UCD School of Computer Science
Scoil na Ríomheolaíochta UCD

University College Dublin,
Belfield, Dublin 4, Ireland

An Coláiste Ollscoile, Baile Átha Cliath,
Belfield, Baile Átha Cliath 4, Eire

T +353 1 716 2928
F +353 1 269 7262

j.murphy@ucd.ie
www.ucd.ie/csi

11th October 2015

phone: +353-1-716-2928
mobile: 087-233-7305

Referee Report on Doctoral Thesis

"Garbage Collection in Software Performance Engineering" by Peter Libic

Technical Description/Evaluation of document

The PhD thesis submitted is concerned with garbage collection (GC) with a special emphasis on the performance modelling of it. GC is usually treated as a constant background factor that a model can either ignore or try and capture its effect by calibration and is an important and active area of research in Engineering and Computer Science. The thesis topic is one of considerable interest and relevance to both the academic and industrial community and there is considerable research published in the literature in recent years. The candidate specifically addresses a number of issues and to me there are three main contributions to the thesis research:

1. **Empirical Investigating GC Overhead**: The candidate evaluates the nature of the GC overhead with respect to its effect on accuracy of performance models. In particular, the possibility to model the GC overhead as a black-box is assessed. Additionally, a set of workload characteristics that contribute to GC performance are identified/discussed (i.e., number/size of live objects, object lifetimes, heap depth, object allocation speed, garbage structure, maximum heap size with constant/ growing heap).
2. **Limits of GC Performance Modelling**: The candidate presents an analytical model of one-generation collector and a simulation model of both one-generation and two-generation collectors. The accuracy of the models is evaluated and an analysis of their sensitivity to the inputs is performed. The obtained results show the gap between understanding the GC overhead based on knowing the algorithm and an actual implementation.
3. **Estimating Effects of Code Additions:** The candidate presents a model to help the developer to predict the effects of adding new code (to a stable code baseline) on the GC overhead of the application.

The thesis consists of 6 chapters; an introduction and a chapter on Collectors. The core of the thesis are the three chapters on the GC overhead, the performance modelling and a set of references (79). There is no appendix.

Written Style and Overall Presentation/Structure/Quality of document

The manuscript is well structured and written, with a good flow and attention to detail. I would not suggest changing the structure of the document as I think it works well. The candidate has shown good appreciation of the relationship of the work to the wider research area.

From a publication perspective, this thesis work produced five papers (four with the candidate being listed as first author).

UCD School of Computer Science
Scoil na Ríomheolaíochta UCD

University College Dublin,
Belfield, Dublin 4, Ireland

An Coláiste Ollscoile, Baile Átha Cliath,
Belfield, Baile Átha Cliath 4, Eire

T +353 1 716 2928
F +353 1 269 7262

j.murphy@ucd.ie
www.ucd.ie/csi

Areas where the thesis could be examined in the defense:

1. **General Comments.**
    a. The work presented in this thesis centered on a single GC algorithm (the parallel GC, which is known as the *throughput collector*). It would be interesting to discuss other GC algorithms (e.g., serial GC, concurrent GC, Garbage First GC) in order to evaluate their similarities (hence obtain more generic contributions) or differences (hence providing more GC findings/observations).
    b. One finding discussed on the section "Empirical Investigating GC Overhead" is that there are differences (somehow expected) between the behaviors of the different JVMs (e.g., overhead). However, in the thesis the contributions are evaluated with different JVMs. For instance, in the "Empirical investigating GC overhead" section, the Hotspot and IBM JVMs are used. Meanwhile, on the "Limits of the GC Performance Modelling" section, the Jikes RVM and OpenJDK JVMs are used. It would be interesting to explore the reasons for this set of JVM's rather than have a homogeneous set of tested JVMs to keep consistency.
2. **Chapter "Empirical Investigating GC Overhead"**:
    a. The empirical investigation performed in this section only used artificial workloads. The defense will explore why some real-world applications were not used to further validate the obtained conclusions.
    b. Regarding the performed empirical investigation of the GC overhead, it would be interesting to explore if there was any research on the differences between this work and similar ones performed in the past. This is because several other studies have previously investigated the factors that influenced the GC behavior. For example, it is well-known that the GC is particularly sensitive to the heap size and even small changes, which might appear trivial, could affect its behavior. As additional context, please find below a non-exhaustive list of references in this area:
        - Jeremy Singer, Gavin Brown, Ian Watson JC. Intelligent Selection of Application-Specific Garbage Collectors. International Symposium on Memory Management, 2007; 91–102.
        - Mao F, Zhang EZ, Shen X. Influence of program inputs on the selection of garbage collectors. SIGPLAN Virtual Execution Environments 2009; :91–100.
        - Lengauer P, Mossenbock H. The taming of the shrew: increasing performance by automatic parameter tuning for java garbage collectors. International Conference on Performance Engineering 2014; :111–122.
        - Andreasson E, Hoffmann F, Lindholm O. To collect or not to collect? machine learning for memory management. JVM Research and Technology Symposium, 2002; 27–39.

UCD School of Computer Science | Scoil na Ríomheolaíochta UCD

University College Dublin,
Belfield, Dublin 4, Ireland

An Coláiste Ollscoile, Baile Átha Cliath,
Belfield, Baile Átha Cliath 4, Eire

T +353 1 716 2928
F +353 1 269 7262

j.murphy@ucd.ie
www.ucd.ie/csi

- Soman S, Krintz C, Bacon DF. Dynamic selection of application-specific Garbage Collection. International Symposium of Memory Management, 2004.
- Blackburn SM, Cheng P, Mckinley KS. Myths and Realities: The Performance Impact of Garbage Collection. SIGMETRICS Performance Evaluation Review 2004; 32(1):25–36.

3. **Chapter "Limits of GC Performance Modelling":**
   a. As mentioned two-generation heaps are the most popular/common in the industry and the defense will explore why the thesis did not focus on this type of heap more. However, approximately half of the work in this section is with single-generation heaps, which are far simpler and less used in practice.
   b. The proposed simulation needs an impractically large amount of data as per the candidate own word. The difficulties of modelling the GC and the fact that the simulation can only give a fair accurate count of minor collections (while not working well for the full collections – MaGC-) will be discussed in the defense.

4. **Chapter "Estimating Effects of Code Additions":**
   a. One of the main limitations of the model is the set of very strong assumptions: in order to provide a reasonable prediction, the model requires that the original code has stable allocation behavior and the new code only allocates short-lived objects. During the defense the fact that the model only works with short-lived objects (i.e. indirectly meaning that it only works for MiGC), its applicability/usefulness will be explored, as it is considerably limited because it misses to assess the impact of the long-lived objects, which are the ones that provoke the Major GC (which has been proved to be the most expensive type of GC).

Overall the research presented here is a valuable contribution to knowledge and the candidate's thesis demonstrated that they are deserving of a doctoral degree. If you would like any more information on this matter, please do not hesitate to contact me.

Yours Sincerely,

____

Prof. John Murphy, CEng FIEI
B.E. (NUI), M.Sc. (Caltech), Ph.D. (DCU)
Associate Professor
FIET, SM-IEEE, FICS, SM-ACM, IBM Faculty Fellow