

Data-quality Guided Load Shedding for Expensive In-Network Data Processing^{*}

Lina Peng K. Selçuk Candan

Department of Computer Science and Engineering
Arizona State University
Tempe, AZ, 85287-8809, USA
{lina.peng, candan}@asu.edu

Abstract

In situ wireless sensor networks, not only have to route sensed data from sources to destinations, but also have to filter and fuse observations to eliminate potentially irrelevant data. If data arrive faster to such fusion nodes than the speed with which they can consume the inputs, this will result in an overflow of input buffers. In this paper, we develop load shedding mechanisms which take into consideration both data quality and expensive nature of fusion operators. In particular, we present quality assessment models for objects and fusion operators and we highlight that such quality assessments may impose partial orders on objects.

1. Introduction

As part of our ARIA project [4, 11], we are developing wired [10] and wireless [15] frameworks to carry out integrated sensing, processing, and routing tasks in a distributed media processing workflow. To maximize system resource utilization, a wireless network has to leverage the inherent redundancy in the sensor network in terms of (a) alternative observation sources, (b) alternative ways to route a message, and (c) alternative ways to process, filter, and fuse data in the network. In general, different alternatives may have different qualities [5, 8]. For instance, the quality of an observation depends on the power available on an adaptive sensor node. Similarly, the quality of a derived (or fused) observation may depend on the density [8], coverage, and power status of a set of sensors covering an observation area. Therefore redundancy in the network is both an advantage (providing alternative data points and fault tolerance) and also a challenge (with increased transmission and processing costs and higher variability).

If the input data arrive faster than the speed with which network nodes can consume (filter, fuse, and route) them, this will result in the overflow of the input queues. In such a case, the node will need to shed (or discard) some of

the input data. Most existing work on semantic load shedding [6, 12, 14] (see Section 5 for the related work) on the other hand tackle the overflow problem in stream databases by aiming to maximize the throughput or at last provide a random sample of the results. In many in-network data processing applications, such as ARIA [11], however, the bottleneck is the processing power of the operators and, in many cases, the throughput is constant. Therefore, since the input data and the fusion operators can be imprecise, shedding must improve the qualities of the output data objects rather than trying to maximize the throughput.

In in-network data processing the qualities of the input objects depend on how they were collected and processed before arriving to the input queues. Due to the redundant ways to sense the environment and the alternative ways to route, process, filter, and fuse data in the network, data quality is inherently variable. Thus, load shedding mechanisms must be cognizant of the quality assessments of the input objects. In this paper, we focus on the quality and throughput constraints of fusion nodes. In particular, we introduce quality-assessment based load shedding, suitable for sensory environments with varying qualities of data. Therefore, we provide fusion operator and object quality assessment models (Sections 2 and 3.1). We highlight that such quality assessments may impose partial orders on objects (Section 3.2) and identify a class of quality assessment merge functions for typical fusion operations (Section 3.3).

2. Fusion Operator Model

Fusion operators have rich analysis, aggregation, and filtering semantics. In particular, they may perform complex media processing, information clustering, and data cleaning tasks. Thus, these operators are more similar in nature to the expensive predicates discussed in [1], than the relational stream operators considered in [14]. Without loss of generality, we model a fusion operator as follows:

Definition 2.1 (Fusion Operator (Φ)). *A fusion operator, Φ , has a number of input queues, $in_qu_i, 0 \leq i \leq m$, and an output queue, out_qu . Φ picks m objects (one from each*

^{*}This work has been supported by NSF Grant # IIS-0308268, "Quality-adaptive Media-flow Architectures to Support Sensor Data Management."

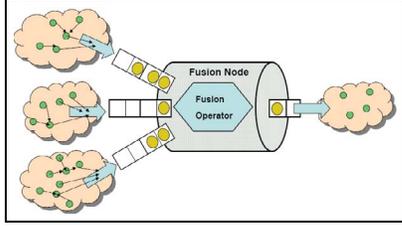


Figure 1. An example of an in-network fusion node.

input queue) and applies a function, f_{Φ} , to the m -tuple of input objects and outputs the result into its output queue.

Figure 1 illustrates a fusion operator which fuses data delivered to its three input queues by the wireless network and outputs fused data into the network. The operator picks a triplet from the input queues and processes them. The fused object is placed in the output queue.

A fusion operator, Φ , has a number of characteristics that determine its runtime properties, including an *input selection model*. This model answers the question “which possible candidate m -tuple in the input window will be processed at a given point in time?” The set of objects from which Φ picks its inputs is generally governed by a time-constraint on the input objects: usually implemented in a time-windowed manner [12]. In this paper, we **do not assume** any specific windowing scheme. Instead, we highlight that, given any scheme, at any given time, there may be more m -tuples for processing than what the fusion operator can handle.

3. Quality, Order, and Rank

As discussed earlier, variations in the data quality can be due to the density of sensors covering the observation area [8] or sensing and processing modules which provide different *quality* levels based on resources [5].

Example 3.1 (Trajectory detection). *Consider an object trajectory detector, which visually tracks an object through cameras and identifies its trajectory. Depending on the observation/sampling rate, the quality (resolution/calibration) of the cameras, and the position of the sensor, the quality of the trajectory labels generated by detector will vary.*

Such trade-offs enable a *graceful degradation* of the sensing and routing services provided by the network, but it also renders the optimization and adaptation challenge. Naturally, quality-aware adaptation requires an *assessment* of the qualities of the objects and the operators.

3.1. Quality Assessment Models

Since the ground truth is not available to judge the qualities of the observations, quality is generally assessed by analyzing the processing characteristics of the operators. If the data is generated through a sensor/operator with a quantifiable quality rate (for instance a function of the available

power), then a scalar-valued quality assessment may be applicable. This is similar to type-1 fuzzy predicates, which (unlike propositional functions which return *true* or *false*) return a membership value between 0 and 1:

Definition 3.1 (Type-1 quality assessment). *In the simplest case, the quality assessment, $qa(o)$, of a given object, o , is modeled as a constant scalar.* \diamond

For example, the quality of an image reading can be quantified in terms of its resolution or the sharpness. Specific usage examples include [5].

A more general quality assessment model would take into account the uncertainties in the quality assessments themselves. These type of predicates, where sets have grades of membership that are themselves fuzzy, are referred to as type-2 fuzzy predicates.

Definition 3.2 (Type-2 quality assessment). *The quality assessment of a given object o is modeled as a normal distribution of qualities, $qa(o) = N_o(q_o, \xi_o)$, where q_o is the expected quality and ξ_o is its variance.* \diamond

Although, the type-2 model can be more general and use different distributions, this specific model (using the normal distribution) relies on the well-known *central limit theorem*, which states that the average of the samples tends to be normally distributed, even when the distribution from which the average is computed is not normally distributed. Various statistical tools, such as *quality control charts*, rely on this. Other quality assessment models are also possible. However, for our purposes in this paper, these two are enough to highlight fundamental differences in different scenarios.

3.2. Comparing and Ranking Assessments

Given a quality assessment model, one way to define the *quality target* for the wireless system is as follows.

Definition 3.3 (Quality Target). *The target of the network is to deliver objects to destinations we can be at least κ confident that the quality of the object arriving to the destination is above a threshold c . Given an object quality assessment model $qa(o)$, this constraint is equivalent to ensuring that the object o satisfies $\int_c qa(o) dqa \geq \kappa$.*

Thus, delivering acceptable objects requires an ability to compare the quality assessments of objects with respect to a given quality target. Below, we show that, a given set, Q , of quality assessments form a totally- or partially-ordered set (Q, \preceq) . Given Q , the standard definitions of equality and order can be adapted to compare its elements:

Definition 3.4 (Equality, Order, Comparability). *Equality ($=$) is a binary relation, such that two quality assessments, qa_1 and qa_2 , are said to be equal iff they are equivalent in some well defined (application depend) sense.*

The binary relation \preceq is said to be a partial order, if given any three quality assessments, qa_1 , qa_2 , and qa_3 , the properties of reflexivity, antisymmetry, and transitivity hold.

Two quality assessments, qa_1 and qa_2 , are said to be incomparable ($qa_1 \not\preceq qa_2$) iff

$$\text{not}((qa_1 \preceq qa_2) \vee (qa_2 \preceq qa_1)).$$

The binary relation \preceq is said to be a total order, if given any three quality assessments, reflexivity, antisymmetry, transitivity and comparability hold.

Definition 3.5 (Comparison of quality assessments). The following are a few samples of comparison relations:

- Let $\forall_i qa_i = N(q_i, \xi_i)$. Then, \preceq_c defined as

$$(qa_i \preceq_c qa_j) \equiv_{def} \int_c^\infty N(q_i, \xi_i) dq \leq \int_c^\infty N(q_j, \xi_j) dq,$$

based on the equality ($=_c$) defined above, is totally ordered. This is because qa_i is mapped to a single scalar value which is totally ordered.

- Let $\forall_i qa_i = N(q_i, \xi_i)$. Then, \preceq defined as

$$(qa_i \preceq qa_j) \equiv_{def} q_j \geq q_i \wedge \xi_j \leq \xi_i$$

is partially ordered. For example, $qa_i = (0.8, 0.1)$ and $qa_j = (0.9, 0.05)$ are comparable by the definition of \preceq , but $qa_i = (0.8, 0.05)$ and $qa_j = (0.9, 0.1)$ are not comparable.

Although the concept of equality and order is application dependent, given an object comparison model, we can identify ranks of assessments in a given set.

Definition 3.6 (Ranking of Quality Assessments). Given a collection (or bag), Q , of quality assessments, a ranking function, $r : Q \rightarrow \mathbb{Z}^+$, is a function which assigns a positive integer to each quality assessment in Q and is anti-monotonic with respect to the quality assessment order:

$$\begin{aligned} \forall qa_i, qa_j \in Q, \quad (r(qa_i) > r(qa_j)) &\Rightarrow (qa_i \preceq qa_j) \vee (qa_i \not\preceq qa_j), \\ \forall qa_i, qa_j \in Q, \quad i \neq j &\Rightarrow r(qa_i) \neq r(qa_j), \\ \forall qa_i \in Q, \quad 1 &\leq r(qa_i) \leq |Q|. \end{aligned}$$

Note that $\forall qa_i, qa_j \in Q$, $(qa_i \prec qa_j) \Rightarrow (r(qa_i) > r(qa_j))$ is implied by the above three conditions. In other words, if a quality assessment is ranked *better* (i.e., its rank value is lower) than another, then the first assessment has either higher or incomparable quality assessment.

Definition 3.7 (Minimum/Maximum Q.A.). Given a collection, Q , of quality assessments, the lowerbound is

$$lbound(Q) = \{qa_i \mid (qa_i \preceq qa_j) \vee (qa_i \not\preceq qa_j), \forall qa_i, qa_j \in Q\}.$$

Given a $lbound(Q)$, $min(Q)$ is a randomly selected quality assessment in $lbound(Q)$.

The upperbound, $ubound(Q)$ and maximum quality assessment, $max(Q)$, are defined similarly.

3.3. Merging Quality Assessments

Quality assessments of the outputs of fusion operations are modeled as *functions* of the assessments of the inputs. Each operator, Φ , has an associated quality assessment merge function, μ_Φ , describing the quality assessment of the output objects in terms of the assessments of the inputs.

Definition 3.8 (Q.A. Merge Function(μ_Φ)). A quality assessment merge function, μ_Φ , of an n -ary operator, Φ , is an n -ary function, such that given n quality assessments, qa_1, \dots, qa_n , $\mu_\Phi(qa_1, \dots, qa_n)$ is also a quality assessment.

Naturally, merge functions depend on the semantics of the fusion operation and can be arbitrarily complex. In the fuzzy logic literature, there are a multitude of functions corresponding to the logical operators, \vee and \wedge , including minimum and product semantics [3, 7].

Definition 3.9 (Q.A. Merge Functions). The following are example quality assessment merge functions:

- Let $0 \leq qa_1, qa_2 \leq 1$ be two quality assessments, then

$$\mu^{min}(qa_1, qa_2) \equiv_{def} \min(qa_1, qa_2)$$

is a merge function. This is the traditional minimum semantics often used in the fuzzy logic literature [7].

- Let $qa_1 = N(q_1, \xi_1)$ and $qa_2 = N(q_2, \xi_2)$ be two quality assessments, then

$$\mu^N(qa_1, qa_2) \equiv_{def} N\left(\frac{q_1 + q_2}{2}, \frac{\xi_1 + \xi_2}{4}\right)$$

is a quality assessment merge function, corresponding to the average semantics used in information retrieval [3].

Depending on the underlying quality assessment model, merge functions can be monotonic or non-monotonic. We adapt the standard definition of monotonicity as follows:

Definition 3.10 (Monotonic Merge Functions). We call an m -ary merge function, μ_Φ , monotonic (given a total or partial order, \preceq , of quality assessments) iff

$$\forall i \in \{1..m\}, qa_{1i} \preceq qa_{2i} \Rightarrow \left(\mu_\Phi(qa_{1i}) \right) \preceq \left(\mu_\Phi(qa_{2i}) \right)$$

Note that the above standard definition of monotonicity does not impose a specific order when incomparable inputs are involved. We next introduce unambiguously monotone merge functions, which specifies that if incomparable inputs are involved, the merged results are also incomparable.

Definition 3.11 (Unambiguously Monotonic M.F.). An m -ary merge function, μ_Φ , is unambiguously monotonic (given a total or partial order, \preceq , of quality assessments), iff it is monotonic and

$$\exists i \in \{1..m\}, qa_{1i} \not\preceq qa_{2i} \Rightarrow \left(\mu_\Phi(qa_{1i}) \right) \not\preceq \left(\mu_\Phi(qa_{2i}) \right)$$

It is known that the fuzzy minimum semantics ($\mu^{min}(qa_i, qa_j)$) is monotonic [7]. When applied to scalars, averaging is also monotonic but when applied to probability distributions, this is not always true:

Example 3.2 (Non-Monotonic Merge Function). Let $qa_i = N(q_i, \xi_i)$, $qa_j = N(q_j, \xi_j)$. Then, the merge function

$$\mu_{\Phi}(qa_i, qa_j) = N\left(\frac{qa_i + qa_j}{2}, \frac{\xi_i + \xi_j}{4}\right)$$

is non-monotonic with respect to \preceq_c in Definition 3.5.

To see this, consider that $\int_c^{\infty} N(q, \xi) dq$ is known to be equal to $1 - F\left(\frac{c-q}{\sqrt{\xi}}\right)$, where the cumulative distribution function, $F(c)$, is defined as $\int_{-\infty}^c N(0, 1) dq$. Therefore, given two quality assessments, qa_i and qa_j , we have

$$\mu_{\Phi}(qa_i, qa_j) = 1 - F\left(\frac{c - \frac{q_i + q_j}{2}}{\sqrt{\frac{\xi_i + \xi_j}{4}}}\right)$$

Now consider three quality assessments, qa_i , qa_j , and qa_k , where $qa_j \preceq_c qa_k$. It is possible to show that $\mu_{\Phi}(qa_i, qa_j)$ is **not** always $\preceq_c \mu_{\Phi}(qa_i, qa_k)$. Let, for instance, the threshold c be 0.8. Let also (q_i, ξ_i) be (0.6, 0), (q_j, ξ_j) be (0.5, 0.4), and (q_k, ξ_k) be (0.6, 0.2), respectively. Since $(c - q_j)/\sqrt{\xi_j} = 0.474 > (c - q_k)/\sqrt{\xi_k} = 0.447$, it follows that $qa_k \preceq_c qa_j$. However, $(c - \frac{q_i + q_j}{2})/\sqrt{\frac{\xi_i + \xi_j}{4}} = 0.79$ and $(c - \frac{q_i + q_k}{2})/\sqrt{\frac{\xi_i + \xi_k}{4}} = 0.894$. Thus, despite the fact that $qa_k \preceq_c qa_j$, $\mu_{\Phi}(qa_i, qa_j) \preceq_c \mu_{\Phi}(qa_i, qa_k)$. Therefore, μ_{Φ} is non-monotone.

4. Future Work: Combination Shedding

In most data flow and streaming architectures, overloads in queues are handled by shedding individual input objects [6, 9, 11, 14]. However, this approach has a big disadvantage when the goal is to maximize the quality of the fusion results: when an object is removed from a queue, then that object cannot join/fuse with any objects in the other queues. Thus, when we consider the assessments of the incoming objects, a more desirable approach would require shedding of (not the objects) but combinations of objects. The challenge of course is to pick the least promising input combinations to eliminate from consideration; and perform this in real-time. We note that when the operators are expensive, there is a significant gap between the time needed for processing input combinations for fusion and for picking combinations for elimination. Unlike existing systems [6, 9, 11, 14], future work will involve leveraging this gap to evaluate and pick combinations for shedding.

In [7], Fagin proposes a set of efficient top- K query execution algorithms for databases with monotonic fuzzy queries. The fusion operations we discuss in this paper have (time-constrained) cross-product semantics. Furthermore, we discuss how to establish rankings when incomparable

inputs are involved. Merging of such partially ordered inputs have been considered in the security domain [2]; however, unlike the work in that area (where there are well-defined mappings between security levels), fusion in wireless networks involves merging of all objects within a given time window.

5. Related Work

In [9], an object-based load shedding scheme is adopted when the input rate exceeds the service rate of the join operators. Semantic load shedding is introduced in [6]; the statistics of the data values arriving in the input data streams are used for maximizing the number of resulting tuples. In [13, 14], the contents of the tuples are also considered for load shedding; the data values with the low utilities are filtered out according to QoS specifications. In [11], we introduced confidence-based early object elimination schemes deployed within ARIA.

6. Conclusion

In this paper, we highlighted that in-network data fusion operators are expensive, quality assessments of the objects are highly variable, and the goal is not to maximize the number of the output results, but to maximize their qualities. With these in mind, we presented quality-assessment models for objects and fusion operators. We showed that in many cases quality assessments are partially ordered and unambiguously monotonic ranking functions may not exist.

References

- [1] R.Avnur and J.Hellerstein. *Eddies: Continuously Adaptive Query Processing*. SIGMOD, 2000.
- [2] P. Bonatti, M.L. Sapino, V.S.Subrahmanian. *Merging Heterogeneous Security Orderings*. J. of Comp. Security 5(1): 3-30, '97.
- [3] K.S. Candan, W.-S. Li. *On Similarity Measures for Multimedia Database Applications*. KAIS 3(1): 30-51, 2001.
- [4] K. S. Candan, G. Kwon, L. Peng, and M.L.Sapino. *Modeling Adaptive Media Processing Workflows*. ICME 2006.
- [5] Y. Chen and H. Sundaram, *A Computationally Efficient 3D Shape Rejection Algorithm*, ICME 2005.
- [6] A. Das, J. Gehrke, and M. Riedewald. *Approximate Join Processing Over Data Streams*, SIGMOD 2003.
- [7] R. Fagin, *Fuzzy Queries in Multimedia Database Systems*, PODS 1998.
- [8] C. Intanagonwiwat, et al. *Impact of Network Density on Data Aggregation in Wireless Sensor Networks*. ICDCS 2002: 457-458
- [9] J. Kang, J. Naughton, and S.D. Viglas, *Evaluating Window Joins over Unbounded Streams*, VLDB 2002.
- [10] G.Kwon and K.S.Candan. *DANS: Decentralized, Autonomous, and Network-wide Service Delivery and Multimedia Workflow Processing*. ACM MM06
- [11] L. Peng and K. S. Candan. *Confidence-driven Early Object Elimination in Quality-Aware Sensor Workflows*, DMSN05.
- [12] U.Srivastava and J.Widom. *Memory-Limited Execution of Windowed Stream Joins*. VLDB 2004.
- [13] N. Tatbul and S. Zdonik. *Window-Aware Load Shedding for Aggregation Queries over Data Streams*, VLDB 2006.
- [14] N. Tatbul et al. *Load Shedding in a Data Stream Manager*, VLDB 2003.
- [15] S. Wu and K. S. Candan. *GMP: Distributed Geographic Multicast Routing in Wireless Sensor Networks*. ICDCS 2006.