SOFTWARE REQUIREMENTS SPECIFICATION

for

<Project>

Version 1.0 approved

Prepared by <author>

<Organization>

April 7, 2018

Contents

1 Introduction **5**
- 1.1 Purpose . . . 5
- 1.2 Project Scope . . . 5
- 1.3 References . . . 5

2 Overall Description **6**
- 2.1 Product Perspective . . . 6
- 2.2 Product Functions . . . 6
- 2.3 User Classes and Characteristics . . . 7
- 2.4 Operating Environment . . . 7
- 2.5 Design and Implementation Constraints . . . 7
- 2.6 User Documentation . . . 7
- 2.7 Assumptions and Dependencies . . . 7

3 External Interface Requirements **8**
- 3.1 User Interfaces . . . 8
- 3.2 Hardware Interfaces . . . 8
- 3.3 Software Interfaces . . . 8
- 3.4 Communications Interfaces . . . 8

4 System Features **10**
- 4.1 System Feature 1 . . . 10
  - 4.1.1 Description and Priority . . . 10
  - 4.1.2 Functional Requirements . . . 10
- 4.2 System Feature 2 . . . 10
  - 4.2.1 Description and Priority . . . 10
  - 4.2.2 Functional Requirements . . . 11
- 4.3 System Feature 3 . . . 11
  - 4.3.1 Description and Priority . . . 11
  - 4.3.2 Functional Requirements . . . 11
- 4.4 System Feature 4 . . . 11
  - 4.4.1 Description and Priority . . . 11
  - 4.4.2 Functional Requirements . . . 12
- 4.5 System Feature 5 . . . 12
  - 4.5.1 Description and Priority . . . 12
  - 4.5.2 Functional Requirements . . . 12

5 Other Nonfunctional Requirements **13**
5.1 Performance Requirements 13
5.2 Safety Requirements 13
5.3 Security Requirements 13
5.4 Software Quality Attributes 13
5.5 Business Rules 14

6 Other Requirements **15**
6.1 Appendix A: Glossary 15
6.2 Appendix B: Analysis Models 15
6.3 Appendix C: To Be Determined List 15

Revision History

Name	Date	Reason For Changes	Version
21	22	23	24
31	32	33	34

1 Introduction

1.1 Purpose

This system is designed to interface with SPECjbb 2015 and make it a more user friendly. The system will provide users with the ability to save an store various run settings for use with SPECjbb 2015.

1.2 Project Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

1.3 References

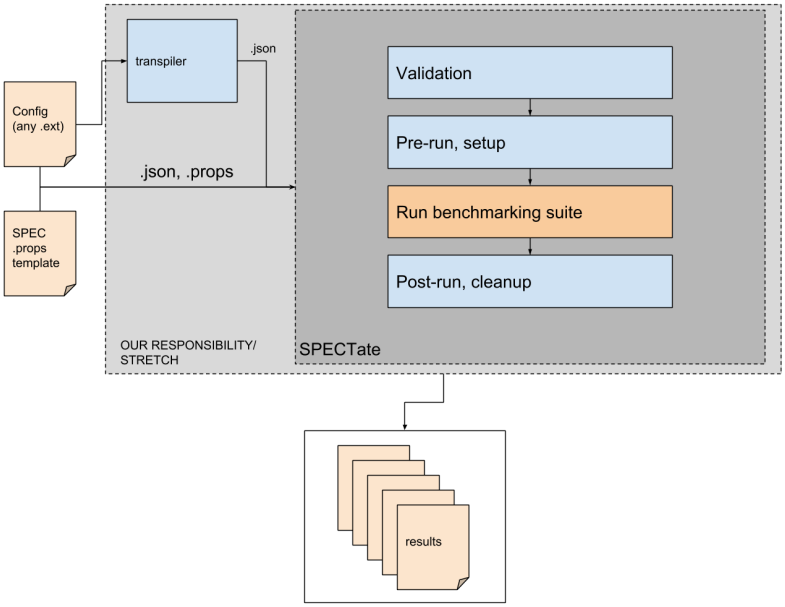
1. SPECjbb2015
2. Documentation

2 Overall Description

2.1 Product Perspective

This product is a new way to manage configuration files, run SPECjbb 2015, and organize output results. It is specified for benchmark runs on SPECjbb 2015.

Figure 2.1: Interface Architecture Diagram



2.2 Product Functions

In general, the product puts focus on a user-friendly way to use SPECjbb 2015, which includes:

- Allows queuing multiple SPECjbb jobs to run sequentially with different configuration.
- Allows users to create and save a new configuration set or load/modify/delete existed sets.
- Allows users to input custom configuration information for SPECjbb.

- Provides options/configuration validation.

2.3 User Classes and Characteristics

The product can be found useful to those who are interested in running benchmarks on SPECjbb 2015 in general. More specifically, who are concerned with performance of specific configurations of hardware likely to use SPECjbb more often and the product is necessary. Besides, students and professors from universities or researchers want to consider using SPECjbb in their research is one of important user classes for this product.

2.4 Operating Environment

The application will always run under a Unix environment. Users may run the application remotely without access to a graphical user interface.

2.5 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

2.6 User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

SPECjbb2015 Benchmark User Guide: https://www.spec.org/jbb2015/docs/userguide.pdf

2.7 Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues And stillaround the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

3 External Interface Requirements

3.1 User Interfaces

The product focuses on the Command Line Interface (CLI) since potential users of the product are usually familiar to Unix operating systems. Further in the process, we would have an alternative solution for nicely designed GUI. Through the CLI, users are able to manipulate configuration sets used for benchmark runs on SPECjbb 2015. The manipulation actions include all the functions related to configuration sets mentioned in Section 2.2.

3.2 Hardware Interfaces

The product performs as a helper for users of SPECjbb 2015 in the processes of preparing configuration sets and organizing results afterward. Thus, the product has no hardware interfaces directly but through SPECjbb if there is any.

3.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

The product will require Python 3 installed on your computers with no further required libraries. The configuration sets will be saved in JSON format files. After parsing configuration sets into parameters, the product will use these parameters, the shared data, as components to run SPECjbb 2015. After SPECjbb 2015 finishes each run, the product will perform the cleanup for the result of the run.

3.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols,

electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

4 System Features

4.1 System Feature 1

Option Sets

4.1.1 Description and Priority

Option Sets include all configuration information selected by the user for a single run of SPECjbb 2015.

4.1.2 Functional Requirements

1. The system shall allow the user to select any valid JVM on their machine. ($JAVA)
2. The system shall validate user provided JVM arguments prior to running SPEC. ($JAVAOPTS)
3. The system shall allow the user to select a run type of HBIR, HBIR_RT, PRESET, or LOADLEVEL
4. The system shall allow the user to select the number of runs to be executed.
5. The system shall allow the user to enable basic data collection. (vmstat -nt 1 > $RUN_NUM.vmstat.log &)
6. The system shall allow the user to select various SPECjbb options.
7. The system shall allow the user to input a run identifier ($TAG)
8. The system shall allow the user to input the number of Nodes.

4.2 System Feature 2

Option Set execution

4.2.1 Description and Priority

Each Option Set will be executed sequentially.

4.2.2 Functional Requirements

1. The system shall inform the user which group and run number is currently executing.
2. The system shall log all information into SUT.txt
3. The system shall create a results directory (named $TAG)
4. Tge system shall create the configuration file 'specjbb2015.props'
5. The system shall place the configuration file in the SPEC configuration directory (RC3/config/) prior to running SPEC.
6. The system shall place a copy of the configuration file in the results directory.
7. The system shall start the SPEC Controller. ($JAVA $JAVAOPTS -jar
8. The system shall execute each Node appropriately.
9. The system shall end all started processes once the Controller exits.
10. The system shall copy all results into the Results Directory (cp result/specjbb2015*/report*/*.html .)

4.3 System Feature 3

Option Set Node execution

4.3.1 Description and Priority

Users may select each Option Set to be run on a number of different nodes, which must be handled accordingly.

4.3.2 Functional Requirements

1. The system shall inform the user which Node number is being started.
2. The system shall execute the TxInjector.
3. The syste m shall execute Backend

4.4 System Feature 4

Save, Create, Delete, and Load Option Sets

4.4.1 Description and Priority

The user needs to be create, save, delete, and load Option Sets for quick execution.

4.4.2 Functional Requirements

1. Create Option Set
2. Load Option Set
3. Delete Option Set
4. Save Option Set

4.5 System Feature 5

User Interface

4.5.1 Description and Priority

The user needs to be able to visualize and organize Option Sets prior to running benchmarks.

4.5.2 Functional Requirements

1. Users need to be able to view available Option Sets
2. Users need to be able to arrange Option Sets in the order to be executed.
3. Users need to be able to indicate that they are ready to being benchmarking.

5 Other Nonfunctional Requirements

5.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

5.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

5.3 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

5.4 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

5.5 Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

6 Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

6.1 Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

6.2 Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

6.3 Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>