
Borg Collective: Filter TreeView, Live Binary Tests and Borgmatic Disaster Recovery

Personal Information

Name: Chirag Aggarwal

Email: REDACTED

Phone: +91 REDACTED

Website: <https://chirag.sh>

Location: REDACTED

Timezone: Indian Standard Time (IST) - UTC + 5:30

Profiles

Github: [jetchirag](#)

LinkedIn: [jetchirag](#)

Other Details

University: Manipal University Jaipur

Major: Computer Science and Engineering (AI & ML)

Year: 2nd year (as of March 2023)

About Me

I am pursuing a Bachelor of Technology degree, majoring in Computer Science and Engineering with a specialization in Artificial Intelligence and Machine Learning. My primary areas of interest include Python, Linux, and AI. I have experience in developing python applications for various purposes which I believe will be a great help in this project. I also have good experience with Linux and production hosting and want to start contributing to Open Source to improve my skills and give back to the community hence I'm participating in this initiative by Google.

I'm also a huge fan of hackathons and such opportunities. I was a finalist in Smart India Hackathon 2022 in which me and my team built a Smart Kiosk using Python that proved to be a success. I recently also won a Codeathon organized by IEEE CS.

Why this project?

I have set up several Linux servers and backups are always my utmost priority after configuration. I started using Borg around a year ago and I've never looked back. Moreover, Vorta has been perfect for my personal use case.

Hence, being able to contribute to a software so crucial for data safety is an opportunity. I'll also be able to learn from the experienced developers behind this project.

Code Contributions:

#	Description	Pull Request	Issue
1	[Vorta] Fixes math error for negative size in diff view in archive tab	#1651	#1650
2	[Vorta] FR: Sources Tab -> The settings for sorting by Path, Size, File Count should be retained when Vorta is restarted	#1649	#1647
3	[Vorta] Introduced Dialog to change borg passphrase for repositories	#1659	#303
4	[Vorta] Introduced password input widget	#1662	1659#issuecomment-1474039819
5	[Borgmatic] Separate endpoint for each primary action for monitor hooks	#660	#518
6	[Borgmatic] Borg 2 date-based matching flags for archive selection	#661	#659
7	[Vorta] Bug Fix: Diff causes critical error/crash on Borg 1.2.4	#1668	#1670

Project Details:

Sub-org name: Borg Collective

I am proposing three project ideas I'd like to complete during this program. I have carefully evaluated and ordered them based on timeline and priority.

- **Search a file in diff and extract view**

Easy **100 Hours**

Vorta has two dialogs which list the files either in comparison (comparing two archives) or listing all files mode (for extraction/restore). User has to manually traverse through the list to find the desired file or folder. Sometimes they may not remember the exact path and manually searching through a large number of items may take a long time.

Issue discussion: <https://github.com/borgbase/vorta/issues/1674>

Task Outline:

- Figure out all required and possible filtering options
 - Search function will allow users to filter items using their name or path.
 - Syntax for filtering to allow advanced search patterns (stretch goal)
 - Filtering by comparing file size (stretch goal)
 - In the comparison view, there will be a "Change" filter to filter only removed, added, modified files.
 - There are other search possibilities which can be introduced later on.
- Modify the GUI to add search options using the Qt UI file
- Add recursive filtering to display view based on enabled filters
- Write tests

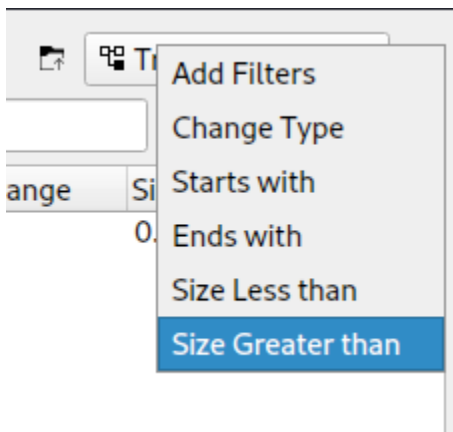
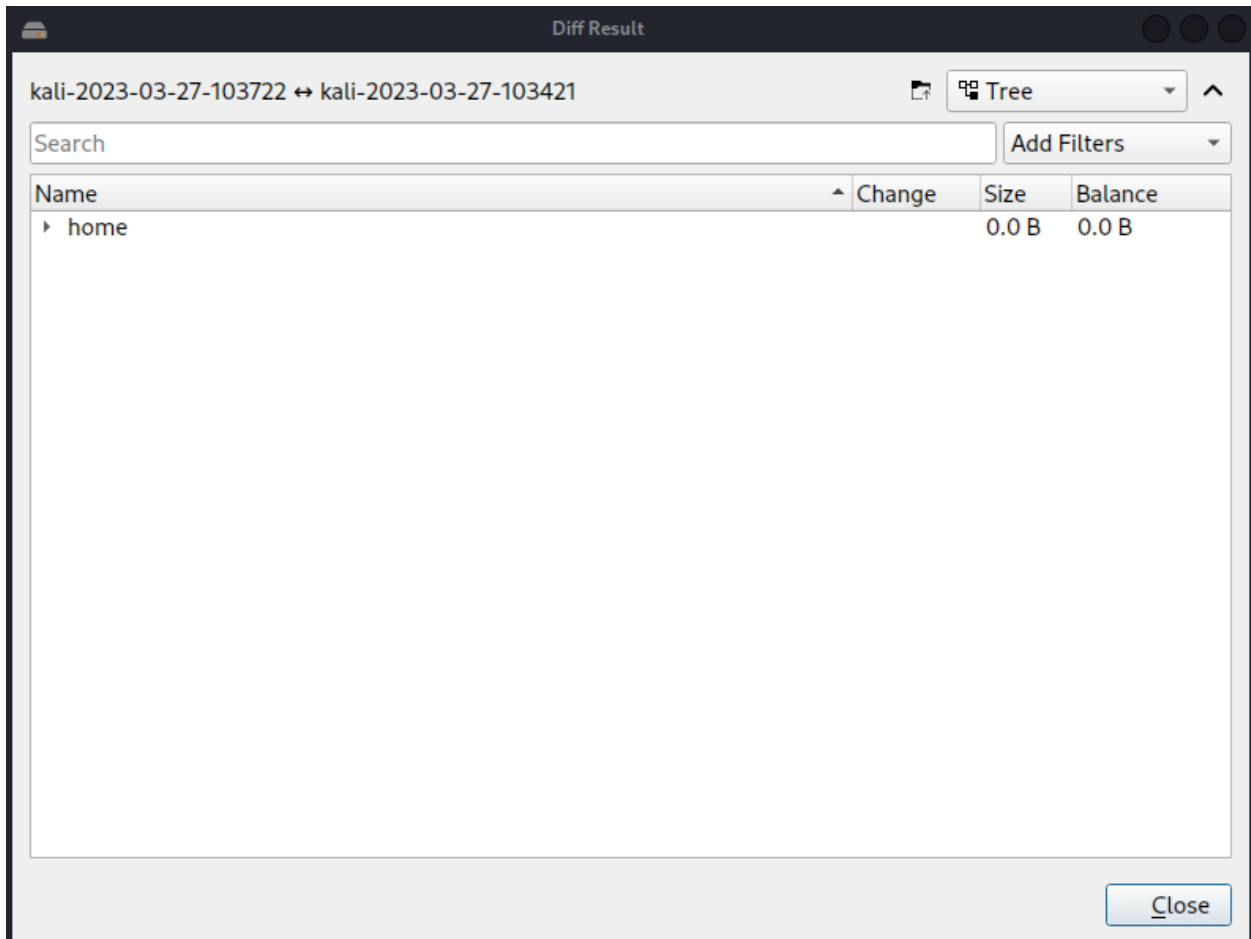
Deliverables:

- GUI modification to add search options
- Implementation of search functionality to filter files by name, path, and size
- Unit tests to ensure the functionality is working as expected.

Stretch Goals:

- Implementation of Size filters
- Search syntax for advanced search functionality

Proposed GUI:



GUI is subject to revisions through suggestions by the community.

- **In Vorta, Improve BorgBackup Testing by Implementing Live Binary Tests**

Medium 100-175 Hours

Currently Vorta contains stdout and stderr mock files for the Borg JSON output of various commands for testing purposes but there's no actual running of the commands. Moreover, it only supports one mock file while Vorta supports multiple Borg versions where output syntax may vary.

This project aims to improve Vorta Testing by implementing live binary tests that run on multiple actual Borg versions/binaries.

Task Outline:

- Research and understand the process of creating multiple environments with different Borg versions to run tests in. This might work differently in the CI (on the server) and local developer machines.
- Design and build a testing utility that can run the existing Vorta tests on multiple versions of Borg.
 - Download and install the specified Borg versions/binaries.
 - Run the Borg commands required for the Vorta tests on each installed Borg version/binary.
 - Compare the results of each Borg version/binary to ensure consistency and accuracy of the tests.
- Integrate the live binary tests with Vorta's CI/CD pipeline so that the tests are automatically run on each pull request and merge.
- Write comprehensive documentation that describes how to set up and run the live binary tests on local developer machines and CI servers.

Deliverables:

1. A testing utility that can run the existing Vorta tests on multiple versions of Borg.
2. Integration of the live binary tests with Vorta's CI/CD pipeline.
3. Comprehensive Documentation on Contribution page

- **Disaster Recovery: Bootstrap a borgmatic restore from nothing**

Medium 80 Hours

The goal of this project is to make restoration of backups generated through Borgmatic easier and more streamlined in case of a disaster or a complete system failure. It would include implementing a solution to store configuration in backup archives, a method to restore the archive using the stored configuration file and also an action to run this action and provide custom arguments.

Related Discussion: <https://projects.torsion.org/borgmatic-collective/borgmatic/issues/345>

Task Outline:

- Research and understand borgmatic's configuration file format and database backup and restore mechanisms.
- Develop a mechanism to store borgmatic configuration in a canonical location within the backup archive, making it available for restore.

-
- Design a way to store borgmatic database configuration along with each backed-up database dump to make it easier to restore. Otherwise, design a way to use the configuration file efficiently.
 - Add an action “bootstrap” to allow specifying a backup archive which will be searched for the restore configuration file and restored.
 - Test the restoration under various environments and configurations.
 - Update the documentation and tests to reflect the changes made including sample config file.

Deliverables:

1. A streamlined workflow for bootstrapping the restore of backed-up files and databases to a blank system.
2. Updated documentation and tests.
3. A "borgmatic bootstrap" action for borgmatic.

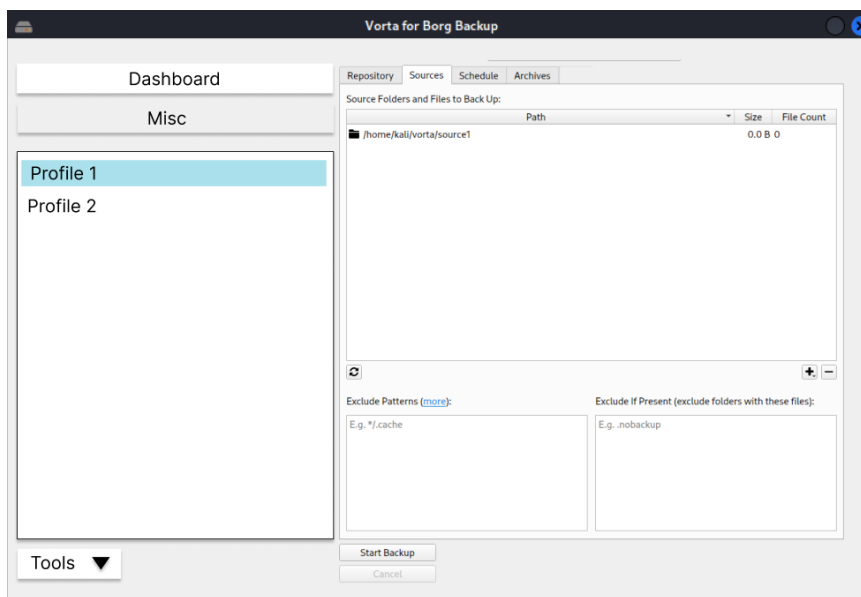
Other Project Ideas I'm interested in:

- [Vorta] Implement profile sidebar

Primary Goals:

- Move Misc to a different view outside of profile to use for global settings
- Implement a sidebar for easy profile selection, and modification with Tools options for actions like import and export.

Rough GUI Idea:

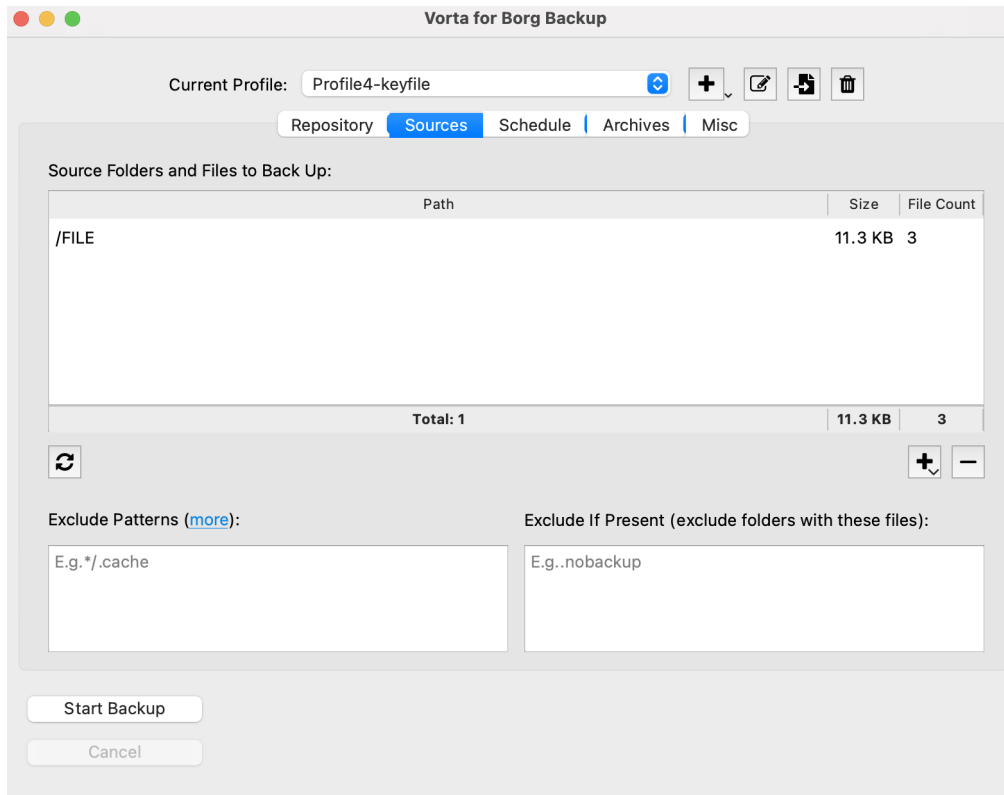


- [Vorta] Implement exclude GUI

This project aims at simplifying and extending functionality of the exclusion rules in Vorta.

- Pre-configured group of rules supporting multiple OS and common non-essential files and folders which user can select to add.
- A file browser integration to add path interactively
- Advanced mode to edit raw list of rules

- **[Vorta] Implement summary line for table in source view**



Above GUI implements a custom QWidget to handle two tables (for our main source list and a summary table which only utilizes the headers without rows of its own) aligned without any margin.

- **[Borgmatic] Restore a database backup to a different database or server**

This project would involve designing the configuration schema (partly designed) allowing setting custom values for database parameters used in restoration. It would unlock multiple restoration possibilities including different servers, databases, users etc.

- **[Vorta] Enhance archive table and archive actions**

- **[Vorta] Back up to multiple repos from one profile**

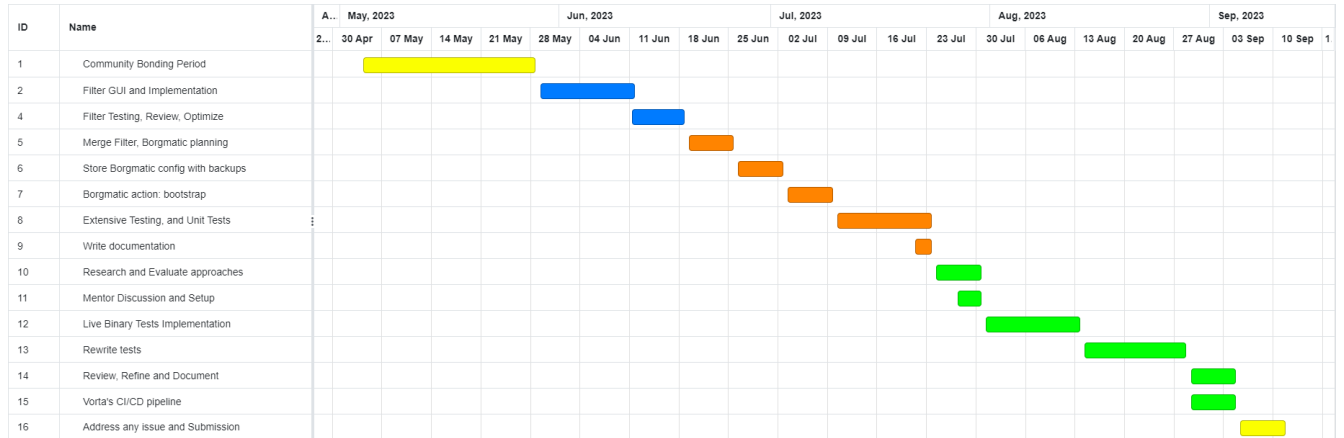
A common use-case for a lot of Vorta users is to backup their source to multiple destinations. Usually a hard-drive or remote location. However it's currently not possible to assign multiple repos per profile. We can implement this by creating a many-to-many relationship between repositories and profiles, adding multiple repos support to backup scheduler and other related changes. GUI would also require a multiselect or checkboxes dropdown.

Timeline:

- **Community Bonding Period (May 4 - May 28)**
 - Communicate with my mentors to understand the scope of the project
 - Get familiarized with the various features of Borg and how it works
 - Get familiarized with Tox, it's features and limitations
 - Gain more experience with PyQt
 - Understand the structure and working of various supported OS
 - [Borgmatic] Deep dive into the codebase and understand archival and restoration process
 - [Filtering TreeView] Create a mockup of search options in GUI
 - Search box for path match
 - Stretch Goal:
 - Mockup for implementing advanced search
 - Parsing advanced syntax and filtering based on other criteria
 - Write a rough overview and implementation plan of all projects
- **Week 1 (May 29 - June 4):**
 - Implement the filtering mockup in UI
 - Connect UI with treeview
 - Investigate methods to filter items in TreeView and test them
- **Week 2 (June 5 - June 11):**
 - Figure out best approach to filter items based on the parameters
 - Collaborate with the mentors to discuss the approach first
 - Implement filtering logic
- **Week 3 (June 11 - June 18):**
 - Test the feature on large data sizes and file counts
 - Figure out if we can optimize the approach
 - Finish up the design and project by taking reviews from mentors and resolving them
- **Week 4 (June 19 - June 25):**

-
- Write Unit Tests
 - Get PR merged
 - Start reviewing Borgmatic project and design configuration schema for disaster recovery
 - Extend the development environment for Borgmatic with virtual servers for various databases and OS to facilitate further development
-
- **Week 5 (June 26 - July 2):**
 - Discuss the approach for storing configuration with mentors
 - Develop a mechanism to store borgmatic configuration in a canonical location within the backup archive and borgmatic database configuration along with each backed-up database dump, if required.
 - Method to extract the information from a given archive file quickly and store it locally or consume directly to use borgmatic commands.
-
- **Week 6 (July 3 - July 9):**
 - Design the User Experience for “bootstrap” action, figure out required information, flags and rough implementation ideas.
 - Check-in with the mentors to review the approach before implementing and get approval
 - Create new bootstrap action as per discussed approach allowing easy restoration and disaster recovery
-
- **Week 7 (July 10 - July 16):**
 - Test the implemented action under different platforms and refine as needed
 - Test supported databases
 - Write tests
-
- **Week 8 (July 17 - July 23):**
 - Refine the implementation and documentation based on feedback from mentors and early adopters.
 - Finalize the implementation and documentation
-
- **Week 9 (July 24 - July 30):**
 - Research and understand the process of creating multiple environments with different Borg versions to run tests in.

-
- Evaluate existing tools and methods such as Tox to create the automation tools for testing purposes.
 - Discuss the final implementation idea with the mentors.
 - Set up a development environment on a local machine and virtual servers to test the implementation of live binary tests.
-
- **Week 10-11 (July 31 - August 13):**
 - Design and build a testing utility that can run the existing Vorta tests on multiple versions of Borg.
 - Implement a simple method to allow adding future versions.
-
- **Week 12 - 13 (August 14 - August 28):**
 - Method to run the Borg commands required for the Vorta tests on each configured Borg version/binary.
 - Rewrite all tests to replace mock files and tests with specified binaries.
 - Begin testing the utility on the local machine.
 - Compare the results of each Borg version/binary to ensure consistency and accuracy of the tests.
-
- **Week 14 (August 29 - September 4):**
 - Refine the implementation based on reviews from mentors.
 - Integrate the live binary tests with Vorta's CI/CD pipeline so that the tests are automatically run on each pull request and merge.
 - Write comprehensive documentation that describes how to set up and run the live binary tests on local developer machines and CI servers.
-
- **Week 14 (September 5 - September 11):**
 - Identify and address any issues or bugs that arise.
 - Push for code freeze
 - Ensure that all deliverables are complete.
 - Polish the project and make any final improvements or adjustments as needed. Proofread all documentations.



Gantt Chart for Proposed Timeline

Please note that the proposed timeline is subject to revision as I continue to learn more about the project and the specific work items. I will make sure to keep my mentor(s) informed of any changes and updates to the timeline and adjust my progress accordingly.

Other Commitments:

The Community Bonding Period starts from May 4th and I have my Academic Holidays from May 6th to July 30th will give me ample time to work on my projects. However starting from July 31st I will have my regular classes 4-5 times a week. I will be able to manage 4-5 hours per day on working days and more on weekends.

This is my first time participating in Google Summer of Code. I am only applying to this Organization and specifically only to Borg Collective Sub-Org.