Nested Intervals Tree Encoding with System of Residual Classes

A. Malikov and A. Turyev
North Caucasus State Technical University, 2 Kulakova st., Stavropol, 355029
Russia

ABSTRACT
This paper describes one of ways to represent tree-like structures in Data Bases. Authors suggest to expand V. Tropashko nested intervals model. Numbers in intervals are described in system of residual classes. It allows to avoid storage of big numbers and easily implemented by parallel-programming algorithms.

1. INTRODUCTION
The majority of information systems data bases are relational nowadays. Temporal property and hierarchies are implemented not effectively. The simple model of adjency lists has been developed for relational systems. Some language structures to support it were included into language standard SQL (SQL:1999). Now we have few popular models: 1) adjency list model and modifications, 2) materialized path model, 3) nested sets model [3], 4) materialized path model extensions (coding by using recurring decimals, etc.).

The materialized path model is used in data type HierarchyID of MS SQL Server 2008 DBMS. The given model has serious limitations: fragment of a key length growing and fragments of a key number growing; efficiency decreasing because of key size growing; it's not possible to model hierarchical data of a distinct from tree form, etc.

2. NESTED INTERVALS CODING
Interesting approach is offered by V. Tropashko in [1] and [2]. The nested sets model is extended by using rational numbers (matrixes or continued fractions). So we simply code materialized path. Author offer some variants to calculate this operations: to use continued fractions or to present conversion in the form of matrix operations.

Let's present a tree.

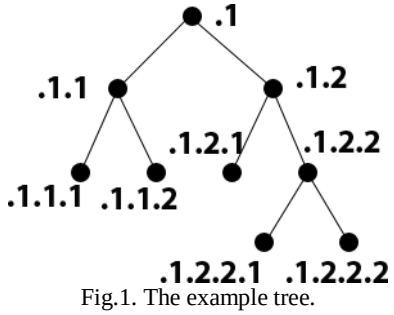

Fig.1. The example tree.

Let's code node .1.2.2.2 by using continued fractionsc (1) and matrix 2×2 (2)

$$1.2.2.2 = 1+\cfrac{1}{2+\cfrac{1}{2+\cfrac{1}{2+\cfrac{1}{x}}}} = \frac{17x+7}{12x+5}; \quad (1)$$

$$\begin{bmatrix}1 & 1\\ 1 & 0\end{bmatrix}\cdot\begin{bmatrix}2 & 1\\ 1 & 0\end{bmatrix}\cdot\begin{bmatrix}2 & 1\\ 1 & 0\end{bmatrix}\cdot\begin{bmatrix}2 & 1\\ 1 & 0\end{bmatrix}=\begin{bmatrix}17 & 7\\ 12 & 5\end{bmatrix}; \quad (2)$$

Results of (1) and (2) are nested intervals like

$$\left(\frac{17}{12},\frac{7}{5}\right); \quad (3)$$

Let's present a tree in Fig.1 with a table. Table 1 consists of 3 columns: N, Path, Interval. The last column describes a way to node using this coding.

Some of the main operations with a tree, implemented by means of the given model are: add new node [1], find parent node [1], reallocate node [4].

Table 1.

N	Path	Interval
1	.1	(1/1)
2	.1.1	(2/1, 1/1)
3	1.1.1	(3/2, 2/1)
4	1.1.2	(5/3, 2/1)
5	1.2	(3/2, 1/1)
6	1.2.1	(4/3, 3/2)
7	1.2.2	(7/5, 3/2)
8	1.2.2.1	(10/7, 7/5)
9	1.2.2.2	(17/12, 7/5)

Add new node.

$$A_{parent}\cdot\begin{bmatrix}a_i & 1\\ 0 & 0\end{bmatrix}=C,$$

$$A_{parent}-parent\ node,$$

$$a_i-new\ node\ index, \quad (4)$$

$$C-new\ node\ \text{interval}\ in\ matrix\ form.$$

Find parent node.

$$A=\begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}\rightarrow B=\begin{bmatrix}A(a_{12}) & x_{12}\\ A(a_{22}) & x_{22}\end{bmatrix},$$

$$A-node, \quad (5)$$

$$B-parent\ node.$$

Reallocate node.

1) Get matrix of node to reallocate.

$$.x1.x2.x3.y1.y.2 - fullpath$$

$$A_{(.y1.y.2)} = \begin{bmatrix} y1 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} y2 & 1 \\ 1 & 0 \end{bmatrix}, \quad (6)$$

$$A_{(.y1.y.2)} - node\ matrix.$$

2) Add node to the new parent.

$$A_{parent} \times A_{(.y1.y.2)} = C, \quad (7)$$

$$A_{parent} - parent\ node,$$

$$A_{(.y1.y.2)} - node,$$

$$C - new\ \text{interval}\ of\ node.$$

3. SYSTEM OF RESIDUAL CLASSES

It is offered to use systems of residual classes [5]. It allows to avoid problems with data types restrictions. It's also possible to use algorithms of parallel processing.

A number is presented like

$$A(\alpha_1, \alpha_2, ..., \alpha_n); \quad (8)$$

$$\alpha_i = A - \left[\frac{A}{p_i} \right] \cdot p_i, \ (\forall i \in [1, n]), \quad (9)$$

$$p_1, p_2, ..., p_n - system\ \text{modules},$$

$$P = \prod_{i=1}^{n} p_i \quad (10)$$

P – volume of system range.

Assume $p_1 = 2, p_2 = 3, p_3 = 5, P = 30$.

Node .1.2.2.2of Table 1 is presented by using (8) and (9).

$$.1.2.2.2 = (17/12, 7/5),$$

$$a_{11} = 17 = (\alpha_1, \alpha_2, \alpha_3),$$

$$\alpha_1 = -\left[\frac{17}{2} \right] \cdot 2 = 17 - 8 \cdot 2 = 1,$$

$$\alpha_2 = 17 - \left[\frac{17}{3} \right] \cdot 3 = 17 - 5 \cdot 3 = 2,$$

$$\alpha_3 = 17 - \left[\frac{17}{5} \right] \cdot 5 = 17 - 3 \cdot 5 = 2,$$

$$a_{12} = 7 = (1, 1, 2),$$

$$a_{21} = 12 = (0, 0, 2),$$

$$a_{22} = 5 = (1, 2, 0),$$

$$A_{.1.2.2.2} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} (1,2,2) & (1,1,2) \\ (0,0,2) & (1,2,0) \end{bmatrix}.$$

The given method allows us to use not big numbers in intervals for big trees (about one million records).

Addition, multiplication operations and exponentiations are identical to the operations executable over system of residuals.

$$A = (\alpha_1, \alpha_2, ..., \alpha_n), B = (\beta_1, \beta_2, ..., \beta_n),$$

$$A + B = (\gamma_1, \gamma_2, ..., \gamma_n), A \cdot B = (\delta_1, \delta_2, ..., \delta_n),$$

$$\gamma_i = \alpha_i + \beta_i (\text{mod}\ p_i) = \alpha_i + \beta_i - \left[\frac{\alpha_i + \beta_i}{p_i} \right] p_i,$$

$$\delta_i = \alpha_i \beta_i (\text{mod}\ p_i) = \alpha_i \beta_i - \left[\frac{\alpha_i \beta_i}{p_i} \right] p_i. \quad (11)$$

Example for numbers *A* and *B:*

$A = 7 = (1,1,2), B = 5 = (1,2,0),$

$$A + B = \left(1 + 1 - \left[\frac{1+1}{2} \right] \cdot 2,\ 1 + 2 - \left[\frac{1+2}{3} \right] \cdot 3,\ 2 + 0 - \left[\frac{2+0}{5} \right] \cdot 5 \right) =$$

$$= (2 - 1 \cdot 2, 3 - 1 \cdot 3, 2 - 0 \cdot 5) = (0, 0, 2)$$

Arithmetical operations in the modular code are produced independently on each of units. Thanks to it, multi sequencing of calculations over residuals is possible. As a result, actions under formulas (4), (5), (6) are fulfilled over lists of residuals.

Add new node.

$$A_{parent} \cdot \begin{bmatrix} \left(a - \left[\frac{a}{p_i} \right] \cdot p_i, \ (\forall i \in [1, n]) \right) & 1 \\ 0 & 0 \end{bmatrix} = C,$$

$$A_{parent} - parent\ node,$$

$$a_i - new\ node\ index,$$

$$C - \text{interval}\ of\ new\ node.$$

Find parent node.

$$A = \begin{bmatrix} a_{11} - \left[\frac{a_{11}}{p_i} \right] \cdot p_i & a_{12} - \left[\frac{a_{12}}{p_i} \right] \cdot p_i \\ a_{21} - \left[\frac{a_{21}}{p_i} \right] \cdot p_i & a_{22} - \left[\frac{a_{22}}{p_i} \right] \cdot p_i \end{bmatrix},$$

$$(\forall i \in [1, n]),$$

$$B = \begin{bmatrix} A\left(a_{12} - \left[\frac{a_{12}}{p_i} \right] \cdot p_i \right) & x_{12} \\ A\left(a_{22} - \left[\frac{a_{22}}{p_i} \right] \cdot p_i \right) & x_{22} \end{bmatrix},$$

$$(\forall i \in [1, n])$$

$$A - node,$$

$$B - parent\ node.$$

Reallocate node.

3) Get matrix of node to reallocate.

$.x1.x2.x3.y1.y.2 - full\ \ path$

$$A_{(.y1.y.2)} = \begin{bmatrix} y1 - \left[\frac{y1}{p_i}\right] \cdot p_i & 1 \\ 1 & 0 \end{bmatrix} . \begin{bmatrix} y2 - \left[\frac{y2}{p_i}\right] \cdot p_i & 1 \\ 1 & 0 \end{bmatrix},$$

$A_{(.y1.y.2)} - node\ matrix.$

1) Add node to the new parent.

$$A_{parent} \times A_{(.y1.y.2)} = C,$$

$$A_{parent} - parent\ \ node,$$

$$A_{(.y1.y.2)} - node,$$

$$C - new\ \text{interval}\ of\ \ node.$$

The given method has been implemented on GPU NVidia. Technology CUDA was used. Linear algebra operations are effectively implemented by using this technology. Besides, residual classes demand on parallel algorithms.

For calculations some kernels have been written. One kernel was used to calculate operations over the numbers presented by residual classes. Other kernel carried out matrix operations. GPU was 5-7 times more productivity than CPU.

This paper is worked out within the limits of federal program realization «Scientific personnel of innovative Russia» for 2009-2013 on a problem «Working out of theoretical bases of functioning of semistructured data control systems».

4. REFERENCES

[1] V. Tropashko, "Nested Intervals with Farey Fractions", 2004.

[2] V. Tropashko, "Trees in SQL: Nested Sets and Materialized Path", 2003a.

[3] J. Celko, "Joe Celko's Trees and Hierarchies in SQL for Smarties", Morgan Kaufmann, 2004.

[4] V. Tropashko, "Relocating Subtrees in Nested Intervals Model", 2003b.

[5] N. Chervyakov, "Modular Parallel Computing Structures of Neuroprocces System", 2003.