Caliper GUI Design[1]

Content

Project Goal And Scope
- Project Expectations
- Key metrics from Caliper application
- Integrating methodology
- Tools
- The Goal-directed Design
  - GDD Questions
  - GDD Product Design Concepts

Deliverables

Milestones And Evaluation Criteria
- Project Timeline and Milestones
- Project Evaluation Criteria
  - Web GUI design
  - Storytelling with Data

Personas of Caliper GUI
- Persona #1

Next Steps

[1]June 4, 2019 created by Jason You

Project Goal And Scope

Project Expectations

Building a GUI for Hyperledger Caliper that can:

- Visualizing benchmarks/performances in real-time;
- Showing clear documentation for Caliper GUI users;
- Assemble and generate configuration files;
- Saving, uploading, and editing the Hyperledger framework configuration files.

Key metrics from Caliper application

- A1
    - Transaction/Read Latency
        - Minimum
        - Maximum
        - Average
        - Percentile
    - Transaction/Read Throughput
        - Minimum
        - Maximum
        - Average
        - Percentile
    - Success rate
        - Success count
        - Fail count
        - Success Ratio
        - Fail Ratio
- A2
    - Resource consumption
        - CPU
        - Memory
        - Network I/O

Integrating methodology

- First determining the main metrics for Hyperledger Blockchain performance visualization.
- Returning JSON formatted data from Caliper-core or Caliper-CLI to Caliper GUI real-time visualization.

Tools

- **Node.js:** formatting and acquiring visualization data from Hyperledger Caliper.
- **D3.js** tools for visualizations:
    - Arbor.js (graph visualization for network relationship)
    - Plot.ly (interactive visualization tool based on D3.js)
        - Plot.ly JSON Chart Schema
        - Bar Chart with Direct Labels
        - Customizing Individual Bar Colors
        - Categorical Dot Plot
        - Circular Gauge Chart (For performance visualization)
        - WebGL With 100000 Points (For dense data analysis)
        - Line and Scatter Plot
        - Bubble Size Scaling on Charts
        - Sunburst with Repeated Labels
        - Table (Alternating Row Colors)
- **React.js**: For interactive GUI development.
    - React Bootstrap
    - React Plot.ly
- Basic front-end development languages and tools (**HTML/CSS/JavaScript/jQuery/Bootstrap 4**)

The Goal-directed Design

GDD Questions

- How will my product help infrequent and inexperienced users understand how to accomplish their goals?
    - For infrequent and inexperienced users, this GUI can help them to accomplish network testing without worrying about integration with existing networks, and it provides easy switching between test networks.
    - It also provides real time metrics visualization with different charts, such as bar charts, time-series, dot plots, etc.
    - This GUI should abstract away connection step when a Hyperledger Blockchain network is already built. (Or at least provide very intuitive visual guidelines for setup)
- Who are my users?
    - Business managers that want to know the performance of their blockchain networks.
    - Blockchain developers that need to present the performance metrics to product managers.
    - Infrequent users of Hyperledger Blockchain Applications that want to do analysis on the underlying network.
- What are my users trying to accomplish?
    - Comparing the performance of blockchain with other existing solutions (such as TPS comparing with other applications that haven't deploy Hyperledger blockchain).
    - Get beautiful visualizations to convince other people to use or invest into their Hyperledger Blockchain applications.
- How do my users think about what they're trying to accomplish?
    - They may find testing tedious and complicated, thus we will need to make it fun and simple.
    - They might just want visualization or they might need accurate JSON output for later usage. (They may even want to have jpg, png, html options for report output)
- What kind of experiences do my users find appealing and rewarding?
    - Inexperienced users might just want to check the result without dealing with bugs in configuration of different frameworks and networks (such as Fabric). Thus simple setup by a few clicks is a must for this kind of users.
    - Experienced users will hope to get accurate and intuitive visualizations with flexible real time demonstration. (We can also provide the JSON format metrics output if they request).
- How will users interact with my product?

    - Need to have Caliper-GUI volunteers to try our MVP.
- How can my product's functions be most effectively organized?
    - TBD
- How will my product introduce itself to first-time users?
    - TBD
- How can my product put an understandable, appealing, and controllable face on technology?
    - TBD
- How can my product provide sufficient depth and power for expert users?
    - TBD

GDD Product Design Concepts

- Research
    - User interview and observations
        - Understanding user needs and behavior
- Modeling
    - Personas
        - User and customer archetypes
        - Patterns in user and customer behaviors, attitudes, aptitudes, goals, environments, tools, challenges
    - Other models
        - Beyond individual users.
        - Workflows among multiple people, environments, artifacts.
- Requirement Definitions
    - Content scenarios
        - Tell stories about ideal user experiences.
        - How the product fits into the persona's life and environment, and how it helps them achieve their goals.
    - Requirements
        - Describe necessary capabilities of the product
        - Functional and data needs, user mental models, design imperatives, product vision, business requirements, technology
- Design Framework
    - Elements
        - Define manifestations of information and functionality
            - Information, functions, mechanisms, actions, domain object models
    - Frameworks
        - Design overall structure of user experience
            - Object relationships, conceptual groupings, navigation sequencing, principles and patterns, flow, sketches, storyboards

        - Key Path and Validation Scenarios
            - Describe how the personna interacts with the product.
                - How the design fits into an ideal sequence of user behavior, and accommodates a variety of likely conditions
- Design refinement
    - Refine specific details
        - Appearance, idioms, interface, widgets, behavior, information, visualization, brand, experience, language, storyboards
- Design support
    - Accommodate new constraints (technological) and timeline/deadline
        - Maintaining conceptual integrity of the design under changing technology constraints.

Deliverables

- **Phase 1: Starting the Caliper GUI project**
    - Basic Web UI design ideas;
    - Basic visualization charts for metrics;
    - Network graph visualization;
    - Consented JSON format input for Caliper visualization GUI.
- **Phase 2: Designing the initial model for the Caliper GUI**
    - Interactive UI with React.js;
    - Real-time visualization functionalities.
- **Phase 3: Integrating the visualization with the Caliper GUI**
    - Integration scripts for Caliper-core, Caliper-CLI, and Caliper GUI.
    - (TBD) Helper functions for integration with Node.js.
- **Phase 4: Refining project**
    - Documentation for integration and usage of Caliper GUI.
    - (TBD) Configuration files upload functionalities for framework testing with Hyperledger Caliper GUI:
        - Adding functionalities that allow users to easily test their results by dragging configuration files of different Hyperledger framework networks in to the Caliper GUI, and get visualization and reports by a few simple clicks.
- **Phase 5: Finalizing and summary**
    - Presentation PowerPoints for Caliper GUI project.
    - Demonstration for the functionality of Caliper GUI.

Milestones And Evaluation Criteria

Project Timeline and Milestones

- **Project Start** [June 3rd, 2019]
    - Deciding the visualization metrics and designing basics statics visualizations with Plot.ly and D3.js.
    - Designing the basic Web UI for Caliper.
    - (TBD) Graph visualization for network relationship
- **1st Quarter** [June 24th, 2019]
    - Using React.js to build the interactive UI.
    - Implementing the real-time visualization functionality.
- **2nd Quarter** [July 15th, 2019]
    - Integrating the Caliper-core or Caliper-CLI with the developed GUI and visualization functionalities.
    - Adding helper functions with Node.js.
- **3rd Quarter** [August 5th, 2019]
    - Creating and editing documentations for Caliper GUI for new users.
    - Refining the GUI and adding additional functionality to facilitate testing configuration for different Hyperledger frameworks.
- **Final** [August 26th, 2019]
    - Preparing project presentation and GUI demonstration.

Project Evaluation Criteria

Web GUI design

- Goal:
    - User friendliness:
        - The metrics visualization should be in real time and should not require too much integration tasks after the launching of any Hyperledger network.
        - Easing the configuration and launching of Caliper. For example, even user without any blockchain technical background can easily build a test network with a few clicks.
        - Provide tutorials that includes everything new users needed to run a test for Hyperledger frameworks in major development environments.
    - Compatibility:
        - Compatible with future updates of Caliper core (modularize each functionality in the GUI and visualization implementation from the very beginning).

        - Making sure the the GUI functions and visualizations are mostly generic for all Hyperledger Blockchain frameworks.

Storytelling with Data

- Understanding the context
    - What background information is relevant or essential?
        - Knowing how to configure the networks with different Hyperledger frameworks that need to be tested.
        - Knowing the functionalities of different configuration files, and knowing which ones to upload to the Caliper CLI/GUI for testing purposes.
    - Who is the audience or decision maker? What do we know about them?
        - Hyperledger developers that need to test their project network.
        - Hyperledger source code contributors who need to know the framework performance.
        - Blockchain application project managers.
        - Blockchain performance and scalability researchers.
        - Blockchain application users that want to know the performance of their application.
        - Blockchain startup investors who want to know the performance of their targeted Blockchain startup products.
    - What biases does our audience have that might make them supportive of or resistant to our message?
        - They maybe analogue Blockchain application performance/metrics with those that based on traditional database architecture, which might not be the same.
    - What would a successful outcome look like?
        - Users with any level of technological backgrounds in Blockchain can easily interpret the visualizations and reports provided in the Caliper GUI.
    - If you only had a limited amount of time or a single sentence to tell your audience what they need to know, what would you say?
        - Hyperledger Caliper GUI visualization can efficiently help developers and users to understand the level of efficiency of the network, and **straightforwardly find the places that need to be optimized**.
    - Storyboard
        - Issues:
            - Hyperledger blockchain developers only have vague understanding to the performance and scalability of their network.
        - Demonstrating issues:
            - Currently there is no efficient way to visualize or measure the performance of multiple network with Hyperledger.
        - Ideas of overcoming issue:

            - Showing the performance benchmark on existing networks and give intuitive references (visually) to help developers and blockchain users to understand the network performance.
        - Describe our Caliper GUI:
            - Caliper GUI provides html reports based on Caliper CLI, and demonstrates precise performance visualizations based on performance metrics.
        - Recommandation:
            - Adding simple test configuration functionalities in the Caliper GUI, and let all Hyperledger users adopting this tool.
- Effective Visualization
    - What is the best strategy to demonstrate our performance metrics outputs?
        - **Rule: choose whatever will be easiest for the audience to read.**
        - Simple Text
            - Using this when only one or two numbers need to be communicated.
            - Exaggerating the number is helpful.
        - Scatter Plot
        - Table
            - Don't use boarder: The data should be what stands out, not the borders.
        - Line
        - Heatmap
            - Helping you to find the higher and lower values in tables.
        - Slopegraph
        - (Stacked) Vertical/Horizontal Bar
            - Good for categorical data showing.
            - It is important that bar charts always have a zero baseline.
            - 100% stacked horizontal bar can provide easy comparison for far left and far right portion of the data.
        - Waterfall
            - Great to demonstrate increments and decrements from start to end status.
        - Square area
        - Interactive or non-interactive
        - Time series

Personas of Caliper GUI

Persona #1

- Name: Jonathan
- Job Title: Blockchain developer
- Where he works: remote
- Age: 26
- Gender: Male
- Location: Seattle, US
- Education: Computer Science, MS
- Primary Value:
    - Building reliable blockchain application and deploy them on his web applications. Still exploring the functionalities of different frameworks in Hyperledger project.
    - Learning experience matters and don't want to waste time on error prone documentations.
- Goal of using Caliper:
    - Storytelling with data to shareholders and his web application users.
    - Better understanding of performance and scalability (need intuitive explain about how scalable the blockchain is right now)
    - Using accurate and beautiful visualizations as marketing tools to get interests from potential web app users and investors.

Next Steps

- **Building**
    - Start to build as soon as we have a clear Goal-directed design road map and UI interface skeleton.
- **Testing**
    - Hopefully we will have time to test multiple times with volunteer users before we give the first release.
- **Release**
    - V0.1.0 of Caliper GUI.