

Master Project Report

# Teakwood: An Web Framework for Handling Many-task Computing

*Submitted in partial fulfillment of  
the requirements for the degree of*

**Master in System Science**  
in

Louisiana State University and Agricultural and Mechanical College  
The School of Electrical Engineering and Computer Science  
Computer Science and Engineering Division

by

---

Rui Guo

---

Under the guidance of  
**Jian Zhang**



Fall Semester 2014



## **Abstract**

Using Linux commands to handle computing jobs can be a hurdle to the scientific researchers who don't have HPC related background. Teakwood provides a solution and beyond. Teakwood is a framework that migrates all the terminal typing work to a web console GUI, and provides user a total control of their jobs, data, computing resources and so on just by clicking buttons. Teakwood is also an open platform that enables user to work co-operatively. Through Teakwood, user can share their models, results, and computing resources within their group and have discussion in Teakwood forum. Teakwood is powered by Django.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Teakwood . . . . .	1
1.3	Feature . . . . .	2
<b>2</b>	<b>Teakwood System</b>	<b>3</b>
2.1	Overview . . . . .	3
2.2	Frontend . . . . .	3
2.3	Backend . . . . .	3
2.4	Data handling . . . . .	3
2.5	Remote Configuration . . . . .	3
<b>3</b>	<b>Backend Mystery</b>	<b>5</b>
3.1	MTV framework . . . . .	5
3.2	Models and Database . . . . .	5
3.3	Django Template Language . . . . .	5
3.4	Request-Response Flow . . . . .	5
3.5	Lose Coupling . . . . .	5
3.6	Powerful Admin . . . . .	5
<b>4</b>	<b>Asynchronous Handling</b>	<b>7</b>
4.1	Celery . . . . .	7
4.2	RabbitMQ . . . . .	7
<b>5</b>	<b>Use Case</b>	<b>8</b>
5.1	Job Submission Flow . . . . .	8
5.2	Job Monitoring . . . . .	8
5.3	Job Report . . . . .	8
<b>6</b>	<b>Conclusion</b>	<b>9</b>

<b>7</b>	<b>Future Work</b>	<b>10</b>
7.1	Docker Hub . . . . .	10
7.2	Visualization . . . . .	10
7.3	Computing on the GO . . . . .	10
	<b>Acknowledgements</b>	<b>11</b>
	<b>References</b>	<b>12</b>

# List of Figures

2.1	Figure 2.1: Caption here	3
2.2	Figure 2.2: Caption here	4
3.1	Figure 3.1: Caption here	6
6.1	Figure 6.1: Caption here	9

# Chapter 1

## Introduction

### 1.1 Motivation

Four years ago, I know nothing about Linux. I then got a job to deal with HPC(High Cerformance Computing) system. Suddenly I jumped in to an UNIX environment. I found all those GUIs(Graphic User Interfaces), which I used to when I use Window OS, are gone, and I have to use Linux commands to control those machines and manage my stuff, which makes me uncomfortable and cumbersome. With the time goes by, I can work with Linux comfortably now. however, I still have the tendency to use GUI.

I am not the only one that had such experience, In my working building, a lot of Scientists and researchers are suffering this pain. Further, cross-platform work adding them many wasteful jobs. For example, scientist A want to share a computing result for researcher B to do visualization. A uses Linux and B uses Window. What should they do? command, command, command, copy, paste, command, command ...

Why not create a GUI platform to reduce those repeated typing work and allow them to work cooperatively?

With all those desires, Here comes the Teakwood framework.

### 1.2 Teakwood

Teakwood is GUI framework that allows user to summit HPC jobs from its web console, and to have a full control of their job status. Teakwood also provides a file management systems for user to organize their job data easily. What's more, Teakwood enables user to work cooperatively by sharing their result, models and computing resources within their group.

Teakwood migrates all the terminal typing work to Teakwood GUI, enables

user to submit the HPC jobs just by simply clicking functional buttons. Teakwood is a web framework, which means user can access from any type of machine as long as the machine has a browser.

## 1.3 Feature

Functionally, Teakwood have the following features:

- **Perfect documentation**

Teakwood homepage provides diverse software information including installation guide, user manual, developer manual, video tutorial, etc. user can easily fetch info and grasp Teakwood soon.

- **Extensible deployment**

Teakwood is loose coupling designed. User can easily add or remove features, models or even python functions without mess the whole system.

- **Neat GUI**

A neat LSU style web console makes your work simple as Windows. Drag, push, click, that's it!

- **Job monitor System**

Job monitor system provides five labels: uploading, queued, running, finish, and Data ready to let the user monitor the job status. Job monitor system also periodically pulls the running message and displays it in the console, user can know more details about their job.

- **Project management system**

All user's project is well organized and web kept in a file server. user can compare, share, and download them as needed.

- **Powerful admin**

The powerful admin system is provided by Django itself. with tiny System configuration, use can activate their models.



# Chapter 2

## Teakwood System

### 2.1 Overview

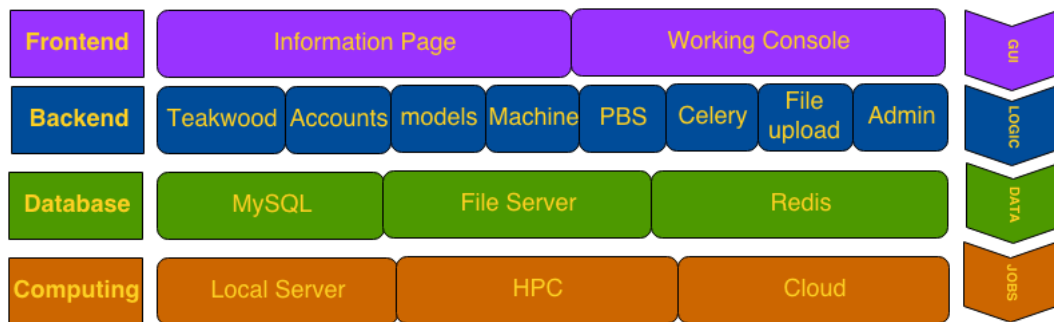


Figure 2.1: Teakwood System Overview

### 2.2 Frontend

### 2.3 Backend

### 2.4 Data handling

### 2.5 Remote Configuration

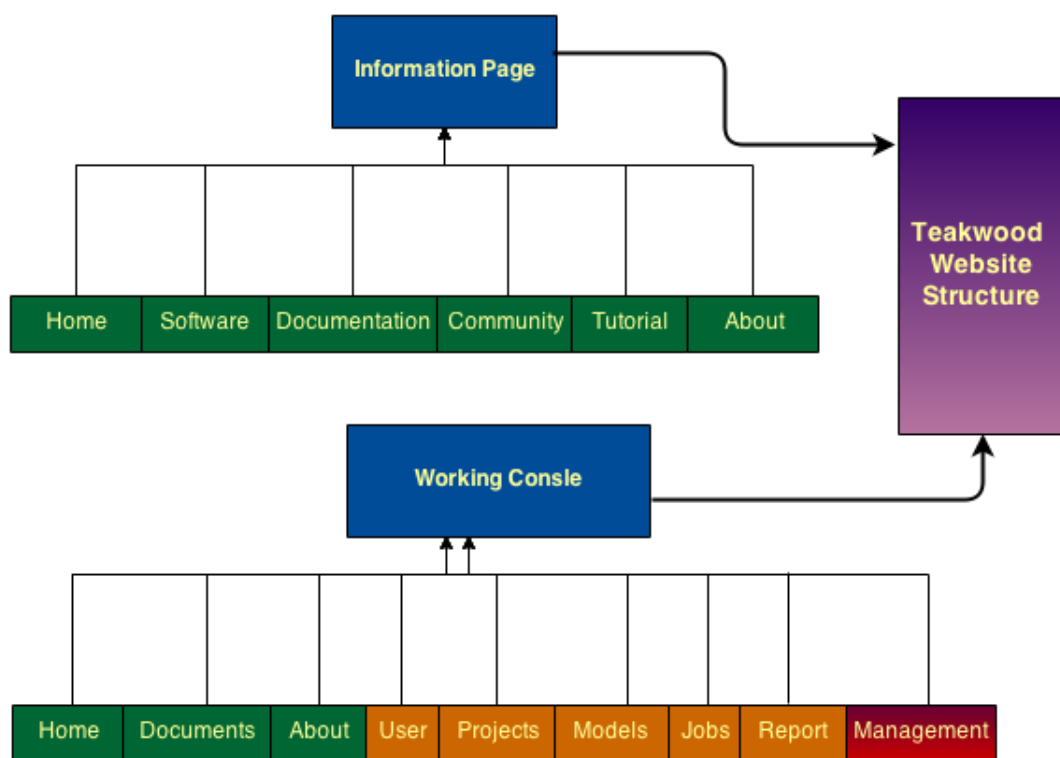


Figure 2.2: Website Strucutre

# Chapter 3

## Backend Mystery

### 3.1 MTV framework

### 3.2 Models and Database

### 3.3 Django Template Language

### 3.4 Request-Response Flow

How Teakwood process a request from user?

### 3.5 Lose Coupling

### 3.6 Powerful Admin

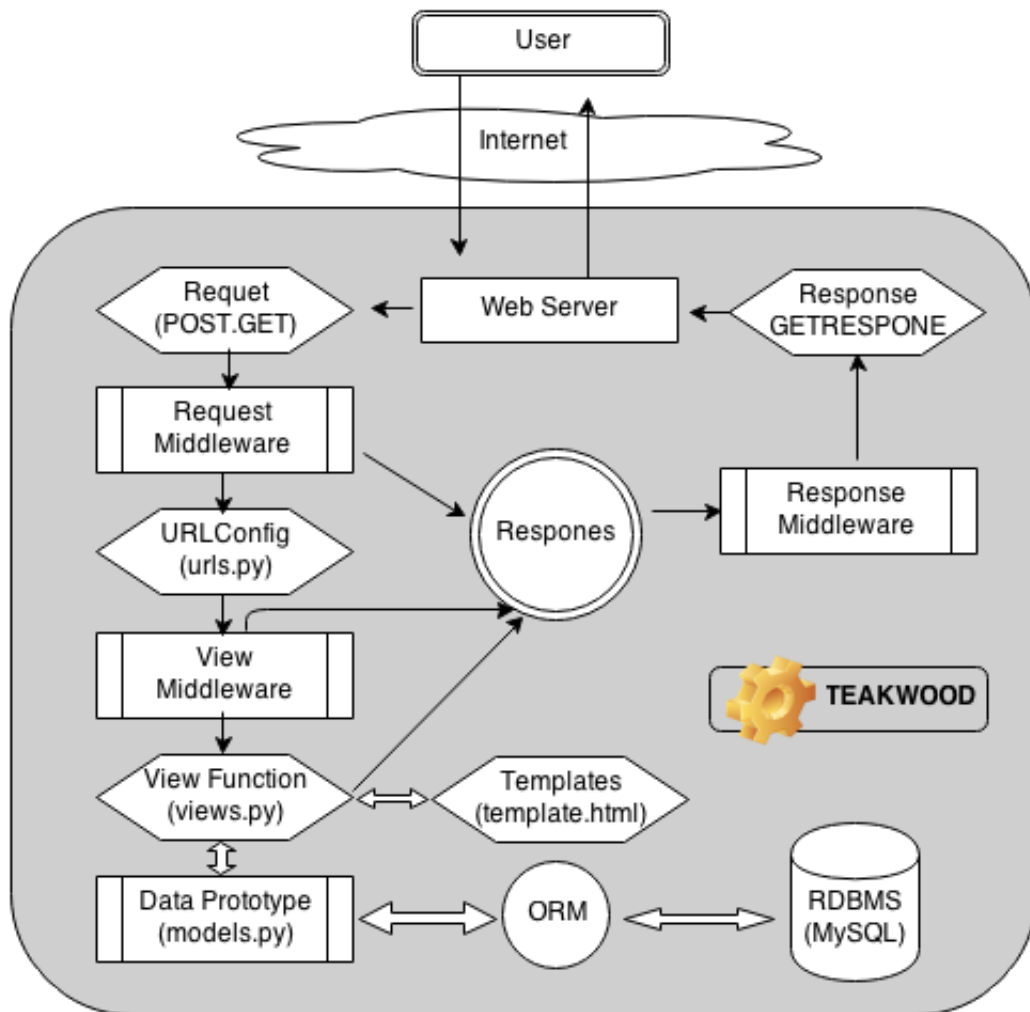


Figure 3.1: ¡Caption here!

# Chapter 4

## Asynchronous Handling

### 4.1 Celery

### 4.2 RabbitMQ

# Chapter 5

## Use Case

5.1 Job Submission Flow

5.2 Job Monitoring

5.3 Job Report

# Chapter 6

## Conclusion

How Teakwood process a request from user?

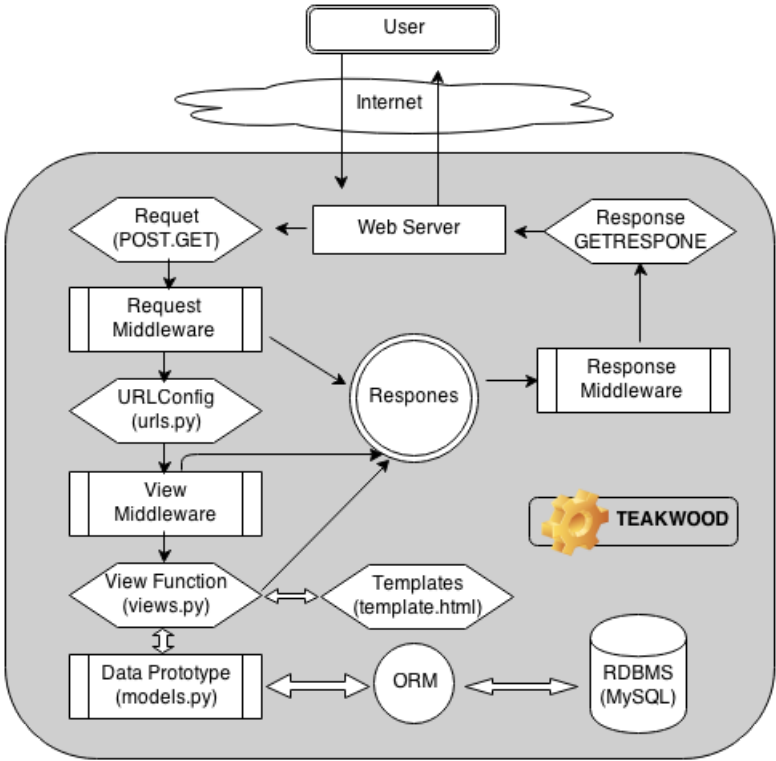


Figure 6.1: ¿Caption here¿

# Chapter 7

## Future Work

7.1 Docker Hub

7.2 Visualization

7.3 Computing on the GO



# Acknowledgments

¡Acknowledgements here¿

¡Name here¿

¡Month and Year here¿  
National Institute of Technology Calicut

# References

- [1] ;Name of the reference here;, <urlhere>
- [2] ;Name of the reference here;, <urlhere>