



Sensor Database System

Sultan Alhazmi

Salha037@uottawa.ca

Outline:

- Definition
- General Examples
- Factory warehouse
- Design Approaches
- COUGAR
- TinyDB
- Antelope



Definition:

- A sensor database involves a combination of stored data and sensor data.
- sensor data is generated by signal processing functions.
- Stored data include the set of sensors that participate in the sensor database together with characteristics of the sensors (e.g., their location) or characteristics of the physical environment.

Some Examples:

- monitoring in-building energy usage for planning energy conservation
- Supervising items in factory warehouse
- Gathering information in a disaster area
- military and civilian surveillance
- fine-grain monitoring of natural habitats with a view to understanding ecosystem dynamics



Some Examples:

- data gathering in instrumented learning environments for children
- Measuring variations in local salinity levels in riparian environments
- **what make some of today's network different :**
 - 1- they operate unattended and untethered.
 - 2- energy-efficiency is becoming a primary design consideration

Factory warehouse:

- The goal is to make sure that items do not overheat
- Temperature sensors are attached on walls and ceiling
- Each item has a stick-on temperature sensor attached to it
- Sensors can do the following:
 - a- Get_Temperature : return the measured temperature
 - b- Detect_Alarm : return the temperature when exceeds certain threshold
- Each sensor can communicate this data and/or store it locally

Factory warehouse:

- The sensor database stores the identifier of each sensor besides their locations
- Some typical queries that are used:
 - query1. Return repeatedly the abnormal temperature measured
 - query2. Every 5 minutes return the measured temperature on the second floor

Approaches:



- **Data Warehousing Approach:**
- processing of sensor queries and access to the sensor network are separated.
- data is extracted from the sensor network in a predefined way and is stored in a database located on a unique front-end server.
- it periodically retrieves data from the sensor network and stores the data at a centralized database

Approaches:



- **Data Warehousing Approach:**
- query processing takes place on the centralized database.
- It is well suited for answering predefined queries over historical data.
- It requires significant communication and that requires energy
- **Limitations:** exhaust the energy of the sensors and produce a lot of redundant data

Approaches:



Distributed Approach:

- the query workload determines the data that should be extracted from sensors.
- Flexible: different queries extract different data from the sensor network
- Efficient: only relevant data are extracted from the sensor network.

Approaches:



Distributed Approach:

- Energy efficient: the query rate is less than the rate at which data was generated
- Traditional distributed database is Unsuitable for large-scale networks because the design has traditionally assumed well-maintained global meta data distribution and network topology

COUGAR:



- The COUGAR System is a platform for query processing techniques over ad-hoc sensor networks
- Treats a sensor network as a distributed database
- A query optimizer is located on the gateway node to generate distributed query processing plans after receiving queries from outside
- The COUGAR forms a clusters out of the sensors to allow intelligent in-network aggregation to conserve energy by reducing the amount of communication between sensors.

COUGAR:

- each sensor type has a standard Abstract Data Type representation which is used at all nodes. It is not possible to insert sensing nodes with new sensing capabilities into the network in an ad hoc manner.

COUGAR:



- Architecture :
- The QueryProxy: a small database component that runs on sensor nodes to interpret and execute queries
- Frontend component: which is a more powerful QueryProxy that permits connections to the world outside of the sensor network
- a graphical user interface through which users can pose ad-hoc and long-running queries on the sensor network.

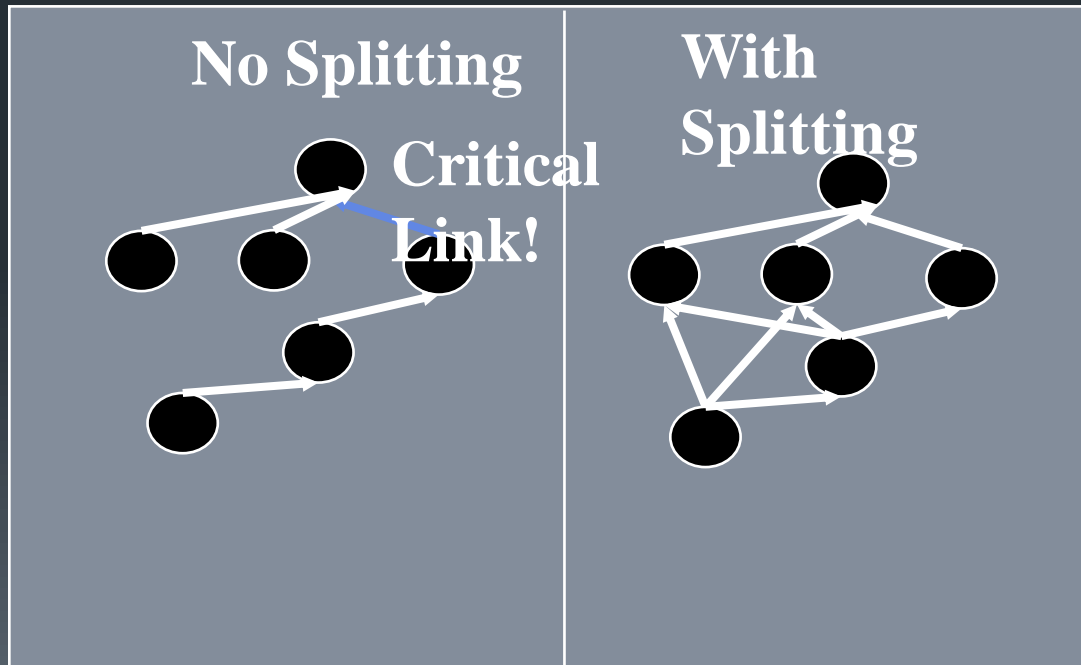
TinyDB



- enquiry processing system for sensor networks that operates on TinyOS
- TinyDB provides a simple SQL-like interface to query sensor data much as you would pose queries against a traditional database
- Query processing system for extracting information from a network of TinyOS sensors
- collects data from motes, filters it, aggregates it together, and routes it

TinyDB

- Every node has an identically structured sensor table containing local sensor data. Each type of sensor corresponds to an attribute (column) in this table



TinyDB



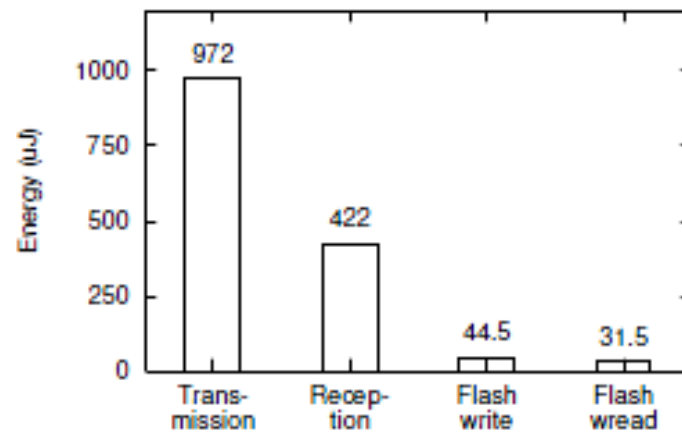
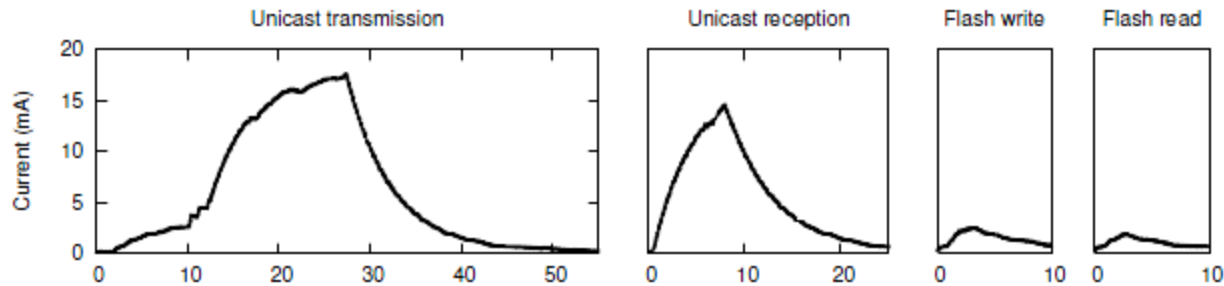
- Motivation:
- The primary goal of TinyDB is to allow data-driven applications to be developed and deployed much more quickly.
- Acquire and deliver desired data while conserving as much power as possible
- TinyDB transforms diverse kinds of sensor networks into user-friendly virtual databases rich with raw information about the real world.

Database in every sensor:



- Deployment experiences show that aggregation is rarely used in practice. Indeed, in many cases each device has a unique task
- Each sensor device should run its own database system.
- low-power flash memory has both rapidly decreased in cost and rapidly increased in storage capacity.

- The energy cost of a query that selects 100 tuple is less than the cost of single packet transmission



Antelope:

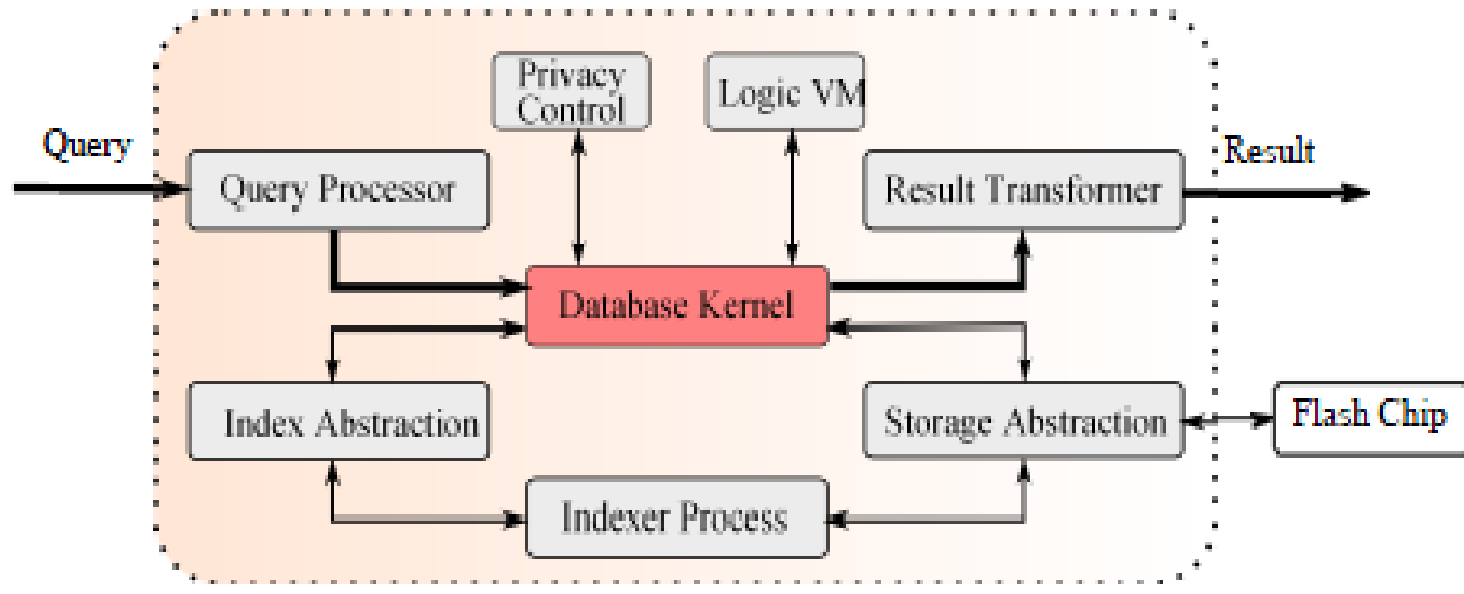


- Antelope contains a flexible data indexing mechanism that includes three different indexing algorithms
- Each node in the sensor network provides a database interface to their stored data and each mote runs a database manager for energy-efficient data querying
- Queries are made to individual nodes instead of to a dedicated sink node

Antelope:

- Antelope consists of eight components:
 - 1- **query processor**: which parse AQL queries
 - 2- **Privacy control**: which ensures that query is allowed
 - 3- **LogicVM**: which executes the query
 - 4- **Database kernal**: which holds the database logic and coordinates query execution
 - 5- **Index abstraction**: which holds the index logic
 - 6- **Index process**: which builds indexes from existing data
 - 7- **storage abstraction**: which contains all storage logic
 - 8- **result transformer**: which presents the results of a query in way that makes it easy to be used by programs

Antelope:





Thank you

Questions:

- Define sensor database and its combination?
- Ans:
- A sensor database involves a combination of stored data and sensor data.
- sensor data is data that generated by signal processing functions.
- Stored data include the set of sensors that participate in the sensor database together with characteristics of the sensors (e.g., their location) or characteristics of the physical environment.

Questions:

What are the main approaches of Sensor Database Systems? And which one of them applies for each of the following characteristics:

1- well suited for answering predefined queries over historical data.

2- data is extracted from the sensor network in a predefined way and is stored in a database located on a unique front-end server.

3- Efficiency

4- Energy efficient

5- require significant communication

6- produce a lot of redundant data

ANS:

The main approaches are: "Data warehousing approach" and "Distributed approach"

1- Data warehousing approach

2- Data warehousing approach

3- Distributed approach

4- Distributed approach

5- Data warehousing approach

6- Data warehousing approach

Questions:

- Antelope is a processing query system. The main idea of this system is to save the data in the sensor? What are the arguments that support this idea to be considered efficient?
- Ans:
- low-power flash memory has both rapidly decreased in cost and rapidly increased in storage capacity.
- The energy cost of a query is less than the cost of single packet transmission