CS 2310: Project Milestone # 2
Multimedia Data Streams

In this milestone, we give the formalization in details of the multimedia data streams problem. The objective of this project is to furnish a formal framework to efficiently design a Multimedia Data Streams (MMDS) schema that achieves an efficient performance in regard with content based retrieval.

Problem Definition:

It is required to generalize the Multimedia Dependency Theory & Multimedia Normalization to be applicable for Multimedia Data Streams (MMDS) and to provide mathematical model for MMDS, based upon which, we can draw some performance guarantees such as minimum output rate per query, maximum supported queries per site, and maximum supported streams per site.

This involves designing the following:

- A Mathematical Model
- A Framework to efficiently utilize MDFs to execute CQs on MMDS.
- A Framework to efficiently archive MMDS for post-event processing
- A framework of evaluation
- A Query Language ➔ Rohit's Project

In this mile stone, we furnish the mathematical model of the multimedia data streams.

Mathematical Model:

In this section we will give some definitions then we will provide the mathematical model of the Multimedia Data Streams problem. At the end, we will try to give a concrete example.

Definitions:

First, we need to define what we mean by a multimedia data stream first. This what constitutes the first set of definitions (Definition 1 through 6). Second, we define some mathematical variables to used in the mathematical model.

Definition 1) Multimedia Data Streams: A Multimedia data stream (MMDS) is a data stream (as defined later) that contains at least one *Micon* (as defined later) as one of its attributes, according to a certain schema.

Please note that the data stream schema refers to the specification of the attributes'/fields' names, and their types that constitutes any tuple that belongs to that stream, as well as the relationships and/or interactions among different streams. In a sense, a stream in the data

streams model is comparable to a relation in the relational database model. However, it should be noted that a relation could be easily modeled as a stream. This will be clarified after we define the data stream below.
An example of a multimedia stream is the video stream captured by some security camera. It would follow the following schema:

Video (frame-number, time-stamp, one-frame-of-video-data, location-id)

This is further illustrated in the Figure 1 below.

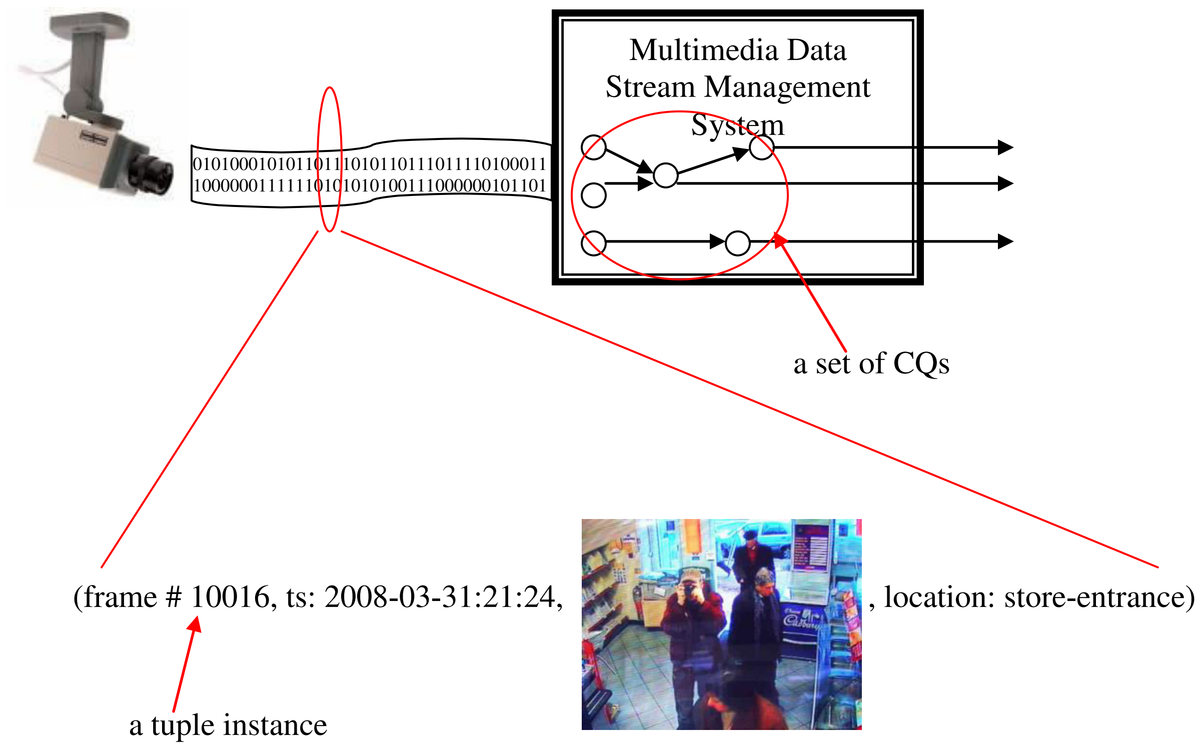

Figure 1: A Multimedia Stream illustration

Now, let us review what is a data stream, and what is a multimedia icon (*Micon*).

Definition 2) Data Stream: A Data Stream is a huge sequence of tuples according to a certain schema that keeps arriving to a Data Stream Management System. Tuples could be of one out of three types: add, delete and the update tuples. Each tuple has both a unique identifier and a timestamp that is used to order the tuples.

Definition 3) Micon: A Micon is a multimedia icon that could be: text (ticon), still image (icon), audeio (earcon), video (vicon), or a multiple of the previous (Multicon).

Generalization or restriction? The question now: does a multimedia data stream generalize the data stream, or the other way around? In fact, the multimedia data stream is a special case of any data stream. However, the MMDSMS allows both multimedia data streams, and non-multimedia (regular) data streams. Hence, the model of the streams that are allowed to be registered in a MMDSMS should be of the general type.

Now, we will furnish some definitions related to a Multimedia Data Stream Management System (MMDSMS), then we will provide some mathematical definitions that will be used in the model we are proposing.

Definition 4) Continuous Query: A Continuous Query (CQ) is a query registered by a user at the MMDSMS, that is to be executed - theoretically - forever. If the CQ include one or more multimedia operators (transformation or fusion operators – as defined in Rohit's project) then the CQ is called a multimedia CQ (*m-CQ*).

Since a data stream in naturally unbounded in size, some query operators can not be applied except for a certain portion (window) of the data, such as aggregate and join operators. Such operators are called windowed operators, and are defined as follows.

Definition 5) Windowed Operators: A windowed operator is to be executed over a certain portion of the data stream (window) as opposed to the whole stream. The window is used to bound an unbounded computation, such as a join of 2 streams or an average (or any aggregate) function of a stream. The window is specified using a range (size) and a slide (step). A window could be time-based, or tuple-based and could be sliding (if the step is less than the size), or tumbling (if the step equals the size).

Tuple-based window is a window specified in terms of number of tuples. While a time based window is specified in terms of time units. Note that it is easier to deal with tuple-based windows, since the size of each window in terms of number of tuples is fixed, while in time-based windows, each window instance size vary based on the arrival process. Now we are ready to define a MMDSMS.

Definition 6) A Multimedia Data Stream Management System: A MMDSMS is a virtual center (could be physically distributed) that receives the tuples of all streams registered within it (both multimedia streams and regular streams). Users use the MMDSMS's user interface to register a set of CQs and the *m-CQ*s. The MMDSMS responsibility is to continuously process the tuples under real-time constraints with respect to the registered CQs, and to disseminate the results back to the users.

The architecture of the MMDSMS is illustrated in Figure 2 below.

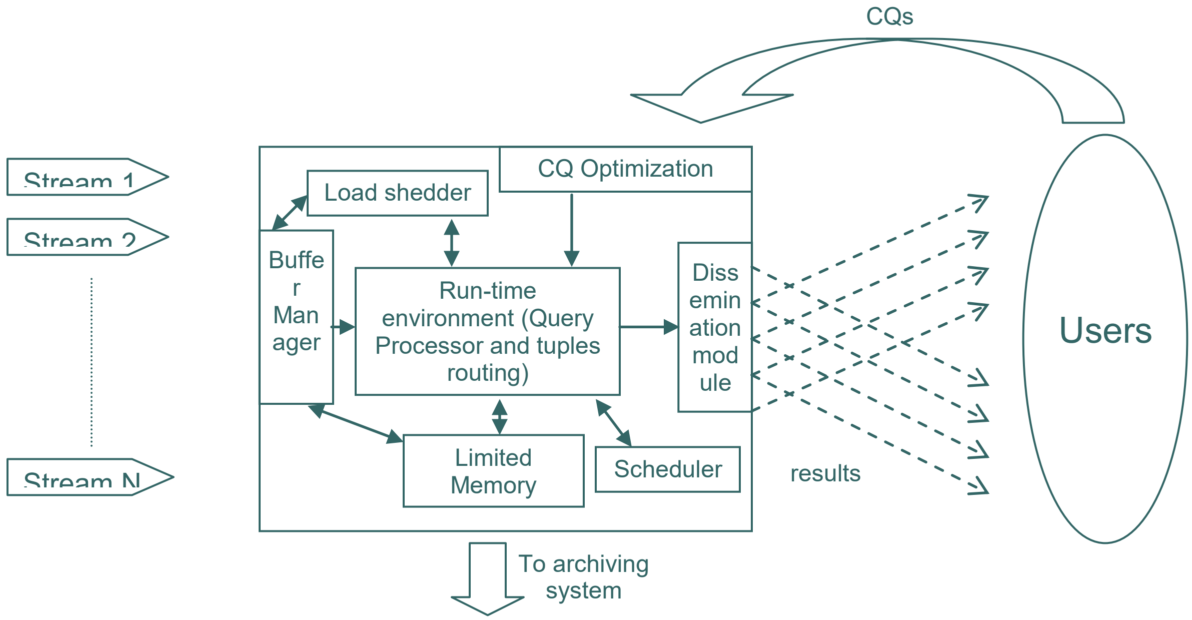

Figure 2: MMDSMS Architecture

We are now ready to give some mathematical definitions:

Definition 7) Multimedia Tool Box of Type 1: A multimedia tool box of type 1 (*mtb1*) is some black box (ready made) tool box for processing the multimedia data of "any" type, with the property that the performance is optimized with respect to Quality of Service (QoS); which is denoted by the processing time here.

Definition 8) Multimedia Tool Box of Type 2: A multimedia tool box of type 2 (*mtb2*) is some black box (ready made) tool box for processing the multimedia data of "any" type, with the property that the performance is optimized with respect to Quality of Data (QoD); which is denoted by the quality of the multimedia objects that belongs to the output stream.

Definition 9) Mathematical terms/variables/Notation:

- Let S be a schema, which is a set of streams and multimedia streams S:$\{S_i\}$

- Each stream S_i consists of a set of attributes, each is of a certain predefined type T_i, where the type could be any traditional data type (such as number, string, Boolean, timestamp, etc) or any multimedia data type (*Micon*), such as: *icon*, *earcon*, *vicon*, etc. Formally:

$$S_i = \{X_i \mid 0 \le i \le n \text{ and } X_i \text{ is of Type } T_i\}$$

- A multimedia continuous query (*m-CQ*) is a computational structure/network/graph, that consists of edges, operators, and at least one input multimedia stream, and exactly one output stream.

  $$m\text{-}CQ: (S_1, S_2, \ldots S_k) \rightarrow S_{new}$$

  Where $m\text{-}CQ = G(V,E)$; that is the *m-CQ* is basically a DAG (Directed Graph), where V (the set of vertices) denotes the operators, and E (the set of edges) define the flow of data. The set V must contain at least one multimedia operator (conversion (σ-operator) or fusion operator (ϕ-operator)).

- Let C_i = Cost ($m\text{-}CQ_i$) denotes the total cost in terms of CPU time consumed to process a single tuple through all operators of $m\text{-}CQ_i$.

- Let S_i = Selectivity ($m\text{-}CQ_i$) denotes the selectivity of $m\text{-}CQ_i$. Where selectivity is defined as the probability of producing one output tuple for processing one input tuple. This is usually less than one, since each tuple is typically matched against some predicates in each operator. However, some special operators (such as Join operators) have selectivity greater than one.

- Let r_i denotes the average response time of tuples produced by $m\text{-}CQ_i$ where the response time is defined as the time span between when the input tuple was first available for processing in the input buffers, and when the corresponding tuple was available for dissemination in the output buffers.

- Let R_i denotes the output rate of producing output tuples of $m\text{-}CQ_i$

Now, we are ready to formulate our problem.

<u>Problem Formalization:</u>

Given a set of streams, with stream S_i a certain multimedia stream of a certain schema $\{X_i\}$, with X^m_i of a *Micon* type, that is T^m_i= *Micon*, and given a Computational Network which constitutes the optimized query execution plan for a set of *CQ*s and *m-CQs*, this Computational Network is composed of both data processing operators (selection, projection, and join) and the special multimedia operators (σ-operator & ϕ-operator), it is required to compute a lower bound on the quality of service provided, measured by the average output rate R_i of each query.

Discussion:

Since there is typically a tradeoff between QoS and QoD metrics, we will assume an adaptive processing algorithm that given a QoS threshold (t_{qos}) and a QoD threshold (t_{qod}), the algorithm starts utilizing mtb1, then when the QoD gets below t_{qod}, the processing switches automatically to *mtb2*. Then as QoS gets below t_{qos}, the algorithm switches back to *mbt1*.

Thus, we need to calculate the following bounds

1- Assuming a certain total cost of a given computation network (*C*), what are the bounds of the output rate, as a QoS.
2- Assuming a desired minimum QoS (output rate) for a given computation network, what is the bound on the number of data streams (maximum possible number of streams to be supported).
3- Similarly, and complementary, given a desired minimum QoS, and a given set of registered multimedia data streams, what is the bound on the complexity of the computation network (i.e. the cost *C*).

If the proposed model is valid, the challenge will be in incorporating the adaptive processing algorithm (switching between the 2 multimedia tool boxes: mtb1 & mtb2) into the formulas.

A concrete example:

Consider the following scheme:

Imagine a stream of video frames captured from a certain security monitoring camera :

```
Video (frame-number, time-stamp, one-frame-of-video-data,
                     location_id)
```

The frame-number attribute here becomes the unique identifier of the video frame (a tuple in the stream).

A derived relation can be computed from the above relation, for example,

```
Aircraft (frame-number, one-frame-of-video-data, type-of-
                        aircraft)
```

There is a multimedia dependency from video-data to type-of-aircraft. The dependency relations tell us how these streams are related.

A possible *m-CQ*$_1$ could be:

```
Notify me every 30 seconds of
all video frames within the
past 30 seconds that contains a
weapon that you receive at
Video stream, from the entrance
or the exit locations.
```

Clearly this *m-CQ* includes the following operations (or manipulations) over the Video multimedia streams:

1) Tumbling Window: a window of size 30 sec, and step 30 sec should be first defined
2) Over the previous window, there is a filter that filters out the frames not coming from the entrance or exit locations. Or in data base language, a Select operator with predicate set such that the location_id field equals to that of the entrance, or that of the exit sites.
3) For all frames that go through the above 2 operators, some mtb1 or mtb2 should be utilized to detect weapon objects. This is considered a transformation operator (σ-operator).

The corresponding Query Graph G($m\text{-}CQ_1$) (or the Computational structure in other words) is illustrated in Figure 3.

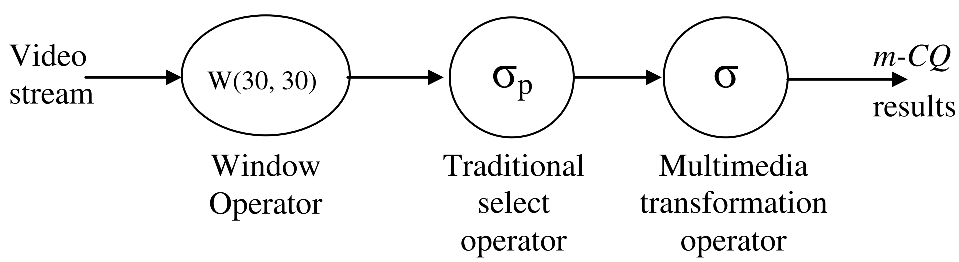

Figure 3: Illustration of $m\text{-}CQ_1$

Now, assuming the cost of this computational network/structure is C_1 then, the MMDSMS can provide the following guarantees:

1) If this is the only stream registered, and this is the only m-CQ registered, then the QoS (output rate) is guaranteed to be:

$$R_1 = \frac{S_1}{C_1}$$

   Where S_1 is the selectivity of the *m-CQ*, and C_1 is the cost.
2) Assuming a desired minimum QoS of R_1 then the maximum possible number of streams that could be supported is limited by the relation that:

$$\text{overhead} = \frac{S_1}{R_1} - C_1$$

Where the overhead represent the additional cost the additional streams processing might induce.

3) Similarly, assuming this stream to be the only stream, then the maximum number of queries is bounded with the same exact relation above (in bound 2). The difference be in the way this overhead is calculated. In the bound of number of streams, the overhead should reflect the cost of adding support for new streams, while here the overhead is basically the cost of the new operators needed to process the additional m-CQs.

+ + +

Another possible $m\text{-}CQ_2$ over the MMDS schema given above, could be:

```
Notify me every minute of all
video frames within the past 10
seconds that contains an
aircraft of type "XYZ".
```

Clearly this *m-CQ* includes similar window operator. However, filtering here requires an additional join operator. Here the multimedia dependency plays a role. Since there is a dependency between the video data, and the aircraft type, we can simply use the derived stream, instead of the original stream, and then we can join based on the *frame_number* attribute to get the rest of the data. The query is illustrated in the figure below.

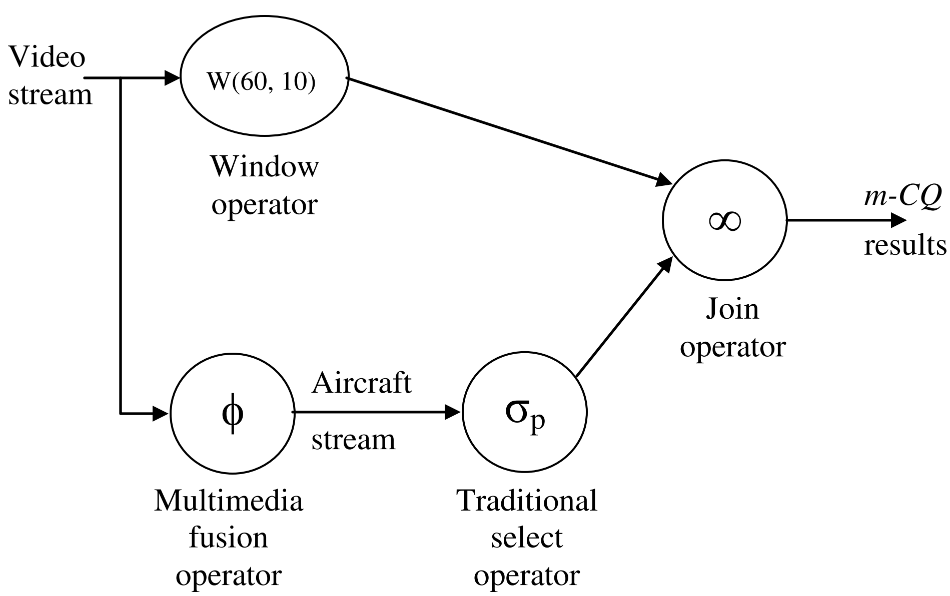

Figure 4: Illustration of m-CQ2