React v.s. Angular - A High-level Comparison of Two Modern JavaScript Frameworks

Chang Yan, Chao Huang, Hongyi Fan, Zhiyu Liu
Department of Computer Science
North Carolina State University
Raleigh, NC, USA
{cyan3, chuang23, hfan4, zliu48}@ncsu.edu

***Abstract*—In this project, we mainly compare two modern JavaScript Frameworks in a new dimension. In order to get more detailed result, we compare them for 4 parts, which are learning curve, resources, maintainability and running performance respectively. For the difficulty levels of learning two frameworks, we get a survey to know the feels of front-end programmers. For resources, we get the count of communities and related packages count for each one. For maintainability, we consider it more in architecture level, because it's more based on how we architect our project and how we develop it. For running performance, we write two same apps using React and Angular respectively, and them analyze them. Finally, we make a conclusion of it and give you some advice about how to choose a front-end framework**

***Index Terms*—React, Angular, Learning curve, maintainability, Performance**

I. Overview

The front end frameworks has been a battleground for JavaScript developers and the debate between the most few acknowledged frameworks has never ended. In this project, we chose the two most popular ones - React and Angular - to provide you with a holistic, high-level comparison, mostly from a developers perspective. The metrics we discuss here are learning curve, resources, maintainability and running performance. We use a survey to consider the learning curve. For resources, we counts the packages number and question number related two framework on stackoverflow. In order to analyze maintainability and performance later, we create the classical MVC apps twice using two frameworks respectively. There are lots of tools to evaluate the maintainability and code smell. Here we use Plato and Eslint. Then we run them on the local server of same PC, and test the performance using Google Chrome developer tool.

We hope that this project could give you more sense of what are some of the aspects that you should consider when choosing a front end framework. The goal is to make you be aware of the difference and enable you to make a more confident choice when necessary. That said, our target audience are mostly web / software developers, or any people that are interested in web development.

II. Approach

A. Learning Curve

Our first concern is how easy and quick it is for developers to pick up some knowledge of a framework and to produce some decent work. The tool chain in React is high as compared to Angular where it is low. Angular has a high learning curve whereas React has a low which takes time for the developer to master initially. The data flow control in React is one way whereas in Angular it is two-way which makes it complex when dealing with the large application. Debugging sometimes can be the toughest thing to do. Angulars runtime debugging tends to provide less information than Reacts assemble time debugging. From statistical result shown on figure 1, the react is more popular than angular, which can partly implies that learning curve of React is more smooth than angular and people want to learn more about React. From an engineering

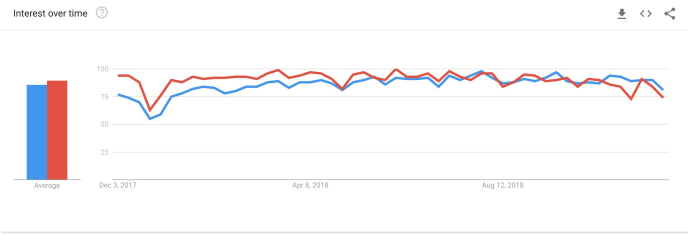

Fig. 1. interest comparison

perspective, this is crucial considering the cost of time and energy could be saved if one is significantly easier than another. According to a survey conducted in the current state of JavaScript in 2018, which is shown in figure 2 and figure 3, developers claim that React is relatively easier than Angular, but with an object-oriented background, one may find Angular more natural and easier to adopt.

B. Resources

The amount of resource reflects how the community, people, and related tools grow around a framework. The more active the community is, the more resource we have. And well consequently have more options to make when facing different issues. We mainly consider three sources here, which are stackoverflow, Github repositories and related Packages. The statistic result is shown on figure 4. When we searched the name of the two frameworks on Stackoverflow, therere 232,309 results tagged with React, nearly twice of that of Angular, about 142,068. Also, 71,247 packages related to React are found at the NPM registry, while 26,170 related to

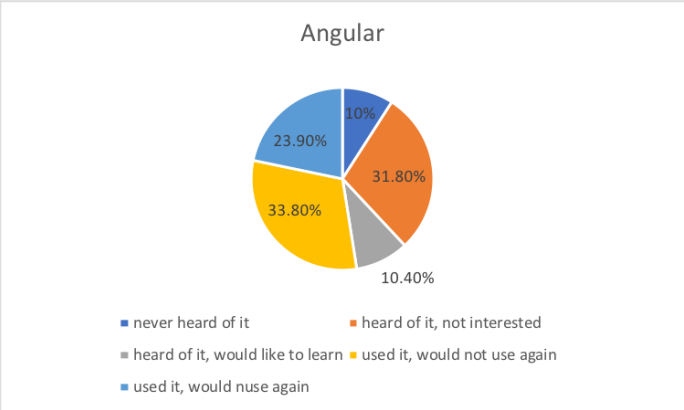

Fig. 2. survey for Angular

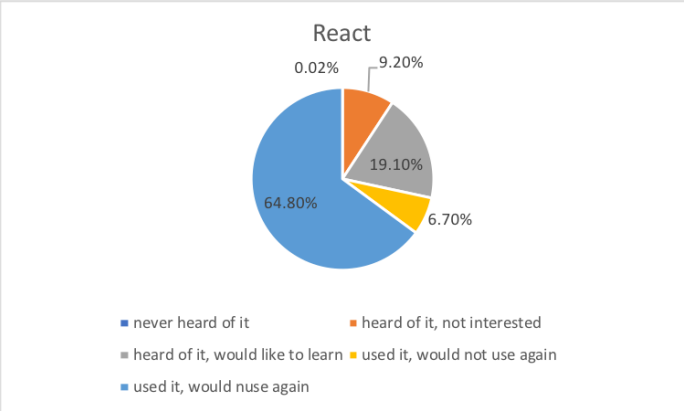

Fig. 3. survey for React

Angular. The open source repositories are also good resources for learning. For Github, we find that the repository number of React is 683,337, much more than Angular, about 449,522. According to the stats from these main sources, we know that React has more resources than Angular.

C. Maintainability

In 1991 Oman and Hagemeister at Idaho University designed the maintainability index to objectively determine the maintainability of a product based on the corresponding source code. We as developers, must be concerned about how well an application could scale. For example, with Angular or React, how to architect your application, how different kinds of tests are written to ensure robustness, and how linting tools are applied to enforce a consistent coding style to prevent smell. For the two apps, we use Plato , a JavaScript source code visualization, static analysis, and complexity tool, to test their maintainability first. The result are shown in figure 5 and figure 6. From result, we know that the app using Angular needs more lines of code than using React, but its average maintainability is a little bit higher. Then we use ESlint to test the code smell of them. In order to make sure our test fair, we use the Google code style as our criterion to evaluate both of them, and the results are shown in figure 7 and figure 8. From ESlint perspective, React is way better than Angualr. Taking those into consideration, we found that both Angular and React have reached to a mature level in that they provide easy-to-set-up scaffolding tools, generic linting tools, etc. But one significant difference is that React is more of a JavaScript library, rather than that Angular is an opinionated framework. React provides you the core essential part but leave you with multiple options for a topic, say, data management, like Redux, MobX, etc. But with Angular, usually you dont have options to choose.

D. Performance

Performance is critical when it comes to web development. It varies on many different factors. When it comes to framework architecture, its hard to measure the performance due to the following reasons:

- The heterogeneity of applications in terms of use cases, scale, etc
- The difficulty for calculating framework performance that is free of the business logic built on top of it
- Machine / network speed, etc

In order to make sure machine and network do not influence our results, we use same machine and local server to run our app. The machin information is in the table below. And we use node.js server as our local server, and run these two applications on it.

PC	*CPU*	*GPU*	*Browser*
MacBook Pro	2.3GHz	Intel Iris+640 1536MB	Chrome 70

We run each application 10 times and use chrome development tool to record the time line. Then we collect the loading time, scripting time, rendering time and painting time from the time line. These four parameters can clearly reflect the performance in different phases and we also can know which part is the most time-consuming. The speed performance are shown in figure 9 and figure 10. Whats more, we also consider the bundle size here, as shown in figure 11, because the size of Webpack is one of factors to influence the speed performance, and lower is better.

That said, the biggest take away is that the performance of application written in either Angular or React will become less dependent on what framework you have chosen, making it trivial when it comes to performance. So, from a developers perspective, youd better not choose one over the other from a performance perspective.

III. Conclusion

Both React and Angular are the top choices as for front end frameworks nowadays. They provides developers with considerably more abundant resource, including tutorials, related libraries, tools, to reason about. But React wins in terms of quantity and is more active in community It also has a more friendly learning curve, and provides more flexibility.

APPENDIX

https://github.com/terryliu1995/React-VS-AngularJs

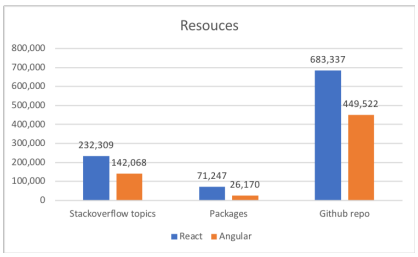

Fig. 4. resources comparison

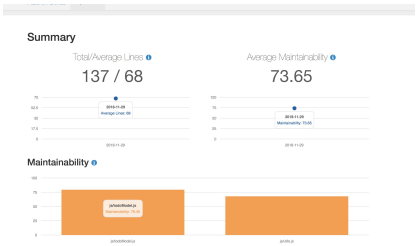

Fig. 5. Plato result for React

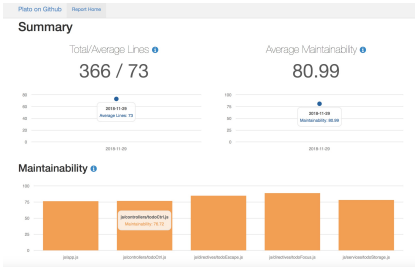

Fig. 6. Plato result for Angular

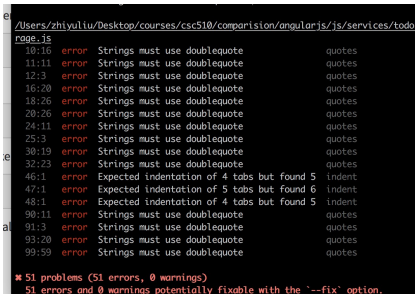

Fig. 7. ESlint result for Angular

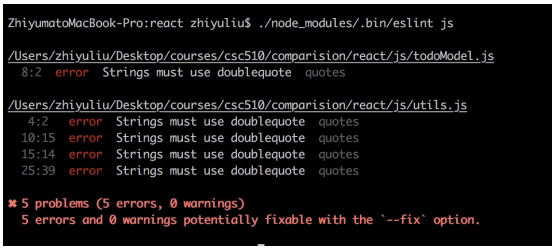

Fig. 8. ESlint result for React

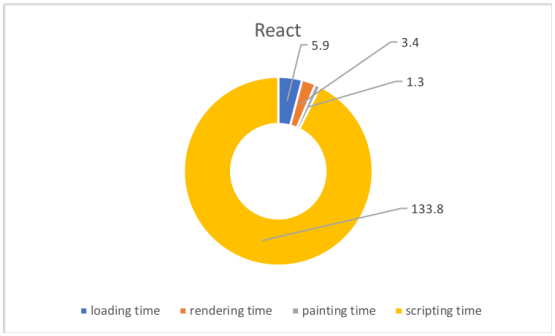

Fig. 9. React performance

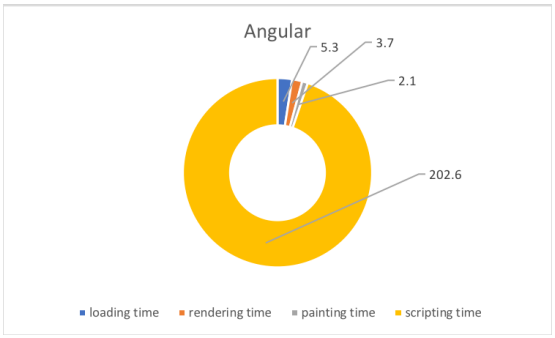

Fig. 10. Angular performance

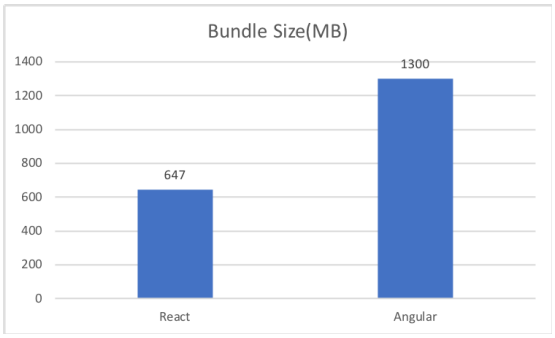

Fig. 11. bundle size comparison