

Análisis de la operación de unión en redes de sensores inalámbricos

Axel Rodríguez¹ , Armando Jipsion² 

¹Estudiante de Doctorado en Ingeniería de Proyectos, ²Docente/investigador
Universidad Tecnológica de Panamá

¹axel.rodriguez2@utp.ac.pa; ²armando.jipsion@utp.ac.pa

DOI <https://doi.org/10.33412/pri.v10.1.2173>



Resumen: El consumo de energía es un factor importante en el rendimiento y la vida útil de las redes de sensores inalámbricos. Las redes de sensores inalámbricos están conformadas por módulos de sensores distribuidos geográficamente en un área determinada con el objetivo de recopilar información necesaria para las distintas mediciones en las investigaciones científicas. La transmisión de los datos es el factor principal en el consumo de energía en una red de sensores inalámbricos. Los datos de los nodos de la red conforman los registros para una base de datos distribuida. La unión de estos registros antes de la transmisión garantiza que se difundan menos datos y, por lo tanto, se requiere menos energía en las redes de sensores inalámbricos. Los sistemas actuales de base de datos para redes de sensores inalámbricos, como verbigracia: TinyDB o Cougar, pueden proporcionar una función de consulta similar a SQL. Por ejemplo, pueden hacer una función agregada GROUP-BY. Sin embargo, no son compatibles con la operación de consulta de unión. En este estudio, analizamos la operación de unión, la cual es de suma importancia en la transmisión de un conjunto de datos determinado fuera de línea. Los resultados de la evaluación experimental para esta investigación demuestran que es útil implementar la operación de consulta de unión en el entorno de las redes de sensor inalámbrico, a fin de que podamos garantizar el ahorro de energía de los sensores y por ende la vida útil de los mismos.

Palabras clave: Red de sensores inalámbricos, consulta de unión, TinyDB.

Title: Analyzing the necessary of join operation in sensor network.

Abstract: Power consumption is a significant factor in the performance and lifetime of wireless sensor networks. Wireless sensor networks entail sensor modules that are geographically distributed over a given area to collect data. Transmission of the data is the primary factor in the power consumption in a wireless sensor network. The data from the nodes of the network make up tuples of a distributed database. Joining these tuples before transmission ensures that less data is transmitted and thus less power is required. Current wireless sensor network database systems, such as TinyDB[1] or Cougar[2], can provide SQL-like

query function. For example, they can do GROUP-BY or Aggregate function. However, they don't support join query operation. In this report, I tried to analyze the necessary of join operation by a given dataset off-line. The experimental evaluation results show that it is useful to implement join query operation in wireless sensor network environment.

Key words: Wireless Sensor Network, Join Queries, TinyDB

Tipo de artículo: original

Fecha de recepción: 9 de agosto del 2018

Fecha de aceptación: 17 de octubre de 2018

1. Introducción

Una red de sensores inalámbricos se compone de varios módulos de sensores que se llaman nodos. Los nodos recopilan datos a través de los sensores y los transmiten a la ubicación deseada llamada sumidero o nodo central. Los nodos pueden tener capacidades de procesamiento y almacenamiento. Un nodo sensor es una fuente de datos que genera los registros. Diferentes nodos en la misma red pueden crear registros del mismo tipo; que a su vez pueden agruparse a fin de conformar una base de datos distribuida. Algunas aplicaciones de la red de sensores inalámbricos incluyen domótica y agricultura. Los registros del sensor se pueden considerar como registros de una base de datos lógica en el sistema de base de datos físicamente distribuido. Una de las principales operaciones en la base de datos que involucra tuplas es una consulta de combinación. Las investigaciones en el campo de la electrónica y las comunicaciones luchan por encontrar estrategias para maximizar el tiempo de batería de las redes de sensores inalámbricos (WSN) con el fin de ahorrar energía. El poder efectuar la consulta de unión de los datos antes de su transmisión nos permitirá efectuar un ahorro de energía es decir se efectuará una depuración de los datos antes de su transmisión.

WSN es una tecnología emergente que se ha considerado como una herramienta muy útil para dominios variantes, como la detección de hábitats y el monitoreo de volcanes [3]. Sin embargo, los nodos de sensores inalámbricos son bien conocidos por sus recursos limitados. Por ejemplo, tienen muy poca potencia, velocidad de procesamiento, capacidad de almacenamiento y ancho de banda de comunicación. Estas limitaciones de recursos conducen a nuevos desafíos de investigación.

En cuanto a la gestión de datos en la red de sensores inalámbricos, existen dos sistemas de bases de datos principales. Uno es TinyDB [1], y otro es Cougar [2]. Ambos sistemas proporcionan una función similar al lenguaje de consulta de base de datos (SQL).

Los usuarios pueden recopilar datos de la red de sensores emitiendo un lenguaje de consulta similar a SQL. Estos sistemas también desarrollaron consultas agregadas a las consultas de base de datos como, COUNT, SUM y AVG. Estas técnicas de agregación dentro de la red pueden reducir significativamente la cantidad de transmisión de datos; ya que reducen la potencia de los nodos del sensor y pueden conducir a su destrucción afectando la transmisión de los datos en las redes WSN.

Sin embargo, en ciertos escenarios, las consultas simples de selección o agregación no son lo suficientemente buenas para manejar todos los requisitos de consulta del usuario. Un escenario es que podríamos interesarnos en rastrear las correlaciones entre los datos del sensor en una ventana de tiempo para hacer la detección del evento. Por ejemplo, en el monitoreo de volcanes, los científicos están interesados en las presiones detectadas dentro de una determinada región alrededor de la montaña volcánica. Después de notar que la actividad volcánica de la montaña se ha incrementado, es posible que deseen saber si las presiones detectadas han pasado un cierto umbral y si aumentan continuamente dentro de un período de tiempo. Esta consulta se puede escribir en esta forma.

```
SELECT P1.pressure, P1.time, P2.pressure, P2.time
FROM Pressure AS P1, Pressure AS P2
WHERE P1.pressure > threshold
AND P2.pressure > P1.pressure
AND P2.time > P1.time
```

En esta consulta, el valor umbral es un valor constante dado por el usuario, y la tabla Presión representa la tabla en el nodo sensor, que recopila información de presión. A partir de esta consulta, podemos observar que hay una operación de unión allí.

En el sistema de base de datos existente en la red de sensores, no proporcione la función de unión. Si uno quiere realizar esta consulta, tiene que pedir a todos los nodos del sensor que envíen de vuelta sus datos a la estación base, y luego dejar que la estación de peaje realice las operaciones de unión. Sin embargo, esto daría como resultado una alta transmisión de datos y reduciría el recurso de potencia para cada nodo. Por lo tanto, es importante implementar la operación de unión para reducir la alta transmisión de datos para este tipo de escenarios.

Las siguientes secciones se resumen a continuación. La sección 2 describe el enfoque para implementar la operación de unión en la red de sensores. La sección 3 describe el entorno de nuestro experimento y es el resultado experimental. Finalmente, concluimos este experimento en la sección 4.

Una unión es una operación de base de datos que implica la coincidencia de registros (tuplas o filas) dependiendo de si se cumple un predicado particular o no. La relación lógica, en este caso, podría ser una tabla. Dadas dos tablas L y R, una unión entre las tablas L y R es una concatenación de las tuplas en L a aquellas en R cuando se cumple una condición específica. En redes de sensores inalámbricos, se realiza una unión entre nodos de sensores en la misma región de la red de sensores inalámbricos. Las uniones se pueden implementar de dos maneras: unión externa y uniones internas. Las uniones externas se implementan en las estaciones base. Eso significa que los datos se transmiten antes de que se implementen las uniones.

Esto conduce a un alto consumo de energía ya que se deben enviar más datos y cada sensor de la red de sensores inalámbricos está involucrado en la transmisión. Las uniones internas / dentro de la red ejecutan uniones en los nodos antes de transmitir los resultados a las estaciones base. Los operadores de

la consulta se envían a la red y cada nodo ejecuta la consulta. El resultado es intermedio para la consulta. Las tuplas del sensor se procesan dentro de la red antes de la transmisión a la estación base. En efecto, se transmiten menos mensajes y la transmisión usa menos energía.

2. El enfoque para implementar la operación de unión en la red de sensores --- TPSJ [4]

En esta sección, describiremos brevemente el enfoque para implementar la operación de unión en la red de sensores. Este enfoque se denomina auto-unión de ventana de dos fases (TPSJ) [4]. La sección 2.1 define el escenario de ejecución. La sección 2.2 describe el enfoque TPSJ.

2.1 Escenario de ejecución

El escenario de ejecución que nos interesa es encontrar las correlaciones entre los datos del sensor dentro de una ventana de tiempo. La correlación aquí significa la relación entre los datos sea una muestra única. El formulario de consulta general de este requisito se puede definir como la siguiente forma:

```
SELECT S1.AT1, S2.AT2
FROM Sensor AS S1, Sensor AS S2
WHERE p1(S1.AT3)
AND p2(S1.attj, S2.atth)
AND window (S1.ts, S2.ts,W)
```

donde Sensor es una tabla en nodos de sensor, p1 es un predicado para el filtrado de datos, p2 es un predicado para la operación de unión, y Windows (S1.ts, S2.ts, W) es un predicado conceptual para restringir la muestra en un período.

2.2. Enfoque de selfjoin en dos fases (TPSJ)

En esta sección discutimos el enfoque TPSJ. Básicamente, como sugiere el título, TPSJ se puede dividir en dos fases, que son la descomposición de consultas y el proceso de autoensamblado. La idea básica de hacer esto es hacer que la consulta original sea más eficiente. La consulta original se descompondrá en las siguientes dos formas:

Forma 1

```
SELECT S.AT1 INTO R1
FROM Sensor AS S
WHERE p1(S.AT3)
```

Forma 2:

```
SELECT S.AT2
FROM R1, Sensor AS S
WHERE p2 (R1.attj, S.atth)
AND window (R1.ts, S.ts,W)
```

De 1 sería la primera fase en TPSJ, y De 2 sería la segunda fase en TPSJ. La primera fase tiene dos propósitos. Un propósito es seleccionar algunas tuplas que puedan contribuir con el resultado final. Estas tuplas se usarán para filtrar datos en la fase dos. Otro objetivo es determinar cuándo iniciar una operación de unión de ventana. Si el formulario 1 devuelve más de uno tupla de regreso a la estación base, luego emitiremos la segunda consulta.

El propósito de la segunda fase es usar la tupla candidata en la fase uno para realizar la operación de unión para hacer la detección de eventos dentro de una ventana de tiempo.

Usemos la forma de consulta de monitoreo de volcanes en la sección 1 para dar un ejemplo. La descomposición de la consulta de fase uno sería la siguiente:

```
SELECT P.pressure, P.time INTO R1
FROM Pressure AS P
WHERE P.pressure > threshold
```

En la figura 1 ejemplificamos la ilustración de la consulta anterior.

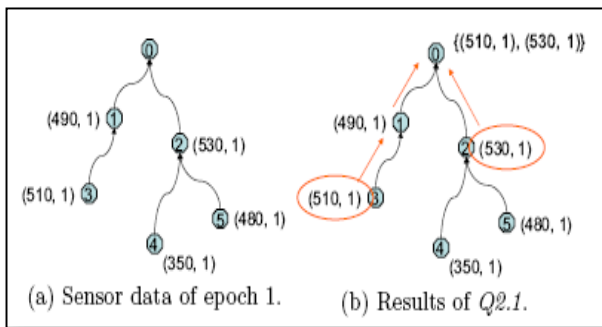


Figura 1. Fase uno tomada de [5]

En la figura 1, cada nodo del sensor mantiene una tabla que consiste en la lectura del sensor y la época. El nodo 0 es nuestra estación base para recopilar datos. El valor de umbral en la figura uno es 500. La figura 1 (a) muestra cada una de las lecturas del sensor en la fase uno, y la figura 1 (b) es el resultado de la fase uno. Los nodos 3 y 2 satisfacen con el predicado, por lo que envían su valor de lectura a la estación base.

La descomposición de la consulta de la fase dos sería la siguiente:

```
SELECT P.pressure, P.time
FROM R1, Pressure AS P
WHERE P.pressure > R1.pressure
AND window (R1.time, P.time, h)
```

Utilizaremos la figura 2 como un ejemplo para la ilustración de la consulta anterior.

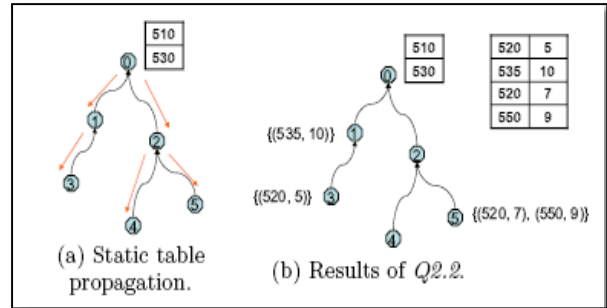


Figura 2. Fase dos tomada de [5]

En la figura 2, el tamaño de la ventana (h) es 10 y en la parte (a), la estación base inyecta los resultados de la primera fase en la red. Cada nodo sensor ahora tiene esta tabla, que tiene 510 y 530 lecturas. Después de diez fases, los nodos 1, 3 y 5 tienen lecturas que son mayores que 510 y 530. Por lo tanto, el resultado final está dado por la lectura de datos del nodo 1, 3, 5.

3. Experimento

En nuestro experimento, utilizaremos el conjunto de datos del laboratorio de Intel en Berkeley California [5] y MySQL para analizar el algoritmo TPSJ. El esquema del conjunto de datos es el siguiente:

Date	time	epoch	id	temp	humidity	light	volt
------	------	-------	----	------	----------	-------	------

En nuestro experimento, solo utilizamos los atributos en color rojo para hacer el análisis. La topología de red se muestra en la figura 3.

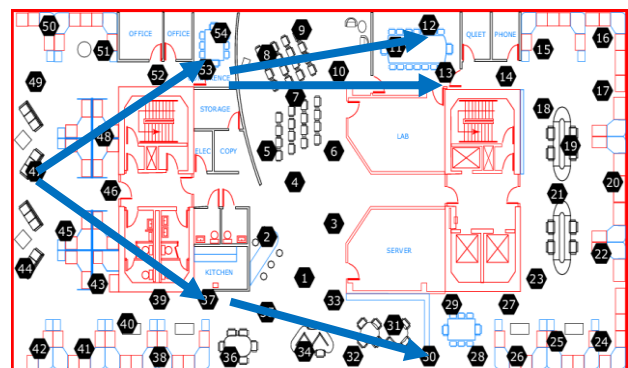


Figura 3. Topología de red

La figura 3 es el mapa del laboratorio de investigación de Intel en Berkeley. Hay 54 nodos de sensor en este laboratorio. Para este experimento, utilizamos el nodo 47, 53, 12, 13, 37 y 30 para formar una topología de árbol. El nodo 47 es la estación base.

La consulta para evaluar el TPSJ en este experimento es la siguiente:

```
SELECT A.id, B.id, A.humidity, B.humidity
FROM sensor as A, sensor as B
WHERE A.humidity > threshold
AND A.humidity > B.humidity
AND window (B.epoch, A.epoch, h)
```

Además de TPSJ, también tratamos de analizar el método tradicional para lograr la operación de unión. Es decir, todos los sensores envían las lecturas de sus sensores a la estación base y permiten que la estación base se una a la operación. Mediante esta comparación, podemos analizar las operaciones de unión necesarias en la red de sensores.

La métrica de evaluación para este experimento es la cantidad de transmisión de datos en toda la red. La Figura 4 muestra el resultado experimental.

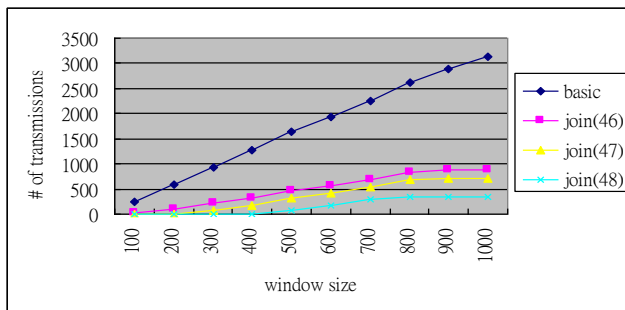


Figura 4. Número de transmisiones de datos en la red

La línea básica indica el método tradicional para lograr la operación de unión. La unión (N) significa que establecemos que el valor umbral sea N en TPSJ. En este caso, el umbral es igual a 46, 47 y 48 respectivamente. Para el tamaño de la ventana, el rango es de 100 a 1000.

En la figura 4, podemos ver que el TPSJ es mucho más externo que el método básico, especialmente cuando la selectividad es baja, es decir, un valor umbral más alto.

4. Conclusión

Nuestro resultado experimental muestra que la operación de unión puede reducir drásticamente la cantidad de transmisión de tuplas en la red. El consumo de energía es un punto importante de consideración con respecto al costo en la implementación de redes de sensores inalámbricos. Las redes de sensores inalámbricos usan la potencia para procesar y transmitir datos. Sin embargo, la cantidad de energía utilizada en la transmisión de datos es mucho mayor que la utilizada en el procesamiento de datos. Las operaciones de uniones de base de datos dentro de la red de sensores, consumen menos energía porque se reduce la cantidad de datos que se transmitirán a la estación base. Por lo tanto, podemos concluir que la operación de unión es necesaria para la red de sensores y garantizar su vida útil.

Referencias

- [1] A. Agrawal, L. Gupta, and B. K. Behera, "Query Processing in Wireless Sensor Network," vol. 3, no. 1, pp. 1–6, 2013.
- [2] D. O. Kim, L. Liu, I. S. Shin, J. J. Kim, and K. J. Han, "Spatial TinyDB: A spatial sensor database system for the USN environment," *Int. J. Distrib. Sens. Networks*, vol. 2013, 2013.
- [3] D. Gupta, "Wireless Sensor Networks 'Future trends and Latest Research Challenges,'" *IOSR J. Electron. Commun. Eng. Ver. II*, vol. 10, no. 2, pp. 2278–2834, 2015.
- [4] S. Mo, Y. Fan, Y. Li, and X. Wang, "Multi-Attribute Join Query Processing in Sensor Networks," *J. Networks*, vol. 9, no. 10, pp. 2702–2712, 2014.
- [5] X. Yang, H. B. Lim, T. M. Özsu, and K. L. Tan, "In-network execution of monitoring queries in sensor networks," *Proc. 2007 ACM SIGMOD Int. Conf. Manag. data - SIGMOD '07*, p. 521, 2007.
- [6] <http://db.csail.mit.edu/labdata/labdata.html>