Tserpe Paraskevi 2068
Kyriakou Aikaterini 2034

Mobile & Pervasive Computing

Project

- **Subject**
  Set-based broadcast scheduling for minimizing the worst access time of multiple data items in wireless environments.

- **Algorithm Details**
  The Bounded Access Time (BAT) algorithm is proposed as a scheduling algorithm for broadcast in multiple channels that minimizes the worst access time ε(π). Given an access pattern *AP(m)={Q1, Q2,..,Qm}* with m query sets to be broadcast by a server with C wireless channels, it denotes a partition π = {P1, P2,…, Pc}. To define this solution, the access pattern is assumed to be given in advance. The worst access time is indicated by the lengths of the partitions ε(π) = Min{Max{|P1|, |P2|, …, |Pc|}}.
  To generate a near-optimal solution to the NP-hard BAT problem, five heuristics are applied in the proposed algorithm.
  The five heuristics are described as follows:

- **Initial phase of the algorithm**
  - **Heuristic 1**
    Overlapped query sets(query sets with common data items) had better be allocated to the same part Pi.
  - **Heuristic 2**
    Large query sets need to be allocated apart even though they are in a connected component ().
  - **Heuristic 3**
    If there exists small disjoint query sets (sets with no common data items), allocate them to smaller parts (such that |Pi|<boundary, where boundary is the minimum channel capacity, boundary = Number_Data_Items / C), one by one and in descending order of |Qj|.

- **Tuning phase of the algorithm**
  - **Heuristic 4**
    In finding the optimal solution, for each temporary partition, try to move an excess query set(Qk, a set with data items that exceed the boundary) from the first channel (P1) to a proper one of tha last several parts (candidate channels).
  - **Heuristic 5**
    Qk had better be an excess query set.

- **Implementation details**
  - **Required Inputs :**
    1. An access pattern (AP(m))
    2. The number of channels (C)
    3. The threshold that indicates the large parts (α)
    4. The number of candidate channels (β)
  - **Output**
    A near optimal partition π and a near optimal access time ε(π)

Algorithm ***MinimalAccessTime***$(C, \boldsymbol{AP}(m), \alpha, \beta)$
INPUT: C broadcast channels;
an access pattern $\boldsymbol{AP}(m)$ containing m query sets;
threshold α (if $|P_i| \geq \alpha$, P_i is regarded as a large part in the partition);
the last β parts, candidates to accommodate the excess query set of P_1.
OUTPUT: a near-optimal partition π and a near-optimal access time ε.
METHOD:
//-------------------------------------Initial Phase --
Step 1 Roughly separate the large query sets ($|Q_j| \geq \alpha$) into C equal parts, i.e., construct the initial π. //Heuristic 2
Step 2 Find the connect components for the remaining query sets. //Heuristic 1
Step 3 Adjust π such that $|P_1| \geq |P_2| \geq \ldots \geq |P_C|$ immediately after appending each connected component to P_C. //Heuristic 1
Step 4 Adjust π such that $|P_1| \geq |P_2| \geq \ldots \geq |P_C|$ immediately after appending each remaining disjoint query set to P_C. //Heuristic 3
//-----------------------------------Tuning Phase ---
Step 5 Set *DONE*=*False*.
Repeat Steps 6-11 **until** *DONE*.
Step 6 Let the last β parts, $P_{C-\beta+1}, P_{C-\beta+2}, \ldots, P_C$, be the candidates. //Heuristic 4
Step 7 Decide which query set $Q^{(k)}$ and candidate P_x are able to decrease $|P_1|$ most as well as to increase $|P_x|$ least. //Heuristic 5
If such $Q^{(k)}$ and P_x exist **then**
Step 8 Move $Q^{(k)}$ from P_1 to P_x according to the decision made in Step 7;
Step 9 Adjust π such that $|P_1| \geq |P_2| \geq \ldots \geq |P_C|$
else
Step 10 Output $P_1, P_2, \ldots, P_C$, and the worst access time $\varepsilon = |P_1|$;
Step 11 Set *DONE*=*True*.
Step 12 **Stop**.

Fig. 4. The heuristic algorithm for solving BAT.