

Stay GUF1 with Performance Regression Testing

Skylar Hagen

Mentors: Dominic Manno and Jason Lee

What is GUFi?

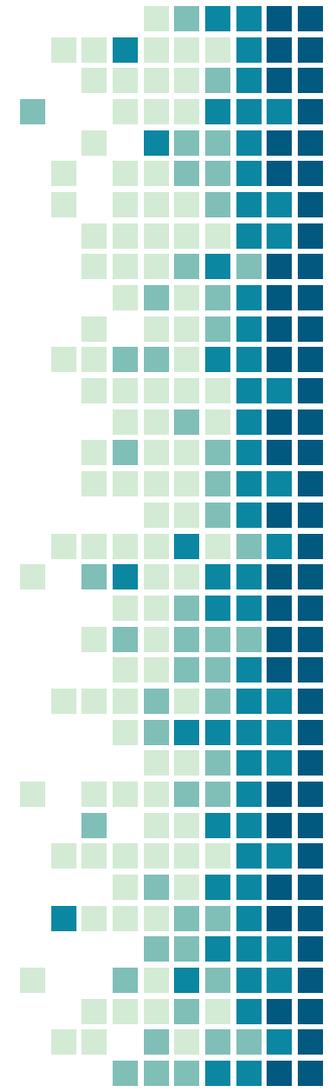
- Grand Unified File Index
- File system index tool
 - Query metadata efficiently on any HPC storage system
 - Secure
 - Outperforms other standard metadata tools
- <https://github.com/mar-file-system/GUFi>



<https://family.disney.com/articles/goofy-weekend/>

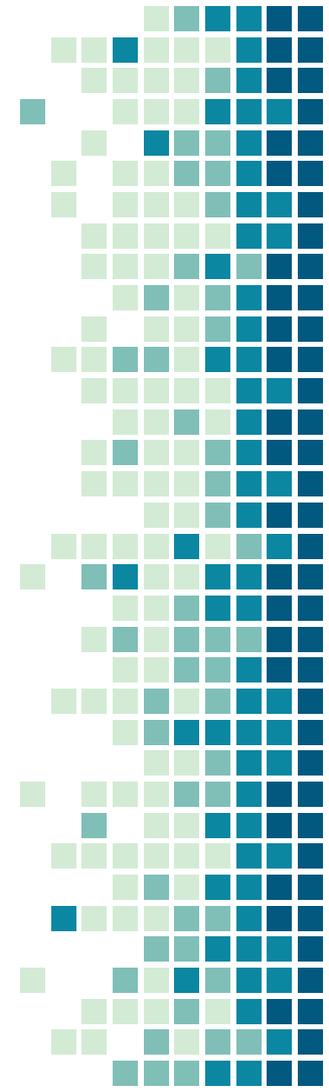
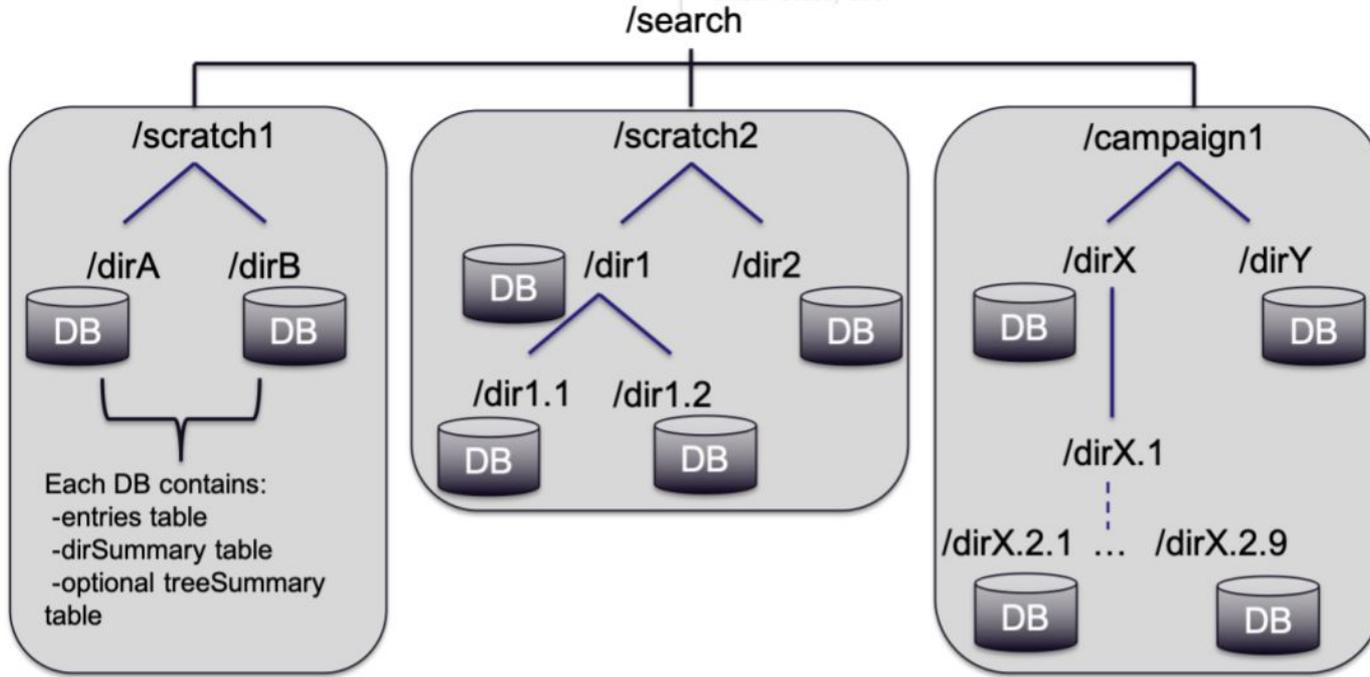
How does it work?

- Re-creates the source tree as an index
 - Index maintains same ownership and permissions
- Files are not copied over
- Directories have an embedded sqlite database that contains file and directory metadata



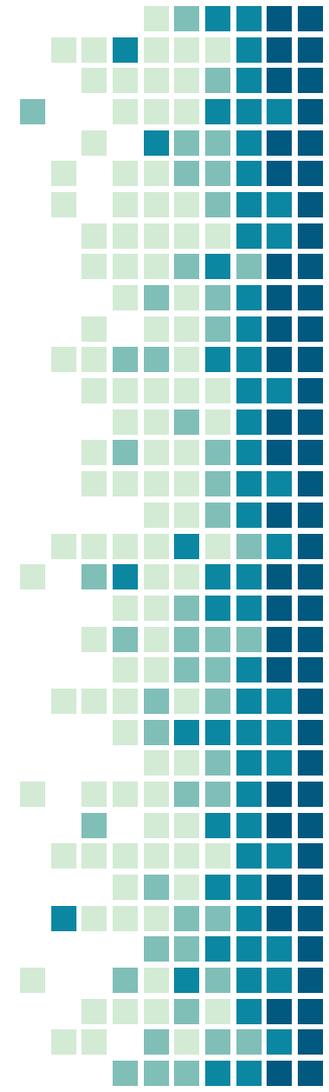
GUFI Design 2019

Simon Clark, LLN



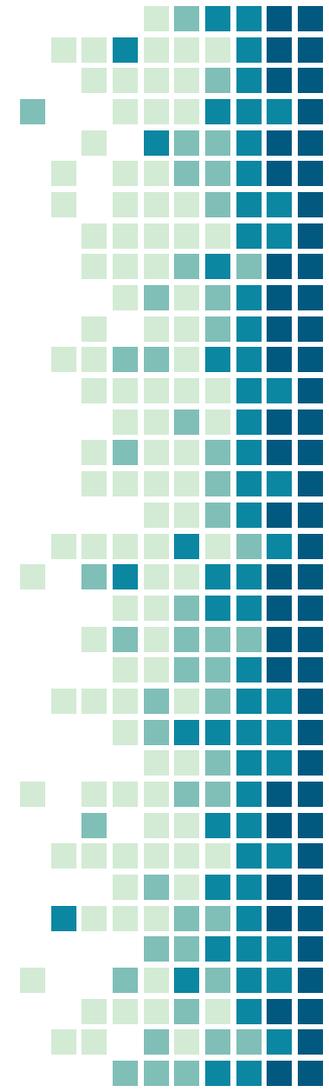
How does it work?

- Run queries to interact with the indexed metadata
- -S "Select * from summary" -E "Select * from entries"
index.gufi



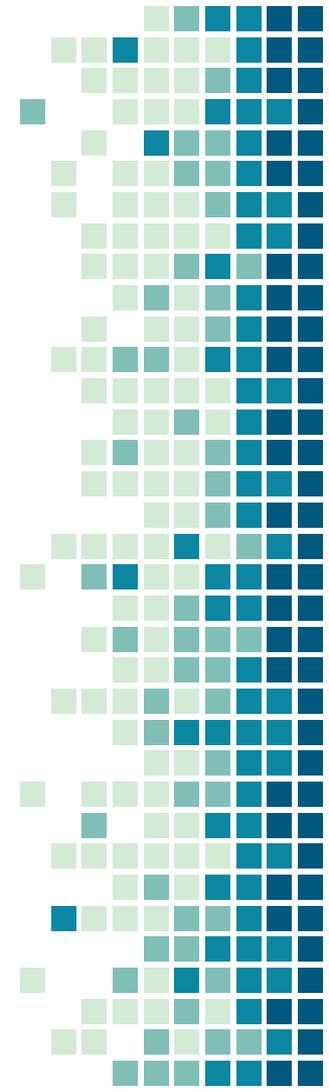
Problem

- GUF1 is efficient but a baseline for a standard set of operations has not been established
- System to track performance improvements and regressions
- The system must have a fixed set of
 - operations (queries)
 - hardware
 - load (trees + queries)



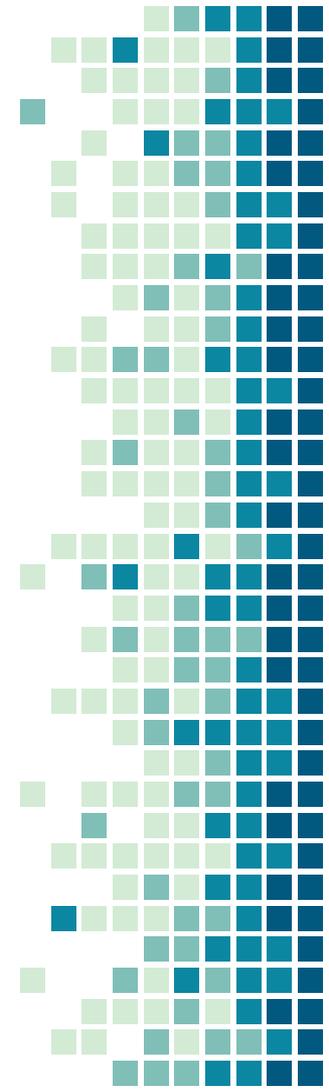
Tree Structures

- Generate trees with ideal size
 - 30 - 60 seconds to run most intensive query
- Deep Tree
 - 12 deep, 4 wide, 3 files per dir
- Wide Tree
 - 6 deep, 20 wide, 3 files per dir
- Balanced Tree
 - 6 deep, 6 wide, 500 files per dir



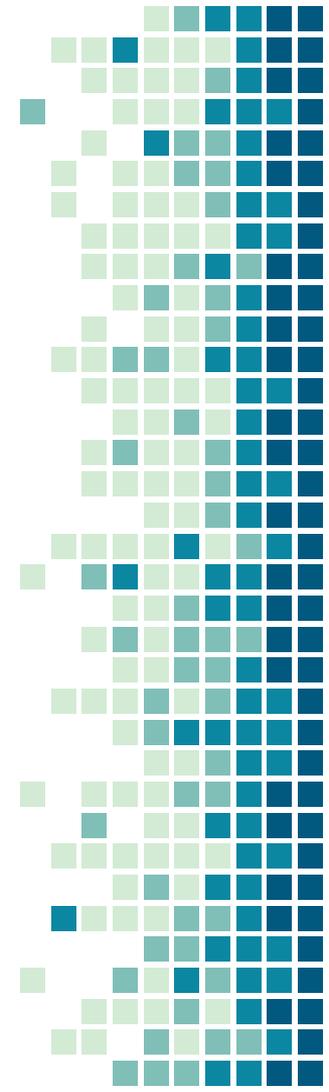
Performance Statistics

- These statistics are the sum of all the threads
 - Open Directories
 - Open Databases
 - Descend
 - Sqlsum
 - Sqlent
 - Close Databases
 - Close Directories
- Real Time - time main function took to run



System Design

- User calls `query_test` script on fixed set of hardware
 - History file
 - Average file
 - Pristine file
- Runs the query and dumps statistics into history file
- Stores averages into an average file with current commit hash
- Commit to pristine file



138.64
1556.00
47.95
0.00
49.57
21.90
4.47
30.00



101.14
1170.41
29.37
0.00
49.73
21.35
4.26
22.88



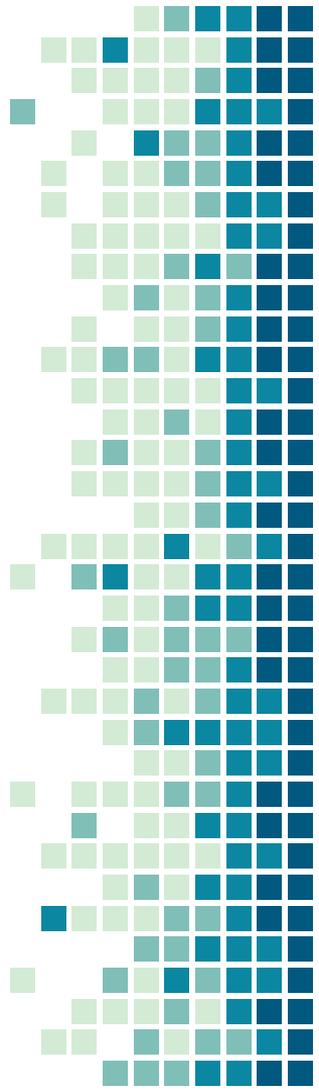
101.78
1180.65
28.03
0.00
47.23
21.35
4.39
23.02



99.70
1166.93
27.99
0.00
46.70
20.88
4.23
22.78

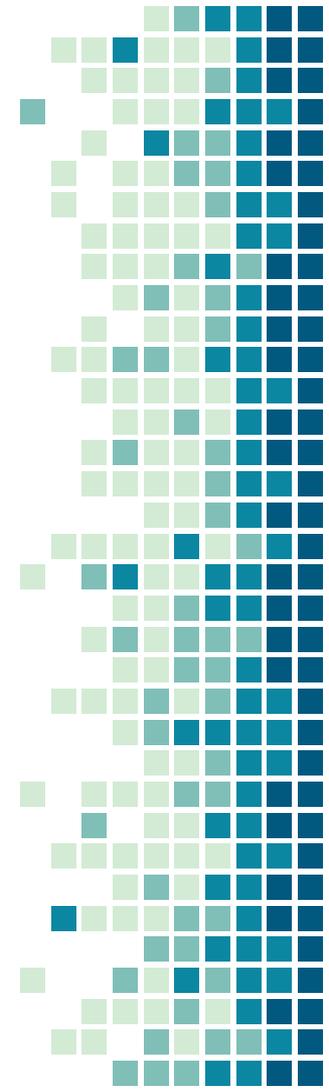


1767950
108.22999999999999
1248.9479999999999
32.458
0.0
48.67
21.386
4.33
24.32



System Design (Comparing)

- Compares stats between current commit and target commit (taken from the pristine file)
- Results of comparison printed to screen
 - Each stat marked with increased performance, regression, or no change
 - Allow variance (user specifies)



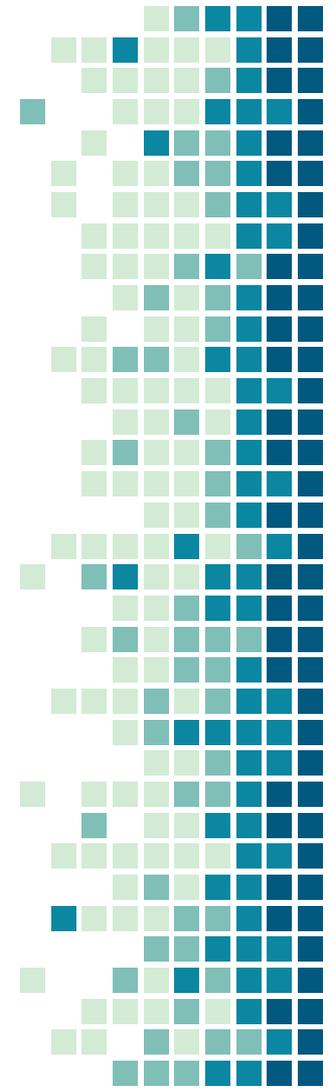
Average File

```
1767950  
99.85  
1171.35 ←  
29.47  
0.0  
49.02  
21.22  
4.24  
22.89
```

Pristine File

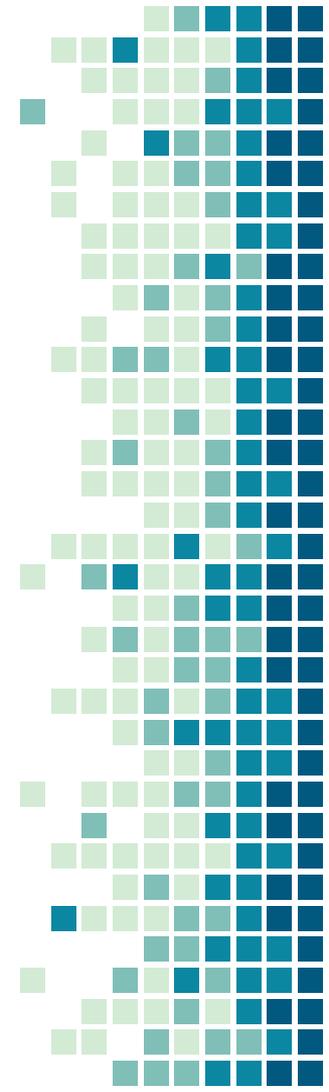
```
a538a75  
99.72  
1167.83 ←  
32.45  
0.0  
50.76  
20.8  
4.24  
22.88
```

```
open directories  
    No change in time  
  
open databases ←  
    Performance regressed by: 3.519999999999982s  
  
descend  
    Performance improved by: 2.9800000000000004s  
  
sqlsum  
    No change in time  
  
sqlent  
    Performance improved by: 1.7399999999999949s  
  
close databases  
    No change in time  
  
close directories  
    No change in time  
  
real time  
    No change in time
```



System Design (Pristine File)

- Commit good results to the pristine file
 - One entry from every commit hash
- Official history file



a13a330
94.34566666666669
1001.4400000000002
28.633000000000003
497.8136666666667
1213.5916666666667
24.40233333333335
4.62533333333334
46.312

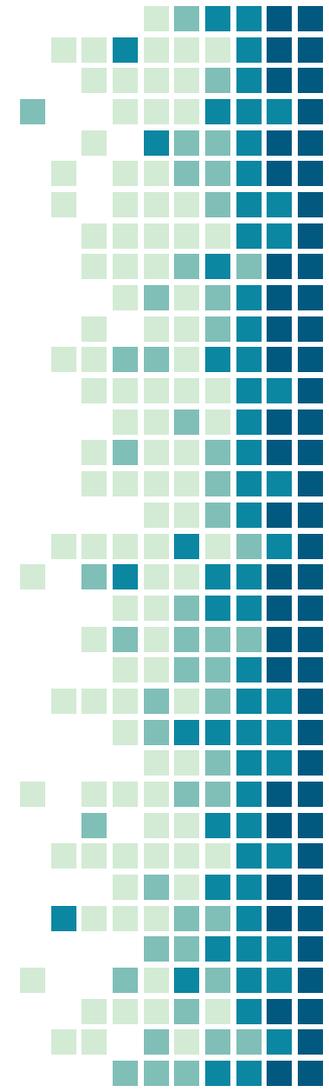
0cdab30
92.099
980.3523333333334
28.20433333333338
504.7606666666667
1240.8826666666662
24.41866666666666
4.566999999999999
46.44633333333332

b4d2883
92.04733333333334
981.2320000000001
28.24633333333334
503.7303333333336
1235.449
24.47733333333334
4.624999999999999
46.37466666666667

a538a75
91.63433333333334
973.0056666666667
28.18333333333337
509.2043333333333
1255.553333333335
24.44433333333336
4.577666666666667
46.62966666666667

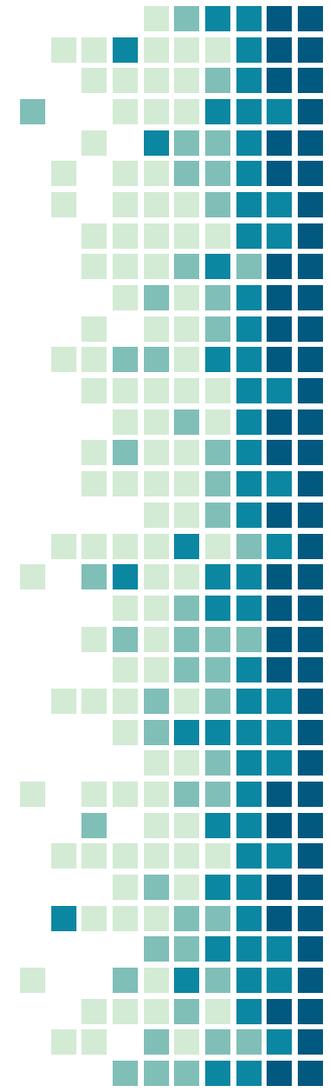


Most Recent



Conclusion

- Tracks and compares statistics between commits
- Detects a performance regression / improvement for each stat based on variance
- Stats vary in timings
 - Real time: 1s window
 - Open databases: 20s window
 - With more runs, we will pin down an acceptable variance value for each statistic



a13a330

94.34566666666669

1001.4400000000002 ←

28.633000000000003

497.8136666666667

1213.5916666666667

24.402333333333335

4.625333333333334

46.312 ←

0cdab30

92.099

980.3523333333334 ←

28.204333333333338

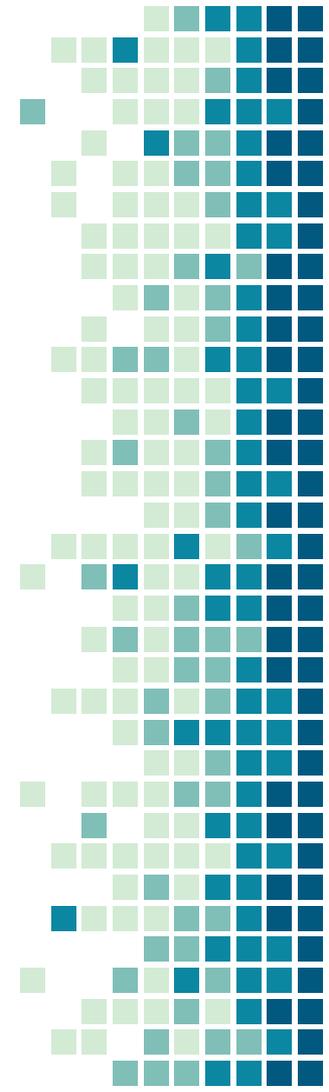
504.7606666666667

1240.8826666666662

24.418666666666666

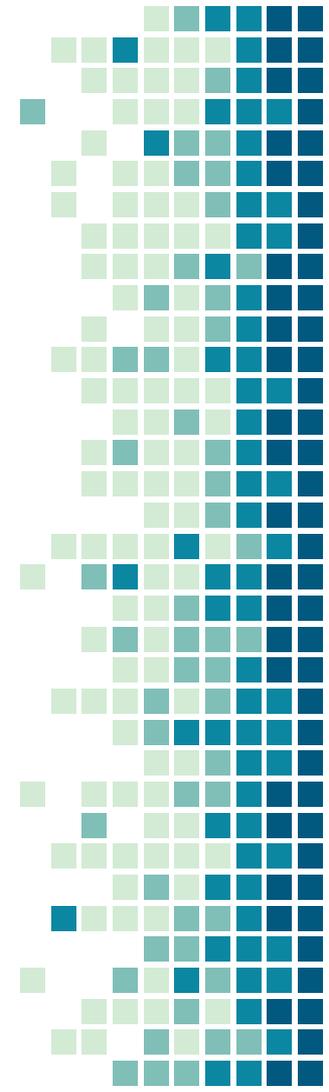
4.566999999999999

46.44633333333332 ←



Future

- Automate the testing process
 - CMake
 - Run on Travis-CI, Jenkins, Gitlab, etc.
- Move stat numbers to a database
- Additional tree structures
 - Different shapes beyond the ones listed
 - Compare results to our trees now
- Explore queries
- Explore variance





Contact / Information

Skylar Hagen

Email: Skylar.Hagen@trojans.dsu.edu

CompSci undergrad: Dec 2020

Master's: Jan 2021

