

Modèles de données multimédia pour l'architecture client/serveur

Soudougou M. KONATÉ et Cao Lieu NGUYEN
Université du Québec à Montréal, Canada.

Résumé

Cet article analyse les modèles de données multimédia proposés récemment et leur application à l'architecture client/serveur, un type de système réparti de plus en plus répandu. Les modèles de données multimédia à base d'intervalles de temps sont d'abord examinés. Puis leur application à l'architecture client/serveur est analysée.

Mots clés: Multimédia, modèles de données, client/serveur, synchronisation.

1. Introduction

Ces dernières années ont vu croître l'intérêt pour les *systèmes d'information multimédia (SIM)*, systèmes réalisant *l'intégration du texte, de l'image, du son et de la vidéo*. Mais avant de réaliser pleinement les bénéfices attendus, les SIM doivent relever plusieurs défis liés à l'acquisition, la compression et le stockage des données multimédia, à l'accès à ces données, et à leur présentation [1],[2]. Un des problèmes à résoudre concerne la gestion de la composante temporelle de ces types de données. Cette composante, caractéristique importante des images animées et du son, est essentielle à la synchronisation des objets au moment de leur présentation à l'utilisateur, surtout dans le cas d'un système distribué sur un réseau téléinformatique[1],[3]. Aussi, doit-on pouvoir stocker la composante temporelle des données multimédia et y accéder au même titre que les autres attributs. Pour ce faire, il faut des modèles appropriés à la manipulation des données temporelles tant au niveau conceptuel qu'aux niveaux logique et physique. Cet article présente les résultats de l'analyse des récents modèles proposés à cet effet et leur application à l'architecture client/serveur.

2. Modèles à base d'intervalles de temps

Les modèles à base d'intervalles de temps proposés par Little et Ghafoor [2] permettent de représenter et manipuler les composantes spatiales et temporelles des données multimédia. Ces modèles utilisent une structure de réseaux de Pétri dont les noeuds sont des intervalles temporels. En spécifiant ces intervalles les uns par rapport aux autres, on parle de *relations temporelles*. Ces relations temporelles, pouvant être atomiques (exemple une image) ou composés (exemple un clip vidéo) sont utilisées pour modéliser les présentations multimédia avec la convention que chaque intervalle représente un élément multimédia. On élabore ainsi un modèle conceptuel arborescent appelé *schéma de synchronisation*. Ce schéma est ensuite unifié au niveau logique avec celui d'un SGBD hôte de type hiérarchique, réseau, relationnel ou orienté-objet.

3. Application des modèles temporels à l'architecture client/serveur

Dans le paradigme client/serveur, un ou plusieurs systèmes clients et un ou plusieurs systèmes serveurs mettent en commun leurs ressources pour former un système composite permettant du traitement partagé [4]. Un client est un processus interagissant avec l'utilisateur, tandis qu'un serveur est un processus ou un ensemble de processus résidant dans une machine et coopérant pour donner du service aux clients. Dans un tel système, certaines tâches sont conçues pour être exécutées par le serveur et d'autres, par le client. Cette division des tâches obéit aux contraintes imposées par l'application et par le réseau téléinformatique. La distribution des tâches étant préétablie, elle est donc plus facilement contrôlable que dans le cas d'un système distribué quelconque.

Les modèles de données présentés plus haut ont été analysés pour leur capacité à faciliter le stockage des informations sur le serveur, l'accès à ces données et la synchronisation des média en tenant compte de la transmission des données entre le serveur et le client. Dans ces modèles, le stockage des données est spécifié dans les noeuds terminaux (feuilles) du modèle de synchronisation. C'est dans ces noeuds que l'on a les informations concernant le type de données et le nom du fichier. On suppose l'existence d'un mécanisme (par exemple des pointeurs) faisant le lien entre ces noeuds terminaux et l'emplacement physique des objets multimédia. Les informations temporelles et de liens entre les objets multimédia sont stockées dans les noeuds intermédiaires. Cette structure s'adapte très bien à un environnement client/serveur. Les différents types de noeuds seront stockés au niveau du serveur. Ensuite, on accède à ces données en utilisant les algorithmes d'accès appropriés [2]. Ces algorithmes, conçus pour un système multimédia centralisé, s'appliquent dans un environnement client/serveur avec quelques aménagements.

En effet, il faut tenir compte de la division des tâches entre le serveur et le client et des contraintes de communication. C'est le processus serveur qui réalisera la recherche proprement dite en exécutant l'algorithme d'accès proposé par Little et Ghafoor. Cet algorithme réalise la traversée du schéma de synchronisation pour atteindre les noeuds concernés par la recherche. Une fois ces noeuds identifiés, on y accède physiquement en se servant des méthodes d'accès du SGBD hôte. Les tâches pré-recherches (formulation des requêtes, optimisation, adressage) et post-recherches (composition spatiale et temporelle, présentation) incombent au client. Il faut aussi tenir compte du temps de communication des requêtes entre le client et le serveur et surtout du temps de communication des résultats du serveur jusqu'au client. Selon la qualité de services offerts par le réseau téléinformatique sous-jacent, ce temps peut avoir un impact majeur sur la qualité de la présentation finale[1],[3]. Dans l'algorithme d'accès proposé par Little et Ghafoor[2], aucune attention spécifique n'est faite à ce problème. Deux solutions sont envisageables pour utiliser cet algorithme d'accès dans un contexte client/serveur:

- la première est une approche optimiste qui considère que le poste de travail sur lequel le client opère a toutes les ressources (espace disque, mémoire...) pour assurer la présentation complète. Dans cette perspective, aucun changement n'est requis au niveau de l'algorithme d'accès proposé;
- la deuxième approche en est une de planification à temps réel dans laquelle le client effectue la présentation au fur et à mesure que les résultats lui parviennent du serveur. Dans ce cas, il faut prévoir un mécanisme visant à compenser les délais de latence dus au réseau téléinformatique. C'est à ce niveau que la qualité de service joue un rôle important.

4. Conclusion

Nous avons présenté et expliqué les modèles de données multimédia à base d'intervalles de temps, principalement ceux proposés par Little et Ghafoor[2]. Ensuite l'architecture client/serveur, type particulier d'architecture de systèmes distribués a été présenté et expliqué. Ces deux étapes nous ont permis d'analyser l'application des modèles de données multimédia à base d'intervalles de temps à l'architecture client/serveur en vue d'identifier les problèmes potentiels à appliquer ces modèles à cet environnement particulier. Fondamentalement, les modèles examinés sont applicables dans un environnement client/serveur, mais un certain nombre de points à considérer ont été relevés. Ces points incluent le partage des tâches entre le processus client et le processus serveur, et la considération du délai de communication des données multimédia entre le serveur et le client, dans les algorithmes d'accès. Dans le cas d'une présentation en temps-réel, ce délai a un impact important, il faut donc en tenir compte dans le modèle de synchronisation.

Références

- [1] Thomas D. C. Little, Arif Ghafoor: "Network Considerations for Distributed Multimedia Object Composition and Communication". *IEEE Network Magazine*, November 1990, pp. 32-49.
- [2] Thomas D. C. Little, Arif Ghafoor: "Interval-Based Conceptual Models for Time-Dependent Multimedia Data". *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5, No 4, August 1993, pp. 551-563.
- [3] Thomas D. C. Little, Arif Ghafoor: "Spatio-Temporal Composition of Distributed Multimedia Objects for Value-Added Networks". *Computer*, October 1991, pp. 42-50.
- [4] Alok Sinha: "Client-Server Computing". *Communications of the ACM*, July 1992, Vol.35, No. 7, pp. 77-98.