Optimisation sémantique des requêtes continues Application aux bâtiments intelligents

Manel Charfi

Optimisation sémantique des requêtes continues Application aux bâtiments intelligents

Manel Charfi*#
*Laboratoire LIRIS, INSA-Lyon, Université de Lyon - CNRS (F)
#Laboratoire LCIS, IUT de Valence, Université de Grenoble (F)
manel.charfi@liris.cnrs.fr
Bourse financée par la région Rhône-Alpes
Début : Octobre 2013

1. CONTEXTE

Les bâtiments dits "intelligents" apparaissent aujourd'hui comme une réponse prometteuse aux enjeux sociétaux liés au bien-être des occupants et à la consommation énergétique, et plus globalement au développement durable, dans le secteur du bâtiment. Ceci explique l'apparition de nombreux projets dans ce domaine, dont plusieurs se sont intéressés à la réduction de la consommation des bâtiments (énergie primaire, fluides...). Un des enjeux actuels est de pouvoir pleinement exploiter les données issues des systèmes de contrôle des bâtiments intelligents afin de fournir des services de qualité aux occupants et aux gestionnaires des bâtiments.

Une maison intelligente peut posséder diverses fonctions selon les habitants et leurs besoins. Une telle maison contient des capteurs (température, luminosité, présence, ...) et des actionneurs (alarme, mise en marche/arrêt chauffage, ...) qui lui permettent respectivement de récupérer les données (par exemple au moyen de requêtes continues à partir des flux de données définis dans le système) et d'agir selon la situation détectée afin de réaliser un objectif donné, par exemple assurer que la température moyenne du logement d'un locataire donné ne descend pas au dessous d'un seuil qu'il aura fixé. Nous nous intéressons à l'optimisation énergétiques d'un bâtiment collectif, avec par exemple 50 locataires. Puisque chaque locataire peut définir ses préférences, nous voulons aborder le problème d'optimisation tout en satisfaisant les préférences des utilisateurs. Nous nous intéressons à un type particulier de requêtes que nous appelons requêtes de monitoring : il s'agit des requêtes continues qui durent très longtemps, aussi longtemps que dure le contrat entre un résident et le gestionnaire du bâtiment (par exemple 3 ans). Ces requêtes, qui n'ont que quelques tuples mais de nombreux effets de bord sur le bâtiment, doivent respecter un contrat donné défini par l'utilisateur qui pourrait à la fois définir ses seuils (température maximale, humidité tolérée, pièce favorite...) et les approximations qui lui conviennent (fréquence d'acquisition de la température, temps de passivité du système...)

Considérons un scénario jouant autour du locataire Bob. Par exemple, pour Bob, il faut que la température à l'intérieur de la maison ne soit pas inférieure à 15°C tout au long de son contrat de location qui dure 3 ans. Une telle contrainte n'est pas une demande qui nécessite une notification en temps réel : une notification par an pourrait être suffisante de la part du gestionnaire du bâtiment. On voit bien que, le système doit interagir avec les actionneurs pour respecter le contrat défini par l'utilisateur.

Ainsi, en cas diminution de la chaleur, le système peut fermer les fenêtres, mettre en marche le système de chauffage, ouvrir les stores... Il est aussi possible de notifier l'utilisateur afin de demander son intervention manuelle pour l'ouverture/la fermeture des stores ou des fenêtres, partant de l'idée que la notification est moins coûteuse que l'activation du levier de fermeture de la fenêtre et que Bob accepte de fermer les fenêtres.

Bob peut aussi décider que la température de chaque chambre à une seconde donnée est la même que sa température lors de la dernière heure. Grâce à une telle approximation, il sera possible d'éviter certaines pratiques courantes dans les bâtiments intelligents qui surchargent le réseau et consomment de l'énergie. En fait, il pourrait être inutile d'user les batteries des capteurs avec une telle fréquence si le service demandé par Bob ne nécessite pas une valeur du capteur chaque seconde.

2. SOLUTIONS EXISTANTES

Revenons à l'approximation définie par Bob : la température de chaque chambre à une seconde donnée est la même que sa température lors de la dernière heure. Cette approximation, qui ne tient évidemment pas dans la pratique, peut soulager la charge de travail du réseau et préserver la batterie de capteurs. La question est : comment arriver à exprimer une telle approximation afin de pouvoir l'utiliser en tant qu'une optimisation exploitable par notre système ?

Cela nous rappelle la technique de délestage ("load shedding") [7] dans les systèmes de gestion de flux, qui interfère généralement dans le plan physique de la requête. Dans notre cas, nous voulons réécrire un ensemble de requêtes continues au niveau logique à l'aide des contraintes exprimées par chaque habitant.

Dans le cadre relationnel classique, le problème est connu comme l'*optimisation sémantique des requêtes* et date des années 80 [4]. L'idée est d'exploiter une certaine information sémantique exprimée par l'utilisateur pour rendre la

requête plus efficace. A notre connaissance, l'optimisation sémantique pour les requêtes continues n'a pas été encore étudiée. Les méthodes déployées pour optimiser les requêtes sémantiquement avaient recours aux contraintes d'intégrité utilisées lors de la conception et la normalisation des schémas de la base de données. L'utilisation de ces contraintes servait à détecter les anomalies présentes dans les requêtes [3] et de mettre en place des techniques de réécriture, telle que l'ajout/la suppression des jointures/sélections [6, 5], des requêtes en question.

3. SOLUTION À APPORTER

Nous voulons intégrer les approximations de l'utilisateur dans les requêtes continues existantes dans notre bâtiment intelligent. Ces approximations seront vues comme des nouvelles contraintes définies par l'utilisateur que le système doit comprendre et intégrer dans son processus d'optimisation des requêtes continues. Il s'agit donc d'utiliser de nouvelles contraintes qui peuvent varier selon l'utilisateur et le temps contrairement aux techniques utilisées dans l'optimisation sémantique des requêtes classiques où on réutilise les contraintes statiques qui ont servi à la conception et la modélisation de la base de données. Il est alors nécessaire de trouver le bon modèle permettant de les définir et facilitant leur intégration dans le processus de l'optimisation.

La première idée simple consiste à exprimer les approximations de l'utilisateur en utilisant les dépendances fonctionnelles (DF) [2], voire les dépendances fonctionnelles conditionnelles (DFC) [1] ou plus souvent les dépendances temporelles (DFT) [8, 9]. Par exemple l'approximation de Bob pourrait être représentée par la DFT $room \rightarrow_{hour} temperature$: selon cette dépendance, la température d'une pièce donnée pendant une heure ne change pas.

La définition de notre problème devient : Étant donné un ensemble M de requêtes de monitoring et un ensemble F de dépendances (DF, DFC, DFT ou autres) : comment réécrire M en un ensemble de requêtes de monitoring M' tel que M' est équivalente à M par rapport à F.

Notre travail en cours consiste à :

1. Modéliser les approximations de l'utilisateur, les flux de données et les requêtes continues,
2. Proposer des règles de réécritures logiques pour les requêtes continues en présence de dépendances,
3. Choisir un système de gestion de flux de données (et les requêtes continues associées),
4. Implémenter un algorithme de réécriture,
5. Réaliser des tests.

4. REFERENCES

[1] Philip Bohannon, Wenfei Fan, Floris Geerts, Xibei Jia, and Anastasios Kementsietsidis. Conditional functional dependencies for data cleaning. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 746–755. IEEE, 2007.

[2] R. Fagin. Functional dependencies in a relational database and propositional logic. *IBM Journal of Research and Development*, 21(6) :534–544, Nov 1977.

[3] Bryan Howard Genet. *Is Semantic Query Optimization Worthwhile ?* PhD thesis, The University of Waikato, 2007.

[4] Michael Hammer and Stanley B Zdonik. Knowledge-based query processing. In *Proceedings of the sixth international conference on Very Large Data Bases-Volume 6*, pages 137–147. VLDB Endowment, 1980.

[5] Barry GT Lowden and Jerome Robinson. Constructing inter-relational rules for semantic query optimisation. In *Database and Expert Systems Applications*, pages 587–596. Springer, 2002.

[6] Sreekumar T. Shenoy and Z. Meral Ozsoyoglu. A system for semantic query optimization. *SIGMOD Rec.*, 16(3) :181–195, December 1987.

[7] Nesime Tatbul, Uğur Çetintemel, Stan Zdonik, Mitch Cherniack, and Michael Stonebraker. Load shedding in a data stream manager. In *Proceedings of the 29th international conference on Very large data bases-Volume 29*, pages 309–320. VLDB Endowment, 2003.

[8] X Sean Wang, Claudio Bettini, Alexander Brodsky, and Sushil Jajodia. Logical design for temporal databases with multiple granularities. *ACM Transactions on Database Systems (TODS)*, 22(2) :115–170, 1997.

[9] Jef Wijsen. Temporal fds on complex objects. *ACM Transactions on Database Systems (TODS)*, 24(1) :127–176, 1999.