tkt-bsc-level

tkt-programme

Tutkielman otsikko

Jussi Timonen

February 4, 2020

FACULTY

UH

supervisors

Prof. D.U. Mind, Dr. O. Why

examiner

Prof. D.U. Mind, Dr. O. Why

ytiedot

address

eaddress

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty faculty	Koulutusohjelma — Utbildningsprogram — Study programme tkt-programme

Tekijä — Författare — Author

Jussi Timonen

Työn nimi — Arbetets titel — Title

Tutkielman otsikko

Ohjaajat — Handledare — Supervisors

Prof. D.U. Mind, Dr. O. Why

Työn laji — Arbetets art — Level tkt-bsc-level	Aika — Datum — Month and year February 4, 2020	Sivumäärä — Sidoantal — Number of pages 15 pp, 1 appendice pages

Tiivistelmä — Referat — Abstract

Tämä dokumentti on tarkoitettu Helsingin yliopiston tietojenkäsittelytieteen osaston opinnäytteiden ja harjoitustöiden ulkoasun ohjeeksi ja mallipohjaksi. Ohje soveltuu kanditutkielmiin, ohjelmistotuotantoprojekteihin, seminaareihin ja maisterintutkielmiin. Tämän ohjeen lisäksi on seurattava niitä ohjeita, jotka opastavat valitsemaan kuhunkin osioon tieteellisesti kiinnostavaa, syvällisesti pohdittua sisältöä.

Työn aihe luokitellaan ACM Computing Classification System (CCS) mukaisesti, ks. https://www.acm.org/about-acm/class, käyttäen komentoa `\classification{}`. Käytä muutamaa termipolkua (1–3), jotka alkavat juuritermistä ja joissa polun tarkentuvat luokat erotetaan toisistaan oikealle osoittavalla nuolella.

ACM Computing Classification System (CCS)

General and reference → Document types → Surveys and overviews

Applied computing → Document management and text processing → Document management → Text editing

Avainsanat — Nyckelord — Keywords

ulkoasu, tiivistelmä, lähdeluettelo

Säilytyspaikka — Förvaringsställe — Where deposited

library

Muita tietoja — övriga uppgifter — Additional information

alko-line

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty faculty	Koulutusohjelma — Utbildningsprogram — Study programme tkt-programme

Tekijä — Författare — Author

Jussi Timonen

Työn nimi — Arbetets titel — Title

Tutkielman otsikko

Ohjaajat — Handledare — Supervisors

Prof. D.U. Mind, Dr. O. Why

Työn laji — Arbetets art — Level tkt-bsc-level	Aika — Datum — Month and year February 4, 2020	Sivumäärä — Sidoantal — Number of pages 15 pp, 1 appendice pages

Tiivistelmä — Referat — Abstract

Use this otherlanguage environment to write your abstract in another language if needed.

Topics are classified according to the ACM Computing Classification System (CCS), see https://www.acm.org/about-acm/class: check command `\classification{}`. A small set of paths (1–3) should be used, starting from any top nodes referred to bu the root term CCS leading to the leaf nodes. The elements in the path are separated by right arrow, and emphasis of each element individually can be indicated by the use of bold face for high importance or italics for intermediate level. The combination of individual boldface terms may give the reader additional insight.

ACM Computing Classification System (CCS)
General and reference → Document types → Surveys and overviews
Applied computing → Document management and text processing → Document management → Text editing

Avainsanat — Nyckelord — Keywords

ulkoasu, tiivistelmä, lähdeluettelo

Säilytyspaikka — Förvaringsställe — Where deposited

library

Muita tietoja — övriga uppgifter — Additional information

alko-line

Contents

1 Introduction

Enormous datasets are a common case in today's applications. Compressing the datasets is beneficial, because they naturally decrease memory requirements but also are faster when compressed data is read from disk (**Zob95**).

One of the leading methods of data compression is variable-length coding (**Sal99**), where frequent sequences of data are represented with shorter codewords. Because the sequences of data have different lenghts when compressed, it is not trivial to determine the exact location of a certain element. If this is required, the usual data compression algorithms are inefficient. Fortunately this is not a requirement compression algorithms usually need to fulfill.

However, random access of compressed data is needed in compressed data structures. In most compression methods, the only way to this is to decompress data from the beginning. An integer compressing method with fast random access is explained and compared existing state-of-the-art methods.

2 Variable-byte encoding

Variable-byte encoding (**Wil99**) (VB) is a method of compressing integers via omitting leading zero bits. A good data set for VB encoding is a list of different sized numbers. As an example, an inverted index used by document search engine stores word frequencies and locations in documents. VB saves space here because index numbers need to support large values, but are usually significantly smaller. It is also proven to improve search speed, because less bytes need to be read from a hard drive (**Sch02**).

VB splits an integer into blocks of b bits. A continuation bit is added to the block to form a chunk c and then chunk non-empty chunks are stored. The continuation bit is set to 1 for the last block of the integer. During decoding, chunks are read and their blocks are joined until a continuation bit is set to 1. An example implementation is showcased in Figure 2.1.

```
Data: this text
Result: how to write algorithm with LaTeX2e
while not at end of this document do
    read current;
    if understand then
        go to next section;
        current section becomes this one;
    else
        go back to the beginning of current section;
    end
end
```

Figure 2.1: VB encoding.

Small lengths of c can yield better compression rate at the cost of more bit manipulation, while longer chunks need less bit manipulation and offer less effective compression. Generally chunk length of 8 has been used because it gives a good average and handling bytes is convenient (**IRbook**).

bl It splits an integer into blocks of b bits and then encodes the block as $b + c$ where c is a bit denoting whether b has the most significant bits of the number or not.

This method loses in compression performance to other methods (**Bri09**), but decoding is fast.

3 Directly addressable codes?

Rank and select are two functions that work on bit arrays. Rank(i) gives the sum of 1 bits from the beginning of the bit array and select(i) gives the index of ith 1 bit in the bit array. Both functions work in constant time (citation?) and they require only a few percents of extra space over the data. The extra bits *c* added by variable-byte encoding conveniently create a bit array, where 1's represent the endings of numbers. An effective version of random access has already been introduced (**Bri09**).

Random access with select query is also possible. By separating the *c* bit array and *b* block array, *b* contains variable-byte integers in readable form. Another upside is that functions next(i) and previous(i) are conveniently available. Rank implementation has

- explain how random access is good

4 Previous Work

bl

5 Algorithm

- modifications to basic implementation

6 Results

- comparison to basic implementation + Bri09

Table 6.1: Results with 100k entries (in milliseconds).

Experiment	128	256	32768	65536	2^{24}	$2^{32}-1$
$7bitVByteencoding$	34.97	49	53.04	52.18	53.08	76.21
$8bitVByteencoding$	33.57	32.47	42.96	43.11	46.15	65.14
$7bitVByteencodingwitharray$	33.39	46.85	51.24	49.03	48.93	66.84
$8bitVByteencodingwitharray$	32.52	31.88	41.54	39.94	41.15	52.86

Table 6.2: Results with 1M entries (in milliseconds).

Experiment	128	256	32768	65536	2^{24}	$2^{32} - 1$
$7bitVByteencoding$	38.17	55.09	64.38	65.36	68.08	159
$8bitVByteencoding$	37.09	37.75	53.44	54.6	59.32	148.7
$7bitVByteencodingwitharray$	38.09	55.42	62.22	61.25	71.72	135.01
$8bitVByteencodingwitharray$	36.13	36.83	50.58	50.73	56.93	103.18

7 Conclusion

- here

8 Future work

- something to improve / research?

8.1 Figures

Figure gives an example how to add figures to the document. Remember always to cite the figure in the main text.

8.2 Tables

Table 6.2 gives an example how to report experimental results. Remember always to cite the table in the main text.

```
}
```

9 Citations

9.1 Citations to literature

References are listed in a separate .bib-file. In this case it is named `bibliography.bib` including the following content:

```
@article{einstein,
    author =       "Albert Einstein",
    title =        "{Zur Elektrodynamik bewegter K{\"o}rper}. ({German})
        [{On} the electrodynamics of moving bodies]",
    journal =      "Annalen der Physik",
    volume =       "322",
    number =       "10",
    pages =        "891--921",
    year =         "1905",
    DOI =          "http://dx.doi.org/10.1002/andp.19053221004"
}

@book{latexcompanion,
    author    = "Michel Goossens and Frank Mittelbach and Alexander Samarin",
    title     = "The \LaTeX\ Companion",
    year      = "1993",
    publisher = "Addison-Wesley",
    address   = "Reading, Massachusetts"
}

@misc{knuthwebsite,
    author    = "Donald Knuth",
    title     = "Knuth: Computers and Typesetting",
    url       = "http://www-cs-faculty.stanford.edu/%7Eknuth/abcde.html"
}
```

In the last reference url field the code `%7E` will translate into ~ once clicked in the final pdf.

References are created using command `\cite{einstein}`, showing as (**einstein**). Other examples: (**latexcompanion**; **knuthwebsite**).

Citation style can be negotiated with the supervisor. See some options in https://www.sharelatex.com/learn/Bibtex_bibliography_styles.

9.2 Crossreferences

Appendix A on page i contains some additional material.

10 From tex to pdf

In Linux, run `pdflatex filename.tex` and `bibtex filename.tex` repeatedly until no more warnings are shown. This process can be automised using make-command.

11 Conclusions

It is good to conclude with a summary of findings. You can also use separate chapter for discussion and future work. These details you can negotiate with your supervisor.

Appendix A Sample Appendix

usually starts on its own page, with the name and number of the appendix at the top. The appendices here are just models of the table of contents and the presentation. Each appendix Each appendix is paginated separately.

In addition to complementing the main document, each appendix is also its own, independent entity. This means that an appendix cannot be just an image or a piece of programming, but the appendix must explain its contents and meaning.