Web Application Frameworks

Aishwarya, Kavya, Rohit, Siri, Somdeep

Objectives

Compare 4 web application frameworks

- nodejs, rails (ruby), php, flask (python)
- Understand the merits and demerits of the various frameworks
- Attempt to rationalize the frameworks used by well-known organizations

Performance

Test Tools

-- Apache Test bench

-- Python script

Parameters

- #Requests
- Concurrency
- Server side computation
- Accessing mongoDb

Security

1) Script Injections - xss
2) Request Forgery - CSRF, OSRF
3) Cryptographic Security - Sessions, passwords
4) Cookie Security - Setting it as HTTP only

Development Issues

- Documentation : all 4 frameworks

– Libraries & Package Management :

- crypto, mongodb
- ruby-gems, npm, pip

– Project Scalability :

- Integration with REST
- Easy deployment - aws, apache, etc
- complex routes, application logic

– Development time :

- Php - high development time

Comparison

	Performance	Security	Libraries & Package Mgmt	Project Scalability	Development Time
node	1 + * + 1 + *	2	2	3	1
php	4 + 4 + 3 + 3 (14)	0	2	0	0
rails	2 + 2 + 2 + 1 (7)	4	2	3	1
flask	3 + 1 + 4 + 2 (10)	2	2	3	1

Analysis

Node:

- Advantages:
    - Good documentation, easy integration with cloud services, growing user-base
    - Good performance - only asynchronous though!
    - One Language to Rule them all - JavaScript
- Disadvantages:
    - Does not work well with large number of linear requests
    - Performance hit with synchronous operations - encryption

Rails:

- Advantages:
    - Offers best security - convention over configuration wins here!
    - Time to develop is less, availability of packages, support are high
    - Concurrency seems to be good - probably because of Apache
- Disadvantages:
    - Poor server-side performance - Interpreted language
    - DB interaction - poor
    - Does not scale

Analysis contd...

Php:

- Advantages:
  - Performance is better - Bytecode/ compiled modules
  - Documentation - good
- Disadvantages:
  - TIme to develop - very high
  - Learning curve is steep
  - Difficult to integrate with new database systems, cloud services
  - Extremely Insecure

Flask:

- Advantages:
  - Reasonable security - not as good as rails though!
  - Time to develop is less; availability of packages, support are high
  - Very efficient server-side computation
- Disadvantages:
  - Poor Concurrency - GIL
  - DB interaction - poor

Use Cases

Rails	● Twitter - initially,Now Twitter is moving to Scala - JVM ● Popular among startups
Php	● Facebook - HipHop ● Startups don’t favor - they don’t need performance, yet!
Flask	● Pinterest - fewer lines of code, less time to production, easy to setup, scales reasonably ● Good choice for startups
Node	● Netflix - UI engineer’s choice, well suited for Microservices, all work in JS, ● Scales well

Thank You!