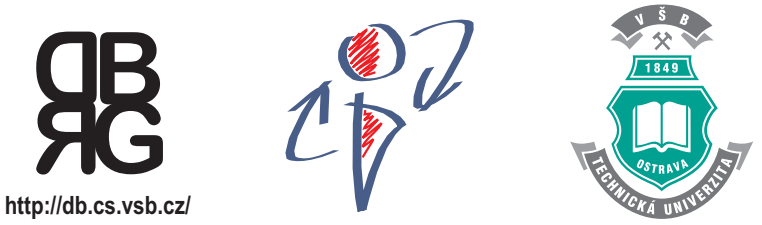
RadegastXDB (ver. 0.3) – User Documentation

Radim Bača, Petr Lukáš, Michal Krátký
radim.baca@vsb.cz, petr.lukas@vsb.cz, michal.kratky@vsb.cz



Database Research Group http://db.cs.vsb.cz
Department of Computer Science http://www.cs.vsb.cz
VŠB – Technical University of Ostrava http://www.vsb.cz

Abstract

This documentation describes our prototype of a native XML DBMS (Database Management System) supporting XQuery and XPath queries. The document describes how to configure and run the prototype and how to capture various statistics. The prototype is able to perform experiments described in [4].

Contents

1 Introduction

RadegastXDB is a prototype of a native XML database management system (DBMS) supporting XQuery and XPath queries. The implementation is at the stage of a prototype, so there are some important characteristics of real DBMSs missing. For example, a transaction processing, a data modification interface and a security. XQuery is not fully supported, no network connection is available, all attributes are treated as common XML nodes and there might be some stability issues. The aim of this prototype is to test physical operators evaluating *twig pattern queries* (see Section 1.2).

Since the whole system is implemented in C++ programming language, there is no overhead of automatic memory management and the core algorithms can run faster than on some similar commercial systems.

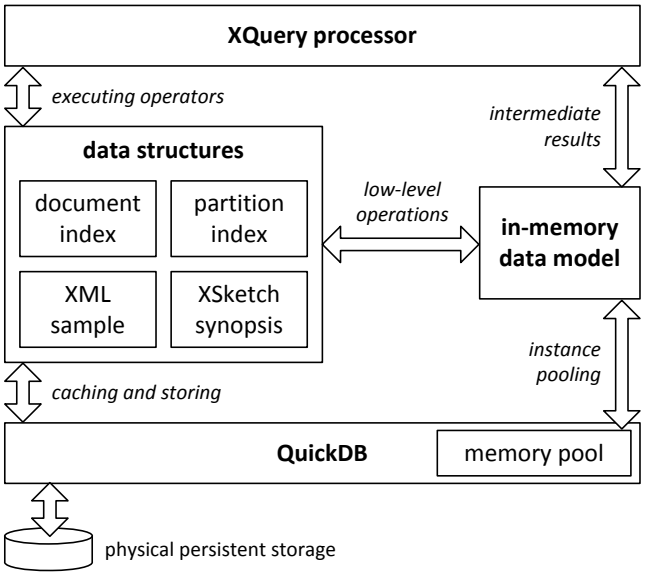


Figure 1: Block structure of the RadegastXDB DBMS

A basic block structure of the RadegastXDB DBMS is given in Figure 1. There are four blocks that the system constists of:

- **QuickDB** [2] – is our internal framework that manages various persistent data structures such as B^+-tree or a paged heap array. There is also a subpart that manages allocation of memory from a reusable memory pool.
- **data structures** – is a block containing persistent data structures. The most important structures are *document index*, *partition index*, *XSketch synopsis* and *XML sample*. The first two structures are used to index an XML collection. For more detail, see [7]. The other two structures are statistics making result size estimations possible [10, 8].

- **in-memory data model** – represents both intermediate and final results of a query execution.
- **XQuery processor** – forms the front-end of the DBMS. It is an algebraic processor of XML queries (XQuery and XPath) that was originally written as a separated application operating over an in-memory DOM representation of an XML document.

1.1 Query Execution

The execution of each query passes through the following steps:

1. **parsing** – the XQuery or XPath query is parsed into a *syntax tree* that consists of non-terminal and terminal symbols according to the XQuery grammar [1].
2. **compiling** – the syntax tree is tranlated into a tree of physical operators that forms an *algebraic query plan*.
3. **optimization** – several rewriting rules are applied to the algebraic query plan. Some of them only rearrange the tree, so more selective operators can be evaluated first, while other rules introduce special operators like holistic join algorithms. This step also includes a phase of cost-based optimization of the holistic joins.
4. **execution** – the optimized algebraic query plan is evaluated.

1.2 Twig Pattern Queries

Our research is currently oriented on holistic twig join algorithms [3, 4] evaluating twig pattern queries (TPQs). TPQ is a simplified model of an XML query used by many approaches [5, 3, 2, 4, 9]. There is one physical operator called *TupleTreePattern* (TTP) [9] that is supported by GTPStack [3] and CostTwigJoin [4] algorithms. This operator is introduced in the optimization step of a query execution by replacing one or more naive tree structure joins and selections.

All XPath queries, containing only `and` logical operations and (child or descendand) axes, and a large class of XQuery queries can be expressed by a simple algebraic operator plan with a single TTP operator argumented by an appropriate TPQ.

2 Usage

The RadegastXDB DBMS is represented by an executable aplication `xdb.exe` on the Windows platform and `xdb` on Linux. The syntax of usage is as follows:

```
xdb.exe <settings.xml> create
xdb.exe <settings.xml> test
xdb.exe <settings.xml> immediate
```

(The `.exe` extension is skipped on Linux)

Two arguments must be always specified. The first argument `<settings.xml>` is a path to a configuration XML file where all settings of RadegastXDB run are specified. All settings of the configuration XML are described in Section 3. The second argument defines a mode of the run:

- `create` reads an XML document and builds all necessary indexes and statistics.
- `test` runs an automated test of a set of queries while some important statistics (query selectivity, execution time, etc.) are captured. For more detail, see Sections 3.2 and 4.1.
- `immediate` runs a single XML query. All settings of immediate querying are described in Section 3.6.

3 Configuration XML

Examples of the configuration XML files can be downloaded from `http://db.cs.vsb.cz/SubPages/Projects/Projects.aspx`. A short description of each tag is also included in the file. A detailed description of all settings is given in this section.

`title`
Represents a user-friendly title that is printed to the standard output when RadegastXDB is started.

`instance_name`
More instances of RadegastXDB can be run sumultaneously on a single machine. However, the prototype does not support any socket connections, so the instance name currently do not play any role. Leave this setting as its default: 'Instance'.

`collection_name`
Similarly as the previous setting, only one collection with a single XML document is supported. Leave this setting as its default: 'Collection'.

`cache_factor`
The database uses a node cache of size 1024 multiplied by this integer constant.

3.1 Path Settings

`paths/xml`
A system path to an XML file that you want to index. All persistent data structures including indexes and statistics are build from this file when RadegastXDB is started in the `create` mode.

`paths/index`
A directory, where physical data files of persistent structures are created. The path must be ended by a backslash '(\)'. Make sure you have a permission to create and modify files in this directory.

`paths/queries`
A text file containing a set of XML queries – each row poses one query. This file is used when RadegastXDB is started with in the `test` mode.

3.2 Test Settings

All settings in this subsection take effect only when RadegastXDB is started in the `test` mode.

`testing/loops`
A positive integer number defining how many times a query is evaluated repeatedly. Afterwards, each time statistic is computed as an arithmetic mean of measured values without the best and the worst case. Therefore, the time statistics can be measured only when this setting is at least 3.

`testing/output_results`
When testing a set of queries, statistics can be captured into a CSV file. The system path of the file is specified in this setting. No CSV is generated when this setting is empty. For more detail about the output, see Section 4.1.

3.3 Cost-based Optimization Settings

The settings described in this subsection are related to the cost-based optimization phase of holistic joins. The cost-based optimization is performed such that a set of plans is explored and for each plan a cost is computed using a *result size estimation method*. The set of plans is determined by a *plan selection method*.

`cb_optimization/optimization_method`
A switch selecting one of three plan selection methods to optimize pattern queries.

- `none` – no cost-based optimization is performed, pattern queries are evaluated using the GTPStack holistic join algorithm [3].
- `greedy` – the Greedy Forward plan selection method is used and pattern queries are evaluated using the CostTwigJoin algorithm.
- `top_k` – the Top-k plan selection method is used ($k = 4$) and pattern queries are evaluated using the CostTwigJoin algorithm [4].

`cb_optimization/estimation_method`
A switch selecting the result size estimation method. This setting does not take any effect when `cb_optimization/ optimization_method` is set to `none`.

- `xsketch` – the f-XSketch estimation method [6, 10] is used. Make sure that the `xsketch/build` setting is `true`.
- `sample` – the sampling-based estimation method [8] is used. Make sure that the `sample/build` setting is `true`.

`cb_optimization/cost_model/alpha`
Specification of the α constant in the cost model (see [4]).

`cb_optimization/cost_model/beta`
Specification of the β constant in the cost model (see [4]).

`cb_optimization/cost_model/omega`
Specification of the ω constant in the cost model (see [4]). This setting takes effect only if `cb_optimization/cost_model/estimate_cpu_cost` is set to `true`.

`cb_optimization/cost_model/estimate_cpu_cost`
When this setting is `true`, the cost of a query plan is evaluated using the 4-variable cost model that considers the CPU cost. Otherwise, when this setting is `false`, the 3-variable cost model is used. 4-variable cost model can be used only if the `cb_optimization/estimation_method` setting is `sample` and the sample statistics is built (`sample/build` setting is `true`).

3.4 XSketch Settings

All settings in this subsection are related to the f-XSketch staitstics that can be used by the cost-based optimization phase of RadegastXDB (see the `cb_optimization/estimation_method` setting) to estimate the query result size. For more detail about the XSketch estimation framework and its fractional variant see [10, 6].

`xsketch/build`
If this setting is set to `true`, the statistics is built an can be used by the cost-based optimization phase. Set this setting to `false` to completely avoid building of the statistics.

`xsketch/training_set`
An integer value specifying the number of randomly generated distinct queries to refine the statistics.

`xsketch/verification_set`
An integer value specifying the number of randomly generated distinct queries to verify an estimation error when the statistics is built.

`xsketch/refinements`
An integer value representing the number of refinement steps.

`xsketch/refs_per_step`
In each refinement step a number of refinements to optimize backward and forward fractional stabilities is peformed. Subsequently, only the best refinement is used in each refinement step. The number of refinements per one step is specified by this setting.

`xsketch/quick_break`
Only for testing purposes.

`xsketch/output_build`
A system path of a CSV file where a building (refining) process of the f-XSketch statistics can be captured. Set this setting to an empty value to not capture the building process. For more detail, see Section 4.3.

`xsketch/output_coarse_xsketch`
A system path of a CSV file where a coarse statistics (before refining) can be captured. Set this setting to an empty value to not capture the coarse statistics. The output is decribed in Section 4.2 in a more detail.

`xsketch/output_refined_xsketch`
A system path of a CSV file where a refined statistics (after refining) can be captured. Set this setting to an empty value to not capture the refined statistics. The output is of the same format as the previous setting.

`xsketch/output_training_set`
The set of queries that are randomly generated to refine the statistics can be captured into a text file specified in this setting. Leave this setting empty to not capture the training queries.

`xsketch/output_verification_set`
The set of queries that are randomly generated to verify the statistics estimation

error can be captured into a text file specified in this setting. Leave this setting empty to not capture the verification queries.

3.5 Sample Settings

All settings in this subsection are related to the sample statistics [8] that can be used by the cost-based optimization phase of RadegastXDB as an alternative to f-XSketch (see the `cb_optimization/estimation_method` setting).

`sample/build`
Set this setting to `true` to build an XML sample from the original XML document. Otherwise, leave this setting set to `false`. The XML sample is created in the working directory of RadegastXDB adding '.sample' suffix to the original XML document file name.

`sample/factor`
An integer nuber n specifying that each n-th XML element with the same name becomes part of the sampled document. $n = 1$ means that no sampling is performed.

`sample/min_cnt`
An XML element with tag t is sampled only if the number of all t elements in the document is at least min, which is specified by this setting.

3.6 Settings of Immediate Querying

All settings in this subsection are important when the application is started in the `immediate` mode.

`immediate/query`
A system path of a text file containing the XML query (XQuery or XPath).

`immediate/print_plan`
When this option is set to `true`, an algebraic query plan is printed to the standard output when a query is evaluated. If you do not want the plan to be printed, set this setting to `false`.

`immediate/print_result`
When this setting is `true`, the query result is materialized and printed to the standard output. If you are interested only in the result size, set this setting to `false` and use `immediate/print_result_size` setting, or surround the original query by `count()` function.

`immediate/print_result_size`
If this setting is `true`, the result size is printed to the standard output after a query is evaluated.

`immediate/repeat`
If this setting is `true`, the immediate querying is performed in a loop, so the user can evaluate more queries. After evaluation of each query, the user is asked to continue with another query or to exit RadegastXDB.

4 Outputs

All table-organized outputs can be captured into CSV files where values of one record are separated by a semicolon '`;`' character.

4.1 Automated Test Results

The location of this output file is specified by the `testing/output_results` setting described in Section 3.2. The file contains the following columns:

- *query* – original query from the `paths/queries` file.
- *status* – 'OK' if the query has been evaluated without any compile-time or run-time errors, otherwise 'ERR'.
- *selectivity* – selectivity of the query that is computed by a fraction n/n_{out} where n is the result size and n_{out} is a number of elements with the same tag as the output query node (sometimes called extraction point). For example, the selectivity of a query `//a//b` is its result size divided by the number of all elements with tag `b` in the XML document. The selectivity is valid only for XPath queries.
- *result size* – result size of the query.
- *time total [s]* – time of the whole evaluation process where optimization and rewritings of the algebraic query plan are included.
- *time plan [s]* – time of execution of the whole algebraic query plan.
- *time TPQ [s]* – time of running of the core algorithm evaluating twig pattern queries (GTPStack or CostTwigJoin).

An inequation *time total* > *time plan* > *time TPQ* is always held.

4.2 Coarse and Refined XSketch

This output is related to the `xsketch/output_coarse_xsketch` and `xsketch/output_refined_xsketch` settings (see Section 3.4). The output completely describes an actual f-XSketch statistics. Since the XSketch is represented by a directed graph, the output contains two parts: nodes and edges.

4.2.1 Nodes

- *node id* – identifier of an XSketch node.
- *term* – term (tag name) related to the XSketch node.
- *term id* – internal identifier of the term.
- *extent* – size of the extent (number of XML elements related to the XSketch node).

4.2.2 Edges

- *edge id* – identifier of an XSketch edge.
- *parent id* – identifier of the first (parent) XSketch node of the edge.
- *child id* – identifier of the second (child) XSketch node of the edge.
- *parent term* – term of the parent node.
- *child term* – term of the child node.
- *f-stable* – number of forward-stable XML nodes.
- *b-stable* – number of backward-stable XML nodes.

4.3 XSketch Refinement Process

This output captures the refinement process of the f-XSketch statistics. The output file is specified by the `xsketch/output_build` setting described in Section 3.4. The columns are as follows:

- *step* – order of the refinement step.
- *time [min]* – minutes from the start of the refinement process.
- *training error* – average relative error per one query of the training set.
- *verification error* – average relative error per one query of the verification set.
- *training time [s]* – average time of estimation of one query from the training set.
- *verification time [s]* – average time of estimation of one query from the verification set.
- *size [B]* – size of the statistics in bytes.

References

[1] Xquery 1.0: An xml query language (second edition). `http://www.w3.org/TR/xquery/`. Accessed: 2015-01-19.

[2] R. Baca, P. Chovanec, M. Krátký, and P. Lukáš. Quickdb–yet another database management system?

[3] R. Bača, M. Krátký, T. W. Ling, and J. Lu. Optimal and efficient generalized twig pattern processing: a combination of preorder and postorder filterings. *The VLDB Journal—The International Journal on Very Large Data Bases*, 22(3):369–393, 2013.

[4] R. Bača, P. Lukáš, and M. Krátký. Cost-based optimizations of holistic twig joins. Submitted in 12/2014.

[5] N. Bruno, N. Koudas, and D. Srivastava. Holistic twig joins: optimal xml pattern matching. In *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pages 310–321. ACM, 2002.

[6] N. Drukh, N. Polyzotis, M. Garofalakis, and Y. Matias. Fractional xsketch synopses for xml databases. In *Database and XML Technologies*, pages 189–203. Springer, 2004.

[7] P. Lukáš, R. Bača, and M. Krátký. Quickxdb: A prototype of a native xml dbms. In *Proceedings of the Dateso 2013 Annual International Workshop*, pages 36–47. Citeseer, 2013.

[8] C. Luo, Z. Jiang, W.-C. Hou, F. Yu, and Q. Zhu. A sampling approach for xml query selectivity estimation. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, pages 335–344. ACM, 2009.

[9] P. Michiels, G. A. Mihaila, and J. Siméon. Put a tree pattern in your algebra. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 246–255. IEEE, 2007.

[10] N. Polyzotis and M. Garofalakis. Xsketch synopses for xml data graphs. *ACM Transactions on Database Systems (TODS)*, 31(3):1014–1063, 2006.