Performance Evaluation of Web Application Frameworks

Student: Jason Wong, Advisor: Simanta Mitra

Project Abstract

The market of web application frameworks is flooded with so many choices. The objective of this project is to compare a few popular server-side web application frameworks and evaluate them against a set of "performance" standards.

Project Objectives

- Perform market research to identify top web application features
- Compare web frameworks based on the top web application features

Methods

- Gathered current research and findings
- Conducted user and professional surveys
- Developed and performed independent testing

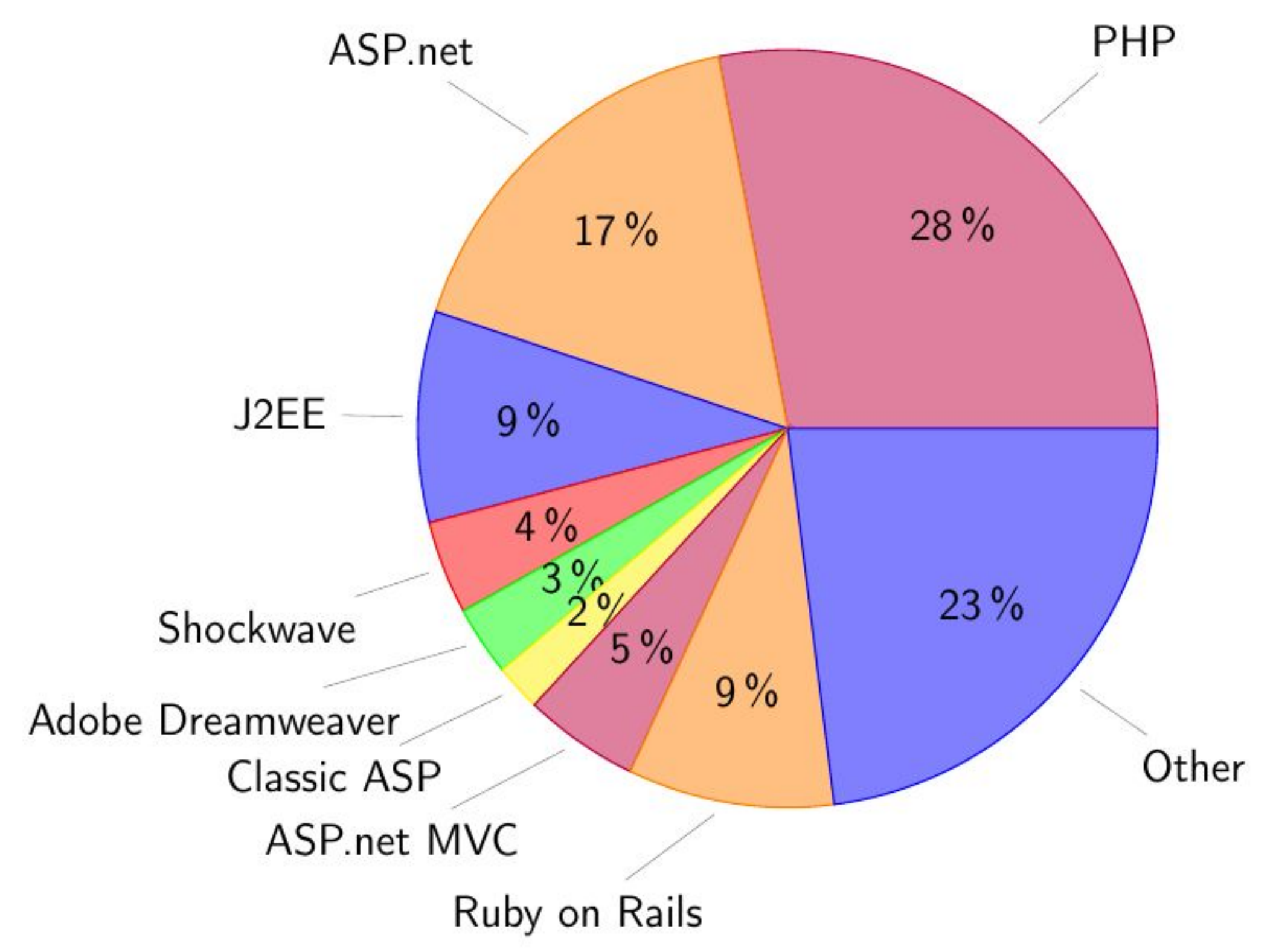
Performance Measures

The performance measures were chosen by surveying 100 Iowa State students who are familiar with web development. The students were asked to rank the importance of various web application features and these were the results:

- Database Access (27% top choice)
- JSON Serialization (REST API) (25% top choice)
- High-Traffic Capability (20% top choice)

Figure: Framework Language Usage Statistics

This pie chart demonstrates the fragmentation of programming languages used to create the web applications for the top 10,000 websites. Choosing a language is only one piece of the puzzle as there are still choices of frameworks within the same language, client-side frameworks, database type, server type, server hardware, etc.



Results Summary

- Frameworks drastically reduce web application performance when compared to their raw language
- Top websites improve traffic handling by using server load balancers
- To see a full report of my results, visit the URL encoded in the QR code below:

Figure: JSON Serialization Performance

Each response is a fresh JSON object serialization that maps the key "message" to the value "Hello, World!" Results were gathered using the open-source TechEmpower Benchmark framework. Notice that raw PHP performed around 11 times better than the PHP framework called CodeIgniter.

Framework	Response/Sec
ulib	2,155,630
Gemini	972,194
go	477,681
NodeJS	226,031
PHP	171,149
Compojure	155,840
Spring MVC	98,304
CodeIgniter	15,612
ASP.net	3047
Symfony2	1,130

Figure: Database Access Performance

Each database access fetches a single row from a simple database table. Results were gathered using the open-source TechEmpower Benchmark framework. Notice that raw PHP performed around 8 times better than the PHP framework called CodeIgniter.

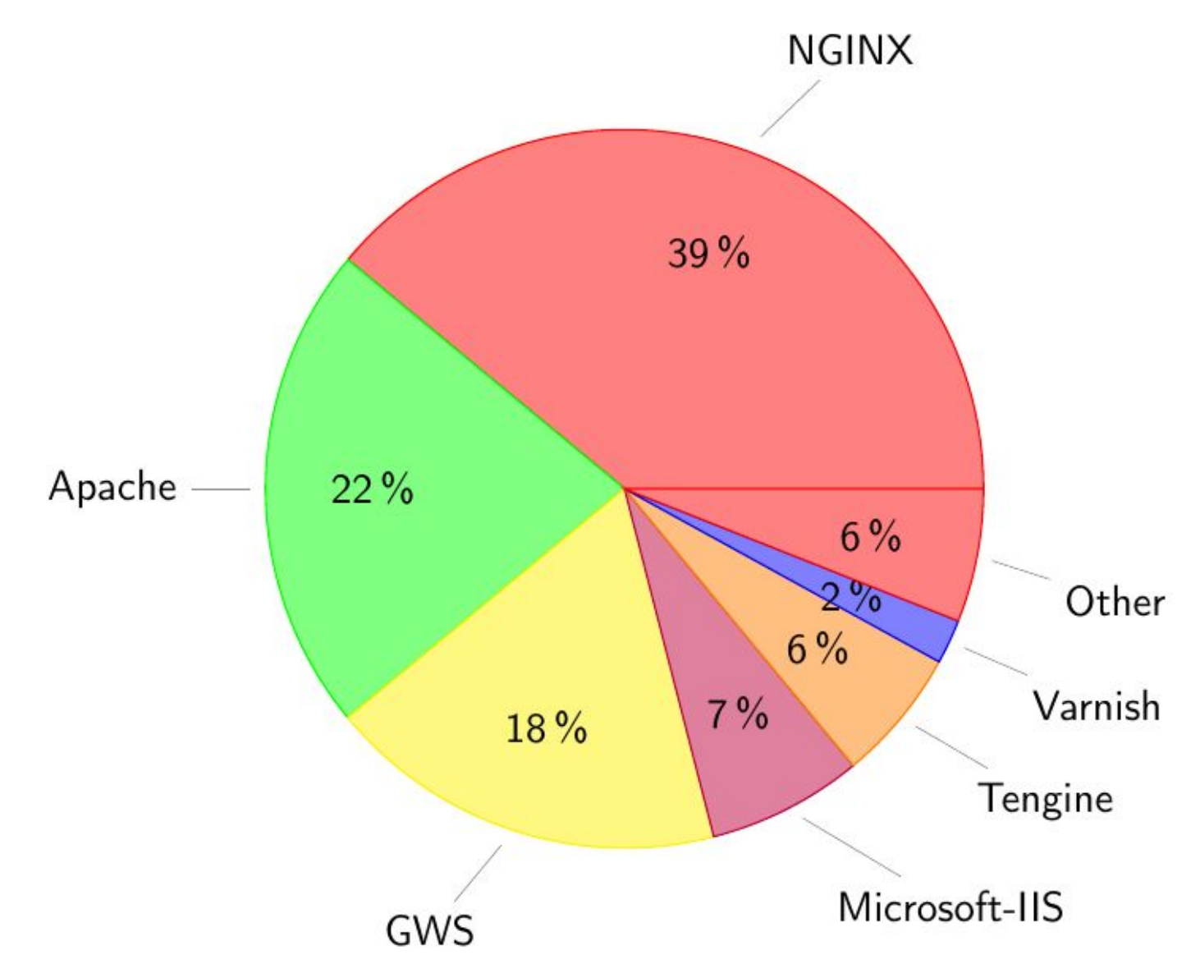
Framework	Access/Sec	Database	Language
uLib	323,820	Postgres	C++
go	201,710	MySQL	Go
Compojure	153,495	MySQL	Clojure
Gemini	147,817	Postgres	Java
PHP-raw	111,218	MySQL	PHP
Spring MVC	54,907	MySQL	Java
NodeJS	38,774	MySQL	JavaScript
NodeJS	31,531	MongoDB	JavaScript
CodeIgniter	14,035	MySQL	PHP
ASP.net	2708	MongoDB	C#
Symphony2	995	MySQL	PHP

Database Access: Independent Testing

Being skeptical of the results above, I developed my own database testing software. In this case, I compared raw PHP to a PHP framework known as Laravel. I used a single query to a local database requesting a single row. In my test of 1000 queries, raw PHP finished in 0.07µs whereas Laravel finished in 0.18µs. This shows a 2.5 times performance advantage for raw PHP.

Figure: Server Type Usage Statistics

This chart is a result of scanning the top 300 visited websites for server type. Nginx is clearly the most commonly used server. Nginx is a high performance HTTP load balancing server. Popular sites which use Nginx include: dropbox.com, yahoo.com, reddit.com, and tumblr.com.



Software Maintainability

- Despite reporting a worse performance, frameworks have the advantage of maintainability. Frameworks often follow a Model-View-Controller design pattern to create modular and easy to update web applications.
- Another benefit to using popular frameworks is the amount of documentation available. When using a custom framework, you rely on the past developers to comment the code and create documentation.

Conclusion

- In regards to performance, avoid the use of frameworks if at all possible. If it is unavoidable, use frameworks with the least amount of features to avoid unnecessary overhead and increase performance.
- The main benefit of frameworks is their maintainability.
- Server load balancing is needed for high-traffic web applications

Key Sources

- TechEmpower Framework Benchmarks. (n.d.). Retrieved April 08, 2016, from https://www.techempower.com/benchmarks/
- Framework technologies Web Usage Statistics. (n.d.). Retrieved March 07, 2016, from http://trends.builtwith.com/framework

Various Definitions

- Framework: a universal and reusable software codebase that provides basic functionality for web applications
- Server-side: a web development term which denotes the computer systems which host and handle the computing for the web application
- Client-side: a web development term which denotes the web browser the end user will use to communicate with the server-side web application
- JSON: a standard format that uses plain human readable text to transmit data
- REST API: a well defined interface to send/request data
- HTTP: an application protocol used for data communication across the world wide web