APPLICATION OF PILE TECHNOLOGY IN SUBSTRINGS SEARCH

Milan Tomeš

Univerzita Pardubice, Fakulta ekonomicko-správní, Ústav systémového inženýrství a inforamtiky

Abstract: *The Pile system brings new alternative, progressive and especial look at data. It works with relations instead of data, and that is why storing it without redundancy. This paper deals with Pile technology and its possibilities in terms of working with text string, which is the first and simplest application of this technology. General principles, and specific work with the text are described. Further here are presented a measurement of the Pile specific implementation.*

Keywords: *Pile, relation, string, text, assimilating*

1. Introduction

This work deals with the use of Pile technology in a field of working with the text. Although this area is explained very well, Pile technology offers a new perspective, which is totally different from the current approach.

Pile technology is very versatile and so its field of application is very broad, from a string matching, despite design of an association database (where data access is not restricted to any key field and a query can be made on any part of the record), to a general description of software systems.

The field of application is therefore quite extensive and its necessary to determine goals at the beginning. The area of interest is so limited and will be evaluated at the end. The basic objectives of this work are as follows:

- Pile technology describing
- Pile technology possibilities in the field of working with text
- Testing and measurement with Pile technology

General description of this technology will be carried out in the first chapter, next chapter will focus on working with the text, further testing a measurements, which was carried out with a specific implementation of Pile technology, will be presented. The findings will be summarized and evaluated at the end.

1.1 General description

Pile technology is a simple mathematical solution providing ordered relations, being related in a recursive manner, to be encoded in only 2+1 dimensions, which together form one logical frame of references. [8]

In this recursive manner of relating also trees are emerged, but such trees are always integrated by their children, since any such child being a relation is a child of its both parents it relates, of which each originates in different tree. [8]

Pile is a new, radically relationist approach to data and structures. Instead of representing and storing data, Pile registers only relations between data elements and stores relational patterns in a novel, non-hierarchic layered structure, which enables to manipulate simulations of data and to reconstruct and generate arbitrary data just from relational patterns. [6]

To understand the Pile approach, we must distinguish between user data and internal data. User data is any data relevant to a user. It's what databases or files usually are storing, e.g. texts, addresses, invoices, music etc. Such kind of data ranging from single characters or numbers to huge images will never be stored in the 'data structures' of Pile. User data remain external and are not any longer required once assimilated into a Pile. [6]

In this case, there are only connections and connections between connections. Data are outside this system. When compared with the traditional work, this system is homogeneous, since here are only connections.

The core of Pile is to register data and to be able to regenerate data, but not to store data. Pile registers data completely and lossless by assimilating it, not by representing it. Assimilating data in contrast to storing or representing data means, that a user data item is decomposed into the relations between its data elements. These relations then are mapped to one or more internal data items representing relations and relations of relations between atomic elements. [6]

Pile brings a new alternative view of the data that works with relations instead with data, which brings great advantage, and that is why storing it without redundancy. In current computing, data is stored redundantly, this is not only an expensive method in terms of required storage capacity, but also in search time. [5]

For example, sentence: "The rain was over and the sun was shining again".

Redundancy is significantly. The letters „a" and „n" occurs 6x, „s" occurs 4x, pair „in" 4x etc. With entire documents, the number of recurring substrings (like words, letters) explodes, since the number of characters used in character sets are hundreds, and the number of words used in a language are thousands. [5]

By contrast our nervous system works with relations. Each interaction of an organism with its environment is represented in the nervous system as a relation of some states of activity. Treating these relations as independent entities and interacting with them (creating new relations with old relations) constitutes thinking.[9]

PILE also works with relations: a PILE connection is a relation between two other PILE connections, it is a referable object and can be treated as an independent entity. A PILE system possesses some features, which are a necessary condition for creating systems with the following characteristics [5]:

1. Connectivity – any two objects in a PILE system can be connected in PILE
2. Heterogeneity – different types of terminal values can be connected in PILE
3. Self-reference – the dynamics of the system is specified by the system itself. PILE can be used as data architecture for modeling self-referential systems.
4. Distributed structure – while having many roots and system definers, a general PILE structure is a distributed structure. It can also be described as uniform: In a neighborhood of any object it looks the same, each object has two parents and eventually a number of children.
5. Growth/scalability – a PILE system can grow extending its structure by adding new PILE connections. A part of a PILE system would have the same growth dynamics (rules) as the whole system would.
6. Multiple inheritance – each PILE object has two parents.
7. Connections and objects are the same instances – the principle of recursivity is used up to its limits.
8. Many roots – there can be many "beginnings" of the system and many internal hierarchies.

1.2 Definition

A PILE structure is a graph, which can be described as a combination of trees. A PILE object is an identification of two nodes from different trees, see Figure 1.

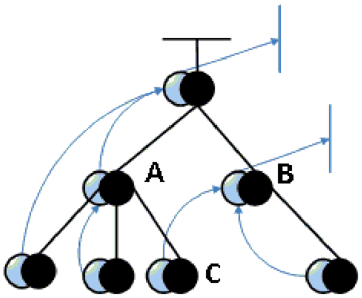

Fig.1. Pile structure, [1]

There is the two trees, black line we call normative system, blue arrow we call associative system. Each node from the normative system is identified with a node of a tree from the associative system. [5]

1. A PILE structure is a combination of the normative and the associative generating systems, so that each normative node is identified with exactly one associative node and vice versa.
2. A pair of nodes, a normative and an associative, which are identified in the PILE structure, is called a PILE object.
3. Only such structures are allowed, by which every (ordered) pair of objects has not more than one common child.

Note: The condition 1 ensures that each PILE object has exactly two parents. The condition 3 says that a child is uniquely defined by its parents. [5]

We say that an object A is directly connected to an object B in PILE, if there is an object C, which is a normative child of A and an associative child of B (Figure **Chyba! Nenalezen zdroj odkazů.**1) **Chyba! Nenalezen zdroj odkazů.**5].

By the definition of the PILE system (condition 3) this object C is unique. We say, the object C is the PILE connection between the objects A and B.

Every PILE object, if it is not a root, connects exactly two other objects in PILE (its parents). So the PILE objects and the PILE connections are the same instances.

If we have a PILE structure in which two objects are not directly connected in PILE, we can always extend the PILE structure by a new object/connection, so that they are connected: we add a new object, which is a normative child of the first object and an associative child of the second object. The objects are connected with respect to their order, so there are always two ways to connect them. [5]

2. Pile technology and text

Pile technology is very general, allows a database design, or information systems design, but the first application of PILE is a fast substring search, promises immediate profit for example in bioinformatics applications.

Here is an example, that build Pile system for string ”SCIENCE”. This is only one of many possibilities of representation this string, see Figure 2. [1]:

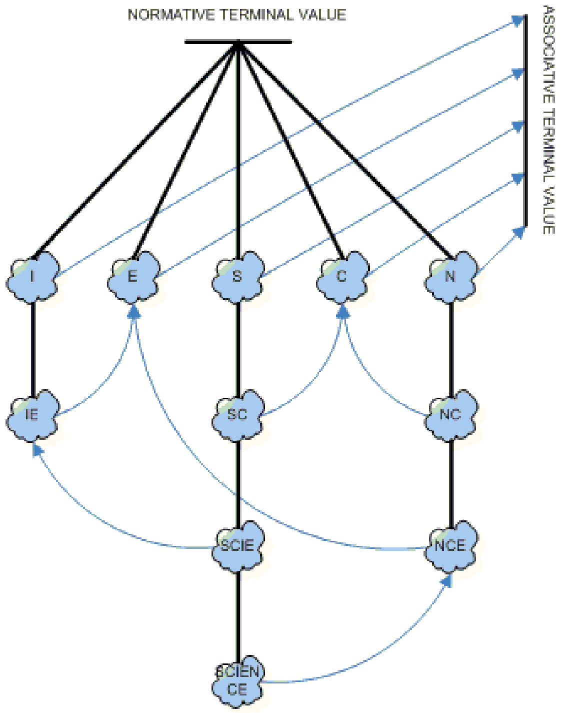

Fig.2. Example of Pile structure, [1]

At first we have to create system definer. In this case we have one normative tree and five associative roots, one for each symbol (“C”, “I”, “E”, “N”, “S”).

We connect the roots representing “I”, “E” via a new object, which will be represents string “IE”. This object is a normative child of “I” and associative child of “E”. In the same way we connect objects representing “SC” and “NC”.

Now we connect new objects “SC” with objects “IE”. By this we get objects “SCIE”, that is normative child of objects “SC” and associative child of objects “IE”. In the same manner, we get objects representing “NCE”.

Finally we connect objects “SCIE” and objects “NCE”. By it, we constructed a Pile structure representing string “SCIENCE”.

Take note that for each symbol there is only one object representing it, and each time, when string already occurs, then this objects is reused. Once created, many times reused.

3. Testing and Measurement

Pile technology is still at an early stage of development, neither the terminology is not expanded, nor any commercial application is available. Yet there are few implementations. But rather serves as an example of the possibilities and feasibility of this technology, than as a functional product.

Here are presented the results of the tests. Those were made by German scientist MSc Dirk Reuter PhD in 2006 with *Pile reference engine from PileSystems Inc.*

3.1 About testing

Two types of experiments were performed [3]:

- assimilating different types of input data and examining the correlation between the size of the input data, the size of the resultant Pile structure, and the number of relations created by the engine to assimilate the input data

- performing substring searches in random data files

The testing platform was a Pentium Celeron 2.4 GHz CPU with 512 MB RAM and WinXP as operating system. To gain a broad perspective of the performance of the Pile System, different input data sources were used. The input data consisted of natural English language, genetic data, and of random Latin letters. [3]

The natural language files were obtained from Project Gutenberg [7] and consisted of 'The Decline and Fall of the Roman Empire' by Edward Gibbon, the 'King James Bible' as compiled by various editors, and 'The Complete Shakespeare' consisting of all 37 plays by William Shakespeare; the genetic sequences were obtained from NCBI Genbank and consisted of genomic data from Pufferfish, Rat, and Arabidopsis, and the random Latin letter sequences were generated with a Python script. In total, ca. 1000 individual measurements were performed. [3]

3.2 Data assimilation results

The parameter determined in these experiments was the dependency between the input data size and the number of relations generated by the engine to assimilate the given input data. As an auxiliary indicator to emphasize the capacity increase of the Pile structure with increasing input size, the ratio between these two values was determined, expressing the number of bytes assimilated on average in one relation generated by the Pile engine. In Figure 3, the depiction of the dependency between these two values is such that the input data size is shown on the ordinate (y-axis) as depending on the number of relations provided by the Pile system to assimilate this amount of input data. [3]

In the sections a) of the figure, the number of relations provided by the engine is shown on the x-axis, and the amount of input data assimilated by these relations is shown on the y-axis. The straight line from the origin to the upper right is the identical function f(x)=x, included as a help for the eye, and the lines covering some of the data points represent the fitted functions. Credible fits could only be yielded for input data sizes larger than ca. 50kB. No polynomial or exponential functions could be found to faithfully approximate input data sizes smaller than this. [3]

In the b) section of the figure 3, the input data size is given on the x-axis, while the y-axis displays the average amount of input data assimilated per relation by the Pile engine. This figure shows that the larger the amount of data already assimilated, the larger the average capacity of each new relation. [3]

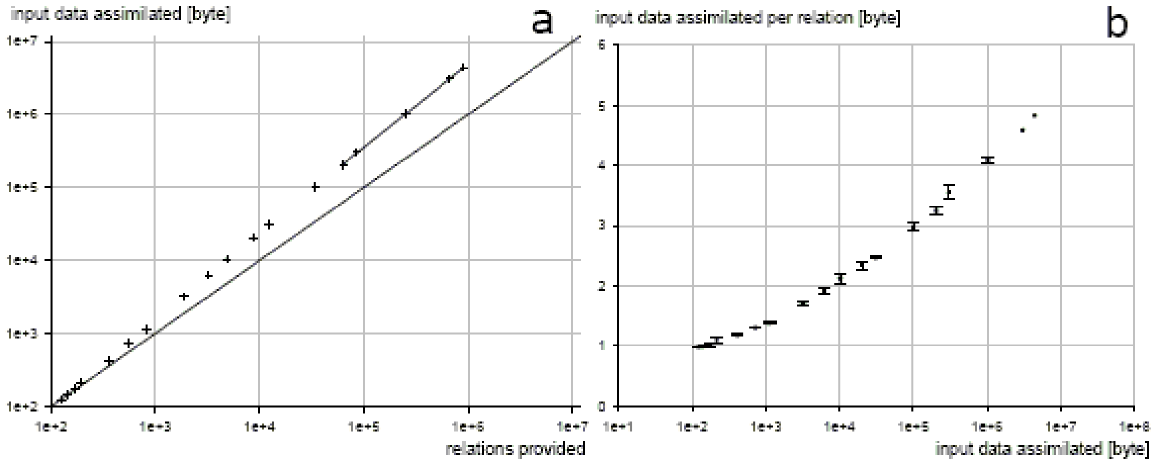

Fig.3. Results (King James bible), [3]

What can be safely concluded from the present data is that the Pile system clearly exhibits an increase of capacity with increasing amount of data already assimilated in the relational structure. [3]

3.3 Substring searches results

The problem of the 'substring search' is one of the most notorious ailments plaguing the field of computerized information processing today. This concerns databases in general, and especially bioinformatics with its massive volumes of genetic code. [3]

Figures 4 and 5 present the numerical results of the substring searches. Left and right part of figure4 present the results for different substring lengths, with one individual plot for each of the file sizes, and figure 5 present the same data again, with fixed file sizes for each figure, and one individual plot for each of the search string lengths. [3]

Figure 4 clearly shows that there is no obvious correlation of the retrieval times of each individual substring occurrence with the size of the file to be searched in. Neither does the number of occurrences have any quantifiable influence on the retrieval times. Figure 5 redisplays the same data again, and reemphasizes the finding, that neither the size of the file to be searched in, nor the size of the substring to be searched have any obvious influence on the retrieval times. [3]

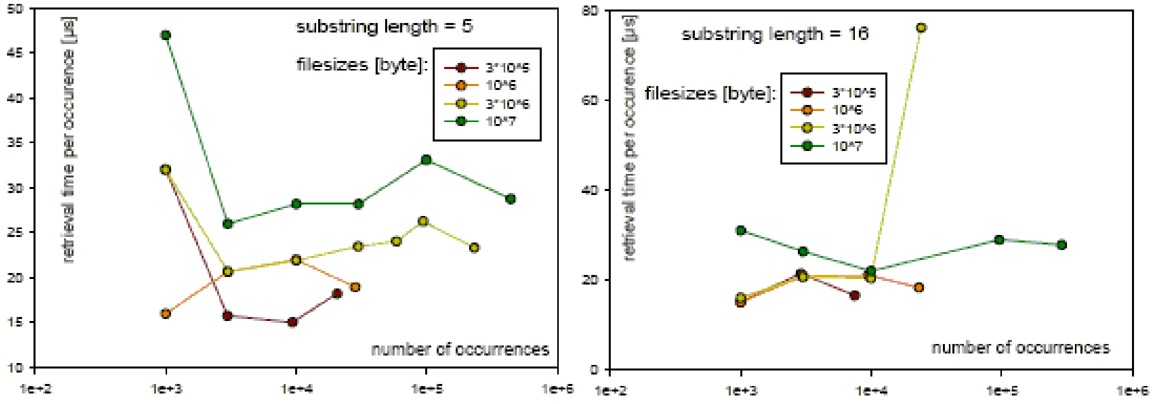

Fig.4. Substring search times with fixed substring lengths, [3]

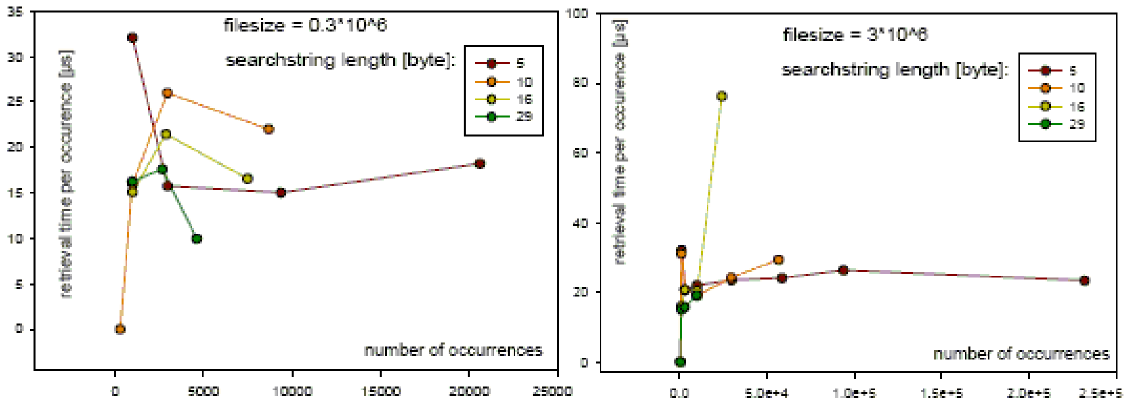

Fig.5. Substring search times with fixed file, [3]

The present experimental design has been chosen with these massive numbers of occurrences of the substring to be searched (up to 450.000) to put some serious strain on the Pile system, because it is too lightning fast for any number of occurrences below several thousand, in which case the retrieval times are smaller than 1 microsecond, which is rounded down to zero. [3]

3.4 Summary

How to interpret these present data? Caution is very much appropriate, because the Pile system is too different to make comparisons with other data processing systems. The relationist approach is in an embryonic stage. Two conclusions can be drawn from the current data [3]:

- There is a capacity increase with increasing size of the structure already assimilated, while maintaining full transparency and accessibility.
- The retrieval times for substring searches depend neither on the size of the substring to be searched, nor on the size of the string to be searched in.

Both of these results are potentially revolutionary for computer science and the IT.

Comparison of Pile with existing data processing systems is not legitimate, as Pile is not, e.g. a compression algorithm; Pile maintains accessibility, and it even lends further transparency to the assimilated data, quite an opposite characteristic of what compression algorithms usually do. Neither is Pile a database (yet), because the data being assimilated into the Pile structure are disintegrated into atomic symbols. [3]

4. Conclusion

From the available resources and tests shows that Pile can be used for working with text, and its use can be very beneficial. This approach appears to be single, universal, homogeneous, flexible and interesting to work with the text.

A development of this technology almost stopped, mainly due to unclear licensing relationships, lack of support both from the scientific, commercial and professional public mistrust in a different view on working with data.

That is why this technology shows considerable immaturity. However, the basics were given and thus considerable scope for further research opens here.

References:

[1] TOMEŠ, M.: The Pile system, Scientific papers of the University of Pardubice – Series D Faculty of Economic and Administration 12, 200--208,(2007)

[2] WESTPHAL, R.: Storing Relations Instead of Data - Just a Cool Idea or a Revolutionary New Data Storage Paradigm?,2005, http://weblogs.asp.net/ralfw/archive/2005/12/08/432665.aspx

[3] REUTER, D.: Processing Data by Assimilating Pure Relations - Benchmarking the Pile System, 2006, http://www.pilesys.com/new/Documents/Pile%20Benchmark.pdf

[4] BEDONI, M., ELUL, E.: Pile for beginners, 2006, http://www.pilesys.com/new/downloads.php?cat_id=3

[5] PROUTSKOVA, P.: The Pile system: A new approach to data and computing, 2004, http://www.pilesys.com/new/documents/Pile%20Math%20Intro.pdf

[6] Pile system Inc., http://www.pilesys.com

[7] Project Gutenberg, http://www.gutenberg.org

[8] ELUL, E.: New approach for internet use based p2p, pile technology and communities, 2008, http://piletech.org/piletech.html

[9] MATURANA, H. R.: Biology cognition, Biological Computer Laboratory Research Report BCL 9.0., Urbana IL: University of Illionois, 1970.

Contact address:

Ing. Milan Tomeš
University of Pardubice
Faculty of Economics and Administration
Studentská 84
532 10 Pardubice
Email: Milan.Tomes@upce.cz