Datavetenskap

Opponent:

Donald F. Ross

Respondenter:

Anders Nilsson

Haris Trbakovic

Optimering av listhantering i telekomapplikation

Oppositionsrapport, C-nivå

2008:01

1 Sammanfattat omdöme av examensarbetet

The project is well defined, namely to optimise the performance of list handling functions in a telecommunications application. The approach taken by the students is clear, namely to analyse different data structures using big-O notation and choose a suitable data structure to replace the existing linked list structure. Certain key points in the system are however not well explained (the meaning and role of the froID and roId and the reason for using “memcpy” in the original implementation for example) and the reasoning behind certain decisions taken during the project is lacking. The techniques used for measuring the performance are not described in sufficient detail to be understandable and the presentation of the results in the diagrams is inconsistent. These points need to be addressed. There are insufficient references to the various ideas presented in the dissertation. Of the 11 references, only one is to a book and 1 to a paper and the author and title of the latter is not given. The dissertation contains a number of unsupported assertions – evidence should be presented in these cases.

2 Synpunkter på uppsatsen knuten till examensarbetet

The implementation details of both the original system and the new data structures implemented to optimise the system are not stated clearly. The motivation for choosing an AVL-tree is not sufficiently clear and neither is the choice of a second linked list for the roid/froid structure. No alternatives to the choice of an AVL-tree were tested and too much emphasis was place on the use of big-O in taking this decision without testing alternatives with the same big-O values.

2.1 Titel

The title clearly reflects the project work.

2.2 Uppsatsens disposition

The layout of the dissertation is well planned and reflects the development of the project as well as a reasonable approach to the problem at hand.

2.3 Begreppsapparat

In general, the use of terminology is reasonable. Now and again however, there are some statements where the authors have clearly not read what they have written. A list if terms for the telecommunications should be supplied as well as references for these terms to allow the reader who is less acquainted with the area to refresh their memory. The authors use the big-O notation later in the dissertation to compare operations on data strtuctures but fail to present any information about big-O in the background chapter (2). This should be remedied!

2.4 Argumentering och slutsatsdragning

The argumentation is generally reasonable however, the authors fail to present the weaknesses of using the big-O notation as a basis for taking decisions. There is no discussion about the frequency of the operations (find, add, remove, update) in the original system, a point which might have also influenced the choice of new data structure. The final choice was an AVL-tree, which is explained plus a second linked list. The reasoning behind the latter not being explained at all. This should be remedied. Would a second AVL-tree have been more efficient?

The importance and role of the froid and roid in the system is never fully explained in the beginning, which makes it hard to judge subsequent arguments about the use of the linked list to translate roid to froid. Insufficient facts and measurements are presented to support the conclusions drawn.

2.5 Sammanfattningen

The development of project seems somewhat minimal and alternatives are not fully investigated. Insufficient facts and measurements are presented to support the conclusions drawn. More references could be given and more sources cited to support the arguments presented. Unsupported assertions should be justified with suitable references. The diagrams used to present the results are not consistent. There is insufficient explanation is given as to g´how these results were derived. The dissertation just meets the minimal requirements.

2.6 Språkbehandling

In general the use language of language is reasonable, with a few exceptions where the authors appear not to have re-read what they have written. With more and better editing, the level of clarity could be much improved.

2.7 Referat och källförteckning

These were minimal and in one case [11] neither the author nor the title to a paper is given. This reference [11] is to an ftp download site whereas a better choice would have been to the original paper – this was easy enough to find on the web!

There are no references to any work on optimisation – why?

2.8 Övriga kommentarer

One has the sense of minimal effort being expended, and a lack of clarity and detail. A little more effort would have produced a much better dissertation.

3 Genomgång av uppsatsen kapitelvis

3.1 Kapitel 1

Good points

- The goals of the poject are well stated as a bullit list
- The layout of the rest of the dissertation is clearly presented

Points requiring attention

- none

3.2 Kapitel 2

Good points

- The overview of the telecommunications system is clearly presented using diagrams (figures 1 & 2)
- A short overview of the SS7 concept is presented

Points requiring attention

- **Page 5, section 2.3.2**: the abbreviations AT&T and ITU are not explained – is it assumed that ”everyone knows” what these are. References to these organisations could easily be given for those readers interested in further information – **FIX!**
- **Page 5, section 2.3.2**: SS5 is referred to but no reference is given – **FIX!**
- **Page 6, section 2.3.3**: some knowledge of SS7 and telecommunications is assumed here BUT it would still be better to give some references to terms such are ISDN, ISUP, SSP, STP, SCP and MTP for the interested reader – **FIX!**
- **Pages 7-10, section 2.3.4**: further references to each of the terms named would be desirable – **FIX!**
- **A list of terms** used in this chapter and elsewhere in the dissertation would be useful – **FIX!**
- **There is no discussion at all** of ideas involved in **optimisation** and data structures, including the big-O notation (with the advantages and disadvantages clearly stated) in the background chapter – **FIX! Definitely required!**

3.3 Kapitel 3

Points requiring attention

- **Pages 11-13 section 3.1**: references to the terms used would be useful – **FIX!**
- **Pages 11-13 section 3.1**: this section is descriptive but NOT EXPLANATORY – this applies especially to the froid and roid – what are these, what is their importance and why do these exist.
- **Page 12 section 3.1:** the froid is described as a **“index”** but the data structure is a linked list – is the froid the **position** ín the linked list? If so, please make this clear! It is confusing later on in the dissertation in Chapters 5 and 6 when the linked list implementation is suggested for the froid/roid information. It is hard to judge the need for this structure.
- **Page 13 section 3.2:** if big-O notation had been presented earlier, O(n) could have been used to describe the linear search. Since big-O notation is used later in the dissertation, it should be used throughout – **FIX!**

- **Page 13 section 3.2: "**I listorna finns det flera olika typer av listobjekt och de kan bestå av allt från 1 till 2048 objekt." Does this mean the the list may contain a maximun of 2048 objects? The sentence is unclear (syftningsfel – vad hänvisar "de" till? – listobjekt? – **FIX!**
- **Page 13 section 3.2: "** Målen med att förbättra listan är att minska omstartstid på systemet och kostnader på CPU cykler." How are these related to the list optimisation? What does system restart have toi do with the list? – **FIX!**
- **Page 13 section 3.2:** what are these "provisional solutions" and what is there importance and impact? This needs to be clearer! – **FIX!**
- **Page 14 section 3.3:** WHAT IS THE FROID? Is this the position in the list? If not what is this and how is it allocated / assigned? – **FIX! DEFINITELY!!!**
- **Page 14 section 3.3: "**Det som är intressant med strukt-medlemmens data är att den innehåller ett annat id som kallas för roid. Roid skiljer sig lite från det vanliga id dvs. det så kallade froid. Roid behöver inte alltid vara unikt och dessutom kan det förändras under körning. När ett objekt skapas är roid 0 och sedan får det ett unikt nummer som ofta men inte alltid är samma som froid." This is downright confusing! What is the roid, how is ther value derived/assigned and what is the function? – **FIX! DEFINITELY!!! –** this is required in order to understand Chapters 5 and 6.
- **Page 14 section 3.3: "** söka och hämta ledigt froid" What does this mean? Why is this important? Where doies the froid come from? Are these values reused, marked as used ("ledigt") – how effective is this?
- **Should** Section 3.2 (Beskrivning av problemet) come AFTER section 3.3 (Beskrivning av den nuvarande implementationen)??? I think it would be better to switch these around. In addition, potential problems could be flagged and then summarised in the following section (Beskrivning av problemet) – **FIX!**
- **Page 17 section 3.3: "**Hämta ledigt froid fungerar på så sätt att funktionen går igenom listan och letar efter första froid som inte finns med i listan, se **Fel! Hittar inte referenskälla.**." This is surely a contradiction in terms!!!! REPHRASE – **FIX!**
- **Page 17 section 3.3:** the code section has no return statement!
- **Page 17 section 3.3:** is this "temporary variable" the position in the list? Is it ever a good idea to have variables called temporary in the first place?

- **Page 18 section 3.3:** “kopierar nästa objekts data” Why is this done/required – never explained!legacy code?
- **Page 18 section 3.3:** is there an else to the code example in figure 17? It would have been better to include this else if it exists – for completeness’ sake!
- **Page 19 section 3.3: “**. Operationen som vi kallar hämta alla objekt använder minneskopiering. Den används för att söka efter ett objekt på roid.” Syftningsfel – ”den” – antagligen ”den operation” är det som menas. Annars syfter ”den” på ”minneskopiering” eller hur?
- **Page 19 section 3.4:** was the 0 (zero) case measured (to determine the startup overhead costs) or not – figure 19 (page 20) begins from 0.
- **Page 20 section 3.4: Figure 19 –** there is no **detailed** description of how these values were obtained. There is no table of values, either in the Chapter or in an appendix, from which the graph was created. Should this figure present information in the interval **512-2048**? This varies in later figures and is thus somewhat confusing to say the least!
- **Page 20 section 3.4:** “Den tiden vi får delar vi med antalet objekt i listan för att få en snittid för att uppdatera ett objekt.” Is this the time to FIND the object to be updated of the time for the update?
- **Page 21 section 3.4:** This is the first mention of O(n) – there is no preceding explanation.
- **Page 21 section 3.4:** “Ett annat test vi gjorde var att vi sökte efter roid och mätte tiden. Roid kan man inte söka på direkt. För att söka på roid måste man använda sig av operationen hämta alla objekt i listan.” Why is it not possible to search directly using the ”roid”? The function of the roid has not really been explained earlier.
- **Page 21 section 3.4: ”** Kurvan i diagrammet stämmer inte helt med teorin eftersom testerna är utförda på en PC med operativtsystem Linux.” Why does the curve NOT agree with the theory? Is this an explanation (PC/Linux) or an observation?
- **Page 21 section 3.4: ”** eftersom testerna är utförda på en PC med operativtsystem Linux.” – was this the ACTUAL environment at Tieto or not? This is also never made clear!
- **Page 21 section 3.4: Figure 20 –** difficult to tell if this curve is O(n) or $O(n^2)$. Do you have an explanation for the shape of the curve?

- **Page 22 section 3.5:** En annan sak som vi har märkt är att listoperationer sådana som insättning, uppdatering, sökning och hämtning av alla objekt i listan använder minneskopiering som är ineffektivt och kostsamt i CPP miljö.” Why is this? You present this as a statement but offer no explanation. In what way is this ineffective and costly? Speculation?
- **Chapter 3:** No attempt was made to present theoretical curves to compare with the actual measurements. Why was this?
- **Page 22 section 3.6:** ” Syftet med detta examensarbete är att undersöka olika datastrukturer och hitta en datastruktur som eventuellt kan ge en förbättring på söktiderna för ett specifikt objekt.” This ignores the other aspects such as the use of ”memcpy” which you name in the previous section!

3.4 Kapitel 4

Points requiring attention

- **Page 23 Chapter 4, introduction section:** What was the environment in which these tests were carried out?
- **Page 23 section 4.1.1:”** En implementation av en hashtabell kallas för hashing.” No – language abuse! Hashing is the whole process! – **FIX!**
- **Page 23 section 4.1.1:** There is no discussion of the distribution of the key values here – this might have made a difference and therefore given a better result – O(1)
- **Page 24 section 4.1.2: “**Alla element refererar till ett index” – Language abuse – Alla element refereras av ett index
- **Page 25 section 4.1.3: Figure 23 –** Huvud + svansen implies a recursive outlook which is not consistent with the earlier view in Chapter 3 of first + next!
- **Pages 25-26 section 4.1.4:** There is no mention here of the creator of the skip list (Pugh) DESPITE an ftp link to a copy of his paper. – **FIX!**
- **Page 25 section 4.1.4: “**De flesta operationer utförs på tidskomplexiteten O(log(n)) med värsta fall O(n)” Which ones? Little too vague!
- **Chapter 4 in general:** there are no/few references to information about each specific data structure. It would help the interested reader!
- **Page 29 section 4.1.8:** No references to literature on AVL-trees DESPITE this being the structure of choice for the implementation – **FIX!**

- **Page 36 section 4.1.8: Table 1 – GOOD! –** there are no references! Where was this information gathered from? What is the weakness of this approach? Not discussed!
- **Page 37 section 4.2: "**Hashtabell har tidskomplexitet som är O(1) för de flesta operationer. Problemet med hashtabell är att vissa operationer i värstafall kan bli så illa som O(n)." This was never tested experimentally – especially with regard to the collisions! Why not?
- **Page 37 section 4.2: "** B-träd och balanserade binära sökträd har i värsta fall en tidskomplexitet som är betydligt bättre än den nuvarande implementationen. Anledningen till att vi tog beslutet att inte implementera B-träd var att de först och främst används för stora mängder data som inte längre får plats på RAM- minnet, t.ex. databaser." Neither figures nor calculation for this statement are presented in support. Although B-trees are used in databases for indexes, they do not imply the need for a database. It would have been interesting to work out the space requirements for the B-tree. In general it would have been interesting to know the space requirements for each structure.
- **Page 37 section 4.2: "** AVL- och röd-svart-träd har samma tidskomplexitet nämligen O(log n). Ett AVL-träd ger bättre söktidskomplexitet då det är bättre balanserade. Man kan visa att höjden av ett AVL-träd är ungefär 1.44log(n+2) – 0.328, som i praktiken är lite mer än log(n). Höjden av ett röd-svart-träd är ungefär i snitt 2log(n+1). " No references are given for these figures. I presume that they are taken from reference [9] p 144 and p503 respectively.
- **Page 37 section 4.2: "** Sökningen görs oftare när systemet är hårt belastat medan borttagning sker ofta när systemet inte är så hårt belastat. Insättning är också viktigt att få ned tiderna på eftersom det också sker när systemet är hårt belastat."

Again an unsupported assertion. Where is the evidence for this? This has not been presented earlier.

- **Pages 37-38 section 4.2: ”** En implementation av ett AVL-träd kommer att ge oss insättningskomplexitet som är lite sämre än röd-svart-träd” How do you know? Again where is the evidence for this? Were any experiments carried out?

3.5 Kapitel 5 (osv.)

Points requiring attention

- **Page 39 Chapter 5 introduction: “** Den nya implementationen innehåller även en hjälplista som är en länkad lista.” Will this not have the same disadvantages as the old system? Why is this required? Why was this not made an AVL-tree as well?
- **Page 39 Chapter 5 introduction ”** Hjälplistan har vi valt att lägga till i den nya implementationen eftersom vi ville få ytterligare förbättringar på söktiden för ett roid.” How do you know? Where is the evidence?
- **Page 41 section 5.2:** ” och hämtning av ledigt froid” - this has still never been explained!
- **Page 41 section 5.2.1: “**Eftersom vi tidigare kunde konstatera att minneskopiering kostar många CPU-cykler konstruerade vi den nya insättningsfunktionen utan minneskopiering, se **Fel! Hittar inte referenskälla.**” – this was an ***observation*** – there is still no explanation for this!
- **Page 42 section 5.2.1: ”** Detta gör att man lättare kan söka via roid.” In what way? This is not explained?
- **Page 42 section 5.2.1:** ” Vi valde att sätta in objektet först i hjälplistan eftersom den inte behöver vara sorterad. Om listan skulle vara sorterad, skulle man då behöva flytta om objektet när ett objekt får ett nytt roid. Detta skulle medföra en massa extra arbete.” Since the function of the froid and

roid has not really been explained, it is difficult to understand this statement.

- **Page 43 section 5.2.1: Figure 43 –** why is there a variable called ”currentTree” in code which purports to add to a list?
- **Page 44 section 5.2.3: “**Den nya funktionen för att hämta ledigt froid utförs endast på AVL-trädet” Why is this necessary? Could a variable be used instead to store the next available froid. Again, since their use and function is not clearly explained anywhere, it is difficult to understand this reasoning.
- **Page 45 section 5.2.4: ”**I den gamla implementationen gjorde man en kopia av dataobjektet vilket man jobbade med.” Why is this done? No explanation is presented.
- **Page 45 section 5.2.5: ”** Med hjälp av dessa funktioner kunde vi gå genom hela AVL-trädet och titta i varje dataobjekt efter det sökta roid.” Surely the point of an AVL-tree is that the whole structure need not be searched!

3.6 Chapter 6

Points requiring attention

- **Page 46 section 6.1: “**I insättningsfunktionen testar vi att fylla listor med objekt som är sorterade på froid. För att se hur funktioner som använder sig av minneskopiering beter sig fyller vi dataobjekt med information på 1 MB.” Why? – no explanation!
- **Page 46 section 6.1:** Is update done by replacement?
- **Page 48 section 6.2: ”**Vi testade även att mäta tiden för uppdatering av ett objekt i endast ett AVL-träd för att se hur lång tid det tar. Tiden vi fick var betydligt bättre än den tiden vi fick när vi uppdaterar ett objekt i både AVL-träd och hjälplista.” What were the results for the AVL-tree? Is the general result not an argument AGAINST the help-list?
- **Page 48 section 6.2: ”** Men vi har i alla fall blivit av med minneskopiering som är ineffektivt och kostsamt i CPP miljö.” Again it is not mentioned WHY this is ineffective!

- **Page 49 section 6.3: ”** Enda testfallet där den gamla implementationen är bättre än den nya är när sökning sker på ett objekt som har lägre froid än vad höjden på AVL-trädet är, vilket inte är vanligt.” Again why was this? There is no explanation given!
- **Page 49 section 6.3: Figure 48 –** why does the curve start at 0?
- **Page 50 section 6.4: “**Tidkomplexitet för den gamla implementationen är O(n) som vi kom fram i kapitel 3.4. För den nya implementationen måste man ta bort ett objekt i hjälplistan och ett AVL-träd vilket ger oss en tidkomplexitet O(n).” So is this really an improvement? There is still no clear argument for the existence of the help-list!
- Should there be a summary paragraph for Chapter 6?

3.7 Chapter 7

Points requiring attention

- **Page 54:** Table 2 – what happened to the other sizes 512, 1024, 1536? Why were results not presented for these? Do these results represent an average case or the worst case (see Table 1)?
- **Page 54:** “En annan stor förbättring vi har gjort är att vi inte längre använder minneskopiering vilket kostade många CPU cykler.” Still no explanation as to why this is!

3.8 Övriga kommentarer

Points requiring attention

- The notation for the froid and roid is inconsistent – sometimes “froid” and “roid” are used and sometimes “froId” and “roId” (capital I)
- Most of the dissertation seems to consist of description and there are few explanations or justifications of certain assertions (unsupported assertions).

4 Slutliga kommentarer

The dissertation has the potential to be interesting but is marred by a lack of explanations. The role of froid and roid is never made clear which makes it difficult to understand the main thread of the dissertation. In Chapter 2, there is no background information on optimisation nor is there information about big-O which is subsequently used to make decisions as to which dat5a structure should replace the list.

5 Questions

- Were other solutions such as cache memory considered as an optimisation?
- Where are these lists stored? Primary or secondary memory? **Not clear!**
- How big is the information used in the objects referred to by the froid (and roid) ?
- Why is “memcpy” used in the old implementation?
- **Page 13 section 3.2:** other “problems” are ”suddenly” presented here – should these not be listed so that the reader knows exactly which problems exists, which are going to be fixed and how?
- What was the test environment used in this dissertation project? At Tieto or on your own PC with Linux? This is not made clear.
- **Page 21 section 3.4: Figure 20 –** difficult to tell if this curve is O(n) or O(n^2). Do you have an explanation for the shape of the curve?
- What were the space requirements for each structure?
- **Page 39 Chapter 5 introduction: “** Den nya implementationen innehåller även en hjälplista som är en länkad lista.” Will this not have the same disadvantages as the old system? Why is this required? Why was this not made an AVL-tree as well?
- Why is there no example with a froid and roid so that the reader can see how they work together?
- **Page 48 section 6.2: ”**Vi testade även att mäta tiden för uppdatering av ett objekt i endast ett AVL-träd för att se hur lång tid det tar. Tiden vi fick var betydligt bättre än den tiden vi fick när vi uppdaterar ett objekt i både AVL-träd och hjälplista.” What were the results for the AVL-tree? Is the general result not an argument AGAINST the help-list