A study on the effects on performance and maintainability the libuv library has on low-cost ARM-based IoT-devices
–

Natanael Log

Handledare : Min handledare
Examinator : Min examinator

Sammanfattning

`Abstract.tex`

Författarens tack

Acknowledgments.tex

Innehåll

1 | Introduction

The Internet of Things (IoT) is a new dimension of the internet which includes *smart devices* - physical devices able to communicate with the outside world through internet [5]. Ericsson foretold there will be around 400 million smart devices by the end of 2016 [1] and a great challenge will be to serve all of these. IoT applications are already influencing our everyday lives; smart cities, smart grids and traffic management are only a few of the large variety of applications in this "multi-tiered heterogeneous system based on open architectural platforms and standards" [3]. The complexity of a smart device varies from passive ID-tags which communicates through near field communication to full scale computers with multithreaded operating systems.

Advanced RISC Machine (ARM) processors are used in different types of electronic devices. RISC is short for *Reduced Instruction Set Computer* and its supported instructions are simple, have uniform length and almost all instructions execute in one clock cycle. This reduces the complexity of the chip and gives more room to performance enhancing features [4]. In an IoT context, ARM processors can be a good choice to be embedded on a smart device due to its large variety in cost, energy consumption, performance and size.

From a network perspective, smart devices can be seen as clients in a traditional client-server architecture. They emit data to a server through an internet protocol. The server receives the data and processes it. When building these kinds of network services it is important for the server to avoid blocking concurrent requests [2]. These requests are both on a network level but also on the general operating system level. Requests from a smart device might be logged on a file or written to a database, meaning the server must start new processes to handle these requests without blocking new requests from the network. However, a server does not necessarily have to be a web server serving requests on the internet protocol. The operating system itself can be viewed as an I/O (input/output) server receiving requests from the I/O layer. This can be data on serial ports and general purpose I/O pins on the processor. A device monitoring a physical environment using sensors can take use of this feature by letting the registers call interrupt handlers when they are ready for reading or writing. Thus an application running on an embedded system listening for data from sensors can be viewed as a web server listening for internet requests. The same challenge regarding non-blocking request handlers holds also for these mentioned applications.

To determine the maintainability of a software system one can, among others, look at the source code and its age since release, the number of non-commented source code statements and its rate of failures per time unit [6].

1.1 Motivation

Since IoT is growing and communication will largely increase, efficiency and performance is an important factor. There exists a number of communication protocols for IoT network architectures. A big challenge is to support massive data streams with minimal overhead in transport. However, this implies for the transport and communication perspective on IoT, but another significant perspective is I/O (file writes, database queries) processing on the machines themselves. If the demand for high performance IoT services is increasing development towards embedded systems will too. A rigor evaluation on two common architectures can therefore be of great value to find what suits best in IoT.

1.2 Aim

To conclude whether libuv is a good option in terms of performance and maintainability when selecting underlying architecture for IoT devices.

1.3 Research questions

1. How does a smart device in a specific context compare in performance when the underlying architecture uses libuv instead of a traditional one?

2. How does an event-driven architecture using libuv compare to a multi-threaded environment in terms of performance?

3. How does an event-driven architecture using libuv compare to a multi-threaded environment in terms of some software metrics?

4. How can the software performance for source code written in libuv for an IoT context be predicted?

5. How does Javascript compare to C++ in an I/O-intensive embedded application in terms of performance?

1.4 Delimitations

Maintainability will be the main metric used. Other metrics might be useful for evaluation of the architectures, but due to time limitations only maintainability will be used. Only one or two types of devices will be used for implementation and testing. This might decrease data resolution.

2 Theory

The main purpose of this chapter is to make it obvious for the reader that the report authors have made an effort to read up on related research and other information of relevance for the research questions. It is a question of trust. Can I as a reader rely on what the authors are saying? If it is obvious that the authors know the topic area well and clearly present their lessons learned, it raises the perceived quality of the entire report.

After having read the theory chapter it shall be obvious for the reader that the research questions are both well formulated and relevant.

The chapter must contain theory of use for the intended study, both in terms of technique and method. If a final thesis project is about the development of a new search engine for a certain application domain, the theory must bring up related work on search algorithms and related techniques, but also methods for evaluating search engines, including performance measures such as precision, accuracy and recall.

The chapter shall be structured thematically, not per author. A good approach to making a review of scientific literature is to use *Google Scholar* (which also has the useful function *Cite*). By iterating between searching for articles and reading abstracts to find new terms to guide further searches, it is fairly straight forward to locate good and relevant information, such as [**test**].

Having found a relevant article one can use the function for viewing other articles that have cited this particular article, and also go through the article's own reference list. Among these articles on can often find other interesting articles and thus proceed further.

It can also be a good idea to consider which sources seem most relevant for the problem area at hand. Are there any special conference or journal that often occurs one can search in more detail in lists of published articles from these venues in particular. One can also search for the web sites of important authors and investigate what they have published in general.

This chapter is called either *Theory, Related Work*, or *Related Research*. Check with your supervisor.

3 Method

In this chapter, the method is described in a way which shows how the work was actually carried out. The description must be precise and well thought through. Consider the scientific term replicability. Replicability means that someone reading a scientific report should be able to follow the method description and then carry out the same study and check whether the results obtained are similar. Achieving replicability is not always relevant, but precision and clarity is.

Sometimes the work is separated into different parts, e.g. pre-study, implementation and evaluation. In such cases it is recommended that the method chapter is structured accordingly with suitable named sub-headings.

4 Results

This chapter presents the results. Note that the results are presented factually, striving for objectivity as far as possible. The results shall not be analyzed, discussed or evaluated. This is left for the discussion chapter.

In case the method chapter has been divided into subheadings such as pre-study, implementation and evaluation, the result chapter should have the same sub-headings. This gives a clear structure and makes the chapter easier to write.

In case results are presented from a process (e.g. an implementation process), the main decisions made during the process must be clearly presented and justified. Normally, alternative attempts, etc, have already been described in the theory chapter, making it possible to refer to it as part of the justification.

5 Discussion

This chapter contains the following sub-headings.

5.1 Results

Are there anything in the results that stand out and need be analyzed and commented on? How do the results relate to the material covered in the theory chapter? What does the theory imply about the meaning of the results? For example, what does it mean that a certain system got a certain numeric value in a usability evaluation; how good or bad is it? Is there something in the results that is unexpected based on the literature review, or is everything as one would theoretically expect?

5.2 Method

This is where the applied method is discussed and criticized. Taking a self-critical stance to the method used is an important part of the scientific approach.

A study is rarely perfect. There are almost always things one could have done differently if the study could be repeated or with extra resources. Go through the most important limitations with your method and discuss potential consequences for the results. Connect back to the method theory presented in the theory chapter. Refer explicitly to relevant sources.

The discussion shall also demonstrate an awareness of methodological concepts such as replicability, reliability, and validity. The concept of replicability has already been discussed in the Method chapter (3). Reliability is a term for whether one can expect to get the same results if a study is repeated with the same method. A study with a high degree of reliability has a large probability of leading to similar results if repeated. The concept of validity is, somewhat simplified, concerned with whether a performed measurement actually measures what one thinks is being measured. A study with a high degree of validity thus has a high level of credibility. A discussion of these concepts must be transferred to the actual context of the study.

The method discussion shall also contain a paragraph of source criticism. This is where the authors' point of view on the use and selection of sources is described.

In certain contexts it may be the case that the most relevant information for the study is not to be found in scientific literature but rather with individual software developers and

open source projects. It must then be clearly stated that efforts have been made to gain access to this information, e.g. by direct communication with developers and/or through discussion forums, etc. Efforts must also be made to indicate the lack of relevant research literature. The precise manner of such investigations must be clearly specified in a method section. The paragraph on source criticism must critically discuss these approaches.

Usually however, there are always relevant related research. If not about the actual research questions, there is certainly important information about the domain under study.

5.3 The work in a wider context

There must be a section discussing ethical and societal aspects related to the work. This is important for the authors to demonstrate a professional maturity and also for achieving the education goals. If the work, for some reason, completely lacks a connection to ethical or societal aspects this must be explicitly stated and justified in the section Delimitations in the introduction chapter.

In the discussion chapter, one must explicitly refer to sources relevant to the discussion.

6 Conclusion

This chapter contains a summarization of the purpose and the research questions. To what extent has the aim been achieved, and what are the answers to the research questions?

The consequences for the target audience (and possibly for researchers and practitioners) must also be described. There should be a section on future work where ideas for continued work are described. If the conclusion chapter contains such a section, the ideas described therein must be concrete and well thought through.

Litteratur

[1] URL: https://www.ericsson.com/en/mobility-report/internet-of-things-forecast.
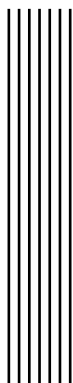
[2] Khaled Elmeleegy, Anupam Chanda, Alan L Cox och Willy Zwaenepoel. "Lazy Asynchronous I/O for Event-Driven Servers." I: *USENIX Annual Technical Conference, General Track*. 2004, s. 241–254.

[3] Gordana Gardašević, Mladen Veletić, Nebojša Maletić, Dragan Vasiljević, Igor Radusinović, Slavica Tomović och Milutin Radonjić. "The IoT architectural framework, design issues and application domains". I: *Wireless Personal Communications* 92.1 (2017), s. 127–148.

[4] T. Jamil. "RISC versus CISC". I: *IEEE Potentials* 14.3 (aug. 1995), s. 13–16. ISSN: 0278-6648. DOI: 10.1109/45.464688.

[5] Hermann Kopetz. *Design Principles for Distributed Embedded Applications*. Springer, 2011.

[6] Paul Oman och Jack Hagemeister. "Metrics for assessing a software system's maintainability". I: *Software Maintenance, 1992. Proceedings., Conference on*. IEEE. 1992, s. 337–344.