

TomuDB: Multi-Resolution Queries in Heterogeneous Sensor Networks through Overlay Network

Yoh Shiraishi^{†,a} Niwat Thepvilojanapong^{‡,b} Yosuke Tamura^{†,c} Tatsuro Endo^{†,c}
 Koichi Yamada^{†,d} Nayuta Ishii^{†,e} Hiroki Ishizuka^{†,e} Keisuke Kanai^{§,e} Yoshito Tobe^{†,f}
[†]CREST, JST, Japan [‡]KDDI R&D Laboratories Inc., Japan [§]Tokyo Denki University, Japan
^asiraisi@csis.u-tokyo.ac.jp ^bwat@acm.org ^c{tamura, endo}@fixstars.com
^dyamada@im.dendai.ac.jp ^e{nayuta, isi, ksk}@u-netlab.jp ^fyoshito_tobe@osoite.jp

Categories and Subject Descriptors: H.2.4 [Database Management]: Systems—*query processing*

General Terms: Design, Performance

Keywords: Multi-resolution queries, overlay sensor network

1 Introduction

Querying in heterogeneous sensor networks is a challenging research issue due to a variety of real-world queries depending on users' preferences. Examples of queries are weather, nearby restaurants, navigation, etc. Users may ask for a breezy path starting from distinct points to distinct train stations. Sensing data collected from wide areas (city or country level) are needed to provide the real-world search as a service for users. Therefore, another challenge is a collaboration of heterogeneous sensor networks because deploying sensors in wide areas is impractical. SensorMap [1] collects sensing data from independent sensor networks by allowing data owners to publish their data on the web. Sensing data from all owners are sent to a centric storage in SensorMap. However, distributed data storage is likely to be an efficient solution comparing to a centric storage in which bottleneck always occurs.

In this paper, we propose to use distributed approach, i.e., sensing data are stored in a sink of each sensor networks and overlay network is constructed amongst such sinks. In order to cope with real-world searches, heterogeneous sensor networks work collaboratively by forwarding queries to another network (sink) that is likely to answer the queries. Each query includes the requirement of resolution. If collected sensing data do not meet the required resolution, interpolation is used to calculate approximate results from coarse resolution. Interpolation is also applied, if sensing data on requested area do not exist.

2 TomuDB Architecture

We propose a TomuDB architecture for constructing multi-resolution storage and query processing among heterogeneous sensor networks through an overlay network. The

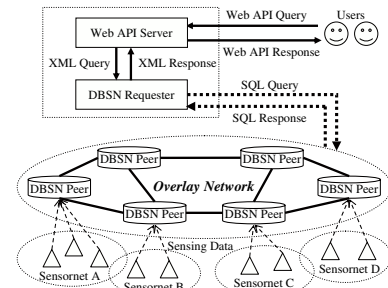


Figure 1. TomuDB architecture.

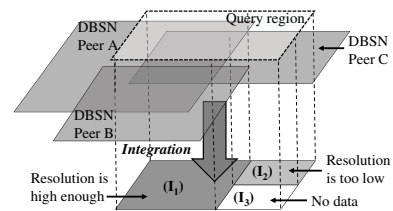


Figure 2. Sensing data from three DBSN peers are integrated into the lowest plane.

architecture composes of the following components (Fig. 1).

Web API Server: Users submit a human-readable query to an intelligent web API server, which in turn converts the query into XML format, and then forwards the XML query to a DBSN (Database Sensor Network) requester. An XML response is converted into human-readable format before sending to the users.

DBSN Requester: The DBSN requester converts an XML query into SQL format so that data retrieval can be executed efficiently. It also converts a SQL response into XML format before sending the response back to the web API server.

DBSN Peer: The basic task of a DBSN peer is to collect sensing data from sensors (or DBSN sources) existing in its network. Each DBSN peer connects to other DBSN peers by using an overlay network. In addition to data collection, DBSN peers are also responsible to answer SQL queries.

DBSN Source: DBSN sources are resource-constrained sensors. The only task of DBSN source is to report its sensing

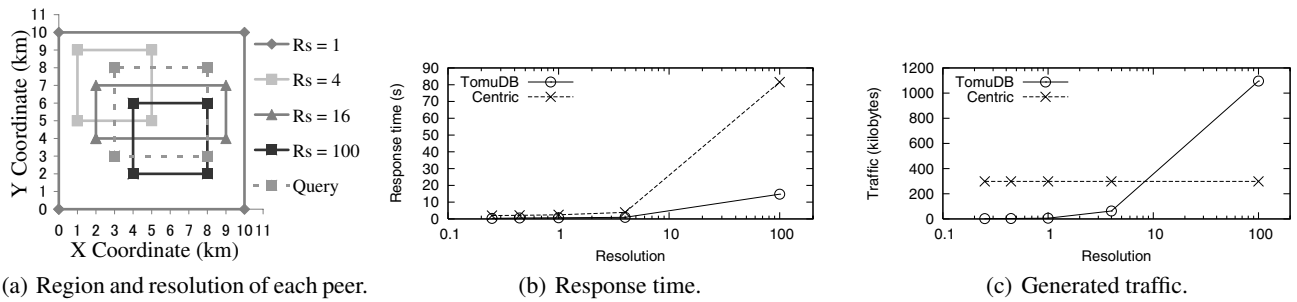


Figure 3. Experiment setup and results.

values to the DBSN peer.

3 Multi-Resolution Queries

Each DBSN peer collects sensing data from all DBSN sources associated with it, and stores data locally in its memory. Hence, each DBSN peer can provide sensing data of limited region. Based on collected data, DBSN peers know resolutions of sensing data in their region. As a result, each DBSN peer keeps the followings in its *peer information table*: DBSN peer ID, sensor type (temperature, light, etc.), region and spatial resolution (R_s). A DBSN peer always exchanges the table with other peers.

Without loss of generality, we assume the region associated with each DBSN peer is rectangular (Fig. 2). A query composes of data type, region, and resolution. As illustrated in Fig. 2, there are three DBSN peers whose regions overlap the query region (the upmost plane). The lowest plane in the figure is a result of integrating data from those three DBSN peers. The results of integration can be divided into three possible cases: (i) resolution of overlapped region is equal or higher than query resolution (I_1); (ii) resolution of overlapped region is lower than query resolution (I_2); (iii) some parts of query region does not overlap any DBSN peers (I_3).

Because TomuDB is a distributed database, users can send a SQL query to any DBSN peer. The query will be forwarded to an *integrator* which is responsible to integrate responses from other DBSN peers. In short, the integrator is a DBSN peer whose region overlaps the largest part of query region. The detailed processes of selecting the integrator are as follows. First, the query region is enlarged to an *extended query region* because we need data of nearby regions for interpolation purpose. Then, the extended query region is divided into small square regions. The query region is also divided into sub-regions based on the boundaries of the overlapped DBSN peer where an owner (DBSN peer) will be chosen to provide a response corresponding to each sub-region. A DBSN peer which is responsible for the most number of sub-regions is the integrator. Three rules are applied when determining the owner of each sub-region. First, if there is only one DBSN peer whose region includes a sub-region, it is the owner. Second, if regions of multiple DBSN peers overlap a sub-region, DBSN peers whose resolution are higher than query resolution are candidates. The owner is a DBSN peer whose resolution is the lowest amongst candidates. If the candidate does not exist, a DBSN peer whose resolution is the nearest to the query resolution is the owner. Third, if a

DBSN peers whose region overlaps the sub-region does not exist, that sub-region does not have an owner.

The integrator sends queries for interpolation to all owners. When owners receive sub-queries, they send responses back to the integrator. If the resolution of owner does not match, *Inverse-Distance Weighting (IDW)* [2] is employed as an interpolation technique. After the integrator receives all responses, it composes a final response. IDW is also used to determine the value of square whose owner does not exist.

4 Performance Evaluation

Sensing data in the simulations are prepared from the AMeDAS which is a surface observation network developed by the Japan Meteorological Agency used for gathering regional weather data throughout Japan. The region and spatial resolution of DBSN peers, and query region are shown in Fig. 3(a). We compared TomuDB with an on-demand centric approach, where corresponding data are sent to a central server on-demand and response processing is executed at the central server. Performance metrics are response time and generated traffic. Response time is the period observed between transmitting a query and receiving a response. Generated traffic is the total bytes of all generated packets.

Fig. 3(b) and 3(c) show the results of experiments. The spatial resolution of query is set to 0.25, 0.44, 1, 4, and 100. The response time of TomuDB was faster than the centric approach for all resolutions. Finer resolutions need a large number of data and interpolation. While centric approach performs interpolation one-by-one, each DBSN peer in TomuDB interpolates data in parallel. As a result, the centric processes much slower than TomuDB. Traffic generated by the centric approach is constant because raw sensor data are sent to the central server. In contrast, DBSN peers in TomuDB interpolate data before sending responses so that packet size is smaller than raw data. However, traffic increases abruptly at resolution of 100 because a large number of data are requested among DBSN peers for interpolation.

5 References

- [1] S. Nath et al. Sensormap: a web site for sensors worldwide. In *Proceedings of SenSys'06*, pp. 373–374, Nov. 2006.
- [2] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 23rd ACM National Conference*, pp. 517–524, Aug. 1968.