УДК 004.422.636.7

И.Н. Бабак, Н.Э. Безверхий

Автоматизированная система анализа хеш-функций для выбора способа хеширования данных в едином информационном пространстве

Национальный аэрокосмический университет им. Н.Е. Жуковского «Харьковский авиационный институт»

Проведен анализ проблемы выбора методов хеширования данных. Предложена разработка автоматизированной системы для сравнения эффективности хеш-функций. Описаны шаги алгоритма работы автоматизированной системы. В системе реализована возможность для пользователя исследовать встроенные функции и свои собственные. В качестве показателей для сравнения функций использованы минимальное, максимальное и наиболее вероятное число коллизий. Визуализация результатов анализа представлена в виде диаграммы распределения данных в ячейках хеш-таблицы. Определены дальнейшие направления улучшения автоматизированной системы анализа хеш-функций.

Ключевые слова: хеширование, хеш-функция, хеш-таблица, хранение информации, коллизия, эффективность хеш-функции, программное обеспечение.

Введение

Перспективным направлением для развития авиационной промышленности, а в частности, для поддержки жизненного цикла проектов беспилотных авиационных комплексов (БАК), является создание единого информационного пространства (ЕИП) [1]. Можно рассматривать ЕИП как совокупность баз и банков данных, телекоммуникационных сетей, технологий их создания и использования, функционирующих на основе единых принципов, обеспечивающих информационное взаимодействие всех участников проекта разработки БАК. При разработке ЕИП одной из задач является необходимость обеспечения единообразного способа взаимодействия всех участников с помощью электронного обмена данными, возможность быстрого поиска данных за счет рационального способа их хранения.

В рамках ЕИП информации разного вида, касающейся изделия, проекта, производства, управления, ресурсов и т.д. будет много, поэтому вопрос о целостности данных и безопасности их передачи, а также организации хранения и обеспечения быстрого доступа является актуальным. Популярной структурой, позволяющей оптимизировать время доступа к данным, является хеш-таблица. Практически все современные языки программирования имеют реализации хеш-таблиц в своих библиотеках. Чаще всего работа с ними осуществляется в виде словарей (или ассоциативных массивов), представляющих собой контейнеры множества пар ключ-значение [2, 3]. Кроме того, хеширование является распространенным способом проверки целостности информации в файлах данных при передаче и их хранении путем использования проверки значения, рассчитанного по определённому алгоритму хеширования [4, 5].

1 Постановка задачи

Идея хеширования основана на распределении ключей в массиве. Распределение осуществляется с помощью вычисления для каждого ключа некоторого значения хеш-функции, которое служит индексом для массива. Лучшей будет та

хеш-функция, которая дает уникальное значение для различных объектов. На данный момент существует много хеш-функций [6, 7]. Если не использовать стандартные, то достаточно просто можно сформировать и собственную хеш-функцию. Однако, нужно проверять, удовлетворяет ли выбранная хеш-функция следующим требованиям: она должна распределять ключи по ячейкам хеш-таблицы как можно более равномерно; должна достаточно просто вычисляться; даже небольшое изменение входных данных должно приводить к существенному изменению значения функции.

Проблема заключается в том, что проверка равномерности распределения ключей на больших объемах данных без использования автоматизированных средств практически невозможна, а программные продукты для проведения такой сравнительной оценки хеш-функций на рынке не представлены. Разработчики программного обеспечения сами проводят исследование и обоснование выбора способа хеширования, а иногда просто используют известную функцию без анализа ее эффективности для применения к конкретным данным.

Поэтому для решения данной проблемы предлагается разработать автоматизированную систему, которая позволит пользователю ввести анализируемую функцию, определить ее эффективность на определенном наборе данных путем анализа количества коллизий, равномерности распределения ключей, и, таким образом, решить вопрос выбора наиболее подходящей хеш-функции.

2 Разработка структуры и алгоритма работы автоматизированной системы анализа хеш-функций

Разрабатываемая система для автоматизации анализа эффективности хеш-функции представляет интерес как для решения задач криптографии, связанными с проверкой целостности данных, так и для ее использования в качестве подсистемы, определяющей, какую хеш-функцию стоит применить для обеспечения минимального времени поиска хранимых данных.

Многие хеш-функции уже пользуются популярностью в решении задач хеширования, например RS, LY, ROT13, H37, FAQ6 и другие [6, 7]. Каждая из этих функций реализует определенный функционал, направленный на то, чтобы с наибольшей вероятностью получить уникальное значение ключа. Аналоги разрабатываемой системы зачастую не распространены в свободном доступе.

Тем не менее, в качестве аналогов можно рассмотреть программы, использующие алгоритмы хеширования и программы, предназначенные для распознавания функций, т.к. предполагается наличие модуля в системе для ввода пользователем функции с клавиатуры

В качестве аналогов данной системы были рассмотрены программы, предназначенные для хеширования, такие как:

- MD5 Hash Generator;
- File Hash Checker;
- HashTab.

В результате анализа был сделан вывод, что на данный момент существует некоторое количество систем, применяющих алгоритмы хеширования, однако они не позволяют провести сравнительный анализ существующих, а, тем более, уникальных пользовательских хеш-функций.

Алгоритм работы данного приложения можно разделить на следующие этапы:

- в качестве входного параметра принимается введенная пользователем

хеш-функция либо выбранная из списка встроенных в программе функций;

– если пользователь ввел свою функцию, то проводится распознавание хеш-функции из введенной пользователем строки;

– на тестовом наборе данных проводится хеширование данных с помощью выбранной хеш-функции с заданным размером хеш-таблицы;

– рассчитываются статистические показатели полученного распределения данных;

– осуществляется вывод графика распределения данных по ячейкам хеш-таблицы;

– проводится расчет и вывод статистических показателей распределения данных.

Обобщенная структура приложения представлена на рис. 1.

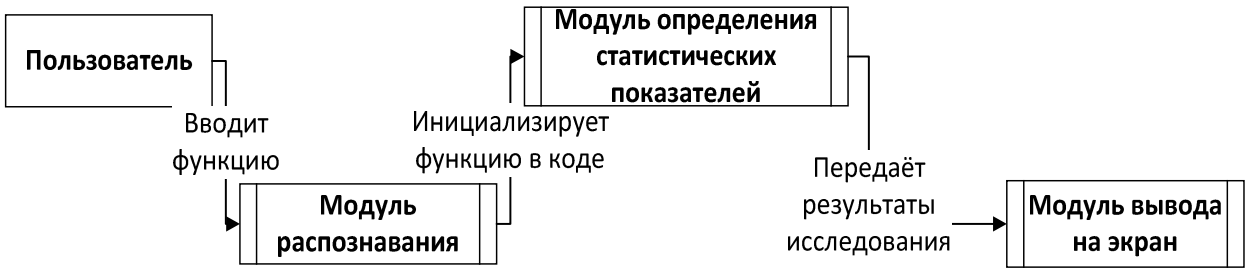

Рис. 1 – Обобщенная структура приложения

В качестве статистических показателей хеш-функции пользователю предоставляются:

– диаграмма распределения количества коллизий по ячейкам;
– модальное значение;
– минимальное количество коллизий;
– максимальное количество коллизий.

Среднее значение коллизий в данном случае не представляет интерес, т.к. для разных функций на одних и тех же данных это будет одно и то же значение. Вместо этого показателя модальное значение будет более информативной характеристикой распределения коллизий, т.к. даст наиболее часто встречающееся число коллизий в ячейке, что будет показывать средне значение времени поиска данных.

Минимальное и максимальное количество коллизий показывает, насколько существенным был разброс между отдельно взятыми ячейками.

Алгоритм распознавания введенной пользователем хеш-функции выполнен следующим образом:

– строка разбирается посимвольно;

– каждый символ, либо совокупность символов попадет в стек операторов либо стек операндов, соответственно;

– результат выражения вычисляется в зависимости от приоритетности операторов.

На данный момент блок распознавания функции поддерживает в качестве основных операций для распознавания сложение, вычитание, деление, умножение, возведение в степень. В качестве аргумента *x* для функции *f(x)* можно выбрать по умолчанию сумму кодов всех символов данных или их произведение.

3 Анализ результатов исследования хеш-функций

В ходе анализа были рассмотрены статистические результаты для разных хеш-функций с одинаковыми входными данными и одинаковой длиной хеш-таблицы, а также был проведен анализ зависимости статистических показателей распределения данных от количества ячеек хеш-таблицы.

В качестве тестовых данных использован американский словарь из проекта Ispell на 62075 слов. В качестве анализируемых функций рассмотрены RS, LY, ROT13, H37, FAQ6. Значения статистических показателей в зависимости от размерности хеш-таблицы приведены в табл. 1.

Таблица 1

Показатели распределения коллизий при изменении длины ячеек хеш-таблицы

Размерность	Хеш-функция	Минимальное количество коллизий	Максимальное количество коллизий	Мода
100	H37	350	4216	412
	FAQ6	301	669	643
	Rot13	301	691	627
	Ly	305	817	643
	Rs	356	685	607
1000	H37	17	3162	36
	FAQ6	24	82	60
	Rot13	25	121	60
	Ly	26	116	60
	Rs	37	106	63

Исходя из результатов, представленных в таблице 1, видно, что при разном количестве ячеек порядок привлекательности функций по показателю максимального количества коллизий изменяется мало: функция H37, имеющая существенные выбросы по количеству ключей, попавших в одну и ту же ячейку, на малом количестве ячеек, показывает максимум и на таблице большей размерности. Наиболее равномерно данные были распределены с помощью хеш-функции FAQ6, т.к. она показала минимальные отклонения как в большую так и меньшую стороны от значения моды.

Исходя из таблицы, можно понять, насколько равномерно распределение данных для той или иной хеш-функции. В программе предусмотрен вывод графической интерпретации распределения данных по соответствующим ячейкам хеш-таблицы. Одного взгляда на диаграмму функции H37 (рис. 2) достаточно, чтобы понять, что данная функция не даёт хорошего распределения и секторы заполнены очень неравномерно. Поэтому ее справедливо можно считать наихудшей из анализируемых функций в данном примере.

На рис. 2 продемонстрирован пример ввода функции пользователем. В данном случае в качестве аргумента *x* выступает произведение кодов символов в слове. Такая произвольная хеш-функция показала более равномерное распределение по сравнению с функцией H37.

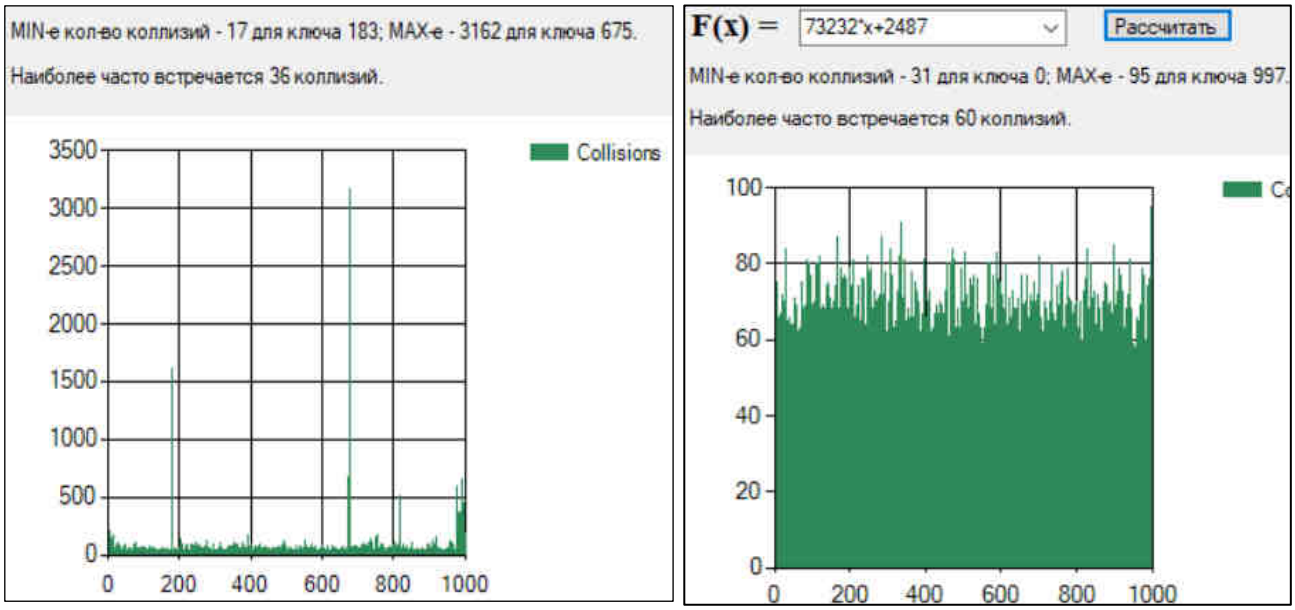

Рис. 2 – Диаграмма распределения коллизий для хеш-функции H37 и произвольной функции пользователя

Наилучшие результаты показали хеш-функции Rs и FAQ6 (рис. 3).

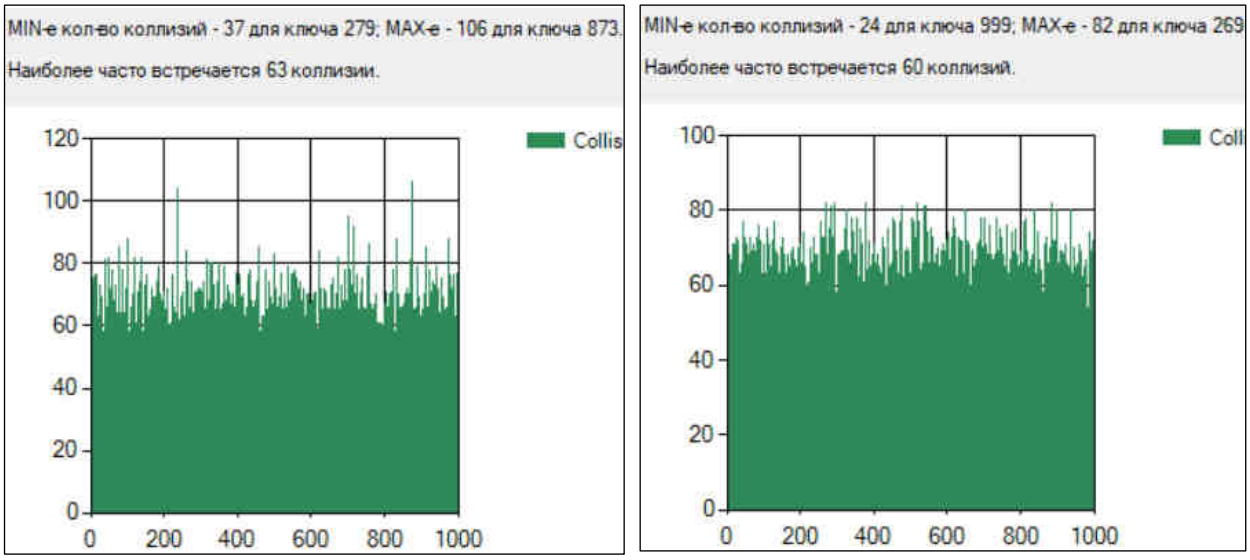

Рис. 3 – Диаграмма распределения коллизий для функции Rs и FAQ6

Стоит отметить, что количество ячеек не должно быть слишком малым или слишком большим по отношению к объему входных данных, т.к. в этом случае по полученным результатам нельзя будет сделать выводы об эффективности функций. Например, если принять значение количества ячеек, равным пяти, то весь объем из 62075 слов будет распределен между пятью столбцами и, очевидно, будет неравномерным для любой хеш-функции. С другой стороны, задав значение количества ячеек 62075, равному количеству слов во входном файле, можно провести проверку на равномерность заполнения таблицы. Пример результатов такого исследования показан на рис. 4.

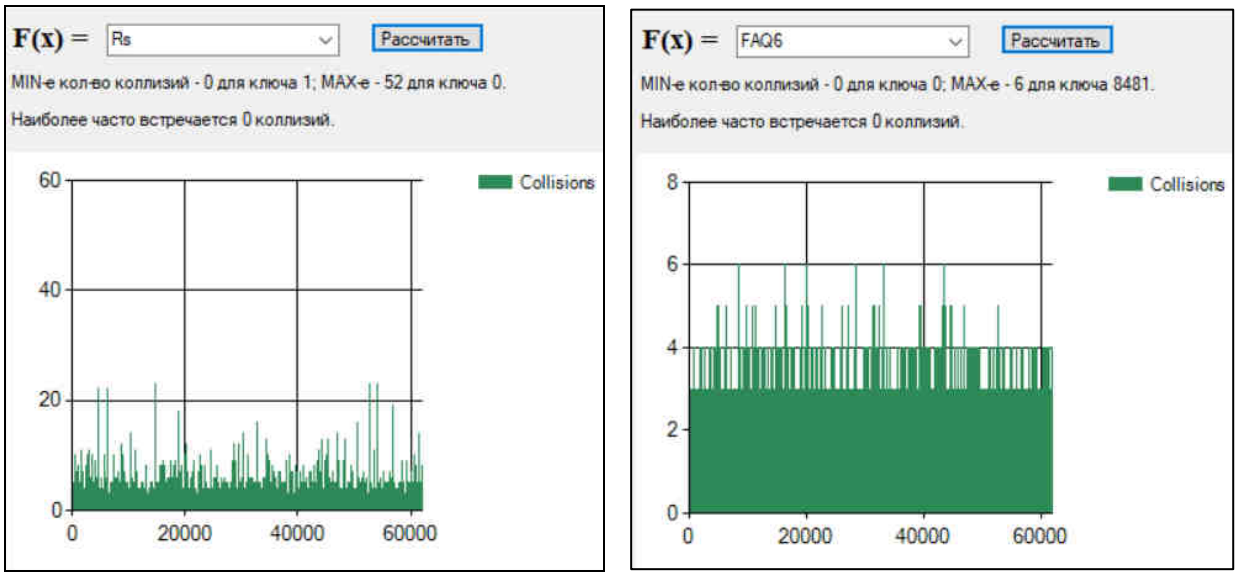

Рис. 4 – Диаграммы распределения коллизий функций Rs и FAQ6 для хеш-таблицы при совпадении количества ячеек таблицы и элементов входных данных

Из результатов, представленных на рис. 4, видно, что, несмотря на достаточное количество ячеек для каждого входного элемента, рассматриваемые функции не гарантируют отсутствие коллизий и остаются ячейки, для которых ключ не был сформирован.

Выводы

Анализ показал, что функции дают разные показатели распределения в зависимости от внутренней реализации, и можно построить рейтинг привлекательности функций с помощью предлагаемой автоматизированной системы исходя из таких показателей как минимальное и максимальное, наиболее частое количество коллизий, а также используя визуализацию диаграммы распределения коллизий.

В качестве перспективных направлений для дальнейшего развития автоматизированной системы анализа хеш-функций планируется предоставить пользователю более широкий выбор настроек для входных данных (задавать количество ячеек, выбирать файлы исходных данных); расширить список распознаваемых программой операций за счет использования регулярных выражений; создать рейтинг просмотренных хеш-функций; предоставить возможность попарного сравнения хеш-функций.

Список литературы

1. Каратанов, А. В. Метод создания единого информационного пространства авиационного конструкторского бюро [Текст] / А. В. Каратанов // Наука і техніка Повітр. Сил Збройних Сил України. – Х., 2013. – Вип. 2 (11). – С. 97–102.

2. Кормен, Т.Х. Алгоритмы. Построение и анализ [Текст] / Т.Х. Кормен, Ч.И. Лейзерсон, Р.Л. Ривест. – М.: «Вильямс», 2016. – 1328 с.

3. Основы криптографии [Текст] / А. П. Алферов, А. Ю. Зубов, А. С. Кузьмин, А. В. Черемушкин. – 2-е изд., испр. и доп. – М.: Гелиос АРВ, 2002. – 480с.

4. Бабенко, Л.К. Современные алгоритмы блочного шифрования и методы

их анализа [Текст] / Л.К. Бабенко, Е.А. Ищукова. – М.: Гелиос АРВ, 2006. – 376 с.

5. Шнайер, Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си [Текст] / Б. Шнайер. – 2-е изд.– М.: Диалектика, 2003. – 610 с.

6. Savage, B. A Guide to Hash Algorithms SANS Institute 2003 [Electronic resource]. – Mode of access: https://www.giac.org/paper/gsec/2853/guide-hash-algorithms/104822.

7. Partow, A. General Purpose Hash Function Algorithms [Electronic resource]. – Mode of access: http://www.partow.net/programming/hashfunctions/.

Поступила в редакцию 30.11.2017

Автоматизована система для аналізу хеш-функцій для вибору способу хешування даних в єдиному інформаційному просторі

Проведено аналіз проблеми вибору методів хешування даних. Запропоновано розробку автоматизованої системи для порівняння ефективності хеш-функцій. Описано кроки алгоритму роботи автоматизованої системи. У системі реалізовано можливість для дослідження користувачем вбудованих функцій та власних. В якості показників для порівняння функцій використано мінімальне, максимальне та найбільш ймовірне число колізій. Візуалізація результатів дослідження представлена у вигляді діаграми розподілу даних в елементах хеш-таблиці. Визначено подальші напрямки покращення автоматизованої системи аналізу хеш-функцій.

Ключові слова: хешування, хеш-функція, хеш-таблиця, зберігання інформації, колізія, ефективність хеш-функції, програмне забезпечення.

The Automated Hash Functions Analyzing System for Choice of Data Hashing Method in the Single Information Space

The automated system development for the hash functions effectiveness comparing is proposed. The automated system algorithm steps are described. The system implements the ability for users to explore the built-in functions and their own ones. The function comparison indicators are the minimum, maximum and most probable number of collisions. The results of analysis are presented as data distribution diagram in the hash table cells. The further directions of hash functions analysis automated system improvement are defined.

Keywords: hashing, hash function, hash table, data storage, collision, hash function efficiency, software.

Сведения об авторах:

Бабак Ирина Николаевна – к.т.н., доцент, доцент каф. 105 «Информационные технологии проектирования», Национальный аэрокосмический университет им. Н.Е. Жуковского «Харьковский авиационный институт», Украина.

Безверхий Никита Эдуардович – студент каф. 105 «Информационные технологии проектирования», Национальный аэрокосмический университет им. Н.Е. Жуковского «Харьковский авиационный институт», Украина.