

Bounds for Overlapping Interval Join on MapReduce

Foto Afrati^{*}
National Technical University
of Athens, Greece
afrati@softlab.ece.ntua.gr

Shlomi Dolev[†] and
Shantanu Sharma
Ben-Gurion University of the
Negev, Israel
{dolev,sharmas}@cs.bgu.ac.il

Jeffrey D. Ullman
Stanford University
USA
ullman@cs.stanford.edu

ABSTRACT

We consider the problem of 2-way interval join, where we want to find all pairs of *overlapping intervals*, *i.e.*, intervals that share at least one point in common. We present lower and upper bounds on the replication rate for this problem when it is implemented in MapReduce. We study three cases, where intervals in the input are: (i) unit-length and equally-spaced, (ii) variable-length and equally-spaced, and (iii) equally-spaced with specific distribution of the various lengths. Our algorithms offer intuition as how to build algorithms for other cases, especially when we have some statistical knowledge about the distribution of the lengths of the intervals. E.g., if mostly large intervals interact with small intervals and not within themselves, then we believe our techniques can be extended to achieve better replication rate.

1. INTRODUCTION

MapReduce [3] is a programming model used for parallel processing of large-scale data. A *mapper* is an application of a (user-defined) map function to a single input and provides outputs in the form of $\langle key, value \rangle$ pairs. A *reducer* is an application of a (user-defined) reduce function to a single *key* and its associated list of *values*. The *reducer capacity* — an important parameter — is an upper bound on the sum of the total number of inputs that are assigned to the reducer. We denote the capacity of a reducer by q , and all the reducers have an identical capacity. Interval join using MapReduce was introduced by Chawda et al. [2].

Example: Employees involved in the phases of a project. We show an example to illustrate temporal relations (a relation that stores data involving timestamps), intervals, and the

^{*}Supported by the project Handling Uncertainty in Data Intensive Applications, co-financed by the European Union (European Social Fund) and Greek national funds, through the Operational Program “Education and Lifelong Learning,” under the program THALES

[†]Supported by the Rita Altura Trust Chair in Computer Sciences, Lynne and William Frankel Center for Computer Sciences, Israel Science Foundation (grant 428/11), the Israeli Internet Association, and the Ministry of Science and Technology, Infrastructure Research in the Field of Advanced Computing and Cyber Security.

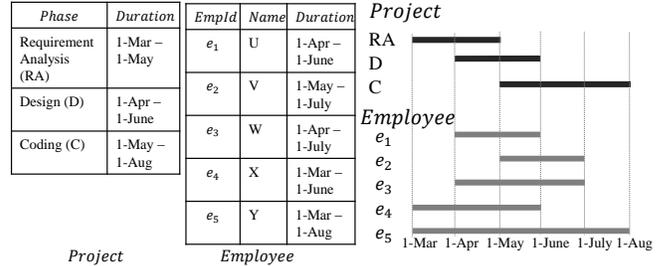


Figure 1: Two temporal relations ($Project(Phase, Duration)$ and $Employee(EmpId, Name, Duration)$) and their representation on a time diagram.

need for interval join of overlapping intervals. Consider two (temporal) relations (i) $Project(Phase, Duration)$ that includes several phases of a project with their durations, and (ii) $Employee(EmpId, Name, Duration)$ that shows data of employees according to their involvement in the project’s phases and their durations; see Figure 1. Here, the duration of a phase or the duration of an employee’s involvement in a phase is given by an interval. It is interesting to find all the employee that are involved in a phase of the project. Formally, a query: find the name of all employees who worked in a phase of the project; requires us to join the relations to find all overlapping intervals of the relations. For example, the answer to the query includes employees U with id e_1 , W with id e_3 , X with id e_4 , and Y with id e_5 are involved in RA phase the project.

Problem Statement. We consider the problem of interval join of overlapping intervals, where two relations X and Y are given. Each relation contains binary tuples that represent intervals, *i.e.*, each tuple corresponds to an interval and contains the starting point and ending point of this interval. Each pair of intervals $\langle x_i, y_j \rangle$, where $x_i \in X$ and $y_j \in Y$, $\forall i, j$, such that intervals x_i and y_j share at least one common time, corresponds to an output.

A MapReduce job can be described by a mapping schema. A *mapping schema*, for this problem, assigns each interval to a number of reducers (via the formation of key-value pairs) so that (i) for each output (*i.e.*, pair of overlapping intervals), there exists a reducer that receives the corresponding pair of overlapping intervals that participate in the computation of this output and (ii) each reducer has a capacity (denoted by q hereon) that constrains the total number of intervals assigned to this reducer. The replication rate of a mapping schema is the average number of key-value pairs for each interval and is a significant performance parameter in a MapReduce job. We analyze here lower and upper bounds on the replication rate for the problem of overlapping intervals.

Our Contribution. We provide lower and almost matching

upper bounds for three cases: (i) unit-length and equally-spaced (Section 3), (ii) variable-length and equally-spaced, and (iii) equally-spaced with specific distribution of the various lengths (Section 4.1). In the third case, we assume that one set contains only small intervals and the other set only large intervals. We offer an algorithmic simple technique that takes advantage of this knowledge to build an algorithm that improves the replication rate of the second case above.

Related Work. Several types of join operations and a detailed review of join algorithms for temporal relations are given in [4]. MapReduce-based 2-way and multiway interval join algorithms of overlapping intervals *without regarding the reducer capacity* are presented in [2]. However, the analysis of a lower bound on replication of individual intervals is not presented; neither is an analysis of the replication rate of the algorithms offered therein.

2. THE SETTING

A (time) interval, i , is represented by a pair of times $[T_s^i, T_e^i]$, $T_s^i < T_e^i$, where T_s^i and T_e^i show the *starting-point* and the *ending-point* of the interval i , respectively. $T_s^i - T_e^i$ is the *length* of the interval i . Two intervals, say interval i and interval j are called *overlapping intervals* if the intersection of both the interval is nonempty.

Mapping Schema. A mapping schema is an assignment of overlapping intervals to some given reducers under the following two constraints: (i) a reducer is assigned only q intervals, and (ii) for each output, we must assign the corresponding intervals to at least one reducer in common.

Replication rate, r : The *replication rate* [1] is the average number of key-value pairs created for an interval.

3. UNIT-LENGTH AND EQUALLY-SPACED INTERVALS

Two relations X and Y , each of n unit-length intervals are given. We assume that all the intervals have their starting-points in a closed interval $[0, k]$, *i.e.*, there is no interval that starts before 0 or after k . Thus, the space between every two successive intervals is $\frac{k}{n} < 1 \ll k$. In other words, the first interval starts at time 0, the second interval starts at time $\frac{k}{n}$, the third interval starts at time $\frac{2k}{n}$, and the last n^{th} interval starts at time $k - \frac{k}{n}$; see Figure 2.

The output we want to produce is a set of all pairs of intervals such that one interval overlaps with the other interval in the pair. The problem is not really interesting if all these intervals exist on the input. The real assumption is that some fraction of them exist, and the reducer capacity q is selected so that the expected number of inputs that actually arrive at a given reducer is within the desired limits, *e.g.*, no more than what can be processed in main memory. In addition, the case of unit-length and equally-spaced interval is not realistic, but is explored because it gives us an idea of what optimal algorithms for more general and more realistic cases would look like.

A *solution* to the problem of interval join of overlapping unit-length and equally-spaced intervals is a mapping schema that assigns each interval of the relation X with all its overlapping intervals of the relation Y to at least one reducer in common, without exceeding q . Since every two consecutive intervals have an equal space ($\frac{k}{n}$), an interval $x_i \in X$ overlaps with at least $2\lfloor \frac{1}{\frac{k}{n}} \rfloor + 1 = 2\lfloor \frac{n}{k} \rfloor + 1$ intervals of Y , where at least $\lfloor \frac{n}{k} \rfloor$ intervals of the relation Y have their ending-points between the starting-point and the ending-point of x_i , at least $\lfloor \frac{n}{k} \rfloor$ intervals of the relation Y have their starting-points between the starting-point and the ending-point of x_i , and an interval $y_i \in Y$ that have identical end-points as x_i (this inequality does not true for the



Figure 2: An example of unit-length and equally-spaced intervals, where $n = 9$ and $k = 2.25$.

intervals that have starting-points before 1 and after $k - 1$). In this section, we will show a lower bound on the replication rate for interval join of overlapping unit-length and equally-spaced intervals. After that, we provide an algorithm, its correctness, and an upper bound on the replication rate obtained by the algorithm.

Theorem 1 (Minimum replication rate) *For two relations, X and Y , of unit-length and equally-spaced intervals, the minimum replication of an interval, for joining each interval of the relation X with all its overlapping intervals of the relation Y , is (i) at least 2 when $2n > q \geq 2\lfloor \frac{n}{k} \rfloor + 2$, and (ii) at least $\frac{2}{q}\lfloor \frac{n}{k} \rfloor$ when $2 < q < 2\lfloor \frac{n}{k} \rfloor + 2$, where each relation holds n intervals, q is the reducer capacity, and k denotes that the starting points of intervals are in $[0, k]$.*

PROOF. First, we consider the case of $2n > q \geq 2\lfloor \frac{n}{k} \rfloor + 2$. When $q \geq 2n$, a single reducer is enough to hold all the intervals of both the relations, and hence, the reducer is able to provide all output pairs (of interval join of overlapping intervals). When $2\lfloor \frac{n}{k} \rfloor + 1 < q < 2n$, a single reducer may hold an interval $i \in X$ and all its $2\lfloor \frac{n}{k} \rfloor + 1$ corresponding overlapping intervals of the relation Y , and such a reducer is enough to provide all-pairs of the interval i with its overlapping intervals. However, at the same time, there must be at least a single interval, say interval j , that is assigned to the same reducer where the interval i is assigned, but the interval j is not assigned with all its corresponding overlapping intervals. Hence, the interval j must be assigned to at least one more reducer to be coupled with all its $2\lfloor \frac{n}{k} \rfloor + 1$ overlapping intervals. Therefore, the minimum replication of an interval is at least 2.

Now, we consider the case of $2 < q < 2\lfloor \frac{n}{k} \rfloor + 2$. Consider an interval i . Since the interval i has at least $2\lfloor \frac{n}{k} \rfloor + 1$ overlapping intervals, all these $(2\lfloor \frac{n}{k} \rfloor + 2)$ intervals cannot be assigned to a single reducer. The interval i can share a reducer with at most $q - 1$ ($< 2\lfloor \frac{n}{k} \rfloor + 1$) intervals (of the relation Y). In order to assign the interval i with all the remaining overlapping intervals, it is required to assign subsets of the $2\lfloor \frac{n}{k} \rfloor + 1$ intervals, each subset with at most $q - 1$ intervals. Such an assignment results in at least $2\lfloor \frac{n}{k} \rfloor + 1 / q - 1$ subsets of $2\lfloor \frac{n}{k} \rfloor + 1$ overlapping intervals. Thus, the interval i must be sent to at least $2\lfloor \frac{n}{k} \rfloor + 1 / q - 1 > \frac{2}{q}\lfloor \frac{n}{k} \rfloor$ reducers. \square

Algorithm 1. We propose an algorithm for interval join of overlapping intervals, where two relations X and Y (each is of n intervals of unit-length and equally-spaced) are inputs. Recall that it is expected that not all possible intervals are present.

We divide the time-range from 0 to k into equal-sized partition of length $w = \frac{q-c}{3\lfloor \frac{n}{k} \rfloor}$, where $c = \lfloor \frac{n}{k} \rfloor + 2$. Consider that by partitioning of the time-range, we have P partitions. We now arrange P reducers, one for each partition. We consider a partition p_i , $1 \leq i \leq P$, and assign all the intervals of the relation X that exist in the partition p_i to the i^{th} reducer. In addition, we assign all the intervals of the relation Y that have their starting or ending-point in the partition p_i to the i^{th} reducer.

Explaining pseudocode of Algorithm 1. A mapper takes an interval $x_i \in X$ (line 2) and produces $\langle key, value \rangle$ pairs (line 4). The *key* represents a partition where the interval x_i exists and the

Cases	Solutions	Theorems	Replication rate
The lower bounds			
Unit-length and equally-spaced intervals		1	$2 \text{ or } \frac{2}{q} \lceil \frac{n}{k} \rceil$
Variable-length and equally-spaced intervals		3	$2 \text{ or } \frac{2}{q} \lceil \frac{l_{min}}{s} \rceil$
The upper bounds			
Unit-length and equally-spaced intervals	Algorithm 1	5	$\frac{3}{qT-s} \frac{s}{2}$
Variable length and equally-spaced (big-small) intervals	Algorithm 2	5	$\frac{3}{qT-s} \frac{s}{2}$
Variable length (different-length) and equally-spaced intervals	Algorithms 3 and 4	5	$\frac{3}{qT-s} \frac{s}{2}$

Table 1: The bounds for interval joins of overlapping intervals.

Algorithm 1: 2-way interval join algorithm for overlapping intervals of unit-length and equally-spaced intervals.

Inputs: X and Y : two relations, each is of n intervals.

Variables: k : A point on the timeline after that no interval can have a starting-point; w : The length of a partition $w = \frac{q-c}{3\lceil n/k \rceil}$, where $c = \lceil \frac{n}{k} \rceil + 2$; P : The total number of partitions and reducers.

- 1 Partition the time-range into P partitions, each of length w
- 2 **Function** $Map_for_X(x_i \in X)$ **begin**
- 3 $z \leftarrow count_partitions(x_i)$
- 4 **for** $j \leftarrow 1$ **to** z **do** $emit(j, x_i)$;
- 5 **Function** $Map_for_Y(y_i \in Y)$ **begin**
- 6 $sp \leftarrow starting_points(y_i), ep \leftarrow ending_points(y_i)$
- 7 $emit(sp, y_i), emit(ep, y_i)$
- 8 **Function** $reduce(\langle key, list_of_values \rangle)$ **begin**
- 9 **for** $j \leftarrow 1$ **to** P **do**
- 10 Reducer i is having
 $\langle i, list_of_values[x_a, x_b, \dots, y_a, y_b, \dots] \rangle$
- 11 Perform interval join over overlapping intervals
- 12 **Function** $count_partitions(x_i)$ **begin**
- $c \leftarrow$ Count the total number of partitions that x_i crosses
- return** c

total number of $\langle key, value \rangle$ pairs for the interval x_i depends on the total number of partitions that the interval x_i crosses, by calling function $count_partitions()$ (lines 3 and 12). Also, a mapper processes an interval $y_i \in Y$ (line 5) and produces at most two $\langle key, value \rangle$ pairs (line 7), where the first pair and the second pair are corresponding to a partition where y_i has the starting-point and the ending-point, respectively (line 6). The *value* represents the interval x_i or y_i itself. In the reduce phase, a reducer i fetches all the intervals of the relations X and Y that have a *key* i (line 10) and provides the final outputs, line 11.

Theorem 2 (Algorithm correctness) *Let $c = \lceil \frac{n}{k} \rceil + 2$ and let $q = 3w\lceil \frac{n}{k} \rceil + c$. Algorithm 1 assigns each pair of overlapping intervals to at least one reducer in common, where each relation, X and Y , holds n intervals, q is the reducer capacity, k denotes that the starting points of intervals are in $[0, k]$, and w is the length of a partition.*

PROOF. Since every two successive intervals have $\frac{k}{n}$ spacing, an interval $i \in X$ can overlap with at most $2\lceil \frac{n}{k} \rceil$ intervals of the relation Y . First, we consider $w < 1$; in a partition, p of length w , an interval i can overlap with at most $2w\lceil \frac{n}{k} \rceil$ intervals of the relation Y . Note that there are at most $w\lceil \frac{n}{k} \rceil$ intervals (of the relation X) that have their starting-points after the starting-point

of the interval i in the partition p , and we called these intervals *post-intervals* of the interval i . Also, there are at most $c = \lceil \frac{n}{k} \rceil$ intervals (of the relation X) that have either their ending-points in the partition p or cross the partition p ; we call these intervals *pre-intervals* of the interval i .

Thus, for $w < 1$, $q = 3\lceil \frac{n}{k} \rceil + c$, we can assign the interval i , post-intervals of i that lie in the partition p , and pre-intervals of i that lie in partition p at a single reducer. Such an assignment occupies $w\lceil \frac{n}{k} \rceil + c - 1$ capacity of the reducer. The remaining capacity, $2w\lceil \frac{n}{k} \rceil + 1$, of the reducer is used to assign all $2w\lceil \frac{n}{k} \rceil$ overlapping intervals of the interval i and an interval, $i' \in Y$ that have an identical starting-point as the interval i . (Note that i' is an overlapping interval for some of the pre-intervals and the post-intervals of i .) Thus, the interval i is assigned to a reducer with all its $2w\lceil \frac{n}{k} \rceil$ overlapping intervals of the relation Y . Further, the interval i will also be paired with all its remaining $2\lceil \frac{n}{k} \rceil - 2w\lceil \frac{n}{k} \rceil$ overlapping intervals at some reducers.

Now, we consider $w \geq 1$. In this case, for a partition p , there must be an interval $i \in X$ that can be assigned to a reducer with all its $2\lceil \frac{n}{k} \rceil$ overlapping intervals of the relation Y . Also, there are at most $\lceil \frac{n}{k} \rceil$ post-intervals and $c = \lceil \frac{n}{k} \rceil$ pre-intervals (of the interval i) that lie in the partition p . Thus, we can assign interval i , post-intervals of i , and pre-intervals of i at a single reducer. In addition, an interval, $i' \in Y$ such that i and i' have an identical starting-point, is also assigned to the reducer. Therefore, the interval i is paired with all $2\lceil \frac{n}{k} \rceil$ overlapping intervals (of the relation Y) at the reducer. \square

4. VARIABLE-LENGTH AND EQUALLY-SPACED INTERVALS

Two relations X and Y , each of n intervals, are given, where all intervals can have non-identical length but equally-spaced. We assume that the first interval starts at time 0, and the space between every two successive intervals is $s < 1$; see Figure 3, where a relation X has 6 intervals, and a relation Y has also 6 intervals. A *solution* to the problem of interval join of overlapping variable-length and equally-spaced intervals is a mapping schema such that each pair of overlapping intervals, one from each of the relations, is sent to at least one reducer in common without exceeding q .

We consider two types of intervals, as follows: (i) *big and small intervals*: one of the relation, say X , is holding most of the intervals of length l and the other relation, say Y , is holding most of the intervals of length $l' \gg l$; we call intervals of the relations X and Y as *small intervals* and *big intervals*, respectively; and (ii) *different-length intervals*: all the intervals of both the relations are of different-length (we will consider the second case in Appendix). In this section, we will provide lower bounds on the replication rate for both types of intervals. We then provide algorithms for interval join of overlapping intervals and show an upper bound on the replication rate. Throughout this section, we will use the following notations: l_{max} : the maximum length of an interval, l_{min} : the minimum length of an interval, and w : length of a partition.

4.1 Big and small intervals

In this section, we consider a special case of variable-length and equally-spaced intervals, where all of the intervals of two relations X and Y have length l_{min} and l_{max} , respectively, such that $l_{min} \ll l_{max}$; see Figure 3. We call the intervals of the relations X and Y as *small intervals* and *big intervals*, respectively.

Since every two successive intervals have an equal space, s , an interval $x_i \in X$ of length l_{min} can overlap with at least $2\lceil \frac{l_{min}}{s} \rceil + 1$ intervals of the relation Y , where at least $\lceil \frac{l_{min}}{s} \rceil$ intervals of the

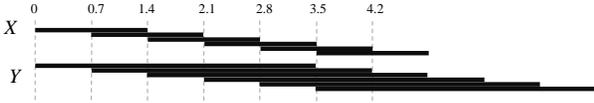


Figure 3: An example of big and small length but equally-spaced intervals, where $n = 6$ and $s = 0.7$.

relation Y have their ending-points between the starting and the ending-points of x_i , at least $\lfloor \frac{l_{min}}{s} \rfloor$ intervals of the relation Y have their starting-points between the starting and the ending-points of x_i , and an interval $y_i \in Y$ has an identical starting-point as x_i . In addition, an interval $x_i \in X$ of length l_{max} can overlap with at most $2\lfloor \frac{l_{max}}{s} \rfloor + 1$ intervals of the relation Y , where at most $\lfloor \frac{l_{max}}{s} \rfloor$ intervals of the relation Y have the ending-points between the starting and the ending-points of x_i and at most $\lfloor \frac{l_{max}}{s} \rfloor$ intervals of the relation Y have the starting-points between the starting and the ending-points of x_i , and an interval $y_i \in Y$ has an identical starting-point as x_i .

Theorem 3 (Minimum replication rate) For a relation X of n small and equally-spaced intervals and a relation Y of n big and equally-spaced intervals, the minimum replication of an interval, for joining each interval of the relation X with all its overlapping intervals of the relation Y , is (i) at least 2 when $2n > q \geq 2\lfloor \frac{l_{min}}{s} \rfloor$, and (ii) at least $\frac{2}{q}\lfloor \frac{l_{min}}{s} \rfloor$ when $2 < q < 2\lfloor \frac{l_{min}}{s} \rfloor$, where q is the reducer capacity, s is the spacing between every two successive intervals, and l_{min} is the length of the smallest interval.

PROOF. First we consider the case of $2n > q \geq 2\lfloor \frac{l_{min}}{s} \rfloor + 2$. When $q \geq 2n$, all the $2n$ intervals of the relations X and Y can be assigned to a single reducer, which is able to provide all output pairs. When $2\lfloor \frac{l_{min}}{s} \rfloor + 2 < q < 2n$, a single reducer cannot hold all the $2n$ intervals of the relations X and Y . Hence, at least a single interval, say j , that is not assigned with all its $2\lfloor \frac{l_{min}}{s} \rfloor + 1$ overlapping intervals must be assigned to another reducer. Therefore, the minimum replication of an interval is at least 2.

Now, we consider the case of $2 < q < 2\lfloor \frac{l_{min}}{s} \rfloor + 2$. Consider an interval i of length l_{min} . Since the interval i has at least $2\lfloor \frac{l_{min}}{s} \rfloor + 1$ overlapping intervals, all these $(2\lfloor \frac{l_{min}}{s} \rfloor + 2)$ intervals cannot be assigned to a single reducer. The interval i can share a reducer with at most $q - 1$ intervals of the relation Y . Hence, in order to assign the interval i with all the remaining overlapping intervals, it is required to assign subsets of overlapping intervals of the relation Y such that each subset holds at most $q - 1$ intervals. Thus, the interval i must be sent to at least $2\lfloor \frac{l_{min}}{s} \rfloor + 1/q - 1 > \frac{2}{q}\lfloor \frac{l_{min}}{s} \rfloor$ reducers. \square

Algorithm 2. Algorithm 2 for interval join of overlapping intervals of a relation X of small and equally-spaced intervals and a relation Y of big and equally-spaced intervals works in a similar fashion as Algorithm 1 performs the join operation. However, Algorithm 2 creates P partitions of the time-range (from 0 to ns), each of length of length $w = \frac{q-c}{3\lfloor \frac{l_{max}}{s} \rfloor}$, where $c = \lfloor \frac{l_{min}}{s} \rfloor + 2$. Note that in Algorithm 2, small intervals are assigned to several reducers corresponding to their partitions that they cross, and large intervals are assigned to only two reducers corresponding to their starting and ending points' partitions. The correctness of Algorithm 2 proves that each pair of overlapping intervals is assigned to at least one reducer in common, where $q = 3w\lfloor \frac{l_{min}}{s} \rfloor + c$, where $c = \lfloor \frac{l_{min}}{s} \rfloor + 2$.

4.2 An upper bound for the general case

In this section, we show an algorithm and an upper bound on the replication rate for the problem of interval join of variable-length

but equally-spaced intervals. We use the following notations: T : the length of time in which all intervals exist, *i.e.*, all intervals begin at some time greater than or equal to 0 and end by time T ; n : the number of intervals in each of the two relations, X and Y ; S : the total length of all the intervals in one relation; and w : the length of time corresponding to one reducer, *i.e.*, we divide T into $\frac{T}{w}$ equal-length segments, each of length w .

Algorithm 3. Algorithm 3 works in a manner similar to Algorithms 1 and 2 do. But this algorithm does more than Algorithms 1 and 2. It finds all intervals that intersect, regardless of whether they overlap, are superimposed, or any other relation. We divide the time-range into $\frac{T}{w}$ equal-sized partitions and arrange $\frac{T}{w}$ reducers, one for each partition. After that, we follow the same procedure as followed in Algorithms 1 and 2.

Theorem 4 (Algorithm correctness) Algorithm 3 assigns each pair of overlapping intervals to at least one reducer in common, where $q = \frac{3nw+S}{T}$, each of the two relations, X and Y , holds n intervals, q is the reducer capacity, S is the total length of all the intervals in one relation, w is the length of a partition, and T is the length of time in which all intervals exist.

PROOF. Following the algorithm, each of the n intervals of the relation Y is sent to at most two reducers. Since there are $\frac{T}{w}$ reducers, a reducer receives $\frac{2nw}{T}$ inputs from Y in average. Since the length of all the intervals of the relation X is S , the average length of intervals is $\frac{S}{n}$. Following the algorithm, an interval of X is sent to $1 + \frac{S}{nw}$ reducers. Since there are $\frac{T}{w}$ reducers, the reducer receives $(1 + \frac{S}{nw})\frac{nw}{T}$ inputs from X in average. Thus, a reducer receives at most $\frac{2nw}{T} + \frac{nw}{T}(1 + \frac{S}{nw}) = \frac{3nw+S}{T}$ inputs, which is equal to the given reducer capacity. \square

Theorem 5 (Replication rate) For $q = \frac{3nw+S}{T}$ and two relations, X and Y , of variable-length but equally-spaced, the replication rate of an interval, for joining each interval of the relation X with all its overlapping intervals of the relation Y is $\frac{3}{qT-S} \frac{S}{2}$.

5. REFERENCES

- [1] F. N. Afrati and et al. Upper and lower bounds on the cost of a map-reduce computation. *PVLDB*, 6(4):277–288, 2013.
- [2] B. Chawda and et al. Processing interval joins on map-reduce. In *EDBT*, pages 463–474, 2014.
- [3] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In *OSDI*, pages 137–150, 2004.
- [4] D. Gao, C. S. Jensen, R. T. Snodgrass, and M. D. Soo. Join operations in temporal databases. *VLDB J.*, 14(1):2–29, 2005.

APPENDIX

We consider a case of different-length intervals, *i.e.*, all the n intervals of each relation, X and Y , can have different-length. For a relation X and a relation Y , each is of n different-length but equally-spaced intervals, the minimum replication of an interval, for joining each interval of the relation X with all its overlapping intervals of the relation Y , is same as given in Theorem 3.

Algorithm 4. We propose an algorithm for interval join of overlapping different-length and equally-spaced intervals, which belong to two relations X and Y , each is of n intervals. Algorithm 4 works identically to Algorithms 1, 2, and 3. However, Algorithm 4 is different from Algorithms 1, 2 and 3, when it divides the time-range from 0 to ns into P partitions, each of length $w = \frac{q-c}{3\lfloor \frac{l_{max}}{s} \rfloor}$, where $c = \lfloor \frac{l_{min}}{s} \rfloor + 2$. The algorithm correctness shows that Algorithm 4 assigns each pair of overlapping intervals to at least one reducer in common, where $q = 3w\lfloor \frac{l_{min}}{s} \rfloor + c$, where $c = \lfloor \frac{l_{min}}{s} \rfloor + 2$.