A technique for n-way joins in wireless sensor networks

Djail BOUBEKEUR, Hidouci Walid KHALED, Loudini MALIK
LCSI Laboratory National High School for Computer Science (ESI) Algiers
b_djail@esi.dz, wk_hidouci@esi.dz, m_loudini@esi.dz

The join queries are operations that require more energy for their execution. In wireless sensor networks, energy is a determinant factor for the network survival. However, high energy consumption caused by such requests needs the implementation of very appropriate techniques for their execution. Research in this field considered especially binary joins. Few studies have addressed n-way joins. In this paper, we propose 'Nway Local Join', an energy-efficient technique for n-way join operations. We adopt an in-network execution at each step of the join operation. We compare our solution with an execution at the sink. NLJ shows the best performance for low selectivity factor.
Keywords: *wireless sensor networks, communication cost, in-network join, n-way join.*

1 Introduction

A wireless sensor network is composed of a very large number of inexpensive and low energy nodes, with processing and memorizing capabilities. The nodes are deployed in a region of interest; they use radio frequencies as communication channels to provide data collection and dissemination.

A sensor is considered as a data source which generates records at different reception actions. The nodes' records of the same type in the same network compose together a distributed table. The sensors network can be regarded as a distributed database system [1]. Collecting received data can be expressed as relational queries, such as selections, projections, union operations, aggregation, and joins.

Joins queries are widely used in wireless sensor networks' applications that require data gathering from multiple nodes.

However, these requests need excessive energy consumption, while energy in wireless sensor networks is a critical factor for the survival of sensors and of the network as well. In wireless sensors network, data transmission consumes more energy than data processing in the node [2]. It is then important to reduce the quantity of transmitted data during a join operation in order to decrease the energy consumed by the sensors. The techniques proposed recently such as Distributed-Broadcast of Coman and al. [3], Mediated Join of Pandit and Gupta [4], Synopsis Join of Yu and al [5], and INJECT of Min and al [6], have been limited to binary joins queries. N-way joins (joins between more than two tables) are rarely addressed.

N-way joins are of a remarkable interest in the networks of sensors. A practical example of the application of this joins type is the traffic control of the vehicles on trajectories made up of several (more than two) zones covered by wireless sensors. Monitoring bird's migrations, through several geographical areas, is another example of the application of this joins type.

This paper presents an energy-efficiency technique for n-way joins execution in wireless sensor networks. The rest of the paper is organized as follows: section 2 provides a description of the general characteristics of joins operations in wireless sensor networks. Section 3 gives related work. Section 4 describes our proposed technique. Section 5 contains the results of the simulation performed, and a discussion of the obtained results. Finally, section 6 concludes the paper.

2 General characteristics of joins in wireless sensor networks

2.1. Basic concepts.

A join of two tables L and R consists of the concatenation of all the tuples from table L

with those of table R where a condition (join predicate) is met on some attributes of the two tuples. When an arbitrary comparison operator (≥, <, =, ..) is allowed in join predicate, the join operation is called theta-join. An equijoin is a theta-join using only the equality operator.
In a wireless sensor network, a join is generally made between two regions of the sensors network. A region consists of a set of sensors of a geographically limited area.

2.2. Joins implementation in wireless sensor networks.

A join operation in the wireless sensors networks can be accomplished according to two implementations:

- External join.

It is the simplest implementation of join operations in wireless sensor networks. The join operation is performed in the base station (sink). This implementation consumes a lot of energy, because of a large amount of transmitted data.

- In-network join.

This is the most complex implementation, but it is more efficient. Join operation is executed by internal nodes of the network before transmitting final results to the base station.

2.3. Join types in wireless sensors network

According to spatial or temporal aspect, the following types of joins operations are distinguished:

- According to a spatial aspect.
  - ✓ Inter-region joins. They represent the joins without spatial predicates; relations are easily identified by assuming that each node knows its location [7].
  - ✓ Unique region joins. Contain spatial predicates. A node receiving a query of this type, can't, in general, determine if it should participate in the query without communicating with other nodes. This may result in a difficult implementation comparing with inter-region joins.
- According to the temporal aspect.
  - ✓ One execution joins. A fixed window is specified for each of the two relations.
  - ✓ Continuous joins. The relations use sliding windows or fixed windows defined for the future.
  - ✓ Periodic joins. Based on jump windows which define a time interval for repeating a query execution.

3. Related works.

Several techniques were proposed to execute join operations in wireless sensors network. They can be divided into two main categories: without filtering and with filtering techniques.
Techniques without filtering execute join operations between all tuples of tables or data streams. This causes a great consumption of energy, due to the high number of transmitted messages. It was the first proposed techniques [1] [4] [8] [9] [10] [11] [12] [13] .
Currently, the tendency for the performance of queries joins in wireless sensor networks, is to filter tuples before the join operation realization. The objective of the filtering techniques is to minimize the number of transmitted messages. These techniques were proposed in several works [3] [5] [6] [14] [15] [16] [17].
All those techniques were suggested for the binary join. Few those developed for nway join. Stern and al. in [18] have suggested a solution allowing processing every joins type in sensors networks (including n-way joins). This solution executes the joins at the base station (sink) in three steps:

i. All join attributes are collected at the base station, to filter some values.
ii. The determined join filter is broadcast in the network.
iii. Each site uses the filter to determine all the relevant tuples and send them to the sink, where the final join is processed.

This technique consumes too much energy because the number of tuples transmitted to the sink is very high.

4. N-way Local Join description.

We propose N-way Local Join (NLJ) to reduce efficiently the consumed energy for n-way joins. We process a one-shot inter-region join having syntax like this:

```
SELECT R1.attrs,
R2.attrs,…,Rn.attrs
FROM  R1, R2, …, Rn
WHERE pred(R1)  AND
pred(R2)  …  AND pred(Rn)
AND join-exp (R1.join-attrs,
R2.join-attrs,…,  Rn.join-attrs)
```

With:

Ri is the relation of an ith region.

pred (Ri) is a predicate of selection of relation Ri, and join-exp is the join condition.

An application example of these joins is the vehicle traffic control through many geographical zones:

```
SELECT Veh1.VehId, Veh1.time,
Veh2.time, Veh3.time
FROM       Veh1, Veh2, Veh3
WHERE (Veh1.time IN i1) and
(Veh2.time in i2) and
(Veh3.time in i3) and
(Veh1.VehId = Veh2. Veh Id)
and (Veh2. Veh Id= Veh3. Veh
Id)
```

Where: i1, i2, and i3 indicate time intervals during which vehicles passed respectively through regions 1,2 and 3.

N-way Local Join (NLJ) consists of executing each join operation at a local node in the network. We also adapt the technique of left linear trees to minimize the number of n-way joins [19], and we consider geographical zone positions of participating zones to determine the execution order of joins (from the nearest to the farthest zone).

By NLJ, a join operation runs in three phases:

Phase 1.

The query is transmitted from the sink to the root node of each area, using a location routing protocol GPSR [20] [21], to ensure the arrival of the query message to the concerned regions. Each region consists of a tree of nodes. Each node must transmit its tuples to the root. We assume that each node knows its location and the locations of its neighbors, via GPS or via localization algorithms [22].

Phase 2.

The join operation is performed with the in-network principle. We also adopt the principle of the left linear trees to determine the execution order of joins. The joins are performed in pairs (Fig. 1). Where an unwinding in two steps is done for each pair of relation R_i and R_{i+1} performing a join operation:

i. The relation R_i is transmitted to the relation R_{i+1} zone.
ii. The join operation is executed in the R_{i+1} region.

Finally, the final result is transmitted to the sink.

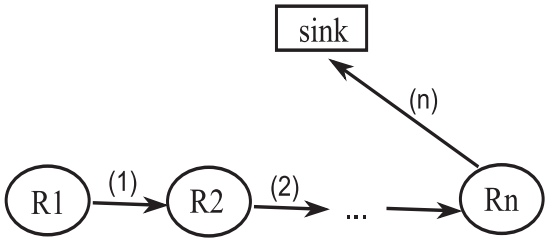

Fig. 1 -N-way local Join (NLJ) execution.

The principle of NLJ will be illustrated with an example. For this purpose, a join between 3 tables is considered.

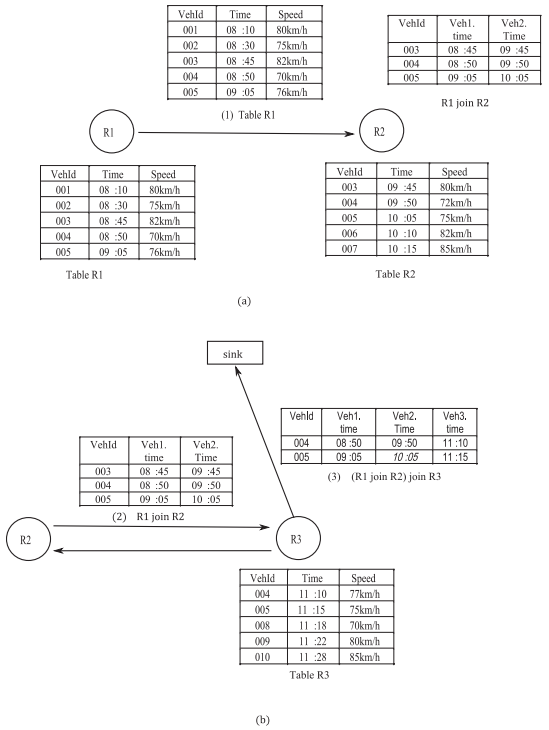

Fig. 2 - An example for N-way local Join (NLJ) execution.

The principle of NLJ will be illustrated with an example. For this purpose, a join between three tables is considered.

In a first step, the join is carried out between the relations R_1 and the relation of the nearest area (R_2). The table R_1 is transmitted to the R_2 area so that the join result between R_1 and R_2 is given (Fig 2 (a)).

Similarly, the join result will be obtained, at the second step, between the table R_3 and the join result of R_1 and R_2 (Fig 2 (b)).

At the last step, the result determined in the R_3 area is sent to the sink (Fig 2 (b)).

5. Performance analysis

5.1. Experimentation environment

To simulate the n-way join execution, we used the NS3 simulator. NLJ has been applied on static tables of 2000 tuples each. The message size is a tuple size, which is 40 bytes each. The column size is assumed to 10 bytes; the size of the result tuple is 30 bytes.

The communication cost was tested considering selectivity factors of intermediates joins in the interval $[10^{-5}, 10^{-4}]$, and then in the interval $[10^{-4}, 10^{-3}]$. Selectivity factors of intermediates joins are generated in a random way. The values of the horizontal axis of results graphs represent averages of the intermediates joins' selectivity factors.

We process separately two simulations:

- 3 tables simulation,
- 5 tables simulation.

For each simulation, we perform 2 cases:

- the extern join simulation,
- and our technique (NLJ) simulation.

5.2. Experimentation results

In the interval $[10^{-5}, 10^{-4}]$ of selectivity factors, (Fig 3 and Fig 4), N-way Local Join perform better than extern join in both simulations: 3 tables and 5 tables. In this interval, selectivity factors are very low.

In the interval $[10^{-4}, 10^{-3}]$ of selectivity factors, (Fig 5 and Fig 6), N-way local join continues to offer the best performance, but not throughout the entire interval. With the high values of selectivity factors, NLJ decrease in performance in favor of extern-join technique.

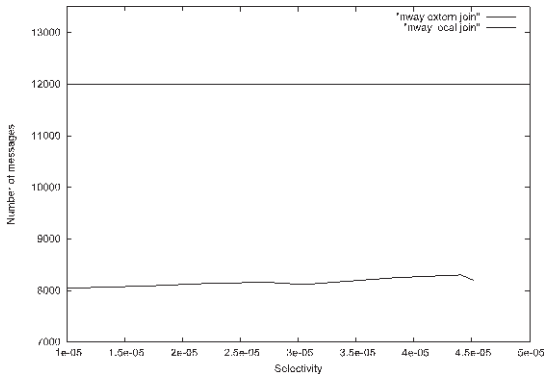

Fig 3. Communication cost for 3 tables following the average of intermediates joins' selectivity factors in the interval $[10^{-5}, 10^{-4}]$

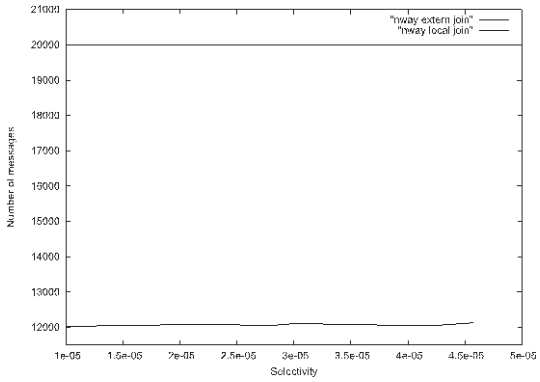

Fig 4. Communication cost for 5 tables following the average of intermediates joins' selectivity factors in the interval $[10^{-5}, 10^{-4}]$

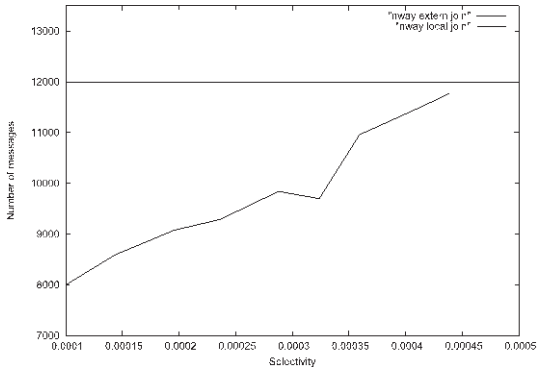

Fig 5. Communication cost for 3 tables following the average of intermediates joins' selectivity factors in the interval $[10^{-4}, 10^{-3}]$

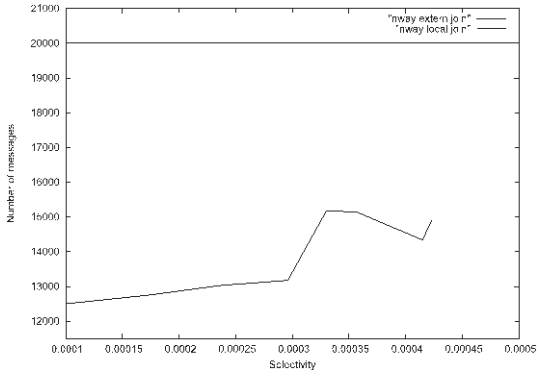

Fig 6. Communication cost for 5 tables following the average of intermediates joins' selectivity factors in the interval $[10^{-4}, 10^{-3}]$

5.3. Discussion

Because join operations are executed at sensors nodes during steps of the join execution, our approach offers the best performance.

Extern-join performs the join operation at the sink. This engenders high energy consumption, due to an important transmitted information volume.

It is important to note that these results are valid for low selectivity values.

For high selectivity values, it is ideal to realize the join query at the sink.

6. Conclusion

We have presented, in this paper, a technique for n-way join execution, in wireless sensor networks. We have adopted the principle of in-network execution to reduce the number of tuples transmitted between sensors and to sink.

Our approach shows better performance compared to extern-join principle for low selectivity factor.

In future work, we plan to a study joins operations between data streams generated by the sensors of each region. The techniques suggested in the case for distributed databases were proposed in [23] and can be adapted to the cases of the databases of sensors network.

References

[1] Y. Yao and J. Gehrke, "Query Processing in Sensor Networks," in *CIDR*, 2003, pp. 233-244.

[2] F. Zhao and L. J. Guibas, *Wireless sensor networks: an information processing approach*: Morgan Kaufmann, 2004.

[3] A. Coman, M. A. Nascimento, and J. Sander, "On join location in sensor networks," in *2007 International Conference on Mobile Data Management*, 2007, pp. 190-197.

[4] A. Pandit and H. Gupta, "Communication-efficient implementation of range-joins in sensor networks," in *International Conference*

on Database Systems for Advanced Applications, 2006, pp. 859-869.

[5] H. Yu, E.-P. Lim, and J. Zhang, "On in-network synopsis join processing for sensor networks," in *7th International Conference on Mobile Data Management (MDM'06)*, 2006, pp. 32-32.

[6] J.-K. Min, H. Yang, and C.-W. Chung, "Cost based in-network join strategy in tree routing sensor networks," *Information Sciences,* vol. 181, pp. 3443-3458, 2011.

[7] H. Kang, "In-network processing of joins in wireless sensor networks," *Sensors,* vol. 13, pp. 3358-3393, 2013.

[8] B. J. Bonfils and P. Bonnet, "Adaptive and decentralized operator placement for in-network query processing," *Telecommunication Systems,* vol. 26, pp. 389-409, 2004.

[9] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "The design of an acquisitional query processor for sensor networks," in *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, 2003, pp. 491-502.

[10] D. J. Abadi, S. Madden, and W. Lindner, "Reed: Robust, efficient filtering and event detection in sensor networks," in *Proceedings of the 31st international conference on Very large data bases*, 2005, pp. 769-780.

[11] V. Chowdhary and H. Gupta, "Communication-efficient implementation of join in sensor networks," in *International Conference on Database Systems for Advanced Applications*, 2005, pp. 447-460.

[12] S. R. Mihaylov, M. Jacob, Z. G. Ives, and S. Guha, "A substrate for in-network sensor data integration," in *Proceedings of the 5th workshop on Data management for sensor networks*, 2008, pp. 35-41.

[13] S. R. Mihaylov, M. Jacob, Z. G. Ives, and S. Guha, "Dynamic join optimization in multi-hop wireless sensor networks," *Proceedings of the VLDB Endowment,* vol. 3, pp. 1279-1290, 2010.

[14] X. Yang, H. B. Lim, T. M. Özsu, and K. L. Tan, "In-network execution of monitoring queries in sensor networks," in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, 2007, pp. 521-532.

[15] M. Stern, K. Böhm, and E. Buchmann, "Processing continuous join queries in sensor networks: a filtering approach," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, 2010, pp. 267-278.

[16] Y.-X. Lai, Y.-L. Chen, and H. Chen, "PEJA: Progressive energy-efficient join processing for sensor networks," *Journal of Computer Science and Technology,* vol. 23, pp. 957-972, 2008.

[17] Y. Lai, Z. Lin, and X. Gao, "SRJA: Iceberg Join Processing in Wireless Sensor Networks," in *2010 2nd International Workshop on Database Technology and Applications*, 2010, pp. 1-4.

[18] M. Stern, E. Buchmann, and K. Böhm, "Towards efficient processing of general-purpose joins in sensor networks," in *2009 IEEE 25th International Conference on Data Engineering*, 2009, pp. 126-137.

[19] M. Steinbrunn, G. Moerkotte, and A. Kemper, *Optimizing join orders*: Citeseer, 1993.

[20] B. Karp and H.-T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, 2000, pp. 243-254.

[21] 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 *Proceedings of the 1st ACM international workshop on*

Wireless sensor networks and applications, 2002, pp. 78-87.

[22] A. Savvides, M. Srivastava, L. Girod, and D. Estrin, "Localization in sensor networks," in *Wireless sensor networks*, ed: Springer, 2004, pp. 327-349.

[23] T. M. Tran and B. S. Lee, "Distributed stream join query processing with semijoins," *Distributed and Parallel Databases,* vol. 27, pp. 211-254, 2010.