To Do or Not to Do: Metadata-guided Query Evaluation in Content Caching Networks

Hui(Wendy) Wang*, Ruilin Liu*, Xiuyuan Zheng†, Yingying Chen†, Hongbo Liu†
*Dept. of Computer Science
Stevens Institute of Technology
Hoboken, NJ 07030
hwang@cs.stevens.edu

† Dept. of Electrical and Computer Engineering
Stevens Institute of Technology
Hoboken, NJ 07030
{*rliu3, xzheng1, yingying.chen, hliu3*}*@stevens.edu*

***Abstract*—To support emerging pervasive computer applications, efficient data access and information sharing is essential in mobile wireless environments. The data-centric storage approach provides energy-efficient data dissemination and organization. However, wireless devices have limited storage and are energy-constrained. To address these issues, we introduce the concept of content caching networks, in which the collected data will be stored by its content in a distributed manner, while the data in the network is cached for a certain period of time before it is sent to a centralized storage space for backup. Further, we propose the metadata-guided query evaluation approach to achieve query efficiency in content caching networks. Our clustering and compression based algorithm for metadata construction helps to minimize the information loss due to data compression, while achieving the memory requirements on wireless devices. We present both theoretical and empirical results to show that our metadata-guided approach is highly effective to perform efficient data queries, and thus demonstrate the feasibility of the content caching networks.**

I. Introduction

With the advancement of wireless technologies, wireless devices are blended into our daily life and spend much time with us when we are working, attending meetings, participating in classes, or socializing. We anticipate that these advances will continue, leading to a world where continuous wireless connectivity will support the collection, storage and sharing of information – thereby driving pervasive computing applications. Further, the increasing sensing capability on wireless devices (e.g., smart phones and bluetooth devices) supports the data-centric nature of the collected information. In data-centric storage, the collected data is stored by attributes or types (e.g., geographic location and event type) at nodes within the network [1]–[3]. Queries for data with a particular attribute will be sent directly to the relevant node(s) instead of performing flooding throughout the network, therefore, data-centric approach enables efficient data dissemination/access.

However, comparing to centralized servers, wireless devices have limited storage and are energy-constrained. To ensure efficient data access and information sharing, we introduce the concept of *content caching networks*, where the collected data will be stored by its *content* in a distributed manner, while the data in the network is *cached*, i.e., the data will be stored within the network for a certain period of time before it is sent to a centralized storage space for backup. The advantage of using content caching networks is two-fold. First, storing the data by content enables efficient evaluation of queries commonly raised on the data content, for example, the queries for specific participants, events, or locations. Second, caching enables real-time query evaluation and eliminates the existence of centralized storage that may become a bottleneck query evaluation or a single target for attacks. Further, by uploading data in a lazy fashion (i.e., once in a while), it avoids frequent data transfer from the wireless devices to the centralized storage, and consequently reduces massive battery power consumption and vastly decreases the communication overhead of the network.

A content caching network may consist of a large number of nodes, moving in and out of the area of interest. To support such large-scale and dynamic content caching networks, it is essential to achieve efficient query evaluation. Although there has been research in wireless sensor networks that are related to data-centric storage [1], [2], [4], [5], most of the work focus on *stable* network topology, assuming data dissemination in a predefined manner, thus are not applicable to mobile wireless environments. In this work, we propose to use *metadata* to guide query evaluation so that only the nodes whose data may contribute to query answers will evaluate the queries. Although it is popularly studied in the scope of statistical network (e.g., P2P network) [6], content caching in mobile network with metadata is novel. To the best of our knowledge, how to design the metadata from the collected content with respect to the resource limitation (e.g., memory space) is not studied by any existing content caching work.

Our approach of metadata-guided query evaluation can not only support efficient query evaluation by only returning relevant nodes containing required data, but also providing rich expressiveness to describe various types of data in content caching networks. Thus, using metadata in content caching networks can exchange and integrate the heterogeneous data from various nodes in the network. On the other hand, the introduction of metadata will add additional overhead on the already memory-limited wireless devices. To address this issue, we propose data clustering and compression for metadata construction, and developed *Clustering, Balancing, and Compression (CBC)* algorithm to meet the memory requirement. In the meantime, we minimize the information loss incurred by data compression. We theoretically analyze the effectiveness of CBC algorithm.

To evaluate the effectiveness and efficiency of our approach, we conduct simulation using trajectory data generated from

a mobile wireless network based on a city environment in Germany. By examining two representative networks, 10-node and 50-node, our results show that our metadata-guided query evaluation approach is highly effective; it dramatically improves query evaluation performance with low false positive rate and high precision of query answers.

The rest of the paper is organized as follows. In Section II, we first provide the background of XML and metadata construction requirements, and then present our CBC algorithm. We next provide an analysis of metadata-guided query evaluation in Section III. We present the simulation evaluation of our approach in Section IV. We conclude our work in Section V.

II. Metadata Construction

To efficiently evaluate queries in large-scale and dynamic content caching networks, the metadata should have the following properties to support the data-centric approach:

- Rich expressiveness: Due to the diversity of data on content caching networks, the metadata should be able to describe data of various types.
- Interoperability: To integrate the data from heterogeneous content caching networks, the metadata should enable exchange of data among distributed heterogeneous wireless devices and with other kinds of information systems.
- Efficient processing: To help discover the wireless devices that have the required data, the metadata should support efficient query evaluation.

Unfortunately, most of the existing index mechanisms (e.g., ring-based index [4] and GHT [5]) cannot satisfy all the above requirements. Thus, instead of index, we propose to use metadata. In this study, we choose extensible markup language (XML) as the representation of metadata, due to its advantages of flexibility, self-description and support for information integration.

A. Background

XML is an HTML-like language with an arbitrary number of user-defined tags [7]. An XML dataset is a collection of data values marked with self-defined tags, which are used to describe the semantics of the data values. The flexibility of tag definition in XML makes it possible to built different semantic layers on top of data. Therefore, XML can be tailored to various categories of applications. Further, XML has been widely accepted and used as the standard for information integration and exchange on the Web. These advantages encourage us to use XML as the representation of the metadata in content caching networks. Figure 1 illustrates an example of XML metadata. It describes the spatial and temporal information of the trajectory data that the sensor node stores. In particular, the spatial information is described by the *lx*, *ux*, *ly* and *uy* elements, while the temporal information is described by the *lt* and *ut* elements. We note that the flexibility of XML makes it easy to extend to support various data types. We only use trajectory database as a running example in this paper to explain the basic idea of our approach. Our approach can be applied to other types of data, e.g., user information data.

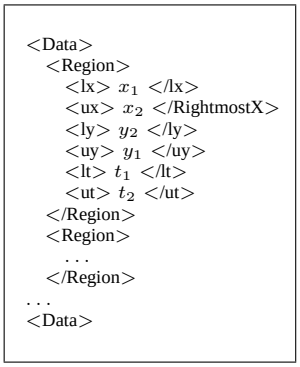


Fig. 1. An example of XML Metadata to represent trajectory data

Answers of XML queries are formalized by using matchings. Informally, a *matching* of a query to the XML metadata is a mapping between the query and the XML data such that both the structural constraints and the value-based constraints in the queries are preserved in XML data. For instance, an XML query "//Region[lx() = x_1]" consists of the structural constraint "//Region[lx]", which specifies the child element *lx* under the element *Region*, and the value-based constraint "lx() = x_1", which specifies the value of the element *lx*. By evaluating this query on the XML data in Figure 1, it returns the first *Region* element. In this study, we consider XPath [8], a node-selecting query language central to most core XML-related technologies. Recent work has shown that XPath queries can be evaluated in polynomial time [9].

B. Requirements and Format

In general, the XML metadata acts as the succinct and complete "summary" of the data. Given a dataset D and the metadata M of D, we say M *conforms* D if for every query Q, if $D(Q)$ (i.e., the answers of evaluating Q on D) is not empty, then $M(Q)$(i.e., the answers of evaluating Q on M) is not empty either. We say D is *relevant* to Q if $M(Q)$ returns non-empty answer. In other words, if there is no answer in $M(Q)$, then D definitely does not satisfy Q. Thus whether there is any answer in $M(Q)$ is the *necessary* condition of whether D has answers to Q. However, it is not the *sufficient* condition; the fact that $M(Q)$ returns non-empty answer does not imply that D must have the answer to Q too. When $M(Q)$ is non-empty but $D(Q)$ is empty, it incurs *false positive*. Thus given the data D, our goal of XML metadata design is two-fold:

- **Requirement 1**: the size of the metadata plus the size of D cannot exceed the available space on sensors, and
- **Requirement 2**: the metadata must conform to D with minimized possibility of false positives.

There exists trade-off between these two requirements; the metadata that is compressed too much (for example, describing all temperatures as a range [-100, 100]) will produce many false positives, while the metadata that is too detailed may be too large and exceed the available memory on wireless devices.

We assume the schema of the metadata is pre-defined based on the prior knowledge of the data that is collected

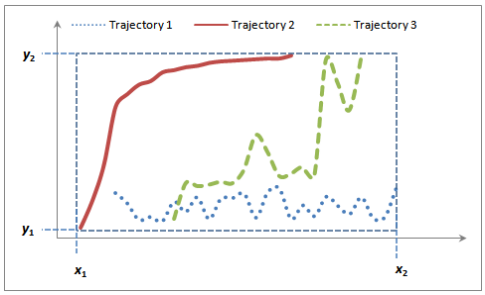


Fig. 2. Illustration of region in metadata versus trajectories in data

in the network. For instance, for the network that collects the trajectories of the mobile devices, assuming in the collected data each trajectory is of the format (x, y, t), where x and y denote the x- and y- coordinates at time point t. Then the collected data on the nodes is a set of trajectories $\mathcal{T} = \{T_i | T_i = ((x_{i_1}, y_{i_1}, t_{i_1}), \ldots, (x_{i_m}, y_{i_m}, t_{i_m}))\}$. Given such set of trajectories, an example of its metadata is shown in Figure 1. Each *Region* element specifies the region, represented with the leftmost and the rightmost $x-$ coordinates (the values x_1 and x_2) as well as the bottom and the top $y-$ coordinates (the values y_1 and y_2), that covers all the trajectories, as well as the range of the timestamps in $\mathcal{T}$ (the values t_1 and t_2). Figure 2 illustrates the relationship between the region in metadata and the trajectories in the data.

C. Algorithm

Given the data D on a wireless device S, both the size of D and the total space N on S are fixed. Thus there exists an upper bound $U = N - |D|$ for the size of the metadata. Then Requirement 1 can be formalized as the following: given the data D and an upper bound U, how to construct the metadata M whose size is no larger than U?

We developed the *CBC* algorithm for metadata construction, which consists of three steps: *Clustering*, *Balancing*, and *Compression*. First, the data is clustered to k groups. Second, the data in each cluster is balanced by transferring. Third, each group is compressed to a single element in the XML metadata. For instance, all three trajectories in Figure 2 are compressed to a single element as shown in Figure 1. Let e be the average size of the elements in the XML metadata. Then the total size of the metadata is $e * k$. Our goal is to make $e * k \leq U$. Due to the fact that e is fixed, we can always fix the value $k = U/e$, i.e., we can meet Requirement 1 by controlling the value of k, the number of clustered groups. The detailed algorithm is presented below.

Step 1 Clustering: To address Requirement 2, we cluster the similar data together in the same group, so that the compressed metadata is as close to the original data D as possible, and the amount of false positives can be minimized. To cluster the data into k groups based on their similarity on trajectory, we use the *k-means* clustering analysis [10]. The *k-means* analysis takes the input parameter k, and partitions a set of n objects into k clusters so that the resulting intracluster similarity is high but the intercluster similarity is low. Using the trajectory data as an example, we define the distance between two trajectories $T_1((x_1, y_1, t_1), \ldots, (x_n, y_n, t_n))$ and $T_2((x'_1, y'_1, t_1, \ldots, (x'_n, y'_n, t_n))$ as the Euclidean distance:

$$Dist(T_1, T_2) = \sum_{i=1}^{n} \sqrt{(x_i - x'_i)^2 + (y_i - y'_i)^2}.$$

The k-means approach clusters the trajectories based on their distances. After clustering, we have k clusters, each containing trajectories that are close to each other. We must note that the distance function in the CBC algorithm can be replaced with other appropriate ones for various types of input data.

Step 2 Balancing: The resulting size of each cluster can be highly unbalanced. For example, some clusters may contain much more trajectories than others. To achieve a balanced compression of metadata, we compare the size of each cluster to the averaged size. If the number of elements in a cluster is smaller than the averaged number, ignore it; whereas if the number of elements is larger than the averaged number, we transfer the excessive number of elements to one or more other clusters by searching for the clusters that are of the least cluster distance to it and with the number of elements smaller than the averaged number.

Step 3 Compression: In this step, each cluster is compressed into an element in the metadata. In particular, for each dimension (i.e., attribute) of the data in the cluster, let its data be $D = \{d_1, \ldots, d_n\}$. We assume that all $d_i (1 \leq i \leq n)$ are of numerical type, which is commonly true in practice. Then D is compressed as a range $[d_{min}, d_{max}]$, where d_{min} and d_{max} are the minimum and maximum value of D. For example, as shown in Figure 1, the leftmost and rightmost $x-$ coordinates are collected as the minimum and maximum value on the $x-$ dimension of the trajectory data. Given the fact that all data values within the same cluster are similar, their generalized ranges must be close to the real values. Thus the possibility of false positive is minimized.

By the data-centric property of the network, all data of the same types/contents are collected on the same device. This enables that the updates on the data still fits the metadata, and thus will not cause much updates on the metadata.

III. Query Evaluation Analysis

In this section, we explain the query evaluation procedure by using metadata, and theoretically analyze the effectiveness of the metadata.

A. Metadata-guided Query Evaluation

The purpose of using metadata is to discover the nodes S that contain the data relevant to queries in content caching networks, so that the queries will not be evaluated on all devices, but only on S. To reach this goal, our query evaluation procedure consists of two steps. We assume the query consists of only keywords.

- **Relevance check on metadata:** the input keyword query Q is translated to the query Q_M on XML metadata. The translation is straightforward; the keywords are defined as the value-based constraints in the queries, with the corresponding structural constraints from the schema. For

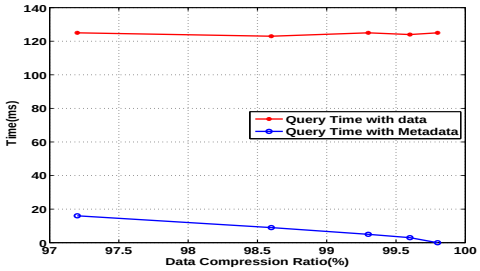


Fig. 3. Comparison of query time on metadata and data itself.

example, for Q that looks for the trajectories containing the coordinates (x, y), it is translated as

Q_M: `//Region[lx()`$\leq x$ `AND ux()`$\geq x$ `AND ly()`$\leq y$ `AND uy()`$\geq y$`],`

which looks for the region that (x, y) falls into. Then we evaluate the query Q_M on the metadata M.

- **Query evaluation on data:** for those nodes whose metadata returns non-empty answer to Q_M, the input keyword query Q is evaluated on their data D. Since the metadata is only a synopsis, it is possible that the returned answers contain false positives. However, as discussed in Section II, since we respect the similarity of data when we construct the metadata, our approach minimizes the possibility of false positives.

B. Analysis of Efficiency

To analyze the effectiveness of using metadata for query evaluation, we consider two scenarios, no metadata is present and the metadata is available, and compare the query evaluation performance in these two scenarios.

When there is no metadata, the query Q will be propagated to the whole network for evaluation. Let n be the total number of sensors in the network, $D_i (1 \leq i \leq n)$ be the size of the data on the i-th sensor node and $t(D_i)$ be the time that Q is evaluated on D_i. The total time of evaluating Q without any metadata is $T_1 = \sum_{i=1}^{n} t(D_i)$.

When there is metadata, the query Q will be first translated to Q_M. Then Q_M will be propagated to the whole network for evaluation. Only the sensors whose metadata satisfies Q_M will evaluate Q on their data to return the final answers. Let m be the number of sensors whose metadata return non-empty answer for Q_M, $M_i (1 \leq i \leq n)$ be the size of the metadata on the i-th sensor node, and $t(M_i)$ be the time that Q_M is evaluated on M_i. The total time of evaluating Q with the presence of metadata is $T_2 = \sum_{i=1}^{n} t(M_i) + \sum_{i=1}^{m} t(D_i)$.

It is straightforward that Q always can be evaluated by scanning D_i once. As shown in Figure 3, our experiments confirm that $t(M_i)$ is always dominated by $t(D_i)$. Therefore, we approximate $t(M_i)$ as a portion of $t(D_i)$, i.e., $t(M_i) = k_i * t(D_i)(k_i < 1)$. We assume the network is of balanced load so that the data on different sensor nodes is of roughly the same size. As the result, all k_is $(1 \leq i \leq n)$ are of similar value. This also applies to $t(D_i)$. Thus we can approximate the ratio of T_2 over T_1 as:

$$T_2/T_1 = k + m/n.$$

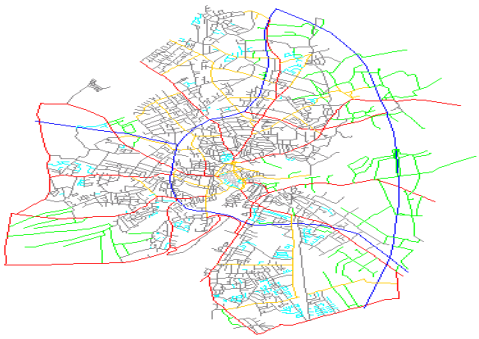


Fig. 4. The experimental data sets are generated based on the city and its vicinity in Germany.

For the worst case that $m \approx n$, T_2/T_1 is always greater than 1. In other words, for those queries whose answers locate on a significant portion of sensor nodes in the networks, evaluating them directly on the data is more efficient than using metadata. However, in practice, most of query answers are stored on only a small number of sensor nodes, i.e., m/n is negligible. Therefore, $T_2/T_1 \approx k$, which is always less than 1. Therefore, in theory, we show that using metadata can achieve more efficient query evaluation than without using metadata.

IV. Experimental Evaluation

In this section, we first describe our experimental methodology and metrics, and then present the results that evaluate the effectiveness of our approach.

A. Methodology

Data Generation. We evaluate the feasibility of our approach using the trajectory data in mobile wireless networks. We note that our approach is generic and can be applied to other types of data as well. We conducted experiments based on mobile nodes generated from a city environment and its vicinity in Germany [11], [12] as shown in Figure 4. The size of the area is $25000m$x$25000m$ and the mobile nodes are moving along the roads in the city randomly at the walking speed(3 feet/sec.).

We collected trajectory data of these mobile nodes for our study. We investigated two networks, one with ten nodes and the other with 50 nodes. To simulate a typical memory-constrained wireless node, we assume the memory on each mobile node is 500KB, similar to a sensor node [13]. In practice, the memory on a mobile node can be larger. Thus, each node stores around $S_{data} = 500KB$ trajectory data, which includes 330 trajectories on average. In our study, each trajectory lasts for 50 time points.

Metadata Construction. We apply the CBC algorithm to construct metadata using various k values ($k \in \{5, 10, 50, 100\}$) during k-means clustering. The smaller the k, the more the data is compressed to metadata. The resulting metadata size is $S_{metadata} \in \{1, 2, 7, 14\}KB$. Our data compression is based on the data containment relationship for each level of k as shown in Figure 5.

Query Evaluation. We perform query evaluation under the scenarios with or without XML metadata on a PC with a 2

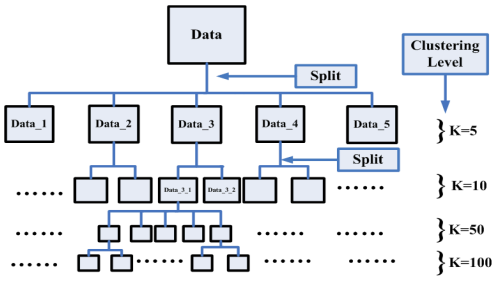
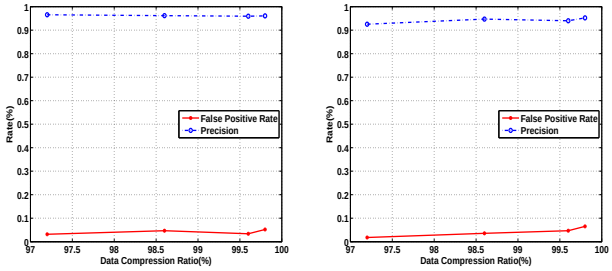


Fig. 5. CBC algorithm: k-means clustering is performed based on the data containment relationship.



(a) 10-node network (b) 50-node network

Fig. 6. False Positive Rate and Precision under metadata-guided query.

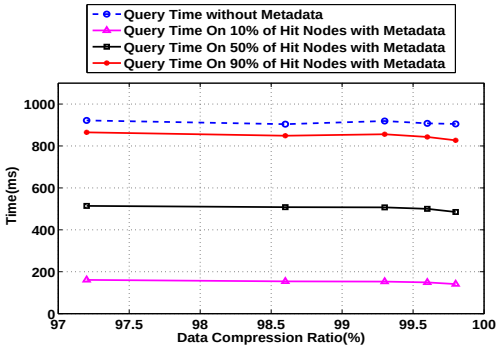


Fig. 7. Query time comparison under various compression ratio of metadata.

GHz Intel Core 2 CPU and 2 GB RAM. When with XML metadata, our evaluation program accesses the XML metadata first. If the metadata returns true, which means there exists a possible answer in the current node, we then read the data stored in this node and search for the query result. Under the scenario without XML metadata, the evaluation program just simply scans all the data stored in every node in the network to answer the query. We run batches of queries with each batch involving different number of nodes. We call the involved nodes in each query *hit node*. Our results are the average of 20 times for each batch test.

B. Metrics

We will utilize the following metrics to evaluate the performance of our metadata-guided approach.

False Positive Rate and Precision. For each query, we define the *false positive* as when the XML metadata on a node returns true, but the subsequent search on the data stored on this node returns zero tuples. Whereas the *true positive* is defined as when the XML metadata on a node returns true, the subsequent data search returns the tuples that match the query. We present false positive in two ways. First, we calculate the false positive rate, which is the percentage of nodes that results in false positive among those nodes return zero tuples in the network in a batch test. Second, we measure *precision*, the percentage of nodes return true positive out of those nodes, which return true from metadata in a batch test. In our study, false positive rate is an important measure of the information loss due to the data compression of the metadata construction.

Compression Ratio. Since each node is memory-constrained, the size of the XML metadata comparing to the size of the original data stored on the node is an important factor to evaluate our approach. We define the Compression Ratio (CR) as

$$CR = 1 - \frac{S_{metadata}}{S_{data}} \quad (1)$$

The CR calculates the percentage of the data that has been compressed to the XML metadata.

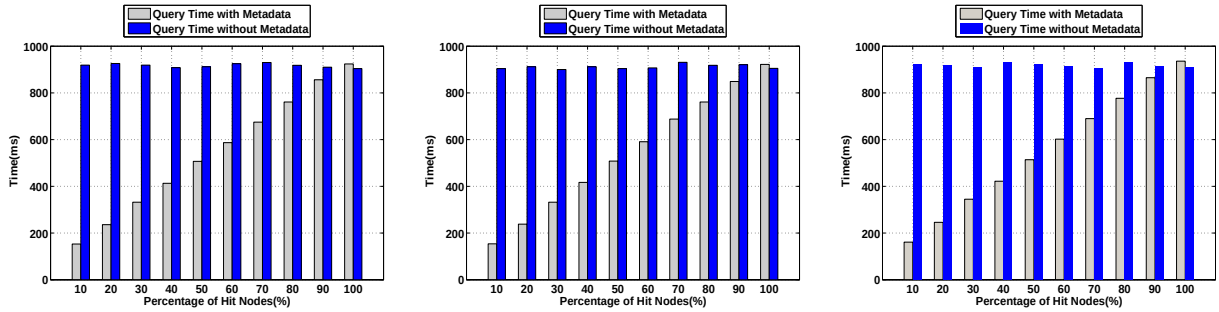
Query Time. We use query time to measure the efficiency of query evaluation. When without metadata, the data stored on all nodes must be accessed. Thus, the query time is equivalent to the total time of evaluating a query on the data of each node in the network. When under the scenario with XML metadata, the query time consists of two parts: the time of evaluating a query on the metadata of each node in the network and the time of evaluating the query on data stored in relevant nodes.

C. Results

False Positive Rate and Precision. Figure 6 presents the false positive rate and precision of metadata-guided queries under various compression ratio for the networks of 10 nodes and 50 nodes respectively. The key observation is that the false positive rate is low and stable, below 6%, under high compression ratio, above 97%. Further, the false positive rate only increases slightly from 4% to 6% when the compression ratio increases from 97% to 99.8%. This is very encouraging as the low false positive rate indicates that the information loss due to the metadata construction is small. On the other hand, we observed high precision, above 96%, under high compression ratios. This indicates that the introduction of metadata only has small impact on the precision when querying data, which may be ignored.

Further, we observed similar trends in both networks, indicating that our approach is not sensitive to the network size. Due to the space limitation, we will only present the results from the 10-node network in the rest of the paper.

Query Efficiency. Figure 7 shows the comparison of query time without metadata to that with metadata guidance under various data compression ratios. The metadata-guided query time is presented with different percentage of hit nodes in the network. We observed that queries without metadata guidance always takes more time than those with metadata guidance. Further, we found that the higher compression ratio corre-



(a) CR = 99.3% (b) CR = 98.6% (c) CR = 97.2%

Fig. 8. Query time comparison under different percentage of hit nodes.

sponds to the less query evaluation time, i.e., there is a slight decreasing trend of the query time when the compression ratio increases. This is because higher compression ratio results in smaller size of the metadata, which needs less time for query evaluation.

Moreover, when using metadata guidance the less percentage of hit nodes, the more efficient our approach is. This observation is further confirmed in Figure 7, which presents the query evaluation time versus the percentage of hit nodes in the network with and without metadata guidance. The query evaluation time without metadata guidance is almost a constant, above 900ms. Under metadata guidance, when the percentage of hit nodes is 10%, the query time is less than 200ms, only 22% of that without metadata guidance. The query time increases as the increasing percentage of hit nodes. This is because in the metadata-guided approach, only those nodes contain the relevant data will be searched for query results, whereas without metadata guidance, the data on all nodes in the network will be evaluated.

Additionally, we observed that when the percentage of hit nodes reaches 100%, the metadata-guided query evaluation time exceeds that without metadata guidance by about 30ms. This is the overhead introduced by the metadata-guided approach, which is only about 3.3% of the query evaluation time on data. Thus, our metadata-guided approach is highly efficient during query evaluation. We believe that with the large-scale data, the performance optimization by our approach metadata will be more significant.

V. Conclusion

In this work, we introduced content caching networks to support data-centric storage for pervasive computing applications in mobile wireless environments. The content caching networks use the metadata-guided query evaluation approach to achieve efficient data dissemination/access. The metadata-guided approach has the characteristic of rich expressiveness that can describe various types of data and capture the diversity of data from heterogeneous wireless devices. Therefore, using metadata in content caching networks can integrate the data from heterogeneous nodes and enable exchange of data among distributed heterogeneous wireless devices. We further developed the CBC algorithm, which is based on data clustering and compression during metadata construction. The CBC algorithm helps to minimize the possible information loss due to data compression, while meeting the memory requirements on wireless devices. Both of our theoretical analysis and empirical results using data generated from a city environment show that the metadata-guided approach is effective in achieving efficient data query and only incurs small overhead, thereby providing strong evidence of the feasibility of content caching networks.

Acknowledgements This work is supported in part by NSF grant CNS-0847211.

References

[1] S. Shenker, S. Ratnasamy, B. Karp, R. Govindan, and D. Estrin, "Data-centric storage in sensornets," *ACM SIGCOMM Computer Communication Review archive*, vol. 33, 2003.

[2] A. Ghose, J. Grossklags, and J. Chuang, "Resilient data-centric storage in wireless ad-hoc sensor networks," in *Proceedings of the 4th International Conference on Mobile Data Management*, 2003, pp. 45–62.

[3] M. Shao, S. Zhu, W. Zhang, and G. Cao, "pDCS: Security and privacy support for data-centric sensor networks," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, 2007.

[4] W. Zhang, G. Cao, and T. L. Porta, "Data dissemination with ring-base index for wireless sensor networks," in *IEEE International Conference on Network Protocols (ICNP)*, 2003.

[5] S. RatNasamy, B. karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker, "GHT: A geographic hash table for data-centric storage," in *ACM International Workshop on Wireless Sensor Networks and Applications*, September 2002.

[6] J. Li, P. Chou, and C. Zhang, "Mutualcast: An efficient mechanism for content distribution in a peer-to-peer (p2p) network," *Microsoft Research TechReport (MSR-TR-2004-100)*, vol. 100, September 2004.

[7] "Extensible markup language (xml)," w3C, http://www.w3.org/XML/.

[8] "XPath 2.0," w3C, http://www.w3.org/TR/xpath.

[9] G. Gottlob, C. Koch, and R. Pichler, "The complexity of xpath query evaluation," in *Proceedings of the ACM International Conference on Principles of Database Systems (PODS)*, June 2003, pp. 179–190.

[10] S. Lloyd, "Least squares quantization in pcm," in *IEEE Transactions on Information Theory*, 1982, pp. 128 – 137.

[11] T. Brinkhoff, "Generating network-based moving objects," in *Proceedings of the 12th International Conference on Scientific and Statistical Database Management*, 2000.

[12] T. Brinkhoff, "A framework for generating network-based moving objects," *GeoInformatica*, vol. 6, no. 2, pp. 153–180, 2002.

[13] "Crossbow Technology Inc." white paper available at http://www.xbow.com.