Previous Formulas

1 Previous version

$$Card(E1 \bowtie E2) = MIN(Card(E1), Card(E2)) \tag{1}$$

$$Cost(E1 \bowtie_h E2) = Card(E1) * CRT + Card(E2) * CRT + 2 * CSQ \tag{2}$$

$$Cost(E1 \bowtie_b E2) = Card(E1) * CRT + \frac{Card(E1)}{BSZ} * CSQ + Card(E1 \bowtie E2) * CRT \tag{3}$$

where the cost for sending a SPARQL query is CSQ, the cost for receiving a single result tuple is CRT and BSZ is the size of a bound join block.

In our experiments CSQ=50, CRT=0.02, BSZ=20

2 Modified version

$$Card(E1 \bowtie E2) = MIN(Card(E1), Card(E2)) * MVK(E1) * MVK(E2) \tag{4}$$

where MVK(E1) and MVK(E2) are multivalue multiplyers of E1 and E2 respectively.

$$MVK(E) = \begin{cases} \frac{1}{\sqrt{2}} & : \text{E is a triple pattern like ?s <p> <o> .} \\ \frac{\text{triple count for given predicate}}{\text{distinct subject count}} & : \text{E is a triple pattern like ?s <p> ?o. The join variable is ?s.} \\ \frac{\text{triple count for given predicate}}{\text{distinct object count}} & : \text{E is a triple pattern like ?s <p> ?o. The join variable is ?o.} \\ 1 & : \text{other cases} \end{cases}$$

$$Cost(E1 \bowtie_h E2) = \frac{1 + TC}{TC} * CSQ + Card(E2) * CRT + (Card(E1) + Card(E2)) * CHT \tag{5}$$

where Card(E1) < Card(E2), TC is the number of threads used to query sparql endpoints, CSQ is the cost of sending a SPARQL query, CRT is the cost of receiving a single result tuple, CHT is the cost of handling received tuple.

Description: the hash join algorithm sends 2 requests for E1 and E2 using TC threads (cost = first summand in the (5)), then recieves results for E1 and E2 in parallel, so cost = Max(Card(E1), Card(E2)) * CRT = Card(E2) * CRT, finally all the tuples received

are handled: the internal implementaion uses hashmap with synchronized access to store data, so cost can be estimated as (Card(E1) + Card(E2)) * CHT

$$Cost(E1 \bowtie_b E2) = CSQ + Card(E1) * CRT + \frac{\frac{Card(E1)+BSZ-1}{BSZ} + CTC - 1}{CTC} * CSQ \quad (6)$$

The bind join algo at first sends the request for E1 (cost = CSQ), receives results for E1 (cost = Card(E1) * CRT), then using TC threads sends resuests for E2 using bunch of BSZ size (cost = third summand in the formula (6))

In our experiments CSQ=100, CRT=0.01, 0.0025, BSZ=20, TC=20