A Generalizable Framework for Automated Cloud Configuration Selection

Supervisors: Adam Barker & Yuhui Lin

Jack Briggs - 140011358

MSc Data-Intensive Analysis

2019-06-06

Abstract

Outline of the project using at most 250 words

Declaration

I declare that the material submitted for assessment is my own work except where credit is explicitly given to others by citation or acknowledgement. This work was performed during the current academic year except where otherwise stated. The main text of this project report is NN,NNN* words long, including project specification and plan. In submitting this project report to the University of St Andrews, I give permission for it to be made available for use in accordance with the regulations of the University Library. I also give permission for the title and abstract to be published and for copies of the report to be made and supplied at cost to any bona fide library or research worker, and to be made available on the World Wide Web. I retain the copyright in this work.

Contents

List of Figures

1 Introduction

Describe the problem you set out to solve and the extent of your success in solving it. You should include the aims and objectives of the project in order of importance and try to outline key aspects of your project for the reader to look for in the rest of your report.

1.1 Background

1.2 Aims and Objectives

1.3 Achievements

1.4 Dissertation Overview

2 Context Summary

Surveying the context, the background literature and any recent work with similar aims. The context survey describes the work already done in this area, either as described in textbooks, research papers, or in publicly available software. You may also describe potentially useful tools and technologies here but do not go into project-specific decisions.

3 Requirements Specification

Capturing the properties the software solution must have in the form of requirements specification. You may wish to specify different types of requirements and given them priorities if applicable.

3.1 Functional Requirements

3.2 Non-functional Requirements

3.3 Use-case

4 Software Engineering Process

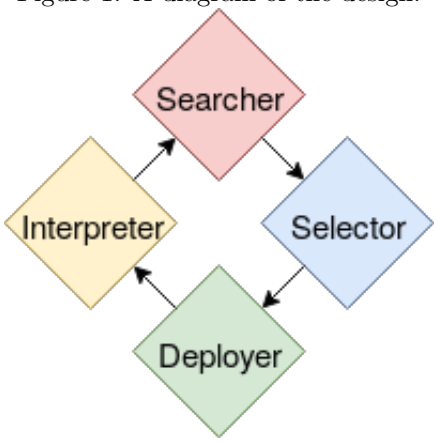
The development approach taken and justification for its adoption.

5 Design

Indicating the structure of the system, with particular focus on main ideas of the design, unusual design features, etc.

5.1 That cherrypick quote about starting specific

Figure 1: A diagram of the design.



6 Implementation

How the implementation was done and tested, with particular focus on important / novel algorithms and/or data structures, unusual implementation decisions, novel user interface features, etc.

6.1 Searcher

For the sake of generalizability, the available implementation of spearmint was found to be outdated and incompatible with the latest versions of various python modules planned to be used in later steps. Because of this, spearmint was first updated to be compatible with Python 3 and newer versions of its dependencies such as Google Protocol Buffers. This implementation of spearmint has been made available.[1] Any searcher

[1] https://github.com/briggsby/spearmint3

6.1.1 Bayesian Optimization - Spearmint

6.1.2 Gradient Descent

6.1.3 Exhaustive search

6.2 Selector

6.2.1 Exact Match

6.2.2 Closest Match

6.3 Deployer

6.3.1 VM Provisioner

6.3.2 Docker Deployer

vBench

Cloudsuite3

Sysbench

6.3.3 Ping server

6.4 Interpreter

7 Evaluation

You should evaluate your own work with respect to your original objectives. You should also critically evaluate your work with respect to related work done by others. You should compare and contrast the project to similar work in the public domain, for example as written about in published papers, or as distributed in software available to you.

With the best option clearly being c5.large, running spearmint with both multiple concurrent jobs (3 simultaneous jobs)and multiple providers (both AWS and GCP), the optimal instance was selection 55% of the time, with m5.large chosen 35% of time. Reducing the search space significantly by reducing the number of providers to only one (GCP) causes the optimal instance to be selected instead 75% of the time. This makes clear that for this experiment a less relaxed stopping condition may be beneficial. To

Figure 2: Distribution of vBench scores

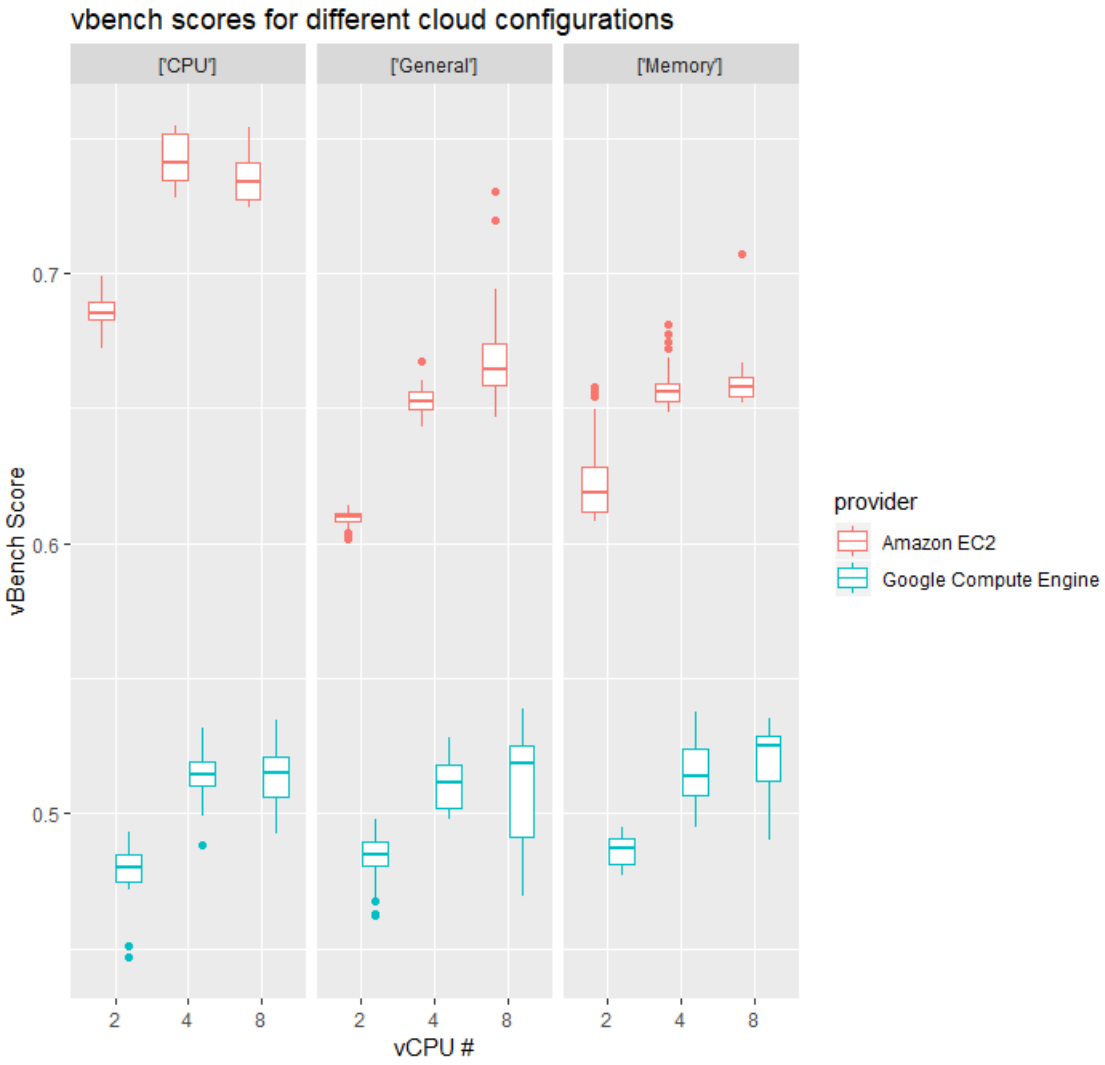
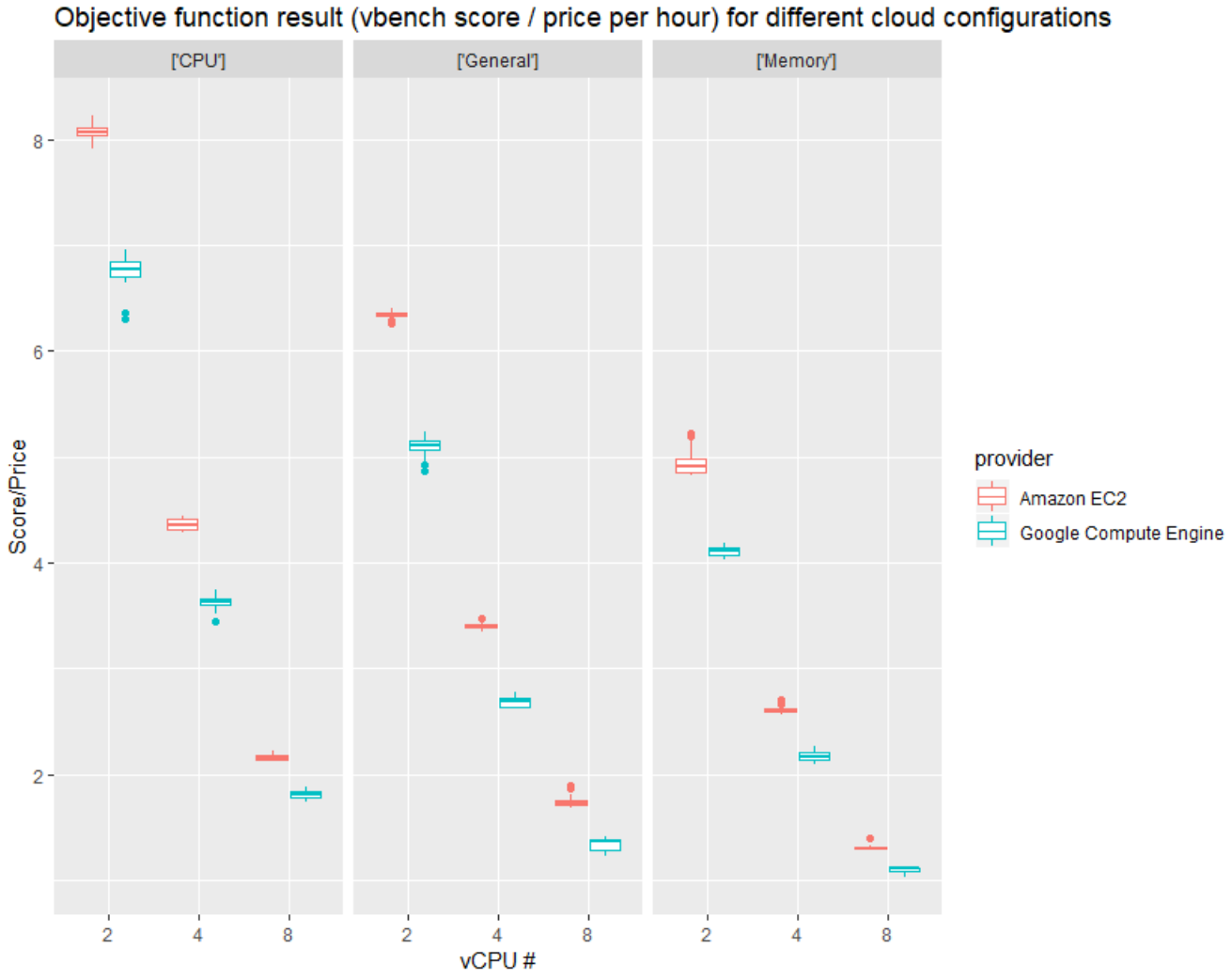


Figure 3: Distribution of objective function values



attain the effectiveness that the CherryPick paper claims, we must not allow concurrent jobs. Preventing multiple jobs running at once instead ensures the optimal instance type is selected 80% of the time.

7.1 Ping Test

In the ping test evaluation of 10 spearmint experiments using the same stopping condition as above, one of two optimal configurations (AWS m5.large or c5.large instance types), with overlapping distributions for the objective value, was successfully chosen in all runs.

8 Critical discussion

You should evaluate your own work with respect to your original objectives. You should also critically evaluate your work with respect to related work done by others. You should compare and contrast the project to similar work in the public domain, for example as written about in published papers, or as distributed in software available to you.

As mentioned, the evaluation above likely does not correspond to comparisons with which to base real deployment decisions on. In reality, a small increase in transcoding speed may lead to a far greater increase in customer uptake, rather than the effectively 1:1 ratio between price and transcoding speed assumed in the experiment. However, the evaluation shows that the methodology works very well with a given objective score measure, and it would be trivial for a new objective function to be implemented with a different relationship between the score, price, and 'value' of a given configuration.

8.1 Future extensions

9 Conclusions

You should summarise your project, emphasising your key achievements and significant drawbacks to your work, and discuss future directions your work could be taken in.

[1]

References

[1] S. Agarwal, S. Kandula, N. Bruno, M.-C. Wu, I. Stoica, and J. Zhou, "Re-optimizing data-parallel computing," in *NSDI'12 Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation*, pp. 281–294, USENIX, 2012.

Appendices

Testing Summary

User Manual