Problem solutions 5

Alberto Defendi - alberto.defendi@helsinki.fi

February 14, 2022

1 Solutions

Exercise 1:

1. ```
   ON EVENT fire-detect(loc):
   SELECT AVG(temp), AVG(co2), event
   FROM SENSORS AS TMP
   WHERE dist(tmp.temp, event.tmp) > 2 AND  dist(tmp.co2, event.co2) > 3
   SAMPLE PERIOD 2s FOR 30s;
   ```

2. ```
       SELECT AVG(CO2), AVG(temp)
   FROM SENSORS AS tmp
   WHERE LOCATION = loc
   AND tmp.temp > threshold.temp AND tmp.co2 >
   threshold.co2
   FRESHNESS 10 seconds;
   ```

3. To support event based queries, the system should support event-based programming. To support lifetime-based queries, the system should should support thread-based programming.

4. It should be made sure that the new devices uses the same operating system (possibly the same version). First, the administrator should check if there are drivers for the control unit for this OS, and check if the sensors record correct values.

Exercise 2:

1. The tasks related to improve safety and driving quality should be performed on device to reduce latency. This can be done since the car PC has good energy resources and computational power. The data

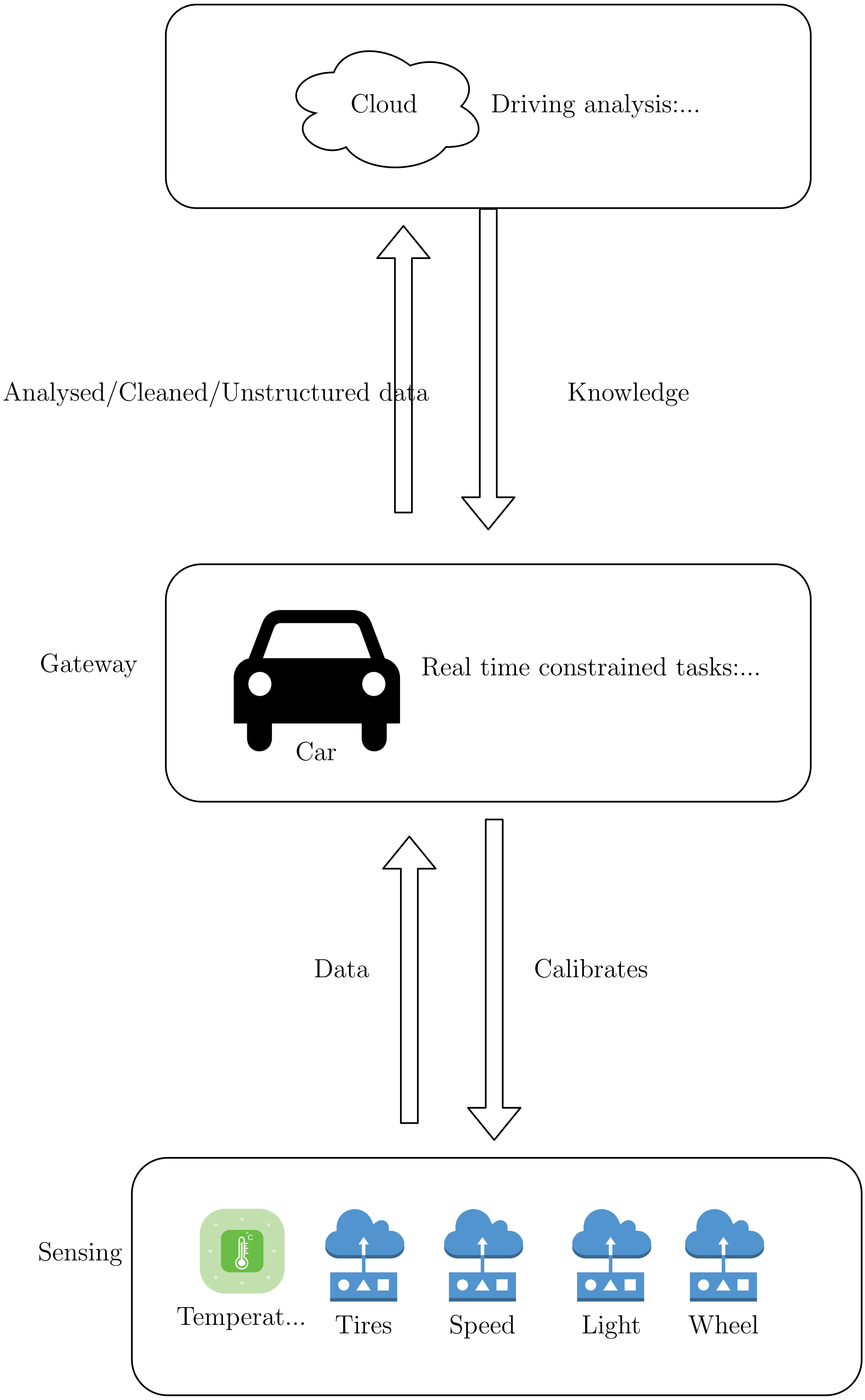

Figure 1: Smart car

and the results from offline training can be sent online to the cloud that improves the upcoming models and finds abnormal situation. This is advantageous because data collected from other users can be used to spot potential problems.

2. See fig. 1.

3. **Locally** Sensor recordings and value checking. As task sensors record values (such as measuring tires pressure) and signal the gateway if the value respects a certain threshold.

   Gateway The gateway is the car's computer, which can perform some analysis with the data coming from the sensors to trigger events. Some tasks can be:

   - Query data from sensors.
   - Use data from other programs (e.g cruise control, collision detect, car status).
   - Send data to server.
   - Re-calibrate sensors.

   Cloud The cloud does more complex calculation and data analysis. Some tasks can be:

   - Listen to car's update.
   - Train data and provide analysis for other applications, such as traffic analysis, Geo-based alerts, fault detection.
   - Send the knowledge back to the car.
   - Organize and store data.

4. Trough a P2P network, where each car gateway is a node of the network. This can only made possible if the manufacturers adopt the service and contribute to the network. The efficiency of the network depends from the number of cars around.

Exercise 3:

1. Warehouse or any centralized model, since a decentralized requires constant communication with the devices, and Sigfox has a limited daily exchange of 4 messages in down-link.

2. It would be easier with LoRa class A, with Class C the receiver can receive data from the server only when the connection is open, which makes hard to have constant updates from the devices.

3. **Value** it is important to determine how the detection of animal movement beneficial to the company. The system can be used to detect the preferred places of the animals, and to improve the spaces from the analysis.

   Volume it is important to establish an interval for the data sampling and to optimize the sensor's position to reduce the volume of data, which grows linearly with the formula $data_{minute} = number_{sensors} \cdot number_{recordings/min} \cdot data_{size}$

   Variety The data should the greatest amount of field, otherwise the measurements would be imprecise.

4. **Real time stream** checking if the animals are currently in a certain zone.

   Batch analysis on the zones preferred by the animals.

Exercise 4:

(a) i. Each node can have an hourly average entry in the table. When a sensor records the temperature, the sensor computes the average summing the value with the measurements recorded in the current hour, and saves it into the table.

```
     Average
-------------
+ average (int)
+ timestamp (datetime)
```

Where for each measurement m, $average_h = \frac{\sum_{i=1}^{n} m_i}{n}$.

ii. The edges have the job of measuring temperature, and the tasks are collecting data, inserting it into the database in a ordered data structure, run the measurement every specific interval of time, and compute the average. The client device runs the program to visualize and find the location with highest average, its task are querying the sensors every hour with a lifetime-based query, send the data in a larger storage, run the algorithm to find the max value, and display the data.

iii. We can use an ordered data structure such as a max-heap, which keeps the highest average as parent node. We can fetch this element in $O(1)$ for every node.

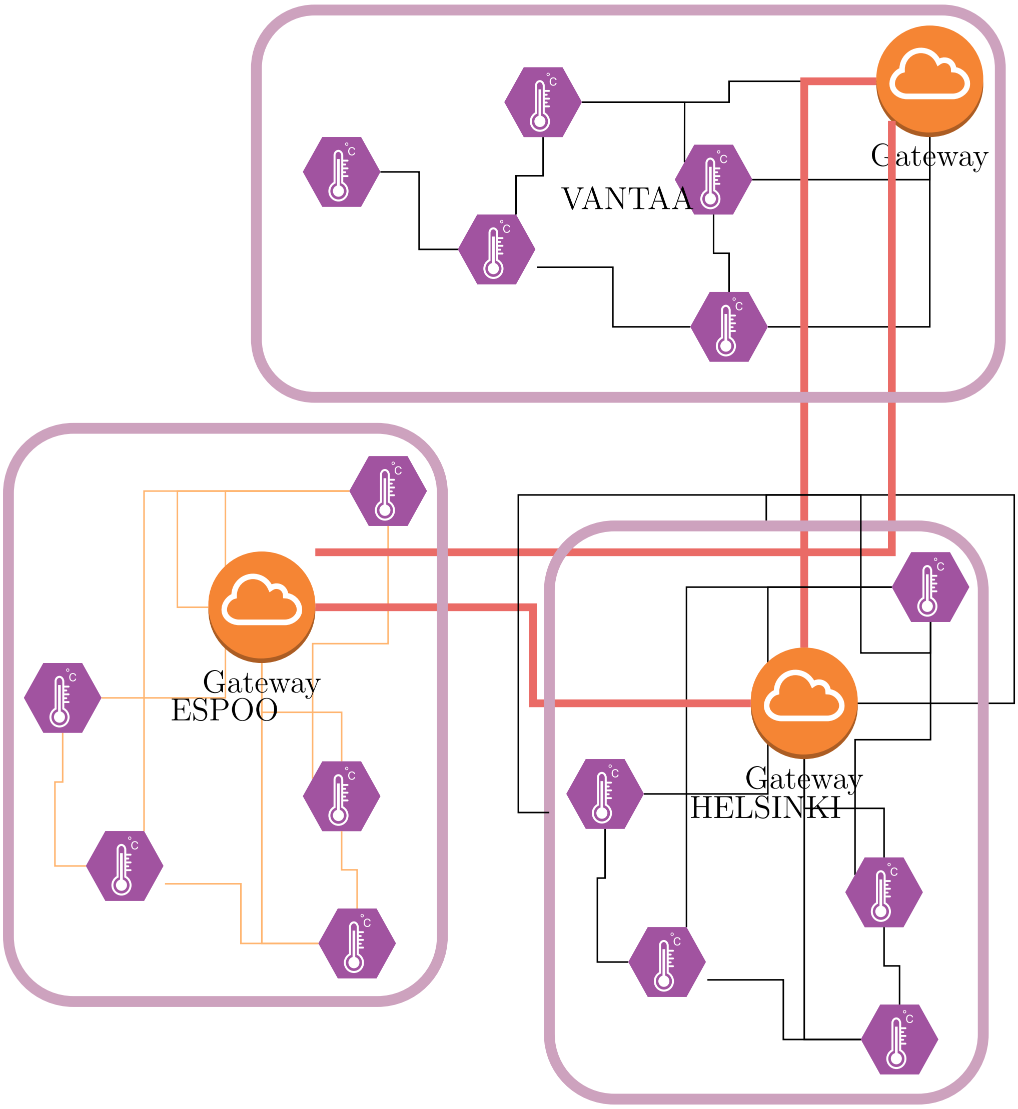

Figure 2: Temperature monitoring, three zones

(b) (i) For each zone, the system is identical to 1, with the difference that each zone has a gateway that is liked with the edges in the area. Every area is independent. The gateway are interconnected in a secondary P2P network, hosting a distributed database that keeps the tables updated with the values from different zones. When users queries the nearest gateway, they can view values from all the zones.

(ii) The database should be distributed, but in my implementation is an extension of the area model, hence it just needed to create another network of coordinator formed by the gateways. A distributed DB is better because the gateway can act as a unique authority, and there is no need to insert a central warehouse as reference to send the data.

(iii) **Volume** it should be defined the sampling time for recording the temperature, as there might be brusque changes in short intervals.

Variety The variety has to be limited to the extremely necessary, namely the temperatures from the three zones of the city.

Velocity The data should arrive in a short amount to provide the users weather forecasting.

Veracity The sensors must record correct data and the platform must keep the value unvaried.