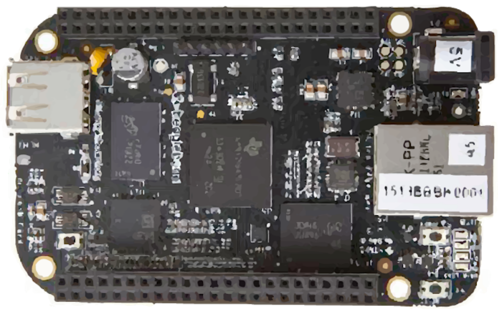
CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG



BB-EMU
A Product Virtualizing System Verification of Embedded Hardware

Bachelor of Science Thesis in Software Engineering and Management

SRI HARI THUCCANI
MARTIN L. KABZIMALSKI

Department of Computer Science and Engineering
UNIVERSITY OF GOTHENBURG
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2018

{A comparison between a virtualized embedded board and an actual embedded board to see how much time is required to test software on the different test environments.}

{SRI HARI THUCCANI, MARTIN L. KABZIMALSKI}
[SRI HARI THUCCANI, MARTIN L. KABZIMALSKI]

Supervisor: {REGINA HEBIG}
Examiner: {MICHAL PALKA}

University of Gothenburg
Chalmers University of Technology
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Cover photo: An image displaying an embedded hardware and the output of test code. The embedded hardware is a Beaglebone Black. The output test code is a sample from the tests conducted in this thesis. Both the embedded hardware and the test codes image quality has been reduced.

Table of Content

Thesis Report

BB-EMU – A Product Virtualizing System Verification of Embedded Hardware

Sri Hari Thuccani
Student of Software Engineering & Management
Gothenburg University
Gothenburg, Sweden
hari.thuccani@gmail.com

Martin Kabzimalski
Student of Software Engineering & Management
Gothenburg University
Gothenburg, Sweden
guskabzma@student.gu.se

***Abstract*— The purpose of this study was to examine if there is a time difference when testing software on a physical embedded board and a virtualized version of the same board. This study conducted research on the topic at hand, organized experiments with the embedded boards and evaluated the results. It was discovered that the virtualized version was performing faster than the physical board. It was also concluded that testing on a virtualized embedded board gives the tester, the freedom to experiment with the configuration of the board and the required time for testing (from setting up the test environment to executing the test suite) is shorter than the time required on the physical board.**

Index terms - artifact, QEMU, BB-EMU, BBB, virtualization

I. Introduction

Testing software before release is a step that is required for assuring that the developed product is capable of achieving the expected output given by the stakeholder and developer [1].
On the other hand, virtualizing hardware for the purpose of testing is a widespread practice, which is used by companies that focus on developing robust hardware products [3].

The project, that will be presented in this work, is part of Diadrom - an expert consultancy company with a focus on diagnostics of products with embedded software. The company was founded in 1999 and has a majority of the customer base in the automotive sector [2]. Diadrom works with wide variety of embedded boards and for this project the company requires us to work with any emulated embedded board we want, as long as it runs on Debian (operating system).

The thesis will be focusing on testing in software development for embedded systems and the industrial environment that the paper will be covering is software development. We will be using test suites developed in Python programming language, papers that focus mainly on subject of virtualization and the virtualization of an x86 chip; QEMU, which is an emulation software for embedded systems with virtualization capabilities; and Virtual Manager which is a manager for virtual machines. This is explained in detail in the coming sections.

A. Motivation

Diadrom states that a considerate amount of time is required to test software on a physical hardware and wanted someone to conduct a study on virtualizing hardware. Diadrom believes that virtualizing hardware, for testing the communication between software and hardware, might be able to drastically reduce the required time for fully setting up the test environment.

B. Problem Domain

Examining and puting embedded systems to the test in the real world is time consuming, for that reason the developers has to physically set up the hardware for each single test, so that the generated software can be tested on a desired part of the hardware. The focus of this paper will be on examining if virtualization of an embedded board can reduce the time spent on conducting the verification procedure of the developed product.

Diadrom is currently working with various embedded boards and requires research to be conducted, in order to discover if time will be reduced while testing software on a hardware by virtualizing that hardware. We are aiming to answer the following Research Question (RQ) so that we can discover if time is reduced for verifying an embedded product.

Can virtualizing the test objects reduce time required for executing a test process, compared to the time required on an embedded board?

II. Background and Related Literature

This section contains the information that is used as a foundation of knowledge for this research paper.

A. Background

To get a better understanding about the term virtualization, first the definitions of emulation and simulation have to be examined. To acquire the actual meaning of the word simulation we examined the dictionary and it defined it as a technique that represents the real world by a computer program [7]. For example, how an embedded board will endure if it was exposed to different weather conditions or how stacks of data will flow through the processor of that embedded board.

Emulation & Virtualization

Emulation on the other hand, is defined as recreating (with software) a product that could replace a real-life product, ”imitating a certain computer platform or a program on another platform or program” [8]. For example, creating a copy of an embedded board, with the help of emulation software, that has the exact same characteristics (CPU speed, RAM, USB ports, etc) and same behavior as the physical board.

Virtualization is what gives the foundation for the emulated models to function on a computer system, “it is the software technology which allows emulated models to run” [9]. When virtualizing an emulated model, the process is separated between a host and a guest. The computer system that has the emulation running is the host and the guest is the emulated model that is running on the host.

In the case of this thesis, the host is a laptop computer which is running on Ubuntu 16.06 OS and the guest is a virtualization of an embedded board with an x86 [10] chip, all done by QEMU . Also, a virtualization manager is used since there were obstacles (explained in the Encountered Obstacles section) with fully implementing QEMU, which is called Virtual Machine Manager or virt-manager for short.

Artifact

Ubuntu is an open source operating system that is a Linux distribution based on Debian architecture [11]. QEMU is a generic and open source machine emulator and virtualizer. It can emulate a wide variety of embedded boards that can run Operating Systems and can virtualize those boards with near native performance by executing the guest code directly on the host [12]. Virt-manager is a desktop user interface for managing virtual machines. It presents a summary view of running virtualization domains, their live performance and resource utilization statistics. It enables the creation of new virtualization domains, configuration and adjustment of a domain’s resource allocation and virtual hardware [13].

The above-mentioned software is used in combination with the Design science research [6] guidelines to create an artefact which will be tested for speed of completion.

To create the artefact we looked into using the available software emulators, which focus on emulating embedded hardware. Using QEMU we can create embedded hardware through emulating a board with a x86 chip, with capabilities of network and USB connectivity. Originally we were planning to use Emul8 as emulating software but we switched to QEMU due to the obstacles faced with Emul8. Following the design science iterative design model guideline we were able to make progressive iterations towards the final version of the artefact. Each iteration contained issues and knowledge that benefit the next iteration. The knowledge was acquired by testing and from outside sources like the employees of Diadrom or stackoverflow [6].

After the completion of constructing the artefact (virtualizing an x86 embedded board), we transferred the application developed by Diadrom to test on the virtualized hardware. The application is developed in C++ programming language, and its main purpose is to transfer small and large amounts of data to and from other devices.

To test the said application, a unittest that was developed on the programming language Python, is used to test the communications, however is mainly used to test the verification of data from the application.

Both the application and the testing software are capable of accessing ethernet and USB ports. Therefore the testing will be conducted through those two primary connections.

These connections are similarly handled with the physical hardware.

Our main purpose for using the test software is to record time taken of the application while it is performing it's full functionality. During this stage, following the evaluation methods prevents the experimentation and testing from deviating the research question.

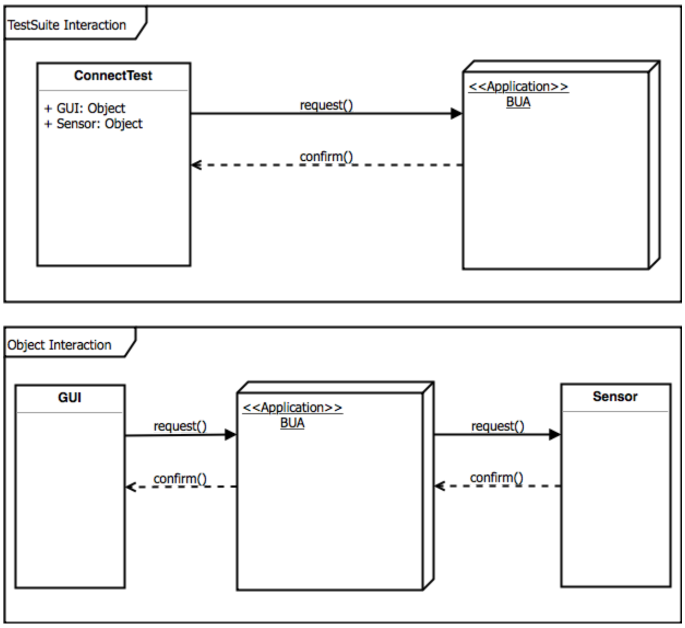


figure 3.

Test Suite

The test cases are specified to target all the functionalities of the application through network and USB communications. The tests creates a mock of a GUI and one or more sensors, and their purpose is to send/receive and manage messages from the app as shown below in *figure 3*. The messages are constructed and serialized by a library called Google Protobuf, which is a protocol buffer that serializes structured data to be sent to and fro objects.

The test cases themselves tests the connections to the application and verifies the correct ID or messages that is sent back and forth. They also test different procedures that the application would undergo during its use. These could range from, large amounts of data transferred from the sensors to the application then to the GUI, to collections of sensor info and GUI info.

The complexity of the test cases are dependant on the number of mock sensors that are used during the test case. As communication can occur not only between the app and the sensors but also between the sensors themselves or the GUI through the application. These kinds of scenarios raises the complexity. Another high complexity are test cases that are designed to create failure during a procedure.

These complexities doesn't only put a strain on the application but also on the emulation due to the number of functionalities that the application has to run on the emulation and the constant communication through network and USB.

The testing method starts from when the computer boots up. At that time we start recording the time up until the completion of all test cases. The test suite contain these test classes:

- ConnectTest - Tests the connectivity (network & USB) of the application with different mock objects, and it also tests failure cases.
- PowerTest - Verifies the different power states that the application can be applied to.
- RequestTest - Requests information from the application e.g. legal seal, application version etc.
- TimeSyncTest - Prompts the application to run a time sync procedure.
- TrigTest - Conducts a scenario of structured data sent from one mock object to another, through the application. It also tests failure cases.

These tests are made by Diadrom and their employee, Hari, during the thesis. They were constructed based upon specifications for the application, received from the company that Diadrom is consulting.

The data collected is used for statistical analysis, which is the science of data collection to present large amounts of data to find an underlying pattern or trends. Using that study we would be able to verify if the artefact is performing either on par or better than the physical hardware.

After the completion of the tests an interview was conducted to gather the opinions of the Diadrom employees who have had experience in working with similar systems. Their opinion is important because they can give additional feedback if virtualized system verification of embedded boards is faster to test software with.

Further details are discussed in the following sections of this study.

B. Related Literature

The following research papers were chosen to be a part of this study because we decided that we have enough information for our work. There is one paper that explains a similar study to ours that uses the same virtualizing software (we did not manage to find more papers like that one) and two other works that give extra information on the related topic. The extra information is knowledge that covers virtualization for testing software but the study is not conducted in the same manner as this thesis. Below you will find the selected papers.

The work of Simon Wallström and Adam Dalentoft [3] focuses on Emulation-based Software Development for Embedded Systems which covers the topic at hand that is being discussed in our paper. Simon Wallström and Adam Dalentoft aim to explore how hardware platform emulation affects software development in terms of platform transition, daily development and testing. They also applied QEMU (embedded emulation software) to their research which is a piece of software that we want to implement in our work as well. Although, this thesis will take a different approach by virtualizing a different board (x86 chipset), with the same emulation program and focusing more on the quality control of the end product (if testers require lesser time to test software when hardware is emulated and if the company spend no additional resources on new hardware).

Another relevant paper is the work of Erek Göktürk et al. [4] which defines emulation as: “recreating a scenario or an object with software, that will have the same characteristics as if it was in the real physical world”. Göktürk et al. also surveys how popular emulators have become and how they create a safe environment for reinvigorating experiments with software, for example: flight simulators, simulation of network connections, chemical reactions, aftermath simulation of a disaster and many more. The author of that paper also presents a comparison between an emulator and a test bed. Although, only the provided information regarding emulation of systems will be considered as relevant and will be used as a resource of knowledge for this thesis.

The work of Edwards et al. [5] is also heading the same direction as the topic of this thesis. The focus of that paper is towards development of embedded systems and the validation of those systems. These types of systems are integrated in almost everything that is surrounding our lives, hence the authors emphasize on the fact that unwanted casualties could happen if not enough testing is done on the software and errors appear in critical systems, for example, life-support systems, airplane cockpit controls, the ECU in a vehicle, and many more. That is why Edwards et al. has conducted this study, to give insight on the importance of being cautious when Designing an embedded system and the necessity of Validating every embedded system pre-release. In other words make sure that it is giving the desired output with no deviations so that the developed product will be safe to use.

III. Methodology

This section discusses the methods used during research, development, testing and evaluation. Furthermore it delves into the choices and changes that occurred during the span of the thesis.

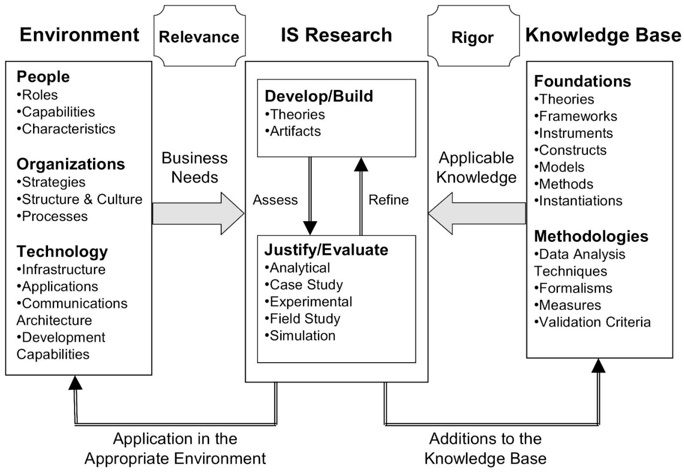


figure 1. [6]

A. Design Science

Design science research [6] are guidelines in the study of an artefact or artefacts within Information Technology (IT), also known as Information Systems (IS), for quality assurance and speed. Design science covers the bases of research, framework and evaluation. These bases are guidelines that help cover the fundamental procedures within the development lifecycle of an artefact.

- Research covers knowledge and understanding of the artefact. IS research guidelines, in design science, provide steps that aid in solving the problem, in our case the research question, by gathering related literature to design the artefact.

- Framework helps in developing and iterating the artefact per obstacle faced. An IS research framework shows the development cycle from research to development to evaluation. It also includes subsection for iterations. This is shown in *figure 1*.
- Evaluation provides methods in experimentation and testing of the artefact, to analysis the results for a conclusion. Design evaluation methods uses the methods from the knowledge base to evaluate the artefact.

These three bases are mentioned throughout the rest of the thesis [6].

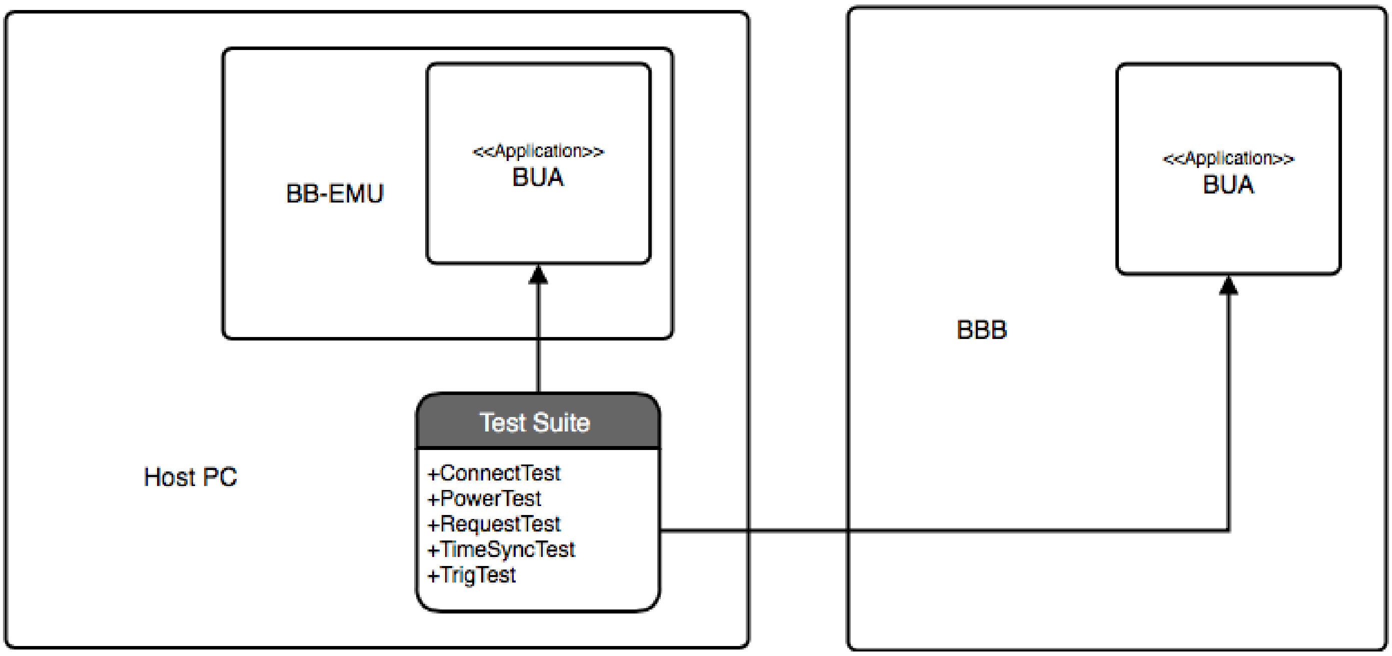


figure 2.

B. *Criteria*

By conducting the research we look into the criteria of the product. By researching the criteria we are determining the importance of certain functions that the artefact requires during the evaluation. The criteria is determined from the first presentation by the CEO of Diadrom.

The criteria for the emulators is dependant upon the basic functionalities of the physical system. E.g. the components required for communication, networking, USB and the capability to run an operating system. The emulations of the specific hardware components e.g. chipset, are not part of the criteria due to the research question focusing on the results of an application running on the emulation with a comparison of the same application running on a physical system.

C. *Iterations*

Iteration is the repetition of a process, in our case the process is the research and change of the artefact.

During the development of the artefact (IS research framework), 6 iterations were made to the artefact some major and some minor changes were made to configuration of the artefact. They are listed as following:

1) Change from Emul8 to QEMU. The time it took for the change was 3 days.
2) The change from ARM A9 to ARM Versatilepb chipset. Lots of discussion were had with one of the employees of Diadrom regarding how tackle the issues we were having with ARM A9. His suggestions was to use an emulation/artefact that has all the functionalities that we require and to not look into the specific chipset. This change occurred after one week.
3) The switch from ARM Versatilepb to x86 chipset. Similarly to the discussion with one of the employees in the previous iteration, we discussed with the CEO of the company, who presented us with this thesis, about the importance of the

artefact and the lack of importance about the chipset. This change took place under one week.

4) Lastly the shift from Debian's default ssh connection to Virt-Manager. The change follows from the discussion from point 3 regarding the importance of achieving a working artefact and not about how it is achieved. This change took place under 3 weeks.
5) After significant trials on the USB connection, between the host and virt manager, it was determined that it would not function in time, for the evaluation period. Therefore the USB connection was removed from the testing phase for both the emulated hardware and physical hardware. This took 4 days to change.
6) The last change was the update from the outdated application version. This took two days for the change to occur.

Further explanations and reasoning of these 6 interactions are detailed below, in *Encountered Obstacles*.

D. Test & Evaluation

The method for the test suite is conducted by running the tests 20 times on the artifact and the physical system. Evaluating the tests, we look upon the time and the correctness of the run. The correctness is feedback of the tests, which also correlates to the time taken. The more failures/errors the more the time taken gets reduced.

E. Interview

For the interview part of our study we collected the opinions of 4 interviewees. We used the interview questions that are mentioned below and everything was recorded through a microphone on a laptop.

Interviewee	Profession	Interview Duration	Diadrom Experience
1	CEO	15:39	19 years
2	Software Developer	11:03	1.5 years
3	Project Leader	11:58	1 year
4	Embedded Developer	13:39	18 years

In the table, interviewee 1 & 2 belong to the same project (the application used in BB-EMU and BBB) and interviewee 4 has experiences within QEMU and embedded systems.

These interviews were conducted on the request that they are not shared outside of Diadrom and are used in consensus for our feedback of the results.

Interview Questions

The interview questions would be conducted after the analysis of the data. The analysis is presented to the interviewees, who are employees of Diadrom that have worked on the application or assisted as a resource for QEMU. Interviewing these set of employees is necessary to determine their opinion on the performance of the artefact. Therefore they would be conducted as semi-structured interviews. Semi-structured interviews are qualitative interviews that has predetermined questions but are open ended to prompt discussions with the interviewee. Therefore the questions that were made for the interview were tailored to prompt discussion and further their opinions to evaluate our artefact and evaluations.

Have you worked with emulated hardware?
Asking this question shows their understanding of the subject.

What challenges have you encountered while working with emulated hardware?
This question is a follow up from the previous question and is only asked if they have worked with emulated hardware. It is to made to prompt further discussion about emulated hardware and evaluate their challenges to ours.

Have you worked with non-emulated hardware?
Similarly to the first question, it confirms their knowledge within the subject.

What challenges have you encountered while working with non-emulated hardware?
Similarly to the second question, it is to prompt further discussions and to evaluate their challenges to ours.

Was the presented time on the emulated version of the hardware to your satisfaction (fast enough)?
The questions was made as an opinion based metric question. The opinion of the interviewees rates the results of the evaluation.

Are the presented results trustworthy?
This question is to check if the method of evaluation is the correct method for the research question.

Would you prefer to use emulation when developing software for embedded boards?
This question shows how the demo presentation of the emulation and the need of the product is for development.

Do you think there is a better approach to test software for embedded boards?
This question is an open question to allow for discussion regarding other possible solutions for testing software for embedded boards.

F. Statistical Analysis

Statistical analysis is the science of data collection, revealing patterns and trends [14]. It is to summarise the data by calculating the collection.

The statistical analysis that we used for compiling the results is known as Descriptive statistics. We also chose to conduct Wilcoxon tests to analyse our data.

Descriptive Statistics

Descriptive statistics the collection of data that is summarised in qualitative description. It is often distinguished from inferential statistics. *'With descriptive statistics you are simply describing what is or what the data shows. With inferential statistics, you are trying to reach conclusions that extend beyond the immediate data alone.'* [15]

Basically the statement describes that descriptive statistics only focus is on presenting the data as fact, while inferential statistics focus is to analysis the data and hypothesis the results. What descriptive statistics does best is to simplify large amounts of data into a more manageable form. This is to avoid distortion and losing important details from the data.

The analysis used in descriptive statistics is univariate analysis. It involves in the calculations of each case of one variable at a time. The three main methods used in univariate analysis are:

- Distribution, which is the summary of the frequency of individual values or range of values for a variable. [15]
- Central Tendency, is the estimation of the "center" of a distribution value. [15]
- Dispersion, is referred to the spread of values from the central tendency. [15]

With these three methods we can present a factual analysis of the results we have evaluated. The use of these three methods for our data can be found below, in *Results*.

Wilcoxon Test

A Wilcoxon rank sum test is a comparison test of two sample data groups. The samples have to be independent of one another, equal in variance and spread; and have a normal distribution[17].

The most significant attribute of the Wilcoxon test is that the samples are independent of one another. Otherwise the rank sum test will not work. Wilcoxon rank sum test is well known to be a non-parametric test and non-parametric test is also known as distribution-free test, which means fewer assumptions are made of the outcome of the tests[17].

Considering these attributes of the Wilcoxon test, they will further analyse our studies and proving our hypothesis that the virtualized embedded board is faster than the physical board, based on the time that is required after a complete test run.

IV. Results

In this section we will discuss the obstacles that were faced during this study and the results that were collected at the end from the conducted tests and interviews.

A. Encountered Obstacles

Virtualization of the embedded board and the execution of the tests proved to be a challenge since we had to have six iterations during this work. In our first iteration we discovered that we had to switch from using Emul8 to QEMU, for the reason that Emul8 was no longer supported by it's developers. E.g. verification of the emulator, lack of community/resources, and no replies by mail or phone.

In our second iteration we came across issues with running the arm chipset on QEMU since the documentation provided from the developers for QEMU was unclear.

In order to run a specific chipset the user is required to configure and set-up the chip manually, and the information provided by the developers for the arm chipset was scarce. Our original goal was to use an arm chip, although the issues we faced while learning how to work with QEMU, made us discuss it with the CEO of the company, which resulted in the decision to use any chipset as long as it is running on Debian. It was determined to use an x86, and everything (from setting up the chip to installing Debian on the virtualized board) went smoothly without any problems. Although, the issue that we came upon after changing to an x86 chip was the configuration for the SSH connection between the host and the guest. Three weeks were spent on trying to figure out how to establish a SSH connection between the host and the guest with no success. Therefore, it was decided to use software, called Virtual Manager or virt-manager, that is responsible for configuring the SSH connection between the computer and the virtualized embedded board.

Virt-manager proved to be effective for setting up the configuration for a SSH connection between QEMU and the host computer. After an SSH connection was established we proceeded with doing the same but through a USB network. What we encountered was that the laptop had a faulty usb controller and it took us three days to discover that issue. What we did to avoid this issue was to install everything on another laptop.

Everything was installed properly on the new laptop and we were able to see that the usb controller detects the usb cable that is plugged in to the system. Although this time we encountered a problem with virt-manager, it was able to detect the usb that was connected to the computer but it was unable to establish a connection between the host and the virtualization of the x86 board. Time was running out at that stage and it was decided to do the testing through an SSH connection only.

Another obstacle that was encountered during this study was that the version of the provided software was older compared to the version of the unit-tests. Two days were spent on discovering and solving the issue with the provided software, and the test cases. It was communicated immediately with the company to provide a newer version of the software and that solved our problem.

The final issue that we faced was that the software, provided by Diadrom, was crashing inconsistently on the physical hardware which led to some results being dramatically shorter than the virtualization of the x86 chip. During the interview the issue was discussed with the employees of Diadrom and there was not enough time to solve it. Results needed to be collected and *figure 3, .figure 4., figure 5, figure 6, figure 7.* present that information.

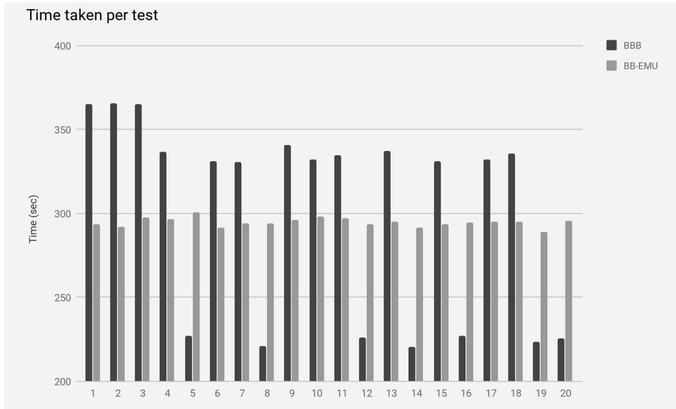


figure 3.

Tests	BB-EMU		Tests	BBB
1	293.76		1	365.02
2	292.23		2	365.55
3	297.42		3	365.07
4	296.37		4	336.78
5	300.46		5	226.93
6	291.65		6	330.94
7	294.07		7	330.77
8	294.01		8	221.15
9	295.92		9	340.77
10	297.87		10	331.99
11	297.01		11	334.59
12	293.53		12	225.85
13	294.89		13	337.08
14	291.58		14	220.52
15	293.48		15	331.24
16	294.82		16	227.09
17	294.93		17	331.88
18	294.91		18	335.91
19	289.24		19	223.45
20	295.54		20	225.38
min value	289.24	>	min value	220.52
quartile1 (q1)	293.5175	>	quartile1 (q1)	226.66
median	294.855	<	median	331.56
quartile3 (q3)	296.0325	<	quartile3 (q3)	336.855
max value	300.46	<	max value	365.55
Standard deviation	2.456113749	<	Standard deviation	56.8624162
Average	294.6845	<	Average	300.398
Variance	6.032	<	Variance	3233.334
Range	11.22	<	Range	145.03

figure 4.

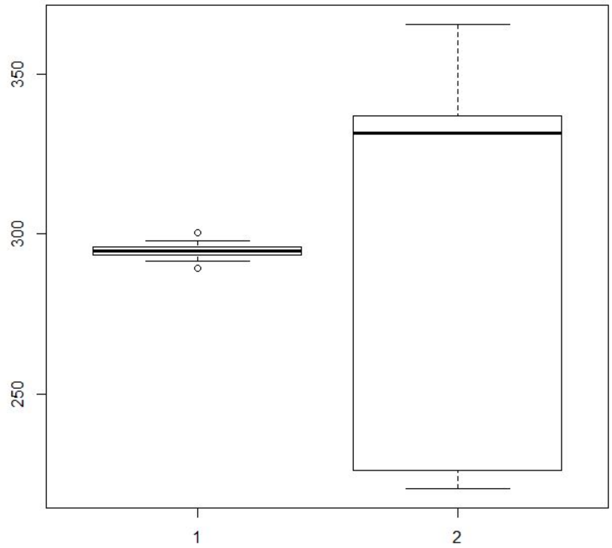


figure 5.

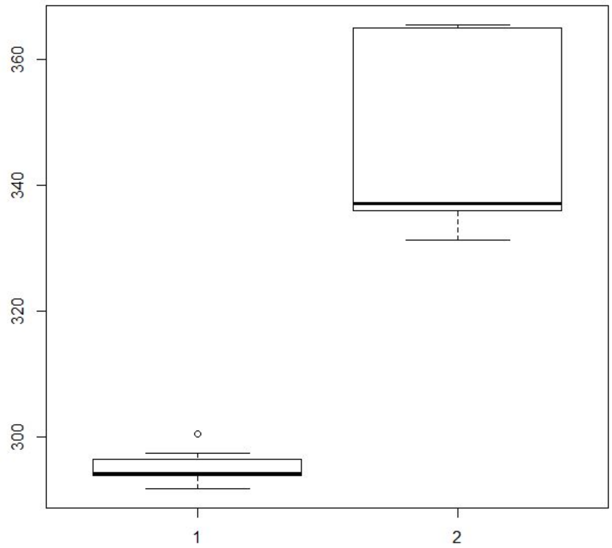


figure 6.

	BB-EMU	BBB
	Failures/errors	Failures/errors
1	0/0	0/0
2	0/0	0/0
3	0/0	0/0
4	0/0	0/0
5	0/0	14/8
6	0/0	1/0
7	0/0	1/0
8	0/0	14/8
9	0/0	0/0
10	0/0	1/0
11	0/0	0/0
12	0/0	13/17
13	0/0	0/0
14	0/0	14/17
15	0/0	0/0
16	0/0	14/18
17	0/0	1/0
18	0/0	0/0
19	0/0	14/18
20	0/0	15/18

figure 7.

B. Collected Results

After the final iteration and conducted tests we are confident that we have created a functioning virtualized testing environment. There is a stable connection between QEMU and the provided software from Diadrom which allowed us to successfully conduct our tests.

When the results were collected it was decided to use Descriptive Statistics[15] to summarize and compare our data. From *figure 4* we can see the total time required to execute a "full test". A "full test" is counted from the setup of the test environment to the completion of the executed testsuite.

In the figures we have the number of conducted tests, BB-EMU as the emulated hardware, BBB as the physical hardware, the time taken (in seconds) to complete a "full test" and the results from the Univariate analyses[15]. There is also a representation of our results with a Box-plot graph in *figure 5* and *figure 6.*

After conducting and comparing the Univariate analyses[15] it is noticeable that the fastest execution is recorded on the BBB side, although it is caused by crashes from the provided software which is uploaded on the BBB, *figure 7*. The time was recorded, after the crash, because the testsuite (when executed) continues to run through all of the test cases that are included and in the end it still displays the total time taken. Since the errors were inconsistent, the BBB results (that are lower than the average) for the BBB are considered as tests interfered by crashes from the provided software. From *figure 7* we can see when the crashes occurred and how much times it usually takes, when there are no problems with the provided software.
BB-EMU is overall faster compared to the BBB, the results from the BB-EMU are consistent with small deviation and variance, also the range of the results is dramatically shorter compared to the BBB.

To further analyze our results, to establish if there is a difference between the two population of results, we decided to use the Wilcoxon rank sum test. The R Project for Statistical Computing[16] was used for carrying out the Wilcoxon tests. The R project is used for statistical computing and is software that works with its unique programming language. The snippets of code, that were used for this thesis, are provided in the appendix section of this work. The first snippet of code is for the whole set of information for both systems. In the snippet, X represents BB-EMU and Y represents BBB. From the Wilcoxon test we can see that there is a difference between those two sets of results for the reason that the hypothesis X < Y (wilcox.test(X, Y, alternative = "less")) is true (alternative hypothesis: true location shift is less than 0). The results of the Wilcoxon test are visualised in *figure 5.* Number 1 in the boxplot presents BB-EMU and 2 presents BBB.

Since there are test runs on the BBB side with failures, it was decided to execute another Wilcoxon test, although this time it was done only on the successful attempts on the BBB. We believe that the failed test runs will not help in answering our research question, because they are not “full tests” (there should be no errors from the software that was provided from Diadrom). There were only 9 BBB full test runs with no failures or errors, so we selected those attempts and paired them with the same attempt number from the BB-EMU side. For example we see that attempt number 4 has complete test runs on both sides so we will pick those times from that test run (test run number 4). If we look again in the appendix, we can see another snippet of code under the first one. The BBB set contains all of the clean test runs and the EMU has the recorded times that were done at the same time with the BBB successful attempts. After implementing another Wilcoxon test it was discovered that there is a difference between the two sets of data and that our hypothesis of EMU < BBB (wilcox.test(EMU,BBB, alternative="less")) is true (alternative hypothesis: true location shift is less than 0). The results of the Wilcoxon test are visualised in *figure 6.* Number 1 in the boxplot presents BB-EMU and 2 presents BBB. In other words, the virtualized version performs faster and the Wilcoxon tests helped us prove it.

We also mentioned that we will talk about the correctness of the test cases, and what we discovered was that they are working properly, however only when there are no issues with the given software from Diadrom. BB-EMU had no issues with the communication on both sides and the final results were without any errors or failures. The version of the testsuite is the same for both systems, therefore the correctness of the testsuite is considered as very accurate.

C. *Interviews*

The overall opinion of all of the interviewees was that they were surprised that there were problems with the application and that our research helped them discover that something is wrong with the software. They also agreed that it is much better to test on an virtualized environment, for the reason that it is hard to have access to new boards that are being developed and having a virtual representation of those prototypes can speed up the software development process for embedded boards. Another opinion that was generally agreed upon from the interviewees was that you can easily automate tests on virtualized embedded boards and that an individual does not have to be busy with testing, and can focus on another task while the automated tests are running. They believe that virtualization can reduce software development time and can improve software quality since there will be more time for further testing or developing.

V. Discussion

The results that we evaluated from the tests were expected to perform at the range shown in *figure 3*. The figure shows how consistent BB-EMU performed and the results didn’t deviate far from our estimation tests. The pattern discovered from these results proves the reliability and stability of the emulators performance.

For the BBB, in *figure 7*, shows the amount of errors/failures occurred during the tests. This affected the performs and results of the testing, since when failure occurs during a test case and the application crashes due to that failure, all other test cases that are are incomplete automatically fails and skips to the end of the testsuite (if the application crashes). This led to a pattern to emerge where the reliability and stability affected the performance of the application.

A. Interviews

These results were also presented to the interviewees during the semi-structured interview, that confirmed the benefits of virtualizing hardware through its quality and speed. The answers and opinions we received from the interviewees mainly came to a consensus that the benefits of a hardware emulation can improve development time and reduce time taken on solving hardware issues. However the correctness data raised the interviewees concerns and interest, on the differentiation of the two systems (BB-EMU & BBB). When discussing about the differentiation, the assumptions that were made was about the differentiation of the chipsets and their clock speed, the speed of processing data. This also made one of the developers question the choice of using the ARM chipset hardware.

B. Research Question

The main focus of the research question, *"Can virtualizing the test objects reduce time required for executing a test process, compared to the time required on an embedded board?"*, is proven by the results of BB-EMU. Time required can be justified by comparing the results from the BB-EMU *total time taken,* which shows the reduction of time taken for the procedure of startup and testing, to BBB *total time taken*. This clearly proves that it is faster to test software by using a virtualized version of the embedded board.

C. Limitations

The studies limitation were contrived by lack of resources from QEMU, incomplete applications and tests, and non functioning USB connections. The resources from QEMU gave us few choices to develop our emulated system, also didn't provide the necessary knowledge to develop the artefact. These limitations affected the outcome of our results and our conclusion.

D. Future Study

We believe that our study is covering just a small section of what can be examined on this topic of software testing. Further research can be conducted on, for example, measuring the similarities in performance between a prototype embedded board (that is about to be released) and a virtualization of that board. Examine if the virtualized version has the same performance as the real prototype. With the results of that research it might be possible to determine whether companies, that create embedded boards, should provide software companies with a virtualized version of their prototypes prior to release. That way software companies should be able to develop future applications before the release of the actual board. Therefore, the creation of products with embedded boards should be faster to release to the public or more time will be available for further developing/improving the product.

E. Benefits

The benefits of using an emulated system are the cost and speed of the system. The cost of the system is a huge factor in the market of embedded boards. Depending on the market, the cost of manufacturing an embedded device can become really expensive, especially when creating several prototypes. The quality of the product can also affect the development/testing process. An example that accumulates both cons, is when a malfunctioning prototype is used for testing however due to manufacturing faults (specifications/components) another prototype is required to be manufactured, adding to the initial cost. Furthermore having the ability to customise the hardware, without being constraint of manufacturing time and cost, allows for more flexible development and can monitor the stability of the hardware before conforming to hardware prototypes or products. This speeds up development time due to the ease of distribution of the emulator without having to buy boards and setup each board.

These benefits were brought up by the interviewee's and all had the same context of the subject.

Another interesting proposal mentioned by interviewee 4, from the interview table, is if the board manufacturer could distribute an emulator, of the prototype, to the customers/developers, before release of the actual board, to test and verify the customers software.

VI. Conclusion

As mentioned in the discussion the main part of the research question is the reduced speed when testing on a virtualized embedded board. This is supported by the test results of time taken since it clearly shows the speed of the test procedures.

Comparing both patterns, in *figure 4*, to each other shows the quality within speed of an emulated hardware to a physical hardware by the failure that occurred. This reveals that using an emulation provides more time on the software development rather than constantly evaluating the issues on the hardware. This also concludes that having an emulation that is constructed with only the necessary modules for the application reduces the time and effort than having to do time consuming research and setup for modules on the physical hardware. Due to the existence of several modules with near similar component names or identifications.

Even though BBB uses an ARM chipset and the BB-EMU emulates an x86 chipset, the difference does not impact the evaluation, based of the research question, since the results from BBB (without failures/errors) can be compared with the BB-EMU's results, which can be analysed with a positive outcome.

References

[1] Basili, V. and Selby, R. (1985). *Comparing the effectiveness of software testing strategies*. College Park, Md.: University of Maryland.

[2] Diadrom. (2018). Hem. [online] Available at: http://diadrom.se/ [Accessed 26 Mar. 2018].

[3] Simon Wallström, Adam Dalentoft. (2016). *Emulation-based Software Development for Embedded Systems*. Department of Computer Science Faculty of Engineering LTH, Lund University.

[4] Erek Göktürk. (2014). *A stance on Emulation and Testbeds, and a Survey of Network Emulators and Testbeds*. Department of Informatics, University of Oslo, Blindern, N-0371 Oslo, Norway.

[5] S. Edwards, L. Lavagno, E. A. Lee, and A. Sangiovanni-Vincentelli. (1997). *Design of Embedded Systems: Formal Models, Validation, and Synthesis.* University of California, Berkeley, USA.

[6] Alan R. Hevner, Salvatore T. March, Jinsoo Park. (2004). *Design Science in Information Systems Research*, MIS Q. 28, City of Co., Rep. 75-105.

[7] Vocabulary.com. (2018). *simulation - Dictionary Definition*. [online] Available at: https://www.vocabulary.com/dictionary/simulation [Accessed 20 Apr. 2018].

[8] Kb.nl. (2018). *What is emulation? | Koninklijke Bibliotheek*. [online] Available at: https://www.kb.nl/en/organisation/research-expertise/research-on-digitisation-and-digital-preservation/emulation/what-is-emulation [Accessed 20 Apr. 2018].

[9] Virtualization in Education. (2007). [pdf] Encinitas, CA 92024: The Greaves Group. Available at: http://www-07.ibm.com/solutions/in/education/download/Virtualization%20in%20Education.pdf [Accessed 20 Apr. 2018].

[10] Mazidi, M., Mazidi, J. and Causey, D. (2010). *The x86 PC*. Boston: Prentice Hall.

[11] Ubuntu.com. (2018). *The leading operating system for PCs, IoT devices, servers and the cloud | Ubuntu*. [online] Available at: https://www.ubuntu.com/# [Accessed 20 Apr. 2018].

[12] Wiki.qemu.org. (2018). *QEMU*. [online] Available at: https://wiki.qemu.org/Main_Page [Accessed 20 Apr. 2018].

[13] Virt-manager.org. (2018). *Virtual Machine Manager Home*. [online] Available at: https://virt-manager.org/ [Accessed 20 Apr. 2018].

[14] Stephanie (2017). *Statistical Analysis: Definition, Examples*. [online] Available at: http://www.statisticshowto.com/statistical-analysis/ [Accessed 24 May. 2018]

[15] Trochim, William M. K. (2006). "Descriptive statistics". *Research Methods Knowledge Base*. Retrieved 14 March 2011

[16] R-project.org. (2018). R: The R Project for Statistical Computing. [online] Available at: https://www.r-project.org/ [Accessed 18 Apr. 2018].

[17] Data.library.virginia.edu (2017). Wilcoxon Rank Sum Test. [online] Available at: https://data.library.virginia.edu/the-wilcoxon-rank-sum-test/ [Accessed 20 June. 2018]

APPENDIX

Statistical Tests

```
.>
> X <-
c(293.76,292.23,297.42,296.37,300.46,291.65,294.07,294.01,295.92,297.87,297.01,293.53,294.89,291.58,293.48,294.82,294.93,294.91,289.24,295.54)
> Y <-
c(365.02,365.55,365.07,336.78,226.93,330.94,330.77,221.15,340.77,331.99,334.59,225.85,337.08,220.52,331.24,227.09,331.88,335.91,223.45,225.38)
> wilcox.test(X, Y, alternative = "less")
	Wilcoxon rank sum test
data:  X and Y
W = 140, p-value = 0.05404
alternative hypothesis: true location shift is less than 0
```

Results from virtualized and physical embedded board tests without failures and errors:

```
> BBB <-c(365.02,365.55,365.07,336.78,340.77,334.59,337.08,331.24,335.91)
> EMU <- c(293.76,292.23,297.42,296.37,300.46,291.65,294.07,294.01,295.92)

> wilcox.test(EMU,BBB, alternative="less")
	Wilcoxon rank sum test
data:  EMU and BBB
W = 0, p-value = 2.057e-05
alternative hypothesis: true location shift is less than 0
```


ACKNOWLEDGEMENT

Supervisors

Software Engineering & Management, Supervisor
Regina Hebig

Diadrom, Supervisor
Henrik Fagrell, CEO