maxwellbond

TRUSTED TECH TALKS

WEBINARS · EVENTS · NETWORKING

UNDERSTANDING REACT FIBER

A DEEP DIVE INTO REACT FIBER AND THE TWO MAIN PHASES OF THE NEW RECONCILIATION ALGORITHM.

JORDAN MATTHEWS

FRONTEND DEVELOPMENT CONSULTANT
AT MAXWELL BOND

KANNAN GANESAN

SENIOR SOFTWARE DEVELOPER
AT SIMPLY BUSINESS

INTRODUCTION

Kannan Ganesan, Senior Software Developer at Simply Business, lead our Trusted Tech Talk on Tuesday 6th October 2020, to present on React Fiber, what it does, and the benefits of implementation. You can now re-watch the full webinar and presentation online at www.maxwellbond.co.uk or read through the highlights overleaf. Please note that the original presentation was heavily visual and used many demonstrations, so please head over to our website to gain a fuller understanding of the topics and the full presentation experience.

WITH THANKS TO:

KANNAN GANESAN

at Simply Business for leading a great session.

Kannan has over 11 years of experience in Frontend development with various domains (E-commerce, Insurance and Financial) and over 6 years working with React. Highly proficient in ReactJS + Context API, ES6 / ES7, Node JS, Express, HTML5, CSS3, GraphQL + Apollo Typescript, Ruby on Rails, Mobile web android and iOS development, Agile Methodologies, CICD, TDD based development. He cares deeply about creating world-class, customer centric and ground-breaking products (UI/UX) that help businesses make a real difference. Now at Simply Business, he's joined a team who are all united by a passion for building technology, which is as innovative today as it will be reliable and trustworthy tomorrow – and enables the world's best commercial insurance experience. Simply Business has over 700,000 micro-business and landlord customers, covering over 1,000 types of business.

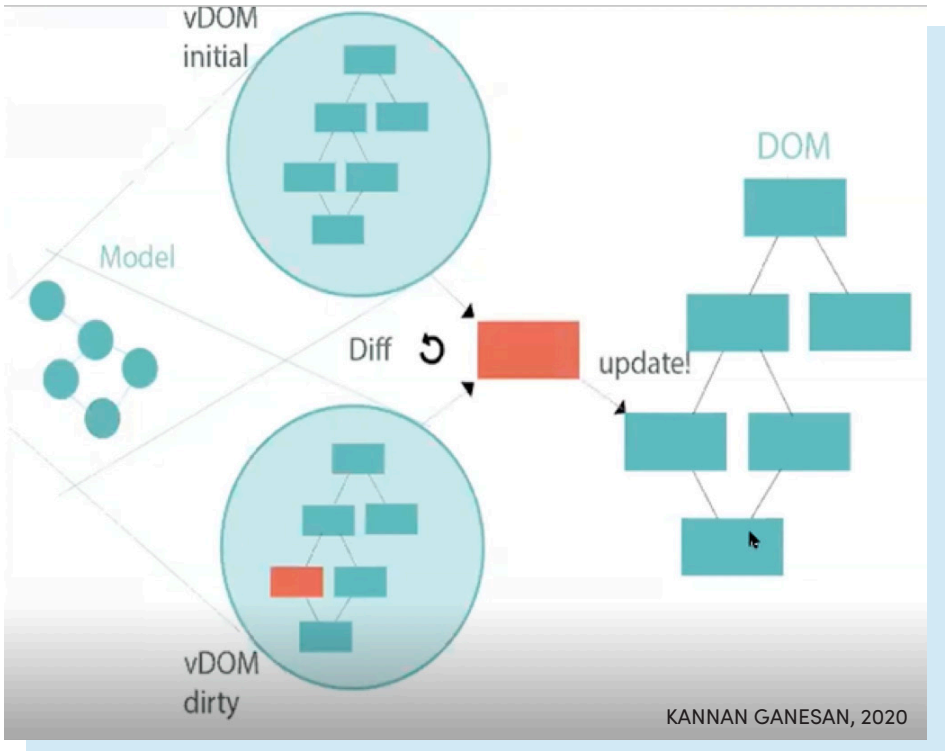
RECONCILIATION

Reconciliation is the process through which React updates the DOM. When a component's state changes, React has to calculate if it is necessary to update the DOM. It does this by creating a virtual DOM and comparing it with the current DOM. In this context, the virtual DOM will contain the new state of the component.

Virtual DOM is the tree of immutable React elements returned from the render method.

The process is as follows:

1. Change happens in the model
2. Model denotes the current state of the component
3. React creates a dirty vDOM and compares this to the initial vDOM to detect what exactly has changed
4. Identifies and updates only the relevant DOM. It's efficient!



BUILDING REACT FIBER

Starting from version 16, React rolled out a new implementation of internal instance trees and the algorithm that manages its code, named Fiber.

RECAP OF HOW BROWSER EXECUTES JS CODE

The JS code written will be executed directly on the main thread. There will be no intermediate person which will convert your code into something more efficient. JavaScript is an interpreted language, not a compiled language. It has no compilation step. Instead, an interpreter in the browser reads over the JavaScript code, interprets it, and runs it. Therefore, when you need to make any changes you have to go back and re-enter the entire DOM because the main thread will just read it directly as there is no difference calculator stage in JavaScript.

React came in to help solve this issue. React allows developers to create large web applications that can change data, without reloading the page and without re-entering the whole DOM. The main purpose is to be fast, scalable, and simple. It works by identifying differences between the old and new tree on the Virtual DOM, and then only updating the relevant browser DOM, thus increasing efficiency.

PROBLEMS WITH REACT STACK

React Stack relies on recursion to update the child components and re-render at each call if necessary. The result is that user experience suffers because everything is waiting for their update to finish and if the CPU is inefficient, there would be a number of side effects including DOM events, and lagging animation. Effectively, the main thread gets stuck waiting for update completion before it can move on and execute the next task.

React identified this issue and wanted to amend the React code in order to reduce waiting times and increase the immediacy of events, which is where Fiber comes in. In comparison, Fiber is faster, more efficient, and much more responsive.

REACT FIBER

React Fiber is the reimplementation of React's core algorithm. It is the culmination of over two years of research by the React team. It has effective project management skills, the ability to minimise and batch the DOM changes, and to keep track of time passed, enabling higher efficiency by the main thread.

It can actively delegate tasks in an efficient manner by splitting large work tasks down into manageable jobs, which is known as incremental rendering: the ability to split rendering work into chunks and spread it out over multiple frames.

MAIN IMPROVEMENTS OF REACT FIBER

PROCESSING ELEMENTS

When a React element is converted into a Fiber node for the first time, React uses the data from the element to create a Fiber in the createFiberFromTypeAndProps function. In the consequent updates React reuses the Fiber node and only updates the necessary properties using data from a corresponding React element. This makes the process sustainable and more efficient.

ALGORITHM TO WALK THE ELEMENTS TREE WITHOUT RECURSION

Instead of recursion, Fiber uses a linked list tree traversal algorithm, which means all fiber nodes are connected using a linked list with the properties: child, sibling, and return. Whilst Stack is a synchronous recursive modeul that relied on the browsers built in stack, Fiber is an asynchronous model with linked list and pointers.

EXECUTE AN UPDATE EFFECT ON ELEMENTS

Here Fiber works across two phases. Reconcile and Commit. In the Reconcile phase the Fiber node will be marked with a side effect and added onto the work in progress tree. Here a list of changes is built up but won't be performed until all changes have been listed. Then it will update everything, in priority order, in a short space of time (approximately 13 milliseconds).

Update Lifecycle:

RECONCILE - Fiber node, marked with side-effects (asynchronously)

- [UNSAFE_] componentWillMount (deprecated)
- [UNSAFE_] componentWillReceiveProps (deprecated)
- getDerivedStateFromProps
- shouldComponentUpdate
- [UNSAFE_] componentWillUpdate (deprecated)
- render

COMMIT - DOM updates visible to user (synchronous)

Performing the changes in the DOM (or native UI)

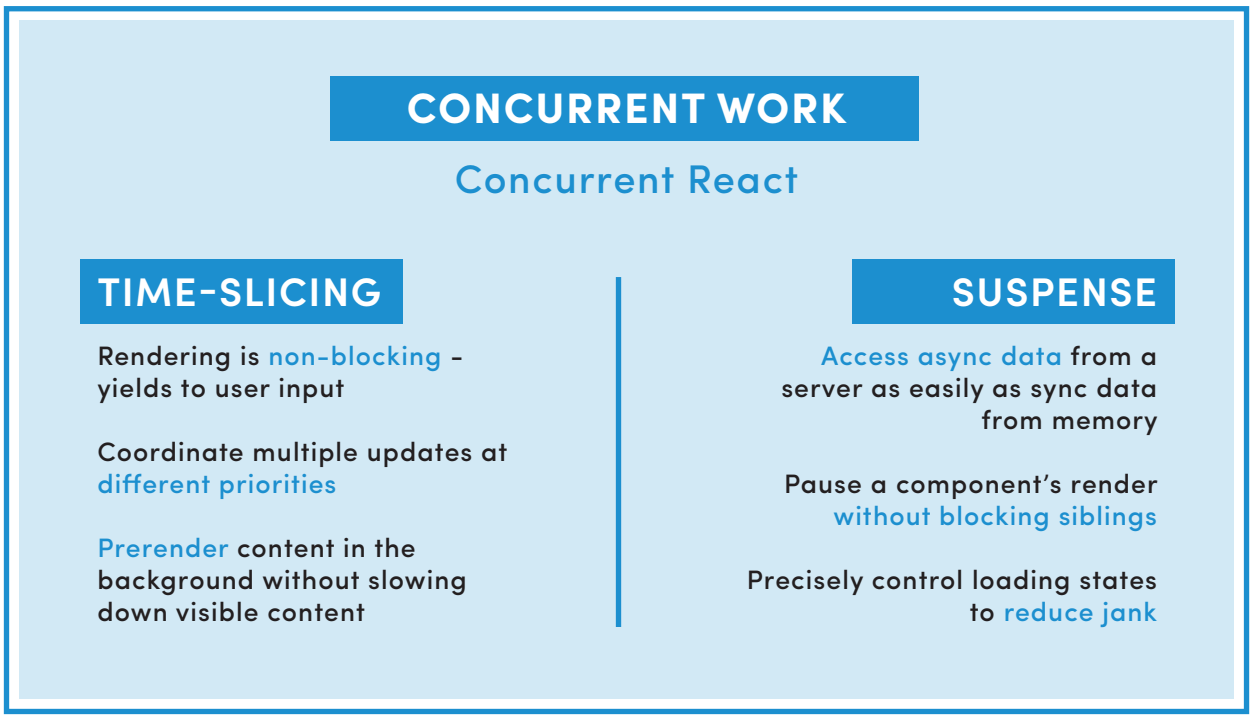
- getSnapshotBeforeUpdate
- componentDidMount
- componentDidUpdate
- componentWillUnmount

REACT FIBER MANIPULATES THE SINGLE THREAD PROCESS TO ENABLE ASYNC BEHAVIOUR

Javascript is a single thread process but Fiber helps fake a multi-thread process because it enables asynchronous behaviour and cooperative multitasking (also knows as non-preemptive multitasking).

WHAT MAKES FIBER INTERESTING?

1. Fiber makes applications more fluid and responsive
2. In the future it could parrellelise work, also known as Time Slicing (improving CPU)
3. It improves startup time whilse rendering components using React Suspense (Improve IO)



REACT VS. REACT FIBER

STACK RECONCILER

- Synchronous model (stack) recursive
- One phase

FIBER RECONCILER

- Asynchronous model - (linked list and pointer)
- Two phases (Reconcile, Commit)
- Reconcile phase can be interrupted
- Cooperative scheduling
- Ability to prioritise, rebase and reuse work in progress
- Ability to split interruptible work in chunks
- Time slicing and Suspense - improve CPU/IO
- Better support for error boundaries

CONTACT US

BUILD HIGH PERFORMING SOFTWARE TEAMS WITH MAXWELL BOND, THE RECRUITMENT PARTNER OF CHOICE FOR STAFFING SOLUTIONS IN THE UK AND GERMANY.

JORDAN MATTHEWS

FRONTEND DEVELOPMENT CONSULTANT,

JORDAN.MATTHEWS@MAXWELLBOND.CO.UK

maxwellbond

TRUSTED TECH TALKS

WEBINARS · EVENTS · NETWORKING