



Международный журнал информационных технологий и энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004.051

## СРАВНИТЕЛЬНЫЙ АНАЛИЗ ВРЕМЕНИ РЕНДЕРИНГА КОМПОНЕНТОВ ПРИ ИСПОЛЬЗОВАНИИ РАЗЛИЧНЫХ ИНСТРУМЕНТОВ УПРАВЛЕНИЯ СОСТОЯНИЯМИ В REACT-ПРИЛОЖЕНИЯХ

<sup>1</sup>Пирюшов А.С., <sup>2</sup>Пирюшов М.С.

ФГАОУ ВО "НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ (ИТМО)" Санкт-Петербург, Россия (197101, город Санкт-Петербург, Кронверкский пр-кт, д. 49 литер а), e-mail: <sup>1</sup>uchenik.ikt@yandex.ru, <sup>2</sup>maximalka111@yandex.ru

**Цели.** В настоящее время веб-технологии достигли такого уровня своего развития, при котором пользователю важно, чтобы система решала определённую задачу и была наиболее эффективной среди аналогов. Одним из вариантов повышения эффективности является оптимизация времени рендеринга компонентов, которые составляют веб-приложение. Большинство таких приложений решают сразу несколько задач, следовательно их структуру можно представить, как композицию нескольких подсистем. Нередко состояние подсистемы оказывает влияние на состояние всей системы в целом. Для доступа в подсистемах к этому состоянию разработчики применяют различные инструменты управления состоянием. Цель работы – сравнить время рендеринга компонентов в React-приложениях, различающихся функциональными возможностями, используя различные актуальные инструменты управления состоянием.

**Методы.** Предлагается исследовать работу веб-приложений, построенных с помощью фреймворка React, при использовании таких инструментов управления состоянием приложения, как Redux Toolkit, Effector и React Context. Экспериментальные результаты получены путём использования инструментов разработчика, предоставляемых Google, в частности вкладки Performance.

**Результаты.** Представлено описание спроектированных веб-приложений, используемых в процессе исследования, приведены экспериментальные результаты и их сравнение.

**Выводы.** Проведенные исследования позволяют говорить о том, что Effector показывает лучший результат в приложениях, в которых необходимо выполнять асинхронные запросы на сервер (среднее значение времени рендеринга компонентов равно 6585,05 мс), Redux Toolkit является наиболее оптимальным решением в случаях, когда веб-приложение состоит из большого и малого количества компонентов без выполнения асинхронных запросов (среднее значение времени рендеринга компонентов равно 114,97 мс и 1,03 мс соответственно).

**Ключевые слова:** Инструмент управления состоянием, оптимизация, рендеринг, менеджер состояния, веб-приложение, инструменты разработчика, React, Effector, Context, Redux.

## COMPARATIVE ANALYSIS OF COMPONENT RENDERING TIME WHEN USING VARIOUS STATE MANAGEMENT TOOLS IN REACT APPLICATIONS

Piryushov A.S., Piryushov M.S.

NATIONAL RESEARCH UNIVERSITY OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS (ITMO) St. Petersburg, Russia (197101, St. Petersburg, Kronverkskiy pr-kt, 49), e-mail: <sup>1</sup>uchenik.ikt@yandex.ru, <sup>2</sup>maximalka111@yandex.ru

**Objectives.** Currently, web technologies have reached a level of development at which it is important for the user that the system solves a specific problem and is the most effective among analogues. One option to improve efficiency is to optimize the rendering time of the components that make up a web application. Most of these applications solve several problems at once, therefore their structure can be represented as a composition of several subsystems. Often the state of a subsystem affects the state of the entire system as a whole. To access this state in subsystems, developers use various state management tools. The goal of the work is to compare the rendering time of components in React applications that differ in functionality, using various current state management tools.

**Methods.** It is proposed to explore the performance of web applications built using the React framework using application state management tools such as Redux Toolkit, Effector and React Context. Experimental results were obtained by using the developer tools provided by Google, in particular the Performance tab.

**Results.** A description of the designed web applications used in the research process is presented, experimental results and their comparison are given.

**Conclusions.** The conducted research suggests that Effector shows the best results in applications in which it is necessary to perform asynchronous requests to the server (the mathematical expectation of the component rendering time is 6585.05 ms), Redux Toolkit is the most optimal solution in cases where the web application consists from a large and small number of components without executing asynchronous requests (the mathematical expectation of the component rendering time is 114.97 ms and 1.03 ms, respectively).

Keywords: State management tool, optimization, rendering, state manager, web application, developer tools, React, Effector, Context, Redux.

## Введение

Под влиянием широкого развития Web в мире, увеличения значимости его концепций в экономической сфере настал момент, когда число всевозможных веб-технологий разрослось, и чтобы заставить пользователя выбрать определённое веб-приложение, оно должно обладать существенными преимуществами перед другими ему подобными приложениями с теми же функциональными возможностями. Для каждого человека так или иначе одной из главных ценностей является время. Перед разработчиками стоит проблема создания веб-приложения, которое помимо решения своих основных задач ещё и является наиболее оптимизированным в своём сегменте. Одним из способов повышения эффективности такого приложения является правильный выбор инструмента для управления его состоянием.

Наиболее распространёнными фреймворками для создания клиентской части веб-приложения являются React, Angular, Vue. React-приложения выделяются среди других решений своей производительностью, фреймворк использует виртуальный DOM для внесения изменений. Возникла потребность в эффективном управлении их состоянием [1].

На данный момент существует множество различных инструментов для управления состоянием React-приложений. Они имеют различные подходы определения доступа к состоянию, обладают собственной экосистемой, различаются способом подписки компонентов на изменения, всё это достаточно подробно рассматривается в некоторых работах [2-5] и учитывается при выборе наилучшего решения, при этом слабо исследуется неоднозначное влияние менеджеров состояний на функционирование разной сложности React-приложений. К тому же не было найдено ни одной работы, в которой исследовалась бы эффективность рендеринга компонентов React-приложений, использующих такой менеджер состояния, как Effector.

## Постановка задачи

Чаще всего на решение разработчика по выбору менеджера состояния оказывает влияние его личный опыт или, наоборот, отсутствие опыта с аналогами. От выбора инструмента управления состоянием React-приложения зависит его быстроедействие. В зависимости от

структуры приложения, его функциональных возможностей оптимальное решение отличается.

Исходя из вышесказанного, возникает задача сравнения времени рендеринга компонентов различных по уровню сложности React-приложений с использованием актуальных инструментов управления их состоянием и выбор наиболее подходящего под тот или иной случай.

### Малокомпонентное веб-приложение

В качестве первого эксперимента предлагается протестировать время рендеринга веб-приложения, которое содержит небольшое количество компонентов. За основу было взято приложение, представляющее собой примитивный счётчик: текстовое поле, отображающее текущее значение счётчика и две кнопки для добавления и вычитания единицы. Пользовательский интерфейс представлен на Рисунке 1.

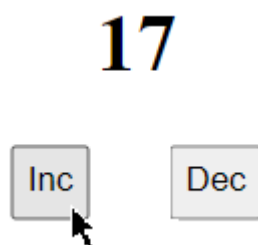


Рисунок 1 – Пользовательский интерфейс простого веб-приложения для эксперимента

Время рендеринга веб-приложения фиксировалось при изменении счётчика на единицу как в большую, так и в меньшую сторону.

Замер времени рендеринга осуществлялся инструментами разработчика, предоставляемыми Google на вкладке Performance.

Для каждого рассматриваемого инструмента управления состоянием было проведено десять экспериментов и вычислено среднее значение. Результаты замеров приведены в Таблице 1.

Таблица 1 – Значения времени рендеринга однокомпонентного веб-приложения

Номер теста	Время рендеринга (мс)		
	Redux Toolkit	Effector	Context
1	1	1,1	1,2
2	1,1	1,4	1,2
3	1,2	1	1,3
4	1	1	1,4
5	0,9	1	1,9
6	1	1,1	1,7
7	1	0,8	1,3
8	1	1,1	1
9	1	0,9	1,2
10	1,1	1,4	1,3

В Таблице 2 приведены средние значения для каждого решения.

Таблица 2 – Средние значения времени рендеринга однокомпонентного веб-приложения

Среднее время рендеринга (мс)		
Redux Toolkit	Effector	Context
1,03	1,08	1,35

### Многокомпонентное веб-приложение

В качестве второго эксперимента предлагается протестировать время рендеринга веб-приложения, которое содержит большое количество компонентов, приближённое к реальному продукту коммерческой разработки.

За основу было взято приложение, представляющее собой инструмент для работы с данными клиентов: имя, возраст, информация о подписке и текущем рабочем положении. Предоставляется форма для заполнения данных, таблица со всеми записями, кнопки для добавления данных в таблицу и удаления записи из неё, а также кнопка для изменения темы приложения (светлая или тёмная). Пользовательский интерфейс представлен на Рисунке 2.

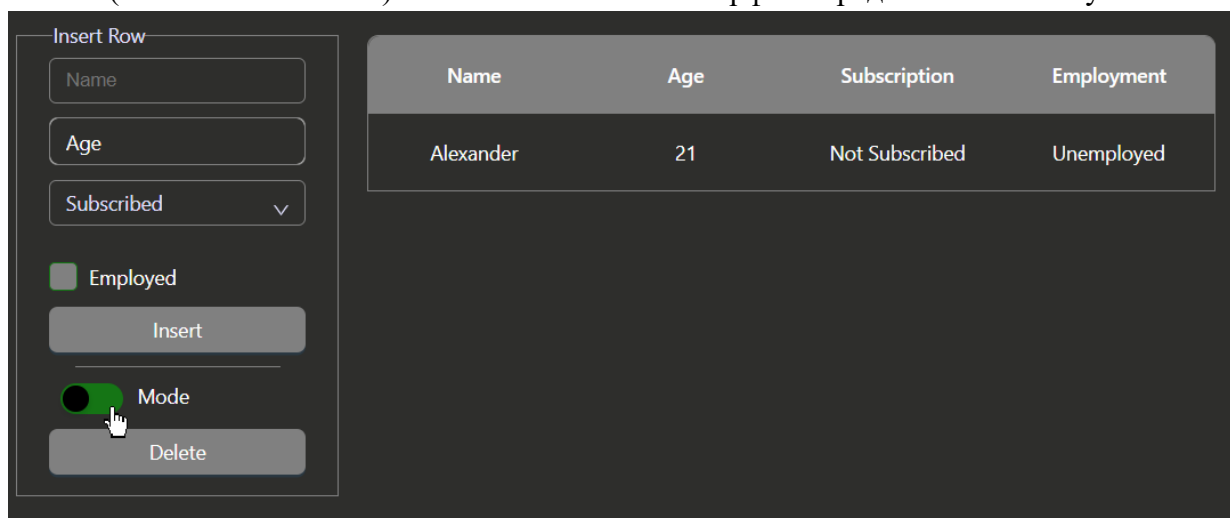


Рисунок 2 – Пользовательский интерфейс многокомпонентного веб-приложения для эксперимента

На Рисунке 3 представлен интерфейс приложения при нажатии на запись в таблице. Всплывает форма для изменения данных текущей записи.

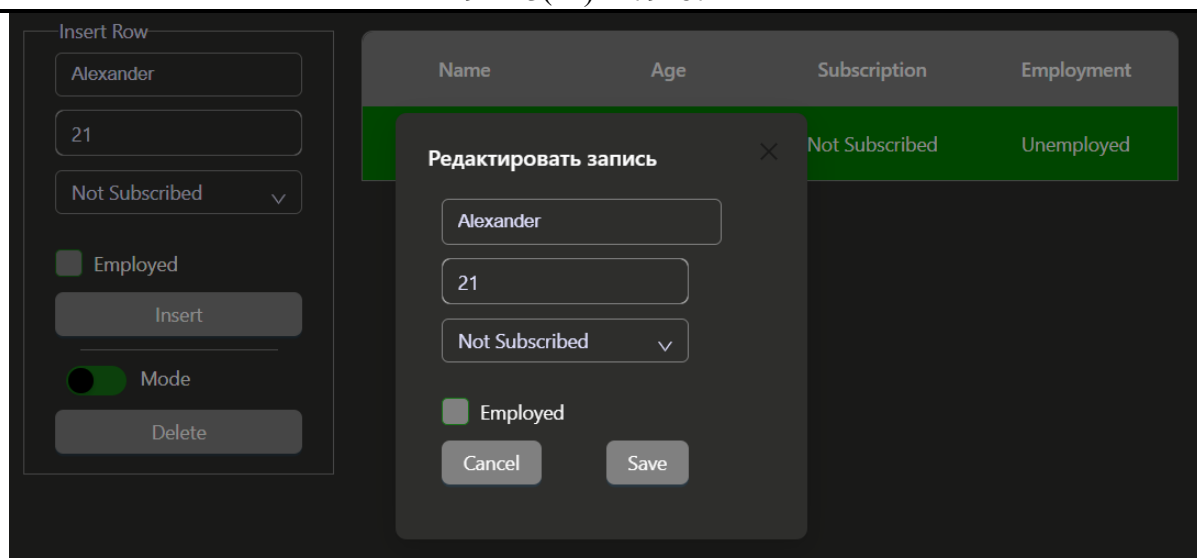


Рисунок 3 – Пользовательский интерфейс с формой для изменения записи в таблице

За рендеринг веб-приложения, время которого вычислялось в эксперименте, были приняты все изменения с начала открытия формы и до появления обновленной записи в таблице.

Замер времени рендеринга осуществлялся инструментами разработчика, предоставляемыми Google на вкладке Performance.

Для каждого рассматриваемого инструмента управления состоянием было проведено десять экспериментов и вычислено среднее значение. Результаты замеров приведены в Таблице 3.

Таблица 3 – Значения времени рендеринга многокомпонентного веб-приложения

Номер теста	Время рендеринга (мс)		
	Redux Toolkit	Effector	Context
1	134,1	123,7	159,1
2	114,3	122,5	161,2
3	109,1	105,9	142,1
4	122,1	126,6	140,1
5	110,8	118,9	163,5
6	106	113,6	154,2
7	110,2	114,6	164,2
8	124,5	116,1	151,4
9	111,1	117,6	142,9
10	107,5	117,3	166,2

В Таблице 4 приведены средние значения для каждого решения.

Таблица 4 – Средние значения времени рендеринга многокомпонентного веб-приложения

Среднее время рендеринга (мс)		
Redux Toolkit	Effector	Context
114,97	117,68	154,49

### Веб-приложение с асинхронным запросом

В качестве третьего эксперимента предлагается протестировать время рендеринга веб-приложения, которое запрашивает данные с определённого веб-сервиса. За основу было взято приложение, в котором есть текстовое поле для ввода количества запрашиваемых картинок и кнопка для осуществления самого запроса. Пользовательский интерфейс представлен на Рисунке 4.

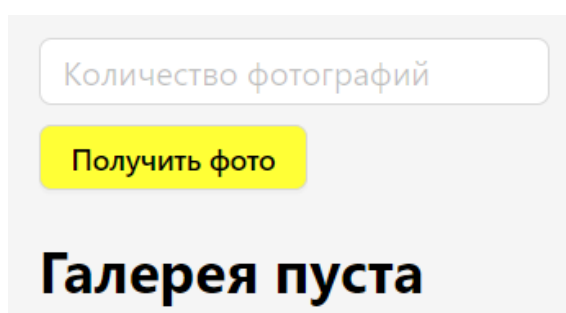


Рисунок 4 – Пользовательский интерфейс веб-приложения с асинхронным запросом для эксперимента

Время рендеринга отсчитывалось с момента нажатия на кнопку для получения фото и до полной отрисовки полученных данных в виде галереи (Рисунок 5).

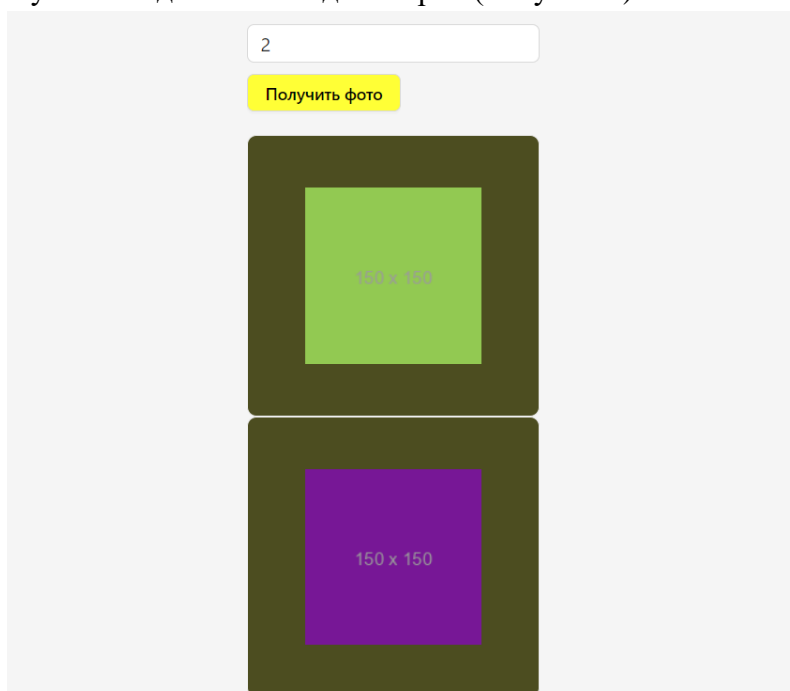


Рисунок 5 – Компонент галерея

Замер времени рендеринга осуществлялся инструментами разработчика, предоставляемыми Google на вкладке Performance.

Для каждого рассматриваемого инструмента управления состоянием было проведено десять экспериментов с количеством получаемых фотографий, равным 5000, и вычислено среднее значение. Результаты замеров приведены в Таблице 5.

Таблица 5 – Значения времени рендеринга веб-приложения с асинхронным запросом

Номер теста	Время рендеринга (мс)		
	Redux Toolkit	Effector	Context
1	6493,3	5646,4	5608,1
2	8629,3	5633,6	7990,5
3	6556,4	6571,1	8557,6
4	6290,1	5337,3	5562,9
5	9886,3	5522,5	8350,3
6	8548,5	7698,2	9572,3
7	8374,7	7884,8	8082,4
8	8983,2	5487,4	8548,7
9	8870,4	7675	5483,3
10	8740,7	8394,2	8189,8

В Таблице 6 приведены средние значения для каждого решения.

Таблица 6 – Средние значения времени рендеринга веб-приложения с асинхронным запросом

Среднее время рендеринга (мс)		
Redux Toolkit	Effector	Context
8137,29	6585,05	7594,59

### Выводы

По результатам первого эксперимента, представленным в таблицах 1,2, можно сделать вывод, что для React-приложений с малым количеством компонент, наиболее оптимальным решением является менеджер состояния Redux Toolkit, среднее время рендеринга компонентов с использованием которого составило 1,03 мс. Сравнимый результат был получен с использованием Effector (среднее время рендеринга – 1,08 мс). Больше всего времени понадобилось при использовании React Context (среднее время рендеринга – 1,35 мс).

Второй эксперимент, проводимый с React-приложением, структура которого по количеству компонент и вложенности сложнее, результаты которого представлены в таблицах 3,4, расположил исследуемые инструменты в том же порядке – Redux Toolkit и Effector также являются наиболее оптимальными инструментами для управления состоянием со средним временем рендеринга 114,97 мс и 117,68 мс соответственно, при использовании Context среднее время рендеринга составило 154,49 мс.

В эксперименте с React-приложением, где осуществлялась работа с данными, полученными в ответе от веб-сервиса на асинхронный запрос (таблицы 5,6), результаты были следующими: рендеринг с использованием Effector занял наименьшее количество времени, среднее значение которого равно 6585,05 мс, Context справляется с этой задачей за 7594,59 мс. Хуже всего показал себя в этом случае Redux Toolkit (среднее время рендеринга – 8137,29 мс).

Подводя итог, можно сказать, что выбор инструмента для управления состоянием приложения зависит от структуры React-приложения и выполняемых им задач. Наиболее стабильным и оптимальным решением во всех проводимых экспериментах оказался Effector.

### Список литературы

1. Бодров, С. Д. Стейт менеджеры в frontend разработке на React. A Posteriori. 2023;6:17-21. URL: <https://elibrary.ru/item.asp?id=53958412>.
2. Кашапов, К. Г., Ким П.Е., Голикова Е.А. Оценка и сравнение менеджеров состояний применяемых при решении проблемы управления состоянием в современном SPA веб приложении. Научно-технические инновации и веб-технологии. 2023;1:70-77. URL: <https://elibrary.ru/item.asp?id=53922922>.
3. Натробина, А. Э., Фёдоров В.О. Redux и новый react context api как инструменты управления состоянием веб-приложения. Компьютерные технологии и моделирование в науке, технике, экономике, образовании и управлении: тенденции и развитие. Материалы международной научно-технической конференции. 2019;299-301. URL: <https://elibrary.ru/item.asp?id=43113854>.
4. Кольшикина, Н.С., Малкова Л.Е. Сравнение библиотек Redux и MobX. Modern Science. 2020;6(2):267-270. URL: <https://elibrary.ru/item.asp?id=43005224>.
5. Ким, А. Выбор менеджера управления состояний в React. Научно-технические инновации и веб-технологии. 2022;2:30-34. URL: <https://elibrary.ru/item.asp?id=50203811>.
6. Vasilijevic, V., Kojic N., Vugdeliija N. A New Approach In Quantifying User Experience In Weboriented Applications. In: Proceedings of the 4th International Scientific Conference IТЕМА, Association of Economists and Managers of the Balkans. 2020. P. 25-30. <https://doi.org/10.31410/IТЕМА.2020>.

### References

1. Bodrov, S. D. State managers in front-end development using React. A Posteriori = A Posteriori. 2023;6:17-21 (in Russ.). Available from URL: <https://elibrary.ru/item.asp?id=53958412>.
2. Kashapov, K. G., Kim P.E., Golikova E.A. Evaluation and comparison of state managers used to solve the problem of state management in a modern SPA web application. Nauchno-tekhnicheskiye innovatsii i veb-tehnologii = Scientific and technical innovations and web technologies. 2023;1:70-77 (in Russ.). Available from URL: <https://elibrary.ru/item.asp?id=53922922>.
3. Natrobina, A. E., Fydorov V.O. Redux and the new react context api as web application state management tools. Komp'yuternyye tekhnologii i modelirovaniye v nauke, tekhnike, ekonomike, obrazovanii i upravlenii: tendentsii i razvitiye. Materialy mezhdunarodnoy nauchno-tekhnicheskoy konferentsii = Computer technologies and modeling in science, technology, economics, education and management: trends and development. Materials of the international scientific and technical conference. 2019;299-301 (in Russ.). Available from URL: <https://elibrary.ru/item.asp?id=43113854>.
4. Kolyshkina, N.S., Malkova L.E. Comparison of Redux and MobX libraries. Modern Science = Modern Science. 2020;6(2):267-270 (in Russ.). Available from URL: <https://elibrary.ru/item.asp?id=43005224>.



5. Kim, A. Choosing a State Manager in React. Nauchno-tehnicheskiye innovatsii i veb-tehnologii = Scientific and technical innovations and web technologies. 2022;2:30-34 (in Russ.). Available from URL: <https://elibrary.ru/item.asp?id=50203811>.
  6. Vasiljevic, V., Kojic N., Vugdelija N. A New Approach In Quantifying User Experience In Weboriented Applications. In: Proceedings of the 4th International Scientific Conference ITEMA, Association of Economists and Managers of the Balkans. 2020. P. 25-30. <https://doi.org/10.31410/ITEMA.2020>.
-