**Dr. Patric Hung’s Comments:**

| **SL.** | **Comments** | **Addressing approach** | **Status** |
| --- | --- | --- | --- |
| 1 | The thesis should introduce the basic concept of synchronization in programming and semaphore definition. | Added few sentences regarding semaphore | Done |
| 2 | Referring to 5.4.1 Prepare Environment, it says "Validation measurement of the clustering is done leveraging the  popular R programming language, and its packages such as cluster, factoextra, NbClust." What is R programming?  There is no description or reference. | Added the references and few lines regarding r language | Done |
| 3 | The concept of unsupervised Machine Learning is not clear in the proposed model. There is no details about it in the  thesis. | Addressed with references | Done |
| 4 | Referring to Page 103 (118 of 129), it says “However, we do believe our dataset can be extended by executing some  concurrent example codes with faults in them.” How large is the dataset? | Grammatical issue. Needs Fixing | Done |
|  | Referring to Page 103 (118 of 129), it says “Moreover, there are not so many real-world java applications with faults;  we can use them as benchmark applications." Is there any references? | Grammatical issue. Needs fixing | Done |
| 5 | There are 41 references (23 are technical handbooks) and there are only 18 academic references. Also there is no  reference in 2019-2021. | We have not found recent academic papers on lock contention detection.  However, added some new references and most of them are academic references | Done |
| 6 | There is a typo in "public void taskOne(Object val)." It should be val1. | Will be addressed | Done |
| 7 | Referring to Page 66 (81 of 129), there is a missing “and” in the following statement.  "Therefore, two factors are important in validating clustering approaches a) Assess  the clustering tendency before the analysis b) validate the quality of the clustering results." | Will be addressed | Done |
| 8 | We use a run-time analysis approach instead of static code analysis because these faults surface at run-time. Why static analysis? | Grammatical issue. We have chosen runtime analysis instead of static analysis. | Done |

**Some other comments by Patrick came in during the defense, address them**

1. **Why did you pick Java? Applicability of your work difference between ADA and other concurrent multithreading programming languages. Addressed**
2. **Metrics definitions (SLOW, and others) are missing : Addressed**
3. **HTTP performance does matter, or not? What are the other influencing factors? Addressed**
4. **Page 13, bugs or faults, clarify/fix Addressed**
5. **Page 38, wait ten seconds, why? Use a general symbol and provide some rationale using the specific numbers Addressed**
6. **Page 40 algorithm, delay metric is improperly used, also again used a number, can be generalized using symbols Addressed**

**Dr. Saana Alwidian’s Comments:**

| **SL.** | **Comments** | **Addressing approach** | **Status** |
| --- | --- | --- | --- |
| 1 | Page 2 (Section 1.1-Introduction): The focus of this thesis is on Lock contention, which typically happens when thread A has a lock and thread B wants to acquire that same lock, thread B will have to wait until thread A releases the lock. While I agree that the most obvious example of contention is on a lock), but have you thought of scenarios when “locks” are not the only resources on which contention can be experienced? For example, thread A may experience slowdowns even if it never has to wait for the other threads (say thread B) to release the lock! This is because a lock protects some kind of data, and the data itself will often be contended as well. | Probably Dr. Alwidian took the idea of “data itself contending” from the link below  <https://stackoverflow.com/questions/1970345/what-is-thread-contention>  General comment and this scenario was not considered and cannot be detected with the tools that are used.  No changes made. | Done |
| 2 | Based on comment #1, I would like to see a brief discussion about the contention problem in general, whether it is caused by “locks” or when it just happened simply when two threads try to access either the same resource or related resources in such a way that at least one of the contending threads runs more slowly than it would if the other thread(s) were not running. | Dr. Alwidian asked to incorporate some description regarding the comment #1, but this requires an example to verify. At this moment we do not have such example to verify | Done |
| 3 | Page 3: It was mentioned that “...Therefore, we believe that it is possible to classify contention fault types into these two causes using a clustering approach that will identify the essential features from the JLM and perf run-time metrics”. Again, what if there are other “non-lock-driven” contentions, how would the classifier behave in this case? | Ref comment#1  Will make some comments on the DISCUSSIONS section regarding this. | Done |
| 4 | Page 4: It was mentioned that “In our approach, we create a dataset considering different features that are responsible for creating these faults”. How scalable is this dataset? What if new features emerged? is there a kind of “combinatorial” features that could cause contention faults? by combinatorial features I mean of you combine two or more features (where each feature per se is not a problem), their combination could cause a problem. | What we have seen so far, features TIER2 and TIER3 combinedly representing fault type 2. If new features emerge and if those can be recognized as a combinatorial for a specific fault then we will consider that and take advantage of them.  Grammatical issue. Need to change/clarify | Done |
| 5 | Page 5: It was mentioned that “Multiple threads have the permission to read simultaneously as long as any thread does not attempt to write at that moment, or there is no incoming write request”. What if after granting a read permission, there becomes lots of incoming write requests? Will this a contention fault again? | See answer at #6 |  |
| 6 | Page 8: According to the suggested solution #5, how would implementing the read-write lock offer a solution for reducing the lock competition and sending access requests to the locks? please explain | It solves the fault type 2 incase of multiple read / write requests at the same time. However, after granting a read permission if multiple write requests come at the same time then it will create fault type 2. So it does not solve the problem in such a scenario.  We need to remove it. | Done |
| 7 | Page 12: I assume that Peter Hover et al., [18] have identified the same contention causes that you've identified. Other than the metrics, what is the difference between their approaches to solve each type of contention, and your proposed approach? Please explain further. | I think the sentence is a little bit decisive in what I have written in the paper. I wrote “their approach is similar to ours” but not that similar. They considered runtime metrics but did not consider performance metrics and also they did not solve fault types identification | Done |
| 8 | Page 29, It is mentioned that “we emulated with 4 threads and applied 1 millisecond inside the critical section”. Why 1 millisecond in particular? you always need to provide a justification/explanation regarding the settings of your experimental parameters. | In our daily experiment we saw if we applied 1 ms to a contended region then JLM metrics appear with contention metrics. It is a general situation, We took 1ms just to make sure that the contention is creating and we can take some picture of that info to attach to our thesis paper nothing else. I will clear it in the paper | Done |
| 9 | Page 30: It was mentioned that “Although we are interested in two classes referring to the two potential faults, our generated data should contain more categories than the two”. What are these more categories? Please give examples. | Need to correct the grammar. However, more categories means subcategories of the two faults we are dealing with | Done |
| 10 | Page 30 and Figure 3.1: It was mentioned that “These steps are shown in a high-level workflow (see Figure 3.1)…”. In the high-level workflow figure (Figure 3.1), I suggest that you illustrate explicitly (using big, dashed boxes) the three main steps (1. acquiring run-time metrics, 2. Filtering of metrics, 3. data processing and classification) on top of the detailed sub-steps that you already had in the figure. | Will be addressed | Done |
| 11 | Page 31: It was mentioned that “It is ideal that the data we fed into an ML model should be streamed from a single source rather than multiple files”. Again, you need to explain the rational behind your choices. Why a single source rather than multiple files? | Grammar problem, will fix it. ALso add some clarification regarding the sentence. | Done |
| 12 | Page 31: It was mentioned that “Before analyzing the data using pure KMeans, we preprocess our data, scale, and reduce the features leveraging Principal Component Analysis (PCA)”. What are the exact criteria of data processing you used in order to get more improved data? For example, what were the feature reduction criteria? | Feature reduction is important as it improves performance and helps visualize the dataset. Maybe this is not mentioned in that sentence, I will clarify | Done |
| 13 | Page 31: “First, we collect performance metrics data and generate a dataset by running some concurrent codes that create contention.”. This is a source of internal threats to validity. Please mention this in the discussion section. | Need a discussion with professor | answered |
| 14 | Page 32 (Section 3.3.1): It was mentioned that “A Java exemplary code emulating lock contention is executed in a controlled environment leveraging”. Is this your own code? if not, what is the source of the lock contention-causing codes? | It is a sample example code written by us, however, if we search on the internet, the structure of contention-causing codes are always similar.  N.B.: In that sentence listing of the code was not referenced. It is now. And the explanation of the code is already given at Chapter 4 “Dataset Generation” | Done |
| 15 | Page 51: You mentioned that the perf command "perf-record" collects lots of symbols that are **not** related to contention faults, and then you mentioned that “a group of symbols appear with a high number of samples in case the code experience bad contention and with fewer samples when it experiences minimal contention”. This is a bit confusing, can you clarify this argument please? | Explanation problem and will be addressed | Done |
| 16 | Page 51: following the same discussion, you also mentioned that "Leveraging these symbols might help us to identify the contention fault types". How? if they are originally not related to contention faults? | We are collecting symbols that are all related to contention. So in our opinion they should help.  N.B.: Explanation problem. Will be addressed | Done |
| 17 | Page 37: In comparison between figure 3.4 and 3.5 which illustrated "perf" snapshots when there's high contention, and low contention, respectively. What is the threshold that is used to decide if there a high vs. low contention?Page 37: In comparison between figure 3.4 and 3.5 which illustrated "perf" snapshots when there's high contention, and low contention, respectively. What is the threshold that is used to decide if there a high vs. low contention? | Grammar + clarification problem. It should be meant, if it experiences some amount of contention or some less amount of contention  N.B.: Added an explanation regarding low and high contention. | Done |
| 18 | Page 38: In the algorithm that automates the steps for faster log generation, in particular, in step 1 of the algorithm where you set the number of threads and sleep time. What is the model you followed to vary the numbers of threads and sleep times? Is it based on something from the literature? | Those are configured by us. However, in real world applications the threads and execution times always vary within the same range we took. So our range is completely fine to emulate the contention cases and perform classification on it.  N.B.: Will add a discussion regarding this after the two configurations we considered. | Done |
| 19 | Page 47: You have already mentioned that you used the DBSCAN as another clustering algorithm (in addition to the KMeans), and that the DBSCAN failed to produce the desired clusters out of the dataset. Have you tired other clustering algorithms such as, for example, Mini-Batch K-Means, Mean Shift, OPTICS, etc.? | We have not. It is our future plan  Added to the future work. | Done |
| 20 | Page 63: It was mentioned that “ JLM metrics do change based on the fault types, and once the metrics related to a particular fault are affected by that fault type, some impacts are  observed on the other metrics at the same time”. This is a very interesting insight! have any one in the literature reached to this conclusion as well? | One of the literatures analyzed the performance metrics but they mentioned that they failed to find any correlations among the performance metrics. But we have found these correlations and already discussed in Chapter 5 “Clustering Results” | Done |
| 21 | Page 64: You had an interesting insight which is: “When the two faults occur at the same time, the metrics related to spin count increase in number **only** when the threads spend shorter period of time inside the critical section.”. But what if the threads spend longer time? what will happen in this case? | Spin-count metrics do not increase | Done |
| 22 | It would be good to mention the several threats to validity that must be assessed to better characterize the limitations of the approach and results presented in this thesis. The candidate can refer to Perry et al.’s paper (Empirical studies of software engineering: a roadmap. In *Proceedings of the conference on The future of Software engineering)*, where the most relevant categories of threats to validity for this work are mentioned, such as: construct validity, internal validity and external validity. | Will be addressed. This is specific to the validation. This was related to her comment about bias. First of all, the work is not an empirical study. | answered |
| 23 | Previous related work should be mentioned using the simple past tense. | Will be addressed | Done |
| 24 | Place your figures closer to the discussion that referenced them. | Will be addressed |  |
|  | I suggest adding a table at the summary section of the literature review chapter, where the candidate summarized the related approaches, and the gaps. | Will be addressed | Done |

Here are some comments I made:

1. You use the term delay in Listing 1 and in the example code you have the term sleep. Need to clarify that these terms. In my view the term delay is not the proper term and can be confusing. Addressed. Changed to SLEEP