

Рисунок 2 – График оптимизации

Таким образом максимальное количество заменяющих символов равно пяти, а минимальное одному. Чем чаще символ встречается относительно других, тем ближе число заменяющих символов будет к максимальному.

Для увеличения максимального количества заменяющих символов достаточно увеличить числитель. Для увеличения чувствительности функции можно заменить число восемь на большее.

Список цитируемых источников

1. Сандруцкий, Д. И. Применение криптографических систем при создании мессенджера / Д. И. Сандруцкий, С. Д. Колдушко, А. И. Калько // Студенческий. – 2017. – № 16(16). – С. 14–16. – EDN KFGYYR.

УДК 004.42

МАНИФЕСТ РЕАКТИВНОГО ПРОГРАММИРОВАНИЯ

В.А. Литвинова

Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Беларусь

THE REACTIVE MANIFESTO

V.A. Litvinava

Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus

Аннотация. В данной работе рассмотрено понятие реактивного программирования. Рассмотрен манифест реактивных систем и представлена его схема. Рассмотрена полезность реактивных системы, методы и средства ее достижения. Рассмотрена формулировка закона Д. Амдала 1967 г.

Ключевые слова: реактивность, реактивное программирование, манифест реактивных систем, отзывчивость, эластичность, отказоустойчивость, обмен сообщениям, событийно-ориентированный подход закон Джина Амдала.

Annotation. This paper discusses the concept of reactive programming. The Reactive Manifesto is considered and its diagram is presented. The value, form and means of reactive systems are considered. The formulation of Amdahl's law of 1967 is considered.

Keywords: reactivity, reactive programming, reactive systems manifesto, responsive, elastic, resilient, message driven, event-driven, Amdahl's law.

За последние годы требования к приложениям сильно изменились. Всего несколько лет назад у крупного приложения были десятки серверов, секунды отклика, часы автономного обслуживания и гигабайты данных. Сегодня приложения развертываются на всех устройствах – от мобильных устройств до облачных кластеров с тысячами многоядерных процессоров. Пользователи ожидают миллисекундного времени отклика и 100% времени безотказной работы. Данные измеряются в петабайтах. Сегодняшние требования просто не удовлетворяются вчерашними архитектурами программного обеспечения.

Необходим последовательный подход к системной архитектуре, как результат которого разрабатывались бы системы, которые являются отзывчивыми, устойчивыми, масштабируемы и управляемые сообщениями (рис. 1) [1]. Реактивное программирование – это подход к программированию, который использует асинхронные потоки данных и событий для создания более отзывчивых приложений. [2]. Системы, построенные как реактивные, являются более гибкими, слабосвязанными и масштабируемыми. Это делает их более легкими в развитии и поддающимися изменениям.

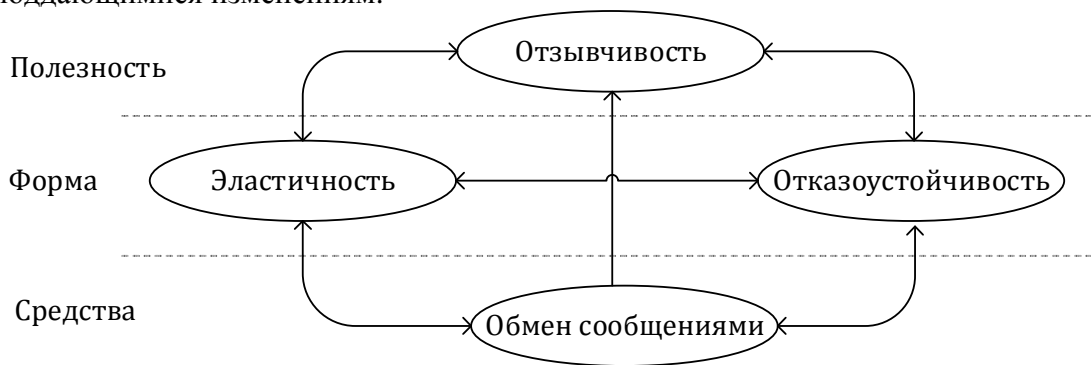


Рисунок 1 – Манифест реактивного программирования

Полезность или ценность реактивных систем «Отзывчивости» – система реагирует своевременно, если это вообще возможно. Отзывчивость является краеугольным камнем удобства использования и полезности, но более того, отзывчивость означает, что проблемы могут быть быстро обнаружены и эффективно устранены.

Метод достижения полезности или форма «Отказоустойчивости» – означает, что система остается отзывчивой в случае сбоя. Устойчивость достигается за счет репликации, сдерживания, изоляции и делегирования. Сбои локализуются внутри каждого компонента, изолируя компоненты друг от друга и тем самым гарантируя, что части системы могут выйти из строя и восстановиться без ущерба для системы в целом. Восстановление каждого компонента делегируется другому (внешнему) компоненту, а высокая доступность обеспечивается репликацией там, где это необходимо. Клиент компонента не обременен обработкой его сбоев.

Метод достижения полезности или форма «Эластичности» – способность сохранять отзывчивость при различной рабочей нагрузке, то есть пропускная способность системы должна автоматически увеличиваться с ростом числа пользователей и уменьшаться – со снижением спроса. Эта особенность улучшает отзывчивость системы, потому что в любой момент пропускная способность системы может вырасти и обеспечить приемлемое среднее время задержки. При отсутствии должной эластичности из-за роста нагрузки увеличится время задержки, которое напрямую влияет на отзывчивость системы. Например, увеличить пропускную способность системы можно, расширяя вычислительные мощности или запуская дополнительные экземпляры. В результате возрастет отзывчивость системы. С другой стороны, если поток пользователей уменьшился, система в ответ должна снизить потребление ресурсов, сократив тем самым накладные расходы. Добиться желаемой эластичности можно путем масштабирования – горизонтального или вертикального. Однако масштабирование распределенной систе-

мы – сложная задача. Обычно ограничиваются узкими местами или точками синхронизации в системе [3]. С теоретической и практической точек зрения эти проблемы объясняются законом Амдала и универсальной моделью масштабирования Гюнтера Нейла.

Наблюдение Джина Амдала – увеличение производительности, достигаемое некоторым усовершенствованием, ограничено потребляемой усовершенствованным компонентом долей общего времени выполнения [4]. Иначе говоря, повышение производительности пропорционально доле, занимаемой улучшаемым компонентом в процессе выполнения программы. Ускорение выполнения программы за счёт распараллеливания её инструкций на множестве вычислителей ограничено временем, необходимым для выполнения её последовательных инструкций. Закон Амдала объясняет, почему составляющие времени отклика следует рассматривать в порядке убывания [5]. В реактивных системах ускорение выполнения программы достигается не только за счет распараллеливания, а благодаря отсутствию блокирующих ввода-вывода. Использование неблокирующего ввода-вывода, позволит минимизировать время, которое потоки тратят на ожидание завершения операций ввода-вывода. Более эффективное использование потоков снижает вероятность "голодания" потоков и увеличивает производительность сервиса.

Средство реактивных систем «Обмен сообщениями» – архитектура системы основана на обмене сообщениями. Такой подход позволяет сервисам работать независимо друг от друга, без необходимости знать о состоянии или доступности других сервисов. Однако реактивная система может быть основана и на событийно-ориентированном подходе.

В данной работе было дано понятие реактивного программирования. Показано как эластичность, отказоустойчивость и взаимодействие с использованием сообщений помогают достичь отзывчивости. Рассмотрена формулировка закона Джина Амдала от 1967 г.

Список цитируемых источников

1. Реактивный манифест [Электронный ресурс]. – Режим доступа: www.reactivemaneifesto.org. – Дата доступа: 23.04.2024.
2. Алпатов, А. Н. Применение реактивного программирования и модели комплексной обработки событий / А. Н. Алпатов, А. С. Сороков // Столыпинский вестник: Освещение вопросов социально-экономических реформ в России. Том 5, №3/2023: сб. ст. / сост.: И. Ю. Мамонтова. – М., 2023. – С. 2692–2699.
3. Докука, О. Практика реактивного программирования в Spring 5 / О. Докука, И. Лозинский. – М. : ДМК Пресс, 2019. – 508 с.
4. Карлин, С. Универсальный закон масштабируемости (USL) [Электронный ресурс] / Сэм Карлин. – Режим доступа: www.perfdynamics.com/Manifesto/USLscalability.html#n93. – Дата доступа: 23.04.2024.
5. Миллсап, К. Oracle. Оптимизация производительности / К. Миллсап, Д. Хольт. – Пер. с англ. – СПб. : Символ-Плюс, 2006. – 464 с.

УДК 004.925.84+159.937.52

ВЛИЯНИЕ СВОЙСТВ ИГРОВОГО ПРОСТРАНСТВА НА ВОВЛЕЧЕННОСТЬ ИГРОКОВ

Н.В. Попеня

Белорусский государственный технологический университет, г. Минск, Беларусь

THE INFLUENCE OF GAME SPACE PROPERTIES ON PLAYER ENGAGEMENT

N.V. Popenya

Belarusian State Technological University, Minsk, Belarus

Аннотация. В статье рассматривается значимость игрового пространства в видеоиграх и его влияние на пользовательский опыт. Описываются ключевые характеристики такого пространства, включая уровни масштабности, сильные центры, границы, повторения, контраст и форму.

Ключевые слова: игровое пространство; вовлеченность игрока; свойства пространства.