An Erlang Distributed Database: How to Better Understand The Cap Theorem

Graham Matthews, PhD; Damian Mirizzi, UG

Abstract—Description of Area of study:
When making large ubiquitous systems such as google calendars or email you come across the issue of storing massive amounts of data. It is inefficient to store all that data on one machine as well as unsafe because if that one system was compromised then all the data would be corrupted. Companies such as Apache or MongoDB have developed open source databases that can scale across multiple situations to solve this problem, while giving end users a simple api to develop with. I will be studying Distributed Systems and algorithms to develop my own interpretation of a distributed database.
The Problem Being Addressed:
There is a Theorem in Distributed systems called the CAP Theorem it is described the balance of these three attributes: Consistency, every read would get you the most recent write. Availability, every node (if not failed) always executes queries. Partition-tolerance, even if the connections between nodes are down, the other two (A & C) promises, are kept.I would like to produce a database that balances the attributes well for large distributed systems with smaller nodes such as IOT devices.

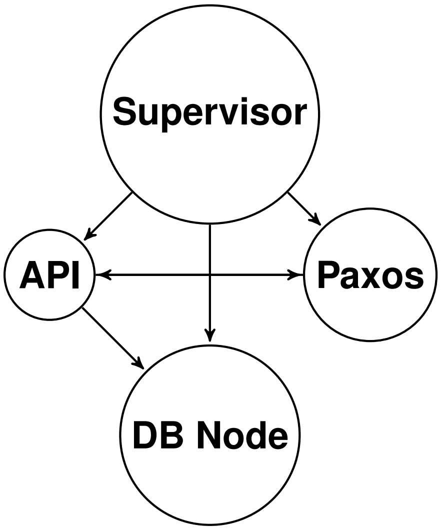
Index Terms—Computer Science, Distributed Systems, CAP.

1 Introduction

THIS demo file is intended to serve as a “starter file” for IEEE Computer Society journal papers produced under LaTeX using IEEEtran.cls version 1.8a and later. I wish you the best of success.

2 Structure

THIS Is the second chap



2.1 Subsection Heading Here

Subsection text here.

3 Conclusion

The conclusion goes here.

Acknowledgments

The authors would like to thank...

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

Damian Mirizzi Computer Science Major at California Lutheran University. Career Objective: Seeking to obtain a position where I may further facilitate my own knowledge of computer science and math while utilizing my skills to contribute to the advancement of innovative and stimulating technology. Profile Java, C; Proficient in Python, LaTeX and Matlab; OOP & Functional. Familiarity of tools such as Git, Unix, Linux, Eclipse, NetBeans, Qt, SSH, Confluence, and Jira. Quick to learn new materials Able to work well in teams while maintaining discipline to work independently Tools Docker, Agile/Scrum, Mob, CSS, Javascript, JSON, XML, Git, Bootstrap, Vim, SSH, VNC, Unix, Eclipse, Ubuntu, Mac, Windows, VMware, Jira, and XConfluence.

Graham Matthews B.Sc. (Computer Science) with First Class Honors, University of Auckland, Auckland, New Zealand.
M.Sc. (Computer Science) with Distinction, University of Auckland, Auckland, New Zealand.
Ph.D. (Mathematics), University of Georgia at Athens, Athens, USA.

- *Graham Matthews is with the Department of Computer Computer Science, California Lutheran University, CA, 91360.*
  E-mail: gmatthew@callutheran.edu
- *D. Mirizzi is with California Lutheran University.*

Manuscript received April 1, 2018; revised April 29, 2018.