香港中文大學
The Chinese University of Hong Kong

MORE ON SPA AND ROUTING

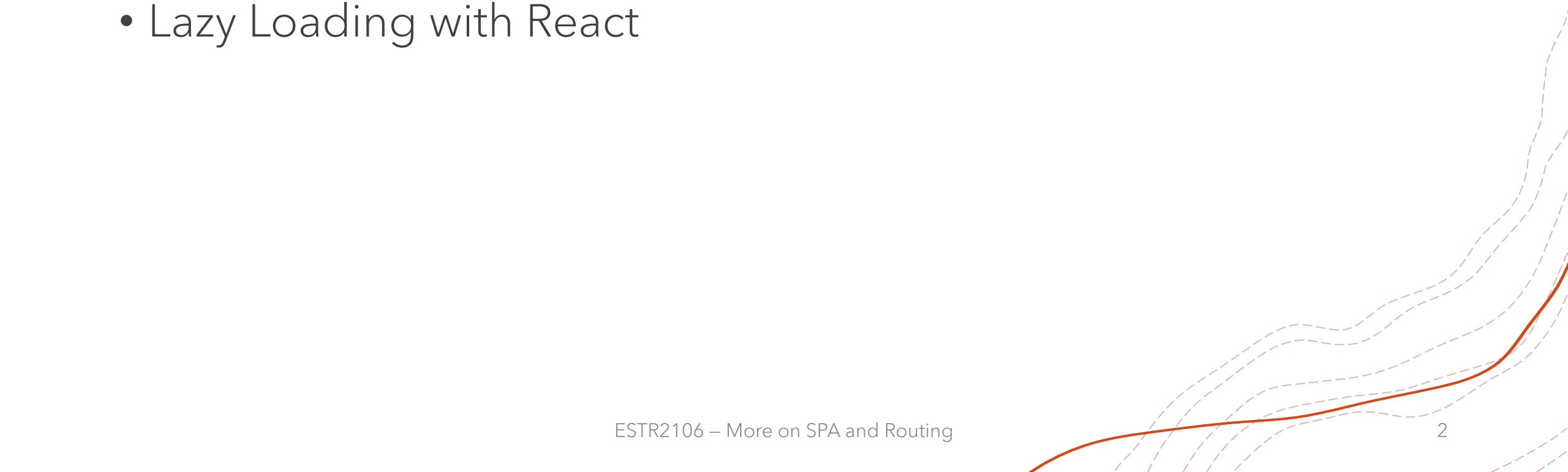
ESTR2106 2022-23 Term 1

Building Web Applications

Dr. Chuck-jee Chau
chuckjee@cse.cuhk.edu.hk

OUTLINE

- Lazy Loading
- Inspecting network activities
- Lazy Loading with React



LAZY LOADING

- In SPA, the app shell is loaded first, and then the data is loaded when needed
- This technique is generally known as ***Lazy Loading***, with improvement in multiple aspects:
  - Initial load time
  - Bandwidth
  - System resources
  - Cost

- Between 2011 and 2019:
  - Median resource weight increased
    - from ~100KB to ~400KB for desktop
    - from ~50KB to ~350KB for mobile
  - Image size has increased
    - from ~250KB to ~900KB on desktop
    - from ~100KB to ~850KB on mobile
  - *See: https://developer.mozilla.org/en-US/docs/Web/Performance/Lazy_loading*

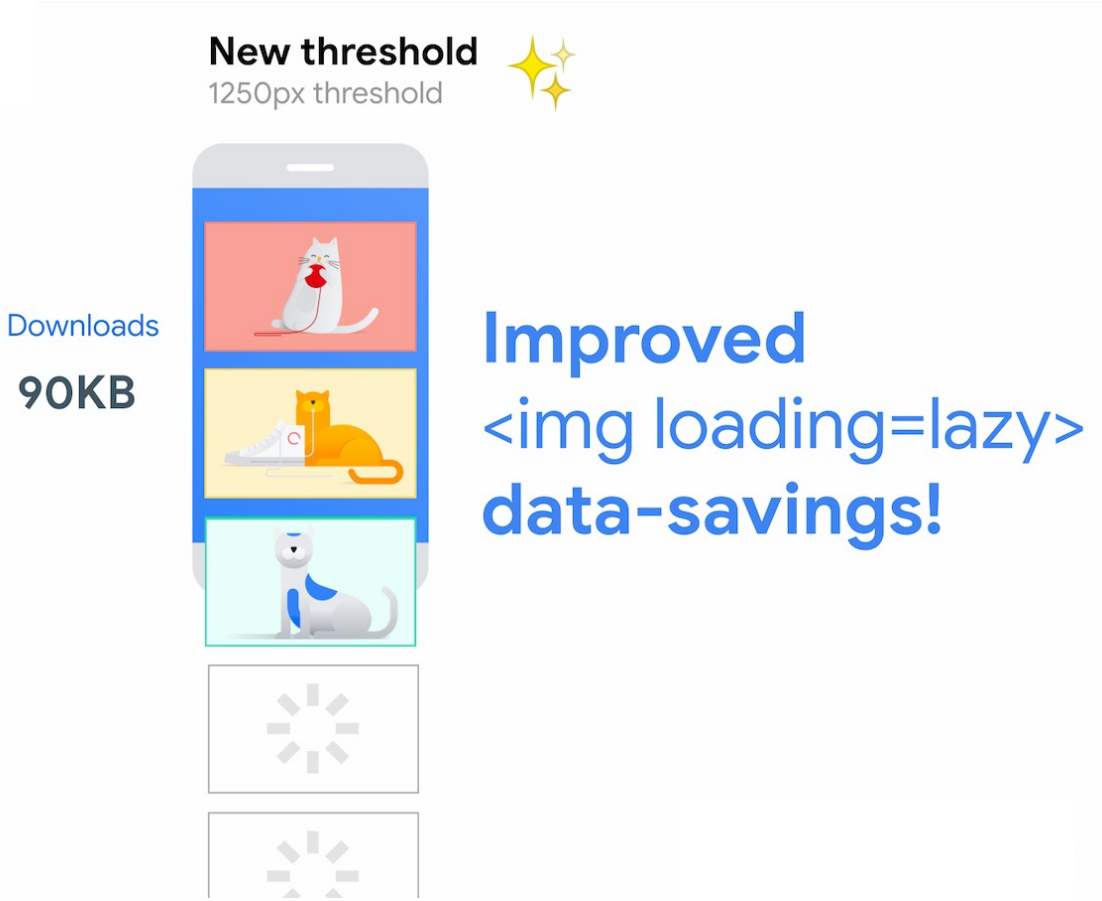
LAZY LOADING

- Contrasting to Eager Loading, Lazy Loading helps with application performance by delaying some elements by:
  - Lazy initialization
    - Initialized as null, and value is only obtained on first access
  - Virtual proxy
    - Initialized as an object with the same interface, and replaced by the real object on first access
  - Ghost
    - Only an identifier is loaded first, and data is loaded on first access

LAZY LOADING

See: https://web.dev/browser-level-image-lazy-loading/

- This can also be done for ordinary HTML elements
  - Images
  - iframes
- They are only loaded when entering o approaching the viewport
- Use **`loading="lazy"`** in `<img>` or `<iframe>` tags
  - `loading="eager"` ensures everything is loaded right away
  - Only supported by some latest version of browsers
  - *See: https://caniuse.com/loading-lazy-attr*

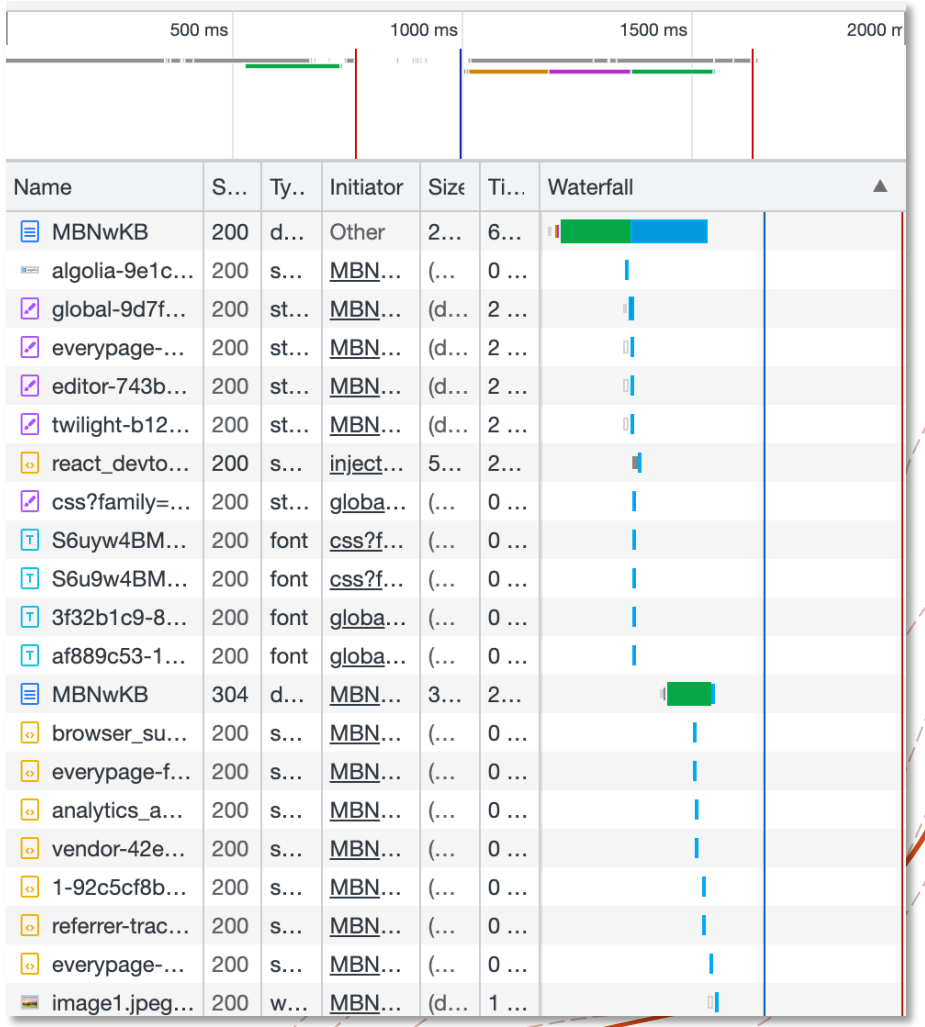


LAZY LOADING IMAGES

- To ensure browser compatibility, this can also be done
  - Preset image/iframe source into an alternate attribute such as data-src
  - JavaScript detects using scroll event, screen height, etc.
    - A new alternative: IntersectionObserver API
  - When the image/iframe comes into a threshold near the viewport (e.g., 1000px), replace **`src`** with **`data-src`** value
  - For example, *see: https://codepen.io/imagekit_io/pen/MBNwKB*
- A special class (e.g., ".lazy") can be used before loading
  - That can also be done on CSS background images
  - For example, *see: https://codepen.io/imagekit_io/pen/RBXVrW*

INSPECTING NETWORK ACTIVITIES

- In Google Chrome, there is a ***Network*** tab in DevTools
  - It shows the *waterfall chart* when your page elements are loaded
  - It is especially beneficial to see when lazy loading happens
- You can even simulate a slow connection to see if the placeholder works properly
  - *See: https://developer.chrome.com/docs/devtools/network/#throttle*



DEFERRING JAVASCRIPT

- Downloading a JavaScript file in **`<script>`** would block the building of DOM
- The downloading can be ***deferred***, i.e., done later

```
<script defer src="script.js"></script>
```

  - *See: https://javascript.info/script-async-defer*

- A JavaScript file dedicated as a module will be loaded deferred

```
<script type="module" src="main.js"></script>
```

  - The **`import`** and **`export`** syntax are relevant to JS modules

LAZY LOADING IN REACT

- In React, you can split components into multiple files for being loaded separately
- They can be loaded on-demand using **`React.lazy()`**

```
import OtherComponent from './OtherComponent'; //before

const OtherComponent = React.lazy(() => import('./OtherComponent')); //after
```

- Then, the component needs to be rendered with **`Suspense`**

```
<Suspense fallback={<div>Loading...</div>}>
  <OtherComponent />
</Suspense>
```

- *See: https://reactjs.org/docs/code-splitting.html#reactlazy*

LAZY LOADING IN REACT

- React-router can also allow components to be lazy loaded, only when the route is chosen
  - *See: https://github.com/remix-run/react-router/tree/dev/examples/lazy-loading*

READ FURTHER…

Complete Guide to Lazy Loading Images

https://css-tricks.com/the-complete-guide-to-lazy-loading-images/

Browser-level Image Lazy Loading

https://web.dev/browser-level-image-lazy-loading/