Reading Report: Hebert13

Albert Bausili Fernández

February 9, 2023

1 Content

1.1 Identify the genre of the document, its purpose, and its target audience.

The document is an article. Its main purpose is to spread information and advices to newcomers to erlang language or more less inexperienced users of the language.

1.2 Summarize the document, indicating the key concepts.

The article talks about the fallacies or common roots of problems that programmers (specially newcomers) may encounter when deploying an erlang program and how can they be addressed and why its important to be aware of them. Here is a quick summary of the eight points that the article focuses into:

- The Network Is Reliable

  As erlang is a language meant (but not necessarily) to be distributed among multiple and even different hardware is important to note that the network that interconnects all these nodes may sometimes be slower or some connections may be down for quite some time, thus the code should be able to overcome these situations by handling it for example as local crashes as the language itself does not provide any special measure except detecting that something went wrong.

- There Is No Latency

  Erlang as a distributed language that it is allows the code to call functions that do not need to be in the same machine as the running code. In this cases there will be a noticeable latency much higher than a call to a local function. The language always expects this delay by design but is important that the programmer is aware of this possible delays when designing the application.

- Bandwidth Is Infinite

  As the technology has progressed significatively these years it is easy to fall into the assumption that sending huge amounts of data comes with no problems besides the time that the data will need to get to the destination. However, this could not be farther from the truth. A very large message may be blocking for all the other messages, specifically you might be blocking the “heartbeats” messages that Erlang nodes send each other periodically. Those messages are the ones responsible for keeping “alive” the nodes and knowing if some nodes are down or not. The solution to this occurrence is by maintaining a relatively small size of messages.

- The Network Is Secure

  The erlang language was conceived in a past were erlang applications used to communicate inside the same network and in extreme situations, thus the designers opted to give more importance to reliability than to security. Nowadays erlang applications can rarely be spread among different datacenters because of decisions made by the original designers. However, there are some alternatives to try to give that needed security layer to the language such as: Switching to SSL, Implementing their own high-level communication layer, tunneling over secure channels and reimplementing the communication protocol between nodes. And even applying this measures it is highly recommended to be careful, as gaining access to a node will grant the intruder access to all the rest of the nodes.

- Topology Doesn't Change

  Hardcoding IP's or any other topological details is usually a bad idea, as machines can be added, retired, hardware can die, etc… There exist some interesting libraries that abstract the names and topology to simplify this problematic.

- There Is Only One Administrator

  In a great part of the enterprises where the distributed software is produced not all the nodes are or must be of the same group, it may be third parties and other branch or many other examples of different agents that may have different versions of the software or with different parts upgraded. To prevent this becoming a problem is important to develop mechanisms for the algorithms to detect differences in the versions and the versions of other components and also providing a tool to enable admins control and diagnose the state of the program.

- Transport Cost is Zero

  Transporting data between nodes comes with a price, in time and in money terms. This was not taken in consideration fully by the original developers and did not implement a compression mechanism to reduce the bandwidth usage, however, they chose to let the programmer implement their own communication layer if they need to.

- The Network Is Homogeneous

  Nowadays there are a huge variety of programming languages and as it can be deduced, by default erlang is not compatible with all of them. That rises a problematic, as developers may find in a situations where they need to integrate different language nodes in a same system. The original designers opted to allow any node that behaves as an erlang node (respects its protocol) to be integrated in the network.

2 Assessment

2.1 Rate the readability of the document: easy, readable, difficult, unreadable.

Easy

2.2 Give your opinion of the reading assignment, indicating whether it should be included in next year's course or not.

I found it a great introduction to the language problematics (And general problems of distributed languages) and some examples of solutions for beginners of the language such as the students. It was pretty fun to read as it contained some funny parts that made it light to go through it.