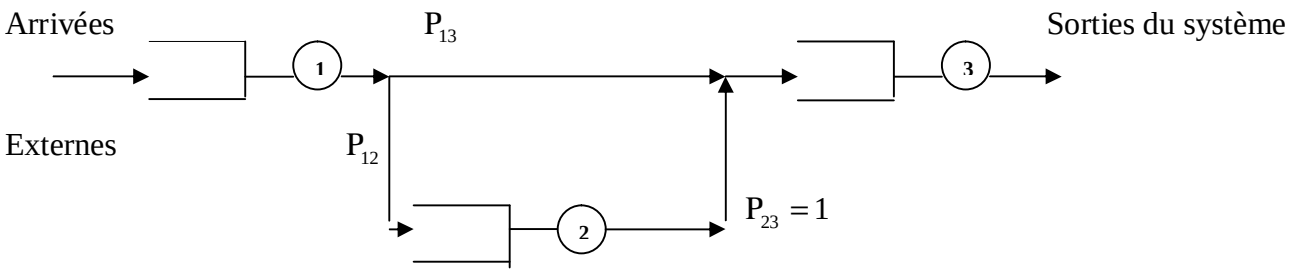


Figure I.5 : Réseaux de files d'attente.

9.2. Réseaux ouverts, fermés et mixtes

Un réseau est ouvert si tout client présent ou entrant dans le système peut le quitter. Un réseau est fermé si les clients ne peuvent le quitter. Dans un réseau fermé, le nombre de clients est généralement fixe et ces derniers sont présents dans le système dès le début de son évolution. Finalement, un réseau est mixte s'il est ouvert pour certains clients et fermé pour d'autres **[S5]**.

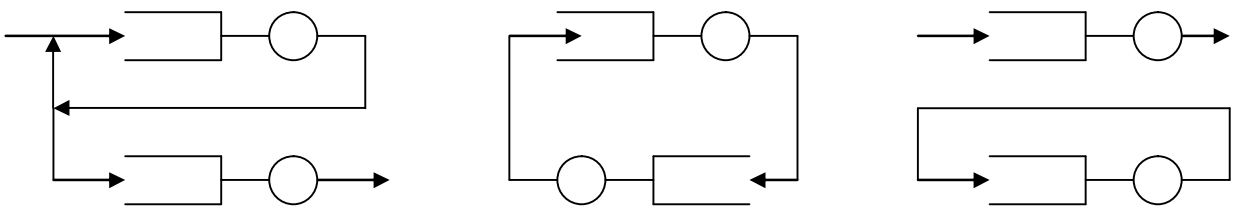


Figure I.6 : Les différents types de réseaux

9.3. Caractéristiques d'un réseau de files d'attente

Les caractéristiques d'un réseau de file d'attente sont :

- $\overline{T}$: temps moyen de réponse dans le système
- $\overline{T_i}$: temps moyen de réponse dans la station *i*
- $\overline{N}$: le nombre moyen de clients dans le système
- $\overline{N_i}$: le nombre moyen de clients dans la station *i*.

9.4. Méthodes approximatives

L'obtention de la solution exacte en étudiant un réseau complexe s'avère très difficile. Certaines méthodes approximatives permettent d'approcher ces solutions. Parmi celles-ci, la méthode de décomposition, la méthode d'isolation, etc.
Le principe de la méthode d'isolation consiste à subdiviser le système global en *L* sous systèmes et à les étudier séparément. La modélisation permettra de connaître pour chaque système :

- les caractéristiques des flots d'entrée ;
- les caractéristiques des temps de services.

Il faut choisir le découpage d'une manière telle que chaque sous-système possède une solution connue. Pour obtenir les équations des sous systèmes, on pourra faire appel à une méthode exacte ou à une méthode approximative. La résolution de l'ensemble de ces équations conduit à la solution numérique du système global. La validation des résultats se fait par comparaison avec ceux obtenus par simulation, ou à partir de mesures effectuées sur un système réel.

Chapitre II

LA SIMULATION

Nous présentons dans ce chapitre, les notions de bases de la simulation. Nous introduirons les concepts fondamentaux de la simulation et les différentes approches de modélisation, puis nous allons citer quelques avantages et inconvénients.

---

1. Introduction

D'après la définition du verbe simuler qui veut dire "faire semblant de", on peut dire que la technique de simulation est l'un des meilleurs moyens qui nous permet de simuler le fonctionnement des systèmes dont l'étude analytique et directe sont assez difficiles, ou parfois impossibles, tels que certains systèmes de files d'attente.

En général, un modèle de simulation regroupe deux grandes phases :

- La phase de génération des nombres aléatoires suivant la ou les lois de fonctionnement du système et qui ont été déduites de quelques observations réelles.
- La phase d'élaboration du modèle de simulation lui-même et qui consiste à mettre à jour tout évènement ayant de l'importance dans le système, ainsi que l'accumulation des valeurs des paramètres sur lesquels porte l'étude du système.

Puisque la simulation est une étude expérimentale, son utilisation est associée aux problèmes qui surgissent au cours d'une expérience. On note en particulier, la validation des données d'entrée, la longueur de la période de l'expérience, le nombre nécessaire et suffisant de répétitions de l'expérience afin d'aboutir à la convergence des résultats, et l'analyse de ces derniers **[3]**.

2. Généralités et définitions

La simulation peut être considérée comme une méthode de modélisation ou comme une *expérimentation sur un modèle* ou même un système réel. Elle étudie le comportement d'un système à travers quelques périodes, en construisant un deuxième système ayant la même structure que l'original, mais qui est plus simple à manipuler. Ce deuxième système est ce qu'on appelle un modèle **[3]**.

2.1. Système

Un système est un regroupement de plusieurs éléments ou composantes. A n'importe quel instant, il est dans un état particulier défini par l'état de ses composantes et des relations qui les relient. L'état du système change quand l'état de ses composantes change **[3]**.

2.2. Modèle

Parfois, il est impossible d'étudier le système directement du fait qu'il soit inaccessible, ou trop coûteux pour qu'on puisse l'utiliser pour faire des expériences. D'autres fois, du fait qu'il change trop rapidement ou trop lentement. Dans ces cas l'étude est faite sur un deuxième système construit pour l'occasion et dont la similitude avec le système original est aussi parfaite que l'étude l'exige. Ce deuxième système est appelé modèle **[3]**.

2.3. Développement d'un modèle

La séquence des étapes prises afin de construire un modèle dépend du problème posé. Mais la majorité des modèles se partagent les étapes décrites par le schéma suivant **[3]**:

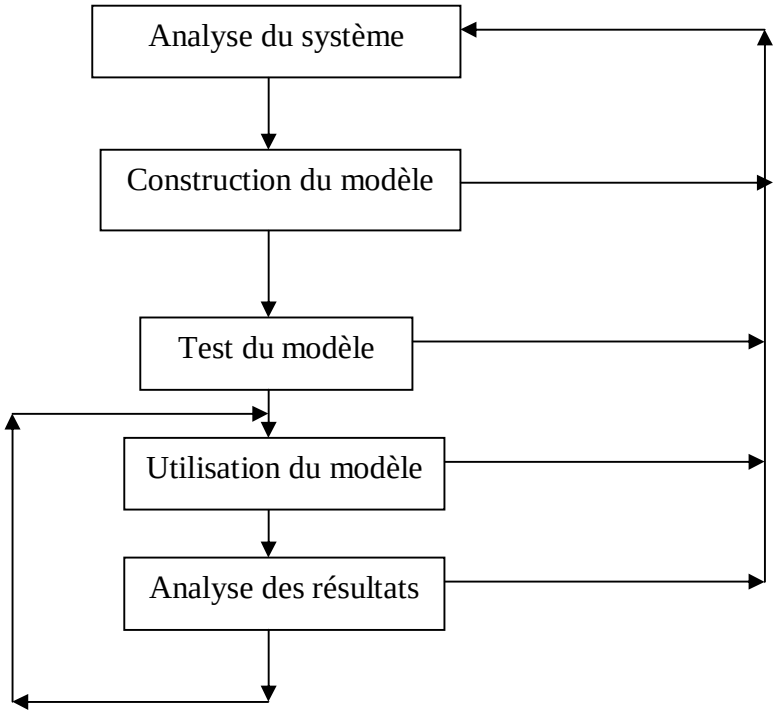


Figure II .1 : Le schéma des étapes des modèles.

a) Analyse du système

Dans cette phase on identifie les composantes du système et les relations existantes entre les différents paramètres ; puis on attribue à ces derniers leurs valeurs actuelles. Dans le cas où ces valeurs sont aléatoires, il faudrait faire des expériences sur le système.

b) Construction du modèle

En construisant un modèle, on devrait faire de notre mieux afin qu'il soit le mieux représentatif du système. On n'entend pas par là qu'il reflète exactement le système, car ce n'est ni nécessaire ni possible dans certains cas ; mais qu'il nous permette de représenter mieux les paramètres sur lesquels porte notre étude.

c) Test du modèle

Après avoir construit le modèle, on devrait tester à quel point il reflète le système. La dispersion entre modèle et système peut-être due à la programmation, à l'analyse du système, ou au mauvais choix des paramètres. La robustesse du modèle est toujours testée en comparant les résultats de la simulation avec des mesures réelles prises sur le système, ce qui laisse supposer que le système existe déjà et que toute mesure est y possible. Si le modèle reflète exactement le système, pour les paramètres étudies, on pourrait affirmer sa validité.

d) Utilisation du modèle

Pour utiliser un modèle, on doit supposer que ses résultats sont stables. Contrairement aux modèles analytiques qui donnent des résultats sûrs, les modèles stochastiques ne donnent de tels résultats qu'après de nombreuses exécutions du programme de simulation. La loi des grands nombres nous assure dans ce cas que les valeurs des paramètres du modèle seront proches de celles du système.

e) Analyse des résultats

La théorie des probabilités met à notre disposition un ensemble considérable de méthodes qui nous permettent d'analyser les résultats d'un modèle de simulation. On remarque que si l'analyse est mauvaise, on doit procéder à des modifications dans la phase d'analyse.

3. Génération des nombres aléatoires

La génération des nombres aléatoires a connu plusieurs étapes :

- Génération suivant un processus physique : la séquence est non reproductible donc impossible à tester.
- Etablissement de tables de nombres aléatoires : ces tables sont difficiles à manipuler.
- Mise au point de méthodes algorithmiques : elles sont aisément programmables et elles conduisent à des séquences qu'on pourrait répéter, ce qui signifie que si on relance le générateur avec les mêmes conditions initiales, on aura la même séquence de nombres **[S4]**.

Un générateur est considéré bon s'il est :

- Simple et rapide : ces caractéristiques peuvent être vérifiées si l'algorithme génère le nombre aléatoire X_i uniquement à partir de X_{i-1}.
- D'une périodicité aussi longue que possible.

3.1. Méthodes de générations des nombres aléatoires

3.1.1. Méthode congruentielle multiplicative

Le ième nombre généré par cette méthode est donné par la formule récurrente :

$$X_i = kX_{i-1} \bmod(m) \qquad i=1, 2, \ldots.$$

qu'on peut représenter aussi par $X_i = \kappa^i X_0 \bmod(m)$.

D'après cette formule, on voit que pour démarrer le générateur, il faut fixer son état initial qui est donné par :

- X_0 : la racine du générateur
- m : le terme de la congruence, qui est une puissance de 2 ou de 10 le plus souvent.
- k : le multiplicateur.

Un bon choix de ces paramètres, nous permet d'avoir des nombres uniformément distribués sur l'intervalle] 0, m [avec une périodicité maximale égale à m-1 **[3]**.

3.1.2. Méthode congruentielle additive

Elle est connue sous le nom de suite de Fibonacci. Sa formule génératrice est donnée par :

$$X_i = (X_{i-1} + X_{i-2}) \bmod (m).$$

Cette méthode est peu utilisée, car la méthode précédente et celle qui va suivre donne des résultats de meilleures qualités **[3]**.

3.1.3. Méthode congruentielle mixte

Le terme générale est donné par : $\begin{cases} X_{i+1} = (a * X_i + c) \bmod (m) & \forall i \geq 0 \\ X_0 \text{ donné} \end{cases}$

où *a*, *c*, *m* et X_0 sont des entiers **[3]**.

3.2. Validation d'un générateur de nombres aléatoires

Un générateur est dit valide, ou de bonne qualité statistique, s'il vérifie les propriétés suivantes :

- les nombres générés sont uniformément distribués dans l'intervalle [0, 1] : une séquence de nombres est qualifiée ainsi, sur un intervalle quelconque, si ayant subdivisé cet intervalle en un nombre fini de sous intervalles de même longueur, chaque terme de la séquence a la même possibilité(ou chance)de se trouver dans l'un de ces sous-intervalles comme dans un autre.
- Ils sont indépendants : absence de corrélation entre les nombres générés **[3]**.

3.3. Génération de nombres aléatoires suivant une loi.

Dans toute simulation, on a besoin de générer des nombres aléatoires suivant une distribution de probabilité bien déterminée. On appelle ce procédé échantillonnage d'une distribution. Dans ce qui suit, on n'envisage pas de présenter des générateurs propres à des distributions spécifiques, mais on projette juste de rappeler quelques méthodes générales utilisée pour obtenir les échantillons d'une variable aléatoire dont on connaît la distribution de probabilité.

3.3.1. Méthode de transformation (ou d'inversion)

Soit une variable aléatoire X dont la fonction de répartition est : $F(x) = P(X \leq x)$.

Posons $F(x) = y$. On peut démontrer que si Y est uniformément distribuée sur [0,1], alors $X = F^{-1}(y)$ a pour fonction de répartition : $F(x) = P(X \leq x) = P(Y \leq F(x)) = F(x)$.

Donc pour une distribution donnée, il suffit de calculer F^{-1} à partir de F et de déduire X à partir de Y par la formule $X = F^{-1}(y)$ **[S4]**.

3.3.2. Méthode de rejection

Cette méthode est utilisée lorsque la fonction de répartition ne s'inverse pas bien, et lorsque la densité f est connue. Si $f(x)$ est bornée et x appartient à un domaine borné, c'est-à-dire $a \le x \le b$, la méthode de rejet peut être utilisée. Elle se résume en quatre étapes **[S4]** :

1. normaliser le domaine de f à l'aide d'une échelle c de sorte que :
   $$g(x) = c[f(x)] \le 1, \text{ sachant que } a \le x \le b$$
2. définir comme fonction linéaire de u ; $x = a + (b-a)u$
3. générer une paire de nombres aléatoires (u1,u2)
4. à chaque fois que l'on rencontre une paire de nombres aléatoires satisfaisant
   $$u_2 \le c[f(a + (b-a)u_1)]$$
   On accepte : $x = a + (b-a)u_1$
   comme une réalisation x de la variable aléatoire X qui a pour densité $f(x)$.

4. La simulation à évènements discrets

La simulation d'un système passe par les trois étapes suivantes **[S7]**:

a. La Modélisation
   Dans cette partie, on construit le modèle qui doit représenter les interactions les plus importantes. Puis le valider.

b. La Programmation
   Cette partie consiste à exprimer le modèle dans un langage de programmation.

c. La Simulation
   - On détermine les conditions de départ de la simulation (état initial) ;
   - On détermine les données en entrée représentées tout au long de la simulation pour le générateur de nombres aléatoires ;
   - On détermine aussi la durée de la simulation ;
   - On exécute la simulation en recueillant à la fin des résultats sous forme de somme, moyenne, **. . .**

Lors de cette étape, on peut faire des ajustements au niveau du modèle et du programme si c'est nécessaire.

4.1. Terminologie de la simulation à évènements discrets

La terminologie utilisée dans la simulation à évènements discrets est très variée. Elle peut être divisée en deux parties principales **[6]**:

. La première partie fournit des étiquettes pour les objets qui constituent le système à simuler.
. La deuxième partie définit les opérations dans lesquelles ces objets s'engagent à travers le temps.

4.1.1. Objets du système

Entités

Les éléments du système qui est simulé. Ils peuvent être identifiés et traités individuellement.
Exemple :

- Machines dans une usine.
- Des travaux en attente d'exécution.
- Des véhicules.
- Des personnes ou toute autre chose qui change d'état avec l'avancement de la simulation.

Un système est vu comme un ensemble d'entités qui coopèrent pour produire des résultats. Les entités qui restent dans le système au cours de la simulation sont dites *permanentes*. D'autres entités traversent seulement le système. Une fois qu'elles ont quitté le système, elles ne présentent plus d'intérêt. Ces entités sont dites *temporaires* .Certaines entités sont traitées par d'autres. Une entité qui subit des traitements est dite *passive* et celle qui effectue un traitement est dite *active*.

Classes

Les entités de même nature sont groupées dans ce qu'on appelle des classes.

Attributs

Chaque entité peut posséder un ou plusieurs attributs qui véhiculent des informations supplémentaires sur l'entité.

Ensembles

Les entités sont regroupées en classes, mais pendant une simulation elles changent d'état. Ces états sont considérés comme des ensembles. Dans certains cas les ensembles peuvent être considérés comme des files dans lesquelles les entités attendent un traitement.

On peut remarquer que cette terminologie est redondante. Ainsi, l'ensemble auquel une entité appartient à un moment donné peut être considéré comme un attribut de l'entité dont la valeur change à chaque fois que celle-ci change d'ensemble.

4.1.2. Opérations des entités

Au cours d'une simulation, les entités coopèrent pour effectuer certaines opérations et suite à ces opérations elles changent d'états. Une terminologie est nécessaire pour décrire ces opérations et décrire l'écoulement du temps dans la simulation **[6]**.

Evènement

On appelle événement tout instant où un changement significatif apparaît dans le système. Par exemple le moment où une entité change d'ensemble, le début d'une opération. Il appartient à l'analyse de déterminer si un événement est significatif ou pas selon l'objectif de la simulation.

Activité

Les entités s'engagent dans des opérations qui leur font changer d'état donc d'ensemble. Les opérations et les événements qui son déclenchés à chaque événement sont appelées activités. C'est les activités qui transforment l'état d'une entité.

Processus

Des fois, il peut s'avérer utile de regrouper une séquence d'événements en un processus. Ces processus sont généralement utilisés pour représenter toute ou une partie de la vie d'une entité temporaire.

L'horloge de la simulation

Elle représente l'instant présent de la simulation. Ainsi, dans une simulation où l'unité de temps est la minute, on peut utiliser le test : " horloge=250 ?" pour savoir si le temps de lancement d'une activité est arrivé.

4.2. Construction du modèle

Diagramme de cycle d'activités

Dans une simulation à événements discrets les différentes entités interagissent à travers le temps de simulation. Cette interaction peut être décrite par ce qui a été énoncé précédemment.

Pour construire un modèle de simulation il est nécessaire de :

- Identifier les classes importantes d'entités.
- Considérer les activités dans lesquelles elles s'engagent.
- Lier ces activités entre elles.

À partir du squelette obtenu, les détails peuvent être introduits progressivement.
Les diagrammes de cycle d'activité offrent un moyen de modéliser l'interaction des entités et sont particulièrement utiles dans les systèmes avec une forte structure de file d'attente.
Les diagrammes de cycle d'activité utilisent deux symboles seulement qui sont **[6]**:

Figure II .2 : Symboles de diagramme de cycle d'activité

Le diagramme représente le cheminement de chaque entité ainsi que l'interaction entre les entités.
Chaque classe d'entités possède un cycle de vie constitué d'une série d'états. Les entités passent d'un état à un autre au fur et à mesure que leur vie se déroule.
Un état actif généralement induit la coopération de différentes classes d'entités. La durée d'un état actif peut toujours être déterminée en avance (en prenant un échantillon d'une distribution de probabilité appropriée).
Dans un système de file d'attente, le service représente un état actif parce qu'il met en œuvre la coopération d'un serveur et d'un client.

D'autre part, un état mort ne met en œuvre aucune coopération entre les différentes classes d'entités. Généralement, c'est un état où l'entité attend que quelque chose se produise.
Les états morts sont souvent vus comme des ensembles ou des files d'attente. Donc, le temps qu'une entité passe dans un état mort ne peut pas être déterminé en avance. Il dépend de la durée des états actifs qui le précède et le suit immédiatement.

Dessiner un diagramme des cycles d'activités revient à lister les états à travers lesquels chaque classe d'entités passe. Normalement ces diagrammes sont dessinés comme une alternance d'états morts et d'états actifs **[6]**.
Le diagramme complet est formé par la combinaison de tous les diagrammes individuels.

Exemple 1 : une file d'attente et un serveur.

On a deux classes d'entités qui sont :

1) Les clients
2) Le serveur

(1) Les clients

Les clients qui se présentent à la file 1 ont deux états actifs :

" **Arrivée** "

" **Service** "

En dehors de ces états actifs ils sont soit à l'extérieur " **Ailleurs** " soit dans la file d'attente " **File** ".

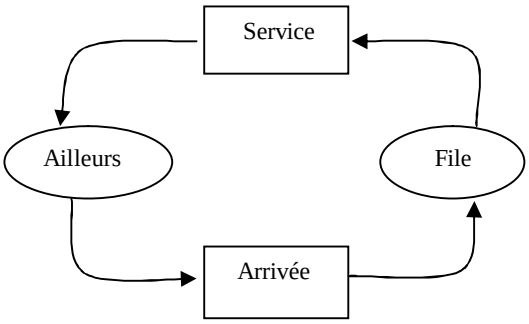


Figure II .3 : diagramme de cycle d'activités (clients).

(2) Le serveur

Le serveur a un état actif qui est :

" **Service client** "

Quand le serveur n'est pas engagé, il est dans un état mort " **Libre** ".

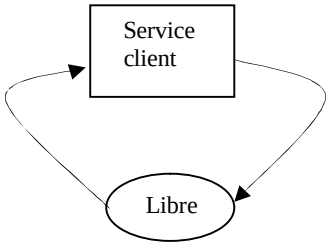


Figure II .4 : diagramme de cycle d'activités (serveur).

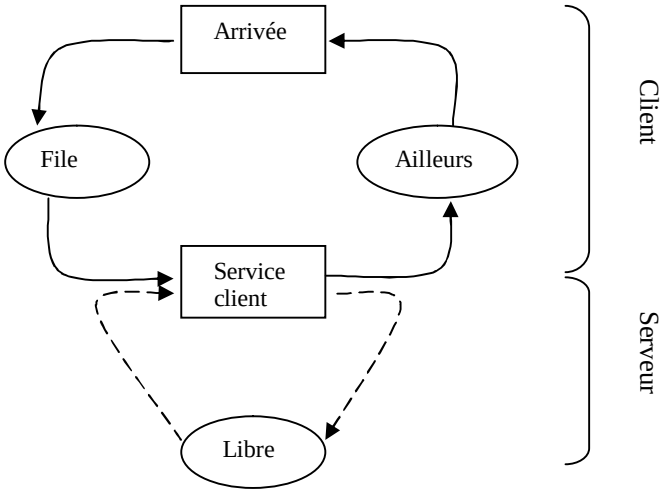


Figure II .5 : Diagramme de cycle d'activités.

Exemple 2 : deux files d'attente et un serveur.

On a trois classes d'entités qui sont :

1) Client1
2) Client2
3) Serveur

(1) Client 1

Les clients qui se présentent à la file 1 ont deux états actifs :

" **Arrivée 1**"

" **Service 1**"

En dehors de ces états actifs ils sont soit à l'extérieur " **Ailleurs**" soit dans la file d'attente " **File 1** ".

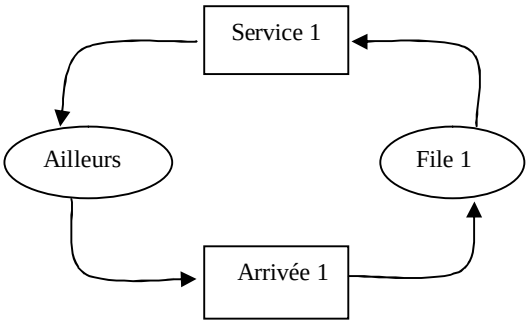


Figure II .6 : diagramme de cycle d'activités (client 1).

(2) Client 2

Les clients qui se présentent à la file 1 ont deux états actifs :

" **Arrivée 2**"

" **Service 2**"

En dehors de ces états actifs ils sont soit à l'extérieur " **Dehors** " soit dans la file d'attente " **File 2** ".

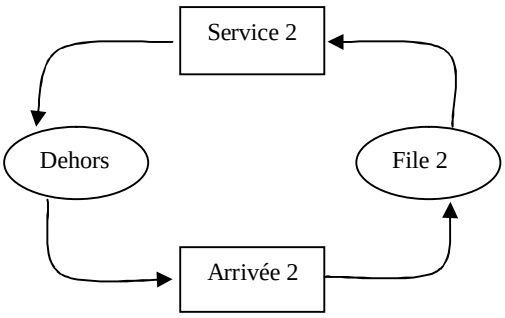


Figure II .7 : diagramme de cycle d'activités (client 2).

(3) Serveur

Le serveur a deux états actifs qui sont :

" **Service client 1** " **et** " **Service client 2** "

Quand le serveur n'est pas engagé, il est dans un état mort " **Libre** ".

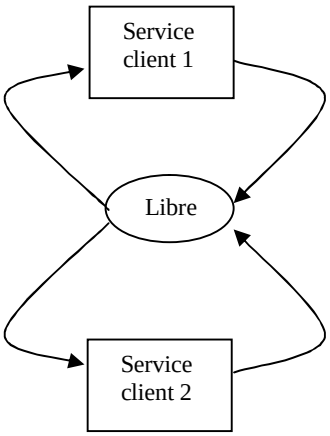


Figure II .8 : diagramme de cycle d'activités (serveur).

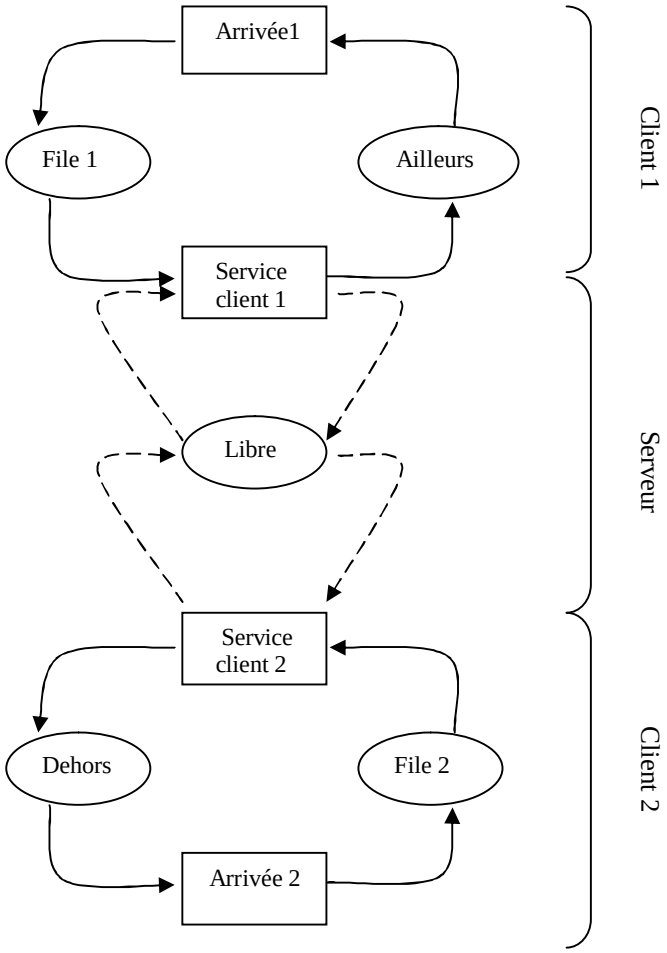
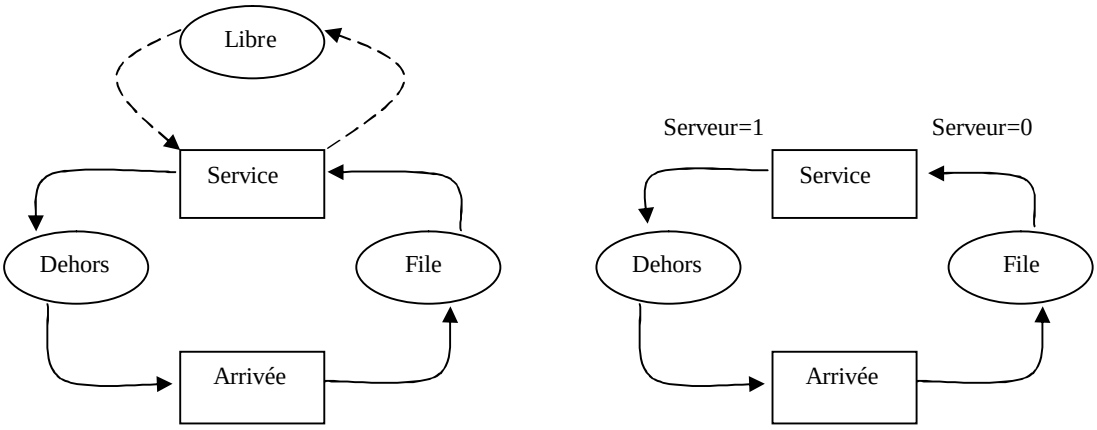


Figure II .9 : Diagramme de cycle d'activités.

Remarque :

Lorsqu'une ressource est limitée on peut utiliser une variable. Dans le cas d'une file d'attente simple où on a un seul serveur ; on peut remplacer le cycle d'activité de serveur par une variable SERVEUR à laquelle on affectera la valeur 1 lorsque le serveur est disponible et la valeur 0 lorsqu'il est occupé.



5. Approches de modélisation

Il existe 4 approches principales **[6]**:

- Approche basée sur les événements « Approche par Événement ».
- Approche basée sur les activités « Approche par Activité ».
- Approche par interaction de processus.
- Approche basée sur les activités et les événements « Approche des trois phases ».

Les quatre approches ont en commun le fait qu'elles produisent toute une structure hiérarchique à trois niveaux :

Niveau 1 : Exécutif (programme de contrôle).
Niveau 2 : Opération.
Niveau 3 : Routines détaillées.

Niveau 1

- Ø Plus haut niveau
- Ø Contrôle les séquences des opérations qui se produisent en cours de simulation.
- Ø Contrôle le niveau 2
- Ø Inclus les routines qui identifient le déclenchement d'un événement et fait en sorte que les procédures correspondantes soient exécutées.

Niveau 2

- Ø Ensemble d'instructions qui décrit les opérations du modèle
- Ø Instructions explicites pour l'ordinateur concernant l'interaction des entités
- Ø Ce niveau constitue le programme de simulation proprement dit
- Ø Concentre l'énergie de l'analyste dans la construction du modèle.

Niveau 3

- Ø Un ensemble de routines utilisées dans le niveau 2
- Ø Ces routines modélisent les détails du système
- Ø Génération de nombres aléatoire
- Ø Production de rapport de simulation
- Ø Collecte de statistiques.
- Ø Ces routines sont appelées par l'analyste à partir du niveau 2.

Pour construire une simulation on dispose de deux alternatives :

Utiliser un système de simulation

- Ø Le programmeur dispose d'un ensemble des sub-routines qui prennent en charge le niveau 1 et 3.
- Ø Le programmeur n'a pas besoin de connaître l'exécutif en détail, la compréhension de l'exécutif suffit.

Utiliser un langage de programmation générale

- Ø Tout est programmé à l'aide d'un langage de programmation comme FORTRAN ou PASCAL
- Ø Tous les niveaux doivent être connus en détail.
- Ø Chaque approche impose sa propre structure du niveau 2
- Ø Chacune des approches impose à l'analyste une méthode de division des opérations du système suivant ses propres composantes de base

Routines d'évènement	à	Approche par évènement
Routines d'activités	à	Approche par activité
Routines de processus	à	Approche par interaction de processus

5.1. Approche par événement

- Ø Cette approche s'intéresse aux changements d'états du système
- Ø Cette approche est populaire au USA
- Ø Le système de simulation SIMSCRIPT est basé sur cette approche
- Ø Le niveau 2 est formé de routines d'évènements
- Ø Chaque routine décrit les opérations dans lesquelles les entités s'engagent quand le système change d'état.
- Ø La routine d'un évènement est définie comme un ensemble d'actions qui peuvent suivre un changement d'état dans le système.

Exemple

Un système de file d'attente :

- Les clients arrivent aléatoirement
- Les clients rejoignent une queue et sont servis sur la base FIFO.

On a deux changements d'états du système :

(1) Arrivée d'un client

(2) Fin de service (départ d'un client)

Chacun de ces évènements nécessite une routine

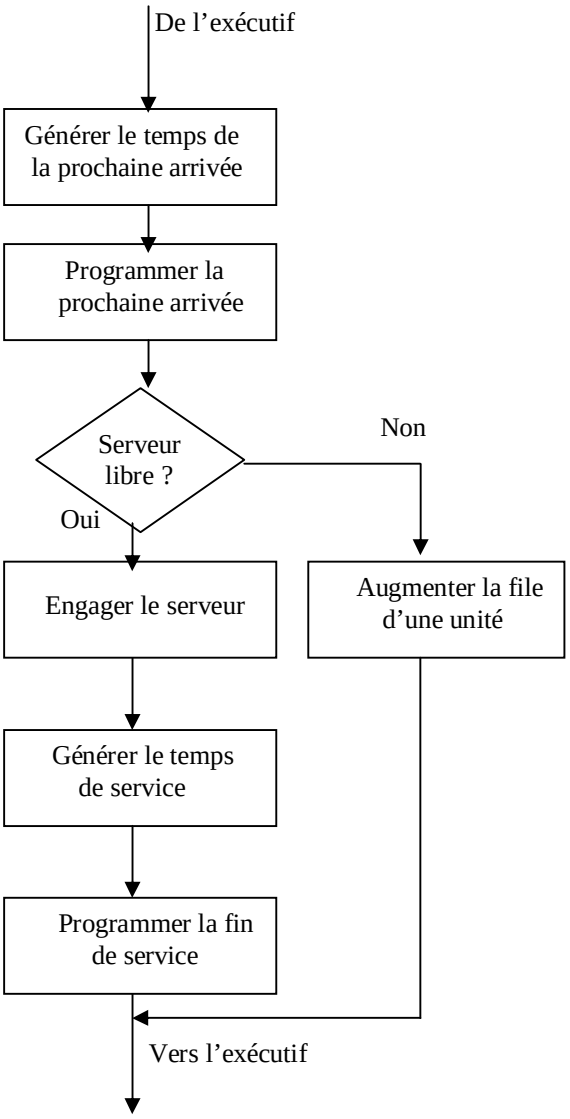


Figure II.10: Routine de l'évènement d'arrivée

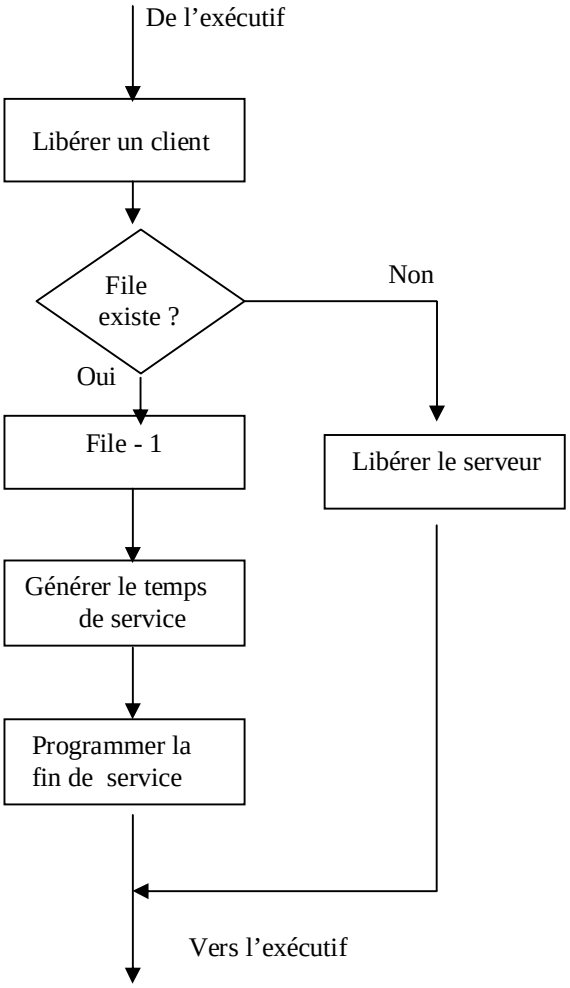


Figure II.11: Routine de l'évènement de fin de service.

Remarque :

- le temps qui sépare deux arrivées successives est généré à l'aide du niveau 3.
- le planning des évènements est géré par l'exécutif.

L'exécutif de l'approche par évènement

Pour contrôler le déroulement d'une simulation, l'exécutif doit effectuer les taches suivantes :

(1) Parcours des temps
- Déterminer le moment de déclenchement de l'évènement suivant.
- Avancer l'horloge à ce moment.

(2) Identification d'évènement
- Identifier correctement les évènements planifiés le moment présent.

(3) Exécution d'évènement
- Exécuter les évènements identifies comme étant programmés pour le moment présent.

Un exécutif simple basé sur une approche par évènement gérera ces taches à l'aide d'une liste d'évènements semblables à un agenda dans lequel les évènements futurs sont reportés. Des évènements sont ajoutés et supprimés au fur et à mesure que la simulation avance.
Chaque évènement reporté sur la liste doit contenir deux informations :
- le moment du déclenchement de l'événement.
- un indice sur le type d'évènement.

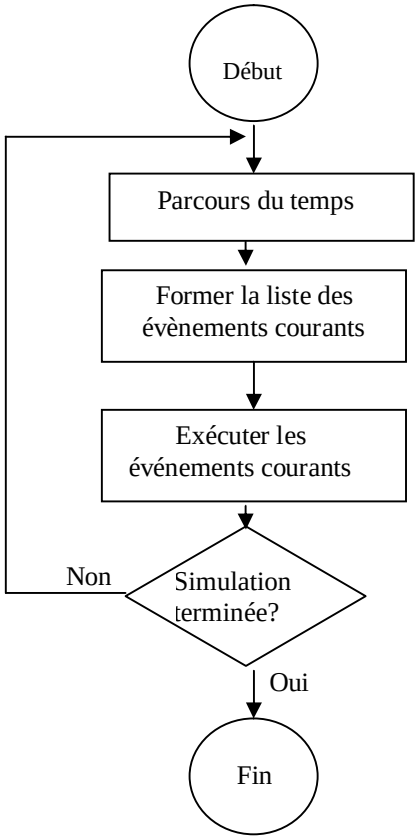


Figure II.12: un exécutif basé sur l'approche par évènements

Une simulation basée sur l'approche par évènement est composée d'une série de procédures de niveau 2. Ces routines sont des segments de programmes dont l'exécution est contrôlée par l'exécutif. Chaque routine décrit exhaustivement les opérations qui peuvent suivre un changement d'état du système.

5.2. Approche par activités

- Ø Approche populaire en UK
- Ø Le système de simulation CSL est basé sur cette approche
- Ø L'approche est basée sur l'interaction des différentes entités : les activités.
- Ø Une activité est l'action qui est déclenchée immédiatement par un changement d'état du système.

Pour l'exemple d'une file d'attente simple on peut définit trois activités :

(1) Arrivée d'un nouveau client.
(2) Début d'un nouveau service.
(3) Fin d'un service.

Chacune des activités a une structure formée de deux parties :

Test d'entrée : constitue la condition à satisfaire pour le déclenchement de l'activité.
Action : les opérations qui constituent l'activité.

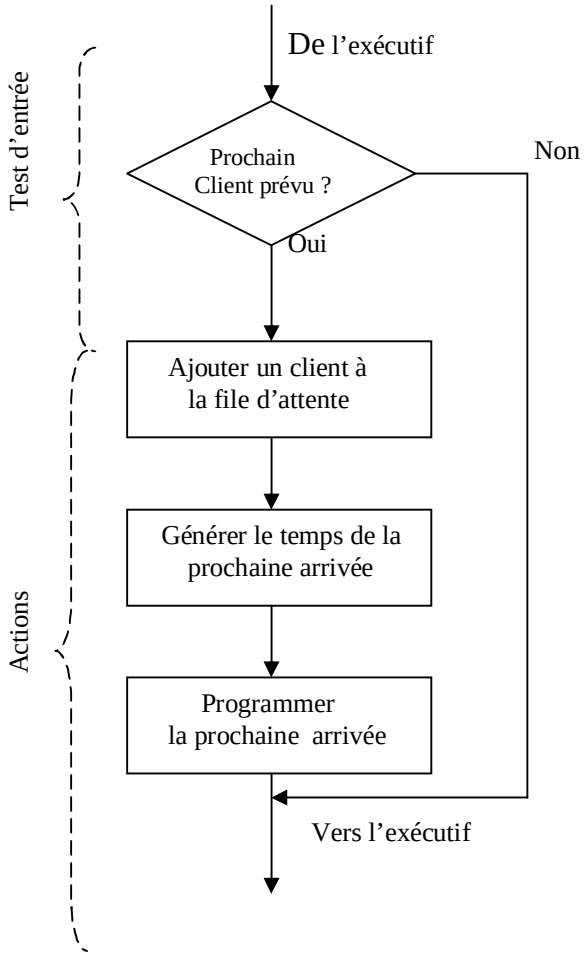


Figure II.13: Arrivée d'un nouveau client

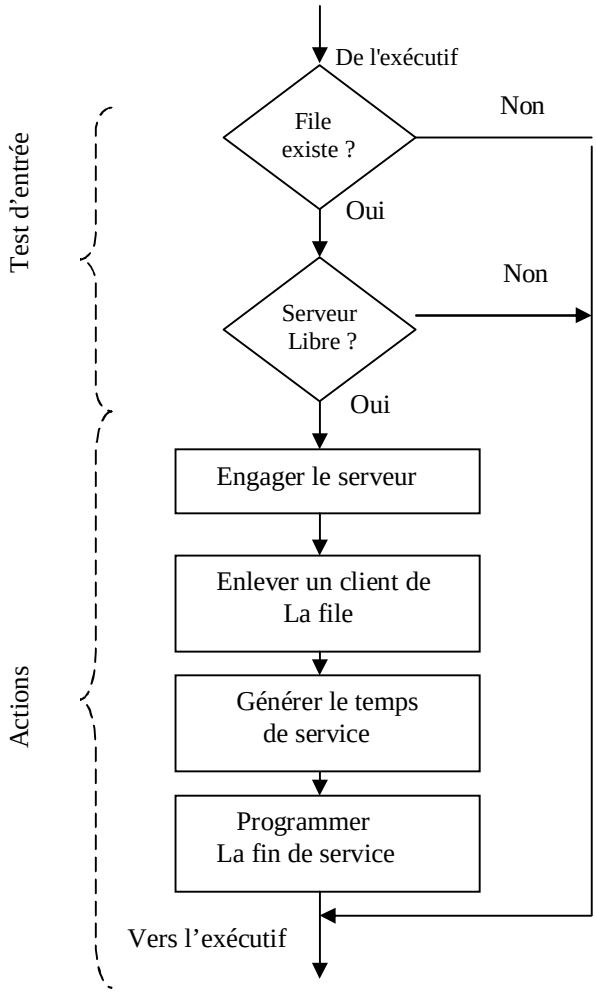


Figure II.14: Début de service

Dans l'approche par activité, chaque activité doit être considérée comme un segment de programme attendant d'être exécuté. L'action sera conduite si et seulement si les tests d'entrées sont passés. Ainsi les actions :

- engagement du serveur
- suppression du client en tête de file
- génération d'un temps de service
- programmation de la fin de service

seront exécutés seulement si

(1) Le serveur est libre

(2) Il y a des clients dans la file d'attente.

Remarque : L'ordre dans lequel les activités sont traitées est important.

L'exécutif d'une approche par activités

Dans une simulation basée sur l'approche par activités, l'exécutif a une tache principale qui est le parcours de temps. Cette tache consiste à identifier le moment du prochain évènement prévu.

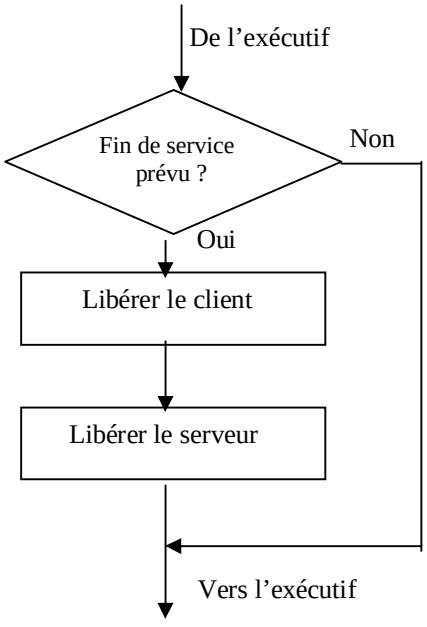


Figure II.15: Fin de service

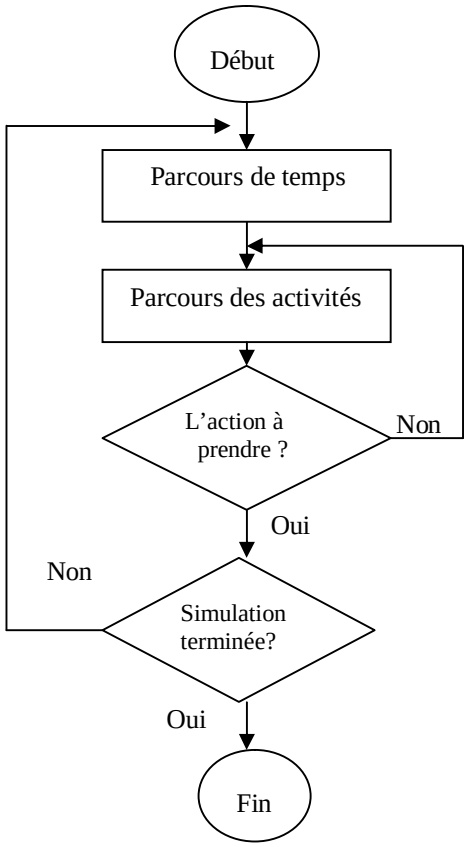


Figure II.16: Exécutif basé sur l'approche par activités.

Pour accomplir cette tache l'approche par activité utilise des cellules de temps en tant qu'attribut des entités permanentes. Cet attribut indique le moment où chaque entité est sensé changer d'état. Ceci est réalisé de deux façons :

(1) En mettant dans la cellule temps le moment de changement d'état de l'entité.

Par exemple :

Serveur.temps := 500 indique que le changement d'état est prévu pour horloge = 500.

Si l'horloge de simulation indique un temps supérieur à la valeur de la cellule temps, ceci indique que l'entité est en attente pour un événement.

Par exemple le serveur est en attente parce qu'il n'y a pas de clients dans la file d'attente.

Dans ce cas le parcours des cellules temps s'effectue de la manière suivante :

POUR toute les entités permanentes avec cellule-temps > Horloge

TROUVER minimum des cellules-temps

Horloge = minimum des cellules-temps

(2) En mettant dans la cellule temps l'intervalle de temps jusqu'au changement d'état de l'entité.

Par exemple :

Serveur.temps := 10 indique que le changement d'état est prévu dans 10 unités de temps.

Un temps cellule négatif indique que l'entité est libre.

Dans ce cas le parcours des cellules temps s'effectue de la manière suivante :

POUR toute les entités permanentes avec cellules-temps > 0

TROUVER minimum des cellules-temps

Horloge = horloge + minimum des cellules-temps

POUR toute les entités permanentes

Cellules-temps = cellules-temps – minimum des cellules-temps

Dans un exécutif simple, les cellules temps peuvent être gérées à l'aide d'une liste **non ordonnée.**

Lors du parcours des temps,

- on ne cherche pas à identifier l'entité qui provoquera le changement d'état.
- on ne cherche pas à identifier l'activité qui est prochainement prévue.

Après le parcours des temps

- l'horloge est avancée au moment du prochain évènement
- et l'exécutif effectue une répétition de parcours des activités jusqu'à ce qu'il ne soit plus possible d'exécuter une quelconque action, c'est-à-dire jusqu'à ce que tous les tests d'entrée soient faux.

5.3. Approche des trois phases

- Ø Première description par Tocher (1963)
- Ø Combine la simplicité de l'approche par activité avec l'efficacité d'exécution de l'approche par évènement.
- Ø L'élément de base est l'activité.

On utilise dans cette approche deux types d'activités

Activité " B " : " Bound activity " **activité forcée** qui est directement exécutée par l'exécutif lorsque son temps arrive.

Activité " C " : activité conditionnelle ou coopérative. Son exécution dépend de la **coopération** de plusieurs entités et de la **satisfaction** de certaines conditions spécifiques.

Chacune des activités est programmée séparément.

Exemple :

Dans un système de file d'attente

L'arrivée des clients est une activité " B "
La fin de service est une activité " B "
Le début de service est une activité " C "

Si on devait produire des diagrammes pour ces activités, on peut, dans ce cas, reprendre les diagrammes des activités de l'approche par activités. La seule différence réside dans les activités " B " qui ne doivent pas contenir de test d'entrée.

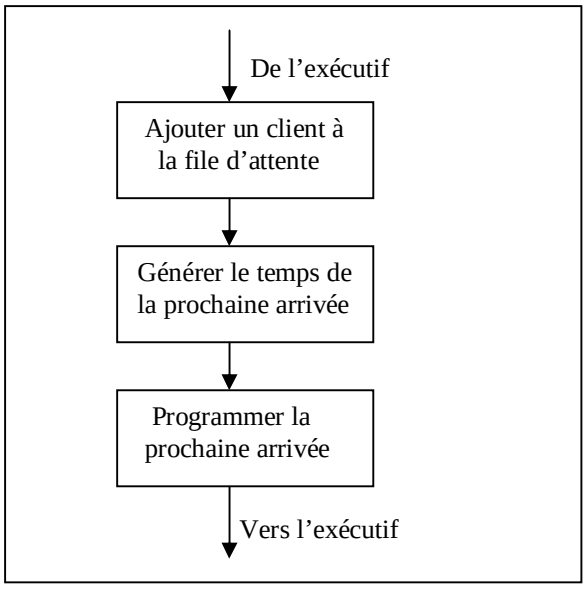


Figure II.17: Arrivée d'un nouveau client

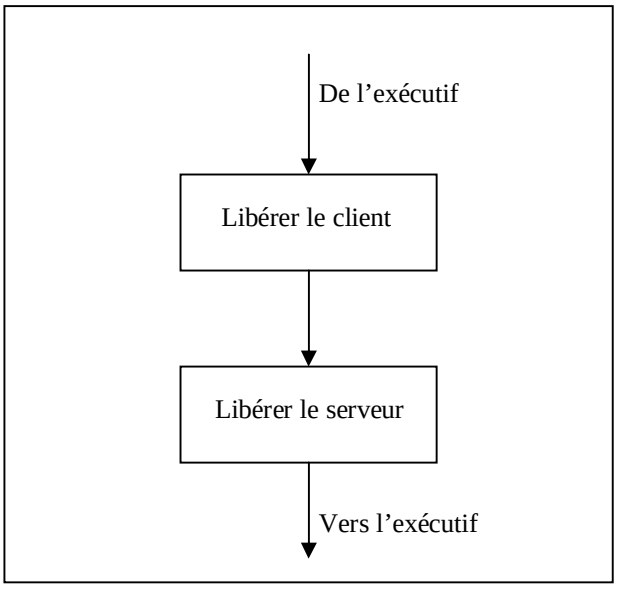


Figure II.18: Fin de service

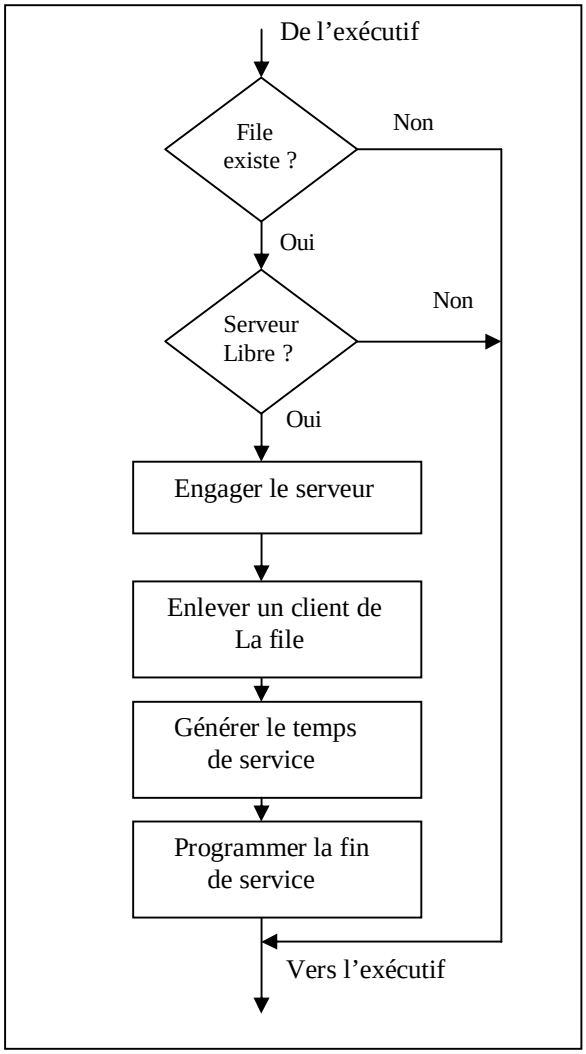


Figure II.19: Début de service

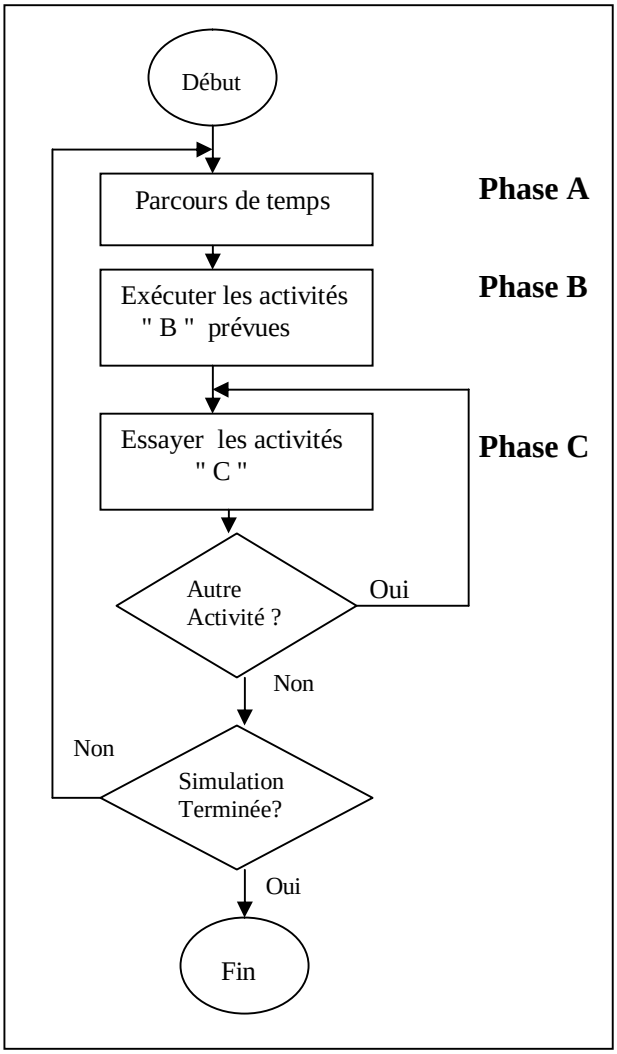


Figure II.20: Exécutif basé sur l'approche des trois phases.

L'exécutif d'une approche de trois phases

Les trois phases sont : phase A, phase B, phase C.

Phase A: Parcours des temps

Détermine le moment de l'évènement prochain et décide lesquelles des activités " B " sont prévues et avance l'horloge de la simulation à ce moment.

Phase B: Appel des activités " B "

Exécute les activités " B " qui ont été identifiées dans la phase " A "

Phase C: Appel des activités " C "

Essaye chacune des activités " C " et exécute celles dont les conditions sont satisfaites, jusqu'à ce qu'il n'y ait plus d'activités "C" à exécuter.

L'implémentation de la méthode des trois phases peut être réalisée en assignant à chaque entité un enregistrement à trois champs

(1)	(2)	(3)
Temps	Activité " B " suivante	Activité précédente

(1) : Le temps prévue pour le changement d'état de l'entité

(2) : Indique l'activité " B " qui est prévue à ce changement d'état. Si l'entité doit prendre part à une activité " C " on met un indicateur pour signifier que la prochaine activité est indéterminée, par exemple -1.

(3) : La dernière activité dans laquelle l'entité est engagée.

6. Evaluation de performances d'un système d'attente par la simulation

Lors de la simulation d'un système de file d'attente, les performances qui sont souvent calculées sont : $\overline{N}, \overline{Q}, \overline{T}, U_j, etU$

Les données nécessaires pour le calcul de ces performances sont les suivantes :

- T : la période de simulation ;
- S : le nombre de serveurs ;
- A : le nombres des arrivées au cours d'une période T ;

- C : le nombre de clients servis au cours d'une période T ;
- t_a : temps de l'arrivée d'un client ;
- t_{ds} : temps du début de service ;
- t_s : Temps de la fin de service ;
- B_j : La somme des intervalles de temps de service de chaque client dans le serveur j,

  $B_j = \sum b_i$, avec $b_i = (t_s - t_{ds})_i$, où i appartient à l'ensemble des clients effectivement servis par le serveur j ;
- Ws_i : l'intervalle de temps qu'un client i passe dans le système

  $Ws_i = (t_s - t_a)_i$
- Wf_i : l'intervalle de temps passé par un client dans la file

  $Wf_i = (t_{ds} - t_a)_i$

Les relations qui permettent de calculer les performances d'une simulation d'un système de file d'attente sont résumées dans le tableau suivant :

Performance	**Calculé par**
u_J	B_j / T
u	$\sum U_J / S$
W_s	$\sum Ws_i / C$
L_s	$\lambda W_s = \sum Ws_i / T$
W_f	$\sum Wf_i / C$
L_f	$\lambda W_f = \sum Wf_i / T$

Tableau II.1 : Calcul des performances d'un système de file d'attente par simulation

7. Avantages et inconvénients de la simulation

L'approche par simulation permet de bénéficier de la plupart des avantages habituels de la modélisation **[S6]**:

- Elle permet de faire varier facilement les paramètres du système et d'évaluer l'impact de ces changements : achat d'une nouvelle machine, etc.
- Elle permet de commettre les erreurs sur le modèle plutôt que dans la réalité, etc.

Par ailleurs, certains avantages sont plus spécifiques à la simulation :

- Elle permet l'évaluation de système pour lesquels on ne dispose pas de solution analytique et pour lesquels la simulation constitue donc la *seule* approche disponible. La simulation est souvent une méthode de dernier recours, utilisée uniquement dans ces conditions.
- Les modèles de simulation sont souvent « visualisable », parfois même avec des possibilités d'animation graphique, ce qui les rend plus acceptables et convaincants aux yeux de décideur que des modèles analytiques plus abstraits (pensez par exemple à des modèles de files d'attente ou de programmation linéaire).

Mais bien évidemment, la simulation connaît également certains inconvénients :

- Chaque 'run' (ou exécution) du modèle est généralement le résultat d'une *expérience aléatoire* et ne produit donc qu'une *estimation* des paramètres étudiés (temps moyen passé dans la file, probabilité de saturation du système, etc.). Ceci est en contraste avec les modèles analytiques (par exemple, les modèles de la file d'attente M/M/S) qui, eux, produisent la valeur exacte de ces paramètres (sous les hypothèses du modèle, bien sûr, et pour autant que le modèle soit résoluble analytiquement).
- Le développement de modèles de simulation requiert du temps et des ressources considérables.
- Lorsqu'un modèle a été développé, la flexibilité de l'outil ainsi que son acceptabilité peuvent engendrer des effets pervers : par exemple, une confiance *injustifiée* dans l'exactitude des résultats calculés (renforcé par l'aspect 'visuel', les animations, etc, qui tendent à réduire la distance entre le modèle et l'objet simulé).
- Le temps de simulation est énorme pour atteindre une stabilisation des résultats.

Chapitre III

L'APPLICATION

Dans ce chapitre nous allons modéliser notre système (serveur web) par un système de réseau de files d'attente, puis le résoudre en utilisant la simulation.

Le simulateur conçu est programmé à l'aide du langage de programmation PASCAL sous l'environnement DELPHI. Il permet d'évaluer les différentes performances du système, ainsi que prédire son comportement après variation de certains paramètres (taux d'arrivée, taux de service).

---

1. Présentation du problème

On considère le modèle simplifié suivant pour un réseau de files d'attente :

- Les requêtes de documents arrivent selon un processus de Poisson de taux λ .
- Le temps de traitement correspondant est supposé exponentiellement distribué avec le paramètre μ_1

Ce système ayant quatre sous systèmes, chaque sous système se compose d'une file d'attente et un serveur tel que :

Le premier représente le cache
Le deuxième représente un disque dur nommé 2
Le troisième représente un disque dur nommé 3
Le quatrième représente la sortie des requêtes analysées.

Lorsque il y a arrivée d'une requête de document dans la file1, et quand le document n'est pas dans le cache, il doit être lu sur un des deux disques avec une certaine probabilité. Avec probabilité $p_{1,4}$ le document est dans le cache. De même, une fois lu sur un des disques, un document est directement envoyé sur le réseau. La discipline de service sera dans ce cas FIFO. Le réseau de files d'attente est représenté dans la figure suivante :

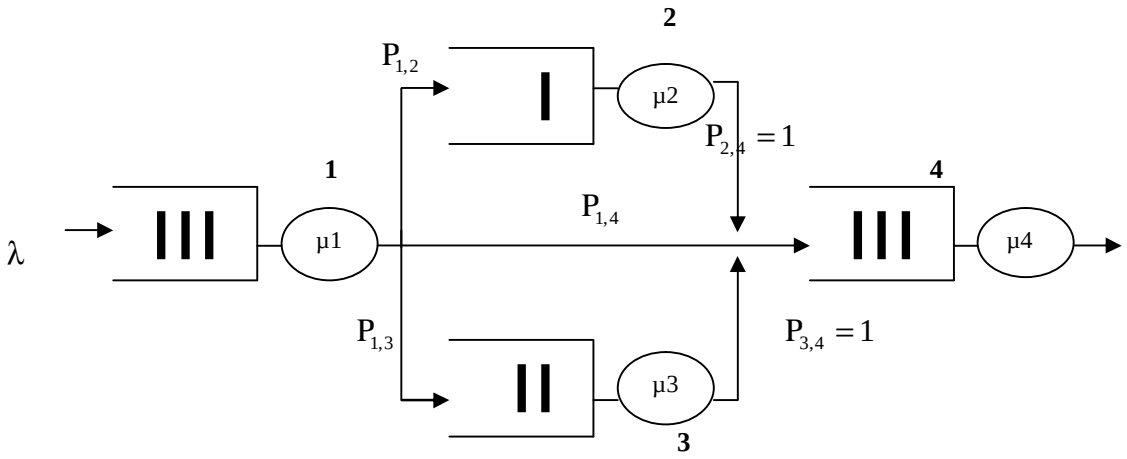


Figure III.1 : Modèle de Réseau de files d'attente.

2. Résolution du problème à l'aide de la simulation

On peut définir pour ce système les classes d'entités suivantes :
Clients (requêtes), serveur1, serveur2, serveur3, serveur4.

(1) *Clients* (requêtes)*:*

- états actifs : "arrivée file1", "service1", "sevice2", "service3", "service4".
- états inactifs : "dehors", "attente file1", "attente file2", " attente file3", " attente file4".

(2) *Serveur i, i=1 ,2 ,3 ,4 :*

- états actifs : "service i"
- états inactifs : "libre"

Le diagramme de cycle d'activités pour chacune des classes est comme suit :

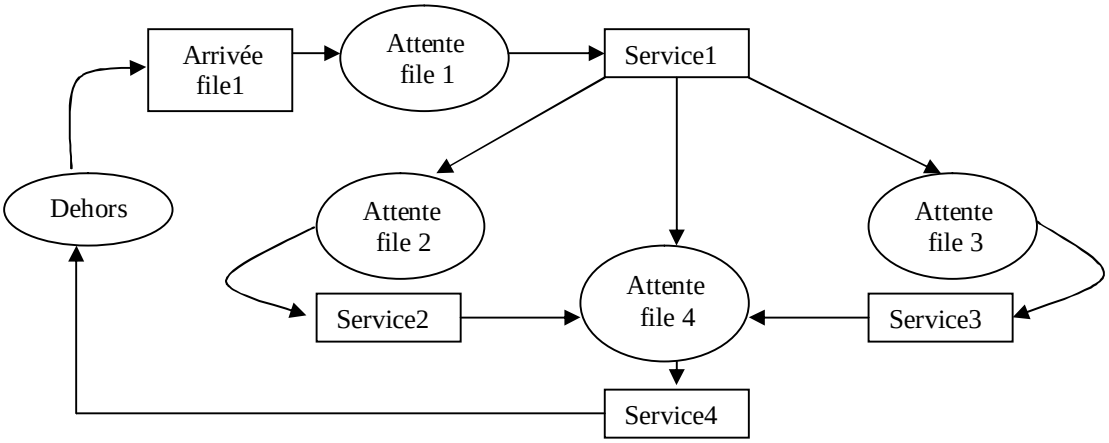


Figure III.2 : Diagramme de cycle d'activités (clients).

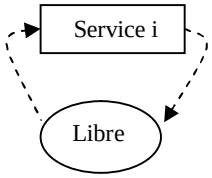


Figure III.3 : Diagramme de cycle d'activités du serveur i ; i= 1, 2, 3 ,4.

Ces diagrammes peuvent être combinés en un seul pour donner le diagramme de cycle d'activités complet.

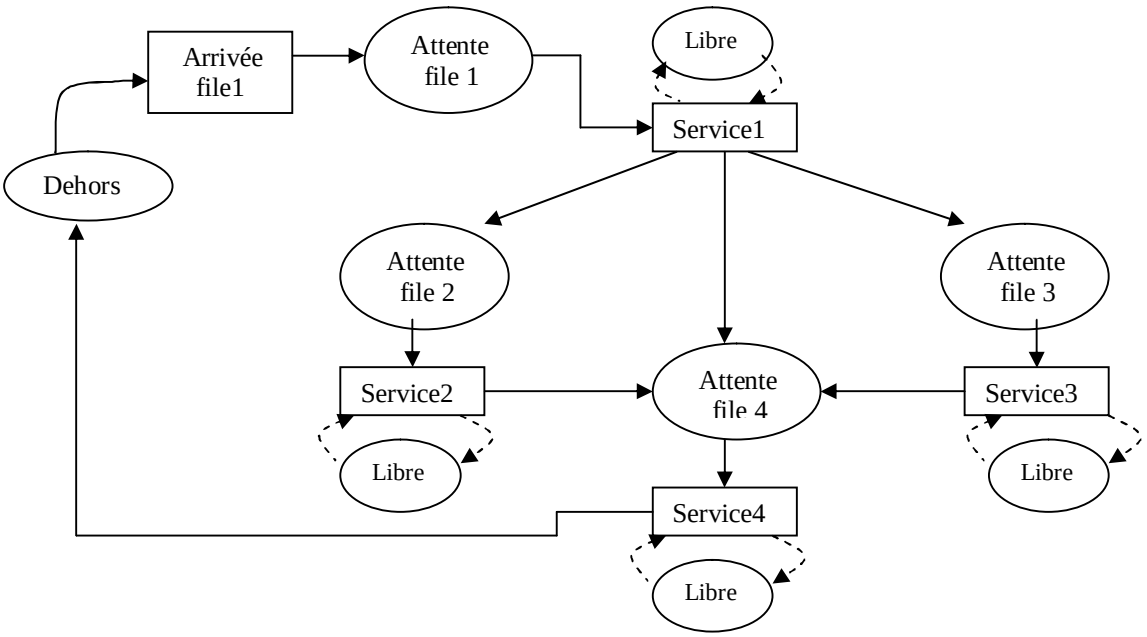


Figure III.4 : Diagramme de cycle d'activités.

2.1. Utilisation de l'approche des trois phases

Lors de l'implémentation du modèle de simulation à évènement discret les activités à prendre en considération sont :

- Les activités de type "B" sont :
    - Arrivée dans la file1
    - Fin de service i, i= 1, 2, 3, 4.
- Les activités de type "C" sont :
    - Début de service i, i= 1, 2, 3, 4.

Chacune de ces activités est décrite par les organigrammes dans les figures qui suivent :

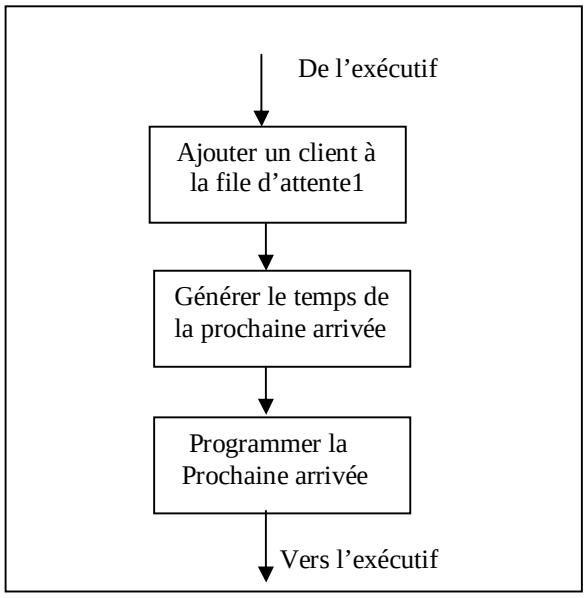


Figure III.5 : Arrivée dans la file 1.

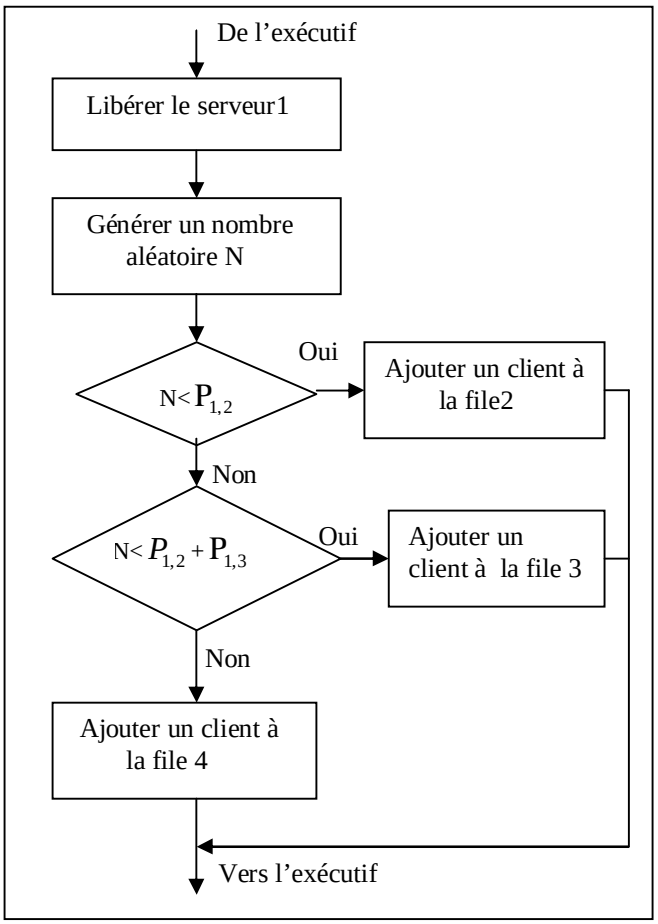


Figure III.6 : Fin de service 1.

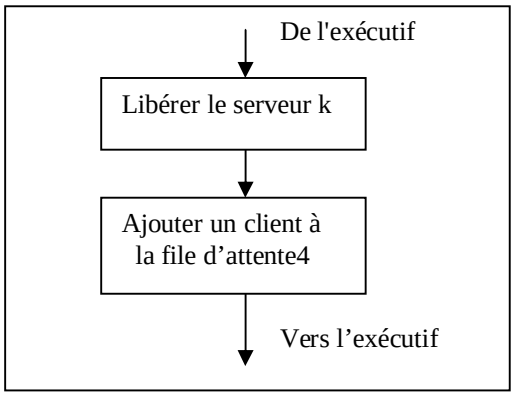


Figure III.7 : Fin de service k, k=2, 3.

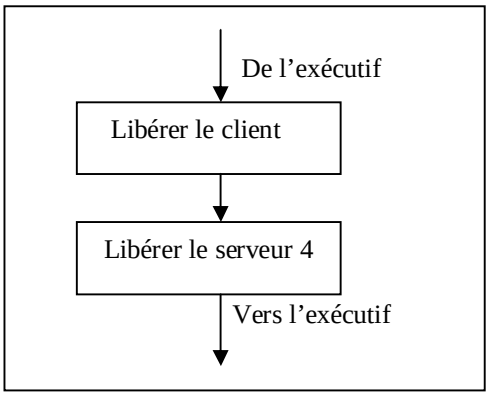


Figure III. 8 : Fin de service 4.

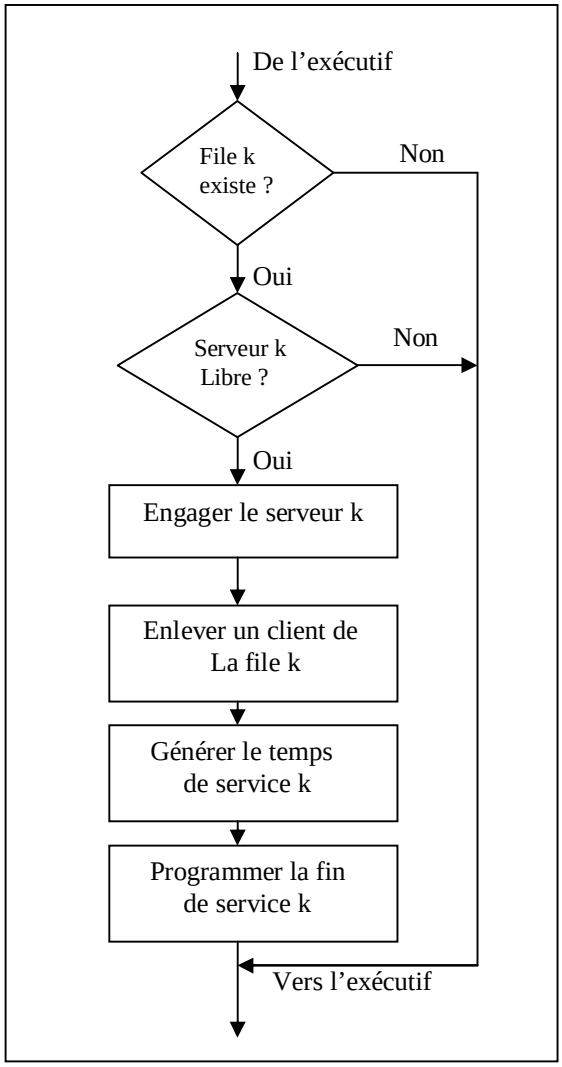


Figure III.9 : Début de service k, k= 1, 2, 3, 4.

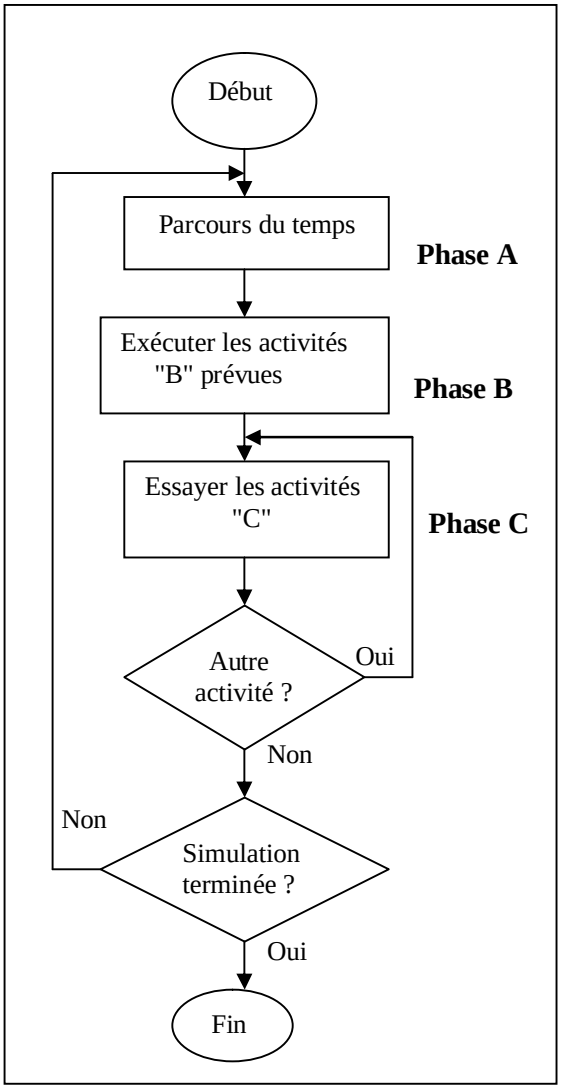


Figure III.10 : Un exécutif basé sur l'approche des trois phases.

2.2. Génération des variables aléatoires

Dans notre étude on a utilisé six générateurs chacun a ses propres paramètres

$$x_{n+1} = ax_n + b[m]$$

Où:

Générateur 1 :	a=134775813;	b=1;	m=229467296;
Générateur 2 :	a=144775813;	b=2;	m=329467296;
Générateur 3 :	a=154775813;	b=3;	m=429467296;
Générateur 4 :	a=164775813;	b=4;	m=529467296;
Générateur 5 :	a=174775813;	b=5;	m=629467296;
Générateur 6 :	a=184775813;	b=6;	m=729467296;

Pour obtenir une suite de nombres aléatoires entre 0 et 1, il suffit de prendre :

$$U_n = X_n / m \qquad n \geq 0$$

Et on a appliqué la méthode d'inversion qui permet de :

- Tirer une variable u uniforme sur]0,1[;
- Prendre $x = F^{-1}(u)$

Tel que $F(x) = 1 - e^{-\lambda x}$, on a alors $x = -\lfloor 1/\lambda \rfloor \log(1-u)$.

3. Présentation du simulateur

Le simulateur a été élaboré à l'aide du langage de programmation Pascal sous l'environnement Delphi. Il a pour objectif de calculer le comportement du système pour une période de simulation T. Il permet de donner les principaux indicateurs de performances suivants :

- Nombre total de requêtes servis ;
- Longueur moyenne de chaque file ;
- Longueur maximum de chaque file ;
- Durée moyenne de séjour d'une requête dans le système ;
- Taux d'occupation de chaque serveur ;
- Durée moyenne d'attente d'une requête dans chaque sous système ;
- Durée moyenne de service d'une requête dans chaque sous système ;
- Durée moyenne de séjour d'une requête dans chaque sous système ;

3.1. Structure de donnée modélisant les files d'attente

On a utilisé les listes chaînées qui représentent une structure de données très adaptée pour gagner de l'espace mémoire et faciliter la manipulation des données. On a utilisé quatre listes chaînées, chacune des listes correspond à un sous système.
L'élément de base de ces listes est un enregistrement de type RECORD qui possède 3 champs qui sont :

- Temps d'arrivée dans le système (le moment où la requête est arrivée dans la file1);
- Temps d'arrivée dans la $File_i$ (l'instant où la requête est arrivée dans l'une des files);
- Pointeur vers l'enregistrement suivant ;

A chaque arrivée un nouvel enregistrement est ajouté à la première liste. A la fin du premier service, le premier élément de la première liste est enlevé de la première liste et il est ajouté dans la deuxième liste et ainsi de suite.

Le schéma suivant illustre la structure d'une liste pour une file.

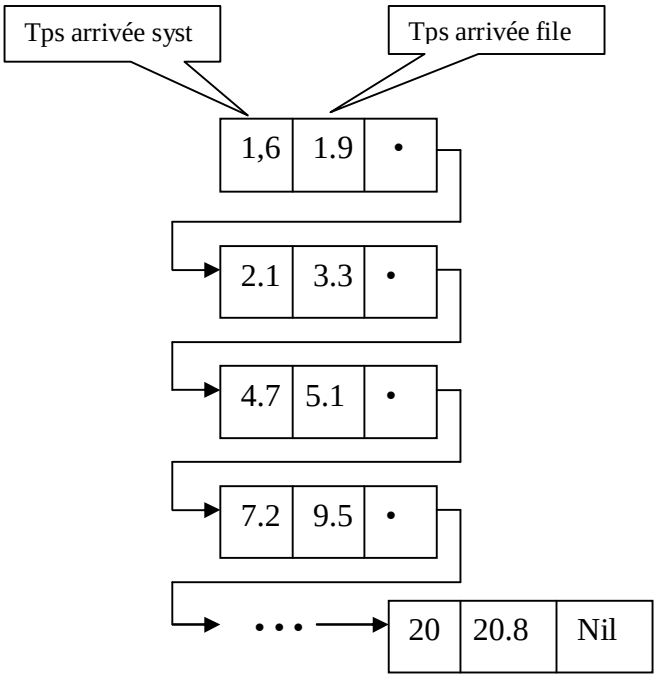


Figure III.11 : La structure de donnée utilisée dans le simulateur

3.2. Algorithme de simulation

Cet algorithme est basé sur l'approche des trois phases, comme nous l'avons indiqué précédemment. La simulation est gérée par l'exécutif que nous avons décrit plus haut (Figure III.10). Les activités à prendre en considération sont :

" B1", " B2 ", " B3 ", " B4 ", " B5 ", " C1 ", " C2 ", " C3 ", " C4 ";

1. Activité "B1" (arrivée dans la $File_1$) :
   - Augmenter la $File_1$ d'une unité.
   - Générer le temps de la prochaine arrivée.
   - Programmer la prochaine arrivée.
   - Mémoriser la date d'arrivée dans le système et le $sous\ \ système_1$ en l'insérant dans la $liste_1$.

2. Activité "B2" : (fin de $service_1$)
   - Libérer le $serveur_1$
   - Générer un nombre aléatoire pour diriger le client vers une des files selon les probabilités de chaque sous système.
   - Augmenter la file correspondante
   - Mémoriser la date d'arrivée dans le sous système et le système
   - Supprimer la tête de la $liste_1$
   - calculer le temps passé dans le sous $Système_1$ qui est donné par la formule suivante :

     Temps passé dans le sous $Système_1$ = horloge - temps d'arrivés dans la $File_1$

3. Activité "B_j" (fin de $service_i$)
   - Libérer le $Serveur_i$.
   - Augmenter la $File_4$.
   - Mémoriser la date d'arrivée dans le sous système et le système.
   - Supprimer la tête de la $liste_i$
   - Calculer le temps passé dans le sous $Système_i$ qui est égal à :

     Temps passé dans le sous $Système_i$ = horloge - temps d'arrivés dans la $File_i$

     Où i (j) peut prendre la valeur 2 ou 3 (3 ou 4) respectivement.

4. Activité "B5" : (fin de $service_4$)
   - Libérer le $Serveur_4$
   - Calculer le temps passé dans le sous $Système_4$ et le système.

     Temps passé dans le sous $Système_4$ = horloge - temps d'arrivés dans la $File_4$

     Temps passé dans le *Système* = horloge - temps d'arrivés dans le *Système*

   - Incrémenter le nombre de clients servis qui est le nombre de clients sortant du système.

5. Activité "C_i" (Début de $service_i$)

Si le $Serveur_i$ est libre et s'il y a une $File_i$:

- Engager le $Serveur_i$.
- Enlever un client de la $File_i$
- Générer le temps de $service_i$.
- Programmer la fin de $service_i$
- Calculer le temps de $Service_i$
- Calculer le temps d'attente dans la $File_i$

Temps d'attente dans la $File_i$ = horloge - temps d'arrivés dans la $File_i$

3.3. Interface du simulateur

L'interface du simulateur d'un réseau de files d'attente :

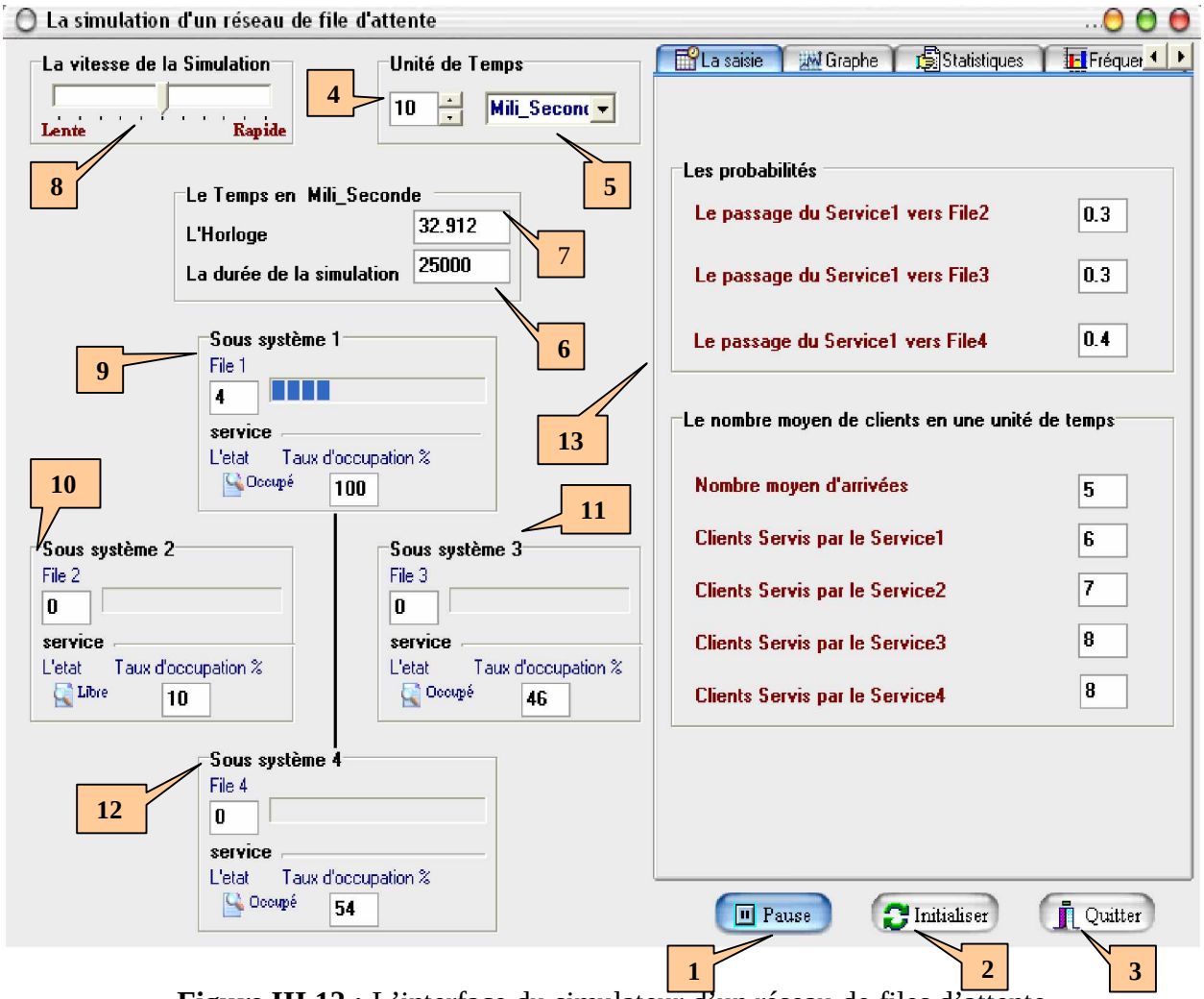


Figure III.12 : L'interface du simulateur d'un réseau de files d'attente.

Les boutons de commande :

(1) Lancer : pour commencer et mettre en pause la simulation,

(2) Initialiser : pour commencer une nouvelle simulation,

(3) Quitter : pour quitter définitivement le programme.

(4) permet de choisir la valeur de l'unité de temps associé aux taux d'arrivées et de service.

(5) pour choisir l'unité de temps (microseconde, milliseconde ou seconde).

(6) pour déterminer la durée prévue de simulation

(7) l'horloge : l'instant présent de la simulation

(8) réglage de la vitesse de la simulation (lente / rapide).

(9) (**10, 11 et 12**) affichage de l'état d'un sous système (on a 4 sous système possibles), et chaque sous système a ses propres données telles que :

- la file : montre la longueur de la file et un indicateur visuel de celle-ci
- l'état du serveur : libre ou occupé
- le taux d'occupation : le pourcentage du temps pendant lequel le serveur a été occupé.

(13) Des pages pour saisir les données de la simulation et afficher les résultats obtenus, on a la page Saisie, la page Graphe, la page Statistique, la page Fréquences

La page Saisie (Figure III.13) :

(14) on saisie les probabilités associés au routage à la sortie du sous système 1

(15) on saisie le taux d'arrivées et les taux de services pour chaque sous système

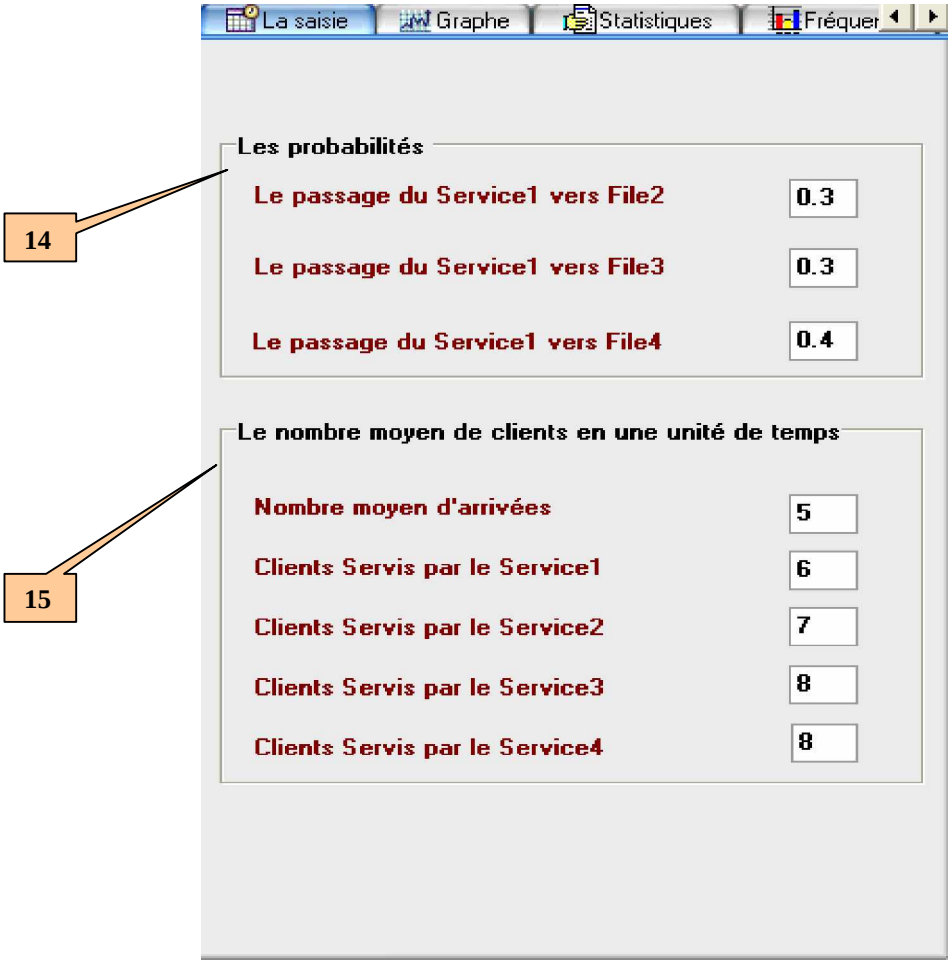


Figure III.13 : Les données nécessaires pour lancer la simulation.

Pour **(14)** si la somme des probabilités de routage est différente de 1 : un message d'erreur est affiché

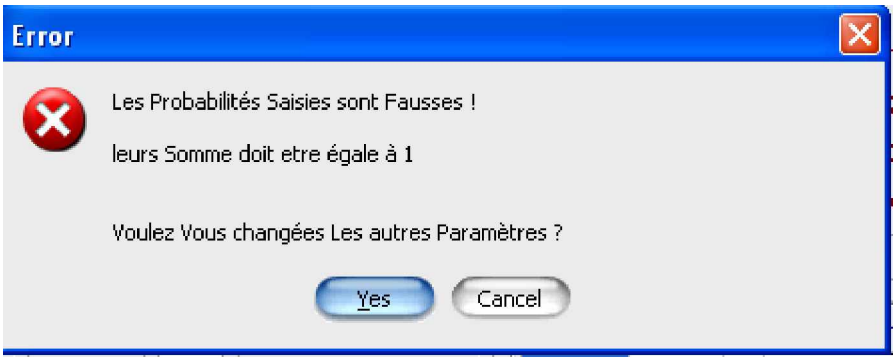


Figure III.14 : Le message d'erreur.

Pour **(15)** si la condition d'équilibre " Taux d'arrivées< Taux de sorties" n'est pas vérifie : un message d'avertissement est affiché.

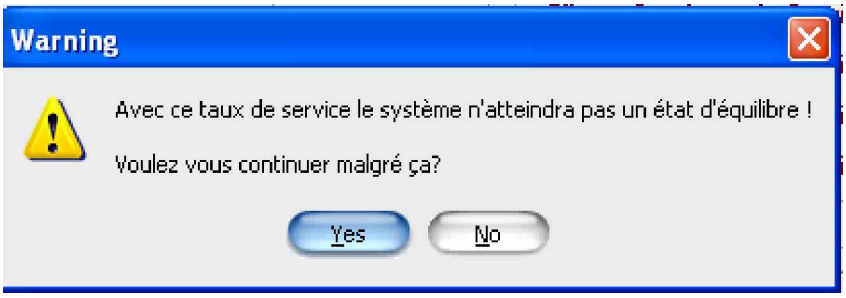


Figure III.15 : Le message d'avertissement.

La page Graphe : On a affiché pour chaque file sa longueur en fonction du temps (l'horloge) et un graphe des longueurs maximales de la file en fonction du temps (l'horloge).

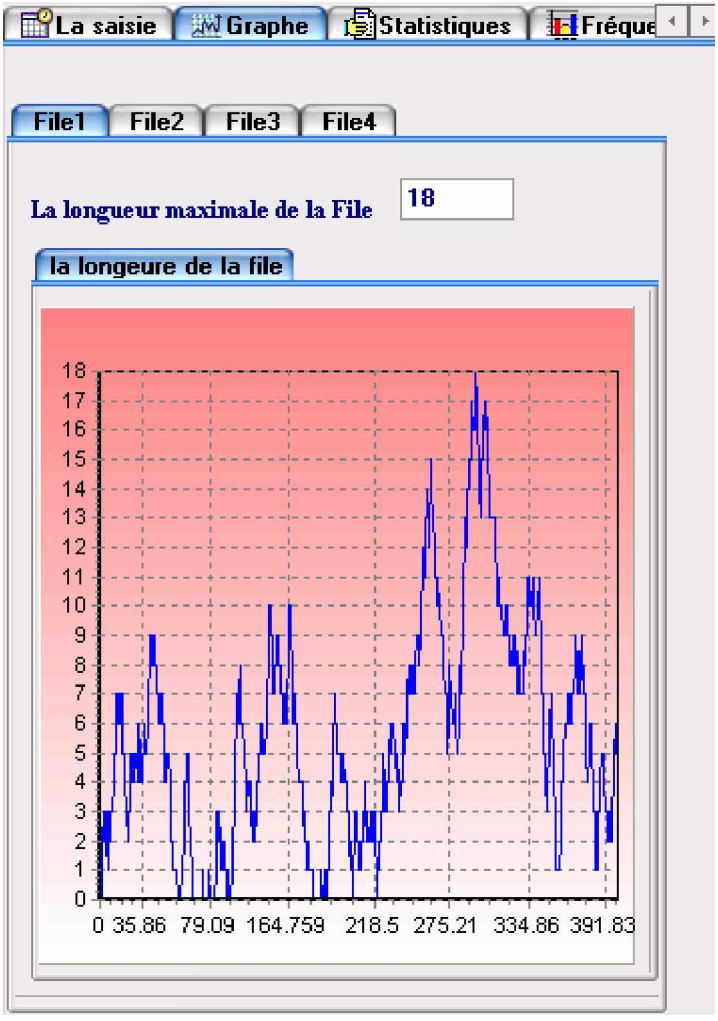


Figure III.16 : Le graphe de la longueur de la file1.

La page Statistiques : On a affiché les résultats obtenus par la simulation, pour chaque sous système on a calculé :

- le temps d'attente moyen dans la file et sa variance
- le temps de service moyen
- le nombre moyen dans la file et sa variance
- le nombre moyen dans le sous système et sa variance
- le temps moyen de séjour dans le sous système et sa variances
- le nombre total des arrivées dans le sous système
- le nombre total de clients servis

On a affiché aussi les valeurs théoriques pour les comparer avec les performances calculées par la simulation.

Et pour le système on a calculé :

- le temps de réponse moyen et sa variance
- le nombre moyen de client et sa variance
- les entrées totales
- le nombre total de clients servis.

Figure III.17 : Les résultats du simulateur pour le sous système1 et le système

La page Fréquence : On a représenté :

- Les temps d'attentes moyen dans chaque file (le temps passé par la requet dans la file jusqu'a la début de service) sous forme d'un histogramme.
- Le temps de service moyen pour chaque service sous forme d'un histogramme
- Le temps de séjour dans le système (le temps passé dans tout le système)
- Les fréquences des files (la densité sa veut dire le nombre de fois qu'on a I clients dans la file).

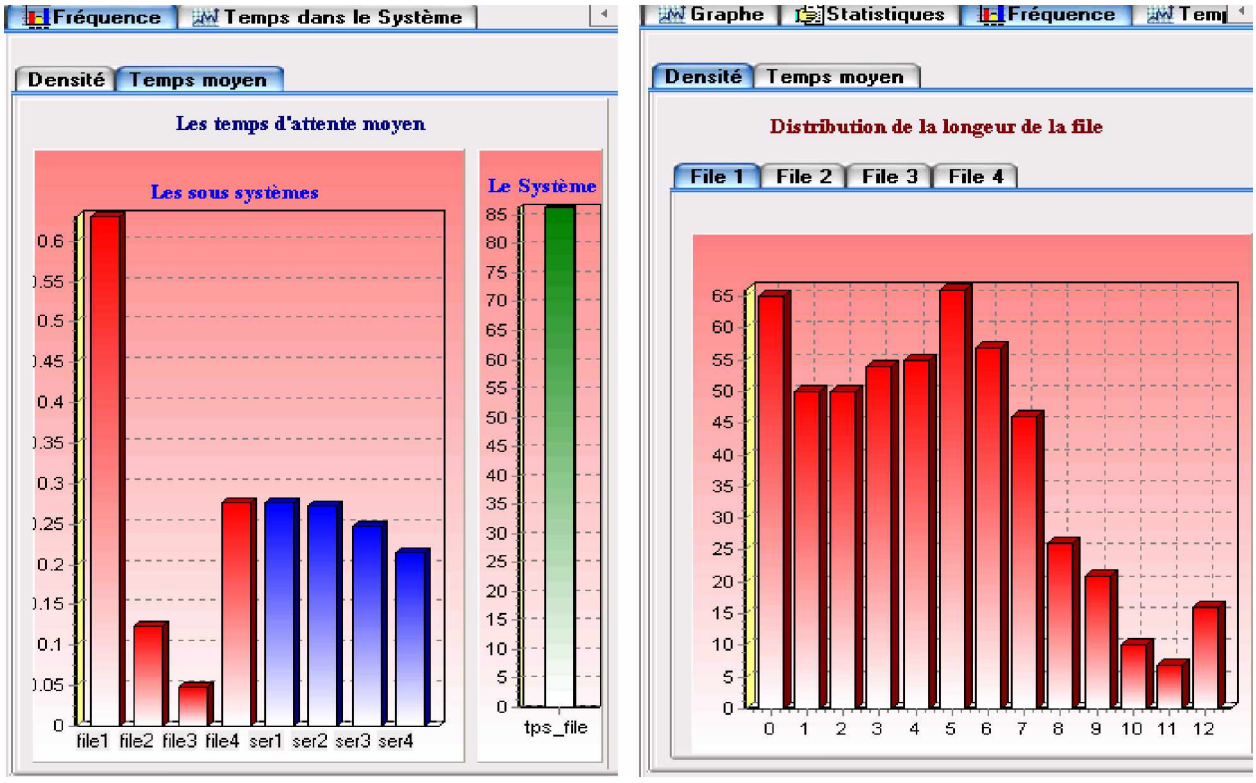


Figure III.18 : l'histogramme des temps moyens et la densité de la file1.

La page Temps dans le système : on a affiché les temps moyens dans le système sous forme d'un graphe.

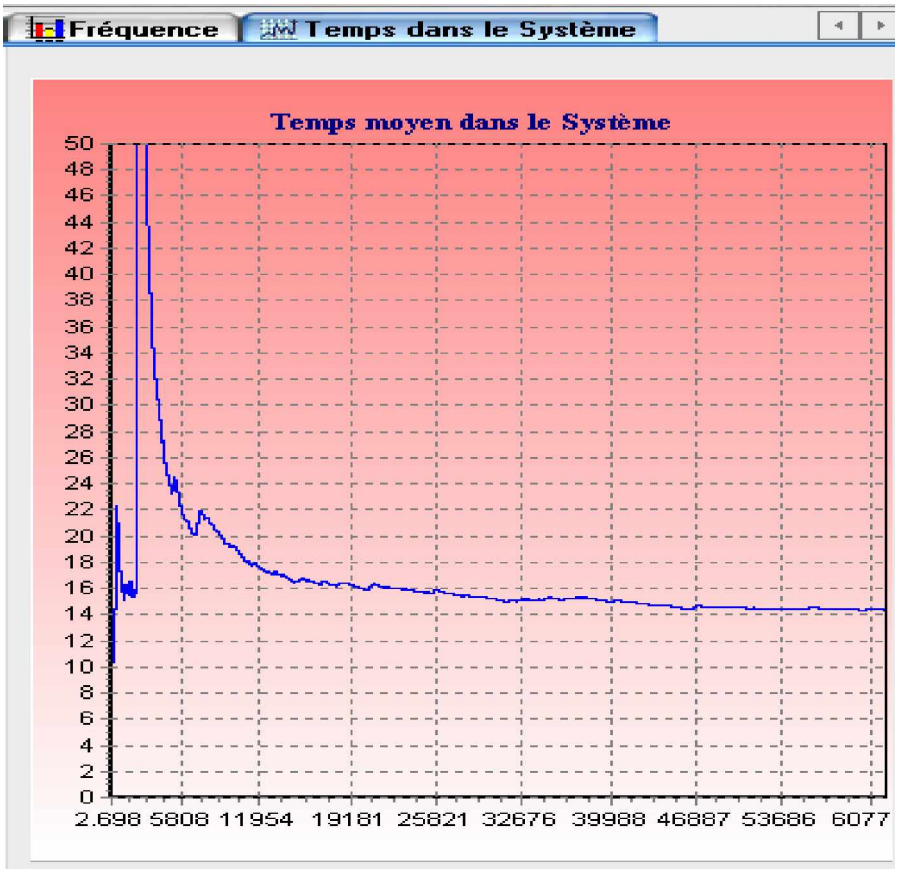


Figure III.19 : Graphe du temps moyen d'attente dans le système.

4. Etude de cas

On a choisit cinq cas de configuration du serveur web à étudier.

Le 1er cas : un disque et pas de cache

Dans ce cas les requêtes de documents arrivent et elles sont analysées et doivent être lu sur un disque (on choisit un des disques 2 ou 3), une fois lu sur un disque le résultat de la requête est directement envoyé sur le réseau.

Les probabilités du passage saisies sont :

- $p_{1,2} = 1$ disque1
- $p_{1,3} = 0$ disque2
- $p_{1,4} = 0$ cache

Le 2ème cas : deux disques et pas de cache

Dans ce cas les requêtes de documents arrivent et elles sont analysées et elles doivent être lu sur un des deux disques *i* avec une probabilité $P_{1,i}$. Une fois lue sur un des disques le résultat de la requête directement envoyé sur le réseau vers la file des requêtes analysées.

Les probabilités du passage saisies sont :

- $p_{1,2} = 0.5$ disque1
- $p_{1,3} = 0.5$ disque2
- $p_{1,4} = 0$ cache

On a choisi d'accorder la même importance à chacun des deux disques.

Le 3ème cas : un cache seul

Dans ce cas on n'a aucun disque dur on a seulement le cache alors les requêtes de documents arrivent et elles sont analysées et directement envoyées vers la file des requêtes servies.

Les probabilités du passage saisies sont :

- $p_{1,2} = 0$ disque1
- $p_{1,3} = 0$ disque2
- $p_{1,4} = 1$ cache

Le 4ème cas : un cache et un disque

Dans ce cas les requêtes de documents arrivent et elles sont analysées et dirigées vers le cache avec une probabilité $P_{1,4}$ et vers le disque dur i (on prend dans ce cas 1 ou 2) avec une probabilité $P_{1,i}$. Le résultat obtenu est alors envoyé sur le réseau. Les probabilités du passage saisies sont :

- $p_{1,2} = 0.5$ disque1
- $p_{1,3} = 0$ disque2
- $p_{1,4} = 0.5$ cache

Le 5ème cas : un cache et deux disques

Dans ce cas les requêtes de documents arrivent et elles sont analysées et dirigées vers le cache avec une probabilité $P_{1,4}$ et vers un des disques durs i avec une probabilité $P_{1,i}$. Le résultat obtenu

est alors envoyé sur le réseau. Les probabilités du passage saisies sont :

- $p_{1,2} = 0.3$ disque1
- $p_{1,3} = 0.3$ disque2
- $p_{1,4} = 0.4$ cache

Nous avons exécuté la simulation pour différentes périodes de temps pour observer le comportement de certains paramètres d'évaluation. Nous avons choisi des périodes de 10000, 20000, 30000, 40000, 50000 et 60000 unités de temps.
Pour étudier le système à l'état d'équilibre, il est nécessaire d'ignorer les premiers instants de la simulation. Nous avons considéré que la période de 0 à 2000 unités de temps est une période transitoire nous l'avons donc ignorée et nous avons commencé la collecte des informations à partir de la fin de cette période.

Les taux d'arrivés et de service saisis

1) le taux d'arrivée dans la file1 est égal à cinq arrivées par unité de temps qui est égal à dix millisecondes par défaut donc $\lambda = 0.5$.
2) le taux de service1 qui est le nombre de client servis par le serveur1 par unité de temps (l'unité de temps est égale à dix millisecondes) est égal à six donc $\mu_1 = 0.6$.
3) le taux de service2 qui est le nombre de client servis par le serveur2 par unité de temps (l'unité de temps est égale à dix millisecondes) est égal à sept donc $\mu_2 = 0.7$.
4) le taux de service3 qui est le nombre de client servis par le serveur3 par unité de temps (l'unité de temps est égale à dix millisecondes) est égal à huit donc $\mu_3 = 0.8$.
5) le taux de service4 qui est le nombre de client servis par le serveur4 par unité de temps (l'unité de temps est égale à dix millisecondes) est égal à huit donc $\mu_4 = 0.8$.

Les résultats obtenus pour le cas d'un cache et deux disques sont présentés dans le tableau ci-dessous et les autres cas sont présentés en annexe. On remarque que les valeurs obtenues commencent à se stabiliser à partir d'une durée égale à 50000 unités de temps comme on peut le constater aussi dans le graphe de la figure III .19 .

Durée de simulation	S/système1	S/ système2	S/ système3	S/système4	Système
10000	8.347	1.558	1.451	3.375	17.560
20000	9.256	1.705	1.521	3.328	16.079
30000	9.088	1.703	1.493	3.334	15.059
40000	9.254	1.725	1.493	3.385	14.869
50000	9.047	1.747	1.490	3.418	14.449
60000	9.185	1.758	1.492	3.371	14.377

Tableau III.1 : Temps d'attente moyen dans chacun des sous système et dans le temps de séjour dans le système pour des différentes durées de simulation choisies.

Les taux d'occupation obtenus sont comme suit :

Durée de simulation	S/ système1	S/ système2	S/ système3	S/ système4
10000	69	18	15	52
20000	76	20	17	57
30000	78	20	18	59
40000	79	20	18	60
50000	80	21	18	61
60000	81	21	18	61

Tableau III.2 : Le taux d'occupation en (%) des serveurs de chacun des sous systèmes pour les différentes durées de simulation choisies.

5. Analyse et interprétation des résultats

Etant donnée que les paramètres commencent à se stabiliser à partir d'une durée de 50000 unités de temps. Nous avons choisi d'étudier les différents cas considérés pour une période de 60000 unités de temps (la période de transition de 2000 unités non comprise).

Nous avons effectué plusieurs exécutions et nous avons calculé la moyenne des résultats obtenus pour produire le tableau suivant qui présente le temps moyen d'attente dans chaque sous système, le temps de séjour dans le système et le taux d'occupation de chaque serveur.

Pour le temps de séjour dans le système nous avons calculé le gain de chaque configuration par rapport à la configuration du cas 1 qu'on a pris comme référence par la formule suivante : $\text{gain}=\left(1-\frac{x}{y}\right)\times 100$ où y est la valeur de référence et x une variable qu'on veut calculer son gain par rapport a la référence

	Sous système1		Sous système2		Sous système3		Sous système4		Système	
	Temps attente	Taux occup (%)	Temps attente	Taux occup (%)	Temps attente	Taux occup (%)	Temps attente	Taux occup (%)	Temps séjour	Gain (%)
Cas 1	**9.490**	**80**	**4.939**	**69**			**3.322**	**60**	**18.740**	**---**
Cas 2	**9.558**	**80**	**2.162**	**36**	**1.782**	**30**	**3.343**	**60**	**15.724**	**16**
Cas 3	**9.966**	**80**					**3.287**	**60**	**14.035**	**25**
Cas 4	**9.643**	**80**	**2.146**	**34**			**3.330**	**60**	**14.785**	**21**
Cas 5	**9.475**	**80**	**1.759**	**20**	**1.533**	**18**	**3.363**	**61**	**14.732**	**21**

Tableau III.3 : Temps moyens et taux d'occupations pour une durée 60000.

On remarque que dans le premier cas le temps moyen de réponse dans le système est le plus élevé par rapport aux autres cas, et de même pour le taux d'occupation de chaque service surtout celui du disque dur qui est de 69%.

Pour le deuxième cas, on remarque que le temps de séjour dans le système a diminué par rapport au 1er cas, l'intégration du deuxième disque a eu une influence sur le taux d'occupation du 1er disque qui est passé de 69% à 36%; mais ceci n'a eu aucune influence sur le taux d'occupation du serveur1.

Dans le troisième cas, on a le plus petit temps de réponse par rapport au autres cas.

Pour le quatrième et le cinquième cas, on a une même interprétation sur les taux d'occupation que le 1er et le 2ème cas, mais l'intégration du cache a eu une influence sur le temps de séjour qui a diminué.

D'après ces résultats, on remarque que la présence du cache a une influence positive sur le temps de séjour dans le système, alors il est préférable d'augmenter le cache que d'ajouter des disques en terme de temps de séjour dans le système et en terme de dépenses.

Conclusion

Le but de notre travail est de modéliser, de simuler un serveur web et d'utiliser cette simulation pour étudier le comportement de ce serveur sous différentes configurations. Pour cela nous avons commencé par étudier les files d'attentes, les réseaux de files d'attentes et l'évaluation de leurs performances.

Le système considéré a été modélisé sous forme d'un système de files d'attentes à quatre services. Afin de simuler le fonctionnement de ce modèle, nous avons conçu et réalisé un simulateur en utilisant la méthode des trois phases et nous avons choisi comme langage de programmation le Pascal sous l'environnement Delphi.

Nous avons intégré dans le simulateur développé les différents paramètres de performances tels que le temps d'attente dans les sous systèmes, le temps de séjour dans le système, la longueur des files, le nombre de clients servis dans chaque sous système et dans tous le système, les taux d'occupation des serveurs etc.

D'autre part, nous avons utilisé le simulateur développé pour étudier différentes configurations du serveur web et nous sommes arrivé à la conclusion que l'utilisation d'un cache apporte toujours une amélioration de performance du serveur. Le cache peut bien sûr être combiné avec un disque dur au cas où l'extension du cache peut être limitée par des contraintes techniques ou économiques.

Le simulateur développé peut être utilisé pour simuler et analyser tous système qui peut être représenté par un système de files d'attentes analogue à celui qu'on a utilisé. Néanmoins, le travail est loin d'être parfait il peut être étendu de façon à ce qu'on puisse prendre en considération d'autres structures en permettant une définition dynamique du système à étudier.
Par ailleurs on peut prendre en considération aussi la priorité dans les files et la limitation des capacités des files qui restent un domaine à développer.

Annexe

Nous présentons dans cet annexe les résultats obtenues par le simulateur pour un disque, deux disques, un cache, un cache et un disque et un cache et deux disques.

Pour chacun des cas on a répété l'exécution quatre fois et les résultats écrits dans les tableaux se sont que des moyennes des résultats obtenus.

Pour tous les cas la simulation est faite par les paramètres suivants :

La durée de la simulation: 60000

Le début du calcul : 2000

Le nombre moyen de clients en unité de temps:

Nombre moyen d'arrivées : 5

Nombre moyen de clients servis par le serveur1: 6

Nombre moyen de clients servis par le serveur2: 7

Nombre moyen de clients servis par le serveur3: 8

Nombre moyen de clients servis par le serveur4: 8

Un disque sans cache

Les probabilités:

Le passage du serveur1 vers file2: 1

Le passage du serveur1 vers file3: 0

Le passage du serveur1 vers file4: 0

Résultats du sous système 1:

	Moyenne	Variance	Ecart type	Valeurs théoriques	Erreur d'estimation
Temps dans file1	**7.882**	**22.007**	**4.691**	**8.333**	**-0.451**
Nombre dans file1	**4.137**	**6.569**	**2.563**	**4.167**	**-0.03**
Temps dans sous_système1	**9.490**	**22.893**	**4.784**	**10.000**	**-0.54**
Nombre dans sous_système1	**4.980**	**6.925**	**2.631**	**5.000**	**-0.02**

Résultats du sous système 2:

	Moyenne	Variance	Ecart type	Valeurs théoriques	Erreur d'estimation
Temps dans file2	**3.547**	**5.692**	**2.386**	**8.751**	**-5.204**
Nombre dans file2	**1.928**	**1.993**	**1.411**	**5.143**	**-3.215**
Temps dans sous_système2	**4.939**	**6.576**	**2.564**	**10.000**	**-5.061**
Nombre dans sous_système2	**2.689**	**2.269**	**1.506**	**6.000**	**-3.311**

Résultats du sous système 4:

	Moyenne	Variance	Ecart type	Valeurs théoriques	Erreur d'estimation
Temps dans file4	**2.004**	**2.319**	**1.523**	**8.750**	**-6.746**
Nombre dans file4	**1.153**	**0.925**	**0.962**	**6.125**	**-4.972**
Temps dans sous_système4	**3.322**	**2.743**	**1.656**	**10.000**	**-6.678**
Nombre dans sous_système4	**1.845**	**1.155**	**1.075**	**7.000**	**-5.155**

Résultats des services :

	Temps moyen	Taux d'occupation (%)
Service 1	**1.607**	**80**
Service 2	**1.391**	**69**
Service 4	**1.211**	**60**

Résultats du Système :

	Moyenne	Variance	Ecart type
Temps dans le système	**18.740**	**29.726**	**5.452**
Nombre dans le système	**9.822**	**10.014**	**3.164**

Entrées	Sorties
30671	**30661**

Deux disques sans cache

Les probabilités:

Le passage du serveur1 vers file2: 0.5

Le passage du serveur1 vers file3: 0.5

Le passage du serveur1 vers file4: 0

Résultats du sous système 1:

	Moyenne	Variance	Ecart type	Valeurs théoriques	Erreur d'estimation
Temps dans file1	**7.95**	**23.491**	**4.846**	**8.333**	**-0.383**
Nombre dans file1	**4.154**	**6.792**	**2.606**	**4.167**	**-0.013**
Temps dans sous_système1	**9.558**	**24.432**	**4..943**	**10.000**	**-0.442**
Nombre dans sous_système1	**4.998**	**7.149**	**2.673**	**5.000**	**-0.002**

Résultats du sous système 2:

	Moyenne	Variance	Ecart type	Valeurs théoriques	Erreur d'estimation
Temps dans file2	**0.770**	**0.696**	**0.834**	**1.071**	**-0.301**
Nombre dans file2	**0.234**	**0.108**	**0.328**	**0.321**	**-0.087**
Temps dans sous_système2	**2.162**	**1.225**	**1.107**	**2.500**	**-0.338**
Nombre dans sous_système2	**0.658**	**0.237**	**0.486**	**0.750**	**-0.1**

Résultats du sous système 3 :

	Moyenne	Variance	Ecart type	Valeurs théoriques	Erreur d'estimation
Temps dans file3	**0.530**	**0.412**	**0.642**	**0.750**	**-0.22**
Nombre dans file3	**0.169**	**0.074**	**0.272**	**0.225**	**-0.056**
Temps dans sous_système3	**1.782**	**0.799**	**0.894**	**2.000**	**-0.218**
Nombre dans sous_système3	**0.550**	**0.185**	**0.430**	**0.600**	**-0.05**

Résultats du sous système 4:

	Moyenne	Variance	Ecart type	Valeurs théoriques	Erreur d'estimation
Temps dans file4	**2.021**	**2.383**	**1.544**	**18.750**	**-16.729**
Nombre dans file4	**1.156**	**0.927**	**0.963**	**14.063**	**-14.907**
Temps dans sous_système4	**3.343**	**2.822**	**1.679**	**20.000**	**-16.657**
Nombre dans sous_système4	**1.851**	**1.157**	**1.075**	**15.000**	**-13.149**

Résultats des services :

	Temps moyen	Taux d'occupation (%)
Service 1	**1.608**	**80**
Service 2	**1.392**	**36**
Service 3	**1.194**	**30**
Service 4	**1.212**	**60**

Résultats du Système :

	Moyenne	Variance	Ecart type
Temps dans le système	**15.724**	**26.946**	**5.190**
Nombre dans le système	**8.324**	**8.505**	**2.916**

Entrées	Sorties
30962	**30956**

Un cache sans aucun disque

Les probabilités:

Le passage du serveur1 vers file2: 0

Le passage du serveur1 vers file3: 0

Le passage du serveur1 vers file4: 1

Sous système 1:

	Moyenne	Variance	Ecart type	Valeurs théoriques	Erreur d'estimation
Temps dans file1	**8.351**	**25.248**	**5.025**	**8.333**	**0.018**
Nombre dans file1	**4.471**	**7.583**	**2.754**	**4.167**	**0.304**
Temps dans sous_système1	**9.966**	**26.157**	**5.114**	**10.000**	**-0.034**
Nombre dans sous_système1	**5.340**	**7.919**	**2.814**	**5.000**	**0.340**

Sous système 4:

	Moyenne	Variance	Ecart type	Valeurs théoriques	Erreur d'estimation
Temps dans file4	**1.978**	**2.270**	**1.506**	**3.750**	**-1.695**
Nombre dans file4	**1.197**	**0.945**	**0.972**	**2.250**	**-1.053**
Temps dans sous_système4	**3.287**	**2.685**	**1.638**	**5.000**	**-1.713**
Nombre dans sous_système4	**1.921**	**1.160**	**1.077**	**3.000**	**-1.079**

Services :

	Temps moyen	Taux d'occupation (%)
Service 1	**1.614**	**80**
Service 4	**1.202**	**60**

Système :

	Moyenne	Variance	Ecart type	Entrées	Sorties
Temps dans le système	**14.035**	**27.823**	**5.274**	**31056**	**31048**
Nombre dans le système	**7.499**	**8.948**	**2.991**		

Un cache et un disque

Les probabilités:

Le passage du serveur1 vers file2: 0.5

Le passage du serveur1 vers file3: 0

Le passage du serveur1 vers file4: 0.5

Sous système 1:

	Moyenne	Variance	Ecart type	Valeurs théoriques	Erreur d'estimation
Temps dans file1	**8.029**	**23.199**	**4.816**	**8.333**	**-0.304**
Nombre dans file1	**4.242**	**6.834**	**2.614**	**4.167**	**0.075**
Temps dans sous_système1	**9.643**	**24.130**	**4.912**	**10.000**	**-0.357**
Nombre dans sous_système1	**5.095**	**7.176**	**2.678**	**5.000**	**0.095**

Sous système 2:

	Moyenne	Variance	Ecart type	Valeurs théoriques	Erreur d'estimation
Temps dans file2	**0.767**	**0.719**	**0.848**	**1.071**	**-0.304**
Nombre dans file2	**0.240**	**6.834**	**2.614**	**0.321**	**-0.080**
Temps dans sous_système2	**2.146**	**1.251**	**1.118**	**2.500**	**-0.353**
Nombre dans sous_système2	**0.673**	**0.243**	**0.493**	**0.750**	**-0.076**

Sous système 4:

	Moyenne	Variance	Ecart type	Valeurs théoriques	Erreur d'estimation
Temps dans file4	**1.969**	**1.221**	**1.105**	**5.417**	**-3.447**
Nombre dans file4	**1.151**	**0.876**	**0.936**	**3.521**	**-2.369**
Temps dans sous_système4	**3.330**	**2.616**	**1.617**	**6.667**	**-3.336**
Nombre dans sous_système4	**1.860**	**1.096**	**1.047**	**4.333**	**-2.472**

Services :

	Temps moyen	Taux d'occupation (%)
Service 1	**1.613**	**80**
Service 2	**1.379**	**34**
Service 4	**1.212**	**60**

Système :

	Moyenne	Variance	Ecart type
Temps dans le système	**14.785**	**26.617**	**5.159**
Nombre dans le système	**7.859**	**8.372**	**2.893**

Entrées	Sorties
30970	**30963**

Un cache et deux disques

Les probabilités:

0Le passage du serveur1 vers file2: 0.3

Le passage du serveur1 vers file3: 0.3

Le passage du serveur1 vers file4: 0.4

Sous système 1:

	Moyenne	Variance	Ecart type	Valeurs théoriques	Erreur d'estimation
Temps dans file1	**7.870**	**23.183**	**4.815**	**8.333**	**-0.463**
Nombre dans file1	**4.160**	**0.702**	**0.838**	**4.167**	**-0.007**
Temps dans sous_système1	**9.475**	**24.099**	**4.909**	**10.000**	**-0.525**
Nombre dans sous_système1	**5.008**	**7.299**	**2.701**	**5.00**	**0.008**

Sous système 2:

	Moyenne	Variance	Ecart type	Valeurs théoriques	Erreur d'estimation
Temps dans file2	**0.379**	**0.331**	**0.575**	**0.495**	**-0.116**
Nombre dans file2	**0.072**	**0.027**	**0.165**	**0.089**	**-0.017**
Temps dans sous_système2	**1.759**	**0.862**	**0.928**	**1.923**	**-0.164**
Nombre dans sous_système2	**0.340**	**0.103**	**0.321**	**0.346**	**-0.006**

Sous système 3 :

	Moyenne	Variance	Ecart type	Valeurs théoriques	Erreur d'estimation
Temps dans file3	**0.270**	**0.200**	**0.447**	**0.363**	**-0.093**
Nombre dans file3	**0.059**	**0.022**	**0.148**	**0.065**	**-0.006**
Temps dans sous_système3	**1.533**	**0.583**	**0.764**	**1.613**	**-0.080**
Nombre dans sous_système3	**0.307**	**0.091**	**0.302**	**0.290**	**0.017**

Sous système 4:

	Moyenne	Variance	Ecart type	Valeurs théoriques	Erreur d'estimation
Temps dans file4	**2.038**	**2.409**	**1.552**	**7.841**	**-5.803**
Nombre dans file4	**1.188**	**0.966**	**0.983**	**5.410**	**-4.222**
Temps dans sous_système4	**3.363**	**2.840**	**1.685**	**9.091**	**-5.728**
Nombre dans sous_système4	**1.896**	**1.192**	**1.092**	**6.273**	**-4.377**

Services :

	Temps moyen	Taux d'occupation (%)
Service 1	**1.604**	**80**
Service 2	**1.379**	**20**
Service 3	**1.197**	**18**
Service 4	**1.215**	**61**

Système :

	Moyenne	Variance	Ecart type	Entrées	Sorties
Temps dans le système	**14.732**	**27.739**	**5.266**	**31009**	**31004**
Nombre dans le système	**7.859**	**8.869**	**2.978**		

Références Bibliographiques

[1]. D.AÏSSANI : Support de cours " *processus aléatoires appliqués* ", 4ème Année Département de Recherche Opérationnelle, Université Abderrahman Mira de Bejaia, 2000/2001.

[2]. G.CULLMANN : " R*echerche opérationnelle* ", Masson, Paris, 1970.

[3]. M. BABES : " S*tatistique, file d'attente et simulation* " , OPU,Alger, 1992.

[4]. P.QUITTARD : " *Eléments de statistiques, Processus stochastique et files d'attente* ", Centre d'études et de recherche en informatique, OPU, Alger, 1983.

[5]. R.FAURE : " *Précis de recherche opérationnelle* ", Dunod, Paris, 1979.

[6]. Y.OUINTEN : Support de cours " *Modélisation et Simulation*",4ème Année Département de Génie informatique, Université Amar Telidji de Laghouat, 2004/2005.

Sites Internet :

[S1]. G.HEBUTERNE, T.ATMACA, la Salle de Cours Virtuelle, Octobre 1998/1999 http://www-rst.int-evry.fr/~hebutern/IT21/Rappels/C2.html.

[S2]. http://membres.lycos.fr/dthiery/filedatt.htm

[S3]. Support de cours, Octobre 1998 http://evalperf.essi.fr/cours/introduction/intro.html

[S4]. La simulation par évènement, http://www-rst.int-evry.fr/~hebutern/IT21/Simu/Coursimu.html

[S5]. J.F.HECHE : " *Recherche Opérationnelle les files d'attentes et les réseaux de files d'attente* ", Institut de Mathématiques, Ecole Polytechnique Fédérale de Lausanne (EPFL), Mars 2004. http://roso.epfl.ch/cours/ro/2003-2004/cours/files_attente-4up.pdf

[S6]. Le Principe fondamental des méthodes de simulation,
http://www.rogp.eplf.hec.ulg.ac.be/Crama/Teaching/RO1lic/Docs/Chap3_simul.pdf

[S7]. A.BOUNCEUR, M.l. Mamasse : " Gestion Optimale des Silos à Céréales de l'Entreprise CEVITAL " , mémoire de fin d'étude d'ingénieur en Recherche Opérationnelle, Département de Recherche Opérationnelle et Informatique, Université Abderrahmane Mira de Béjaia 2002 .
http://tima.imag.fr/rms/perso/bounceur/doc/rapport_ing.pdf