

Méthode formelle d'évaluation de la qualité de service des applications multimédias réparties

Qualité de service d'un assemblage de composants

Sophie Laplace—Marc Dalmau—Philippe Roose

LIUPPA -Département Informatique de l'I.U.T. de Bayonne
Château Neuf – Place Paul Bert-64100 Bayonne – France
Téléphone : 05-59-57-43-38, 05-59-57-43-26, 05-59-57-43-28
Télécopie : 05-59-57-43-29
{laplace, dalmau, roose}@iutbayonne.univ-pau.fr

Résumé : La qualité du service fourni est la principale préoccupation de l'utilisateur en particulier dans le domaine des applications multimédias réparties. En effet, dans ce cas, le support essentiel, l'Internet, n'est pas parfaitement adapté à la transmission des flux en jeu. Afin de résoudre ce problème, nous plaçons la qualité de service au centre de notre méthode de conception d'applications multimédia réparties. Ainsi, pour assurer une qualité de service à l'utilisateur, et ceci malgré les variations du contexte d'exécution, nous nous appuyons sur une conception à base de composants logiciels. Ce sont alors des reconfigurations de l'application supervisées par une plate-forme qui doivent nous permettre d'adapter l'application aux conditions d'exécution.

A tout moment, la plate-forme choisit la configuration qui présente la meilleure qualité de service. Cet article présente une méthode d'évaluation de la qualité de service de tous les assemblages possibles de composants permettant d'assurer un service donné. Cette méthode est basée sur un modèle de qualité de service comportant deux critères représentant les caractéristiques intrinsèques de l'assemblage et les caractéristiques dépendant du contexte d'exécution. Nous développons et illustrons par un exemple la principale difficulté qui est l'extrapolation de la qualité de service d'un assemblage de composants non utilisés à partir de mesures réalisées sur les entités actuellement utilisées et des informations sur les composants et les flux dont dispose la plate-forme dans notre modèle d'application.

Mots-clés : qualité de service, applications adaptatives, applications multimédias réparties, méthode de conception, composant.

Abstract : Quality of service is the main user concern for distributed multimedia application because Internet does not suit to multimedia transmission. That is the reason why we propose a component based design method centred on quality of service for distributed multimedia applications. Then the application is adapted to user and to run-time conditions thanks to reconfigurations supervised by a platform.

The platform choose the configuration with the best quality of service. It is why we present in this article a quality of service evaluation method for component assembly which allows to evaluate the application configuration. So we propose a quality of service model with two criteria, intrinsic characteristics and run-time context dependant characteristics. With an example, we develop and illustrate the main difficulty which is the extrapolation of the quality of service of an assembly of non executed components. For this, we use measures achieved from executed entities and informations on components and flow holded by the platform in our model.

Keywords : quality of service, adaptive application, distributed multimedia applications, conception method, component

Introduction

La qualité du service fourni est la principale préoccupation de l'utilisateur en particulier dans le domaine des applications multimédias réparties. En effet, dans ce cas, le support essentiel, l'Internet, n'est pas parfaitement adapté à la transmission des flux en jeu. Afin de résoudre ce problème, nous plaçons la qualité de service au centre de notre méthode de conception d'application multimédia répartie.

Ainsi, pour continuer à assurer une qualité de service correcte à l'utilisateur malgré les variations du contexte d'exécution, nous nous appuyons sur une conception à base de composants logiciels. Ce sont alors des reconfigurations de l'application supervisées par une plate-forme qui doivent nous permettre d'atteindre nos objectifs.

Dans cette perspective, nous avons proposé un modèle d'architecture d'applications multimédia [5] où les composants sont assemblés pour fournir une fonctionnalité à un premier niveau puis un service au niveau supérieur. La communication entre ces assemblages répartis sur le réseau est alors réalisée par des flux d'informations. De plus, nous avons établi une méthode formelle permettant de générer toutes les configurations possibles de l'application [6].

Notre objectif est actuellement de définir une méthode d'évaluation de la qualité de service des assemblages de composants en cours ou non d'exécution de manière à pouvoir choisir la composition future de l'application. Pour cela, il faut tout d'abord être capable d'évaluer le service fourni par un composant en regard des souhaits de l'utilisateur puis de mesurer l'influence du contexte d'exécution de ce composant et en particulier de la circulation des flux d'informations pour pouvoir enfin définir la qualité de service d'un assemblage de composants.

C'est pourquoi, nous présentons dans une première partie la notion de qualité de service et le modèle que nous comptons utiliser pour représenter celle d'une entité logicielle puis dans une seconde partie, nous exposerons une méthode d'évaluation de la qualité de service d'un assemblage de composants. Enfin, la troisième partie nous permettra d'illustrer notre propos par un exemple issu de notre prototype de vidéoconférence.

1. Modélisation de la qualité de service

1-1- Définition de la qualité de service

La qualité de service est « l'effet collectif des performances du service qui détermine le degré de satisfaction de l'utilisateur du service. » [2]. Pour la décrire dans le contexte des applications multimédias réparties, il faut donc déterminer « l'ensemble des caractéristiques quantitatives et qualitatives d'un système multimédia distribué qui permettent d'atteindre la fonctionnalité requise pour une application » [9].

Nous pensons donc que la qualité de service traduit l'adéquation entre le service souhaité par l'utilisateur et celui qui lui est fourni. Définir la qualité de service d'un assemblage de composants, consistera donc à définir l'ensemble des propriétés, des caractéristiques, des performances du service fourni au client présentées en comparaison avec le service attendu par celui-ci [4]. Une caractéristique du service fourni ne sera donc pertinente pour la qualité de service qu'à condition d'avoir des conséquences sur le service attendu par le client.

La méthode d'évaluation de la qualité de service contiendra donc obligatoirement une première étape de spécification des souhaits de l'utilisateur en terme de service pour chaque assemblage de composants puis une seconde étape de comparaison du service fourni par l'application avec les caractéristiques définies précédemment. L'évaluation de la qualité de service constituera alors le résultat de cette comparaison.

1-2- Modélisation de la qualité de service

1-2-1- Les deux types de critères de qualité de service

Nous considérerons qu'à un instant donné, la qualité de service d'une entité dépend de deux types de critères.

Le premier décrit la qualité que l'entité propose de par sa conception : nous parlerons de **critère intrinsèque**. Sa principale caractéristique est d'être indépendant du contexte c'est-à-dire du réseau, du matériel ou de l'environnement non informatique de l'application.

Ce critère intrinsèque permet d'évaluer, par exemple, l'adéquation de la fonctionnalité de l'entité avec les vœux de l'utilisateur. Il peut alors s'agir de caractéristiques aussi diverses que la langue des sous-titres ou le format de compression d'image. Il peut aussi traduire la conformité du support matériel comme les dimensions d'un écran ou l'adéquation des performances comme la rapidité de suivi d'un locuteur.

A l'inverse, le second critère traduit l'influence du contexte d'exécution sur le service fourni: il sera appelé **critère contextuel**. Il décrit le respect des contraintes temporelles autant en local qu'en réseau et le respect du service malgré les perturbations amenées par l'environnement matériel telles que le bruit ou les erreurs de transmission.

Pour un rôle ou une fonction donnée, les critères de qualité de service étudiés seront définis à partir des souhaits de l'utilisateur. Les critères contextuel et intrinsèque seront donc plus ou moins nombreux selon les composants disponibles et les exigences de l'utilisateur.

1-2-2- Représentation de la qualité de service

La qualité de service a donc deux dimensions indépendantes, intrinsèque et contextuelle, qu'il est intéressant de conserver lors des comparaisons car deux entités peuvent fournir la même qualité mais l'une étant à sa qualité intrinsèque optimale et l'autre étant uniquement limitée par le contexte. Or le potentiel d'évolution est un argument de comparaison entre deux configurations. Nous proposons donc une représentation de la qualité de service à deux dimensions comme un couple de nombres correspondant chacun à une note. Ce couple pourra alors être représenté par un point du plan représentant les qualités de service disponibles pour un rôle ou une fonction.

Pour chacune de ces dimensions, la qualité de service traduit une distance par rapport à une qualité optimale qui est celle spécifiée par l'utilisateur pour un rôle donné. Nous choisissons donc de la représenter par deux réels positifs, I_n et C_o , compris entre 0 et 1. La note de 1 sera octroyée lorsque le service est inacceptable pour l'utilisateur, la note de 0 lorsque le service est celui attendu par l'utilisateur. La qualité de service idéale, souhaitée par l'utilisateur, est donc représentée par le point (0,0).

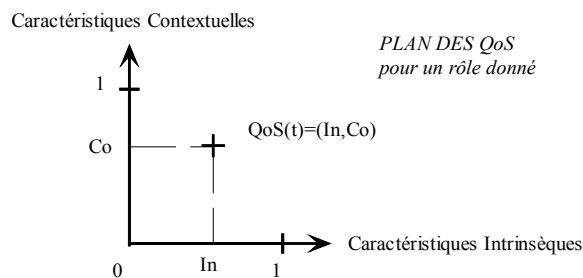


Figure 1 : Représentation de la qualité de service fournie à un instant t

Une entité logicielle sera alors caractérisée par l'ensemble des points représentant le service qu'elle propose et la dépendance de ce service envers le contexte. Cette caractérisation est tout à fait comparable à celle que l'on connaît pour les composants ou circuits électroniques à la différence qu'elle ne décrit pas le rôle mais la qualité.

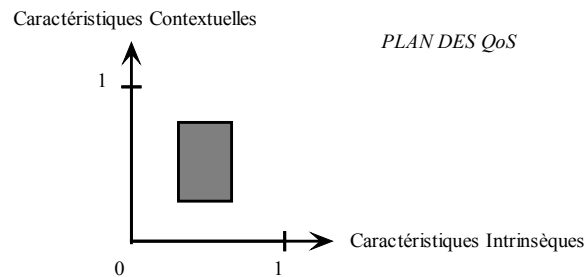


Figure 2 : Représentation des qualités de service pouvant être fournies par une entité

1-2-3- Intérêt de cette représentation pour l'évaluation

Cette représentation permet d'adapter la méthode d'évaluation au cas étudié et à l'utilisateur. La qualité de service est assimilée à la distance entre le point représentant l'entité et le point (0,0). Toutefois, entre deux entités placées à la même distance et donc sur le même arc de cercle centré en (0,0) il est possible de différencier celle offrant le plus grand potentiel comme étant celle placée le plus loin de l'axe des abscisses. En effet, si les conditions liées au contexte s'améliorent, elle réduira sa distance au centre offrant ainsi une qualité de service supérieure.

2. Méthode d'évaluation d'un assemblage de composants logiciels

De même que le rôle d'un assemblage de composants se décompose en rôles des composants le constituant [6], le respect des différentes contraintes par l'entité dépend du respect de ces contraintes par les composants la constituant. De plus, parmi les contraintes à respecter figurent des contraintes temporelles. Or c'est le réseau qui apporte la majorité des perturbations dans ce domaine. Les caractéristiques contextuelles des flux entre composants doivent donc être prises en compte dans l'évaluation de la qualité de service de l'entité. Les caractéristiques contextuelles d'un assemblage dépendent donc des caractéristiques contextuelles des composants et des flux le constituant.

2-1- Présentation de la problématique et de la méthode

2-1-1- Problématique

En cours d'exécution, la plate-forme supervisant une application doit évaluer la qualité de service offerte par les assemblages de composants utilisés et la qualité de service qui pourrait être offerte par les autres assemblages possibles. Ainsi, à tout moment, elle est capable de restructurer l'application en choisissant l'assemblage le plus adéquat.

Cependant, l'évaluation de la qualité de service des assemblages de composants est rendue délicate par la répartition des composants sur différents sites. En effet, l'environnement et le réseau influent grandement sur les flux de données échangés entre les constituants d'une entité et conditionnent donc le service fourni. Or pour les assemblages non actuellement utilisés mais dont on veut évaluer la qualité de service, aucune mesure n'est possible.

Pour évaluer la qualité de service dans le contexte de l'architecture d'applications multimédias que nous avons proposée, la plate-forme dispose de documents XML [3] décrivant les composants et leur rôle, les assemblages en cours d'exécution ou possibles et les flux de données. De plus, elle peut effectuer des mesures sur les entités en cours d'exécution et sur les flux.

2-1-2- Méthode

La méthode que nous proposons comporte quatre étapes.

La première permet à l'utilisateur de choisir les caractéristiques du service auxquelles il attache de l'importance.

Dans un second temps, il s'agit de spécifier le comportement actuel ou supposé d'un assemblage de composants logiciels en regard de chaque caractéristique intrinsèque ou contextuelle de qualité de service.

La troisième étape consiste à déterminer, à partir de ces données, une note de qualité de service pour chaque caractéristique en comparant le service fourni ou potentiellement fourni avec le service souhaité par l'utilisateur.

Enfin, grâce à une moyenne des notes des caractéristiques pondérées par l'utilisateur, chacun des deux critères, intrinsèque (In) et contextuel (Co), est évalué. Il est alors possible de déterminer le point définissant la qualité de service de l'entité $QoS(t)=(In, Co)$.

Nous développons ici la principale difficulté qui est d'extrapoler la qualité de service d'un assemblage potentiel de composants à partir des données dont nous disposons.

2-2- L'étape centrale : Description du service fourni par l'assemblage

Pour chaque caractéristique, il s'agit de spécifier le comportement actuel ou supposé de l'assemblage de composants.

2-2-1- Pour les caractéristiques intrinsèques

Que l'assemblage soit exécuté ou non, ses caractéristiques intrinsèques sont définies à partir des documents XML décrivant les composants. En effet, certains composants influenceront sur certaines caractéristiques en fonction de la décomposition fonctionnelle de l'entité.

2-2-2- Pour les caractéristiques contextuelles

La méthode d'évaluation des caractéristiques contextuelles sera distincte selon que l'assemblage sera exécuté ou non.

Dans le premier cas, une mesure des propriétés des flux permettra d'évaluer chaque caractéristique.

Dans le second cas, il sera nécessaire d'extrapoler, par approximation linéaire par exemple, le service fourni par un composant à partir de mesures réalisées sur le réseau et sur le matériel et à partir des informations caractéristiques décrivant le comportement du composant dans le document XML. Un diagramme des flux représentant l'assemblage de composants permettra alors de définir les spécificités du service fourni par cet assemblage.

Cependant, pour chaque caractéristique, il est nécessaire de définir une méthode d'évaluation spécifique. Nous proposons que le temps de traitement puisse ainsi être déterminé à partir de la somme de tous les temps de traitement, composants et flux, de la chaîne élémentaire. Le débit ou la quantité d'information traitée par unité de temps sera équivalente au minimum des débits d'une chaîne. Enfin, le taux de perte sera assimilé à une multiplication des taux de perte de la chaîne.

2-2-2-1- Caractérisation du service d'un composant

Le service d'un composant est caractérisé à partir des informations du document descriptif XML décrivant le composant. Ce document peut regrouper les caractéristiques matérielles minimales pour un bon fonctionnement, les performances nominales pour des conditions données, les courbes décrivant le flux de sortie en fonction du flux d'entrée ainsi qu'une méthode d'extrapolation adaptée au composant. Le service fourni par le composant est caractérisé à partir de ces informations et des mesures faites sur le réseau ou le matériel en cours d'exécution.

2-2-2-2- Caractérisation d'un assemblage

Pour cette étude, l'assemblage est représenté par un graphe des flux créé à l'aide des documents XML décrivant les flux. Ce graphe est orienté et doublement valué. Les sommets représentent les composants de l'entité étudiée et les arcs orientés représentent les flux d'information échangés entre ces composants.

La valuation des sommets est un vecteur noté V_i représentant les caractéristiques du composant modélisé par le sommet. Elle traduit d'une part l'influence du contexte local sur les caractéristiques de qualité de service telles que le respect des contraintes temporelles et le respect du service malgré les perturbations et d'autre part les caractéristiques intrinsèques.

La valuation des arcs notée Vf_j représente l'influence des caractéristiques du flux d'information représenté par l'arc. Elles constituent le critère contextuel de qualité de service du flux si l'on considère que celui-ci remplit la fonction de transmission de données. Ce flux peut être matériellement sur la même machine ou transiter par le réseau. Cependant la qualité de service d'un flux est mesurable si ce flux est en cours d'exécution. Dans le cas contraire, on extrapole à partir des flux existants pour définir les débits ou les délais.

Nous proposons que le flux soit caractérisé par deux informations contextuelles : les pertes qui traduisent un débit insuffisant et les délais. Ces informations ont l'avantage d'être mesurables [1].

Le graphe décrivant l'assemblage de composants et les flux est bi-valué comme dans l'exemple suivant:

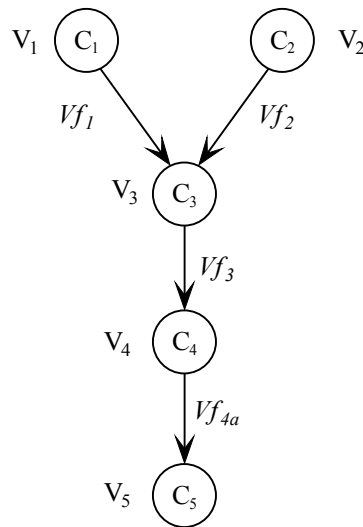


Figure 3: Graphe des flux d'un assemblage de composants

Chaque vecteur est composé différemment selon qu'il traduit la qualité de service d'un composant ou d'un flux :

- pour les composants : V_i

| | | |
|----------|---|--------------------------------|
| X_{i1} | } | caractéristiques intrinsèques |
| X_{i2} | | |
| ... | | |
| Y_{i1} | } | caractéristiques contextuelles |
| Y_{i2} | | |
| ... | | |
- pour les flux Vf_j

| | | |
|----------|---|--------------------------------|
| Y_{j1} | } | caractéristiques contextuelles |
| Y_{j2} | | |
| ... | | |

Pour chaque caractéristique, intrinsèque X ou contextuelle Y , le concepteur de l'application doit choisir une méthode d'évaluation permettant de définir la caractéristique de l'assemblage de composants à partir des caractéristiques des composants et flux et du parcours du graphe précédent.

Pour le temps de traitement et délais d'acheminement, il pourra ainsi s'agir du maximum entre les deux sommes des temps associés aux chemins élémentaires pris entre les sommets initiaux, C_1 et C_2 , et le sommet final, C_5 .

Lors de l'étude du débit, le concepteur pourra choisir pour la caractéristique associée aux flux, Y , une fonction s'appliquant aux caractéristiques des entités situées en amont car les flux modifient les performances contextuelles des composants.

3. Exemple : Assemblage de composants pour la restitution de l'image

3-1- Présentation de l'exemple étudié

L'exemple que nous présentons pour illustrer notre propos est issu de l'étude d'une vidéoconférence via Internet [7]. Nous nous intéressons ici à l'assemblage de composants qui assure la restitution de l'image de la conférence à l'utilisateur.

| Pour réaliser cette fonction, le | | Rôle du composant | |
|----------------------------------|---------|-------------------------------|-----------|
| Nom | symbole | Définition | symbole |
| Caméra | CA | Capture des images | R_{111} |
| Carte d'acquisition | AC | Acquisition numérique | R_{112} |
| Compression 3 | TI_3 | Traitement avant transmission | R_{12} |
| Compression 2 | TI_2 | Traitement avant transmission | R_{12} |
| Élaboration du clone | TI_C | Traitement avant transmission | R_{12} |
| Émission vidéo | EV | Émission d'images | R_{131} |
| Émission du clone | EC | Émission d'images | R_{131} |
| Réception vidéo | RV | Réception d'images | R_{132} |
| Réception du clone | RC | Réception d'images | R_{132} |
| Restitution vidéo | RE | Traitement après transmission | R_{14} |
| Restitution du clone | RL | Traitement après transmission | R_{14} |
| Carte vidéo | CV | Restitution numérique | R_{151} |
| Écran | ER | Visualisation | R_{152} |

Tableau 1. Composants disponibles pour la restitution de l'image

La conception de l'application définit les différentes compositions possibles pour cet assemblage sous la forme des différents chemins élémentaires entre le sommet d'entrée et le sommet de sortie du graphe suivant [6]:

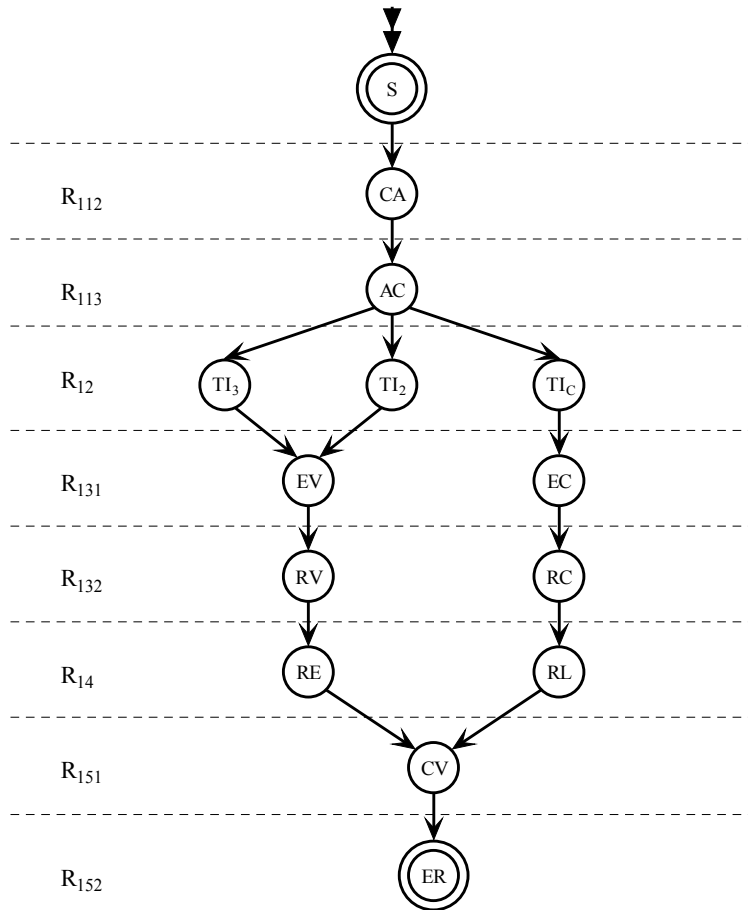


Figure 4. *Différentes compositions de la restitution de l'image*

Le concepteur dispose donc de trois configurations pour cette entité et donc de trois qualités de restitution de l'image. La première qualité est imposée par le composant de compression TI_3 , la seconde par le composant de compression TI_2 et la dernière correspond à la transmission d'un clone élaboré par le composant TI_C

3-2- Analyse de la qualité de service

3-2-1- Qualité du service rendu

Le critère intrinsèque de cet assemblage est l'adéquation de la fonctionnalité avec celle attendue par l'utilisateur tandis que le critère contextuel concerne le respect des contraintes temporelles.

3-2-2- Qualité attendue par l'utilisateur

Etant donné les composants disponibles, le concepteur définit les critères de qualité de service qui vont varier en cours d'exécution.

Tout d'abord, le seul critère intrinsèque sera l'adéquation de la fonctionnalité. En effet, il y aura soit une transmission de l'image avec deux qualités différentes soit une transmission du clone. Le critère intrinsèque In décrit donc une seule caractéristique.

Ensuite, les caractéristiques contextuelles sont le nombre d'images par seconde, la fréquence, et le taux de perte d'information qui sont des caractéristiques de transmission disponibles avec JMF [8] support de notre prototype.

Avant l'exécution de la vidéoconférence, l'utilisateur a indiqué ce qu'il souhaite et quelle note il donne aux différentes possibilités. Ainsi il précise que la qualité liée au composant TI_3 le satisfait pleinement alors qu'il ne souhaite pas recevoir une image issue du clone. Les notes sont alors les suivantes :

Transmission du clone : $In_{TI_c} = 1$

Transmission de l'image avec TI_2 : $In_{TI_2} = 0,5$

Transmission de l'image avec TI_3 : $In_{TI_3} = 0$

De même, l'utilisateur précise ses souhaits relatifs aux deux caractéristiques contextuelles. Il choisit d'attribuer la note 0 à une fréquence de 25 images par seconde et au-dessus, et une note de 1 à une fréquence de 20 images par seconde et en dessous. De plus, il tolère une perte maximale de 10% des données.

Enfin, l'utilisateur indique l'importance relative qu'il attribue aux deux caractéristiques du critère contextuel en précisant le poids dans la moyenne pondérée qu'il souhaite leur attribuer : 2 pour la fréquence et 1 pour le taux de perte.

3-2-3- Evaluation de la qualité de service de l'assemblage de composants pour la restitution de l'image

3-2-3-1- Méthode d'évaluation des deux critères

La note In relative au critère intrinsèque sera celle attribuée à la fonctionnalité : pour le clone, 0,5 pour TI_2 , 0 pour TI_3 puisqu'il n'y a qu'une caractéristique à prendre en compte.

La note relative au critère contextuel Co est la moyenne pondérée des notes accordée à la fréquence, Co_A , et au taux de perte Co_B .

$$Co = \frac{2.Co_A + 1.Co_B}{2 + 1}$$

Pour déterminer Co_A et Co_B , on utilise une proportionnalité à partir des indications données par l'utilisateur.

Co_A va dépendre du nombre d'images par seconde n :

- si $25 < n$ alors $Co_A = 0$
- si $20 \leq n \leq 25$ alors $Co_A = \frac{25 - n}{25 - 20}$
- si $n < 20$ alors $Co_A = 1$

Co_B va dépendre du taux de perte en % t :

- si $t \geq 10$ alors $Co_B = 1$
- si $t < 10$ alors $Co_B = \frac{t}{10}$

3-2-3-2- Exemple d'évaluation de la configuration en cours d'exécution

On suppose qu'à un instant t_1 , la configuration de l'assemblage choisie par la plate-forme est celle contenant le composant TI_2 . Le critère intrinsèque vaut alors $In = 0,5$.

Les mesures faites sur le réseau donnent $n=25$ et $t=0$ ce qui permet d'évaluer le critère contextuel :

$$Co = \frac{2.Co_A + 1.Co_B}{2 + 1} = \frac{2.0 + 1.0}{2 + 1} = 0$$

La qualité de service est donc donnée par le point $QoS(t_1) = (In, Co) = (0,5, 0)$ représenté sur le schéma suivant dans le plan des qualités de service du rôle de la restitution d'image.

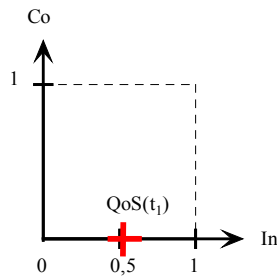


Figure 5: Qualité de service de la première configuration

La qualité de service obtenue n'est pas la qualité de service idéale pour l'utilisateur. La plate-forme va donc chercher à l'améliorer. Elle va évaluer la qualité de service de la configuration ayant une note intrinsèque meilleure que celle en cours, c'est-à-dire celle contenant le composant TI_3 , puisque la note contextuelle ne peut être améliorée.

3-2-3-3- Evaluation de l'autre configuration possible

Du point de vue intrinsèque, la configuration étudiée est ici celle contenant le composant TI_3 , elle réalise donc une transmission des images.

Du point de vue contextuel, il est nécessaire de déterminer les caractéristiques de fréquence et de taux de perte des composants logiciels et flux intervenant dans l'assemblage.

Cet assemblage est décrit par le graphe des flux suivant :

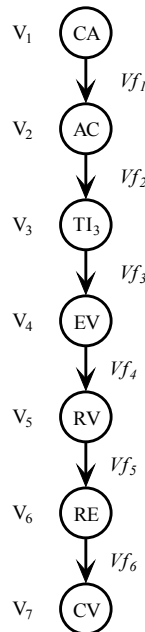


Figure 6: Evaluation de la configuration non exécutée

Chaque valuation d'un sommet est un vecteur de la forme :

$$V_i \begin{cases} TI_i \text{ Type de compression si pertinent} \\ n_i \text{ fréquence} \\ t_i \text{ taux de pertes} \end{cases} \begin{cases} \text{caractéristiques intrinsèques} \\ \text{caractéristiques contextuelles} \end{cases}$$

Pour les arcs, les vecteurs ne décrivent que les caractéristiques contextuelles :

$$Vf_j \begin{cases} n_j \text{ fréquence} \\ t_j \text{ taux de pertes} \end{cases} \text{caractéristiques contextuelles}$$

L'évaluation des fréquences n_i et des taux de pertes sera différente selon le positionnement du composant ou du flux dans la chaîne et selon qu'il est ou non en cours d'exécution.

En effet, pour les composants en cours d'exécution et les flux qui les relie, les mesures faites sur le réseau suffisent à définir les caractéristiques, nombre d'images par seconde et pertes. Ainsi les vecteurs V_1 , Vf_1 , V_2 et Vf_2 sont définis puisqu'ils concernent des composants ou flux en amont du changement de composant du point de vue de la chaîne de traitement d'image et donc leurs comportements ne devraient pas ou peu changer.

Nous choisissons, dans ce cas simple, de faire la même hypothèse pour les composants et flux en aval du changement de composant. Nous supposons donc que les observations sur le réseau permettront de définir V_4 , Vf_4 , V_5 , Vf_5 , V_6 , Vf_6 et V_7 .

En revanche, pour le composant qui sera créé et le flux qui en est issu, il est nécessaire d'extrapoler pour définir V_3 , Vf_3 . Le document XML décrivant le composant de traitement d'image doit permettre de définir le nombre d'images traitées par seconde. Le flux attaché à ce composant sera évalué à partir de tests effectués sur le réseau. Ces derniers permettront d'évaluer s'il existe une marge permettant d'envoyer plus de données ou si le réseau est saturé. Dans le premier cas, le flux sera considéré comme équivalent au flux issu du composant TI_2 . Dans le cas contraire, on utilisera les caractéristiques de délai et de perte du flux pour définir les nouvelles valeurs du nombre d'images par seconde et du taux de perte pour ce composant.

L'étape suivante de l'évaluation permet de noter les caractéristiques de l'assemblage.

La configuration étudiée est ici celle contenant le composant TI_3 , la note du critère intrinsèque est donc $In=0$.

Deux caractéristiques contextuelles sont donc à évaluer : le nombre d'images par seconde n et le taux de perte t et leurs notes respectives Ch_A et Ch_B .

$$Co = \frac{2.Ch_A + 1.Ch_B}{2 + 1}$$

Pour évaluer le nombre d'images par seconde, n , on dispose des différentes fréquences n_i des composants et flux appartenant au chemin étudié. Ce nombre d'images est imposé par le composant ou flux le plus lent donc $n = \min(n_i)$. Cette évaluation permet de déterminer la note Ch_A .

Pour évaluer le taux de perte, t , on considère les taux de pertes t_i des composants et flux du chemin étudié. Le

taux global est alors le produit des taux de pertes de chaque composante du chemin : $t = \prod_{i=1}^{13} t_i$. Une fois t

déterminé, on peut définir la note Ch_B et calculer Co .

La qualité de service est alors entièrement déterminée par les deux valeurs $QoS = (In, Co)$.

Conclusion

Pour une application répartie, il existe différentes façons d'obtenir la qualité de service désirée par l'utilisateur. Ainsi, les méthodes de négociation et de réservation de ressources apportent des solutions lorsque l'environnement est stable et qu'il est possible de définir a priori une qualité de service minimale du réseau. En

revanche, pour les applications grand public utilisant Internet et du matériel hétérogène, ces méthodes ne sont pas applicables. Il est alors nécessaire de proposer des solutions dynamiques capables de s'adapter en temps réel au contexte.

Nous réalisons cette adaptation par une restructuration de l'application par la plate-forme qui la supervise dont l'objectif est de fournir à l'utilisateur le service le plus adapté à ses vœux. Nous avons défini précédemment une méthode de détermination des différentes compositions possibles de l'application en fonction des composants logiciels disponibles. Toute la difficulté réside alors à évaluer la qualité de service des différentes configurations possibles.

En effet, la qualité de service d'un système complexe ne peut être la somme des qualités de service offertes par chacun de ses composants tant il est vrai que deux composants de haute qualité peuvent donner, une fois assemblés, un mauvais résultat. De plus, l'application étant structurée en assemblages de composant, le problème central est l'évaluation de la qualité de service d'un tel assemblage.

Notre propos a d'abord consisté à définir le modèle de qualité de service que nous allons utiliser. Celle-ci doit en effet tenir compte des caractéristiques de service dépendant du contexte et de celles qui n'en dépendent pas. Nous avons donc abouti à une représentation graphique sous forme de points dans un plan afin de refléter l'indépendance de ces deux critères.

Nous avons ensuite proposé une méthode d'évaluation de la qualité de service permettant en particulier l'évaluation d'un assemblage de composants logiciels non actuellement utilisés. En effet, la plate-forme ne dispose pas, dans ce cas, de mesure directe des flux d'entrée et de sortie. En revanche, à l'aide d'un graphe représentant les composants et les flux qu'ils échangent, la plate-forme peut évaluer la qualité de service en se basant sur les informations dont elle dispose, qui décrivent les composants et les flux, et sur des mesures réalisées sur le réseau.

Si elle permet de contrôler et d'ajuster la qualité du service offert en cours d'utilisation de l'application, l'approche que nous préconisons ne peut cependant fonctionner que si la qualité de service a été prise en compte à tous les niveaux de conception et de réalisation. C'est pourquoi nos travaux portent actuellement sur la définition de modèles de composants intégrant la qualité de service mais aussi sur l'étude d'une méthode de conception permettant de structurer l'application autour des concepts de fonctionnalités et de services. Cette méthode doit aussi aboutir à l'élaboration de documents descriptifs que la plate-forme utilisera ensuite pour assurer sa tâche de supervision.

Bibliographie

- [1] C. Chassagne: Qualité de Service dans l'Internet; CNRS-UREC, 1997
- [2] Comité Consultatif International Télégraphique et Téléphonique : General Aspects of Quality of Service and Network Performance in Digital Networks ; Rec.I.350, International Telecommunication Union, Geneva, 1989.
- [3] M.Dalmai, P. Roose, F.Luthon : A software architecture for component based multimedia applications ; 6th international conference on Integrated Design and Process Technology (IDPT 2002), 23-24 June 2002 Pasadena (California) , USA,- ISSN 1090-9389 (8 pages)
- [4] A.Hafid, G. von Bochmann, R. Dssouli : Distributed Multimedia Application and Quality of Service : A Review ; *Electronic Journal on Networks and Distributed Processing*, n°6, 1998, p. 1-50
- [5] S. Laplace : Architecture logicielle intégrant la qualité de service pour les applications multimédias réparties reconfigurables ; Mémoire de D.E.A., Université du Maine, 2002
- [6] S.Laplace, M.Dalmai, P.Roose : Quality of Service in distributed multimedia applications. A formal approach ; poster, MMNS 2003, 6th IFIP/IEEE International Conference on Management of Multimedia Networks and Services, 7th, 10th September 2003, Queen's University of Belfast, Northern Ireland

- [7] P.Roose, M. Dalmau, F. Luthon : A distributed Architecture for Cooperative and Adaptative Multimedia Applications ; 26th Annual International Computer Software and Application Conference (COMPSAC 2002), IEEE Computer Society Press, 26-29 August 2002, Oxford, England , ISBN : 0-7695-1727-7, ISSN : 0730-3157, pp. 444-449
- [8] Sun Microsystems :, Java Media Framework API Guide Version 2.0 ; novembre 1999.
- [9] A. Vogel, B. Kerherve, G. Bochmann, GECEL., : Distributed multimedia applications and Quality of Service :- A Survey ; IEEE Mutimedia Journal, august 1995