Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«Сыктывкарский государственный университет имени Питирима Сорокина»

(ФГБОУ ВО «СГУ им. Питирима Сорокина»)

Институт точных наук и информационных технологий

Кафедра прикладной информатики

Исследовательская работа

**Search Engines: Elasticsearch, Splink, Solr, Sphinx, Algolia**

Направление подготовки

09.03.03 Прикладная информатика

Направленность (профиль) программы

Прикладная информатика в экономике

Исполнитель:

Гончаров Игорь Валерьевич \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Личная подпись

Сыктывкар

2023

Оглавление

[**Тезаурус** 3](#_Toc153460724)

[**Введение** 4](#_Toc153460725)

[**1. Поисковые системы** 5](#_Toc153460726)

[**2. Механизмы поиска** 8](#_Toc153460727)

[**3. Обзор поисковых систем** 11](#_Toc153460728)

[**3.1 Elasticsearch** 12](#_Toc153460729)

[**3.2 Solr** 14](#_Toc153460730)

[**3.3 Sphinx** 16](#_Toc153460731)

[**3.4 Algolia** 18](#_Toc153460732)

[**4. Результаты экспертной оценки** 21](#_Toc153460733)

[**Заключение** 22](#_Toc153460734)

# **Тезаурус**

Данные – поддающееся многократной интерпретации представление информации в формализованном виде, пригодном для передачи, связи, или обработки.

Информация – сведения, независимо от формы их представления.

База данных – совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных.

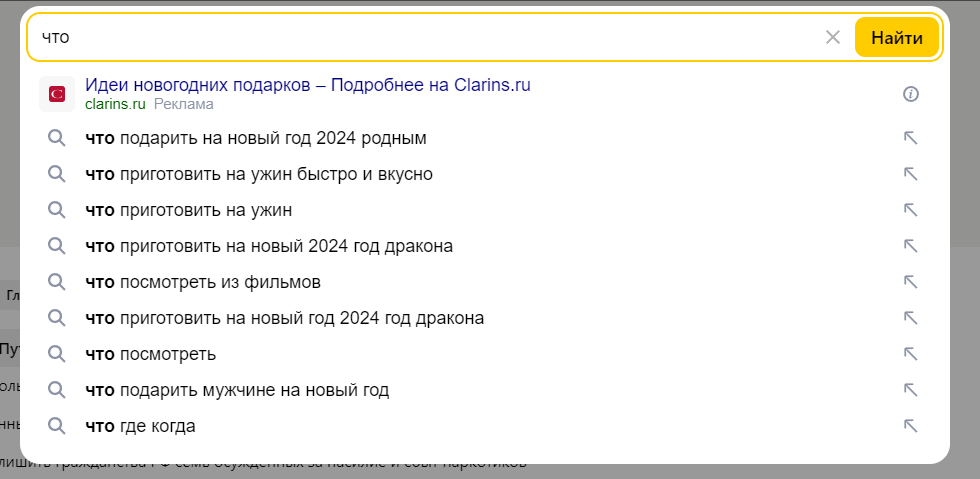
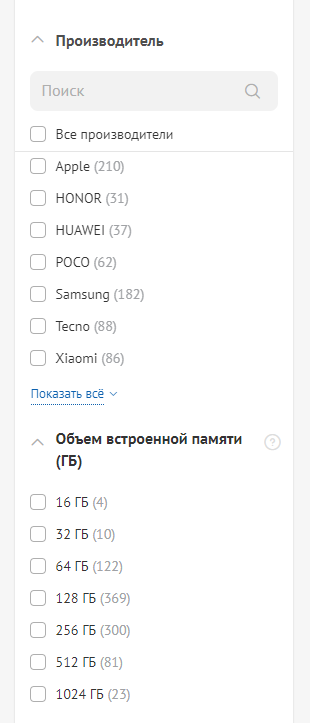
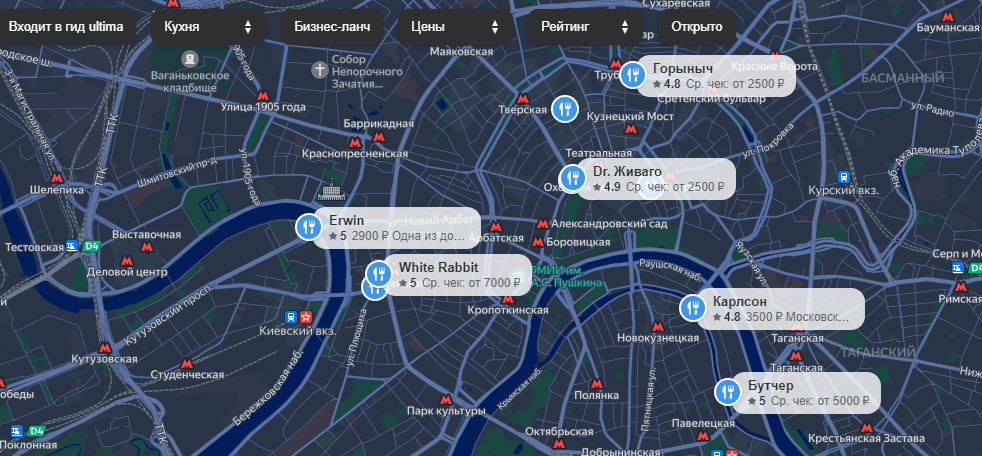
# **Введение**

Поисковая система (Search Engine) – алгоритмы и реализующая их совокупность компьютерных программ (в широком смысле этого понятия, включая аналоговые системы автоматизированной обработки информации), предоставляющая пользователю возможность быстрого доступа к необходимой ему информации при помощи поиска в обширной коллекции доступных данных. Одно из наиболее известных применений поисковых систем — веб-сервисы для поиска текстовой или графической информации во Всемирной паутине. Существуют также системы, способные искать файлы на FTP-серверах, товары в интернет-магазинах, информацию в группах новостей Usenet.

С точки зрения бизнеса, эффективная поисковая система, это мощный инструмент, способный принести больше прибыли владельцам веб-сайтов. Если поисковый механизм сайта не дает нужных результатов или его эффективность поиска слишком низкая, люди покидают веб-сайт и переходят к его конкуренту.

# **1. Поисковые системы**

Основная цель поиска — найти наиболее релевантные совпадения с запросами пользователя, исключая другой общий контент с веб-сайта. Самыми популярными функциями, которыми обладают современные поисковые системы являются:

* **полнотекстовый поиск** (по простым словам, фразам, или нескольким формам слова или фразам);
* **многополевой поиск;**
* **выделение** (визуальное указание слов, введенных в поле поиска);
* **поиск по синонимам;**
* **предложения автозаполнения**
* 
* **фасетный поиск** (поиск по атрибутам, например, на сайтах электронной коммерции фасеты используются, чтобы сообщить покупателям, сколько элементов определенной модели, размера, цвета и других атрибутов найдено)
* 
* **нечеткий поиск** (опечатки, орфографические ошибки);
* **орфографические исправления;**
* **геопространственный поиск** (для определения местоположения объекта по широте и долготе)
* 

Система должна иметь возможность сужать поиск, используя диапазоны (цена, даты, размеры и т.п.), сортировать (по популярности, дате, цене) и фильтровать (включая только желаемые параметры) данные.

Когда мы говорим о веб-приложениях, в которых информация изменяется динамически (цены, подробное описание, наличие товаров), чрезвычайно важно обновлять данные близко к реальному времени; например, в электронной коммерции или системах бронирования, чтобы показать товары и услуги, доступные в текущий момент.

Помимо общих функций, перечисленных выше, поисковые движки могут предоставлять рекомендации при поиске наиболее интересных продуктов или информации, чтобы улучшить взаимодействие с пользователем.



# **2. Механизмы поиска**

Данные, как правило, организованы в виде документов (веб-страница, word файл и т.д.). Если мы попытаемся поискать по всей странице в браузере, поиск будет осуществляться по всему содержащемуся тексту. И это удобно для большинства кейсов.

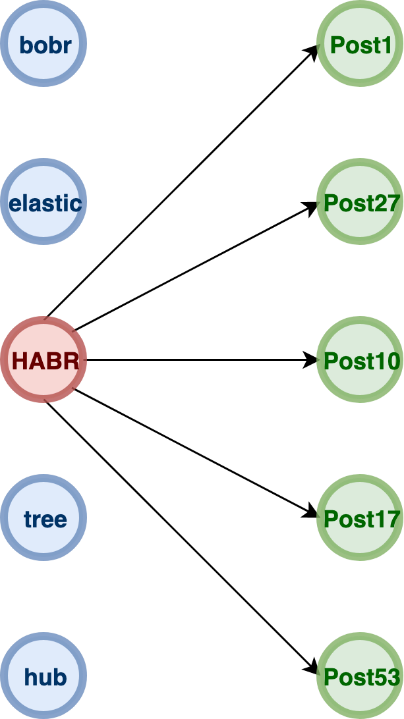
Примерно так же работают многие поисковые системы, поиск происходит по всему тексту проиндексированных страниц, а не по отдельным полям, тегам или заголовкам. Это называется полнотекстовым поиском.

Искать предстоит по огромному количеству документов и было бы разумно запомнить, что в каком документе лежит. В реляционных СУБД мы привыкли оптимизировать подобный поиск индексами.

Индекс – это сбалансированное дерево, то есть дерево, в котором длина путей (количество шагов межу узлами) не будет отличаться больше чем на один шаг.

Например, если проиндексировать любой текст, мы получаем дерево, листьями которого являются используемые в нем слова. Иными словами, мы будем знать заранее, какие слова находятся в документе и как их быстро в нем найти. Не смотря на такую удобную структуризацию данных, обход дерева звучит как не самое лучшее решение для максимально быстрого поиска.

Можно поступить иначе — собрать список всех используемых слов и узнать, в каких документах они встречаются. Да, индексация займет больше времени, но нас в первую очередь интересует именно скорость поиска, а не индексации.



Такой индекс называется обратным индексом и используется для полнотекстового поиска.

**Распределенный индекс**

Для обеспечения горизонтального масштабирования необходимо иметь возможность размещать данные на разных машинах. Нет разницы, как данные хранятся физически, важно чтобы мы имели единое логическое хранилище. Каждый инстанс поисковой системы должен стать частью одного большого индекса, или осколком(shard) разбитого индекса. Шард будет выполнять непосредственно операции по поиску и записи данных.

https://habrastorage.org/r/w1560/webt/al/ja/fv/aljafvzws50myp43grmy4hvhbrk.png

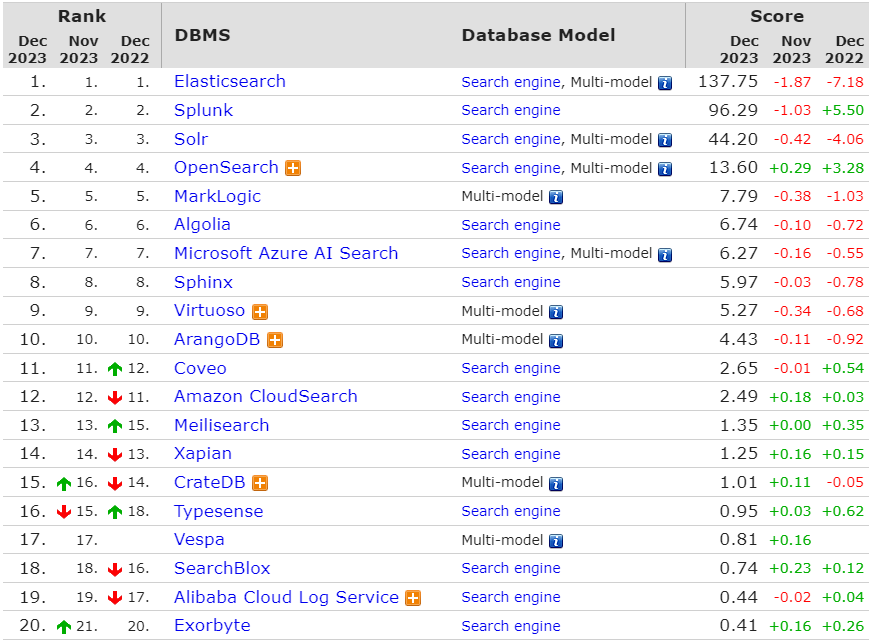
Shard в Elasticsearch — это логическая единица хранения данных на уровне базы, которая является отдельным экземпляром Lucene.

[Index](https://www.elastic.co/blog/what-is-an-elasticsearch-index) — это одновременно и распределенная база и механизм управления и организации данных, это именно логическое пространство. Индекс содержит один или более шардов, их совокупность и является хранилищем.  
  
Классическое сравнение индекса с другими базами выглядит примерно так.

|  |  |  |
| --- | --- | --- |
| **Elasticsearch** | **SQL** | **MongoDB** |
| Index | Database | Database |
| Mapping/Type | Table | Collection |
| Field | Column | Field |
| Object(JSON) | Tuple | Object(BSON) |

# **3. Обзор поисковых систем**

<https://db-engines.com/en/ranking/search+engine>



В интернете можно найти сведения о нескольких десятках поисковых инструментов, но, если необходимо найти надежное и эффективное решение для своего веб-приложения, рассмотрим некоторые наиболее популярные варианты.

Все они являются поисковыми решениями с открытым исходным кодом, с хорошей поддержкой от собственных сообществ участников. Эти решения отличает высокая производительность, масштабируемость и гибкость, но каждая из них все же имеет свои особенности, которые могут повлиять на проект.

## **3.1 Elasticsearch**



Разработчик: Elasticsearch разработан компанией Elastic.

Преимущества:

Масштабируемость и производительность.

Распределенная архитектура.

Поддержка RESTful API.

Богатый функционал для поиска и аналитики.

Интеграция с другими продуктами Elastic Stack (Kibana, Logstash, Beats).

Недостатки

Высокие требования к ресурсам, особенно к оперативной памяти.

Сложность настройки и оптимизации.

Не самая лучшая поддержка ACID-транзакций.

Алгоритм поиска

Elasticsearch основан на Apache Lucene и использует инвертированный индекс для поиска. Текст разбивается на токены, которые затем индексируются и хранятся вместе с указателем на документ.

Хранение данных

Elasticsearch хранит данные в формате JSON и разбивает их на шарды, распределяя по узлам в кластере.

Применение

Поиск и аналитика в логах.

E-commerce: поиск товаров, аналитика покупательского поведения.

Enterprise search: поиск документов, информации о сотрудниках и клиентах.

**Elasticsearch (ES)**, абсолютный лидер рейтинга поисковых систем, подтверждает свое название как действительно «эластичный», поскольку может работать в любой среде. Это технология с открытым исходным кодом, в которой используется библиотека Apache Lucene. Многие всемирно известные компании, такие как TripAdvisor, Mozilla, Github, eBay, и Netflix используют ES для своих приложений.

**Основные преимущеcтва ES**

Индексация практически в реальном времени. ES может индексировать динамически изменяющиеся данные практически мгновенно (менее чем за 1 секунду). Это делает ценным его использование в проектах, где база данных постоянно обновляется.

Высокая масштабируемость. Очевидно, что когда размер базы данных растет, то искать становится труднее. Но ES масштабируется по мере увеличения вашей БД, поэтому скорость поиска не снижается.

Хранение. ES можно использовать не только как индексатор, но и как хранилище данных. Однако, с точки зрения безопасности и надежности, целесообразно хранить данные в основной БД, используя ES только для индексации данных и хранения журналов.

Визуализация данных. Это одна из модных сегодня функций, которая хорошо реализована в ES. Elastic Stack (комбинация плагинов ES, Logstash и Kibana) — отличный инструмент для аналитики. Он позволяет в режиме реального времени отслеживать трафик приложения (общее количество посетителей, количество уникальных посетителей, IP-адреса, самые популярные запросы, наиболее запрашиваемые страницы, используемые устройства и браузеры, журналы трафика по времени суток и многое другое). Эта информация визуализируется в диаграммах, картах и ​​таблицах на панели управления.

Аналитика безопасности. Elastic Stack также является полезным инструментом для анализа безопасности. Аналитика и визуализация логов практически в реальном времени позволяет выявлять угрозы безопасности (проблемы с веб-сервером, неработающие ссылки, попытки несанкционированного доступа, места атак и т. д.).

Машинное обучение. ES может воспользоваться функциями машинного обучения, предоставляемыми коммерческим плагином X-Pack. Алгоритмы машинного обучения ориентированы на обнаружение аномалий и выбросов в данных временных рядов.

Amazon Elasticsearch Service. Amazon Elasticsearch Service обеспечивает быструю и простую настройку, работу и масштабирование ES в облаке без необходимости настраивать собственные серверы.

Однако, увы, в этом мире нет ничего идеального. Хотя ES в настоящее время является наиболее востребованным поисковым приложением, все же это еще молодая и развивающаяся технология. Не все желаемые функции доступны по умолчанию, и многие из них следует добавлять с помощью различных расширений. Например, в ES нет сообщения «Возможно Вы имели в виду ….?».

## **3.2 Solr**



Разработчик

Solr создан Apache Software Foundation.

Преимущества

Быстрый и масштабируемый.

Распределенная архитектура.

Поддержка RESTful API.

Поддержка схем и NoSQL.

Фасетный поиск и богатые возможности для фильтрации.

Недостатки

Меньше функциональности по сравнению с Elasticsearch.

Отсутствие визуализации и аналитики на уровне продукта.

Алгоритм поиска

Solr также использует Apache Lucene и инвертированный индекс для поиска. Текст разбивается на токены, которые затем индексируются и хранятся вместе с указателем на документ.

Хранение данных

Solr хранит данные в формате XML, JSON или CSV и разбивает их на шарды, распределяя по узлам в кластере.

Применение

Enterprise search: поиск документов, информации о сотрудниках и клиентах.

E-commerce: поиск товаров, аналитика покупательского поведения.

Поиск и аналитика в области медицины, образования, исследований.

**Solr** — это еще одна поисковая система, основанная на Apache Lucene, и поэтому она имеет много общих функций с ES. Но, тем не менее, они разные по архитектуре. Среди компаний, использующих Solr — Cnet, CitySearch, Bloomberg, Zappos, AOL, eTrade, Disney, Apple, NASA, MTV и другие.

**Сильные стороны Solr**

Фасетный поиск. Solr имеет расширенные возможности многогранного поиска, что делает это решение идеальным для веб-сайтов электронной коммерции, которые используют Solr для поиска и навигации по сотням тысяч стилей своих продуктов.

Богатый набор функций. Solr может похвастаться богатыми функциями полнотекстового поиска, которые легко настраиваются даже по сравнению с ES. Solr поддерживает различные реализации подсказчиков, функции выделения (визуальное указание слов, введенных в поле) и средства проверки орфографии / «Возможно Вы имели в виду?» (которые отсутствуют в ES).

Документы с разнообразным содержанием. Solr — одна из немногих поисковых систем, которая может читать документы в разных форматах, включая PDF, Word, XML или простой текст. Это идеально подходит для проектов, в которых необходимо просмотреть большое количество файлов PDF или Word на веб-сайте (включая контракты, резюме, учебные материалы, электронные книги и т. д.).

Визуализация данных. Banana — это инструмент визуализации (ответвление Kibana), который работает для Solr и позволяет администраторам отслеживать события в режиме реального времени.

Машинное обучение. Solr в сотрудничестве с Bloomberg реализовал машинное обучение (плагин Learning-to-Rank), используя концепцию повторного ранжирования документов в соответствии с оценкой более сложного запроса. В результате пользователям предоставляются возможности значительного ускорения их поиска.

Не без ложки дегтя… Solr не так быстр, как ES, и лучше всего работает со статическими данными (которые не требуют частого изменения). Причина в разных стратегиях использования кеша.  В Solr кеш является глобальными. Что означает, что когда в нем происходит даже малейшее изменение, вся индексация требует обновления. Понятно, что это не быстрый процесс. В отличие от Solr, ES обновляет кеш по сегментам.

## **3.3 Sphinx**



**Sphinx** не входит в тройку самых популярных поисковых систем, хотя и является мощной и популярной технологией. Sphinx используется в таких известных системах, как Joomla.org, CouchSurfing.org, Wikimapia.org, Tumblr.com и сотнях других приложений.

**Сильные стороны Sphinx**

Мощный и быстрый. Sphinx эволюционировал в последние годы и стал обеспечивать поиск в режиме, близком к реальному времени. Его скорость составляет более 500 запросов в секунду по 1 миллиону документов, при этом наибольшее зарегистрированное количество индексированных документов оценивается более, чем в 25 миллиардов.

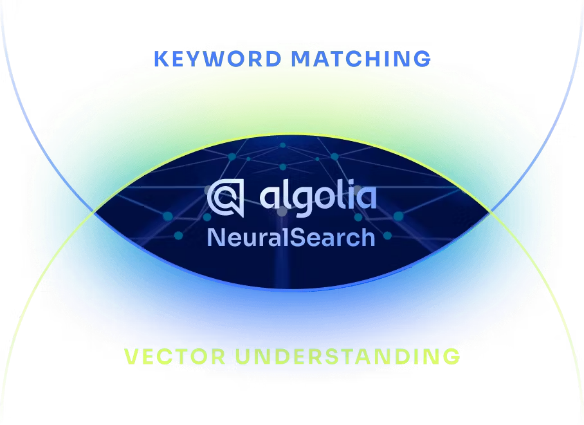
Фасетный поиск. Sphinx имеет большой опыт работы с возможностями многогранного атрибутного поиска.

Ничего бесполезного. Если нам нужны в основном общие функции поиска и не требуются какие-либо дополнительные специфические функции, такие как визуализация и анализ данных, использование Sphinx целесообразно. Он обладает достаточной быстротой и мощностью для индексирования и обработке поисковых запросов огромных объемов документов с использованием ограниченных вычислительных ресурсов, в отличие от ES, который потребляет достаточно много памяти.

Увы, Sphinx хорош в основном для структурированных данных (предопределенные текстовые поля и нетекстовые атрибуты), но это не совсем не лучший выбор для проектов, имеющих дело с неструктурированными данными (DOC, PDF, MP3 и т. д.), поскольку настройка и обработка такого типа данных требует значительных усилий от разработчиков. Это, в совокупности с определенными сложностями в настройке, делает Sphinx менее удобным в использовании, чем ES и Solr.

## **3.4 Algolia**

Algolia



Разработчик

Algolia разработан компанией Algolia.

Преимущества

Быстрый и масштабируемый.

Поддержка RESTful API.

Удобный интерфейс для настройки и управления.

Поддержка инстантного поиска и автоматического дополнения запросов.

Интеграция с популярными платформами и фреймворками.

Недостатки

Зависимость от структуры цен и условий Algolia.

Меньшая гибкость настройки по сравнению с Elasticsearch и Solr.

Алгоритм поиска

Algolia использует свой собственный алгоритм поиска, основанный на триграммах и дополнительных метриках, таких как процент совпадения, расстояние между словами и других. И еще не так давно они запустили Neural Search

Хранение данных

Algolia хранит данные в своих собственных облачных серверах и автоматически разбивает их на шарды.

Применение

Enterprise search: поиск документов, информации о сотрудниках и клиентах.

E-commerce: поиск товаров, аналитика покупательского поведения.

Мобильные приложения и SaaS-продукты.

Neural Search от Algolia

Я заинтересовался этим инновационным продуктом потому решил выделить его в отдельный пункт.  
Algolia Neural Search - это инновационный подход к полнотекстовому поиску, разработанный компанией Algolia. В основе этого подхода лежат искусственные нейронные сети и машинное обучение, которые улучшают традиционные методы поиска, делая их более интеллектуальными и точными.

Как работает Algolia Neural Search

Подготовка данных: Algolia Neural Search начинается с индексации данных. Во время этого процесса текстовая информация разбивается на индексируемые токены, а также проходит процедуры очистки, стемминга и лемматизации.

Обучение нейронных сетей: Algolia использует обученные нейронные сети для определения семантической близости между запросами и документами. Эти сети могут быть обучены на общедоступных корпусах данных или на данных, специфичных для конкретной отрасли или компании.

Ранжирование результатов: Когда пользователь отправляет поисковый запрос, Algolia Neural Search применяет нейронные сети для вычисления семантической близости между запросом и документами в индексе. На основе этой оценки алгоритм ранжирует документы, предоставляя пользователю наиболее релевантные результаты.

Дополнительные функции: Algolia Neural Search также предлагает дополнительные возможности, такие как автоматическое дополнение запросов, исправление опечаток и фильтрация результатов на основе пользовательских предпочтений и контекста.

Преимущества Algolia Neural Search заключаются в его способности учитывать семантическую близость и контекст запросов, что делает поиск более точным и релевантным. Кроме того, использование машинного обучения и нейронных сетей позволяет алгоритму постоянно улучшаться, адаптируясь к изменяющимся потребностям пользователей и поведению поиска.

# **4. Результаты экспертной оценки**

# **Заключение**

Выбор полнотекстового поискового движка зависит от множества факторов, таких как размер проекта, требования к производительности, масштабированию, функционалу и стоимости. Важно тщательно изучить возможности каждого движка и сопоставить их с потребностями вашего проекта. Elasticsearch и Solr являются мощными и гибкими решениями, подходящими для крупных проектов и комплексных задач. Amazon CloudSearch и Algolia предоставляют удобные облачные решения с простым управлением и интеграцией.

Как видно из этой таблицы, различия между ES, Solr и Sphinx минимальны. Все они выполняют свою главную задачу — обеспечивают эффективный и быстрый поиск.

На мой взгляд, для большинства проектов лучшим выбором будет ES —  быстрый, гибкий и простой в работе, не только обеспечивает быстрый поиск, но и сам по себе может служить средством хранения. За счет эффективной визуализации в режиме реального времени ES удобен при поиске данных в журналах логов для быстрого выявления проблем с приложениями.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Система | Назначение и возможности пакета | Преимущества | Недостатки | Алгоритм поиска | Хранение данных |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |