Architecture et étude des performances d'un serveur multimédia pour réseaux à haut débit

Hichem KADDECHE et Monique BECKER

Département informatique, Institut National des Télécommunications
9, rue Charles Fourier, 91011 Evry Cedex-FRANCE
et
Laboratoire MASI
4 place jussieu, 75252 Paris Cedex - FRANCE

Tél: (33 1) 60 76 47 42, Fax: (33 1) 60 76 47 80
E-mail: kaddeche@etna.int-evry.fr et mbecker@etna.int-evry.fr

Résumé

Ce travail rentre dans le cadre d'un contrat de recherche entre le CCETT (Centre Commun d'Études de Télédiffusion et Télécommunication) et l'INT (Institut National des Télécommunications). Le travail consiste en la présentation de l'architecture et l'étude de performances d'un serveur multimédia capable de gérer plusieurs utilisateurs connectés simultanément à distance à travers un réseau à haut débit. Les applications sont des séquences vidéo. L'architecture et le principe de fonctionnement du système ont été développés en collaboration par le CCETT et l'INT. Le serveur a été modélisé par un réseau de files d'attente. Les différentes simulations effectuées nous ont permis d'étudier les performances du système, d'étudier les avantages et les inconvénients de l'architecture adoptée et d'analyser l'effet des différents paramètres du système afin d'optimiser son fonctionnement.

Mots clés: serveur multimédia, séquences vidéo, architecture parallèle, modélisation, évaluation de performances, simulation.

Abstract

The work presented here is a joint work between CCETT (Centre Commun d'Etudes de Télédiffusion et de Télécommunications) and INT (Institut National des Télécommunications). It consists in the design and the performance analysis of a multimedia server, a device able to offer audio-visual sequences to several users simultaneously, through a high speed network. The architecture and the operating principles were developped by CCETT and INT. A queueing network model has been developed and simulated. From many simulations the study of the behaviour of the system and the optimisation of its implementation were derived.

Key words: multimedia server, audio-visual sequences, parallel architectures, modelling, performance evaluation, simulation.

Table des matières

1. Introduction

Le domaine du multimédia est en pleine expansion grâce au développement matériel et logiciel informatique. La perspective des réseaux grande distance à haut débit, les travaux menés actuellement dans le domaine de la compression des données favoriseront le développement d'applications multimédia mettant en jeu des transmissions numériques (texte, vidéo et son). Les applications actuelles sont en général limitées à des réseaux locaux voir des stations isolées. L'objectif est de proposer des services multimédia de qualité, sur des grandes distances et pour plusieurs utilisateurs simultanés. Ceci sera réalisé grâce à des serveurs multimédia qui seront capables de traiter plusieurs utilisateurs simultanément et grâce à un réseau de transmission à haut débit qui permettra un cheminement rapide des flux d'information entre utilisateurs et serveur. Les applications futures seront nombreuses et variées: la vidéo à la demande, les informations à la demande, les services de consultation multimédia ...

Dans cette étude nous nous intéressons à un serveur numérique de séquences vidéo. Les séquences sont compressées suivant le format MPEG1 (Motion Picture Group) à débit constant et sauvegardées sous forme de fichiers binaires [5]. L'importante quantité de données manipulée et les haut débits mis en jeux font que l'architecture parallèle convient très bien à ce système [1]. Le système sera composé de plusieurs disques et chaque fichier de séquence vidéo sera fragmenté en blocs de même taille répartis sur les différents disques [4] et [6].. Cette configuration permet de repartir l'opération d'extraction d'un fichier sur les différents disques et d'avoir ainsi un débit global d'extraction beaucoup plus élevé que celui d'un seul disque. Un autre avantage de la fragmentation des fichiers sur les différents disques est la possibilité d'extraction presque simultanée d'un même fichier pour plusieurs utilisateurs qui demandent la même séquence. Cette configuration sert donc aussi à favoriser l'aspect multi-utilisateurs. Le serveur sera à base de disques et transputers interconnectés.

Nous nous intéresserons dans cette étude principalement à deux critères: le temps de réponse du serveur et la charge maximale qu'il peut supporter. Le temps de réponse du serveur est le temps écoulé entre la demande de visualisation d'une séquence et le début de son émission. La charge maximale du serveur est le nombre maximum de connexions (utilisateurs) simultanées que peut traiter le serveur avec zéro rupture. Nous avons une rupture dans l'émission d'une séquence lorsqu'à un moment le serveur n'arrive plus à assurer l'émission du restant des blocs de la séquence considérée au débit MPEG requis (1,5 Mbits/s). Le fonctionnement du serveur ainsi présenté dépend de plusieurs paramètres.

Le but recherché est d'avoir un comportement stable du serveur vis-à-vis des facteurs extérieurs, d'avoir une charge maximale optimale et des temps de réponse acceptables. Nous nous intéresserons donc dans cette étude aux différents paramètres du système, à l'évolution de ses performances en fonction du nombre de disques et à son comportement vis-à-vis des facteurs extérieurs (applications et utilisateurs). La modélisation du système sera faite par un réseau de files d'attente et la résolution sera effectuée par des simulations. Notons que cette étude de performances a été motivée par une étude antérieure, effectuée sur une architecture voisine [2].

Le rapport se présente comme suit: une présentation du système et de son principe de fonctionnement, et une présentation de sa modélisation en files d'attente. Une première étude préliminaire sur un serveur de huit disques pour avoir une idée sur le comportement global de l'architecture et le choix des critères de performances. Nous passerons par la suite à l'étude des paramètres intrinsèques de pilotage du système, à savoir: la taille des blocs, le nombre de blocs anticipés et la politique de placement des blocs. Nous étudierons par la suite l'évolution des performances en fonction du nombre de disques du serveur. Nous passerons enfin à l'étude du comportement de l'architecture vis-à-vis des facteurs extérieurs, à savoir: le profil des séquences et leur nombre, et le profil des consultations. Nous terminerons par une conclusion et la présentation des points éventuels de développements futurs.

2. Présentation du système et de son principe de fonctionnement

Le serveur est constitué de disques et de transputers interconnectés. Les séquences vidéo sont stockées sous forme de fichiers binaires MPEG1. Chaque fichier (représentant une séquence) est découpé en plusieurs blocs de même taille qu'on répartit sur les différents disques suivant une politique de placement donnée.

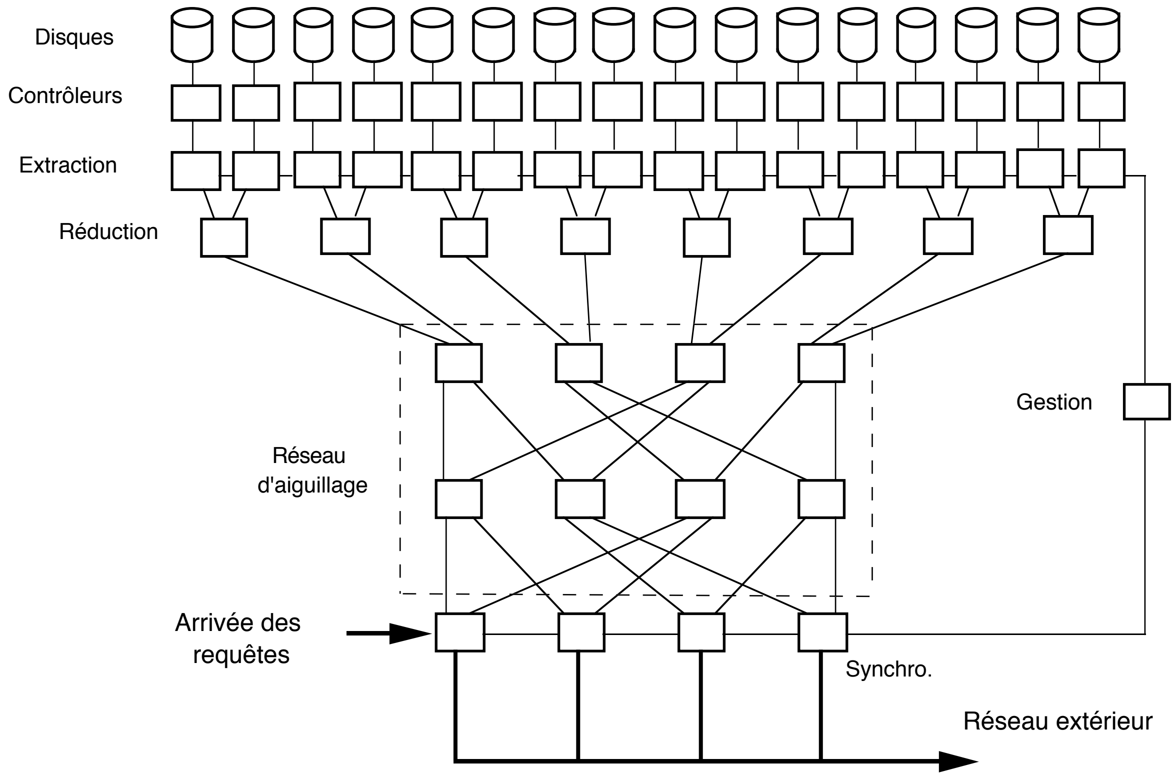

Figure 1. Serveur à 16 disques

Chaque disque est relié à un transputer contrôleur qui a pour fonction de contrôler et de gérer l'opération d'extraction des blocs au niveau des disques. Les transputers d'extraction exécutent les ordres d'extraction provenants du module de gestion et retransmettent à leurs voisins (de gauche) les ordres qui ne leur sont pas destinés. Les blocs qui leur sont ensuite envoyés par les contrôleurs sont transmis aux transputers de réduction. Les transputers de réduction ont pour rôle de réduire le nombre d'entrées du réseau d'aiguillage et ainsi sa taille. Les transputers d'aiguillage ont pour rôle de faire parvenir tout bloc extrait de n'importe quel disque au transputer de synchronisation demandeur. Les transputers de synchronisation ont la charge de gérer l'extraction des blocs et de gérer leurs transmission isochrone dans le bon ordre et au débit MPEG sur le réseau extérieur. Le transputer de gestion a pour rôle de transmettre vers les transputers d'extraction adéquats les ordres d'extraction provenants des transputers de synchronisation.

Le principe de fonctionnement est le suivant: à son arrivée une requête utilisateur est transmise au transputer de synchronisation le moins chargé. Le transputer réserve un espace mémoire pour stocker et gérer les blocs de la séquence demandée. Toutes les opérations concernant cette demande de séquence seront gérées à partir de ce transputer et de cet espace réservé qui sera modélisé par la suite par une file d'attente qu'on appellera file de synchronisation. Les A premiers blocs de la séquence considérée sont alors demandés simultanément. Cette opération est effectuée par l'envoi simultané de A ordres d'extraction au transputer de gestion. Le nombre A d'anticipation est un paramètre constant du système qu'il faudra fixer au préalable. Lorsque les A premiers blocs sont arrivés dans le transputer de synchronisation, on peut démarrer l'émission isochrone des blocs sur le réseau extérieur. A chaque émission d'un bloc, le transputer de synchronisation demande au module de gestion l'extraction du bloc suivant. Cette opération est répétée jusqu'à l'extraction de la totalité des blocs. L'émission isochrone des blocs est réalisée de sorte à assurer le débit MPEG (1,5 Mbits/s). Cette émission s'arrête avec la fin de la séquence ou avec une rupture dans l'émission. Nous avons une rupture d'une émission de séquence lorsque nous n'arrivons pas à émettre le bloc requis à temps pour respecter le débit MPEG.

L'aiguillage, depuis les disques vers les transputers de synchronisation, des blocs extraits est assuré par un réseau de transputers connectés suivant la topologie oméga. Cette topologie est caractérisée par des règles de connexion entre les transputers des différents étages du réseau et des règles de routage des blocs qui y circulent (voir figure 1). C'est une architecture monochemin. C'est à dire que d'une entrée il y a un seul chemin qui mène vers une sortie donnée. Le nombre d'entrées est égal au nombre de sorties. Ce nombre est une puissance de deux. Pour un réseau à 2^N entrées nous avons N étages de transputers. Notons cependant que pour notre serveur nous n'avons pas besoin du dernier étage du réseau puisque les deux sorties des différents transputers de cet étage vont vers le même transputer de synchronisation donc sont équivalentes. Nous avons ainsi pour un réseau à 2^N entrées N -1 étages de transputers. Notons que le choix de cette architecture n'est pas définitif. Il est intéressant d'étudier d'autre solutions d'aiguillage tel que l'architecture banyan ou un aiguillage en crossbar etc. ...

Cette présentation nous permet de relever les paramètres intrinsèques agissant sur le fonctionnement du système et qu'on développera dans la suite à savoir: la taille des blocs, le nombre de blocs anticipés et la politique de placement de ces blocs sur les disques.

3. Présentation du modèle

Le serveur sera modélisé par un réseau de files d'attente. Sachant que les temps de traitements aux niveaux des transputers sont négligeables par rapport aux temps de transmissions aux niveaux des liens inter-transputers, les entités réelles modélisées par les files d'attente ne seront pas les transputers mais les liens inter-transputers. Les disques auront une représentation en files d'attente pour modéliser le temps de lecture des blocs et de leur transfert dans les contrôleurs. Pour simplifier l'écriture du modèle ainsi que son exploitation, on va réserver statiquement un nombre fixe surdimensionné de files de synchronisation pour chaque transputer de synchronisation au lieu de les créer dynamiquement.

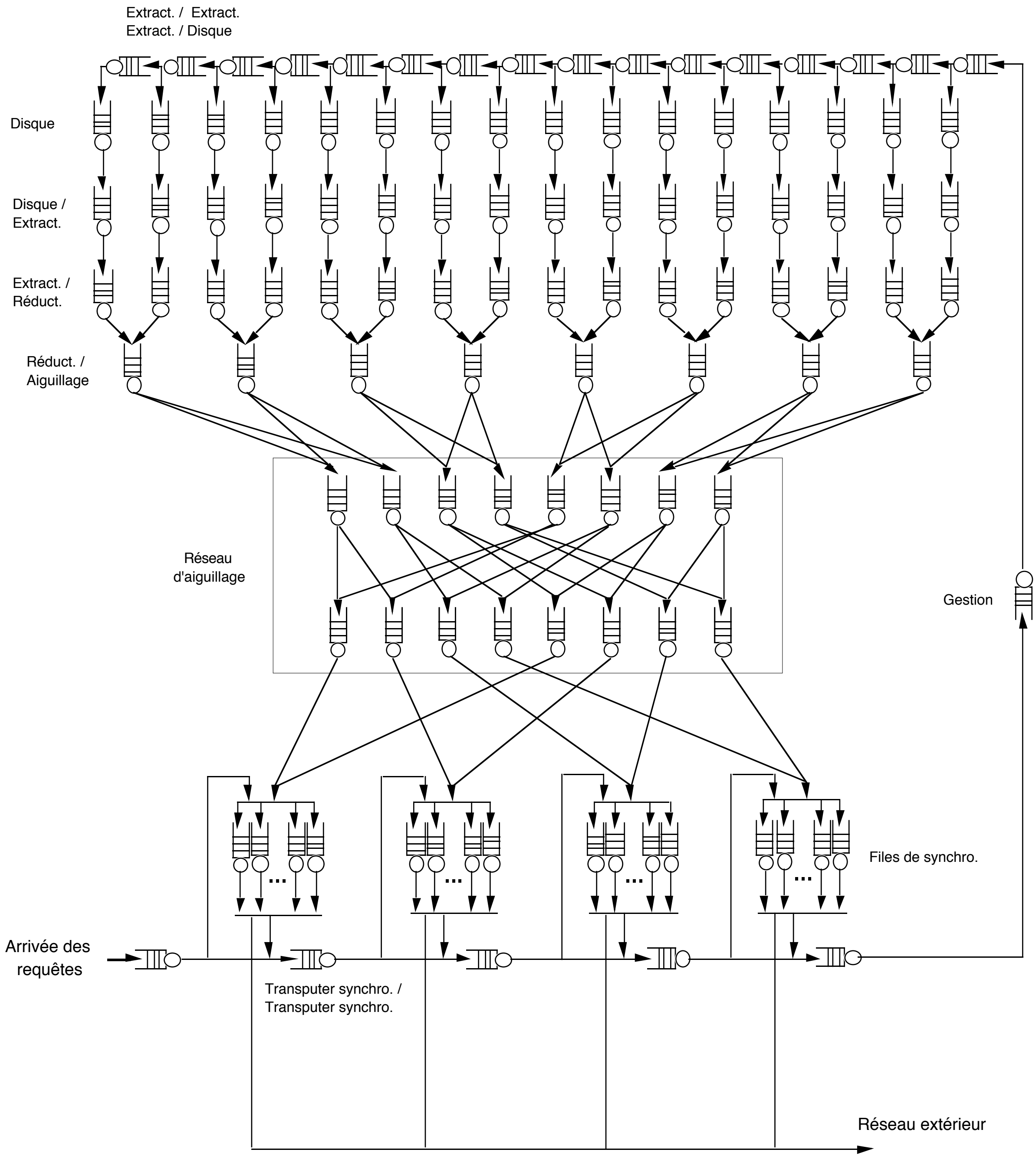

Figure 2. Modèle du serveur à 16 disques

Les données techniques exprimant respectivement le temps d'extraction d'un bloc dans un disque et le temps de transfert sur un lien de transputer sont les suivantes:

$TpsD$ (ms) $= 0{,}68.10^{-3} \times tb + \theta$. (1)

$TpsL$ (ms) $= 0{,}8.10^{-3} \times tb + 0{,}33$. (2)

TpsD représente le temps moyen nécessaire pour l'extraction d'un bloc de taille tb à travers les contrôleurs de disques. TpsL représente le temps moyen de transfert d'un bloc de taille tb sur un lien de transputer. tb représente la taille des blocs exprimée en octets. θ représente le temps de latence et de positionnement de la tête de lecture des disques qui suit une loi uniforme dans l'intervalle [4 ; 42,7] (ms). Le temps d'extraction d'un bloc dans un disque est composé du cumul d'un temps de latence et de positionnement de la tête de lecture (θ) et d'un temps de lecture. Le temps de transfert à travers un lien de transputer est composé d'un temps de latence (0,33 ms) et d'un temps de transfert de données. Pour une taille de bloc de 30 Ko nous obtenons un débit d'extraction moyen de 5 Mb/s pour les disques et leurs contrôleurs et un débit moyen de transfert de 10 Mb/s pour les liens de transputers (équ. 3 et 4). Ces données ont été déterminées grâce à des mesures effectuées au CCETT sur les composants retenus actuellement: des contrôleurs de disques T222 et des transputers T425.

Une résolution analytique du modèle n'est pas possible du fait de la présence de mécanismes de synchronisation, de temps de service constants et de création et suppression de clients. On a donc eu recours à la simulation à événements discrets. La résolution est effectuée en réseau fermé (figure 3). Le réseau comprend un nombre donné de clients qui représente le nombre de connexions (requêtes, utilisateurs) simultanées ou encore la charge du serveur. A la fin de l'émission d'une séquence une nouvelle requête est crée. Une requête est une demande de visualisation d'une séquence donnée. Ceci nous permet d'avoir un nombre de connexions constant dans le système, le long d'une simulation. Les simulations ont étés réalisées avec le simulateur QNAP2 (Queuing Network Analysis Package 2nd generation) [7]. Un modèle général paramètré a été développé afin de nous permettre d'étudier l'effet des différents paramètres et facteurs qui nous intéressent.

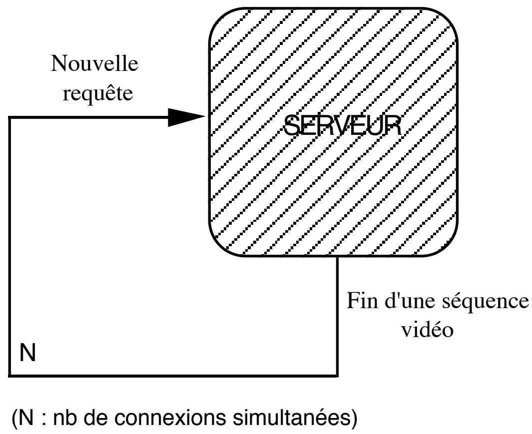

Figure 3. Résolution en boucle fermée

4. Étude préliminaire

Le but de cette étude préliminaire est d'une part de nous familiariser avec le comportement global du serveur et d'autre part de voir de près l'évolution des performances du système et de justifier ainsi les critères choisis.

4.1. Étude du temps de réponse

Nous allons dans cette partie nous intéresser à deux points: l'évolution du temps de réponse en fonction de la charge du système et les variations du temps de réponse pour une même charge donnée.

Le temps de réponse du système est croissant avec la charge à laquelle est soumis ce dernier (figure 4). Ceci s'explique par une plus forte sollicitation des disques et des liens des différents composants et ainsi une augmentation de leurs taux d'occupation et de leurs temps de réponse lorsque la charge augmente. Nous constatons sur la figure 4 et sur d'autres simulations effectuées sur des serveurs de plus grande taille que la dégradation du temps de réponse moyen avec l'augmentation de la charge est plutôt continue. Nous constatons aussi dans certains cas, une dégradation accentuée du temps de réponse juste au voisinage très proche du début d'apparition de ruptures d'émissions de séquences. Ce comportement fait du critère rupture d'émission le principal critère limitatif du nombre de connexions simultanées.

Nous avons réalisé dans toute cette étude près de 1000 simulations, soit près de 300 heures de fonctionnement des différents serveurs étudiés confondus, dans différentes conditions et avec différents paramètres. Pour toutes ces simulations le temps de réponse maximal a toujours été borné par: 2,25 x temps de réponse moyen.

Les trois figures qui suivent présentent les résultats de simulations réalisées sur un serveur à 8 disques avec 400 séquences uniformément réparties entre 10 et 30 secondes, des blocs de 30 Ko de taille, un nombre d'anticipation égal à 10 blocs et une consultation uniforme des différentes séquences (toutes les séquences ont la même probabilité d'être consultées).

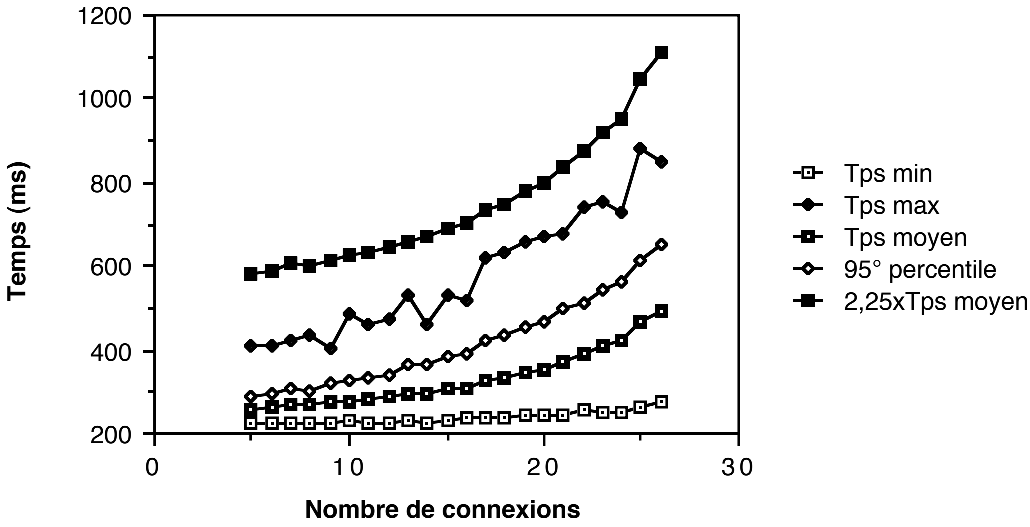

Figure 4. Temps de réponse fonction de la charge

La comparaison des courbes: temps de réponse moyen, temps de réponse maximum, temps de réponse minimum et le 95° percentille du temps de réponse, montre que le temps de réponse est bien concentré autour du temps de réponse moyen avec une dégradation du coefficient de variation (écart type/moyenne) avec l'augmentation de la charge qui reste cependant inférieur à 0,2 (figure 5).

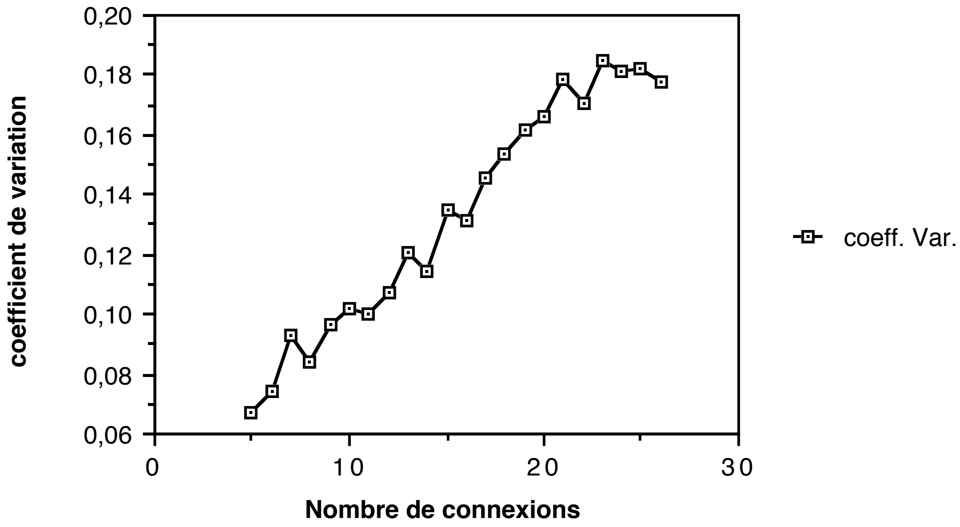

Figure 5. Coefficient de variation du temps de réponse en fonction de la charge

Pour mieux voir les variations du temps de réponse pour une charge donnée nous avons dressé l'histogramme de répartition du temps de réponse pour le serveur précédent pour une charge de 20 utilisateurs simultanés. Cet histogramme a été dressé pour un fonctionnement de 30 minutes soit un nombre de 1700 séquences émises.

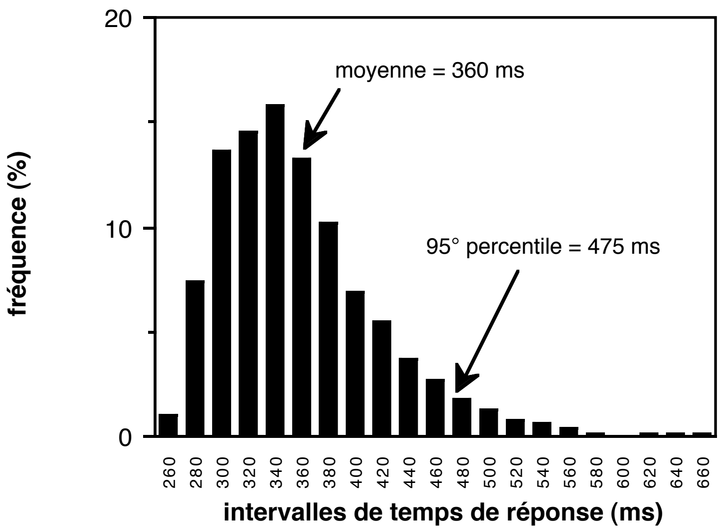

Figure 6. Répartition du temps de réponse

Nous remarquons sur la figure 6 la forte concentration du temps de réponse autour de la moyenne et en particulier en dessous de celle-ci. 60% de la population est inférieure à la moyenne et 95% inférieure à 0,7xtemps de réponse maximum.

Suite à cette étude nous pouvons dire que le temps de réponse moyen est un assez bon indicateur du comportement global du temps de réponse du système. Dans la suite nous retiendrons ce critère pour l'étude du temps de réponse du système.

4.2. Intervalle de confiance

Pour valider les résultats relatifs aux temps de réponse, un calcul d'intervalles de confiance a été effectué. Les durées des simulations sont fixées de sorte à avoir un intervalle de confiance (à 95%) de rayon inférieur à 5% du temps de réponse moyen (sans hypothèse de normalité).

4.3. Étude de la saturation du système

Les différentes simulations effectuées lors de cette étude montrent que la saturation du système évolue très rapidement lorsque la charge augmente. Le passage d'un fonctionnement à 0 rupture à celui à 100% de ruptures se fait au plus avec une augmentation de 5% de la charge au début de l'apparition de ruptures. Cela correspond à 2 connexions supplémentaires pour un serveur à 16 disques et 20 pour un serveur à 128 disques. La dégradation rapide du taux de rupture est accompagnée par une forte dégradation du temps de réponse. Il est donc non intéressant d'étudier le comportement du serveur dans ces conditions de fonctionnement. Dans la suite nous nous intéresserons au fonctionnement du système avec 0 rupture d'émission. La charge maximale du système sera le nombre maximum de connexions que pourra supporter le système avec 0 rupture d'émission. Le point de saturation sera le point de fonctionnement correspondant à cette charge maximale.

4.4. Répartition de la charge dans le système

Avant de passer à l'étude détaillée du comportement du système vis-à-vis des paramètres internes et des facteurs externes, nous donnons un bref aperçu sur la répartition de la charge à l'intérieur du système et la localisation des goulots d'étranglement. L'analyse des différents critères de performances (taux d'occupation, temps de réponse, débit à la sortie, ...) des différentes files du modèle pour les différentes simulations réalisées, montre une bonne répartition de la charge sur les différents composants équivalents dans le système. Les différents disques présentent pratiquement la même charge. Nous avons la même chose pour les différents liens équivalents d'un même étage du serveur. Cette constatation est une preuve du bon fonctionnement parallèle du système.

Une analyse du réseau d'aiguillage permet de relever des temps de réponse plus élevés pour les premières couches d'aiguillage. Ceci s'expliquerait par des lois d'arrivée des blocs de plus en plus régulières pour les couches inférieures d'aiguillage. En effet plus nous descendons dans le réseaux d'aiguillage mieux sont séparés les flux de blocs correspondants aux différentes séquences et plus régulière est donc l'arrivée des blocs. Sur un lien de la première couche transitent les blocs de n'importe quelles séquences alors que sur un lien de la dernière couche ne transitent que les blocs des séquences gérées par le transputer de synchronisation auquel est connecté ce lien.

La comparaison des taux d'occupation des différentes files du modèle, pour les différentes simulations effectuées avec les différents paramètres, nous a permis de localiser le goulot d'étranglement du système qui se situe soit au niveau des disques soit au niveau des liens de réduction et d'aiguillage suivant la taille des blocs.

5. Étude des paramètres de pilotage

Étant donné un serveur de dimension (nombre de disques) fixé, trois paramètres sont alors à fixer pour permettre son fonctionnement: la taille des blocs, le nombre d'anticipation et la stratégie de placement des blocs sur les disques. Ce sont les paramètres de pilotage du système qu'il faudra choisir de façon à optimiser le fonctionnement de ce dernier. Dans la suite de ce paragraphe, nous allons étudier l'effet de ces différents paramètres et essayer de les optimiser.

5.1. Étude de la taille des blocs

Pour étudier l'effet de la taille des blocs nous avons considéré un serveur à 8 disques avec 400 séquences de durées uniformément réparties entre 10 et 30 secondes, un nombre de blocs anticipés égal à 10 et une consultation uniforme des séquences. Pour une taille donnée des blocs nous faisons augmenter la charge et nous relevons le temps de réponse moyen correspondant jusqu'à saturation du système.

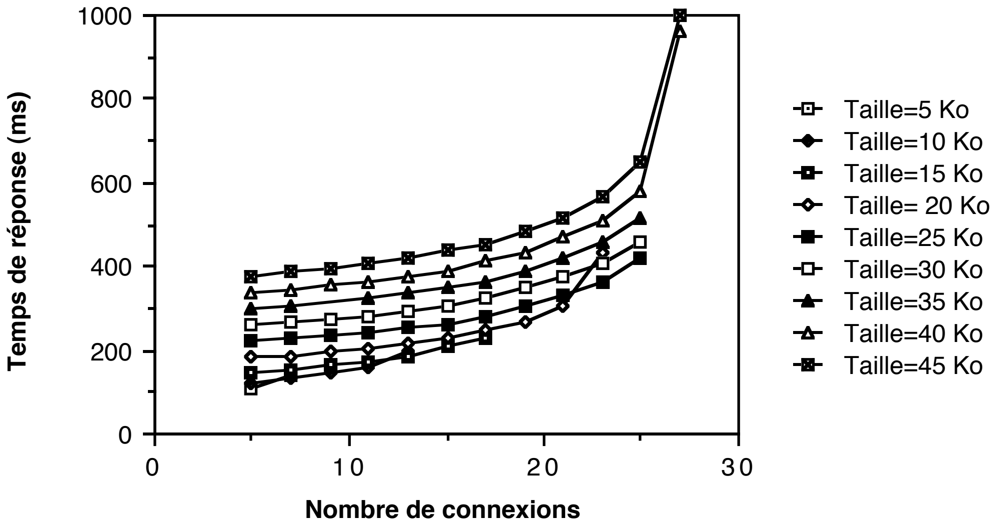

Figure 7. Effet de la taille des bloc sur le temps de réponse

Nous constatons (figure 7) une augmentation du temps de réponse moyen avec la taille des blocs. En effet en augmentant la taille des blocs nous augmentons la quantité de données collectée dans les transputers de synchronisation avant le début de l'émission d'une séquence. Le temps de réponse qui n'est autre que le temps de collecte de ces blocs d'anticipation se trouve alors augmenté. Cette augmentation du temps de réponse s'explique aussi, mais en moindre partie, par une plus importante sollicitation des disques et des liens de transmission qui est due à l'augmentation de la quantité de données requise simultanément lors des phases d'anticipation. La collecte des blocs d'anticipation entraîne une plus forte charge au niveau des composants du système vu quelle se fait d'une façon simultanée. En effet les ordres d'extraction des blocs d'anticipation sont envoyés tous simultanément contrairement aux ordres d'extraction des autres blocs restants qui sont envoyés d'une façon espacée à la suite de l'émission de chaque bloc et donc avec une période qui correspond à la période d'inter-emissions. Travailler avec une grande taille de bloc entraîne une charge plus importante au niveau du système qui dégrade ainsi le temps de réponse. Cette dégradation au niveau du temps de réponse est encore plus sensible à l'approche de la saturation où le système est fortement sollicité. Cette interprétation est faite sur la base d'une comparaison des temps de réponse et des taux d'occupation des files goulots d'étranglement pour les différentes simulations effectuées. Ce comportement est encore plus visible avec l'augmentation du nombre d'anticipation pour une taille de bloc constante (paragraphe suivant).

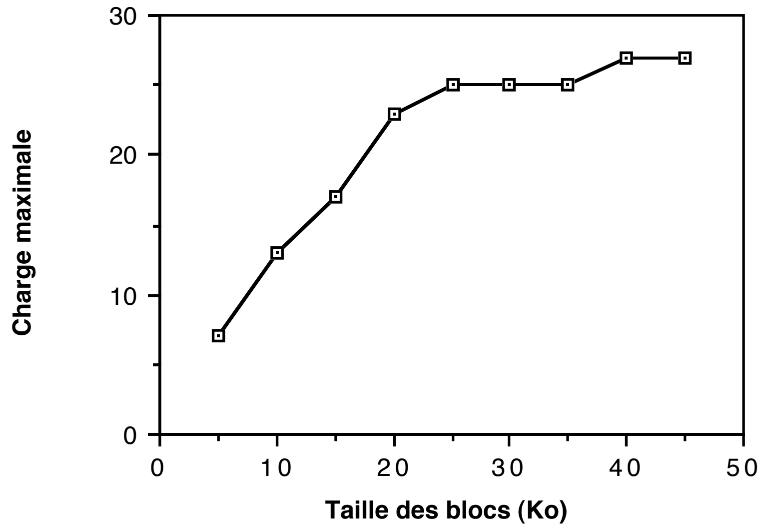

Figure 8. Effet de la taille des bloc sur la charge maximale

Nous constatons aussi (figure 8) l'augmentation de la charge maximale supportée par le serveur avec l'augmentation de la taille des blocs. Cette augmentation peut s'expliquer par les trois faits suivants qui accompagnent l'augmentation des tailles des blocs. Le premier fait est l'augmentation des débits moyens des disques (DmD) et des liens (DmL), en effet lorsque la taille des blocs augmente le temps de latence et de positionnement des têtes des disques ainsi que le temps de latence des liens deviennent négligeables devant les temps de transferts effectifs des données d'où l'augmentation des débits moyens. Ces débits moyens sont le quotient de la taille des blocs par le temps d'extraction des disques respectivement le temps de transmission des liens:

$$DmD\ (\text{oct./ms}) = tb \,/\, (0{,}68.10^{-3} \times tb + \theta). \quad (3)$$

$$DmL\ (\text{oct./ms}) = tb \,/\, (0{,}8.10^{-3} \times tb + 0{,}33). \quad (4)$$

Le deuxième fait est l'augmentation de la masse de données rassemblée avant le commencement d'émission. Ceci permet d'avoir une période d'émission garantie plus longue et de ramener ainsi les blocs restants à un débit plus faible. Le troisième fait est l'augmentation de la période inter-émissions des blocs pour assurer le débit MPEG. Cette période s'écrit comme le quotient de la taille des blocs par le débit MPEG.

Nous remarquons aussi que cette croissance est plus importante pour les petites tailles de bloc. Une analyse des taux d'occupation et des temps de réponse des files d'attente du modèle permet de constater le passage du goulot d'étranglement depuis les disques pour les petites tailles aux liens de la couche de réduction et du réseau d'aiguillage pour les tailles supérieures. Une comparaison entre le débit moyen global des disques et de celui des liens des transputers de réduction ou d'un étage d'aiguillage permet de retrouver cette constatation. Nous pouvons donc dire que le débit moyen global du serveur est régi par les disques pour les petites tailles de blocs (inférieures à 25 Ko) et par les transputers de réduction et le réseau d'aiguillage pour les tailles supérieures. Le comportement de la charge maximale constaté ci-dessus s'explique par le fait que le débit moyen des disques respectivement des liens de l'étage de réduction et du réseaux d'aiguillage , soit des goulots d'étranglement du système, croissent beaucoup plus vite avec les petites tailles de blocs qu'ils ne le font avec les grandes tailles. Ceci est du à la nature hyperbolique des fonctions donnant le débit moyen des disques (DmD) et les liens des transputers (DmL) en fonction de la taille des blocs (équ. 3 et 4). Notons que cette interprétation basée sur les débits moyens est indépendante du nombre de blocs anticipés et de la taille du serveur (nombre de disques). Cette indépendance a été vérifiée par les simulations nécessaires en faisant varier ces deux paramètres.

Le choix d'une taille optimale de bloc est un compromis entre la charge maximale et le temps de réponse qu'elle engendre. La superposition des deux courbes précédentes nous permet de proposer une taille optimale des blocs comprise entre 25 et 30 Ko indépendamment des autres paramètres. En effet il est inutile de travailler avec des tailles supérieures puisque la perte en temps de réponse ne justifie pas le très faible gain en charge maximale.

5.2. Étude du nombre de blocs anticipés

Pour étudier l'effet du nombre de blocs anticipés nous avons repris le serveur décrit dans le paragraphe précédent à savoir: 8 disques avec 400 séquences de durées uniformément réparties entre 10 et 30 secondes, des blocs de taille 30 Ko et une consultation uniforme des séquences. Pour un nombre d'anticipation donné nous faisons augmenter la charge et nous relevons le temps de réponse moyen correspondant jusqu'à saturation du système.

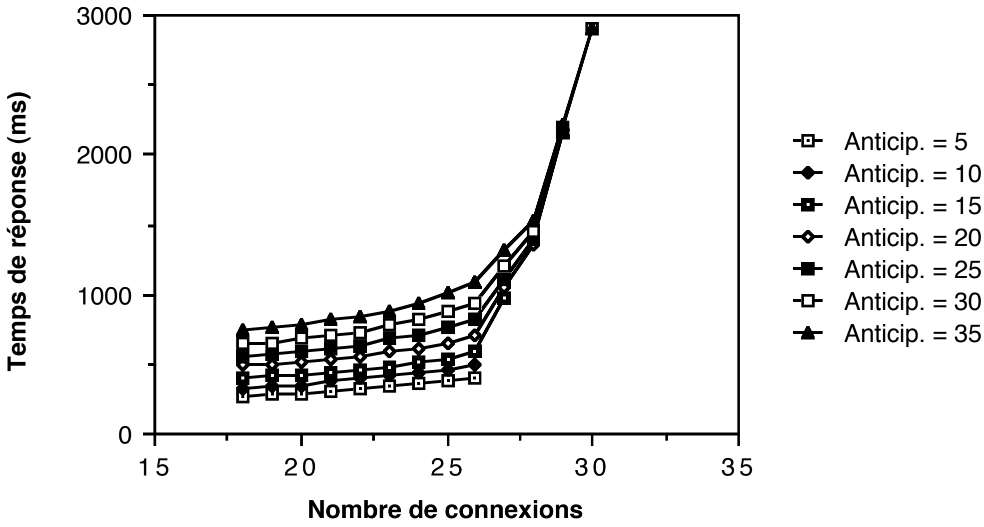

Figure 9. Effet du nombre d'anticipation sur le temps de réponse

Nous constatons, comme pour la taille des blocs, une augmentation du temps de réponse moyen avec le nombre d'anticipation (figure 9) et ceci pour les mêmes raisons décrites dans le paragraphe précédent.

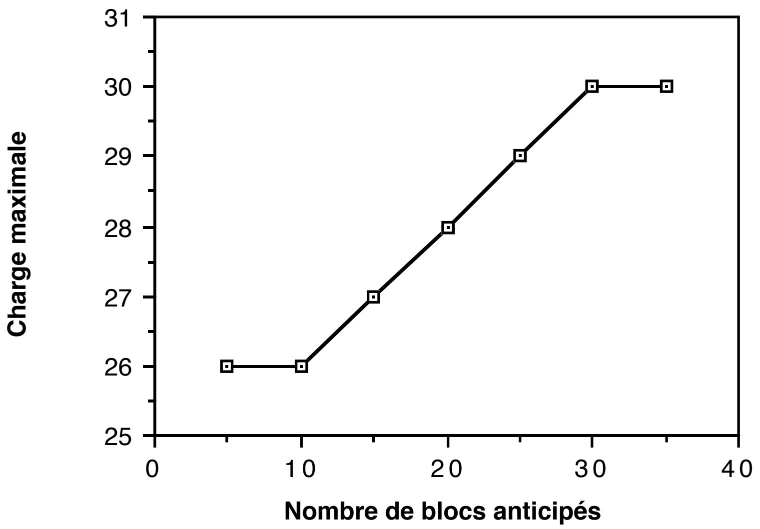

Figure 10. Effet du nombre d'anticipation sur la charge maximale

Nous constatons une augmentation de la charge maximale avec l'augmentation du nombre d'anticipation. Ceci peut s'expliquer, par l'une des raisons citées pour l'augmentation de la taille des blocs à savoir l'augmentation de la quantité de données collectée avant le début d'émission des séquences. Là aussi le choix du nombre d'anticipation est un compromis entre la charge maximale et le temps de réponse qu'il engendre. Nous constatons qu'il n'est pas intéressant de travailler avec une très grande anticipation en effet le gain en charge ne justifie plus la détérioration du temps de réponse engendrée particulièrement à l'approche de la saturation. La comparaison des deux courbes précédentes nous laisse proposer les nombres de 10 et 15 blocs pour la taille de 30 Ko.

5.3. Stratégie de placement des données

Nous appelons stratégie de placement des données la façon suivant laquelle les différents blocs sont affectés aux différents disques. La première stratégie de placement et la plus simple consiste à placer les blocs d'une manière séquentielle. C'est la stratégie utilisée dans toutes les simulations citées précédemment. On place le premier bloc sur le premier disque, le suivant sur le disque suivant et ainsi de suite. On reprend avec le premier disque lorsqu'on arrive au dernier. Pour la séquence suivante on continue avec le disque auquel on s'est arrêté. Pour un serveur à huit disques cela donne l'affectation suivante pour les huit premiers blocs (12345678). La deuxième stratégie que nous proposons a pour but de mieux distribuer et ordonnancer, sur les composants du système, les différentes opérations relatives à une même émission. A priori il n'est pas judicieux de placer deux blocs consécutifs d'une même séquence sur des disques reliés au même transputer de réduction. En effet sachant qu'ils seront demandés simultanément lors de la phase d'anticipation et l'un après l'autre durant la phase de transmission ils seraient en conflit sur le lien commun réduction-aiguillage. L'idée est donc de placer les blocs consécutifs sur des disques alternés, ce qui donne l'affectation suivante pour notre exemple (15263748). En regardant de plus près l'architecture, nous constatons qu'il est a priori judicieux aussi de mieux distribuer la charge sur les transputers d'aiguillage. Sachant que chaque quatre disques sont reliés au même transputer d'entrée du réseau d'aiguillage, pour assurer cette nouvelle distribution on mettra deux blocs consécutifs sur des disques espacés de quatre. Pour notre exemple nous aurons l'affectation suivante qui entrelace à la fois les transputers de réduction et les transputers d'aiguillage (15372648). C'est la seconde stratégie que nous proposons. Pour comparer ces deux stratégies, nous avons réalisé les simulations nécessaires sur un serveur à 64 disques avec 400 séquences uniformément réparties entre 10 et 30 secondes, avec une taille de bloc de 30 Ko, une anticipation de 10 blocs et une consultation uniforme des séquences.

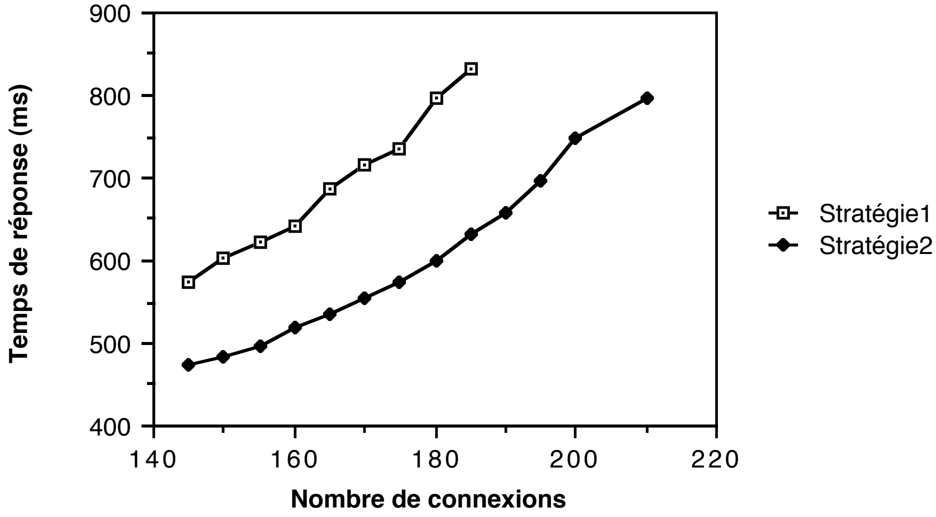

Figure 11. Effet de la stratégie de placement des blocs

Nous remarquons une sensible amélioration du temps de réponse avec la seconde stratégie de placement. Cette réduction varie entre 17 et 25% de la valeur du temps de réponse obtenu avec la première stratégie. Nous remarquons aussi l'augmentation de la charge maximale qui passe de 185 à 210 utilisateurs simultanés, soit une augmentation de 13%. L'analyse des paramètres de performances des différentes files du système montre une réduction du temps de réponse des files modélisant les liens de réduction et les liens du réseau d'aiguillage. Ces résultats confirment l'explication donnée a priori précédemment à savoir l'établissement d'un meilleur ordonnancement des différentes opérations relatives à une même émission.

Notons que l'amélioration des performances apportée par la seconde stratégie est due essentiellement à l'amélioration de l'ordonnancement des opérations pendant la phase d'anticipation où les A (nombre d'anticipation) premiers blocs sont demandés simultanément. Pendant la phase de transmission l'amélioration apportée par la seconde stratégie est moins importante à cause du fait que le taux d'arrivée des ordres d'extraction de blocs relatifs à une même requête est bien plus faible que le taux de service des disques. Pendant la phase de transmission la période de demande d'extraction de bloc n'est autre que la période d'émission des blocs (à chaque envoi de bloc on demande le prochain bloc à extraire). Pour les tailles qui nous intéressent, ce temps d'émission est largement supérieur au temps d'extraction du bloc. Pour une taille de 30 Ko, nous avons un temps d'émission de

160 ms (30 Ko / 1,5 Mbits/s) contre un temps d'extraction moyen de 43 ms et un temps d'extraction maximum de 63 ms (équation 1). Deux blocs consécutifs auront donc une faible chance de rentrer en conflit pour une même ressource pendant la phase de transmission. Avec un serveur à 8 disques et une anticipation de 10 blocs nous perdons le gain apporté par la seconde stratégie lors de la phase d'anticipation puisque nous sollicitons simultanément les 8 disques et ainsi les autres composants en question lors de l'anticipation de des 10 blocs, indifféremment de la stratégie adoptée. Le double entrelacement perd tout son intérêt lors de la phase d'anticipation. Les mêmes simulations précédentes réalisées sur un serveur à 8 disques avec une anticipation de 10 blocs, montrent une diminution moins importante du temps de réponse (maximum 5%) et aucune amélioration au niveau de la charge maximale. Ce qui vient confirmer ce qui précède.

En conclusion nous disons que la seconde stratégie de placement des blocs est plus intéressante que le placement séquentiel. Cette stratégie est encore plus intéressante pour les systèmes de grande taille puisqu'ils nous permettent de profiter des apports de cette stratégie lors de la phase d'anticipation. Dans la suite de notre étude et pour des raisons de simplification nous continuerons à utiliser la première stratégie qui consiste à placer les blocs d'une façon séquentielle sur les disques. Ce choix n'affecte en rien le comportement global du système et son interprétation.

6. Évolution des performances du système avec le nombre de disques

Dans cette partie nous nous intéressons à l'évolution du comportement du système et de ses performances en fonction de sa taille (nombre de disque). Pour ce faire nous allons étudier le comportement de cinq serveurs de différentes tailles: 8, 16, 32, 64 et 128 disques. Cette progression du nombre de disques en puissance de deux est imposée par l'architecture d'aiguillage oméga dont le nombre d'entrées varie en puissance de deux.

Dans cette étude nous avons travaillé avec un nombre d'anticipation égal à 10, 400 séquences de durées uniformément réparties entre 10 et 30 secondes et une consultation uniforme. Nous avons dressé pour chaque serveur le temps de réponse moyen en fonction de la charge pour des taille de blocs variant de 15 à 45 Ko. Nous observons le même comportement global (voir figures 7 et 8 pour serveur à huit disques) pour les différents serveurs avec des temps de réponse moyens et des charges maximales plus grands pour les serveurs a plus grand nombre de disques. Le système présente donc globalement le même comportement indépendamment de sa taille. C'est sur la base de ce résultat que nous avons effectué la majorité de notre étude sur des serveurs de 8 et 16 disques afin de gagner en temps de simulation. Dans ce qui suit nous nous intéressons à l'évolution de nos critères de performances en fonction du nombre de disques.

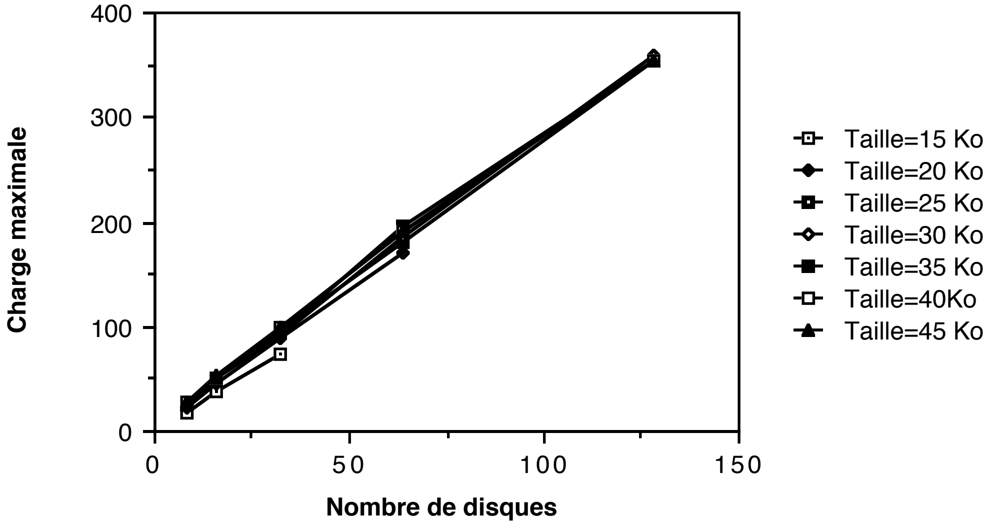

Figure 12. Évolution de la charge maximale en fonction du nombre de disques

Nous constatons une croissance pratiquement linéaire de la charge maximale en fonction du nombre de disques (figure 12). Cette évolution linéaire s'interprète par le fait que le serveur émet à la saturation pratiquement au débit de son goulot d'étranglement et que ce débit évolue linéairement en fonction du nombre de disques puisque le goulot d'étranglement n'est autre que les disques ou les liens de réduction et d'aiguillage. La pente de la droite est très bien approchée par le quotient du débit du goulot d'étranglement rapporté à un disque par le débit MPEG. Pour les tailles de blocs inférieures à 25 Ko le goulot d'étranglement se situe au niveau des disques. Le débit des disques étant sensiblement croissant pour ces tailles (équ.3), c'est ce qui explique la croissance de la pente observée sur la figure 12 pour cette gamme. Pour les tailles supérieures à 25 Ko, le goulot d'étranglement passe aux liens de réduction et d'aiguillage. Le débit des liens étant peut sensible à la variation de la taille des blocs supérieures à 25 Ko (équ.4), ceci explique alors la constance de la pente pour ces tailles.

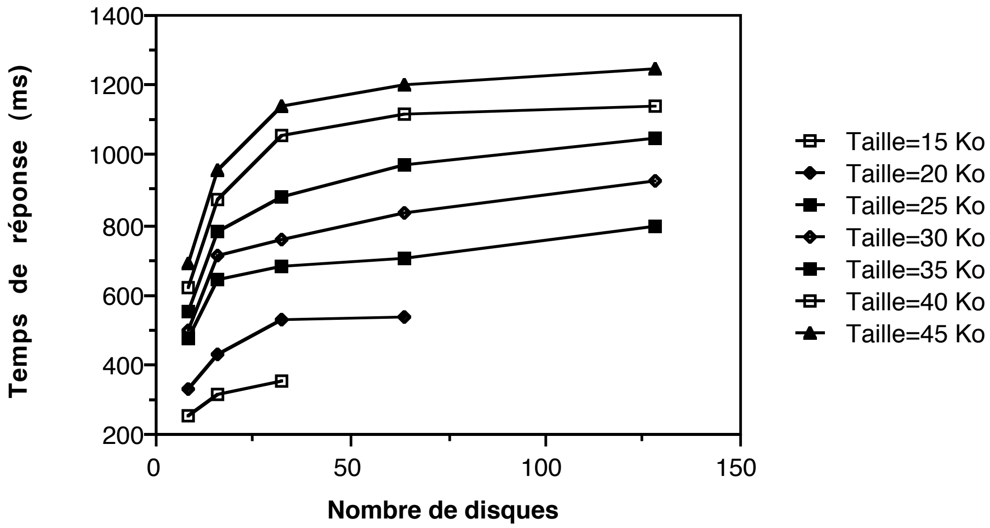

Figure 13. Évolution du temps de réponse moyen en fonction du nombre de disques

Sur la figure 13 nous avons porté le temps de réponse moyen juste avant la saturation en fonction du nombre de disques pour les différentes tailles de blocs étudiées. Nous nous sommes intéressés au temps de réponse moyen avant la saturation pour ne pas comptabiliser le saut de ce dernier à la saturation du système qui fausserait l'interprétation de ces résultats. Contrairement à la charge maximale qui évolue linéairement en fonction du nombre de disques, le temps de réponse moyen croit asymptotique par rapport à l'axe des x. Ce résultat est très intéressant puisqu'il nous permet de garantir une certaine qualité de service pour les serveurs à grand nombre de disques. Nous avons une augmentation linéaire de la charge contre une augmentation asymptotique du temps de réponse moyen. A titre d'exemple pour un serveur de 128 disques et pour une taille de bloc de 30 Ko, nous avons un temps de réponse moyen de 930 ms pour une charge de 360 utilisateurs.

Pour un serveur donné tous les blocs traversent le même nombre de couches de composants. Le temps de réponse du système est fortement dépendant de ce nombre de couches. Le comportement asymptotique du temps de réponse en fonction du nombre de disques s'explique alors par la relation logarithmique qui lie le nombre d'étages d'aiguillage au nombre de disques. En effet le nombre d'étages d'aiguillage (nbA) s'écrit en logarithme base deux en fonction du nombre de disques (nbD): $nbA = \log_2(nbD) - 2$. Dans notre étude, le passage d'un serveur à un autre est accompagné par le dédoublement du nombre de disques et l'ajout d'une seule couche de transputers au niveau du réseau d'aiguillage.

7. Étude de l'effet du profil de l'application

Dans cette partie nous allons nous intéresser au comportement du serveur vis-à-vis de l'application installée. Le but c'est d'étudier la stabilité de l'architecture vis-à-vis de ce facteur. L'application considérée pour ce serveur est un ensemble de séquences vidéo. Deux paramètres sont alors à étudier: le nombre de séquences et la distribution des durées des séquences. Pour la distribution des durées des séquences nous nous intéresserons à trois points, à savoir: l'effet de la longueur des durées, l'effet de la dispersion uniforme des durées et l'effet de la répartition des durées.

7.1. Étude du nombre de séquences

Nous allons dans ce paragraphe nous intéresser à l'effet du nombre de séquences sur le comportement du serveur. L'étude est faite sur un serveur à 8 disques, le nombre d'anticipation est pris égal à 10 et la taille des blocs à 30 Ko. Les séquences sont uniformément réparties entre 10 et 30 secondes et la consultation est supposée uniforme. Sachant que la charge maximale de ce serveur est aux alentours de 30 utilisateurs simultanés, nous avons étudié le comportement du serveur avec un nombre de séquences égal à: 100, 30, 20, 10 et 5. L'idée est de voir si le comportement du serveur se dégrade lorsque nous avons un faible nombre de séquences partagées entre un grand nombre d'utilisateurs simultanés et ainsi un conflit continu sur les séquences. Pour chaque nombre de séquences installées sur le serveur nous faisons augmenter la charge jusqu'à la saturation et on relève pour chaque charge le temps de réponse moyen.

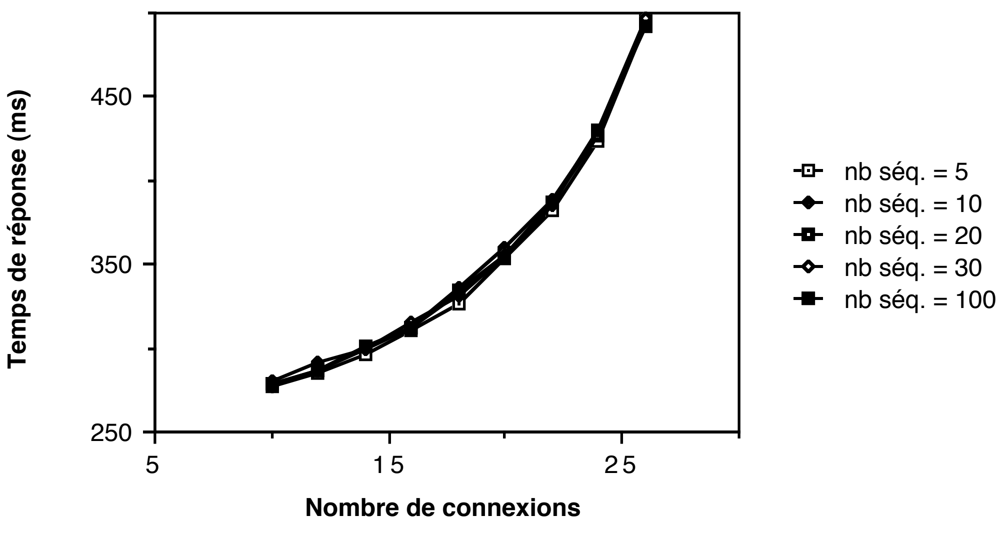

Figure 14. Effet du nombre de séquences

Nous remarquons sur la figure 14 que le serveur présente le même comportement, même temps de réponse moyen et même charge maximale, avec les différents nombres de séquences. Ce résultat confirme l'efficacité et la robustesse de l'architecture et du fonctionnement parallèles de notre système.

7.2. Effet de la longueur des durées des séquences

Pour toute cette étude de l'effet de la distribution des durées, nous allons considérer un serveur à 16 disques avec 300 séquences, une anticipation de 10 blocs de 30 Ko de taille. La consultation sera supposée uniforme.

L'intérêt de cette étude est de voir si le serveur se comporte différemment avec les séquences courtes et les séquences longues. Pour ce faire nous avons considéré 4 familles de séquences de mêmes durées constantes égales respectivement à : 10, 20, 30 et 40 secondes. Pour chaque famille de séquences nous procédons à une série de simulations en faisons augmenter la charge jusqu'à la saturation et en relevant le temps de réponse moyen pour chaque charge.

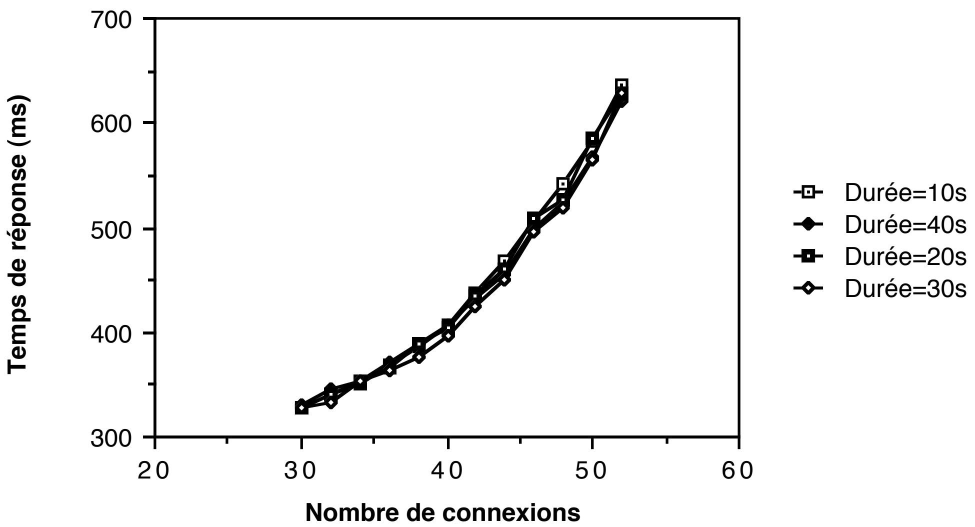

Figure 15. Effet de la longueurs des durées des séquences

Nous voyons sur la figure 15 que nous avons pratiquement le même temps de réponse et la même charge maximale pour les différentes familles de séquences. Le serveur est donc indifférent aux longueurs des séquences.

7.3. Effet de la dispersion uniforme des durées

L'intérêt de cette étude est d'analyser l'effet de la dispersion des durées sur le comportement du serveur. Pour ce faire, nous avons considéré 4 familles de séquences de durées différemment dispersées par rapport à la moyenne 20 secondes. La première famille est constituée de séquences de même durée constante égale à 20 seconde. Les autres familles sont constituées de séquences de durées de dispersion croissante. Ces durées sont uniformément réparties respectivement dans: [15s, 25s], [10s, 30s], [5s, 35s]. Nous avons procédé aux mêmes simulations que précédemment. Les résultats sont portés sur la figure ci-dessus.

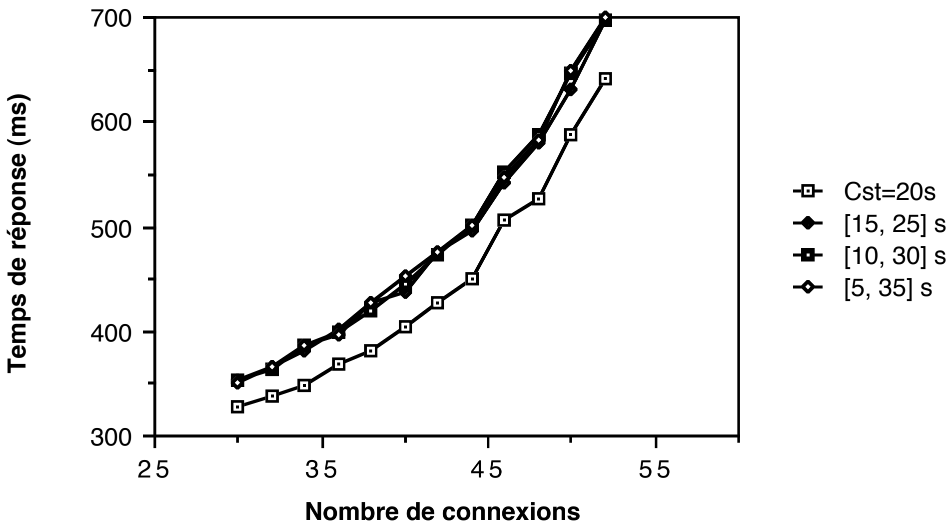

Figure 16. Effet de la dispersion uniforme des durées

La figure 16 montre que le serveur présente les mêmes performances pour des dispersions uniformes dans l'un ou l'autre des trois intervalles. D'autres simulations réalisées avec des données plus serrées autour de la moyenne 20s donnent le même résultat. Le serveur présente cependant une légère mais systématique amélioration du temps de réponse avec les durées constantes. L'analyse des critères de performances des différentes files du modèle nous permet de relever une légère baisse du temps de réponse des files du réseau d'aiguillage (goulot d'étranglement) qui expliquerait cette amélioration au niveau du temps de réponse du système. Cela s'explique peut être par le fait qu'avec des séquences de même durée le système fonctionne avec une meilleure répartition instantanée de la charge sur les différents composants et particulièrement au niveau du réseau d'aiguillage.

7.4. Effet de la répartition des durées

Dans toute cette étude nous avons considéré trois types de profils de répartition des durées des séquences: un profil constant ou toutes les séquences ont la même durée égale à une constante, un profil uniforme où les durées sont réparties uniformément dans un intervalle donné et un profil en gaussienne où la répartition des durées est donnée suivant un histogramme en gaussienne ou en portion de gaussienne.

La construction du profil uniforme se fait par tirage des durées suivant une loi uniforme dans l'intervalle considéré. La construction du profil en gaussienne se fait par la discrétisation d'une loi normale $N(m,\sigma)$ de moyenne m et d'écart type σ. L'intervalle [a,b] dans lequel varient les durées est subdivisé en nbI sous-intervalles adjacents équidistants I_i de centres x_i (i=1, ... nbI). Le nombre de séquences n_i uniformément réparties dans l'intervalle I_i est alors donné par: $n_i = (N(x_i) / \sum N(x_i)) \times \text{nbSeq}$, après arrondissement et normalisation ($\sum n_i = nbSeq$), où nbSeq est le nombre total de séquences. Le paramètre m positionne la gaussienne dans l'intervalle [a,b], pour m= (a+b)/2, nous avons une gaussienne centrée. Le paramètre σ agit sur la largeur de la gaussienne et ainsi sur la répartition des durée autour de m . Pour les petites valeurs de σ nous avons une forte concentration de durées autour de m , pour les grandes valeurs de σ nous avons une concentration plus homogène sur l'intervalle [a,b] jusqu'à une répartition uniforme pour σ très grand.

Pour étudier l'effet de la répartition des durées et ainsi de la composition de l'application, nous avons considéré des profils de même moyenne m et nous avons fait varier σ. Pour ce faire nous avons considéré les quatre familles de séquences suivantes: la première est constituée de séquences de durées uniformément réparties dans l'intervalle [10s, 30s] (ou gaussienne avec $\sigma = \infty$), les autres familles sont constituées de séquences de durées réparties suivant des histogrammes en gaussiennes de plus en plus serrées autour de la moyenne 20 secondes (σ décroissant) (figure 17).

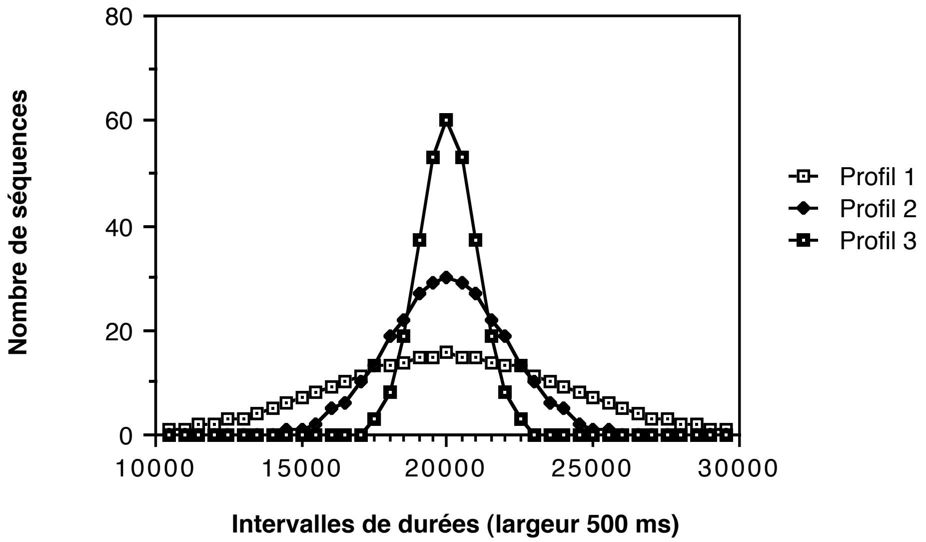

Figure 17. Histogrammes de répartition des durées en gaussienne

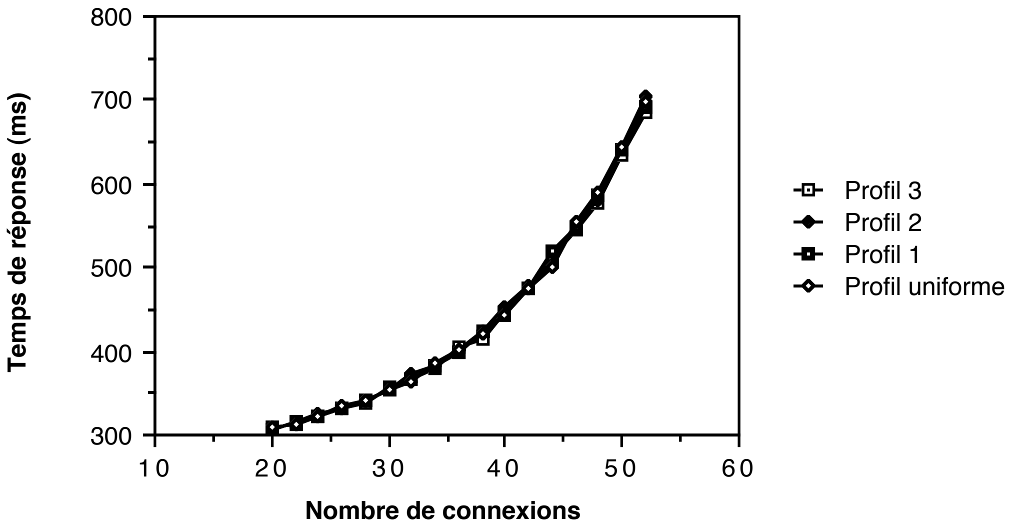

Figure 18. effet du profil de répartition

On voit sur la figure 18 que le serveur est indifférent à la répartition des durées des séquences. Il présente le même temps de réponse et la même charge maximale indifféremment de ce facteur.

Cette étude du profil d'application met en évidence la forte stabilité du système vis-à-vis de l'application installée. Elle prouve la robustesse de l'architecture vis-à-vis de ce facteur.

8. Étude de l'effet du profil de la consultation

Dans cette partie nous nous intéressons à l'étude du comportement du serveur vis-à-vis du comportement des utilisateurs. Cette étude est très importante puisqu'elle nous permet d'évaluer l'efficacité de la stabilité et de la robustesse cherchée dans le comportement du système vis-à-vis des utilisateurs, à travers l'architecture considérée.

8.1. Modélisation de la consultation

Pour modéliser la consultation deux approches sont possibles: une modélisation globale et une modélisation par utilisateur.
(a) modélisation globale de la consultation: cette approche consiste à modéliser globalement la consultation en définissant directement la probabilité de consultation de chaque séquence. Les probabilités sont déterminées suivant une loi donnée. Nous avons considéré deux profils type de consultation: un premier profil uniforme, toutes les séquences ont la même probabilité p d'être consultées (p =1/nbSeq). Et un second profil en gaussienne où les probabilités de consultation sont déterminées suivant une loi ou une portion de loi normale discrète.

La construction de ce deuxième type de profils se fait par la discrétisation d'une loi normale $N(m,\sigma)$. La probabilité de consultation d'une séquence numéro s_i est donnée par: $p(s_i) = N(pr(s_i)) / \sum N(s_i)$, où pr est une permutation dans l'ensemble des numéros des séquences {1, ..., nbSeq}. Le paramètre m permet de positionner la gaussienne dans l'intervalle [1,nbSeq] et ainsi de choisir la portion de gaussienne à considérer. Le paramètre σ permet de jouer sur la largeur de la gaussienne et ainsi sur la composition des probabilités. Pour les petites valeurs de σ nous avons une consultation très pointue vers une famille donnée de séquences. Pour les grandes valeurs de σ nous avons une consultation plus homogène jusqu'à avoir une consultation uniforme pour les très grandes valeurs de σ. Nous pouvons aussi avoir des probabilités en gaussienne constantes par groupe de g séquences en prenant $p(s_i) = N(pr(s_i) \text{ modulo g}) / \sum N(pr(s_i) \text{ modulo g})$.

Cette approche a l'avantage de permettre une vue globale de la consultation. Elle permet ainsi une étude fine du comportement du serveur vis-à-vis de situations particulières de consultations. Elle permet ainsi de visualiser l'effet d'un accès pointu à une famille donnée de séquences. Elle a aussi l'avantage de permettre une confrontation simple du profil consultation avec le profil de l'application. Elle permet par exemple des confrontations du type, les séquences les plus longues sont les plus regardées etc. ... Pour notre réseau fermé ce modèle fonctionnera comme suit: à la fin de son service le client (utilisateur) est à nouveau envoyé à l'entrée du serveur, nous tirons suivant le profil adopté la séquence à consulter et nous générons ainsi une nouvelle requête.

(b) Modélisation par utilisateur: La seconde approche consiste à passer par la modélisation du comportement des différents utilisateurs en modélisant leurs préférences. On définit plusieurs types d'utilisateurs. Chaque type étant caractérisé par un profil de consultation qu'on construit comme décrit précédemment. Les profils décrits précédemment ne seront plus les profils de la consultation globale mais les profils de consultation de différents types d'utilisateurs. La deuxième étape consiste à distribuer ces différents profils sur la population d'utilisateurs. Pour ce faire il faut fixer les proportions de chaque type d'utilisateurs et distribuer les profils en conséquence. L'avantage de cette approche c'est qu'elle permet une modélisation plus fidèle à la réalité. Notre population d'utilisateurs est composée de plusieurs profils types qui représentent les différentes classes de préférences de ces derniers. Le profil global de consultation sera donc une combinaison de ces différents profils. L'inconvénient de cette approche c'est qu'elle ne permet pas une vue globale a priori de la consultation et ne permet pas ainsi d'effectuer des études fines sur ce facteur consultation. Pour notre réseau fermé ce modèle fonctionnera comme suit: à la fin de son service le client est à nouveau envoyé à l'entrée du serveur, nous regardons son profil de consultation et nous tirons suivant ce profil la séquence à consulter, nous générons ainsi une nouvelle requête. Dans la suite nous utiliserons la modélisation globale de la consultation.

8.2. Comportement du serveur

Pour l'étude de l'effet du comportement des utilisateurs sur les performances du serveur, nous avons considéré quatre différents profils de consultation. Un profil de consultation uniforme et trois profils en gaussiennes qui expriment une consultation sélective de plus en plus orientée vers une famille de séquences données qui seront plus sollicitées que les autres (σ décroissant). Les simulations sont effectuées pour un serveur à 16 disques avec 400 séquences. Les séquences ont une durée constante égale à 20 secondes afin de mieux isoler l'effet du paramètre étudié. La taille des blocs est de 30 Ko et le nombre d'anticipation est de 10 blocs.

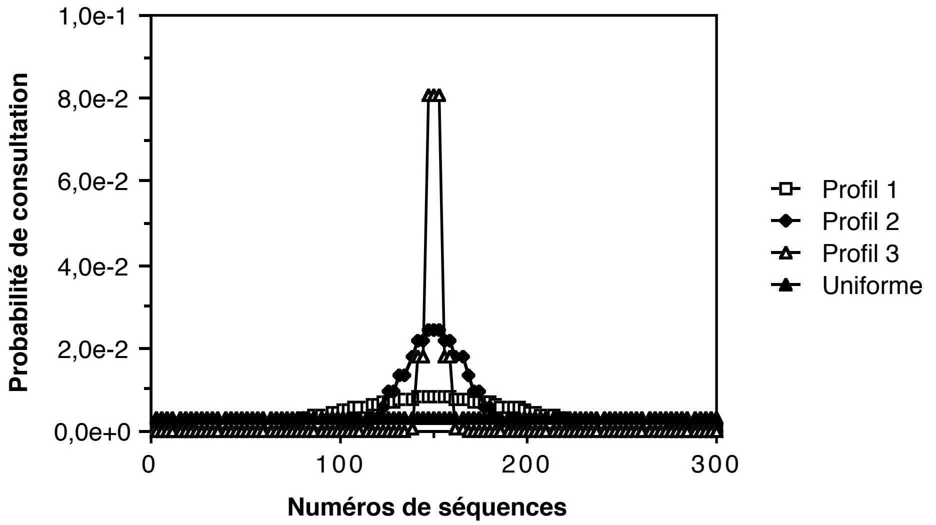

Figure 19. Profils de consultation

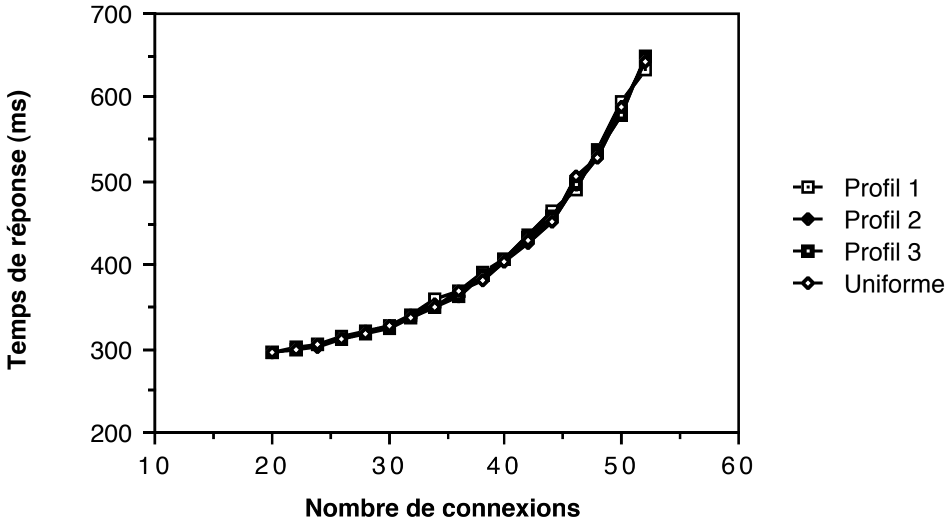

Figure 20. Comportement du serveur

Nous remarquons sur la figure 20 que le serveur réagit exactement de la même façon vis-à-vis des différents profils de consultation considérés. Ce comportement s'explique par la bonne répartition des séquences sur les différents disques. Un accès accentué vers une famille donnée de séquences ne modifie pas la répartition de la charge au niveau des disques ni donc au niveau du serveur. C'est un des avantages de la topologie parallèle de notre système. Le serveur est donc indifférent au profil de la consultation. Cette propriété est très intéressante puisqu'elle nous garantit une certaine stabilité dans le comportement du système vis-à-vis du facteur extérieur: comportement utilisateurs. Cette étude vient encore confirmer l'efficacité et la robustesse de l'architecture et du fonctionnement parallèle de notre système.

9. Conclusion

Cette étude nous a permis de mettre en évidence l'efficacité de l'architecture et du principe de fonctionnement adoptés. Le découpage des fichiers vidéo en blocs et la stratégie d'anticipation se sont montrés très efficaces pour l'augmentation du débit d'extraction et l'amélioration du comportement multi-utilisateurs du système. L'architecture et l'organisation adoptés permettent un bon fonctionnement parallèle du système. L'analyse des performances des composants du système montre une très bonne répartition de la charge globale sur les différents composants équivalents.

Nous nous sommes intéressés principalement à deux critères à savoir: le temps de réponse du système et la charge maximale qu'il peut supporter. Une étude préliminaire nous a permis d'analyser en détail le comportement de ces deux critères. Pour une charge donnée, le temps de réponse du système est relativement concentré autour de sa moyenne. Il est croissant avec la charge du système avec une augmentation de son coefficient de variation. La saturation du système est relativement tardive et rapide ce qui nous procure une large marge d'utilisation du système avec zéro rupture d'émission.

Nous avons considéré dans cette étude tous les paramètres internes et externes qui agissent sur le système. Nous avons étudié les paramètres internes de fonctionnement à savoir: la taille des blocs, le nombre d'anticipation et la stratégie de placement des blocs sur les disques. Nous avons étudié par la suite l'évolution des performances du système avec sa taille. Pour finir nous avons étudié l'évolution du comportement du système avec l'application qui y est installée et la consultation à laquelle il est soumis. Nous résumons dans ce qui suit les principaux résultats établis. L'augmentation de la quantité de données anticipées par l'augmentation de la taille de blocs ou du nombre d'anticipation, permet d'augmenter indéfiniment la charge maximale du système au détriment d'une augmentation du temps de réponse. L'étude détaillée de ce comportement nous a permis de proposer une taille de bloc et un nombre d'anticipation optima. La stratégie de placement des blocs s'est avérée importante. Elle permet d'avoir un meilleur ordonnancement des tâches au niveau du système. Un placement intelligent des blocs sur les disques permet d'améliorer sensiblement les performances du système. L'analyse de l'évolution des performances avec la taille du système montre une croissance pratiquement linéaire de la charge maximale avec le nombre de disques contre une croissance logarithmique du temps de réponse. Ce comportement est très intéressant surtout pour les serveurs de grandes dimensions puisqu'il permet d'avoir une importante charge maximale avec un temps de réponse acceptable. La dernière étape de cette étude concerne l'effet des deux facteurs extérieurs application et utilisateurs, sur le comportement du système. Pour l'étude de l'effet du profil de l'application installée nous nous sommes intéressés au nombre de séquences, à la longueur des durées, à la dispersion des durées et finalement à la répartition des durées. Le système est pratiquement insensible à tous ces paramètres. Pour l'étude de l'effet du facteur utilisateur, nous avons considéré différents profils de consultation des séquences: de la consultation uniforme à la consultation extrêmement orientée vers une famille très restreinte de séquences. Là aussi le système présente les mêmes performances indépendamment du profil de la consultation. Ces résultats sont très intéressants puisqu'ils nous garantissent une forte stabilité vis-à-vis de l'environnement extérieur et particulièrement vis-à-vis du comportement des utilisateurs. Tous ces résultats viennent confirmer encore l'efficacité et la robustesse de l'architecture et du fonctionnement parallèle de notre système.

Notons que le principe de modélisation utilisé dans cette étude a été validé sur une architecture voisine. La validation repose sur des tests effectués sur un prototype du système considéré.

Le réseau d'aiguillage joue un rôle important dans le fonctionnement du système. Il est à la fois goulot d'étranglement du système et à l'origine d'une majeure partie du temps de réponse du système. Il serait intéressant d'approfondir l'étude du réseau en comparant différentes éventuelles architectures.

Bibliographie

[1] Boral H., Parallelism and data management, Proceedings 3rd Conference on Data and Knowledge Bases, June 1988.

[2] Damm G., Babonneau G., Beylot A.L. and Becker M., Performance evaluation of a multimedia server for ATM networks, Proceedings of the 27th Annual Simulation Symposium, La Jolla, April 1994, pp. 41-50

[3] Ghafir H. and Ghadwick H., Multimedia Servers - Design and performance, Proceedings of IEEE Globecom, 1994, pp. 886-890.

[4] Kim M., Parallel operation of magnetic disk storage devices: Synchronized disk interleaving, Proceedings of the Fourth International Workshop on Database Machines, Springer Verlag, March 1985.

[5] Legall D., MPEG: a video compression standard for multimedia, Communications of ACM, Vol.34, No. 4, April 1991, pp. 47-58.

[6] Livny M., Khoshafians S. and Boral H., Multi-disk management algorithms, Proceedings of the 1987 ACM SIGMETRICS International Conference on Measurement and Modelling of Computer Systems, May 1987, pp. 69-77.

[7] Potier D. and Veran M., QNAP2: a Portable Environment for Queuing Systems Modelling, International Conference on Modelling and Performance Analysis Tools, May 1984.

[8] Tobagi F.A., Pang J., Baird R., Gang M., Streaming RAID / sup TM/ - a disk array management system for video files, Proceedings ACM Multimedia 93, pp. 393-400.