A Static Data Structure for Discrete Advance Bandwidth Reservations on the Internet

Andrej Brodnik
Andreas Nilsson

CSEE Luleå University of Technology
Introduction

- Differentiated services
- Quality of service (QoS)
- Bandwidth brokers
Bandwidth brokers

• Manages the bandwidth reservations on one link
Interaction between bandwidth brokers

- Usually the route between two computers consists of several links
- Entire route has to be reserved
Definitions

- A reservation \( R \) is a time interval during which constant amount of bandwidth \( B \) is allocated throughout the entire interval \( I \).
- In the data structure \( D \) we use slotted time, that is fixed granularity \( g \) to gain aggregation.
- Bounded universe, maximum interval size \( |M| \) has the interval of the root \( M \).
Operations

- $\text{Insert}(D, R)$ increases the reserved bandwidth during the interval $I$ for $B$
- $\text{Delete}(D, R)$ decreases the reserved bandwidth during the interval $I$ for $B$
- $\text{MaxReserved}(D, I)$ returns the maximum reserved bandwidth during the interval $I$
Advanced Segment Tree (AST)

- Modified segment tree
- The tree is static, i.e. it is
  - only built once, and in advance, no nodes are added or deleted and therefore the tree is always perfectly balanced
- All nodes on a level $l$
  - have the same number of children
  - represents a time interval
  - cover intervals of the same size
  - have intervals that mutually don't intersect
  - have intervals which’s union is $M$
  - have an interval that is contained within one interval on level $l'$, where $l' < l$
Nodes in the AST

- Each node $n$
  - has a pointer to each child
  - has a value $nv$ – the amount of bandwidth reserved exactly the interval covered by $n$
  - has a value $mv$ – the maximum amount of bandwidth reserved on the interval covered by $n$
Nodes and their values

node value = 10
max value = $\max(50 + 200, 0 + 220) = 250$

node value = 50
max value = $\max(60 + 90, 120 + 80) = 200$

node value = 0
max value = $\max(0 + 30, 20 + 200) = 220$

node value = 60
max value = 90

node value = 120
max value = 80

node value = 0
max value = 30

node value = 20
max value = 200

90 0 0 80 0 0 30 0 0 0 200 0
Insert \((D, R)\)

\[
B_R = 50
\]

\[
nv = 10 \\
mv = \max(100+200, 0+220) = 300
\]

\[
nv = 50+50 \\
mv = \max(60+90, 120+80) = 200
\]

\[
nv = 0 \\
mv = \max(50+30, 20+200) = 220
\]

\[
nv = 60 \\
mv = 90
\]

\[
nv = 0+50 \\
mv = 30
\]

\[
nv = 20 \\
mv = 200
\]

\[
nv = 90 \\
mv = 0
\]

\[
nv = 0+50 \\
mv = 200
\]

**time** \(O(\log n)\)

**space** \(O(n)\)
MaxReserved\((D, I)\)

\[ \text{MaxReserved} = \max(10+\text{MaxReserved}(t_4, t_5)), 10+\max(300, 80) = 10+300 = 310 \]

\[ \text{MaxReserved} = 0 + \text{MaxReserved}(t_4, t_5) = 0 + 80 = 80 \]

\[ \text{MaxReserved} = 50 + (30 + 0) = 80 \]

\[ \text{time O}(\log n) \]

\[ \text{space O}(n) \]
Implicit data structure

\[ \delta_l = \begin{cases} \sum_{j=2}^{l} \left( \prod_{i=1}^{j-1} X_i \right) + 1 \quad & l = 1 \\ \sum_{j=2}^{l} \left( \prod_{i=1}^{j-1} X_i \right) + 1 \quad & 1 < l \leq L \end{cases} \]

\begin{align*}
\text{time} & \quad O(\log n) \\
\text{space} & \quad O(n)
\end{align*}
Implicit data structure
Conclusions

• Easily implemented as an implicit data structure
• AST is generic solution
• Worst case time complexity for all operations is $O(\log n)$ (where $n$ is the number of leaves)
• The solution is used in the real world
Thank you

Time for questions

Andreas.Nilsson@sm.luth.se
Andrej.Brodnik@sm.luth.se

http://www.sm.luth.se/~andreas/publications/