Dynamic Source Selection to Handle Changes of User's Interest in Continuous Query

Kosuke Ohki[1], Yousuke Watanabe[2], and Hiroyuki Kitagawa[1,3]

[1] Graduate School of Systems and Information Engineering, University of Tsukuba
[2] Global Scientific Information and Computing Center, Tokyo Institute of Technology
[3] Center for Computational Sciences, University of Tsukuba
ohki@kde.cs.tsukuba.ac.jp, watanabe@de.cs.titech.ac.jp,
kitagawa@cs.tsukuba.ac.jp

1 Introduction

The volume of stream data delivered from information sources has been increasing. A demand for efficient processing of stream data has become more and more important. Stream processing systems [1] can continuously process stream data according to user's requests. A request is usually specified as a continuous query written in SQL-like language.

In conventional frameworks, the user must specify information sources in advance, and the user cannot change information sources during query processing. However, there are many cases in which target information sources the user is most interested in change over time. We therefore need an additional framework to deal with changes of the target information sources for the same query.

We propose dynamic source selection for this purpose. Our contributions are as follows: (1) **Proposal of ASSIGN operator to switch information sources dynamically**, (2) **Development of a connection management scheme to reduce connections to unnecessary information sources**, (3) **Implementation of the proposed framework on our stream processing system**, and (4) **Experiment to measure performance in stream environments**.

2 Dynamic Source Selection

We consider an application to track video data of moving objects as a sample case in which target information sources change over time. We assume many network cameras, a database that contains information on locations of the cameras, and location sensors attached to moving objects.

The query processing in our proposed framework is illustrated in Fig.1. When the system receives the current location of the target object "A", it finds IDs of cameras near the target "A". In this example, the system is receiving a tuple with the camera ID "Camera1". The connection management unit makes a connection to "Camera1" to obtain its video data. The ASSIGN operator then selects "Camera1" and gets the video data from "Video" in information sources

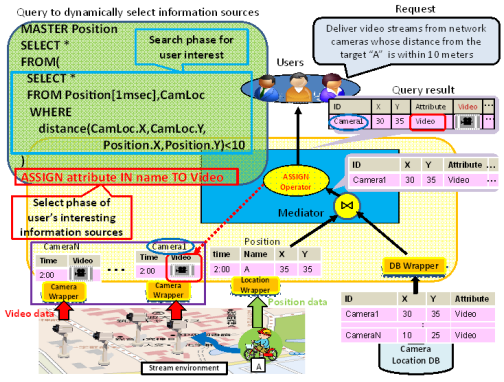

Fig. 1. Tracking application in the proposed framework to dynamically select target information sources

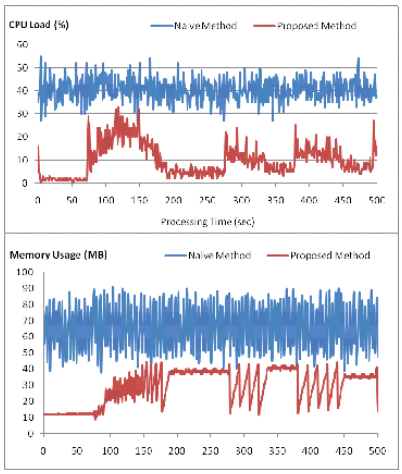

Fig. 2. CPU Load and Memory usage in comparison of naive and proposed methods

"Camera1". The ASSIGN operator outputs a tuple attached with the "Video" attribute. The system repeats this process every time the current location of the target object arrives.

We have implemented the proposed framework in our stream processing system called as StreamSpinner [2].

3 Experiment

We prepared ten cameras and location data of a thousand people in the virtual environment. We compare system loads of the proposed method and a naive method. The naive method here means that the system makes connections to all running network cameras. The result is Fig.2. We can see our proposed method contributes to reduce system resource consumptions.

Acknowledgments

This research has been supported in part by Japan Science and Technology Agency (CREST), the Grant-in-Aid for Scientific Research from JSPS (#1820005) and MEXT (#19024006).

References

1. D. Abadi, et al. "Aurora: a new model and architecture for data stream management ", VLDB Journal Vol.12,No2,pp.120-139,2003.
2. http://www.streamspinner.org/