COMS E6125 Web-enHanced Information Management (WHIM)

Web Development Frameworks

Swapneel Sheth

swapneel@cs.columbia.edu

@swapneel

Spring 2012

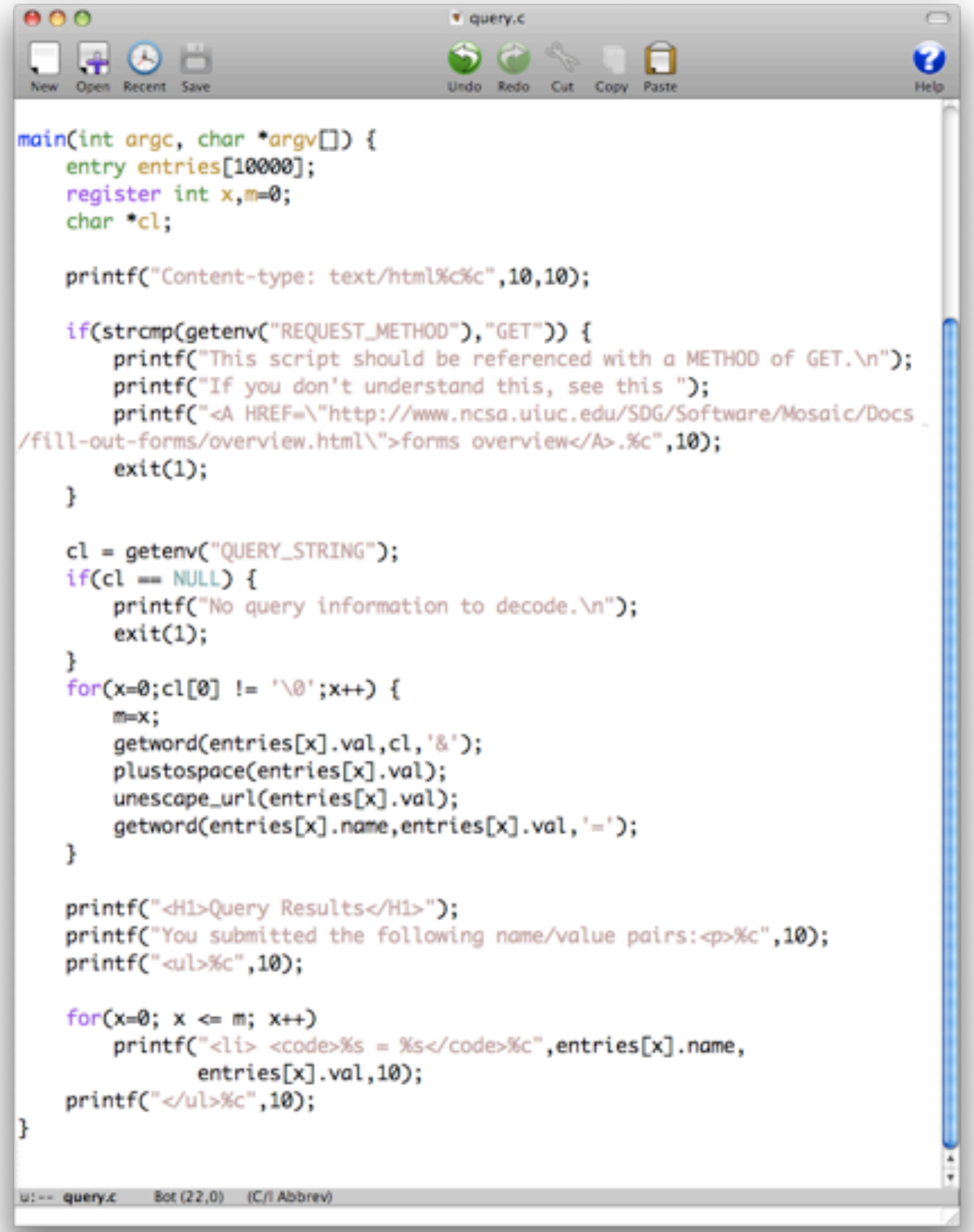
Topic 1 – History and Background of Web Application Development

- Static HTML Document
- Web Servers would retrieve the text file and send it to the user
- Needed some mechanism to output dynamic information from queries executed in real-time

Common Gateway Interface (CGI)

- Standard for external gateway programs to interface with information servers such as HTTP servers
- CGI programs can be written in any language and can either be compiled or “interpreted”
  - Compiled Languages: C, C++, Fortran, etc.
  - Scripting Languages: Perl, Shell scripts, etc.

Common Gateway Interface (CGI)



```
main(int argc, char *argv[]) {
    entry entries[10000];
    register int x,m=0;
    char *cl;

    printf("Content-type: text/html%c%c",10,10);

    if(strcmp(getenv("REQUEST_METHOD"),"GET")) {
        printf("This script should be referenced with a METHOD of GET.\n");
        printf("If you don't understand this, see this ");
        printf("<A HREF=\"http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs
/fill-out-forms/overview.html\">forms overview</A>.%c",10);
        exit(1);
    }

    cl = getenv("QUERY_STRING");
    if(cl == NULL) {
        printf("No query information to decode.\n");
        exit(1);
    }
    for(x=0;cl[0] != '\0';x++) {
        m=x;
        getword(entries[x].val,cl,'&');
        plustospace(entries[x].val);
        unescape_url(entries[x].val);
        getword(entries[x].name,entries[x].val,'=');
    }

    printf("<H1>Query Results</H1>");
    printf("You submitted the following name/value pairs:<p>%c",10);
    printf("<ul>%c",10);

    for(x=0; x <= m; x++)
        printf("<li> <code>%s = %s</code>%c",entries[x].name,
               entries[x].val,10);
    printf("</ul>%c",10);
}
```

```
u:-- query.c      Bot (22,0)    (C/l Abbrev)
```

Common Gateway Interface (CGI)

- Advantages
  - Ability to provide dynamic real-time content
- Disadvantages
  - Too low level
  - Various security issues
  - Each invocation needs to fork a new process, thus sub-optimal

Server Side Includes (SSI)

- Dynamically add small amounts of content in static pages
- Special code gets executed on the server and dynamically replaced with real content
- NOT a replacement for CGI – an easier way to include small amounts of dynamic information when CGI is overkill
- Similar to JSPs

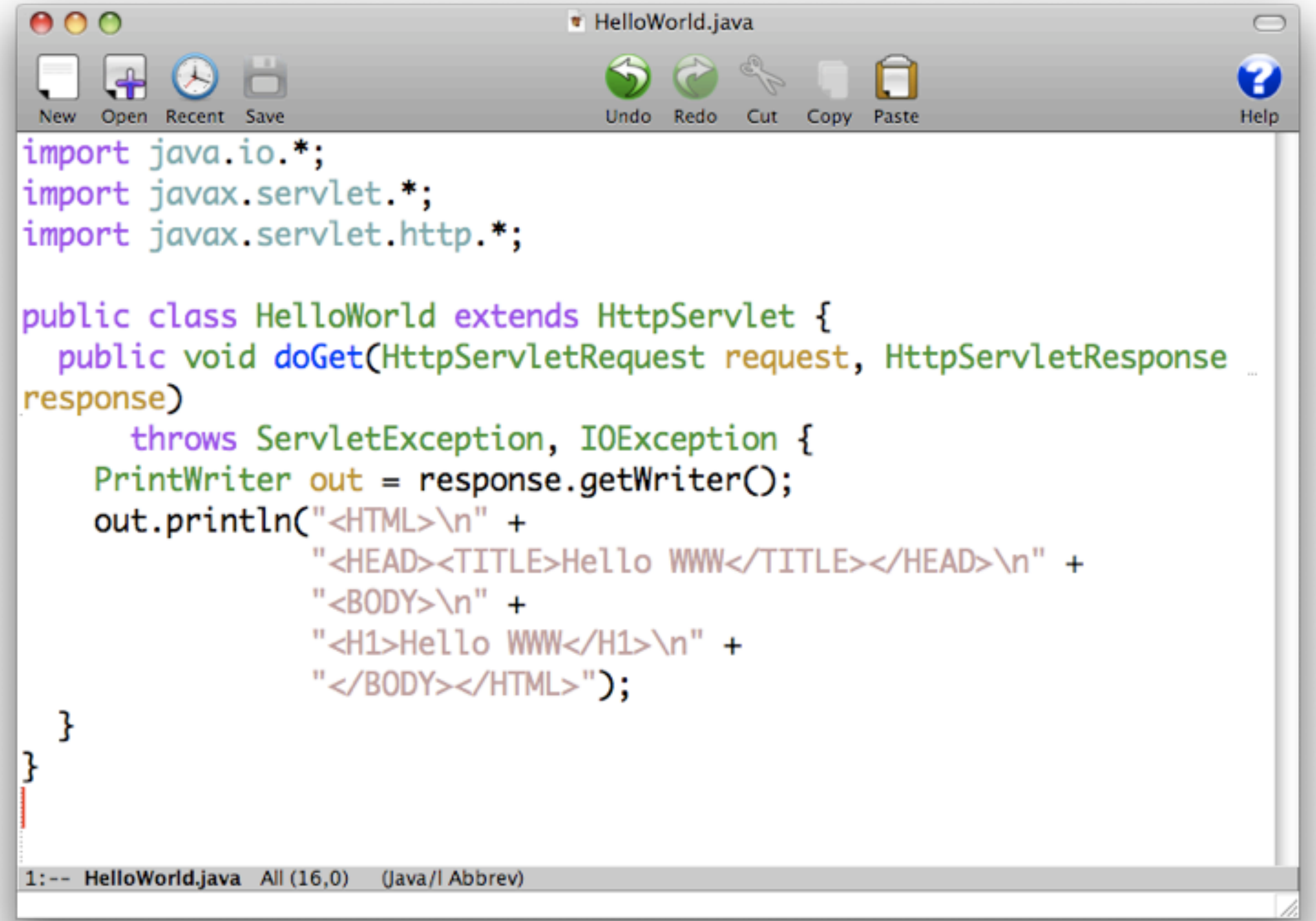
Active Server Pages (ASP)

- ASP was Microsoft's first server-side script engine for dynamically-generated web pages
- Originally released as an addon to IIS Server
- Most pages written in VBScript, but other languages are allowed (e.g., JScript, PerlScript)

Java Servlets

- Java's solution for generating dynamic web content
- Servlet 1.0 specification finalized in June 1997
- Servlet is an Object that receives a request and generates a response based on that request
- Servlets can maintain state across requests
- Can be automatically created from JavaServer Pages (JSPs)

Java Servlets



```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorld extends HttpServlet {
  public void doGet(HttpServletRequest request, HttpServletResponse
response)
      throws ServletException, IOException {
    PrintWriter out = response.getWriter();
    out.println("<HTML>\n" +
                "<HEAD><TITLE>Hello WWW</TITLE></HEAD>\n" +
                "<BODY>\n" +
                "<H1>Hello WWW</H1>\n" +
                "</BODY></HTML>");
  }
}
```

```
1:-- HelloWorld.java   All (16,0)    (Java/l Abbrev)
```

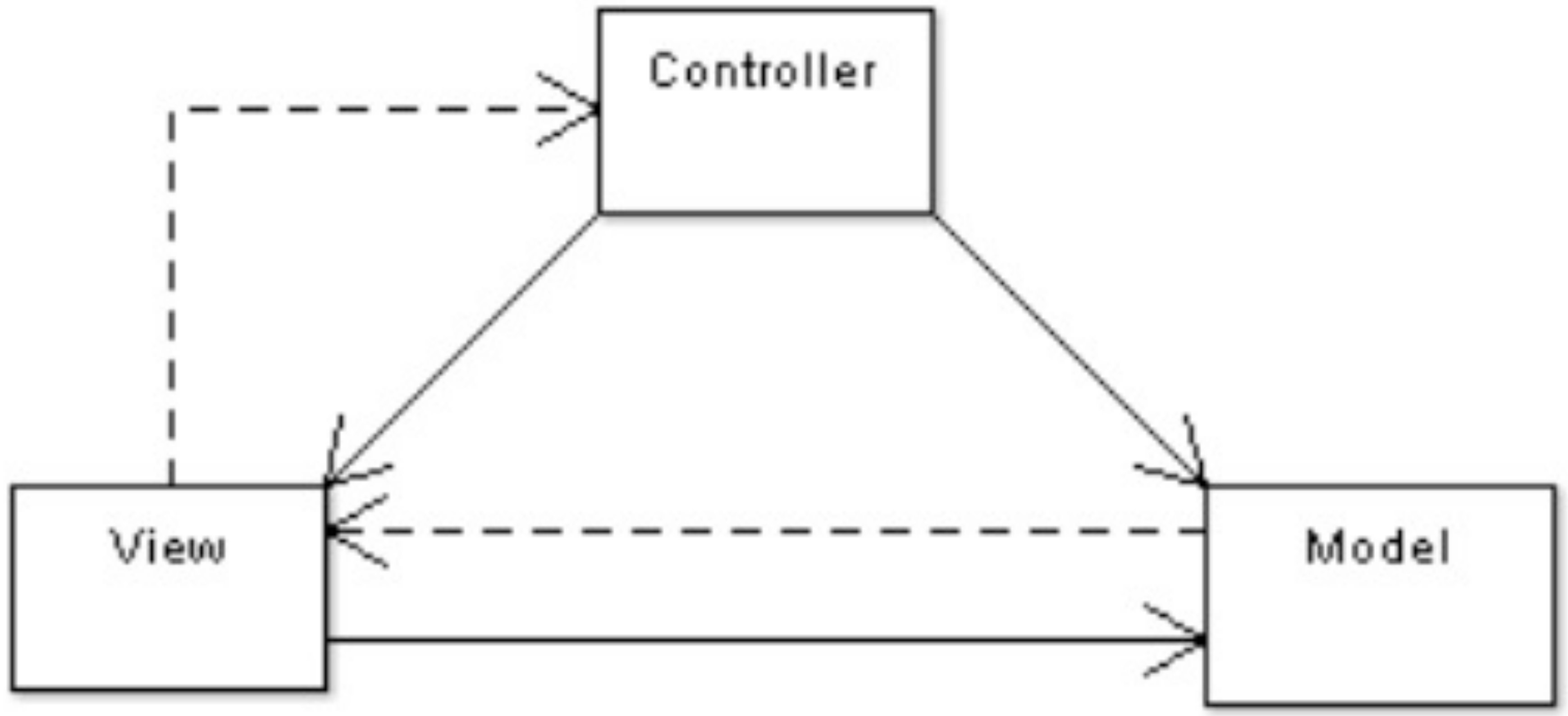
Topic 2 – MVC Frameworks

- What is MVC?
  - Model – View – Controller
  - Architectural and Design Pattern
  - Described in 1979 by Trygve Reenskaug who was working on SmallTalk at Xerox PARC
- MVC – Then and Now
  - "Rediscovered" for web app development

Quotes from Trygve Reenskaug

- "MVC was conceived as a general solution to the problem of users controlling a large and complex data set."
- "The hardest part was to hit upon good names for the different architectural components. Model-View-Editor was the first set."

MVC Architecture



MVC Architecture – Model Layer

- Corresponds to the database – some form of data persistence
- Can be a real database like MySQL, PostgreSQL, etc.
- Can alternatively be an XML file, flat files, etc.

MVC Architecture – Model Layer (2)

- Decouple the data storage and retrieval from the other aspects such as the UI
- UI does not change depending on whether the data comes from an XML file or from an Oracle DB
- Central place to do all the validations such as integrity constraints and null checks

MVC Architecture – View Layer

- Corresponds to the User Interface
- For web apps, this is typically a web page
- The web page designer need not be concerned about things like business logic
- Programmers typically use tools like Eclipse and emacs; Web page designers use different tools like Adobe Dreamweaver
- Allow the web page designers to use whatever they are comfortable with

MVC Architecture – Controller Layer

- Corresponds to the "business logic"
- Theoretically lets the programmers use any language they are comfortable with
  - There are no dependencies with the View or the Model Layers
- In practice, this is not true as picking an MVC framework forces you to use a fixed programming language

TURBO GEARS

tapestry

symfony

CakePHP

Stripes

Struts

RAILS

pylons POWERED

Spring

merb

seaside

django

catalyst Web Framework

JBoss a division of Red Hat

nitro

Cappuccino

Topic 3 – Ruby on Rails

- Web Application Framework created by David Heinemeier Hansson (DHH) at 37signals
- Extracted from real-world web application called Basecamp and made open source in 2004
- Some 37signals applications
    - Basecamp (project management)
    - Ta-Da List (personal todo list)
    - Campfire (business oriented online chat service)

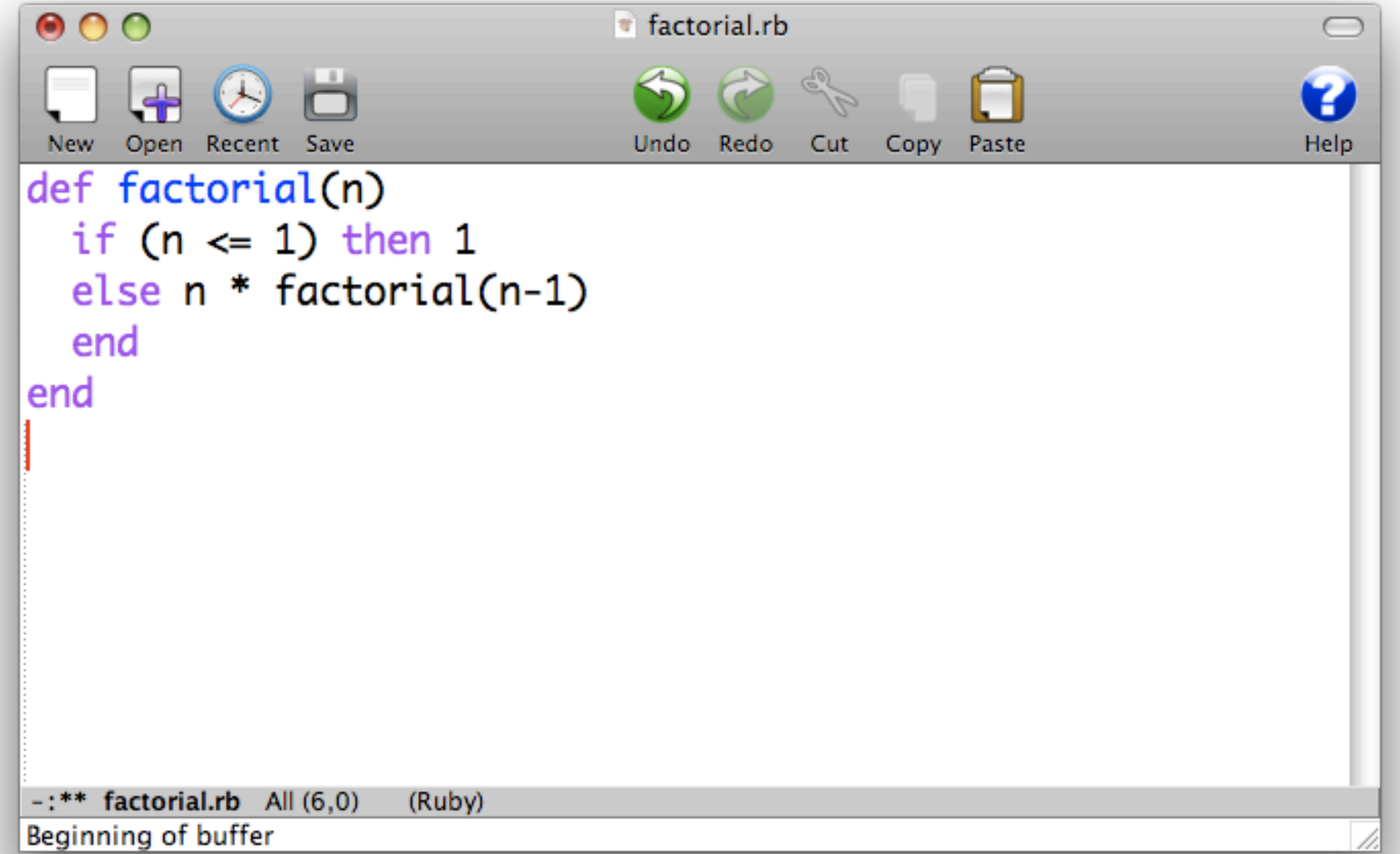
Ruby on Rails

- Uses Ruby
- Ruby is a dynamic, object-oriented programming language
- Created by Yukihiro Matsumoto (Matz) in 1995
- Based on Perl, Smalltalk, Eiffel, Ada, and Lisp
- Supports multiple programming paradigms – functional, OO, imperative, etc.
- Strong support for reflection and Metaprogramming

Design Philosophy of Ruby

- "I wanted a language more powerful than Perl and more object-oriented than Python. Then, I remembered my old dream and decided to design my own language." – Matz
- Principle of Least Surprise
- Make programming fun!

Sample Ruby Code



```
def factorial(n)
  if (n <= 1) then 1
  else n * factorial(n-1)
  end
end
```

```
-:**  factorial.rb   All (6,0)     (Ruby)
Beginning of buffer
```

Design Philosophy of Ruby on Rails

- Don’t Repeat Yourself (DRY)
  - Very Little Duplication
  - “Every piece of knowledge in a system should be expressed in just one place”
- Convention over Configuration
  - Sensible Defaults for Everything
  - “Follow the conventions and you can write a Rails application using less code than a typical Java web application uses in XML configuration”

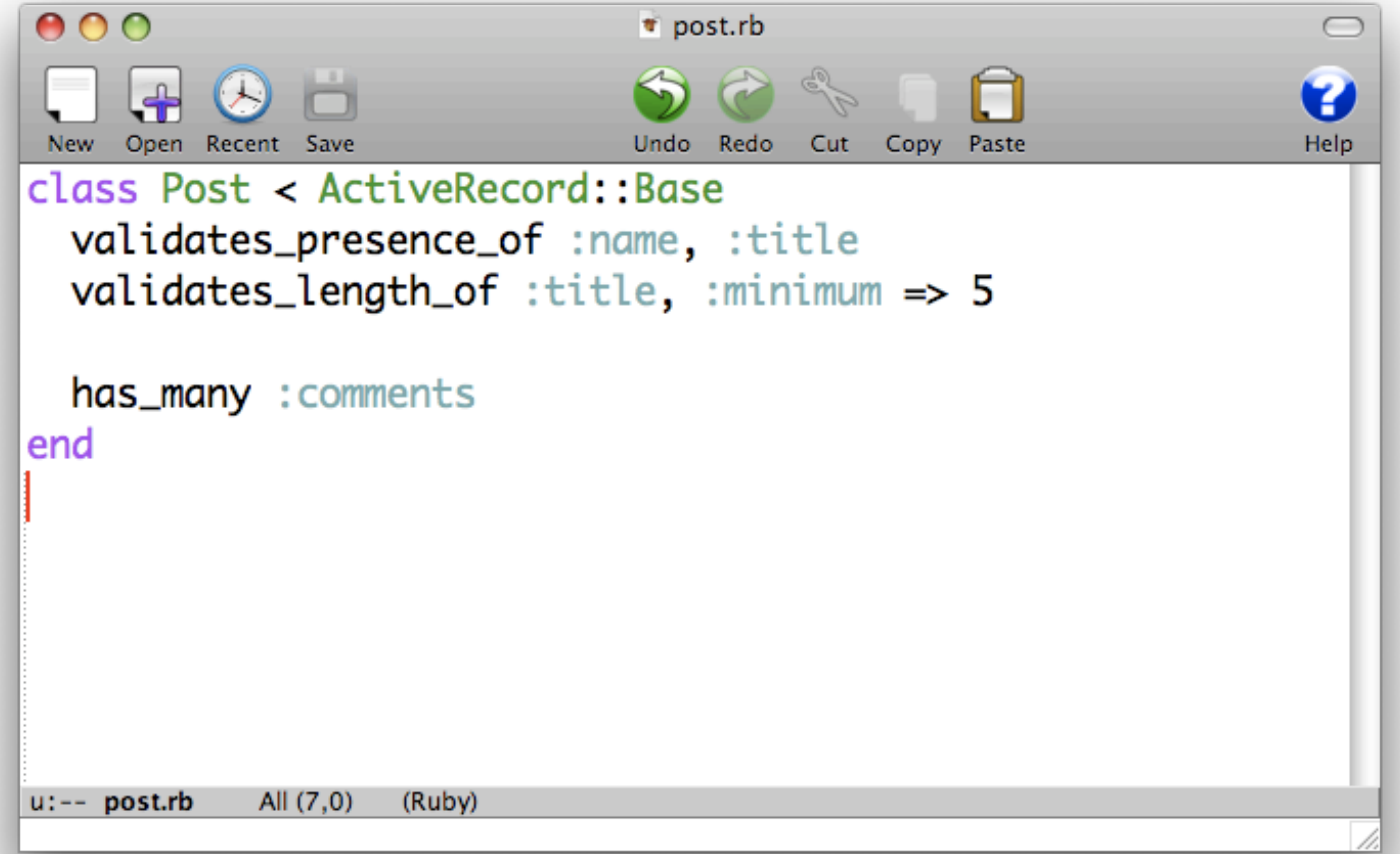
Design Philosophy of Ruby on Rails

- Inspired other MVC frameworks
- Most notable ones include
  - Symfony
  - CakePHP
  - PHP on TRAX
  - Merb

Ruby on Rails – Model Layer

- Active Record is the default Model Component in Rails and is the Base Class for all models
- Provides Object-Relational Mapping (ORM)
  - Mapping between tables in the database and the classes in the application
  - Classes correspond to Tables
  - Attributes correspond to columns of the table
  - Objects correspond to rows of the table
- Provides database independence, basic CRUD functionality, advanced finding capabilities, etc.

Ruby on Rails – Model Layer (2)



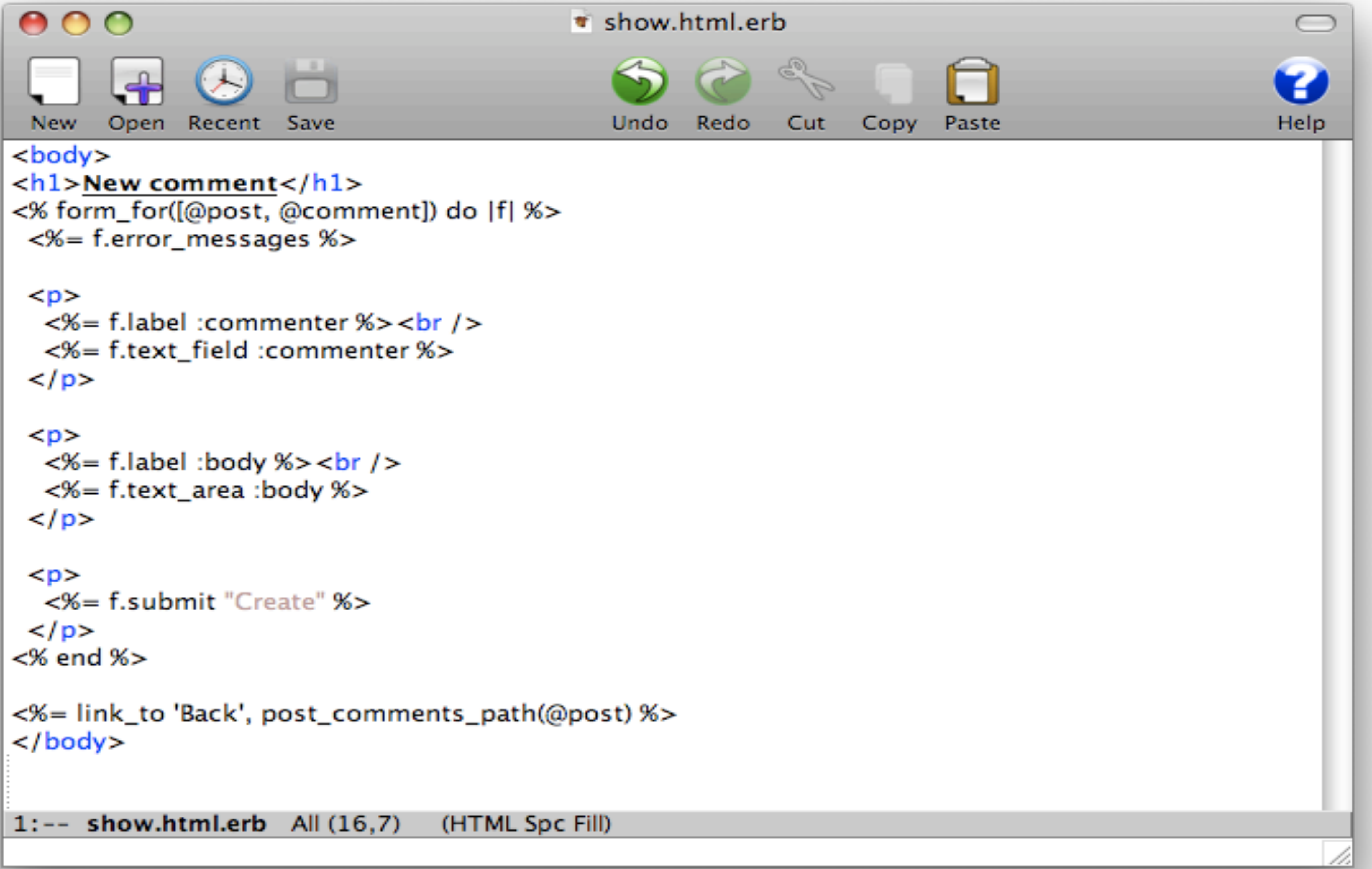
```
class Post < ActiveRecord::Base
  validates_presence_of :name, :title
  validates_length_of :title, :minimum => 5

  has_many :comments
end
```

```
u:-- post.rb      All (7,0)     (Ruby)
```

Ruby on Rails – View Layer

- Action View manages the views in Rails applications
- Can create both HTML and XML output by default
- Manages rendering templates, including nested and partial templates, and includes built-in AJAX support
- Can embed Ruby code in HTML for the View Layer (similar to JSPs, etc.)

Ruby on Rails – View Layer (2)



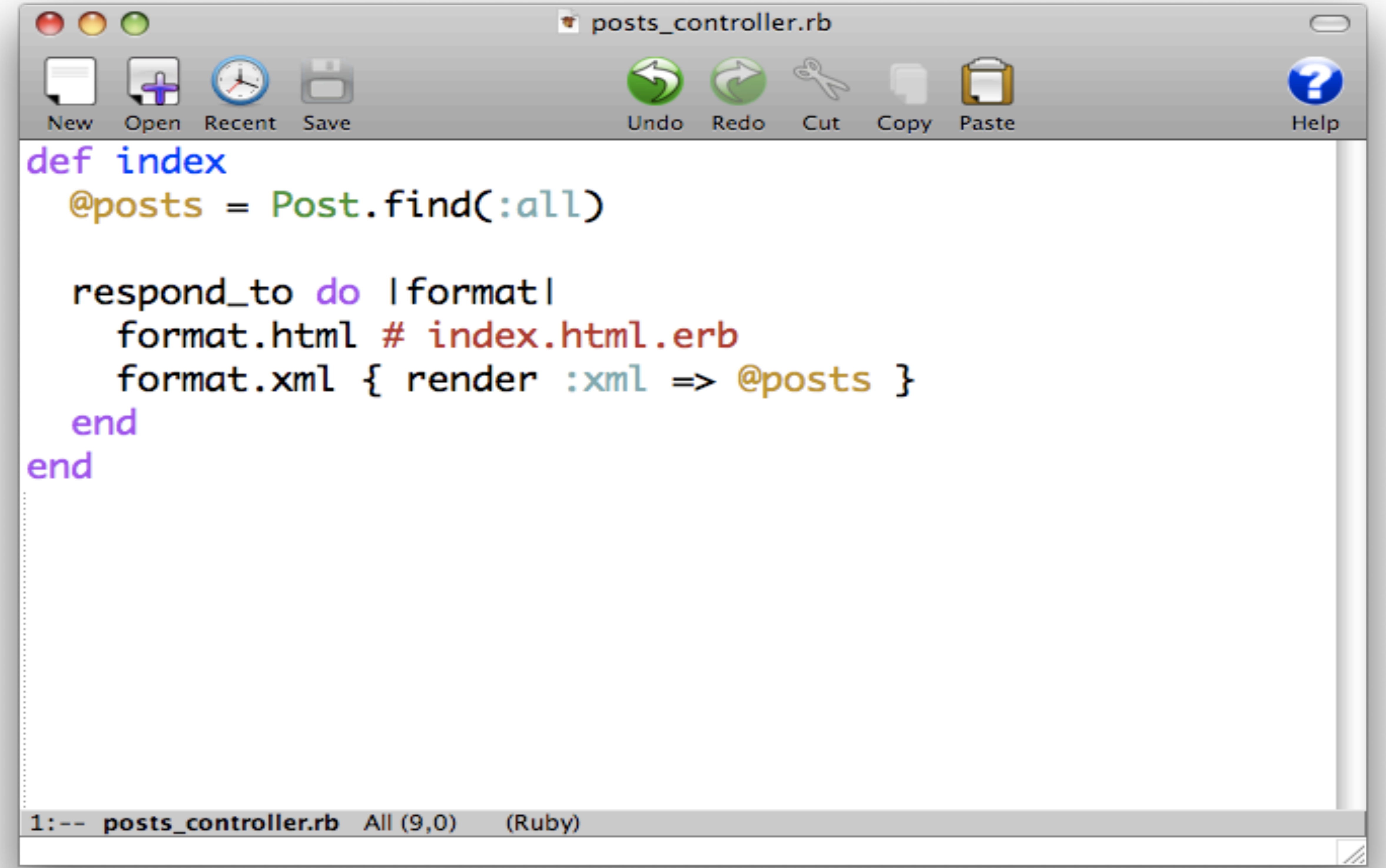
```
<body>
<h1>New comment</h1>
<% form_for([@post, @comment]) do |f| %>
  <%= f.error_messages %>

  <p>
    <%= f.label :commenter %><br />
    <%= f.text_field :commenter %>
  </p>

  <p>
    <%= f.label :body %><br />
    <%= f.text_area :body %>
  </p>

  <p>
    <%= f.submit "Create" %>
  </p>
<% end %>

<%= link_to 'Back', post_comments_path(@post) %>
</body>
```

```
1:--  show.html.erb   All (16,7)    (HTML Spc Fill)
```

Ruby on Rails – Controller Layer

- Action Controller 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.

Ruby on Rails – Controller Layer (2)



```
def index
  @posts = Post.find(:all)

  respond_to do |format|
    format.html # index.html.erb
    format.xml { render :xml => @posts }
  end
end
```

```
1:-- posts_controller.rb   All (9,0)      (Ruby)
```

Ruby on Rails – Other Components

- Action Mailer
  - Framework for building e-mail services.
- Active Resource
  - Framework for managing the connection between business objects and RESTful web services
- Action Web Service
  - Server-side support for SOAP and XML-RPC protocols in Rails applications

Ruby on Rails – Deployment

- Many Web Servers and hosting options
- WEBrick bundled with Rails
- Other options include Apache (with mod_rails or FastCGI), Mongrel, nginx, lighttpd, etc.
- Dedicated Rails hosting companies: Rails Machine, Engine Yard, etc.

Topic 4 – MVC Framework Comparison

- There are LOTS of web application frameworks
- Picking which one to use is not trivial
- Many factors come into consideration when picking a framework
    - Familiarity with programming language
    - Legacy Code
    - Easy of Use
    - Documentation
    - Fun Factor!

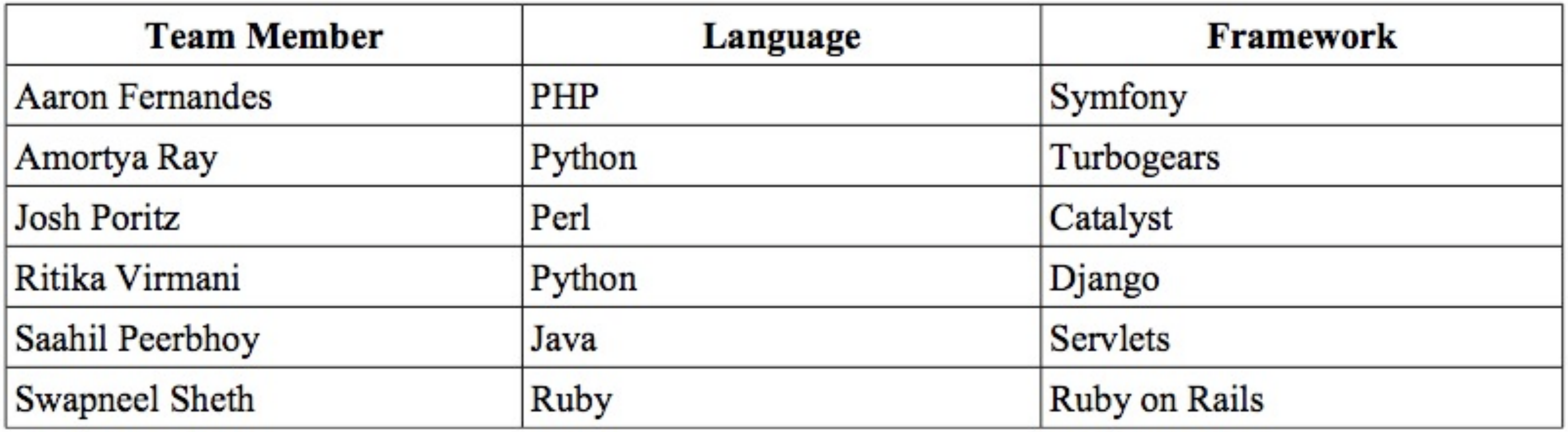
Six Degrees of Separation

- Project done for WHIM in Spring 2007
- Basic Idea
  - Implement the exact same web application in 6 different frameworks
  - Compare the frameworks on criteria such as
    - Lines of Code, Number of Methods
    - Performance Benchmarks like throughput, latency, cpu and memory usage

Six Degrees of Separation

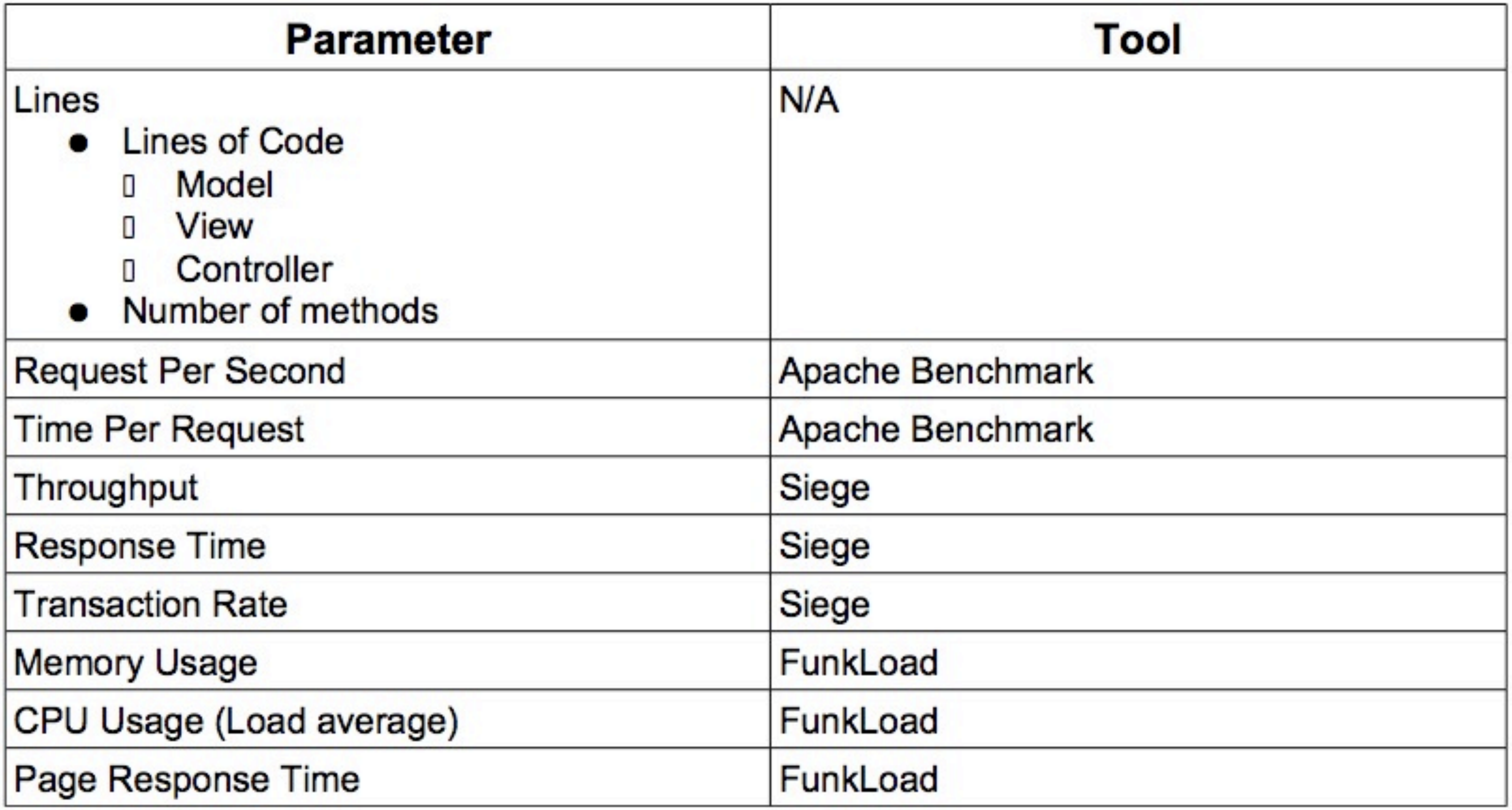
- Phase 1
  - Build a CRUD application for creating Music Catalogs
  - Application should have only basic features like Searching and Sorting
- Phase 2
  - Benchmark using Apache Benchmark, Siege, Funkload

People



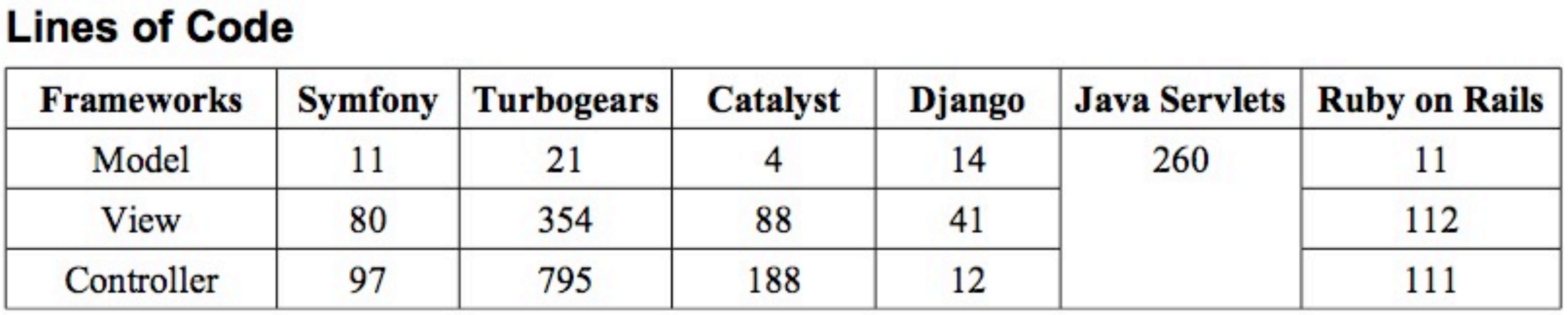
Team Member	Language	Framework
Aaron Fernandes	PHP	Symfony
Amortya Ray	Python	Turbogears
Josh Poritz	Perl	Catalyst
Ritika Virmani	Python	Django
Saahil Peerbhoy	Java	Servlets
Swapneel Sheth	Ruby	Ruby on Rails

Benchmarks



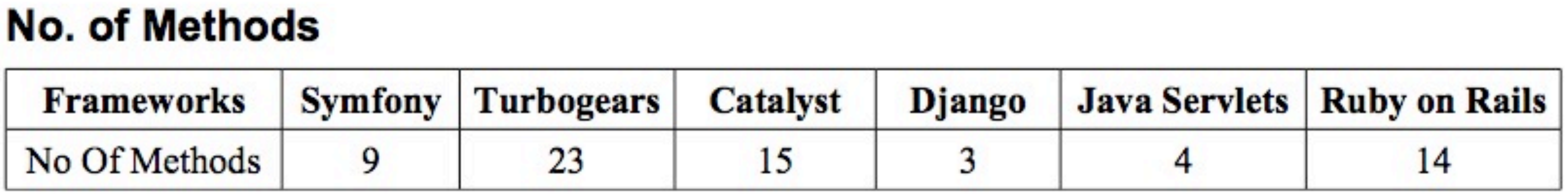
Parameter	Tool
Lines ● Lines of Code ▯ Model ▯ View ▯ Controller ● Number of methods	N/A
Request Per Second	Apache Benchmark
Time Per Request	Apache Benchmark
Throughput	Siege
Response Time	Siege
Transaction Rate	Siege
Memory Usage	FunkLoad
CPU Usage (Load average)	FunkLoad
Page Response Time	FunkLoad

Benchmark Results – Lines of Code



Lines of Code

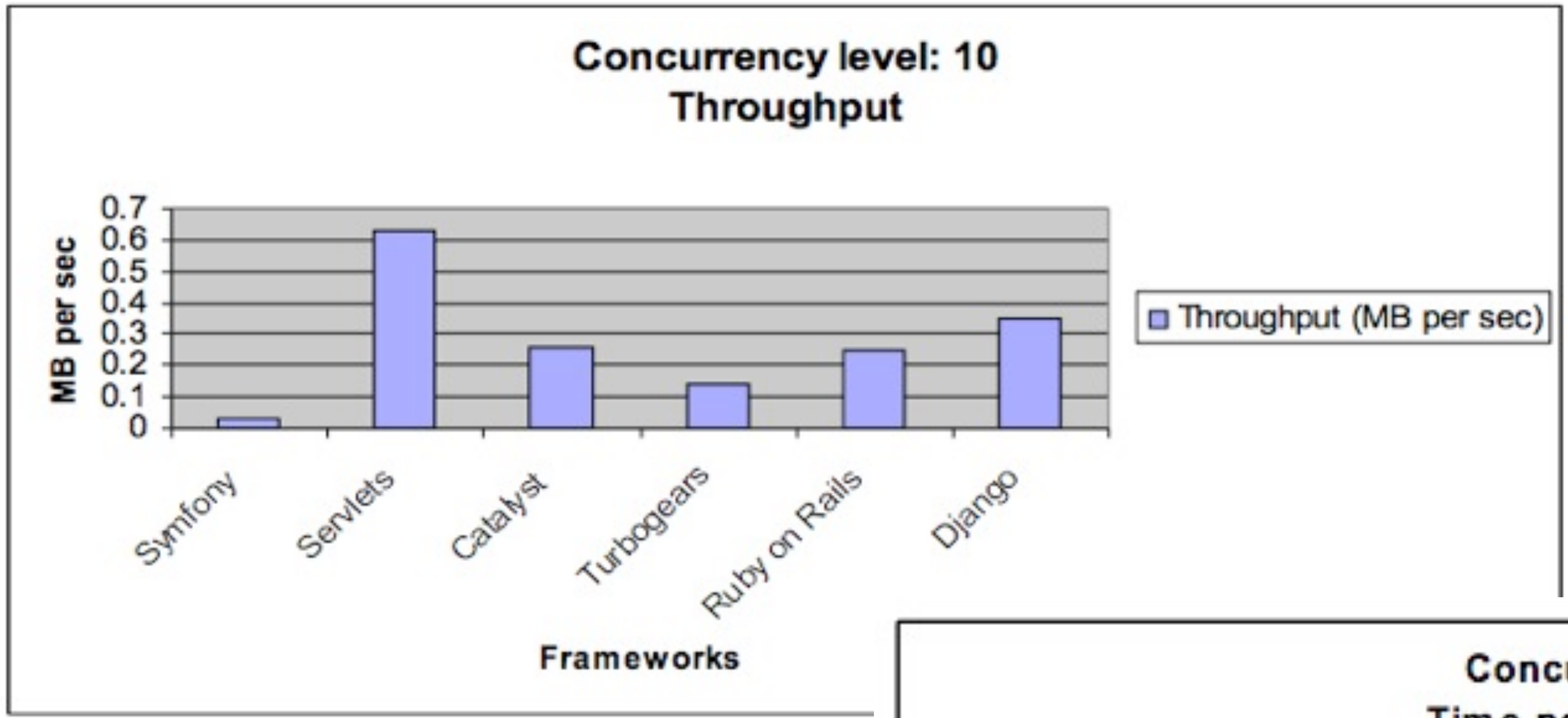
Frameworks	Symfony	Turbogears	Catalyst	Django	Java Servlets	Ruby on Rails
Model	11	21	4	14	260	11
View	80	354	88	41		112
Controller	97	795	188	12		111

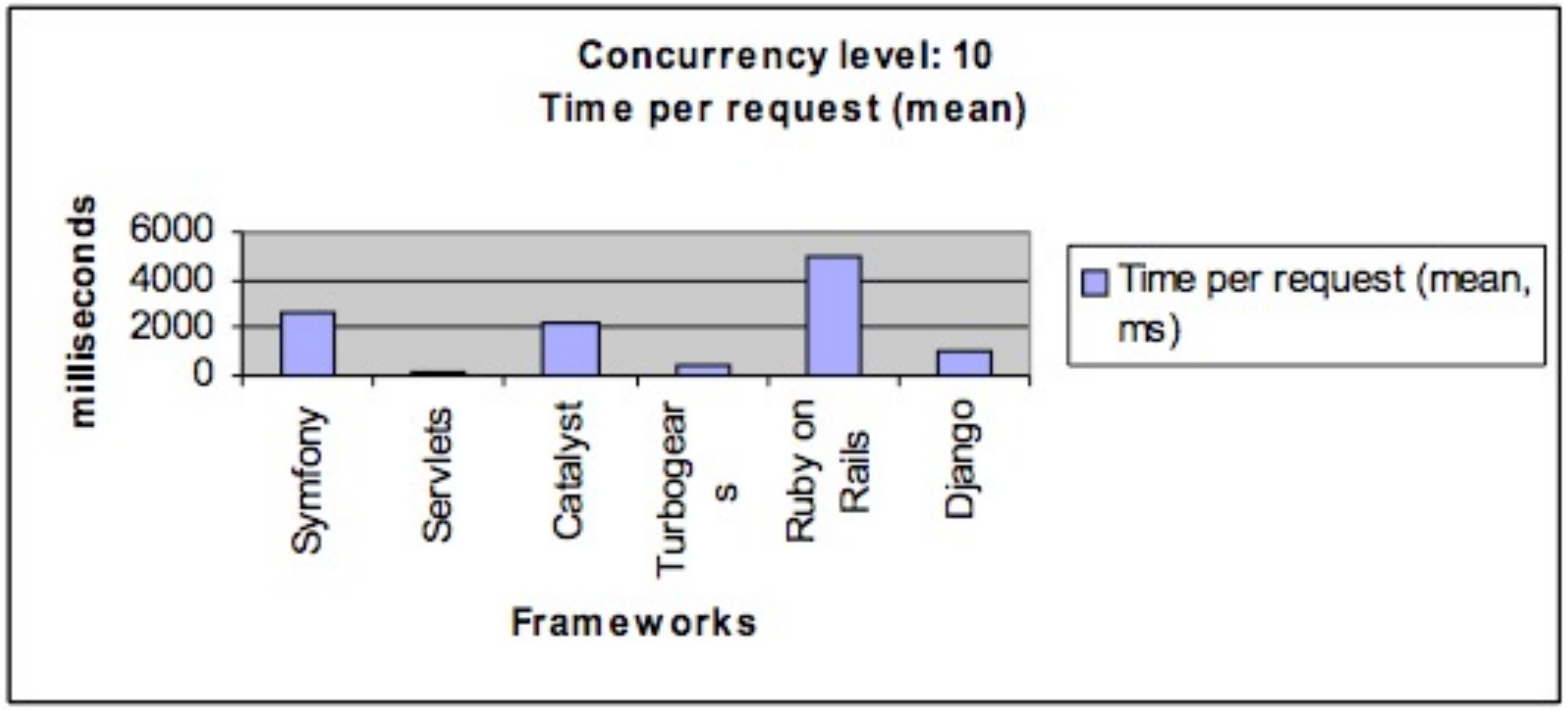


No. of Methods

Frameworks	Symfony	Turbogears	Catalyst	Django	Java Servlets	Ruby on Rails
No Of Methods	9	23	15	3	4	14

Benchmark Results – Throughput, Latency





Benchmark Results – CPU Usage

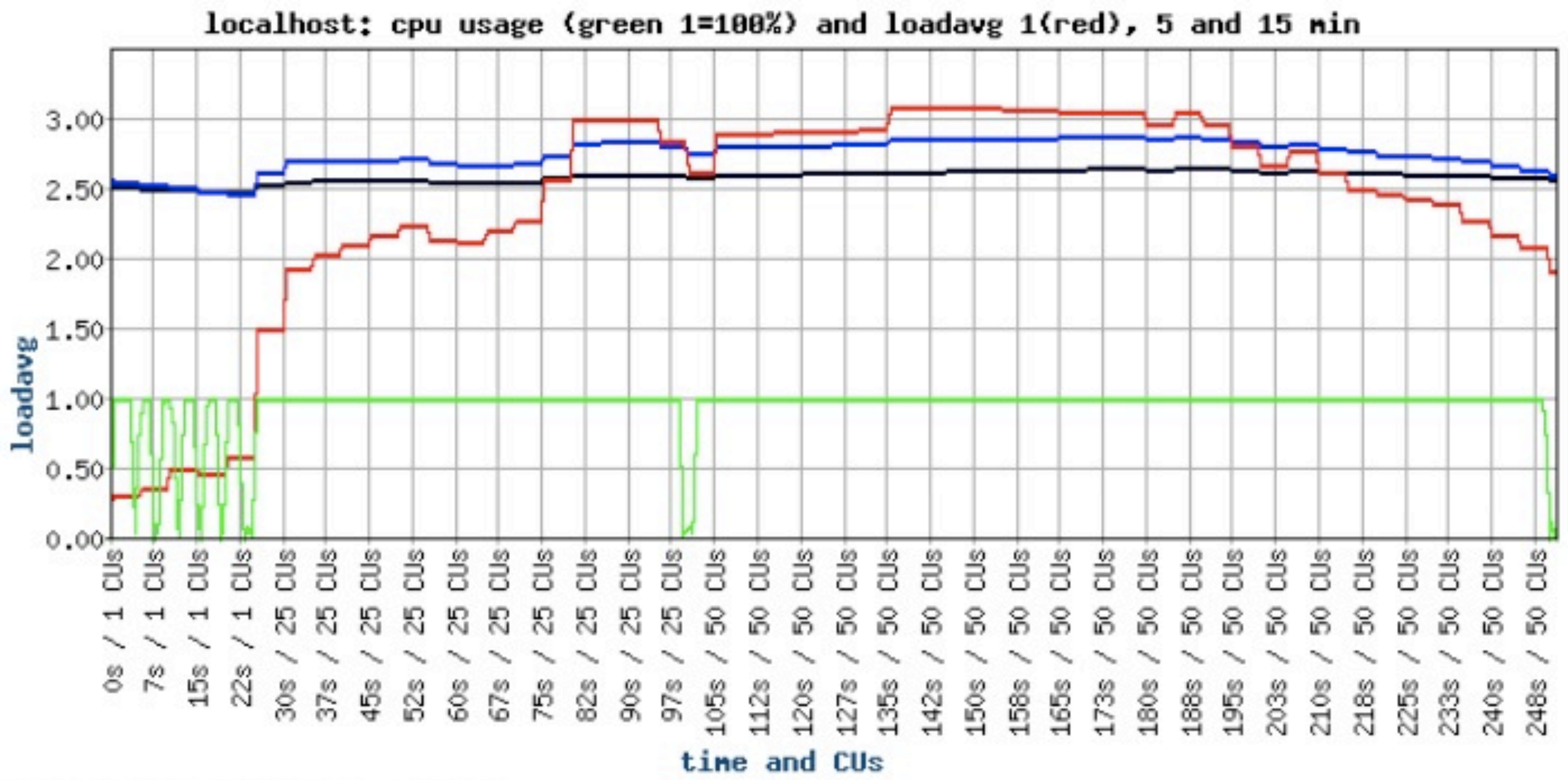


Illustration 7: Ruby on Rails

Benchmark Results – CPU Usage

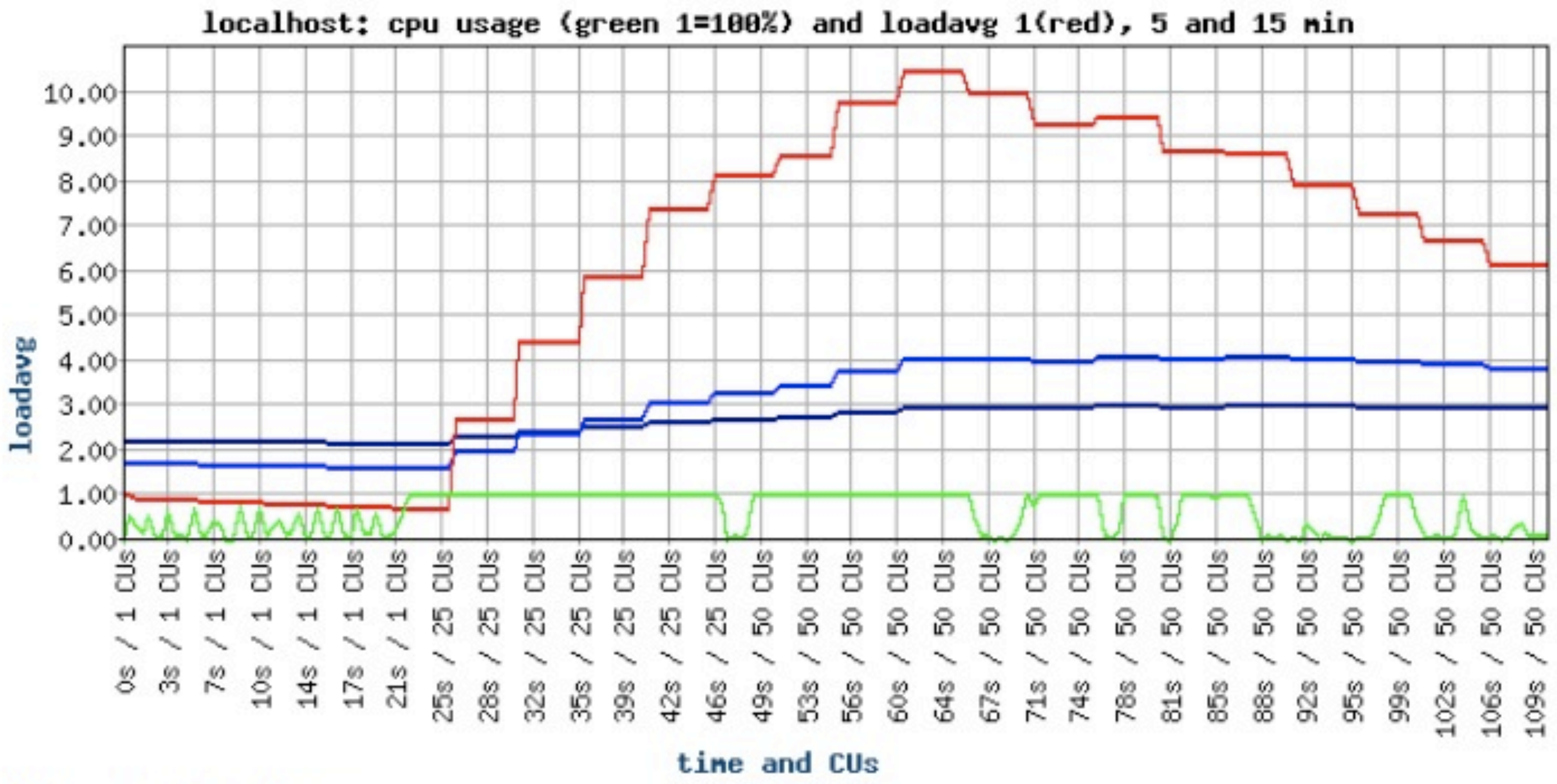


Illustration 5: Django

Benchmark Results – CPU Usage

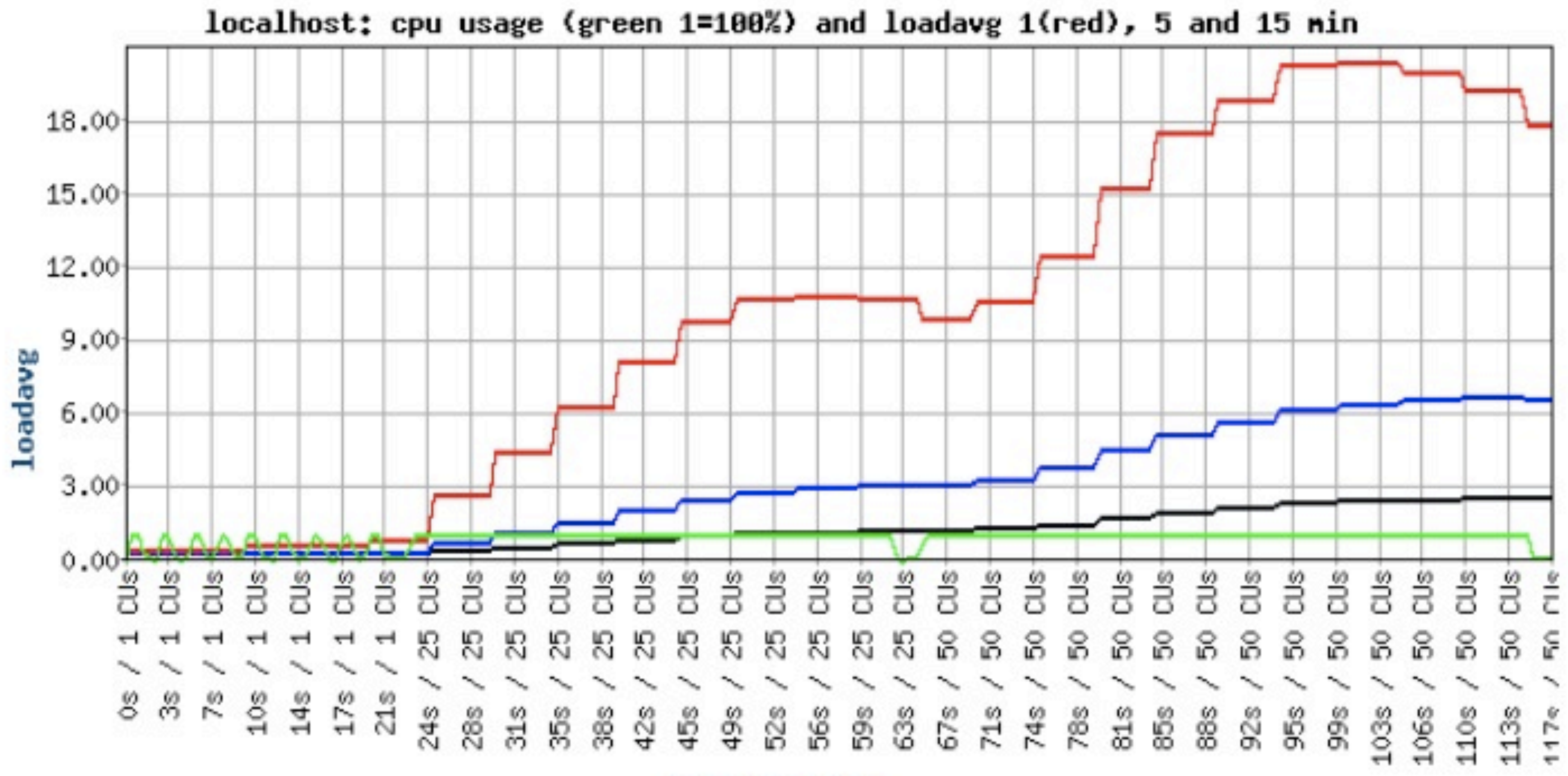


Illustration 2: Symfony

Benchmark Results – Memory

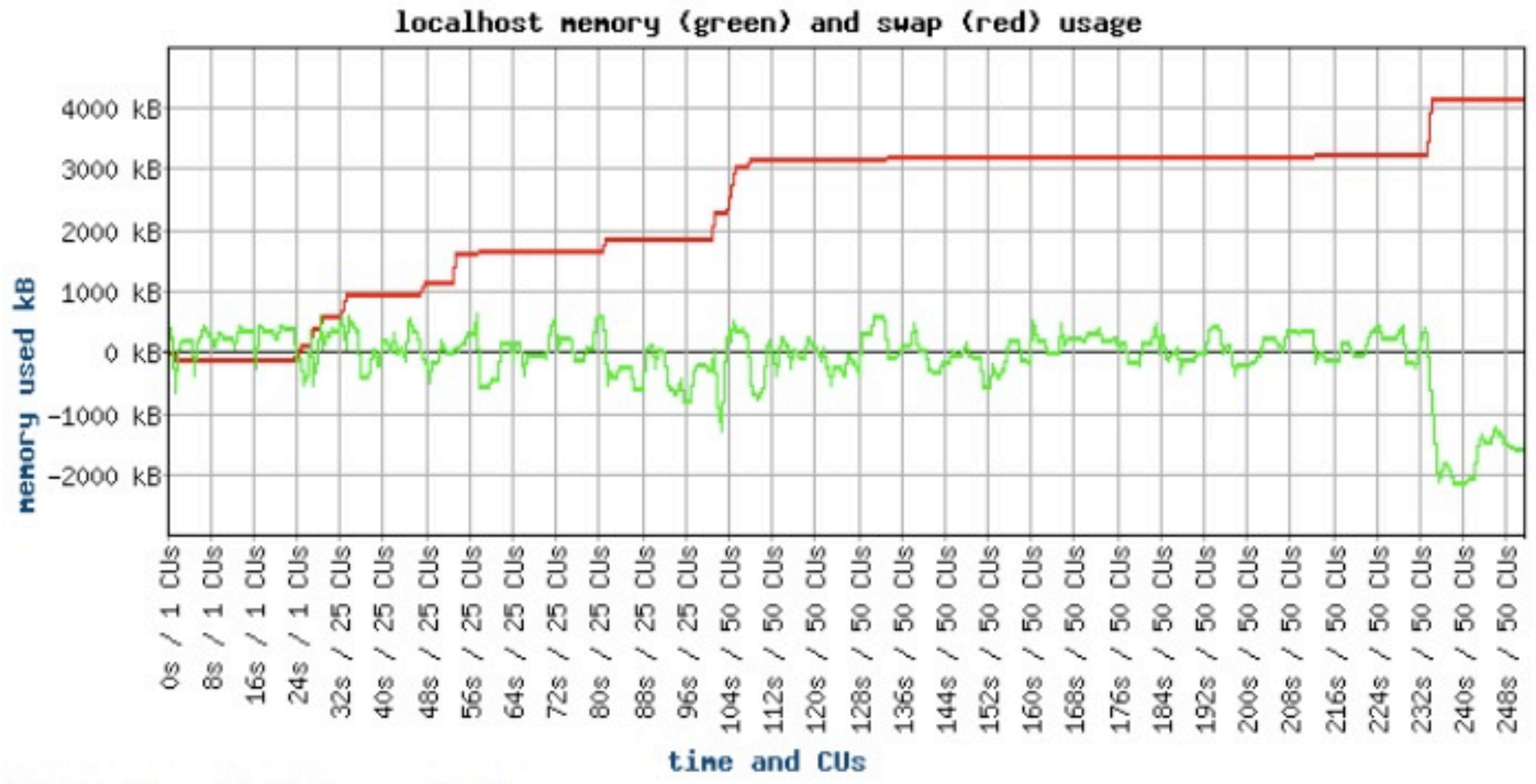


Illustration 13: Ruby on Rails

Benchmark Results – Memory

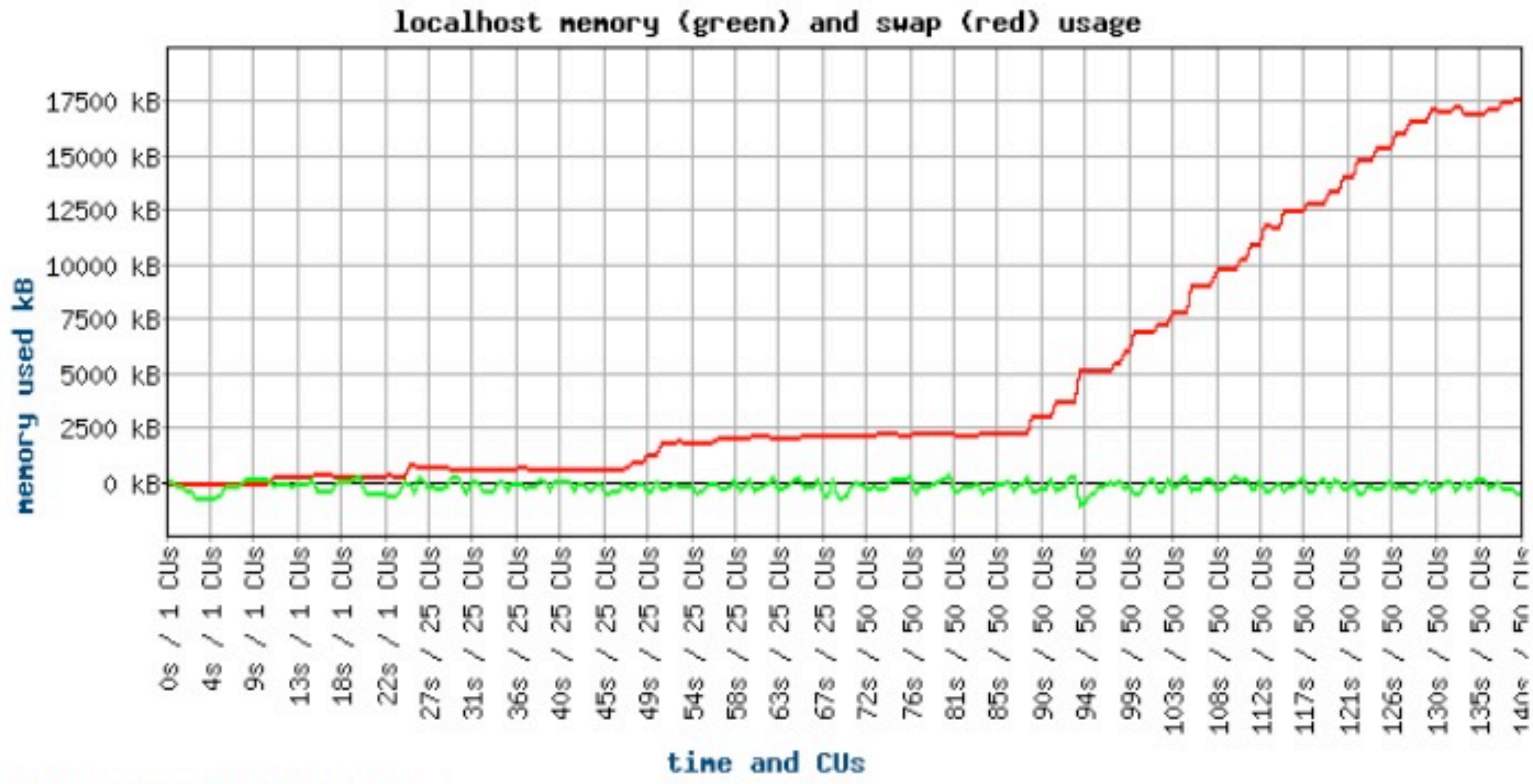


Illustration 10: Catalyst

Benchmark Results – Memory

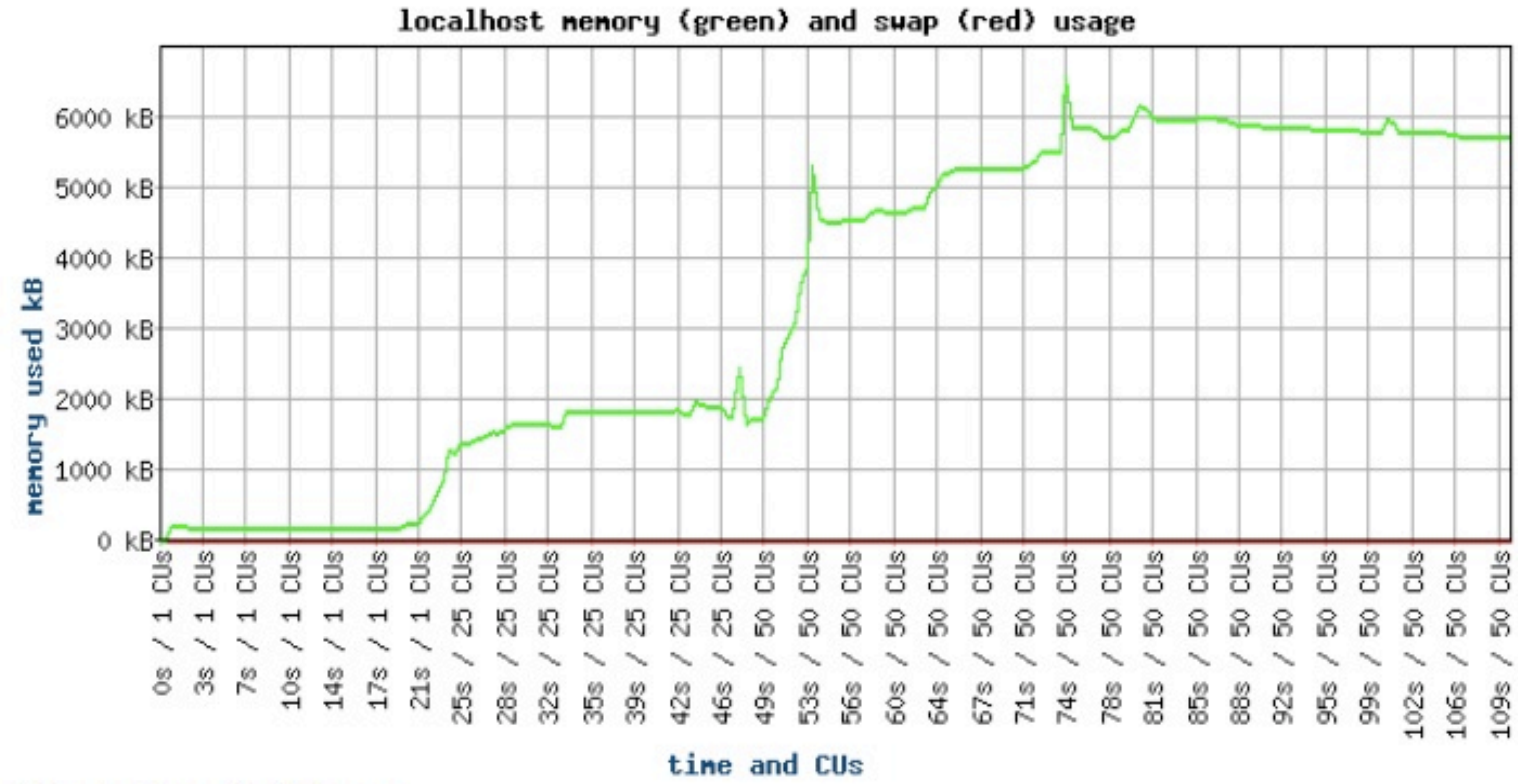


Illustration 11: Django

Better Web App Development

- Screencast by Sean Kelly
- Sean Kelly is a technologist at NASA's Jet Propulsion Laboratory
- Compares Java J2EE, Ruby on Rails, Zope/Plone, TurboGears, Django
- Link to Video

COMS E6125 Web-enHanced Information Management (WHIM)

Web Development Frameworks

Swapneel Sheth

swapneel@cs.columbia.edu

@swapneel

Spring 2012