Outlier Detection in Virtual Machine Resource Data

A dissertation submitted in partial fulfilment of

the requirements for the degree of

BACHELOR OF ENGINEERING in Computer Science

in

The Queen's University of Belfast

by

Liam Reid

2022 May 3rd

SCHOOL OF ELECTRONICS, ELECTRICAL ENGINEERING and COMPUTER SCIENCE

CSC3002 – COMPUTER SCIENCE PROJECT

Dissertation Cover Sheet

A signed and completed cover sheet must accompany the submission of the Software Engineering dissertation submitted for assessment.

Work submitted without a cover sheet will **NOT** be marked.

|  |  |  |  |
| --- | --- | --- | --- |
| Student Name: | Liam Reid | Student Number: | 40231992 |
| Project Title: | Outlier Detection in Virtual Machine Resource Data. | | |
| Supervisor: | Dr Son. T. Mai | | |

Declaration of Academic Integrity

Before submitting your dissertation please check that the submission:

1. Has a full bibliography attached laid out according to the guidelines specified in the Student Project Handbook
2. Contains full acknowledgement of all secondary sources used (paper-based and electronic)
3. Does not exceed the specified page limit
4. Is clearly presented and proof-read
5. Is submitted on, or before, the specified or agreed due date. Late submissions will only be accepted in exceptional circumstances or where a deferment has been granted in advance.

**By submitting your dissertation you declare that you have completed the tutorial on plagiarism at** [**http://www.qub.ac.uk/cite2write/introduction5.html**](http://www.qub.ac.uk/cite2write/introduction5.html) **and are aware that it is an academic offence to plagiarise. You declare that the submission is your own original work. No part of it has been submitted for any other assignment and you have acknowledged all written and electronic sources used.**

1. If selected as an exemplar, I agree to allow my dissertation to be used as a sample for future students. (Please delete this if you do not agree.)

|  |  |  |  |
| --- | --- | --- | --- |
| *Student’s signature* | Liam Reid | *Date of submission* | 3/5/22 |

Acknowledgements

To those who have helped the author during the project and the preparation of the dissertation and to anybody who has given financial support.

Abstract

A summary (100 words) which provides an outline of the subject matter and the results, findings and/or conclusions of the dissertation.

Contents

A complete list of chapters, sections, appendices etc. with page numbers.

Main Text (see below)

The main body of the dissertation as described below organised as a sequence of chapters each normally containing several sections. The main text should not normally exceed 45 pages (it may be less).

References

A list of references to documents (books, papers, web pages etc.) which are referred to in the main body of the text. Use the IEEE citation style as detailed here <https://www.ieee.org/documents/ieeecitationref.pdf>. There is some guidance on referencing at <http://www.qub.ac.uk/cite2write/home.html>.

**The first citation should be the URL to the software code repository which should contain the code and any other resource required to run the software.**

Appendices

These should include as appropriate:

(a) A User manual giving details on how to use the software, including details of input data, output formats and error messages.

(b) Test results, if appropriate.

(c) Other information which is not convenient or appropriate to include in the main body of the dissertation.

The Main Text

**The Examiners will be looking for quality rather than quantity in your dissertation.** You should try to keep the main text of your dissertation as concise as possible. Spelling should be correct, sentences grammatical, and formulae, figures and tables accurate. All figures, tables and appendices should be given numbers and headings. Your writing should be precise, concise and fluent. Avoid the first person (i.e. say what was done, rather than that you did it). Some dissertations may be organised differently from that headings given below. In particular, some parts might have more emphasis than others for some projects Further advice will be available from your supervisor. Below is a sample of what might be expected.

1. Introduction and Problem Area
   1. Introduction

Cloud computing, although not fundamentally new, has become one of the fastest growing technologies in the 21st century [1]. Applications and services deployed on cloud-based platforms are getting increasingly complex with the escalation in popularity. Consequently, cloud engineers are constantly working to make their platforms robust and failsafe, but problems with cloud services are still prevalent as we move into 2022 [2].

On October 4th, 2021, Facebook, WhatsApp, and Instagram suffered an outage due to a problem with their shared cloud infrastructure. The effects were felt by its billions of users globally, claiming they felt “discomfort and displeasure” during the 6-hour period [2][3]. Facebook shares fell 5.4% and loses in revenue were estimated to be around $99.75 million [4].

Maintaining cloud systems within an organisation is the responsibility of the Cloud Systems Administrator. A Cloud Systems Administrator deploys services, monitors and analyses cloud resource performance and resolves any issues reported. This role has become so important in recent years that organisations dedicate entire teams of Cloud System Administrators to manage their cloud infrastructure [5].

With the increased complexity of this role, a software solution has been developed in order to assist Cloud Systems Administrators in the monitoring of cloud services. This solution provides a platform to detect discrepancies in cloud resource data using newly implemented outlier detection techniques. The detection can run in real time to monitor CPU usage and identify outliers. The solution is also a platform for performing experiments on new outlier detection techniques with a variety of datasets. The proposed solution will aid a Cloud Systems Administrator in performing their role and automate the process of detecting discrepancies in cloud resource usage.

* 1. Cloud Computing in 2022

Cloud computing has been a popular technology used for many years, and the demand is continuing to grow. Over 90% of organisations use some type of cloud service for a variety of different applications [6]. “Data backup, disaster recovery, email, virtual desktops, software development and testing, big data analytics, and customer-facing web applications [7]” are among a few listed by AWS (Amazon Web Services).

In recent years, millions of people around the world have been forced to work from home because of the coronavirus crises [8]. Remote working was forced upon many unprepared organisations and their workforces. Consequently, the use of cloud-based virtualisation technology was utilised to provide staff with the resources necessary to carry out their roles [9]. BYOD (Bring Your Own Device), paired with a VDI (Virtual Desktop Interface), provided employees with an interface to their organisation’s network, applications and resources [10], [11]. Thus, with the power of Cloud technology, office workers around the world quickly adapted to our ever-changing way of life.

* 1. Risks / Mitigation of Hosting Cloud Resources

Resource exhaustion is a major risk to a VM host server, where software using a server will exhaust its resources and effect the availability of other Virtual Machine’s sharing the same resources [12]. Resource exhaustion may occur for several reasons, for example, an infinite loop that prevents a process from terminating or a cyber-attack that results in a server’s resources being misused. If availability is impacted, a systems software may fail and introduce vulnerabilities to the system [13]. An effective Outlier Detection algorithm paired with a monitoring dashboard can reduce the risk to availability by notifying the user of a Virtual Machine on a server exhausting more resources than it normally needs.

Physical resources for Virtual Machines are managed by software called the hypervisor. The hypervisor is responsible for provisioning new VMs and distributing the resources where necessary. Hypervisor security can be breached by ‘various malicious attacks.’ “Session hijacking, man-in-the middle attack, flooding attack (and) Malware-Injection attack(s)” can allow malicious intruders unauthorized access to a server’s resources. In session hijacking, a hacker will steal a legitimate user’s session ID when it is generated after login. Once the session is hijacked, the hacker can copy the VM and gain access to all the data. A hacker can use a man-in-the-middle attack to intercept data and gain access to a VM. A hacker can perform a Malware Injection attack by inserting malicious code into an application. This makes the resources, applications, and data on the VM vulnerable

[14]. These events are liable to happen often and when they do, they can go undetected. An Outlier Detector would mitigate the damage done to a system by notifying the user of any suspicious/anomalous behaviours.

Misuse or unauthorized access of a VM management console can be a serious risk. A hacker could gain access to a VM management console using techniques like phishing, malware attacks, brute force attacks or by learning login credentials from a system or website breach. Not just hackers, but an insider from an organisation with authorised access to VM management tools can also be a risk. With access to a VM management console, a person could cause a lot of damage by deprovisioning VMs running important processes. They could also provision additional VMs that are not needed, this would result in a huge cost for the organisation as resources are scaled to meet demand and if resources restrictions are set, availability of resources is decreased for other VMs running important processes [15]. These changes could easily be detected by a VM Outlier Detector dashboard and the risk and cost to the organisation can be reduced.

VM sprawl is the process where a VM is duplicated then forgotten about. These unmanaged VMs can operate on a network for weeks or even months without being detected. Without proper monitoring, these VMs can miss important security patches. As a result, there is an increased risk to security. There are methods to prevent VM sprawl, processes to govern VM lifecycle management using automated scripts can reduce the risk of VM sprawl, but this comes at a cost [12]. A dashboard monitoring Virtual Machine data in real time could quickly identify these forgotten VMs and notify the user.

[1] P.K. Paul, A. Kumar. and M. Ghosh. "Cloud Computing: the 21st Century Friend for Virtualization." *About Mewar University* (2012). Available: [Paper Title (use style: paper title) (researchgate.net)](https://www.researchgate.net/profile/Ajay-Kumar-80/publication/325083266_Cloud_Computing_the_21st_century_friend_for_virtualization/links/60eeb60f16f9f3130080302c/Cloud-Computing-the-21st-century-friend-for-virtualization.pdf)

[2] Cast AI. (2022, Jan. 11) “The cloud in 2021: 21 game-changing outages, security issues, and highlights” [Online]. Available: <https://cast.ai/blog/the-cloud-in-2021-21-game-changing-outages-security-issues-and-highlights/>

[3] N.M. Shousha. and L.R. Abdelgawad. (2021) "DOWN IN MINUTES, OUT FOR SIX HOURS: A BRIEF REPORT ON FEELINGS DURING THE OUTAGE OF WHATSAPP, INSTAGRAM, AND FACEBOOK." *British Journal of Psychology Research Vol.9, No.2, pp. 38-44.* Available: [DOWN IN MINUTES, OUT FOR SIX HOURS](https://www.researchgate.net/profile/Nayera-Shousha/publication/356193254_Down_in_Minutes_out_for_six_Hours_A_Brief_report_on_feelings_during_the_outage_of_WhatsApp_Instagram_and_Facebook/links/619203fe61f09877209a3736/Down-in-Minutes-out-for-six-Hours-A-Brief-report-on-feelings-during-the-outage-of-WhatsApp-Instagram-and-Facebook.pdf)

[4] The Guardian. (2021, Oct. 10) Impact of WhatsApp, Facebook and Instagram outage on businesses [Online]. Available: <https://guardian.ng/technology/social-media/impact-of-whatsapp-facebook-and-instagram-outage-on-businesses/> (Accessed 28th March 2022)

[5] J. Frank. (2019, Jul. 29) Your Next Move: Cloud Systems Administrator[Online]. Available: <https://www.comptia.org/blog/your-next-move-cloud-systems-administrator>

[6] 451 Research. (2017) Voice of the Enterprise: Cloud Transformation survey [Online] Available: <https://451research.com/images/Marketing/press_releases/Pre_Re-Invent_2018_press_release_final_11_22.pdf>

[7] AWS. (n.d.) Who is using cloud computing*?*[Online]. Available: <https://aws.amazon.com/what-is-cloud-computing/>

[8] S. Soares, F. Bonnet, J. Berg. (2021, Apr. 25) Working from home during the COVID-19 pandemic: Updating global estimates using household survey data[Online]. Available: <https://voxeu.org/article/working-home-during-covid-19-pandemic-updated-estimates>

[9] D. Anderson, C. Kelliher (2020), "Enforced remote working and the work-life interface during lockdown", *Gender in Management, Vol. 35 No. 7/8, pp. 677-683*. Available: <https://doi.org/10.1108/GM-07-2020-0224>

[10] P.J. Cho, et al. "Demographic Imbalances Resulting from Bring-Your-Own-Device Study Design." *JMIR mHealth and uHealth* (2021). Available: <https://s3.ca-central-1.amazonaws.com/assets.jmir.org/assets/preprints/preprint-29510-accepted.pdf>

[11] T. Mangan (2020, Mar. 30) The Tech Supporting Remote Workers[Online]. Available: <https://www.nutanix.com/theforecastbynutanix/technology/modern-remote-workforce-is-powered-by-virtual-desktop-technology>

[12] A. Chaudhuri. (2015). "Best Practices for Mitigating Risks in Virtualized Environments" - *Cloud Security Alliance, USA.* Available: <https://downloads.cloudsecurityalliance.org/whitepapers/Best_Practices_for%20_Mitigating_Risks_Virtual_Environments_April2015_4-1-15_GLM5.pdf>

[13] EBA/GL. (2017) Guidelines on ICT Risk Assessment under the Supervisory Review and Evaluation process *(SREP) (Page 11). Final Report on ICT Risk Assessment Guidelines.* [Online] Available: [EBA BS 2017 131 (Final Guidelines on ICT Risk Assessment under SREP) (europa.eu)](https://www.eba.europa.eu/sites/default/documents/files/documents/10180/1841624/ef88884a-2f04-48a1-8208-3b8c85b2f69a/Final%20Guidelines%20on%20ICT%20Risk%20Assessment%20under%20SREP%20%28EBA-GL-2017-05%29.pdf?retry=1) (Accessed: 4 January 2022).

[14] N. Arya., M Gidwani., S.K. Gupta. (2013) “Hypervisor Security - A Major Concern” *International Journal of Information and Computation Technology. Volume 3, Number 6, pp. 533-538.* Available: <https://www.ripublication.com/irph/ijict_spl/08_ijictv3n6spl.pdf>

[15] T. Morrow. (2018) 12 Risks, Threats, & Vulnerabilities in Moving to the Cloud. *Carnegie Mellon University's Software Engineering Institute Blog* [Online]. Available: [http://insights.sei.cmu.edu/blog/12-risks-threats-vulnerabilities-in-moving-to-the-cloud/](http://insights.sei.cmu.edu/blog/12-risks-threats-vulnerabilities-in-moving-to-the-cloud/%20) (Accessed March 28, 2022)

Background material should be given which introduces the problem area, its context and background. You should identify the particular problem under consideration along with information about the problem area that enables the reader to understand the problem scope and nature. If your project involves a particular domain, algorithm, method theory etc., you may describe it in the introduction (alternatively or additionally, it may be described later, if appropriate). For best marks the student should show that they have systematically researched and fully analysed the problem, synthesising the relevant information. The very best dissertations will be of a standard suitable for part of a peer reviewed or professional publication.

1. System Requirements and Specification

You should provide a precise description of the system developed. Note that this is likely to be different from those that you started with, since you now have considerably more knowledge and understanding. The final dissertation should therefore contain an updated set of requirements matching the final system delivered. You can list these as a *requirements definition* from the domain perspective but you should also derive a *specification* for the software. The software requirements specification establishes the basis for what the software product is to do (and is not expected to do). You should list any assumptions made about the problem and any system constraints. Overall your requirements, functional AND non-functional should be correct, complete, consistent, clear, feasible and objectively verifiable. Content depends on your project but could include:

* A complete set of function definitions (as use cases if preferred), as far as possible written so as to be testable
* Measurable and testable non-functional requirements
* Description of interfaces required such as with other software or systems
* Any specific user interface requirement
* User characteristics

The target to aim for here is to describe a solution that satisfactorily solves the problem. Ideally your solution will be convincing and creative. Your requirements could be the basis for a contract or handing to external developers to complete. The best dissertations will show outstanding work, approaching that of the best professionals.

* 1. Example sub heading

Use sub headings as appropriate

Subsub heading

For a sub sub heading just make it bold without numbering.

1. Design

This section should describe the design of your proposed system. Normally this several parts, depending on your project:

1. Architectural Description of the system – textual and/or diagrammatic. This could be a simple diagram showing the components and how they relate or it could describe the choice of architectural style or pattern used.
2. User Interface Design (if applicable). Show sketches of the design or screenshots with explanations of choices made, if necessary.
3. Software System Design. The role of each component and the interfaces between components should be described. There should be a clear correlation between your design and your specification.
4. Where applicable give a critical discussion of key design decisions/styles/patterns used; data model; UI design, external Interfaces, other important issues e.g. concurrency, event handling, data persistence, error and exception handling, fault tolerance, security, distribution of components.

The design should be linked to requirements and, where applicable give a critical discussion of key design decisions/styles/patterns used. There might be a data model, a UI design, details of external interfaces, and of other important issues e.g. concurrency, event handling, error and exception handling, security, data persistence. No particular notation or tool is mandated. A satisfactory design will show a grasp of the main design issues. For top marks aim for outstanding design documentation approaching that of the best professionals. Prove that you have a very strong grasp of the design issues and aim for documentation that could be passed on to a developer without the need for further explanation

1. Implementation

You should describe any languages, packages, and libraries etc. that are used in the development of your system. There is no need to describe your code in detail. You may highlight data types and implementation techniques that are of special interest. If appropriate, you may provide:

1. Choice of implementation language(s)/ development environment(s)
2. Use of software libraries;
3. Key implementation decisions
4. A description of how some important functions and algorithms were implemented.
5. A description of how each component is implemented.

Program code can be accessed by the assessors via the git repository **so there is no need to include code listings**. It is recommended that you comment code appropriately (not excessively). Programs should be written in a clear style with good program structure and well-defined data structures. The program code should reflect its design and show an understanding of relevant implementation issues.

1. Testing

This section will be judged in tandem with other evidence including evidence of unit tests and/or test documentation on the Repo. There should be a discussion of Test Approach e.g. unit testing, system testing, regression testing etc; Test cases should be described and justified; Include Testing tools used and provide evidence that testing coverage was complete. Provide proof that testing was completed, either showing sample test history and/or describing automated tests.

1. System Evaluation and Experimental Results

Provide a summary evaluation of the success of the project with respect to criteria identified in the introduction. Different projects will have a different emphasis. In all cases you are expected to provide empirical results and to draw conclusions from those results. You may use your software to generate experimental results. Be sure to describe the methodology of your evaluation or experimentation. An experiment is typically described in terms of its goals, the hypotheses being tested, the subject of the experiment, what is being measured and what is controlled, the results obtained and the analysis and interpretation of those results. A discussion of the significance of your experimental results may be appropriate or why the new system you have developed improves on what was already there. Do your results agree with other previous work or ideas? How does your system compare with similar ones?

Alternatively (or additionally), you can assess the product in terms of how it compares with other similar products and/or in terms of user feedback (e.g. via a survey or interviews) or some measurable quality aspect such performance efficiency or reliability.

Draw conclusions on the *process* used in the project as well. What went well? What did not go well? What are the strengths of your solution or conclusions? What are the weaknesses? Suggestions for further work should also be discussed. You can be critical and draw a negative conclusion. Not all projects will be successful. A well-explained failure is as an acceptable an outcome as a spectacular success. Assessors are looking for excellence in a critical appraisal of the work and a convincing argument for the significance of contribution in the context of wider work. This section should be objective, fair and comprehensive.

In all cases, societal implications and commercial and economic aspects should be evaluated. Has your project an outcome that potentially could improve some community or group of people? Perhaps your project can impact on the lives of others for example In education, employment, health, public policy or services, security, the environment, general wellbeing etc. There may be commercial opportunities arising from your product or findings. Describe these and include how the project could eventually brought to deployment and to deliver value. Discuss the feasibility of doing that. It may be that your project could make some process more efficient. Try to quantify the savings or improvements, generally or in one or more scenarios. You should be realistic though and include the risks and any negative impacts of your work and the potential impact as well.

Your supervisor can guide you on what is appropriate, but typically the very best projects have shown results derived using scientific method, that could be publishable with little or no work or show an exemplary empirically based evaluation of a software product. Those projects will also fairly and honestly assess the potential impact of the work socially or economically.

Any publication of results of the student's work is left to the discretion of the supervisor, but you can expect appropriate credit to be given to your work.

1. Appendices

Appendices will not be marked but may be referred to by the assessor to aid their understanding. They are useful if there is something that helps in understanding earlier parts of the dissertation, but if included inline might break the flow or readability of the document. For example, there may be large tables of data, design documents, evidence of testing etc etc.