Groovy & Grails
Scripting for Modern Web Applications

11, 12 Dec. 2009

Pune, India

IndicThreads.com Conference On Java Technology

Rohit Nayak

Talentica Software

Agenda

- Demo: Quick intro to Grails
- Scripting, Web Applications and Grails/Groovy
- REST service in Grails
  - Demo
  - Internals
- Web Client in Grails
  - Demo
  - Internals
- Perspective

Demo

Quick Intro to Grails

Maggi

- grails create-app Maggi
- Domain class: Noodle, Packaging
- grails generate-all Noodle, Packaging
- grails run-app

Web Frameworks with Scripting

- Ruby on Rails (2004)
- CakePHP (2005)
- Django / Python (2005)
- Groovy on Grails (2006)

Power of these frameworks

- Baked Experience
- The Language
- Agility / Productivity

Baked Experience

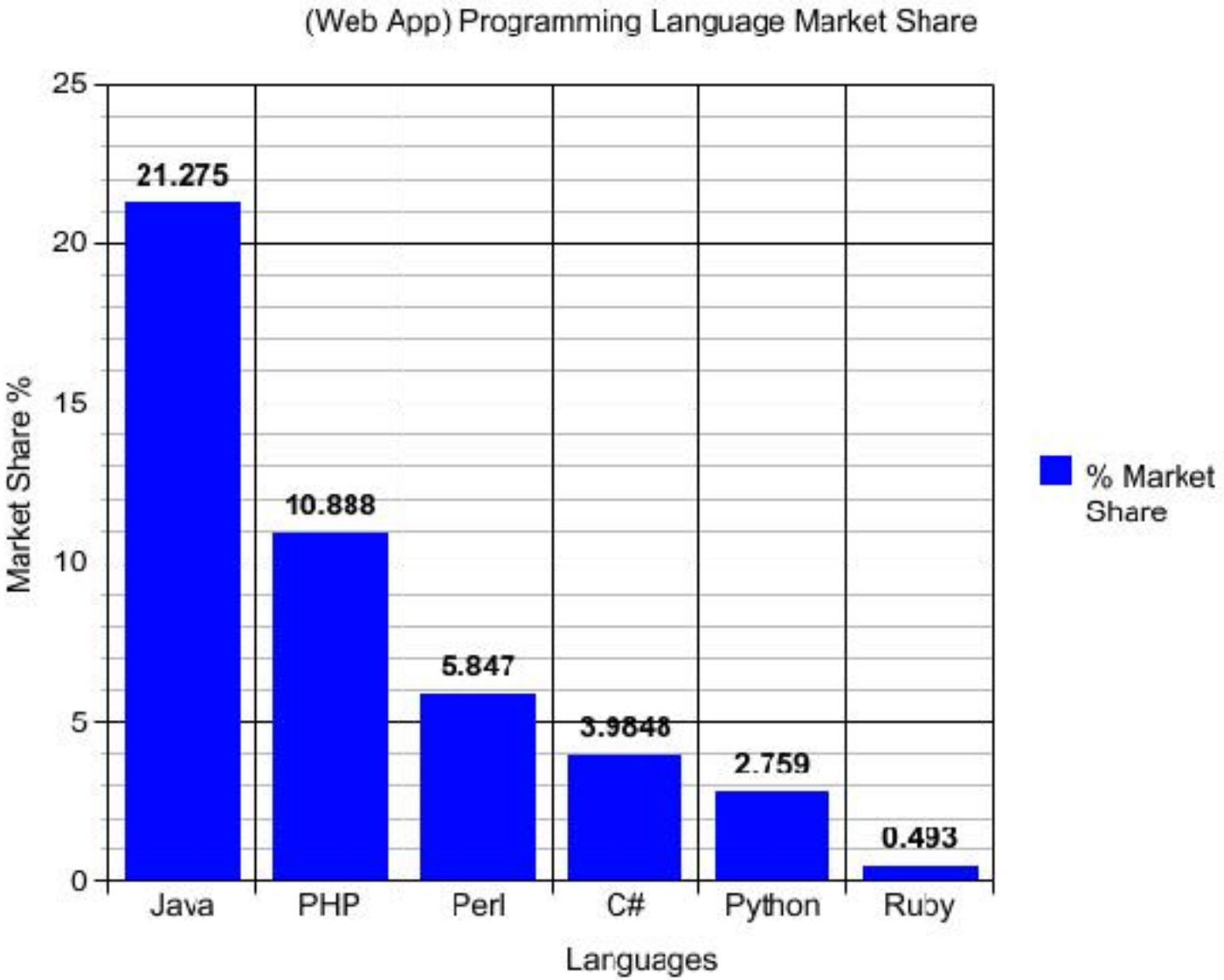
- Model View Controller
- Object-Relational Mapping
- Templates
- Layout
- URL rewriting
- Ajax support
- XML / JSON support

The Language

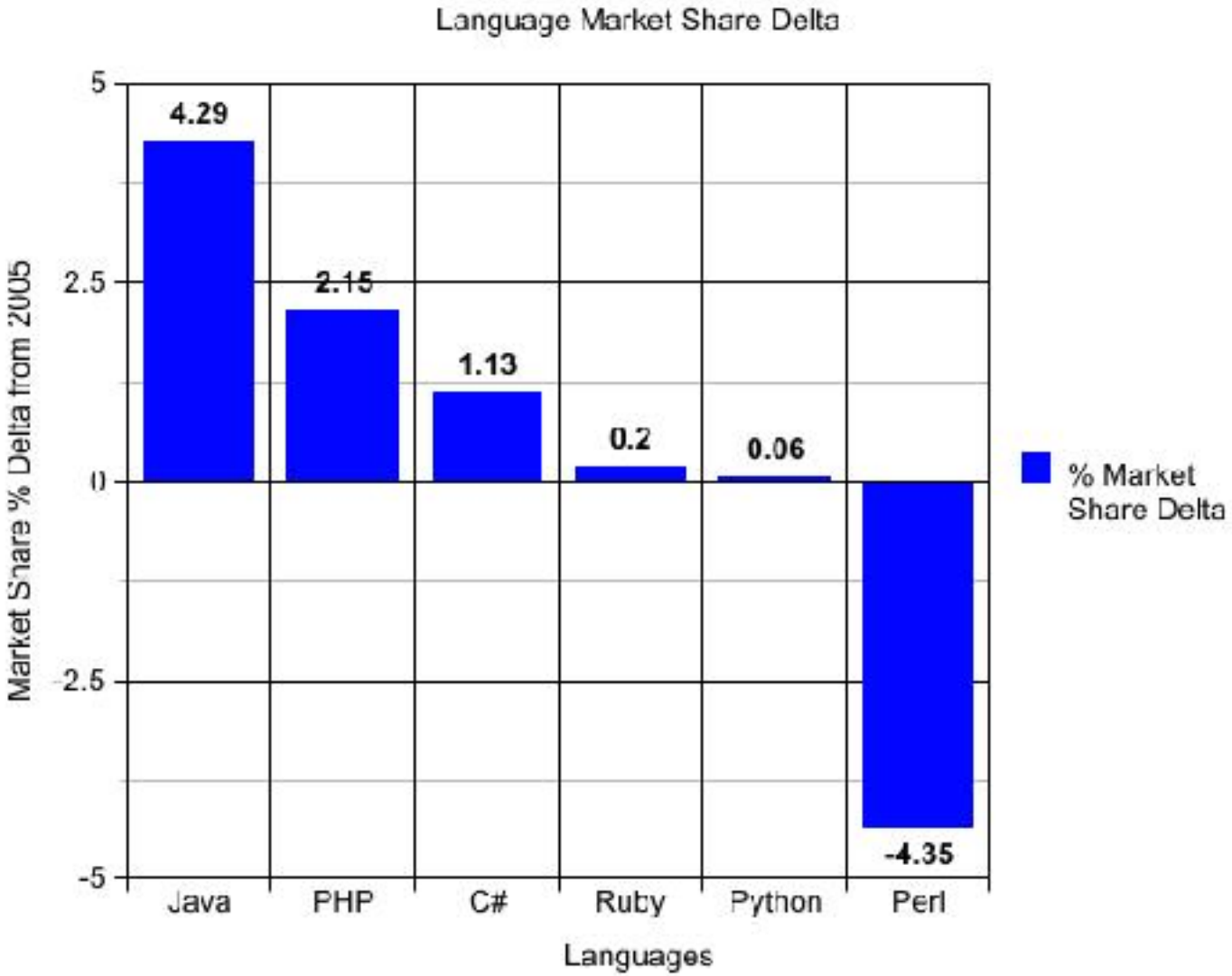
- Dynamic
- More expressive code
- Smaller code
- Native support for Lists, Hashmaps
- Lang. support for IO, Net, XML
- Idioms for common Design Patterns

Agile

- Scaffolding
- Unit tests
- No compile cycles
- Built-in webservers
- Fail faster!



http://www.zacker.org/ruby-on-rails (Nov 2nd)



http://www.zacker.org/ruby-on-rails (Nov 2nd)

HelloWorld.java

```
public class HelloWorld {

    String name;

    public void setName(String name)
        { this.name = name; }

    public String getName() { return name; }

    public String hello()
        { return "Hello "+ name; }

    public static void main(String args[]) {

        HelloWorld helloWorld = new HelloWorld();

        helloWorld.setName("Java");

        System.out.println( helloWorld. hello() );

    }

}
```

HelloWorld.groovy

```
class HelloWorld {
    String name
    def hello() { "Hello $name" }
}
def helloWorld = new HelloWorld(name:"Groovy")
println helloWorld.hello()
```

Key Groovy Features

- Java-like syntax
- Complements Java
- Object-oriented
- Targets Java VM (JSR-241)
  - Invoke Java class within Groovy
  - Invoke Groovy class within Java
- Dynamic
- Scripting (JSR-223)
- Brevity

Brevity

- Optional semicolons, package prefixes
- Automatic imports (java.util.*, java.net.*, java.io.*, groovy…)
- GroovyBeans (generated accessors)
- Optional typing
- Optional return

Groovy Gravy

- GStrings: ”$book.title: $book.author (${book.reviewers.length})”
- Regular expressions: assert ‘12345’ =~ /\d+/
- Only objects: primitives converted to Reference Types
- Lists: def list = [1, 2, 'hello', new java.util.Date()]
- Maps: def map = ['name':‘Indic Threads', 'location':‘Pune']
- Closures [1,2,3].collect {it*2} ===> [2, 4, 6]
- String literals – single, double, triple quotes

Closures

- Block of statements
- First class objects
- Parameters including default values
- Carries local context

```
def foo(n) {    // method
        return { n += it }    // returns closure
}
def accumulator = foo(1)  //curried closure
assert accumulator(2) == 3   //assert used for self-documenting, unit-testing
assert accumulator(1) == 4
```

Dynamic Programming

- Add methods, properties to classes/objects at run-time
- Mixins to inject behaviour
- Can extend class field-access mechanism
- Dynamic method invocation

Poolster

- Online “football pools” application
- Entities: Game, User
- Game → Stake, Option, Ends
- To join User chooses an Option
- REST Webservice backend
- Clients: iPhone, Grails, Android, Silverlight

Demo

The Poolster Webservice

Grails – Philosophy

- Convention over Configuration
  - Magic directories
  - Implicit table names, column names
- Don't Repeat Yourself
  - Database maps to domain
  - hasMany defines relationship **&** declares variable
  - Layout, form validations
- Lightweight
  - Modify and F5
- Strong shoulders
  - Spring (Grails MVC, DI, Transactions)
  - Hibernate (GORM)
  - Ant, JUnit, SiteMesh

Grails - Key Features

- Database constraints, composition
- MySQL integration
- URL Mapping
- Authentication / Filters
- Bootstrapping
- Unit test
- Logging

Demo

Poolster Web Client

Grails Web Application

- Custom Tag libraries
- Session variables
- Sitemesh layout
- Templates
- Internationalization

Unseen Gravy

- JUnit test cases, Mocking/Stubbing
- Web testing with Canoo Webtest
- Bootstrapping
- Pagination

Cons

- Learning curve
- Performance ?!
- Early adopter issues

Scripted In Groovy

- Canoo WebTest
- Tellurium
- Ant / Maven config files
- SoapUI script step
- Spring beans
- <YOUR APP HERE> ☺

Invoking Groovy Scripts

```
import java.io.File;
import groovy.lang.Binding;
import groovy.util.GroovyScriptEngine;

public class ScriptEngineEmbedGroovy{
    public static void main(String args[]) throws Throwable{
     String[] paths = {"C:\\groovy"};
     GroovyScriptEngine gse = new GroovyScriptEngine(paths);
     Binding binding = new Binding();
     Object[] path = {"C:\\music\\mp3"};
     binding.setVariable("args",path);
     gse.run("Songs.groovy", binding);
    }
}
```

Getting Started

- groovy.codehaus.org
- grails.org
- Free e-books
  - Beginning Groovy and Grails (Apress)
  - Getting started with Grails (InfoQ)
- refcardz.com cheat sheets
- ibm.com Mastering Grails, Practically Groovy

Thanks