Exjobb opposition report

Reviewer name: Klas Segeljakt

Date of review: 2018-01-17

Title: Decentralized stream processing: algorithms, methods and tools

Author: Pietro Cannalire

1 Evaluation

Category	Rating
Relevance of content	5
Disposition	4.5
Evaluation of published results	3.5
Abstract	4
Conclusion	4.5
Presentation of related work	5
Language	3.5

I think the report covers an interesting topic. It presents a smart solution to one of the problems associated with geographically distributed systems. The text follows the "red-line". All of the background work is phenomenal, and the figures used in the report are very descriptive. The main lacking parts are the abstract, the evaluation, and the language. The report has potential to be great if these are corrected.

2 Recommendations

2.1 Relevance of Content - 5

The problem which the thesis addresses is how to decentralize distributed streaming applications. This should be highly relevant to the Software and Computer Systems programme which the author is taking.

- No recommendations.

2.2 Disposition - 4.5

The disposition is good. I think all chapters have a good length, and the text is nicely divided into subsections.

- I think section 6.5.3 should be renamed into "Evaluation", since it not only summarizes the results, but also makes claims about them.
- Add a section about Ethics and Sustainability, it is required for a passing grade I think. For ethics, you could for example write about maintaining the privacy of user-data. For sustainability, you could write about how decentralization might impact energy consumption.
- Add sections about goals/objectives/aims in the introduction, although this might be optional.

2.3 Evaluation of published results - 3.5

The evaluation is overall good. It is great that you executed experiments on various configurations and algorithms.

- More information about the setup should be added:
  - Add specifications about the two computers, e.g, OS, CPU, RAM, HDD/SSD.
  - Add information about the request rate, stream element rate, and latency.
  - It is not obvious if the experiments were simulated on a virtual machine, or if they were run on actual machines.
  - Is Kafka run on the same cluster as Spark?
  - Does the setup resemble a geographically distributed system?
- The result from each algorithm and configuration is visualized in its own figure. Since the thesis is about comparing different configurations with each other, it would be better to place all results for each algorithm in the same figure. I understand that the reason for this might be due to technical limitation of Grafana, which is only able to plot results per run.
- If it is not possible, it would be good at least to put all figures for each algorithm on the same page.
- Another option is to show the job time side by side for each of the three configurations, then processing rate, etc.
- The evaluation does not refer to the results by their figure number. If this is added, it will be easier to follow the reasoning.
- It could also be good to discuss the validity of the results, i.e, the possibility of measurement errors.

2.4 Abstract - 4

The abstract contains the problem description and motivation.

- Add a brief description of the proposed solution.
- Also add some information about the experiments and their results.

2.5 Conclusion - 4.5

The conclusion is well written and addresses advantages of the solution.

- One improvement might be to also address the disadvantages of decentralizing. For example, is the solution flexible? Which problems can it/can it not solve?
- Some discussion could be added about how the solution affects the privacy of user-data, as it was a mentioned concern in the abstract.

2.6 Presentation of related work - 5

The related work is well presented. There is no deep technical description of it, but I don't think it is necessarily needed.

- No recommendations.

2.7 Language - 3.5

The language is syntactically correct, and all the content can be understood. There are some grammatical errors which, if corrected, would improve the flow of the text.

- I have marked all the errors and strange sentences that I noticed in the attached report. These errors are mainly represented by:
  - Instances where plural is mixed with singular.
  - Missing "the", "a", and "an" before common nouns.
  - Wrong adjective/adverbial ordering.
- There are also some long sentences which could be shortened. Although this is optional, it might improve the readability of the report. I would recommend to try and keep all sentences below 30 words.

3 Detailed comments for the author

- Reference 11, 31, and 40 seem to be missing information.
- Some references break the horizontal page margin. It is not crucial, but it would look better if they did not.
- The second page seems to be a repeat of the first page.
- Add a Swedish abstrakt and nyckelord if possible.
- It would be good to only list the title of the figure, and not the description, in the List of Figures.
- Be consistent with how items in lists are capitalized. On page 3, the items are not capitalized, on page 4, they are capitalized.
- In chapter 5, it is mentioned that standing queries are performed by a sequence of operations in the Scala language. This is partially true, but it would be better to write that they were performed in Scala with the Spark API.
- Section 5.3.2 has "Snippet of code is in ??".

- The report refers to the "next section" multiple times. For clarity I think it is better to refer to the next section explicitly, e.g, "section 5.4.1.1".
- It would aid the reader if a Problem Statement was added to section 1.1. It should just briefly describe the problem to be solved in one or two sentences.
- I think you should mention something about lazy execution as it is a central aspect of stream processing frameworks such as Spark.

More detailed comments are in the attached report. Spelling errors are marked in yellow, other issues are marked in purple.

4 Questions for the Opposition

4.1 Question 1

"When talking of workers, be noticed that, independently from their total number in each Spark cluster, the amount of memory assigned to each of them is 1 GB and the maximum number of assigned cores is 10." - Section 6.4

Does this mean that each worker gets the same CPU/MEM? If so, why does the distributed solution outperform the centralized? They should in this case use the same amount of MEM/CPU and the centralized does not suffer from network costs.

4.2 Question 2

Why does the decentralized configuration get twice as many workers?

4.3 Question 3

Did the experiment simulate a geographically distributed system? E.g, what was the request rate and latency?

4.4 Question 4

Are there any downsides to decentralization. And are there downsides to increasing the levels of aggregation?

4.5 Question 5

What is the time complexity/memory complexity (approximately) of the centralized/distributed/decentralized solutions? How do their performance improve when scaling up/scaling out?

4.6 Question 6

Are there any other issues with geographically distributed systems which need to be addressed in future work? For example:

- Do you need to make any further considerations for reliability, or is it all handled by Kafka/Zookeeper/Spark?
- How do you handle event and processing time between time zones?

4.7 Question 7

In reality, clusters may consist of more than a thousand geographically distributed computers.

- What aspects do you need to consider when determining which computers to assign to a cluster?
- Do you know any clustering algorithm which could be used for assigning computers to clusters?
- What should you consider when assigning the level of aggregation for a cluster?