Vision and Scope

101 Solutions

June 23, 2013

Contents

1 Architecture requirements

1.1 Architectural scope

- Providing infrastructure for copying, synchronizing and managinge files over multiple computers from one master

1.2 Quality requirements

1.3 Integration and access channel requirements

Our system must interact with Subversion (SVN), which is where the clients store the various builds of their applications. We will also be connecting to a MySql database where the information concerning confiugurations on slave pc's will be stored.

- We can use the SVN client library to connect to Subversion.
- Connector/C++ 1.1.3 is available to connect to MySql for both Windows and Linux.

The system will be accessible to users via a Graphical User Interface as specified by the client.

1.4 Architectural constraints

Overall constraints include the use of QT 4.8.4 with C++ throughout the project. The use of other technologies are allowed if need be such as 7zip for zipping of files and unzipping of files, however it would be advisable to use a QT API such as the file handling API to do so. Furthermore the project must be deployable on windows and linux and be capable of working on each of them.

For phase 2 in the project, the project must be capable of running applications on the slave computers that have been distributed from a master computer to a slave. The configuration for the application to be run should also be used when running the program so that different environments can be selected.

2 Architectural patterns or styles

3 Architectural tactics or strategies

4 Use of reference architectures and frameworks

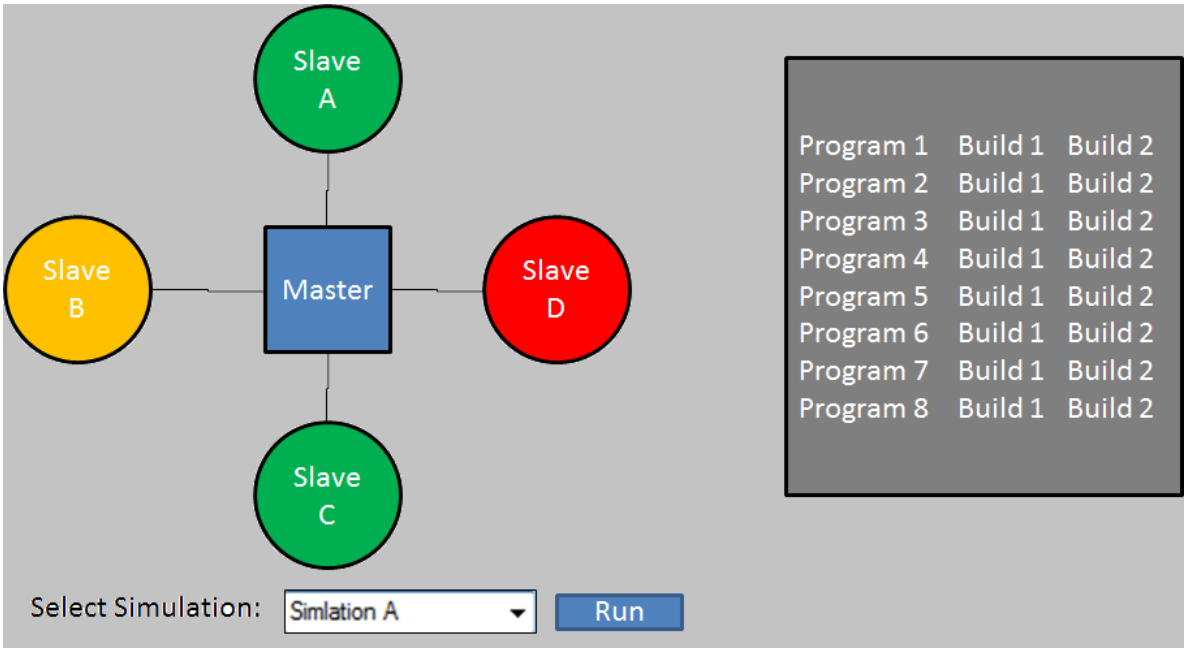
QT is a framework much the same as Java and their motto stated "Code Less. Create More. Deploy Everywhere." which means the framework allows cross platform development and deployment. This project will make use of QT 4.8.4 framework with C++ that will enable the project to be used on multiple platforms. We will strive to make use of the QT API's to provide a project with capabilities of compiling and running on multiple platforms.

QT 4.8.4 contains file API's such as QFile and QFileSystemWatcher and QDirIterator which can be used for synchronization of files and folders accross multiple computers. QFile will help with writing and editing of files that may be used in this project. The QFileSystemWatcher can be used to monitor a directory structure that may be specified by a user and if changes occur the changes can then be spread accross to the slave computers. The QDirIterator can be used to manipulate folders and directories if need be.

QT also contains an extensive API for networking that we can make use of when implementing the project. This will allow the slave and master computers to communicate and transfer files from one computer to the other. The applications can be run on multiple slaves by making use of QT networking capabilities. Examples of those networking API's are QUdpSocket and QTcpSocket that can be used to create connections between computers. The QNetworkInterface class can be used to obtain information on networking interfaces that a computer has if the need arises.

5 Access and integration channels

Below is a rough depiction of how the GUI will look. The different nodes each have a colour to indicate status. Green means ready, Yellow means some error has occured, Red means it is offline. A program build can be clicked on the list and placed on the slave it is needed on. This will initiate the copying sequence. Next a simulation is chosen and can be run.



6 Technologies