Robust Application with RoR

Prof. P.Karthikeyan* and Rama Pandian M
School of Information Technology & Engineering
VIT University
Vellore, India
mailme_kk@yahoo.com
ramapandianm2007@vit.ac.in

Abstract: This paper describes the advantages of using Ruby on Rails in commercial and enterprise class web application development. This paper also describes the power and advantages of Open source in the field of web development. Rails is an open source web application framework for the Ruby programming language which is typically integrated with database server such as MySQL and a web server such as Apache. This paper includes the overview and features of rails. Some useful rails plugins are also discussed.

Keywords: database, open source, web application framework, web server.

I. INTRODUCTION

SDLC is the traditional process of developing software or web applications by including research to identify and define the application requirements, information analysis, architectural design and specifications blueprint, workflow, team involvement, programming, testing and bug fixing, system testing, implementation and maintenance. Traditionally starting a new web application is a fairly heavy weight process, you typically need to survey and choose your various software components to solve the common architectural problems. Now the new trend is instead of investing more money in the development framework most of the leading enterprises adopt open source frameworks and soft wares. Thus they can concentrate highly on the business logic, maintenance and deployment. Rails framework helps them to achieve it and the main point of this paper is to explore the Ruby & Rails "Ecosystem".

II. WHY OPEN SOURCE?

Open source applications are now available for every common type of enterprise software -- from databases, application servers and Web servers, to Web browsers and office applications, to network monitoring software and security software. Open Source technologies offer enterprises cost savings with lower "total cost of ownership" and a release from the lock-in of traditional proprietary vendors. Enterprises find that open source softwares consistently get great value -- and the desired ROI(Return On Investment). Open source applications can be even more secure than their commercial equivalents. Open source communities fixed security vulnerabilities twice as quickly as commercial software vendors did. Open source software has been proven to provide better value, lower costs, and improved security, addressing the most important enterprise considerations today.

Following are some of the advantages of open source[6]
Lower total cost of ownership
Reduced dependence on software vendors
Easier to customize
Higher level of security

Fig. 1 shows the importance of using open source.
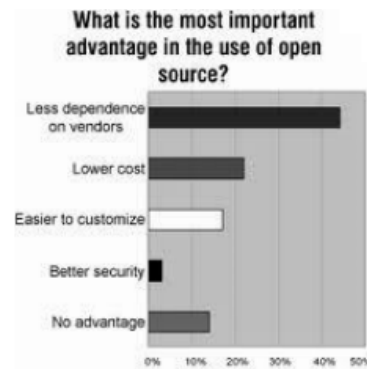

Figure. 1

III. LINUX

Linux is an open-source version of the UNIX operating system. This means the source code for Linux is publicly available for inspection and even improvement. The advantages of open-source products range from the benefits of creative input from programmers all over the world, to security. With the source code readily available in the public domain, open-source products can be vigorously tested by programmers all over the world. An operating system is the first piece of software that the computer executes when you turn the machine on. Linux is portable to any hardware platform.

IV. MYSQL

The MySQL database is the open source database which became famous due to its consistent fast performance, high reliability and ease of use. MySQL runs on more than 20 platforms including Linux, Windows, OS/X, HP-UX, AIX, Netware, giving you the kind of flexibility[7]. Since MySQL is an open source database server, one can customize it by addind their requirements to it. It has a unique storage-engine architecture which allows the database professionals to configure the MySQL database server. Due to its high performance, MySQL became the de-facto standard for high-traffic web sites. It supports specialized web functions like fast

full text searches and it suits to data warehousing environments. MySQL is a strong standout for both web and business intelligence applications. Ruby and Rails are Open Source so the virtual default database used by the majority of developers is the Open Source MySQL database[7].

V. RUBY

Ruby is a open source and pure Object oriented programming language introduced by Yukihiro Matsumoto (known widely as "Matz") in 1993. Ruby is a dynamic interpreted language which has many strong features of various languages. It is a scripting language like Python. The OOPs concept from C++ and Java helps ruby to maintain the security of code[1][2].

Ruby runtimes

A Ruby runtime is a program that accepts Ruby source files, converts them into executable form, and then provides an environment in which they can execute. So to run a program written in Ruby, including Rails, you need one of the Ruby runtimes that exists or is in the works. With one exception, these runtimes are Open Source[1].

VI. RUBY ON RAILS

Rails is a web application framework written in the Ruby language. It allows us to write less code than many other programming languages and frameworks. Ruby on rails projects are created quickly, very simple and work efficiently. Rails became very popular because it is open source framework. Software Centralization give cheaper per unit costs. Since it is open source it can be work on all platform(even on MS-DOS).

The main purpose for what Rails was created is to solve the real business problem and it grew into a major development tool that was promoted and taken up by Web 2.0 and agile businesses[1]. It became the powerful tool to create Web applications both internal(Enterprise) and external (products or services) to the commercial or non-profit worlds.

Rails allows to generate web applications rapidly which also includes advanced Ajax (it allows a part of a webpage to be updated with new information without the need to refresh the entire page) and has many components like Object-relational mapping, Web services, JSON(JavaScript Object Notation) and other APIs(Application programming interface) built-in. Integration between these components will reduce development time and effort significantly. The templating technology built into Rails can be used to generate web applications and other documents that requires dynamic content.

RoR is based on 2 main principles:
Convention over Configuration means a developer only needs to specify unconventional aspects of the application. For example, if there's a class Sale in the model, the corresponding table in the database is called sales by default.

Don't Repeat Yourself (DRY, also known as Single Point of Truth) is a process philosophy aimed at reducing duplication, particularly in computing.

MCV architecture

Rails uses a **MCV** (Model-View-Controller) style of architecture. In this the database and its representation -model, the user interface -view, and the functional logic-controller of the application are all handled separate effectivey. This architecture style typically helps create products that are clean, portable and sharable.

MVC architecture keep the Business Logic and the User Interface seperated to an extent.

The business logic just mentioned is a part of the "Model", along with this is the data that it interacts with. In web application it is a data in the database and the algorithms used to manipulate it. The "view" is geared at the User Interface. That is, the text on the web page, text boxes and other content. In web application it is the front end HTML pages. The "Controller" interfaces between the model and the view. It lets the model know what the user is doing in the view. It gathers the data based on user input that used as content for the HTML in the web application.

Agile programming methodology

Rails tie in with the rise and use of Agile programming methodologies, which were used to create the complex software projects rapidly. Agile processes use short iterations by teams of programmers and stakeholders to develop complex software applications rapidly. This became famous because it could adjust to constantly changing requirements.

VII. COMPONENTS OF RAILS[9]

Action Pack - Action Pack is a single gem that contains Action Controller, Action View and Action Dispatch. The "VC" part of "MVC".

Action Controller - Action Controller is the component that manages the controllers in a Rails application. The Action Controller framework processes incoming requests to a Rails application, extracts parameters, and dispatches them to the intended action. Services provided by Action Controller include session management, template rendering, and redirect management.

Action View - Action View manages the views of your Rails application. It can create both HTML and XML output by default. Action View manages rendering templates, including nested and partial templates, and includes built-in AJAX support.

Action Dispatch - Action Dispatch handles routing of web requests and dispatches them as you want, either to your application or any other Rack application.

Action Mailer - Action Mailer is a framework for building e-mail services. You can use Action Mailer to receive and process incoming email and send simple plain text or complex multipart emails based on flexible templates.

Active Model - Active Model provides a defined interface between the Action Pack gem services and Object Relationship Mapping gems such as Active Record. Active Model allows Rails to utilize other ORM frameworks in place of Active Record if your application needs this.

Active Record - Active Record is the base for the models in a Rails application. It provides database independence, basic CRUD functionality, advanced finding capabilities, and the ability to relate models to one another, among other services. Some of the outstanding properties of Activerecord- Table columns map to Object attributes, mapping class names to table names, pluralized table names, integer primary keys, classname_id as foreign keys, for simplicity and ease of use, hides low level implementation.

Active Resource - Active Resource provides a framework for managing the connection between business objects and RESTful web services. It implements a way to map web-based resources to local objects with CRUD semantics.

Active Support - Active Support is an extensive collection of utility classes and standard Ruby library extensions that are used in the Rails, both by the core code and by your applications.

Railties - Railties is the core Rails code that builds new Rails applications and glues the various frameworks and plugins together in any Rails application.

VIII. SOME RAILS PLUGINS

ActiveScaffold[3] - ActiveScaffold is a plugin that provides AJAXified table interface for creating, reading, updating, and deleting objects in the database tables. It is easy configuration-driven framework. In just a few easy steps, you can create a full web interface for your database. ActiveScaffold is perfect for creating a backend.

It provides the following goodness - Sorting, Search and Pagination, Graceful JavaScript degradation, RESTful API support (XML/YAML/JSON) baked in and beautiful CSS styling and theming support.

The github wiki that outlines common use cases: Prototyping, Admin Interfaces, Embedded Widget-Style, Data-Heavy Applications

Captcha - This plugin provides the security by validating the humanity of your visitors with either the common text rendered in an image or a simple question. This can be used in the web applications or websites where preventing Commenting Spam in blogs, protecting Website Registration, online Polls, etc.. are needed

Acts as Authenticated - The whole idea behind the plugin is that you generate the code once and add in your application specific authentication actions. This provides the authentication benefit to your rails application.

Acts As Rateable – This plugin allows ActiveRecord models to be rated between 1 upto some application specified number. This plugin will calculate the average rating for a record. This can be used in the websites where music, photos can be downloaded.

Rails PDF – This ActiveView plugin allows rails developers to create actions that produce PDF documents. This is a handy plugin for applications that want to generate reports.

Calendar Helper – This is an ActiveView plugin that will produce stylish calendars.

Graphs Rails – This ActiveRecord plugin is used to generate CSS powered bar graphs.

default_value_for - The default_value_for plugin allows one to define default values for ActiveRecord models in a declarative manner.

EventCalendar - This plugin will easily show multiple, overlapping events across calendar days and rows.

It will give good look to the calendar.

Restful-authentication - This widely-used plugin provides a foundation for securely managing user authentication:

Login / logout

Secure password handling

Account activation by validating email

Account approval / disabling by admin

Rudimentary hooks for authorization and access control.

IX. RAILS IS DATABASE DRIVEN

Rails framework can be used to develop "database driven" Web applications. As a framework, it is the quick way to develop and deploy the complex Web applications, typically it will reduce upto one-third to one-tenth time and expense taken by other alternatives. Once built, these sites are easier to maintain. Rails uses activerecord to generate SQL and send it to the database. It also creates a default primary key named id for each and every table[1]. The MySQL database server provides the ultimate in scalability. Databases interact with the User Interface through the Business Logic. It controls how data is gathered and manipulated.

X. BEHAVIOR DRIVEN DEVELOPMENT(BBD) WITH RAILS

RSpec(a Behavior Driven Framework for Ruby) became an important aspect of RoR to support Agile development methodologies. BDD provides collaboration among all people who involved in product development. It also encourages developers to think about the behavior of the component they are developing, and the roles of other components they interact.

BDD makes testing very easy and it reduce the testcases. Further it will explain the behavior of the developing program. RSpec provides two frameworks namely Story and a Spec framework[1]. This allows us to write stories with different scenarios and it will assure that the test will pass. The logic of the code being written can be tested with the story and spec framework.

Support from Behavior Driven Development and Agile methodologies, and the nature of Web 2.0, Ruby on Rails became a good platform of choice to build new products and services for the next generation.

XI. TEST DRIVEN DEVELOPMENT

It is the part of Agile development. Testing and Development go on in parallel. Actually in this test cases are developed before development. Tests are based on requirements and not on development. Development is focused on getting tests to pass.

XII. DEPLOYMENT

The actual deployment of a Rails application requires some specific knowledge about the environment and dedication because we need numerous components to make the application works in the server[1].

We need a ruby runtime to run the ruby program. The version is very critical and will become more complex as the alternative Ruby runtimes reach maturity. The operating system and the Web server also are version sensitive. In the case of the Web server configuration process involved. The stable database is the absolute requirement. Actually we need some tool like Capistrano (an open source tool for running scripts on multiple servers) to deploy the rails application. Then also you need a software called an application server like Mongrel (an open-source HTTP library and web server written in Ruby) to tie everything to make it all work together.

Rails applications runs on a server. If the application get to a certain size or function, it will hosted on a data center or server farm that sells access to their machines. Figure 2 shows the Rails deployment clearly.
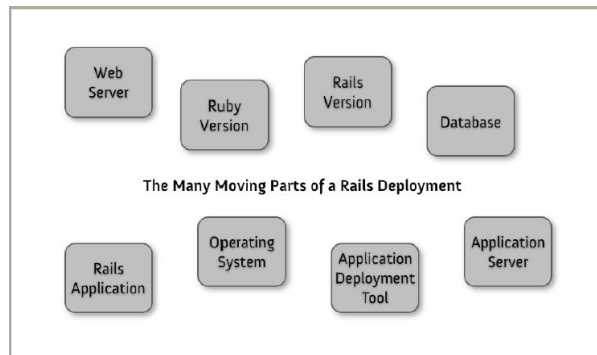
The Many Moving Parts of a Rails Deployment

Figure 2

XIII. DEVELOPMENT TOOLS WITH RAILS

Rails is used to create many developer tools like "text editors," "Integrated Development Environments (IDEs)" and "project management systems". These are used together if they all share some common functions used in software development.

Redmine is a integrated Open Source project management tool that was recently introduced which is a combination of project management and bug tracking, and it is being used increasingly in conjunction with text editors and/or IDEs.

In the Ruby and Rails Ecosystem, TextMate (by MacroMates) is the most popular developer tool which is a text editor for the Mac OS X operating system.

XIV. SPONSORS [1]

Sponsors enter: Sun Microsystems developed the JRuby runtime to allow it to more easily run the vast number of Java applications and libraries available.

Microsoft is in the early stages of an IronRuby runtime to optimize .NET applications.

One of the most important milestones, however, was the decision by Apple in October 2007 to begin shipping Rails along with the main Ruby interpreter in the Mac OS X 10.5 operating system (known as *Leopard*) that comes with all Apple computers.

Further, Apple is working on MacRuby, a Ruby runtime optimized to its OS X operating system.

XV. RESULT

Rails is the powerful ruby framework to implement web applications and VoIP. Since it is open source we can recommend it to use in the companies to reduce the cost of ownership. It is helpful for them to spend the money on other phases of the project.

XVI. REFERENCES

[1] The Ruby & Rails Ecosystem White Paper, Initial Release Version - Fall 2009, Developed and Distributed by:InfoEther, Inc.

[2] 10-reasons-for-ruby-on rails www.abletech.co.nz/2006/07/white-paper-10-reasons-for-ruby-on-rails

[3] http://activescaffold.com/2007/5/24/how-to-approach-active-scaffold

[4] Selina D'Souza - WHITE PAPER New Product Development with Ruby on Rails

[5] Key Advantage of Open Source is Not Cost Savings http://www.computereconomics.com/article.cfm?id=1043

[6] MySQL white papers

[7] Guide to Ruby on Rails-http://guides.rubyonrails.org/getting_started.html