Clustered network-attached storage: fault tolerance and performance.

Meeting Notes

Jonathon A. T. Carter

UP896692

School of Computing

Final Year Research Project (PJE40-R)

May 4, 2022

Supervised by

Dr Rinat Khusainov

Info

A collection of meeting notes taken throughout the research project from 11/10/2021 - 06/05/2022

Project repository

All project code and data can be viewed within the Github repository (Carter, 2022).

Consent to share

I consent for this project to be archived by the University Library and potentially used as an example project for future students.

Table of Contents

Chapter 1

17/11/2021

1.1 Progress to date

- Started literature review: got a plan and topics to cover; 16 papers to reference, most from IEEE journals and some conferences
- Made a good start with using Latex; enjoying it!
- Plan to have acronyms page
- Lit review question - "What are the most important aspects of a distributed file system?"

1.2 Plan for next week

- Talk about the processes needed futher down the line in the lit review
- Add/link the diagrams we discussed
- Aiming to have background research/ lit review draft by 30 Nov
- Start focusing more on problems/question we are investigating, what comparison metrics/criteria can be associated with them, and what experiments we could do to get the data
- Fault tolerance mainly focuses on user satisfaction / interaction whereas software focuses on the design and user requirments

Chapter 2

25/11/2021

2.1 Questions

- Who do I talk to about the lab computers I need to use?

  Email Jack

- Is a "heterogeneous system" a hardware or performance related area?

  https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6424849 - diverse different

2.2 Progress to date

- Continued work on background research

2.3 Plan for next week

- Complete the first draft of the background research
- Make a start on the methodology
- Clarify access arrangements to the lab computers
- Get better clarity on what we will be focusing on in the project

Chapter 3

01/12/2021

- Artefact

  Software A, B & C - justify why selected - eg popular or easily configurable

  Measuring fault tolerance of a system

  Tests to take offline nodes (1,2 or 3) - measures how the system copes with load, responsiveness and total rebuild time taken

  Comparison of software with / without erasure coding - measuring total recovery time and system load

  Measuring avaliable software protocol connections

  Compare software connection methods - measures read / write between each software and respective OS protocol.

- First draft 3/4 of background research done
- Contacted Vasileios Adamos about computer lab
- Jack suggested VM's, I think this could be an easier method?

  However I'm not sure if this will cause anomalies in my results if 2x nodes are running with shared CPU and RAM.

3.1 Plan for next week

- When selecting software candidates, look for features that can make a difference for our experiments
- Aim for a variety in the above respect
- Focusing on two aspects:

  Fault tolerance

  Performance

- Continue with background research, aim to finalise
- Look further into the lab setup

Chapter 4

08/12/2021

4.1 Questions

- Iterative model good for this type of project? - with mixture of agile sprints
- Unsure on how I will tie in software setup and working on the artefact, should I have the artefact fully complete first before running tests?
- Automated testing for python and how to test this specific kind of artefact?

4.2 Progress to date

- Found out who I need to contact directly for the computer labs, will need to work after 5pm / weekends to get hands on. Need to get lab timetables
- Found out how I can get google cloud to do setup testing on (set up of the DFS software)
- Methodology drafted up with justified method, though I'm unsure on if this will work for my specific case of artefact finished first
- Some further look into Softwares chosen GlusterFS / Ceph, need to pick a performance Software
- Literature review drafted
- Still waiting on marking scheme and not sure if I'm going off track a little

- Have some system/user requirements drafted, need to work more on functional / non-functional
- Draft the methodology for selecting candidate cluster storage systems
- Draft the research objectives
- Start planning experimental scenarios
- Finalise selection of the candidate systems
- Look into installation/OS/etc requirements for the selected systems
- Based on the above, look at the requirements for the testbed
- Make a start with testbed design

Chapter 5

05/01/2022

5.1 Progress to date

- Done specification of research questions (and discussion)
- Made a start on research design
- Got preliminary requirements for the testbed

5.2 Plan for next week

- Finish research design
- Finalise selection of the candidate systems
- Finalise into installation/OS/etc requirements for the selected systems
- Finalise requirements for the testbed
- Start looking into characterising different load patters

  We are looking to identify key characteristsis (such as operation frequency, mix between reads and writes, files sizes etc)

  Get an idea of mapping between different load patters and real world applications
- Make a start with testbed design

Chapter 6

12/01/2022

6.1 Questions

- Research analysis strategy what would be the most straight forward and presentative for this project? or any suggestions?
- Do I tak about VM usage for running the candidates in research design or discussion of the testbed design and implemenation suppporting

6.2 Progress to date

- Drafted research design
- Chosen candidate systems GlusterFS, MooseFS and Ceph (reasoning behind this explained in research design)
- Researched candidate systems installations for ubuntu 18.04 (can replicate in VM)
- Requirements drafted

6.3 Plan for next week

- Completely flesh out requirements with reasoning
- Draft some hypothesis

- Start testbed design
- Start looking into characterising different load patters

  We are looking to identify key characteristsis (such as operation frequency, mix between reads and writes, files sizes etc)

  Get an idea of mapping between different load patters and real world applications

Chapter 7

19/01/2022

7.1 Progress to date

- half drafted testbed design
- some hypotheses made
- found some more tools to intergrate into testbed glances, telegraf
- found papers on characterising load patterns will be using FIO, found similar tool IOR but is difficult to implement
- started to think about result analysis / presentation

  found formula for evaluating performance degragation of a system which I will use
- have contacted about the lab equipment and had a response

7.2 Plan for next week

- Finalise the design
- Read through workload characterisation papers and put together out ideas for this
- Make a start on the environment setup with VMs

7.3 Discussion

For fault tolerance: we can distinguish between 3 phases: before failure, failure mitigation, and after failure; 1 and 3 can be measured separately for any given capacity. 2 can also be measured separately (in terms of duration, and system performance during that phase). Then, the total system performance for failure at any time can be estimated as the combination of these three phase in the corresponding proportions.

Chapter 8

26/01/2022

8.1 Questions

- 10:30am Wed doesn't work anymore due to IoT what is another good time to do a weekly meet on?
- idea of using telegraf ¿ influxDB ¿ grafana for metric collection and presentation

8.2 Progress to date

- tested out multiple virtual machine deployment from clone, have a test base image to deploy that comes with setup.sh script

8.3 Plan for next week

- Check about additional computer to act as the client (load generator)
- Send the new timetable for scheduling meeting in TB2
- Check precision of the 'telegraf'

Chapter 9

11/02/2022

9.1 Progress to date

- Cluster is on private subnet
- Testing of telegraf sucessful
- Some exploration into Ubuntu image and base configuration,
- Script setup to include static ips in the range 10.1.100.130 164, architecture adjusted to include this change and the standalone data collection machine

9.2 Plan for next week

- Look into FIO command building, search against the papers cited/used to be cited
- Ask about the extra disks
- Offline testing at home with influxDB and Grafana

9.3 Demo work

- Showcase idea of having live demo
- Diagrams, actual photo of cluster and grafana visuals

Chapter 10

18/02/2022

10.1 Progress to date

- Cluster machines named, setup and disks added need to double check seperate disk for proj-management
- FIO information / papers found need to convert this into commands
- Started on a python testbed where I can run and control different aspects of the cluster (fairly basic atm)
- Installed InfluxDB and Grafan onto the proj-data machine, need to setup the link between Influx and Grafana but working otherwise
- Image for the nodes is ready, need to deploy to the cluster
- Have the information for setting up each file system, will practice/test at home before I deploy

10.2 Plan for next week

- Setup cluster nodes, link with influxDB and Grafana
- Setup for demo where launch FIO and then measurements showing in Grafana
- Setup MooseFS (master), test
- Setup singular instance for backup

- Summary for showcase

Chapter 11

25/02/2022

11.1 Progress to date

- Cluster nodes setup with MooseFS
- Tesed FIO into mfs mount point
- Progress demo slides and video completed

11.2 Plan for next week

- Test r/w traffic with a Raspberry Pi
- RK/LY look through video/slides and feed back
- FIO read/write patterns finalise and email when ready
- Finish testbed
- Wireshark/network sniffer

Chapter 12

04/03/2022

12.1 Progress to date

- Tested network with wireshark and iperf3 (found the bottleneck to be the 100Mbps switch)
- Found further Result Analysis graph tool + FIO benchmarking scripts on Github (intend to use)
- Played with FIO command breakdown and have sped up / increased file size
- Separate client machine added to interact with cluster
- Little bit slow this week

12.2 Plan for next week

- Continue on validation of FIO patterns and the FIO benchmarking script usage / sources to back up its usage
- Relate FIO-plot / bench-fio to found papers
- Finish tesbed continued
- Start result collection against MooseFS, store and sort results safely
- Paper formatting / examples

Chapter 13

11/03/22

13.1 Progress to date

- Testbed finished
- MooseFS results gathered

13.2 Plan for next week

- Paper formatting / examples
- Continue on validation of FIO patterns and the FIO benchmarking script usage / sources to back up its usage
- Relate FIO-plot / bench-fio to found papers
- GlusterFS setup?
- Focus on writeup

Chapter 14

18/03/2022

14.1 Progress to date

- GlusterFS setup
- MooseFS results gathered
- MooseFS results applied to FIO-plot and Fault Tolerance formula -¿
- Design section has been written up
- FIO patterns has been drafted

14.2 Plan for next week

- Focus on showcase demo

  https://www.youtube.com/watch?v=I2XuW9EJ8xo

- Focus on the research design

  How are you going about solving your question

  How are you chosing your systems and why

  How are you going to compare the systems

  The metrics that are being used

  How we deal with the results

Talk about FIO patterns in here with workloads

- Focus on the distributed systems coursework

  https://www.overleaf.com/read/pzrvggnpkrwy

Chapter 15

25/03/2022

15.1 Showcase feedback

- Results are hard to understand and read
- Performance Degradation should not be negative ie there is no performance degradation
- Present as a matter of computation time? Should I therefore have multiple tests for each type of I/O
- Measure results against normal completetion time?
- Could I reference my distributed systems coursework which is about replication, this could be another factor that contributes to time taken

15.2 Progress to date

- Research showcase + feedback
- Finished Dist systems coursework
- Drafted another idea for research design

15.3 Plan for next week

- Find out how MFS & GFS operate

- Hypothesis for each typology
- Draw up tests for each typology hypothesis
- Gatherfurther min/avg/max results on time taken during failure
- Write up implentation of MFS & GFS
- Write up implentation of testbed

Chapter 16

31/03/2022

16.1 Progress to date

- Research design redrafted need to do further additions
- testbed further developed to be automated

16.2 Plan for next week

- Break down the research dfs table specifially to my design
- further comparisons between master and p2p
- Automate time during failure, possibly via array?
- remove performance degradation formula

Chapter 17

05/04/2022

17.1 Progress to date

- Research design additions added, final draft
- Results gathering continuing
- Added further error handling into the testbed due to slowdown of collection (had a bad bug with glusterfs)
- Testbed fully automated for result collection

17.2 Plan for Easter break

- Finish Result collection
- Finish Testbed / Cluster design docs writeup what tools I am using
- Finish Implementation writeup
- Add in research question section to latex
- Add in break down of DFS in table with different aspects after literature section
- Ethical section inside research design

17.3 Plan by the 29th

- Have above finished

- Finish Result presentation
- Finish Result discussion, conclusion
- Finish Future work and project evaluation
- Gather images and snippets for doc
- Finish Introduction
- Code commenting

17.4 Plan from 30th - 6th

- Latex
- Polish, Grammar ect
- Check References work
- Acronyms section

Chapter 18

26/04/2022

18.1 Progress to date

- All progress from easter finished (see chapter 17, p24)
- Final draft ready

Chapter 19

03/05/2022

19.1 Amendments

- Put Literature into background research
- Chapter 3 (Research Questions) more context for the Research Questions
- Chapter 4 (Research Design) hypotheses

  Shuffle sections around (test details closer to experiments table)

  More context for ethics needed
- Chapter 9/10 (Results, Result Discussion) Merge

  Change layout
- Update Introduction with changes and rest of document links
- Reference Github

References

Carter, J. (2022). *Clustered network-attached storage: fault tolerance and performance.* https://github.com/jnoc/PJE40-R