GSoC 2023 Proposal: Wiki farm support for Canasta

Chenhao Liu

Personal Information

- **Name:** Chenhao Liu
- **Email:** chl178@ucsd.edu
- **GitHub:** https://github.com/chl178
- **Time Zone:** UTC−07:00
- **University:** University of California San Diego

Synopsis

Canasta is a Docker-based MediaWiki distribution designed to streamline the process of deploying a comprehensive MediaWiki instance on virtually any server. However, it currently does not support the operation of multiple wikis (also known as a wiki family or wiki farm) within a single container. This project’s objective is to introduce this absent feature and augment the Canasta command-line interface (CLI) for effectively managing individual wikis in a wiki farm environment. The multiple wikis within the same container will be differentiated by one of the following methods:

- Different directories (e.g., example.com/a, example.com/b)
- Different subdomains (e.g., a.example.com, b.example.com)
- Different domains for each wiki (e.g., example1.com, example2.com)

In addition to supporting multiple wikis in a single container, the Canasta CLI will be enhanced to include commands for creating, configuring, and deleting wikis in a wiki farm setup. The project will involve modifying the Canasta configuration and setup process, as well as making necessary changes to the web server configuration to handle different directories, subdomains, and domains for individual wikis.

This project aims to significantly improve Canasta’s functionality and flexibility, making it an even more powerful solution for hosting and managing MediaWiki instances.

**Mentors:** Yaron Koren & Jeffrey Wang

Deliverables

By the end of this GSoC project, the following deliverables are expected:

1. Support for running multiple wikis within the same Canasta container, with wikis differentiated by different directories, domains and subdomains.

2. A Canasta command-line interface (CLI) with the following features:
- Support for the wiki farm setup
- Commands to create, configure, delete, and manage individual wikis in a wiki farm

3. Modified Canasta configuration and setup process to support multiple wikis in a single container.

4. Necessary changes in the web server configuration to handle different directories, subdomains, and domains for individual wikis.

5. Comprehensive test cases to ensure the functionality and stability of the new features.

6. Updated Canasta documentation to reflect the new wiki farm capabilities and CLI options.

**Timeline**

- **May 4th - May 28th: Community Bonding Period**
  
  - **Further understanding Canasta and MediaWiki:** Study the Canasta codebase, MediaWiki core code, and relevant extensions. This will help me to identify the components that need to be modified or extended to support multiple wikis within a single container.
  
  - **Wiki Farm Management:** Investigate how other wiki farms are set up and managed, with a focus on addressing common challenges and incorporating best practices into the project design.
  
  - **Technology Familiarization:** Brush up on my knowledge of Docker, PHP, and web server configurations, as these technologies will be crucial in the implementation of the project.
  
  - **Collaboration with the Community:** Engage with the Canasta community and my mentors through mailing lists, chat platforms, and video calls. This will help me gather feedback on my project proposal, clarify any doubts, and establish strong working relationships with my mentors and other community members.
  
  - **Refining the Project Proposal:** Based on the feedback and insights gathered during the community bonding period, I will refine my project proposal, timeline, and deliverables to ensure the project aligns with the needs of the Canasta community and adheres to MediaWiki standards.

- **May 29th - June 4th: Design Architecture & Implement Wiki Differentiation**
  
  - Explore solutions for running multiple wikis within the same Docker container.
  
  - Investigate Apache web server configuration to handle different directories, subdomains, and domains for individual wikis.
  
  - Design a unified PHP codebase that can differentiate wikis by directories, subdomains, or domains.
  
  - Investigate methods for managing storage in wiki farm environments, including common resources and databases, and evaluate the most effective approaches for execution.

- **Jun 5th - Jun 11th: Implement Wiki Differentiation**
– Complete PHP code and Docker configuration for supporting wikis in different directories, subdomains, and domains.
– Test the differentiation functionality, ensuring compatibility with Docker, Apache, and PHP configurations.

• **Jun 12th - Jun 18th:** Implement Storage
  – Implement the chosen storage solution to support shared assets and tables for multiple wikis.
  – Test the storage solution to ensure proper functionality and compatibility with the existing Canasta stack.

• **Jun 19th - Jun 25th:** Refine Implementation & Address Issues while testing
  – Refine the PHP code and Docker configuration to ensure smooth user experience and maintainability.
  – Test the wiki differentiation implementation and storage solution together to identify any potential issues or bugs.
  – Address any issues or bugs found during testing and make any necessary adjustments to the code and configurations.

• **Jun 26th - Jul 2nd:** Modify Canasta Config & Web Server Config
  – Prepare for the midterm evaluation.
  – Modify the PHP code, Docker configuration, and Apache web server configuration to support the new multiple-wiki setup process.

• **Jul 3rd - Jul 9th:** Enhance Canasta CLI & Implement Wiki Commands
  – Design and implement the features for the Canasta CLI to support wiki farm setup.
  – Implement commands for creating and configuring individual wikis

• **Jul 10th - Jul 16th:** Implement Deletion & Management Commands; Write CLI Test Cases
  – Implement commands for deleting and managing individual wikis.
  – Write test cases for the new CLI features.

• **Jul 17th - Jul 23rd:** Test CLI & Setup; Address Issues
  – Perform thorough testing of the new features.
  – Address any issues or bugs discovered during testing.

• **Jul 24th - Jul 30th:** Polish Features; Write Comprehensive Test Cases
  – Refine the configurations for a smoother user experience.
  – Fix any remaining bugs or issues discovered during testing.
  – Start writing comprehensive test cases to ensure the functionality and stability of the new features, focusing on compatibility and integration between different parts.

• **Jul 31st - Aug 6th:** Complete Test Cases; Extensive Testing; Refactor Code
- Finish writing comprehensive test cases for the new features.
- Perform extensive testing of the project, ensuring compatibility and integration.
- Refactor code as needed to improve performance and maintainability.
- Address any issues or bugs discovered during the extensive testing phase.

- **Aug 7th - Aug 13th:** Update Documentation; Submit for Review
  - Update the Canasta documentation to include information on the new wiki farm capabilities, CLI options, and configurations for Docker, Apache, PHP, and Go.
  - Submit the code and documentation for review by the community and mentors.

- **Aug 14th - Aug 20th:** Address Feedback; Prepare for Final Submission
  - Incorporate feedback from the community and mentors, making any necessary adjustments to the code, configurations, and documentation.
  - Ensure the project meets the quality standards expected by the Canasta community and MediaWiki.
  - Prepare the project for final submission, including code, test cases, and documentation.

- **Aug 21st - Aug 28th:** Final week - Wrap Up; Submit Final Report
  - Complete any last-minute fixes or adjustments.
  - Submit the final report, summarizing the work completed during the GSoC project.

- **Sep 5th:** Initial results of Google Summer of Code 2023 announced.

**Goals**

I’ve divided the timeline into different stages, breaking down the whole task into smaller, manageable stages with specific goals. This will help maintain a steady pace and make it easier to track progress.

**Part 1: Design, Familiarization, and Implementation (May 4th - June 25th)**

1st Goal: Establish a strong foundation in the project’s technologies, design a flexible architecture, and implement the wiki differentiation and storage solution.

**Part 2: Enhancements, Configuration, and Testing (June 26th - July 30th)**

2nd Goal: Modify Canasta’s configuration and setup process, enhance the Canasta CLI, and conduct comprehensive testing to ensure a smooth user experience.

**Part 3: Documentation, Review, and Final Submission (July 31st - August 28th)**

3rd Goal: Complete the test cases, refine the code, update the documentation, and prepare the project for final submission and evaluation.
About Me

Current Education

I am currently pursuing a Master of Science (M.S.) in Computer Science at University of California San Diego (UCSD) with an estimated graduation date in 2024.

How did you hear about this program?

During the winter quarter, I began taking a network programming course. A classmate recommended the GSoC to me on Piazza, as the GSoC is an excellent platform for enhancing one’s professional skills.

Will you have any other time commitments, such as school work, another job, planned vacation, etc, during the duration of the program?

If I am accepted for GSoC, I will be working from the United States this summer. At the moment, I don’t have any other plans, such as internships or vacations, and Canasta is the only GSoC project I am applying for. I am committed to making GSoC my first priority this summer.

We advise all candidates eligible for Google Summer of Code and Outreachy to apply for both programs. Are you planning to apply to both programs and, if so, with what organization(s)

Yes, I am eligible for both the programs, but have only applied to GSoC.

What does making this project happen mean to you?

Participating in Google Summer of Code (GSOC) for the first time holds significant meaning and value to me for several reasons. By making this project happen, I will be able to achieve personal and professional growth, contribute to the open-source community, expand my network within the tech industry and boost my confidence.

Experience

<table>
<thead>
<tr>
<th>Pull Request</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Added propagating verbose to subcommands support #72</td>
<td>Completed</td>
</tr>
<tr>
<td>Provided Real-Time Output for External Subcommands #75</td>
<td>Completed</td>
</tr>
<tr>
<td>Added ”canasta elasticsearch index” command #76</td>
<td>Completed</td>
</tr>
<tr>
<td>Added maintenance-scripts directory #228</td>
<td>Working</td>
</tr>
<tr>
<td>Added option to choose version to install #77</td>
<td>Completed</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enhancement for Issue #61: Provide Real-Time Output for External Subcommands #73</td>
</tr>
</tbody>
</table>
Describe any relevant projects that you’ve worked on previously and what knowledge you gained from working on them.

I have implemented a cloud-based file storage application called SurfStore, which is based on Dropbox and the RAFT protocol, allowing users to sync files to and from the cloud and utilizes distributed consensus to ensure the safety and availability of data.
I have also developed a web server that implemented a subset of the HTTP/1.1 protocol specification.

Describe any open source projects you have contributed to as a user and contributor (include links).

The exposure to the open-source community has been a remarkable strength in my development journey since I studied in the United States, providing me with valuable insights into various technologies, such as Go, Docker, and Apache. Canasta, a complex and well-structured project, was the first open-source project I contributed to. And I have made a few contributions as mentioned above in the section.

Any Other Info

The proposal is also publicly published on Phabricator.