

УДК 004.89

DOI 10.52575/2687-0932-2023-50-2-476-486

## Имитационная модель распределенного процесса сбора данных

Оболенский Д.М., Шевченко В.И., Ченгарь О.В.

Севастопольский государственный университет,

Россия, 299053, г. Севастополь, ул. Университетская, д. 33

E-mail: denismaster@outlook.com, VIShevchenko@sevsu.ru, OVChengar@sevsu.ru

**Аннотация.** В данной работе авторы рассматривают процесс сбора и обработки данных из различных веб-источников. Исследована простая модель сбора данных, основанная на циклическом переборе, определены её основные недостатки. Предложена модель распределенного сбора данных как многоканальной системы массового обслуживания с неограниченной очередью. Модель использует множество узлов для доступа к онлайн-ресурсу, а для хранения информации о задачах и их балансировке между узлами используется очередь сообщений. Распределенная модель также обладает свойством отказоустойчивости и горизонтального масштабирования. Авторы сравнивают обычную и распределенные модели при помощи средства имитационного моделирования AnyLogic. Также в процессе моделирования используются различные распределения времени отклика внешнего ресурса. Результаты имитационного моделирования показывают эффективность распределенного процесса сбора данных как по критерию времени, так и по критерию удельной стоимости.

**Ключевые слова:** распределенный сбор данных, имитационная модель, сбор данных, распределенные системы, системы массового обслуживания

**Для цитирования:** Оболенский Д.М., Шевченко В.И., Ченгарь О.В. 2023. Имитационная модель распределенного процесса сбора данных. Экономика. Информатика, 50(2): 476–486. DOI: 10.52575/2687-0932-2023-50-2-476-486

---

## A Simulation Model of the Distributed Data Collection Process

Denis M. Obolensky, Victoria I. Shevchenko, Olga V. Chengar

Sevastopol State University,

33 University St, Sevastopol, 299053, Russia

E-mail: denismaster@outlook.com, VIShevchenko@sevsu.ru, OVChengar@sevsu.ru

**Abstract.** In this paper, the authors consider the process of collecting and processing data from various web sources. A simple data collection model based on cyclic iteration is investigated, its main disadvantages are identified. The model of distributed data collection as a multi-channel queuing system with unlimited waiting is described. This model uses multiple nodes to access an online resource, and a message queue is used to store information about tasks and to balance them between nodes. The distributed model is also fault tolerance and horizontal scalable. The authors compare simple and distributed models using the AnyLogic simulation tool. Additionally, various distributions of the server response time of an external web-resource are used in the modeling process. The simulation results show the effectiveness of the distributed data collection process both by the criterion of time and by the criterion of unit cost.

**Keywords:** distributed data collection, simulation model, data collection, distributed systems, queuing model

**For citation:** Obolensky D.M., Shevchenko V.I., Chengar O.V. 2023. A Simulation Model of the Distributed Data Collection Process. Economics. Information technologies, 50(2): 476–486 (in Russian). DOI: 10.52575/2687-0932-2023-50-2-476-486

---

## Введение

Процесс сбора данных с различных веб-сайтов и API [Государев, 2020] (интерфейсы прикладного программирования) может быть автоматизирован с помощью различных инструментов, которые позволяют разработчикам получать доступ к некоторым онлайн-ресурсам и извлекать данные в структурированном, полуструктурированном и неструктурированном форматах [Ткачук, 2018].

Данный процесс – одна из актуальных задач в современном мире, поскольку он позволяет организациям и частным лицам собирать ценную информацию, например, информацию о конкурентах, тенденциях отрасли, последних вакансиях и новостях и многом другом [Бенфилд, Жлемко, 2006]. Эти данные могут быть использованы для различных целей, таких как маркетинговые исследования, бизнес-аналитика и принятие решений, системы построения рекомендаций, образовательные экосистемы [Оболенский, 2020] и т.д.

Например, компания может использовать веб-скрапинг [Государев, 2020] для сбора отзывов клиентов с онлайн-площадок, чтобы улучшить свои продуктовые предложения или отслеживать репутацию своего бренда. Исследователь может использовать веб-API для сбора данных с платформ социальных сетей для изучения общественного мнения по определенной теме [Ткачук, 2018; Бенфилд, Жлемко, 2006].

В целом, сбор данных с веб-сайтов и API обеспечивает экономичный и действенный способ сбора больших объемов информации, которые могут быть использованы для анализа и принятия обоснованных решений.

Существует несколько методов сбора веб-данных, включая:

1. веб-скрапинг (web-scraping) [Государев, 2020]: этот процесс включает в себя использование программных средств для извлечения данных с веб-сайтов. Получение данных с онлайн-ресурсов может быть выполнено вручную или автоматизировано с использованием таких языков программирования, как Python;

2. использование API [Государев, 2020; Ткачук, 2018]: многие веб-сайты и онлайн-сервисы предоставляют API, которые позволяют разработчикам получать доступ к данным в структурированном формате и извлекать их. API-интерфейсы могут использоваться для сбора данных с платформ социальных сетей, веб-сайтов электронной коммерции, онлайн-порталов поиска работы и других онлайн-сервисов [Ткачук, 2018; Бенфилд, Жлемко, 2006];

3. интеллектуальный анализ данных (ИАД) включает в себя использование статистических методов и методов машинного обучения для извлечения информации из больших наборов данных. ИАД может использоваться для анализа посещаемости веб-сайта, поведения пользователей и так далее;

4. краудсорсинг (crowdsourcing) – сбор данных от большой группы людей с помощью онлайн-опросов или других форм пользовательского контента. Краудсорсинг может быть использован для сбора данных о потребительских предпочтениях, мнениях и другой демографической информации;

5. веб-аналитика включает в себя использование таких инструментов, как Google Analytics для отслеживания трафика веб-сайта, поведения пользователей и других показателей. Веб-аналитика может предоставить ценную информацию о производительности веб-сайта и вовлеченности пользователей.

Однако существует несколько ограничений при использовании различных методов сбора веб-данных, включая [Государев, 2020; Купер и др., 2006]:

1. качество данных. Собранные наборы данных могут содержать ошибки, несоответствия и неточности, которые могут повлиять на точность и надежность анализа;

2. технические ограничения. Сбор данных может потребовать специальных навыков и инструментов для преодоления технических проблем, таких как ограничения скорости доступа к веб-сайту, структура веб-сайта, форматирование данных и объем данных;

3. Непредсказуемые изменения. Веб-сайты могут изменить свою структуру, содержание или политику в любое время, что может повлиять на доступность и качество данных;

4. конфиденциальность данных. Сбор персональных данных с веб-сайтов без разрешения может нарушать законы о конфиденциальности и этические соображения;

5. юридические проблемы. Удаление данных с веб-сайтов может нарушать законы об авторском праве и соглашения об условиях предоставления услуг;

6. объемы данных. Сбор действительно больших объемов данных требует значительных инвестиций в инфраструктуру, а также особые подходы для реализации процесса сбора.

Для действительно большого объема данных процесс сбора может занять значительно больше времени, особенно в случаях, когда веб-сайт или API имеют ограничения по скорости. Эта статья посвящена сравнению распределенной обработки сбора данных с базовым конвейером сбора, чтобы выяснить, какое решение является наиболее подходящим для крупномасштабного сбора данных в условиях ограничений сайта-источника на доступ к нему.

### Вопросы, поднимаемые в статье

В этой статье поднимаются следующие вопросы:

1. рассмотреть, что такое ограничение скорости доступа к веб-API;
2. исследовать, что такое распределенный процесс сбора веб-данных;
3. сравнить простой процесс сбора данных с распределенным.

### Основная часть

Ограничение скорости веб-сайта или API относится к максимальному количеству запросов, которые могут быть сделаны к данному онлайн-ресурсу или API в течение определенного периода времени. Это ограничение обычно устанавливается владельцами веб-сайта или поставщиком API для предотвращения злоупотреблений, защиты своих серверов от перегрузки и DDoS атак, а также для обеспечения справедливого использования для всех пользователей. При превышении лимита скорости веб-сайт или API могут выдать сообщение об ошибке или временно заблокировать дальнейшие запросы до тех пор, пока лимит не будет сброшен. Это может повлиять на конвейеры сбора данных, когда объем данных очень велик.

Рассмотрим, как работает базовый процесс сбора данных. Схема процесса сбора данных показана на рисунке 1 ниже.

Процесс сбора данных может начинаться с запроса метаданных из некоторого API, который предоставляет подробную информацию о данных, подлежащих сбору. Эти метаданные могут включать такие сведения, как тип данных, формат, в котором они доступны, и любые ограничения на доступ к данному датасету.

Как только метаданные получены, следующим шагом является получение списка страниц, содержащих интересующие нас данные, с помощью API. Как только у нас есть список страниц, мы извлекаем список элементов, содержащих данные, которые мы хотим собрать. Этими элементами могут быть текст, изображения, видео, документы или любой другой тип контента, доступный на данной странице.

После того, как мы определили соответствующие элементы, мы извлекаем каждый элемент с помощью API и сохраняем результаты в системе хранения. Эта система хранения может быть базой данных, файловой системой или любым другим типом хранилища данных, подходящим для хранения собранных данных в рамках поставленной задачи.

На протяжении всего процесса сбора данных необходимо предусмотреть обработку ошибок и исключительных ситуаций, возникающих из-за проблем с подключением к сети, ошибок сервера или других факторов. Также необходимо реализовать механизмы ограничения скорости запросов к целевому онлайн-ресурсу, чтобы избежать перегрузки API или возникновения других проблем с производительностью. Это может быть сделано при помощи добавления искусственных задержек в программный код базового конвейера сбора данных.

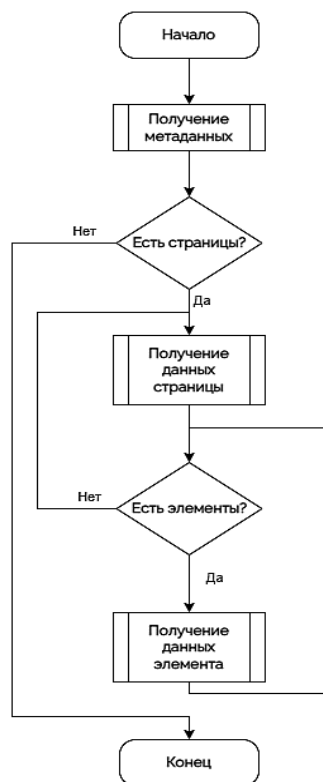


Рис.1. Алгоритм работы базового конвейера сбора данных  
 Fig.1. Basic web data collection processing algorithm

Распределенный процесс сбора веб-данных использует несколько распределенных узлов, также называемых агентами [Евзеньях, 2021; Артамонов, 2016]. Каждый узел или агент отвечает за сбор небольшой части данных, которые затем агрегируются и используются для создания всеобъемлющего набора данных. Этот подход часто используется в крупномасштабных проектах веб-скрапинга, где объем собираемых данных слишком велик для обработки одним компьютером или пользователем. Распределяя рабочую нагрузку между несколькими узлами, процесс может быть завершен гораздо быстрее и эффективнее [Евзеньях, 2021].

Использование распределенного подхода имеет следующие преимущества [Артамонов, 2016]:

- масштабируемость. Распределенный конвейер сбора данных позволяет собирать большие объемы данных даже из нескольких источников одновременно, что может быть выгодно для крупномасштабных проектов;

- устойчивость. При распределенном подходе сбор данных может продолжаться, даже если один узел или сервер выходит из строя. При возникновении проблем задача может быть обработана повторно необходимое количество раз;

- скорость. Распределяя рабочую нагрузку между несколькими узлами или серверами, сбор данных может быть завершен быстрее, чем при централизованном подходе;

- гибкость. Распределенный подход обеспечивает большую гибкость с точки зрения того, где и как собираются данные.

Однако распределенный процесс сбора данных имеет некоторые недостатки [Евзеньях, 2021; Артамонов, 2016]:

- сложность. Распределенный подход может быть значительно более сложным в настройке и управлении, чем централизованный подход, а также требует дополнительных ресурсов и опыта использования.

- синхронизация. Необходимо обеспечить, чтобы все узлы не собирали несколько раз один и тот же фрагмент данных, а также чтобы данные были согласованы на всех узлах;

– стоимость. Распределенный подход может потребовать больше ресурсов и инфраструктуры, настройка и обслуживание которых могут быть дорогостоящими.

Распределенный конвейер сбора данных, рассмотренный в этой статье, показан на рисунке 2 ниже.

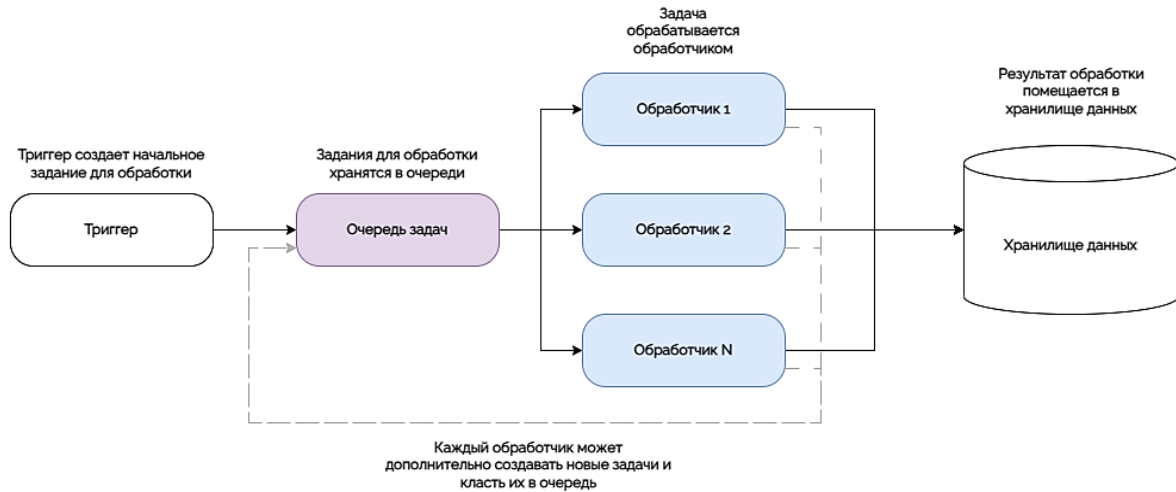


Рис. 2. Архитектура распределенного конвейера сбора данных  
 Fig. 2. A distributed data collection pipeline architecture

Конвейер распределенного сбора данных начинается с некоторого триггера, которым может быть событие *webhook*, событие ручного запуска или запланированное cron-задание, инициирующее процесс сбора данных. Триггер добавляет новую задачу в очередь задач, которая содержит все задачи для последующей обработки агентами.

Затем набор узлов извлекает задачи из очереди задач и обрабатывает их. Каждый узел имеет возможность обрабатывать задачу независимо, не полагаясь на другие узлы. Как только узел завершает задачу, он либо помещает следующую задачу в очередь задач, либо сохраняет итоговые результаты в хранилище.

Хранилище используется для хранения итоговых результатов конвейера сбора данных. Это может быть база данных или файловая система, в зависимости от типа собираемых данных. Хранилище получает данные от узлов и сохраняет их таким образом, чтобы они были легко доступны для дальнейшей обработки.

Архитектура такого распределенного конвейера сбора данных предназначена для обработки больших объемов данных и обеспечения эффективного и надежного сбора данных.

В общем случае распределенный процесс сбора данных может быть представлен в виде многоканальной системы массового обслуживания (СМО) с неограниченной длиной очереди вида  $M / M / n / \infty$  в нотации Кендалла, где  $n$  – число каналов. Рассчитаем характеристики работы данной СМО в предельном режиме.

На вход данной СМО поступает пуассоновский поток заявок с интенсивностью  $\lambda = 150$  заявок в минуту. Пусть число каналов  $n = 8$ . Среднее время обработки одной задачи  $t_{\text{обсл}}$  составляет 2,8 секунды. Тогда интенсивность потока обслуживания  $\mu = \frac{60}{t_{\text{обсл}}} = 21,43$  заявок в минуту.

Пусть  $\psi = \frac{\lambda}{n \cdot \mu} = 7$  – показатель нагрузки на один канал. Тогда предельные вероятности можно рассчитать следующим образом:

Вероятность простоя каналов:

$$p_0 = \left( \sum_{k=0}^n \frac{n^k \psi^k}{k!} + \frac{\psi^{n+1}}{n!(n-\psi)} \right)^{-1} = 0,0001.$$

Вероятность отказа в обслуживании:

$$p_{\text{отк}} = 0.$$

Вероятность возникновения очереди:

$$p_{\text{queue}} = \frac{\psi^{n+1}}{n!(n-\psi)} = 0,1021.$$

Относительная пропускная способность:

$$Q = 1 - p_{\text{отк}} = 1.$$

Среднее число заявок в очереди:

$$L_{\text{queue}} = \frac{\psi^{n+1}}{n!(1-\frac{\psi}{n})^2} p_0 = 0,8164.$$

Среднее число заявок в системе:

$$L = L_{\text{queue}} + \psi = 7,8164.$$

Имитационная модель может быть использована для сравнения двух различных подходов к процессу сбора данных [Карпов, 2005]. Имитационная модель – это математическое представление реальной системы или процесса. Она используется для моделирования поведения системы в различных условиях и для прогнозирования результатов различных сценариев. В контексте сбора данных имитационная модель может использоваться для моделирования поведения пользователей и серверов, а также для имитации сбора данных из различных источников [Лейман, 2015; Машенко, 2012; Бочкарева, 2009].

Имитационная модель также может быть использована для тестирования различных стратегий сбора данных и оценки их эффективности. Например, модель может быть использована для сравнения производительности различных алгоритмов веб-скрапинга или для оценки влияния различных ограничений на доступ к данным или форматов данных на процесс сбора [Лейман, Бочкарева, Сучкова, 2015; Бочкарева, 2009].

Мы используем следующие критерии для сравнения двух подходов:

1. эффективность по времени работы;
2. эффективность по стоимости;
3. сложность модели.

Теперь мы можем сравнить обе модели и проверить их в соответствии критериям, определенным выше. В таблице 2 мы можем видеть параметры API, используемые в обеих имитационных моделях.

Таблица 1

Table 1

Параметры API, используемые в имитационной модели  
Simulation model API parameters

Количество элементов на одной странице	100
Общее число элементов	140000
Число страниц	1400
Среднее время ответа сервера, с	
50-й перцентиль	0,1
95-й перцентиль	0,15
99-й перцентиль	0,7
Задержка между запросами, с	1
Время обработки одного элемента, с	1

Процентили в ответе Web API [Лопез и др., 2019] относятся к распределению времени отклика для выбранной конечной точки (endpoint) используемого API. 50-й процентиль, также известный как медиана, представляет время отклика, при котором половина запросов занимает меньше, а половина – больше. 95-й процентиль представляет время отклика, при котором 95% запросов занимают меньше времени и только 5% – больше. Аналогично, 99-й процентиль представляет время отклика, которое у 99% запросов занимает меньше времени и только у 1% – больше [Лопез и др., 2019].

На рисунке 3 изображена имитационная модель распределенного конвейера обработки данных. Данная модель разработана и протестирована с использованием среды имитационного моделирования AnyLogic [Карпов, 2005].

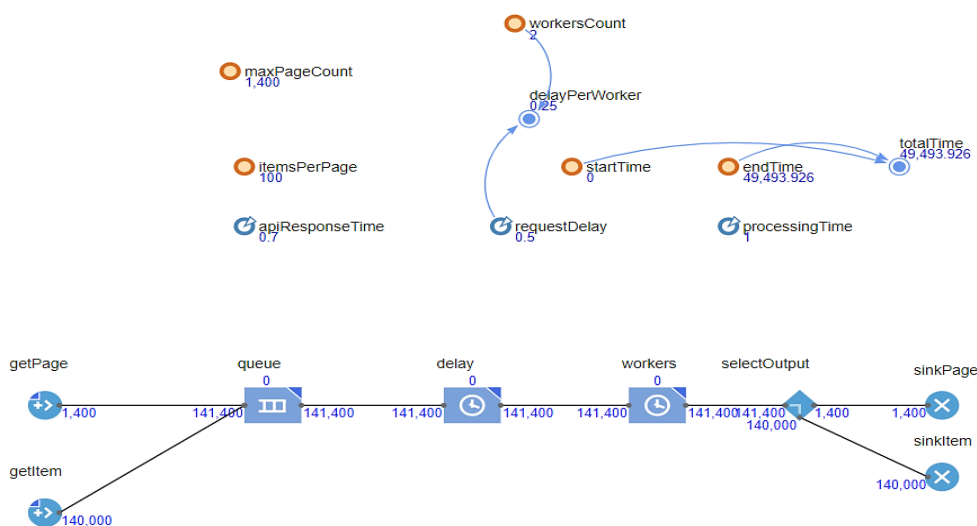


Рис. 3. Имитационная модель распределенного конвейера сбора данных  
 Fig. 3. Simulation model of the distributed data collection pipeline

Результаты моделирования простого конвейера обработки данных с помощью Excel представлены в таблице 2.

Таблица 2  
 Table 2

Результат моделирования простого конвейера сбора данных  
 Simple data collection pipeline model results

Процентиль	50	95	99
Получение страниц	1539	1609	2379
Получение элементов и обработка	293999	300999	377999
Суммарное время, с	295538	302608	380378
Суммарное время, минут	4925,63	5043,47	6339,63
Суммарное время, часов	82,094	84,058	105,661
Округленное время, часов	83	85	106
Округленное время, дни	3,46	3,54	4,42
Количество ядер VM	2		
Стоимость часа использования ядра [Правила тарификации Compute Cloud, 2023]	1,12		
Объем памяти (в Гб)	1		
Стоимость часа использования 1 Гб	0,39		
Стоимость использования, рубли	218,29	223,55	278,78

Результаты моделирования распределенного конвейера обработки данных с помощью AnyLogic представлены в таблице 3. Помимо использования процентилей, выполнено также моделирование ответов сервера при помощи распределения Эрланга.

Таблица 3  
Table 3

Результат моделирования распределенного конвейера сбора данных  
 Distributed data collection pipeline model results

Процентиль/Распределение	50	95	99	Эрланг(2, 0.07)	Эрланг(2, 0.1)
Суммарное время, с	42420,601	45955,651	84841,201	46665,193	49493,926
Суммарное время, минут	707,01	765,93	1414,02	777,75	824,90
Суммарное время, часов	11,784	12,765	23,567	12,96	13,75
Округленное время, часов	12	13	24	13	14
Округленное время, дней	0,50	0,54	1,00	0,54	0,58
Число элементов в очереди	141401				
Стоимость 1000 запросов к очереди	0,04876				
Использование очереди, рубли	2,02				
Количество воркеров	8				
Количество ядер VM	2				
Стоимость часа использования ядра	1,12				
Процентиль/Распределение	50	95	99	Эрланг(2, 0.07)	Эрланг(2, 0.1)
Объем памяти (в Гб)	1				
Стоимость часа использования 1 Гб, рубли	0,39				
Стоимость использования, рубли	254,50	275,54	506,98	275,54	296,58

Сравнение времени работы базового и распределенного конвейеров представлено на рисунке 4. Сравнение стоимости работы базового и распределенного конвейеров представлено на рисунке 5. Также на этих диаграммах представлены столбцы для результатов, полученных в результате моделирования ответов API для распределенного процесса с помощью распределения Эрланга.

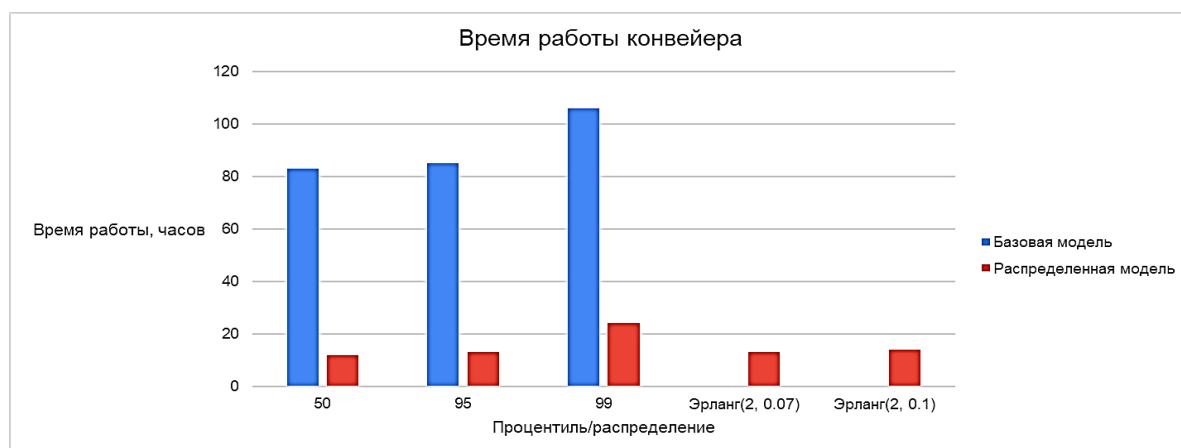


Рис. 4. Сравнение общего времени сбора данных для базовой и распределенной модели, часы  
 Fig. 4. Overall data collection time comparison for simple and distributed models in hours



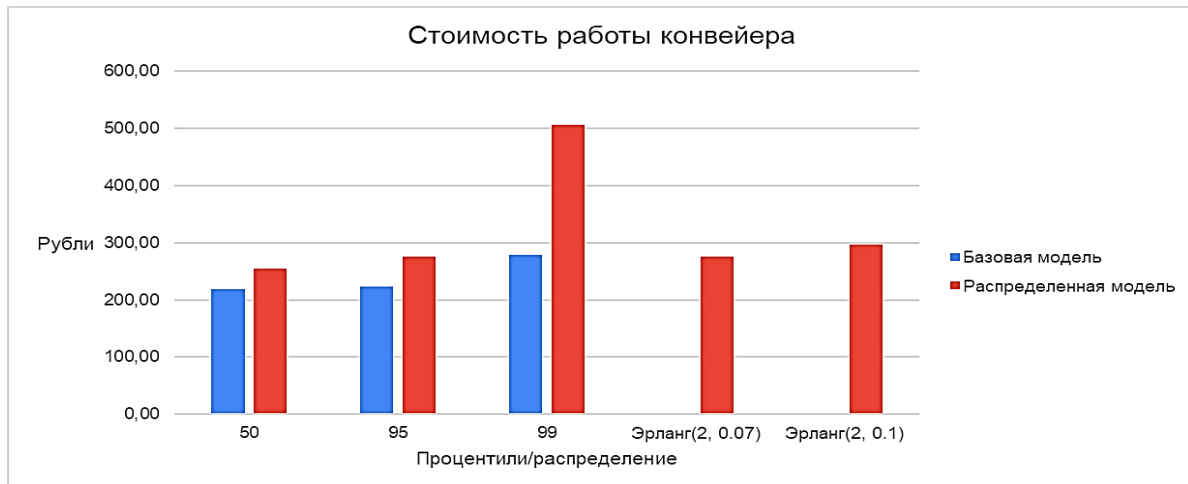


Рис. 5. Сравнение стоимости работы базовой и распределенной модели, рубли  
Fig. 5. Overall data collection cost comparison for simple and distributed models, rubles

При сопоставимой общей цене использования, распределенная модель сбора данных значительно выигрывает у обычной модели, основанной на циклах, по критерию времени работы. Однако распределенная модель архитектурно сложнее, поэтому может потребовать более высокого уровня экспертности разработчиков.

### Заключение

Таким образом, распределенный процесс сбора данных прекрасно подходит для сбора огромного количества данных в условиях ограничения на частоту доступа к веб-ресурсу:

- в распределенном процессе сбора данных запросы распределяются по множеству узлов при помощи очереди сообщений, что сокращает объем трафика от отдельного узла к некоторому онлайн-ресурсу или API. Это гарантирует, что ограничения на частоту или скорость доступа не будут превышены и что сбор данных может продолжаться без перерывов;
- распределенный конвейер сбора данных позволяет агрегировать данные сразу из нескольких источников, сокращая количество запросов, отправляемых к отдельным API или веб-сайтам;
- распределенный конвейер может горизонтально масштабироваться вверх или вниз по мере необходимости при помощи увеличения или уменьшения количества узлов, обеспечивая большую гибкость при сборе данных. Это означает, что при достижении ограничений могут быть добавлены дополнительные ресурсы для обработки возросшей рабочей нагрузки.
- распределенный процесс сбора данных может обеспечить большую устойчивость данных и избыточность. Если в процессе доступа к API или онлайн-ресурсу процесс выходит из строя или завершается с ошибкой, данный элемент может быть обработан повторно, собран из другого источника или обработан на другом узле.

В целом, распределенный конвейер сбора данных может обеспечить более надежный, эффективный способ сбора данных с различных веб-сайтов и API, избегая проблем, связанных с ограничениями скорости, и обеспечивая большую масштабируемость и устойчивость, обладая при этом сопоставимой стоимостью на час времени.

### Список литературы

- Артамонов, Ю. С. 2016. Разработка распределенных приложений сбора и анализа данных на базе микросервисной архитектуры / Ю. С. Артамонов, С. В. Востокин // Известия Самарского научного центра Российской академии наук. – 2016. – Т. 18, № 4-4. – С. 688-693. – EDN YGSQTV.
- Бочкарева Е.В. 2009. Применение имитационного моделирования для исследования процесса сбора и обработки данных микроконтроллерными устройствами / Е.В. Бочкарева, Л.И. Сучкова, А.Г. Якунин // Научно-технические ведомости СПбГПУ. - Санкт-Петербург: Изд-во СПбГПУ, 2009. - №3. с. 158-166.

- Государев И. Б. 2020. WEB SCRAPING как программный процесс извлечения и обработки данных в вебе / И. Б. Государев, Н. Н. Жуков, П. А. Бабарицкий // Современное образование: традиции и инновации. – № 1. – С. 146-152. – EDN THLLEG.
- Карпов, Ю. Г. 2005. Имитационное моделирование систем : введение в моделирование с AnyLogic 5 / Ю. Г. Карпов ; Юрий Карпов. – СПб. : БХВ-Петербург, 2005. – ISBN 5-94157-148-8. – EDN QMPBSH.
- Лейман, А. В, Сучкова Л. И. 2015. Имитационная модель сбора и обработки данных в распределенной системе / А. В. Лейман, Л. И. Сучкова // Ползуновский альманах. – № 1. – С. 45-47. – EDN VLAHSH.
- Лейман А.В., Бочкарева Е.В., Сучкова Л.И. 2015. Имитационная модель распределенных систем сбора и обработки данных / Лейман А.В., к.т.н. Бочкарева Е.В., д.т.н. Сучкова Л.И. // Научный журнал «БЪДЕЩИТЕ ИЗСЛЕДВАНИЯ - 2015». София: «Бял ГРАД-БГ» ООД - 96 стр.
- Мащенко, Е. Н. 2012. Исследование автоматизированных процессов управления качеством ИТ-сервисов с использованием имитационного моделирования / Е. Н. Мащенко, В. И. Шевченко // – № 125. – С. 142-147. – EDN TPNAFX.
- Оболенский Д.М., Шевченко В.И. 2020. Концептуальная модель интеллектуальной образовательной экосистемы. Экономика. Информатика. 47(2): 390–401. DOI: 10.18413/2687-0932-2020-47-2-390-401.4e4e
- Правила тарификации для Yandex Compute Cloud. 2023. URL:<https://cloud.yandex.ru/docs/compute/pricing>
- Benfield, J. A., & Szlemko, W. J. 2006. Internet-based data collection: Promises and realities. Journal of Research Practice, 2(2), Article D1. Retrieved [date of access] from, <http://jrp.icaap.org/index.php/jrp/article/view/30/51>
- Cooper, Charles & Cooper, Sharon & del Junco, Jesús & Shipp, Eva & Whitworth, Ryan & Cooper, Sara. 2006. Web-based data collection: Detailed methods of a questionnaire and data gathering tool. Epidemiologic perspectives & innovations : EP+I. 3. 1. 10.1186/1742-5573-3-1.
- Eyzenakh, D. S. 2021. High performance distributed web-scraper / D. S. Eyzenakh, A. S. Rameykov, I. V. Nikiforov // Proceedings of the Institute for System Programming of the RAS. – Vol. 33, No. 3. – P. 87-100. – DOI 10.15514/ISPRAS-2021-33(3)-7. – EDN SIPWXY.
- Lopez C., Morato D., Magana E., Izal M. 2019. Effective analysis of secure web response time / C. Lopez, D. Morato, E. Magana, M. Izal // TMA 2019 - Proceedings of the 3rd Network Traffic Measurement and Analysis Conference : 3, Paris, 2019. – P. 145-152. – DOI 10.23919/TMA.2019.8784652. – EDN VBOMXZ.
- Tkachyk, D. A. 2018. Software development for collection financial historical data using Web-scraping / D. A. Tkachyk, Yu. P. Noskovenko, O. V. Zaharchuk // 3rd International Conference on Innovations and development patterns in Technical and Natural Sciences, Berlin, 20 апреля 2018 года. – Berlin: Premier Publishing s.r.o. – P. 6-8. – EDN YVPGDJ.

### References:

- Artamonov, Yu. S. 2016. Razrabotka raspredelennykh prilozheniy sbora i analiza dannykh na baze mikro-servisnoy arkhitektury / Yu. S. Artamonov, S. V. Vostokin // Izvestiya Samarskogo nauchnogo tsentra Rossiyskoy akademii nauk. – 2016. – Т. 18, № 4-4. – S. 688-693. – EDN YGSQTV.
- Bochkareva E.V. 2009. Primenenie imitatsionnogo modelirovaniya dlya issledovaniya protsessa sbora i obrabotki dannykh mikrokontrollernymi ustroystvami / E.V. Bochkareva, L.I. Suchkova, A.G. Yakunin // Nauchno-tehnicheskie vedomosti SPbGPU. - Sankt-Peterburg: Izd-vo SPbGPU, 2009. - №3. с. 158-166.
- Gosudarev I. B. 2020. WEB SCRAPING kak programmnyy protsess izvlecheniya i obrabotki dannykh v vebe / I. B. Gosudarev, N. N. Zhukov, P. A. Babaritskiy // Sovremennoe obrazovanie: traditsii i innovatsii. – № 1. – S. 146-152. – EDN THLLEG.
- Karpov, Yu. G. 2005. Imitatsionnoe modelirovanie sistem : vvedenie v modelirovanie s AnyLogic 5 / Yu. G. Karpov ; Yuriy Karpov. – СПб. : BKhV-Peterburg, 2005. – ISBN 5-94157-148-8. – EDN QMPBSH.
- Leyman, A. V, Suchkova L. I. 2015. Imitatsionnaya model' sbora i obrabotki dannykh v raspredelennoy sisteme / A. V. Leyman, L. I. Suchkova // Polzunovskiy al'manakh. – № 1. – S. 45-47. – EDN VLAHSH.
- Leyman A.V., Bochkareva E.V., Suchkova L.I. 2015. Imitatsionnaya model' raspredelennykh sistem sbora i obrabotki dannykh / Leyman A.V., k.t.n. Bochkareva E.V., d.t.n. Suchkova L.I. // Nauchnyy zhurnal «B""DESHCHITE IZSLEDVANIYA - 2015». Sofiya: «Byal GRAD-BG» OOD - 96 str.
- Mashchenko, E. N. 2012. Issledovanie avtomatizirovannykh protsessov upravleniya kachestvom IT-servisov s ispol'zovaniem imitatsionnogo modelirovaniya / E. N. Mashchenko, V. I. Shevchenko // – № 125. – S. 142-147. – EDN TPNAFX.
- Obolenskiy D.M., Shevchenko V.I. 2020. Kontseptual'naya model' intellektual'noy obrazovatel'noy ekosistemy. Ekonomika. Informatika. 47(2): 390–401. DOI: 10.18413/2687-0932-2020-47-2-390-401.4e4e

- Pravila tarifkatsii dlya Yandex Compute Cloud. 2023. URL:<https://cloud.yandex.ru/docs/compute/pricing>
- Benfield, J. A., & Szlemko, W. J. 2006. Internet-based data collection: Promises and realities. *Journal of Research Practice*, 2(2), Article D1. Retrieved [date of access] from, <http://jrp.icaap.org/index.php/jrp/article/view/30/51>
- Cooper, Charles & Cooper, Sharon & del Junco, Jesús & Shipp, Eva & Whitworth, Ryan & Cooper, Sara. 2006. Web-based data collection: Detailed methods of a questionnaire and data gathering tool. *Epidemiologic perspectives & innovations* : EP+I. 3. 1. 10.1186/1742-5573-3-1.
- Eyzenakh, D. S. 2021. High performance distributed web-scraper / D. S. Eyzenakh, A. S. Rameykov, I. V. Nikiforov // *Proceedings of the Institute for System Programming of the RAS.* – Vol. 33, No. 3. – P. 87-100. – DOI 10.15514/ISPRAS-2021-33(3)-7. – EDN SIPWXY.
- Lopez C., Morato D., Magana E., Izal M. 2019. Effective analysis of secure web response time / C. Lopez, D. Morato, E. Magana, M. Izal // *TMA 2019 - Proceedings of the 3rd Network Traffic Measurement and Analysis Conference* : 3, Paris, 2019. – P. 145-152. – DOI 10.23919/TMA.2019.8784652. – EDN VBOMXZ.
- Tkachyk, D. A. 2018. Software development for collection financial historical data using Web-scraping / D. A. Tkachyk, Yu. P. Noskovenko, O. V. Zaharchuk // *3rd International Conference on Innovations and development patterns in Technical and Natural Sciences*, Berlin, 20 апреля 2018 года. – Berlin: Premier Publishing s.r.o. – P. 6-8. – EDN YVPGDJ.

**Конфликт интересов:** о потенциальном конфликте интересов не сообщалось.

**Conflict of interest:** no potential conflict of interest related to this article was reported.

#### ИНФОРМАЦИЯ ОБ АВТОРАХ

**Оболенский Денис Михайлович**, аспирант кафедры информационных технологий и компьютерных систем Севастопольского Государственного университета, г. Севастополь, Россия

**Шевченко Виктория Игоревна**, кандидат технических наук, доцент, Заведующий базовой кафедрой «Корпоративные информационные системы» Севастопольского государственного университета, г. Севастополь, Россия

**Ченгар Ольга Васильевна**, кандидат технических наук, доцент кафедры «Корпоративные информационные системы» Севастопольского государственного университета, г. Севастополь, Россия

#### INFORMATION ABOUT THE AUTHORS

**Denis M. Obolensky**, Postgraduate Student of the Department of Information Technology and Computer Systems, Sevastopol State University, Sevastopol, Russia

**Victoria I. Shevchenko**, Candidate of Technical Sciences, Associate Professor, Head of the basic department "Corporate Information Systems", Sevastopol State University, Sevastopol, Russia

**Olga V. Chengar**, Candidate of Technical Sciences, Associate Professor of the department "Corporate Information Systems", Sevastopol State University, Sevastopol, Russia