TextCritical.net - Bug #2681

Poor performance

04/23/2020 01:30 AM - Luke Murphey

Status:	Closed	**Start date:**	04/22/2020
Priority:	Normal	**Due date:**	
Assignee:	Luke Murphey	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	4.0		
Description			

History

#1 - 04/23/2020 01:35 AM - Luke Murphey

I suspect that this is caused by excessive re-rending of the Reader view.

I also may want to debounce some of the calls: https://stackoverflow.com/questions/23123138/perform-debounce-in-react-js

#2 - 04/29/2020 06:41 PM - Luke Murphey

Pain points:

- Searching the works list

#3 - 05/04/2020 08:02 PM - Luke Murphey

https://medium.com/@anuhosad/debouncing-events-with-react-b8c405c33273

#4 - 05/04/2020 08:10 PM - Luke Murphey

Getting this error: https://github.com/babel/babel/issues/9849

Seems fixed by importing regeneratorRuntime

#5 - 05/04/2020 08:52 PM - Luke Murphey

I denounced the fetch call. This isn't the issue since I am filtering on the client side.

```
const fetchWorks = () => fetch(ENDPOINT_WORKS_LISTS());
const fetchWorksDebounced = AwesomeDebouncePromise(fetchWorks, 1500);
```

```
  async loadInfo() {
    try {
      const response = await fetchWorksDebounced();
      const works = await response.json();
      this.setState({ works });
    } catch (e) {
      this.setState({
        error: e.toString(),
      });
    }
  }
```

I updated to filter onChange. This is better but a search for "josephus" is still slow.

#6 - 05/04/2020 09:18 PM - Luke Murphey

Tried building an index. This is still slow.

- https://nikitahl.com/how-to-find-an-item-in-a-javascript-array/
  - Says forLoop is best
- https://developer.mozilla.org/en-US/docs/Web/API/Performance
  - Performance measuring API
- https://www.andygup.net/fastest-way-to-find-an-item-in-a-javascript-array/
- https://andrewdupont.net/2006/05/18/javascript-associative-arrays-considered-harmful/
  - Says to use an associative array
- https://koukia.ca/top-6-ways-to-search-for-a-string-in-javascript-and-performance-benchmarks-ce3e9b81ad31
  - Regexes are faster but not by much
- https://hackernoon.com/3-javascript-performance-mistakes-you-should-stop-doing-ebf84b9de951
  - For Loop and While is fast; Map is slow

#7 - 05/04/2020 11:03 PM - Luke Murphey

Obs:

- Searching took 3 ms according to console.time()
- The UI seems to hang; like CPU is held up
- This seems to be a problem for the first render, but is better after the first slow search
- Seeing tons of LazyLoad componentWillUnmount()
- I see lots of commitNestedUnmounts() which each take 50ms

Qs:

- What is taking so long?
  - searching takes 3ms
  - rendering is not the problem either
- **[Yes]** Is CPU usage high?
  - Two cores seem to hit 50% for a while
- **[No]** Does removing LazyLoad help?
  - No
- Why is the first load so much faster?
- **[Yes]** Does reducing the number help much?
  - Very much so
- **[No]** Does removing the handlers help?
  - No
- **[No]** Is it better the the search matches few entries?
- Use https://react.semantic-ui.com/behaviors/visibility/
- Does rendering almost nothing help?
  - Very much so
  - Its the image that is causing the problem

Solns:

- **[No]** Keep rendered components in a list; don't re-render
  - Performance seems no better
- Do the search on the server-side
  - This would allow me to load related works easier too
- Use another component
  - https://cmichel.io/lazy-load-images-with-react/

#8 - 05/05/2020 02:33 AM - Luke Murphey

- File index.jsx added

Attaching modified Reader/index.jsx

#9 - 05/17/2020 07:41 PM - Luke Murphey

I could have these load on infinite scroll with https://react.semantic-ui.com/behaviors/visibility/

#10 - 05/19/2020 06:54 AM - Luke Murphey

```
[Violation] 'setTimeout' handler took 50ms
react-dom.development.js:174 [Violation] 'setTimeout' handler took 52ms
react-dom.development.js:174 [Violation] 'setTimeout' handler took 53ms
react-dom.development.js:174 [Violation] 'setTimeout' handler took 51ms
react-dom.development.js:174 [Violation] 'setTimeout' handler took 51ms
react-dom.development.js:174 [Violation] 'setTimeout' handler took 50ms
react-dom.development.js:174 [Violation] 'setTimeout' handler took 51ms
react-dom.development.js:174 [Violation] 'setTimeout' handler took 51ms
react-dom.development.js:174 [Violation] 'setTimeout' handler took 17306ms
[Violation] 'setTimeout' handler took 17332ms
[Violation] Forced reflow while executing JavaScript took 16757ms
```

#11 - 05/24/2020 03:40 PM - Luke Murphey

The

Qs:

- Is there another callback that seems to work?

Solns:

- Use another call back
  - onOnScreen: same
- Show the first few regardless

Refs:

- https://www.digitalocean.com/community/tutorials/react-components-viewport-react-visibility-sensor

#12 - 05/24/2020 03:40 PM - Luke Murphey

- Status changed from New to Closed

- % Done changed from 0 to 100

Files

index.jsx	4.27 KB	05/05/2020	Luke Murphey