Odd Thesis

Bob's stumbling through JBoss technologies, doing random stuff.

http://oddthesis.org/

jboss.org

oddthesis

Background on Bob

- Bob was the manager for the jboss.org community.
- Bob decided he shouldn't be management & took a break.
- Bob spent his summer building RoR apps.
- Bob regained some sense; proposed some ideas to Sacha.

Some of the ideas...

- Rails isn't going away.
- JBoss technologies could support Rails.
- JBoss could provide more enterpriseyness to Rails.
- Profit!

Also...

- Rails development is super zippy.
- Rails deployment is extremely annoying.
- "Rails can't scale" is an opportunity to show how JBoss can bring scale, even to Rails.
- I don't own a cluster, but I can rent one from Amazon by the hour.

Why Odd Thesis?

- Every project starts with a logo...
- I blog a lot.
- I should eat my own dog-food.
- I already have my own RHEL server/sandbox at Contegix.
- Domain names are cheap.

The Broad Thesis

Since you can develop an app in a day using RoR, you should be able to deploy it to a highly scalable, clustered cloud environment in *mere minutes*.

<sarcasm/>

Thus the theses...

1. Need to get Rails running in JBossAS.

2. Need to get JBossAS running on EC2.

3. Need to give people reasons to run Rails on JBoss.

Rails on JBoss

Thesis I

- JRuby is quite functional.
- Ruby frameworks are standardizing on the **Rack** API for gluing to servers.
- Nick Sieger wrote **JRuby-Rack** to glue the Rack API to the Servlets API.

JBossMC Deployer

- AS5, being built upon JBoss Microcontainer is fantastically excellent.
- A deployer to load up JRuby, JRuby-Rack, and deploy a RoR app in-place was almost trivial.
- Deploys into the standard web container already running.
- Creates & deploys the same metadata that a WAR's web.xml does; *without* creating a web.xml.

Benefits

- The "live" development of Rails is respected. Editing views, controllers, models is immediately reflected in the running app; *without* actual redeployment.
- Existing clustered sessions with JBossCache *just work*.
- The Rails functionality is small (8MB, including dependencies) & self-contained.

JBoss on EC2

Thesis 2

- Installing JBoss is easy (unzip jbossas.zip).
- Homogeneous nodes on EC2 are easy.
- Another group in Red Hat has already started working on "appliances" for virtualized environments (KVM, Xen, EC2, VMWare), including a JBoss appliance (JBoxx).

Thincrust: http://thincrust.net/

Creating Appliances

- Builds upon Fedora or RHEL repositories.
- Starts with a “just enough” minimal RPM set
- Plus *Puppet* for updating & reconfiguring.
- Appliances defined through Kickstart descriptions and Puppet recipes.

Which means RPM...

<sigh/>

- The JPackage guys are creating an RPM for every JAR known to man.
- JON/JOPR expects a "normal" installation of JBoss (unzip jbossas.zip).
- Turning our normal .zip into an offensive-to-sysadmins .rpm is not too difficult.
- So I took the easy way out...

JBoxx5-bob Appliance

- The Thincrust guys have JBoss AS4 in an appliance, but I need JBoss AS5.
- I created an appliance with...
  - AS5
  - JBoss Rails deployer
  - An appropriate init.d script
- It boots up, ready to roll.

EC2 Limitations

- *Boots from DHCP*. Need to reconfigure /etc/jbossas.conf at each boot to control which IP address to bind to.
- *Multicast is disabled*. Must use a JGroups Gossip server to rendezvous nodes. (EC2 allows injecting the Gossip server's address into nodes at boot time)

Further Work

- While JBoss in a cluster might be homogeneous, you also need appliance images for...
  - A web front end (httpd & mod_cluster)
  - A database or two (involves Amazon EBS)
- Plus support to easily deploy a variety of cluster configurations.
- And resize existing deployments.

Enterprisey Rails

Thesis 3

If clustering and instant cloud support isn't enough to give traditional Rails developers a reason to use JBoss for deployment...

- Integrate awesome Java-only packages without comparable quality Ruby equivalents: Drools, ESB, jBPM, etc
- Solve problems (scheduling, management) common to non-JBoss deployments.
- Support Windows-based Rails development better than traditional Ruby.

Frequently Asked Questions

Why Ruby & Rails?

- It's a nice language.
- Rails may not be ideal, but the framework and the components around it lead to fast development.
- The Rails community is still in flux, changing technologies often, looking for "the best" solution.
- JBoss has "the best" solution to the general problems.
- We can easily bind JBoss to Rails and win.

Why not start with Goovy & Grails?

- Groovy & Grails users tend to already be Java developers.
- And they already either use us or a competitor. They've established their Java environments already.
- Ultimately does not enlarge the potential market.
- Ironically, I know Ruby better than Groovy.

What integration points?

- Integration with Drools will bring a Ruby-friendly syntax to the power of our rule engine.
- Integration with Messaging and Remoting will help integration with legacy (enterprisey) systems, along with filling a gaping hole in traditional Ruby systems.
- I'm open to suggestions...

Why are you using *git*?

- The Ruby community has pretty much entirely moved to *git* for development.
- Github provides nice free repositories.
- I'm often offline, and *git* supports that amazingly well.

Less-Frequently Asked Questions...?

Resources

Website/Blog/Documentation

- http://oddthesis.org/

Source Code

- http://github.com/bobmcwhirter/jboss-rails
- http://github.com/bobmcwhirter/jboss-rails-plugin

My out-of-date Thincrust bits

- http://github.com/bobmcwhirter/thincrust-ace
- http://github.com/bobmcwhirter/jboss-rpm

jboss.org

oddthesis