электронный журнал
МОЛОДЕЖНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51038.

УДК 004.415

Особенности реализации информационной системы онлайн кластеризации

***Кулажский А.А.**, студент*
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Автоматизированные системы обработки информации и управления»

Научный руководитель: Филиппович А.Ю., к.т.н., доцент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана
philippovich@list.ru
chernen@bmstu.ru

Информационная система онлайн кластеризации представляет собой веб-приложение, предоставляющее специалистам по интеллектуальному анализу данных функции кластеризации данных в числовых форматах методами MST[1] и Fuzzy-C-Means[2].

С точки зрения реализации, приложение имеет несколько ключевых особенностей, а именно: клиент-серверная архитектура, технология CGI[3] в основе программы, необходимость использования сторонних библиотек по машинному обучению. Освещение перечисленных аспектов стало предметом данной статьи.

Обоснование выбранного средства разработки

Для начала приведем доказательства необходимости использования C++ в качестве средства разработки данной системы.

Наличие библиотек. Для облегчения процесса разработки требуется подключать внешние библиотеки. Поэтому очень важным критерием является наличие соответствующих бесплатных библиотек по машинному обучению.

Скорость выполнения. Программа должна выполняться максимально быстро, чтобы не загружать процессор.

Возможности интеграции. Средство разработки должно позволять интегрироваться CGI-программе с внешней средой, получать доступ к переменным окружения, соединяться с базами данных, соединяться с другими серверами по сети и т.п.

[1] http://www.ics.uci.edu/~eppstein/161/960206.html
[2] http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/cmeans.html
[3] http://www.w3.org/CGI/

Межпроцессорное взаимодействие. В один и тот же момент времени может быть запущено несколько экземпляров одной и той же программы, поэтому средство разработки должно иметь возможность межпроцессорного взаимодействия.

Переносимость. Важным фактором является переносимость, которая может быть от версии к версии операционной системы и сопутствующего ПО, например БД. Переносимость между клонами ОС FreeBSD <=> Linux. Переносимость на принципиально другую платформу Unix <=> Windows.

На основе всего вышесказанного критериям оценки качества можно присвоить следующие весовые коэффициенты.

Таблица 1

Критерии оценки качества и их коэффициенты

1. Скорость выполнения	2 α	0.2
2. Возможности интеграции	2 α	0.2
3. Межпроцессорное взаимодействие	α	0.1
4. Переносимость	2 α	0.2
5. Наличие библиотек	3 α	0.3

Согласно методу взвешенной суммы выполнено следующее условие: $\sum \alpha_i = 1$

На основании перечисленных критериев составим сравнительную таблицу характеристик для различных средств разработки.

Таблица 2

Сравнительная таблица

Средство разработки	1	2	3	4	5
C++	Хорошо	Отлично	Отлично	Хорошо	Отлично
MS VisualC	Отлично	Отлично	Отлично	Плохо	Плохо
Delphi	Хорошо	Отлично	Отлично	Плохо	Плохо
Java	Плохо	Плохо	Плохо	Отлично	Отлично
Unix Shell	Плохо	Отлично	Плохо	Плохо	Плохо
Perl	Плохо	Отлично	Отлично	Отлично	Хорошо
PHP	Плохо	Отлично	Плохо	Отлично	Плохо
ASP	Плохо	Отлично	Плохо	Плохо	Плохо

Таблица 3

Шкала перевода качественных показателей в количественные

Качественный показатель	Отлично (соответствует в полной мере)	Хорошо (соответствует с некоторыми ограничениями)	Плохо (не соответствует)
Количественный показатель	1	0,6	0,3

Таблица 4

Итоговая таблица сравнения средств разработки

	α	C++	MS VisualC	Delphi	Java	Unix Shell	Perl	PHP	ASP
1 Скорость выполнения	0,2	0,12	0,2	0,12	0,06	0,06	0,06	0,06	0,06
2 Возможности интеграции	0,2	0,2	0,2	0,2	0,06	0,2	0,2	0,2	0,2
3 Межпроцессорное взаимодействие	0,1	0,1	0,1	0,1	0,03	0,03	0,1	0,03	0,03
4 Переносимость	0,2	0,12	0,06	0,06	0,2	0,06	0,2	0,2	0,06
5 Наличие библиотек	0,3	0,3	0,09	0,09	0,3	0,09	0,18	0,09	0,09
Сумма	1	**0,84**	0,65	0,55	0,65	0,44	**0,74**	0,58	0,44

Таким образом, средство разработки C++ является лучшим для реализации программы с функциями кластеризации данных. Язык Perl был выбран в качестве средства разработки модуля загрузки пользовательского файла, так как в данном случае не требуются библиотеки машинного обучения. Для реализации алгоритмов кластеризации и для работы с технологией CGI были выбраны открытые C++ библиотеки mlpack[4] и cgicc[5] соответственно.

Архитектура системы и технология CGI

Система имеет клиент-серверную архитектуру, так как существует необходимость выполнения сложной логики алгоритмов кластеризации. В ходе разработки приложения возникли некоторые трудности, связанные с выполнением программы на стороне сервера: пользователям запрещается запускать exe-файлы, хранящиеся на сервере. Таким образом, стандартными способами невозможно было запустить desktop-приложение разработанное автором изначально. Поэтому для организации интерактивного взаимодействия с

[4] http://mlpack.org/

[5] http://www.gnu.org/software/cgicc/

приложением через веб-браузер была использована технология CGI, позволившая пользователям загружать файлы с данными для обработки на сервер и получать итоговые кластеры в веб-браузере после выполнения алгоритма.

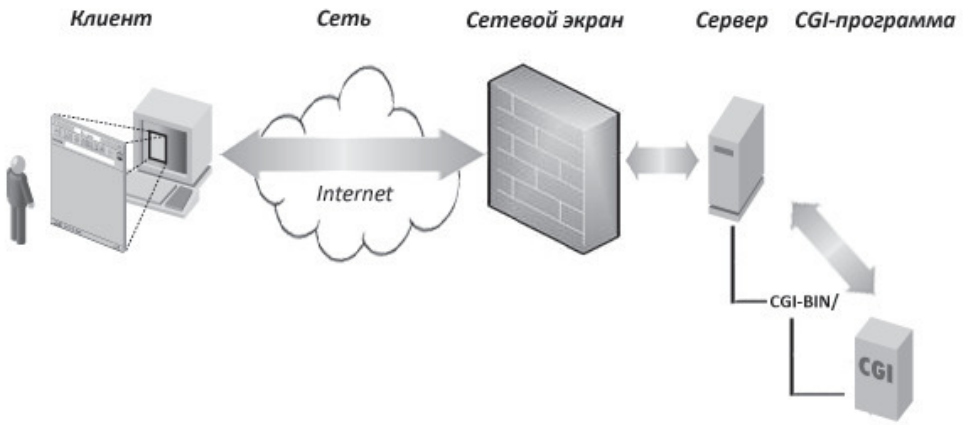

Рис. 1. Общая архитектура

CGI - стандартизированный метод, который позволяет выполняемым файлам на сервере генерировать веб-контент. Такие файлы называют CGI-скриптами или CGI-программами. Для работы с CGI необходимо предварительно задать соответствующие настройки сервера: разрешить выполнение CGI-скриптов из определенной директории. После этого требуется сформировать html-страницу для правильного обращения к CGI-приложению.

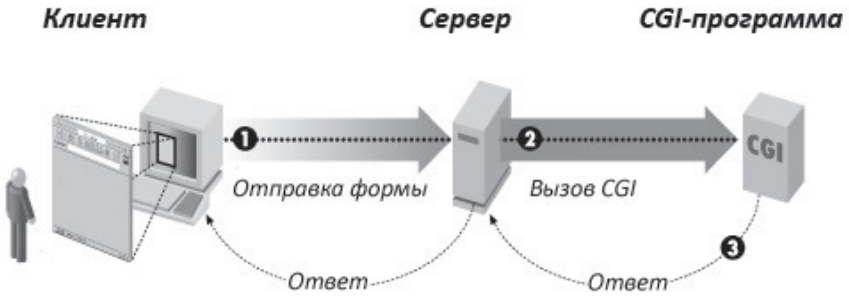

Рис. 2. Вызов CGI программы

Последовательность действий при работе с CGI выглядит следующим образом.

Действие1: Пользователь просматривает HTML-документ и встречает ссылку на страницу, содержащую форму (используется тэг <FORM>). Пользователь вводит в полях ввода на форме требуемые данные и нажимает на кнопку SUBMIT (отправить) или графическую кнопку IMAGE. HTML-тэг <FORM> имеет два обязательных атрибута:

METHOD и ACTION. Атрибут ACTION определяет URL, в качестве которого должно быть имя CGI-программы, размещаемой в каталоге. Атрибут METHOD, принимающий значение GET или POST, определяет механизм передачи данных серверу.

Действие2: Веб-браузер собирает введенные на форме данные, определяет способ передачи данных в зависимости от указанного метода (GET или POST) и передает вызов Веб-серверу.

Действие3: Веб-сервер получает вызов через сокетное соединение. Сервер разбирает сообщение на части и определяет, что это метод POST или GET. Далее запускается CGI-взаимодействие.

Действие4: Веб-сервер задает переменные окружения. Переменные окружения (environment variable) играют роль доски объявлений при обмене данными между Веб-сервером и CGI-программой. Обычно используют следующие переменные: server_name, request_method, path_info, script_name, content_type, content_length и ряд других. Когда CGI-программа вызывается посредством формы (наиболее распространенный вариант), браузер передает серверу длинную строку, в начале которой указан полный путь до CGI-программы. Далее следуют другие данные, называемые данными пути, и передаются CGI-программе через переменную окружения path_info.

Действие5: Веб-сервер запускает CGI-программу, располагаемую по умолчанию в каталоге \сервер\cgi-bin. Однако можно создавать и свои виртуальные каталоги.

Действие6: CGI-программа анализирует переменные окружения и определяет, что отвечает, например, на POST.

Действие7: CGI-программа получает тело сообщения через стандартный поток ввода (stdin). Переменная окружения content_length сообщает, сколько данных находится в сообщении.

Действие8: CGI-программа выполняет некоторые действия, в нашем случае, происходит выполнение основного модуля системы - кластеризации данных

Действие9: CGI-программа вне зависимости от метода GET или POST возвращает результат всегда через стандартный поток выхода (stdout).

Действие10: Веб-сервер возвращает результат Веб-браузеру.

Структура приложения

Структурно информационная система состоит из двух модулей – модуля загрузки пользовательского файла на сервер и модуля обработки пользовательского файла. Результат работы первого модуля передается в качестве входного параметра второму

модулю. Модуль обработки запрашивает параметры кластеризации и повторно вызывается с указанными параметрами.

Модуль загрузки пользовательского файла на сервер выполняет функции выбора пользователем файла на компьютере и загрузки данного файла во временную директорию на сервер. Модуль реализован на языке Perl и представляет собой CGI-скрипт, сохраняющий выбранный файл во временную директорию с соответствующими правами. В конце своей работы модуль передает путь к загруженному файлу в качестве параметра модулю обработки пользовательского файла и предлагает пользователю начать обработку загруженного на сервер файла.

Модуль обработки пользовательского файла получает путь к файлу в качестве параметра из ‘html-form’ по нажатию кнопки типа ‘submit’. После предобработки данных модуль выводит в браузер html-страницу с формой для задания параметров кластеризации. Аналогично предыдущему случаю, модуль вызывается повторно с параметрами кластеризации и происходит кластеризация данных. По окончанию работы модуль выводит в браузер html-страницу с результатами кластеризации, хранящимися в матрицах, в виде таблиц с номерами кластеров, значениями объектов и их порядковыми номерами.

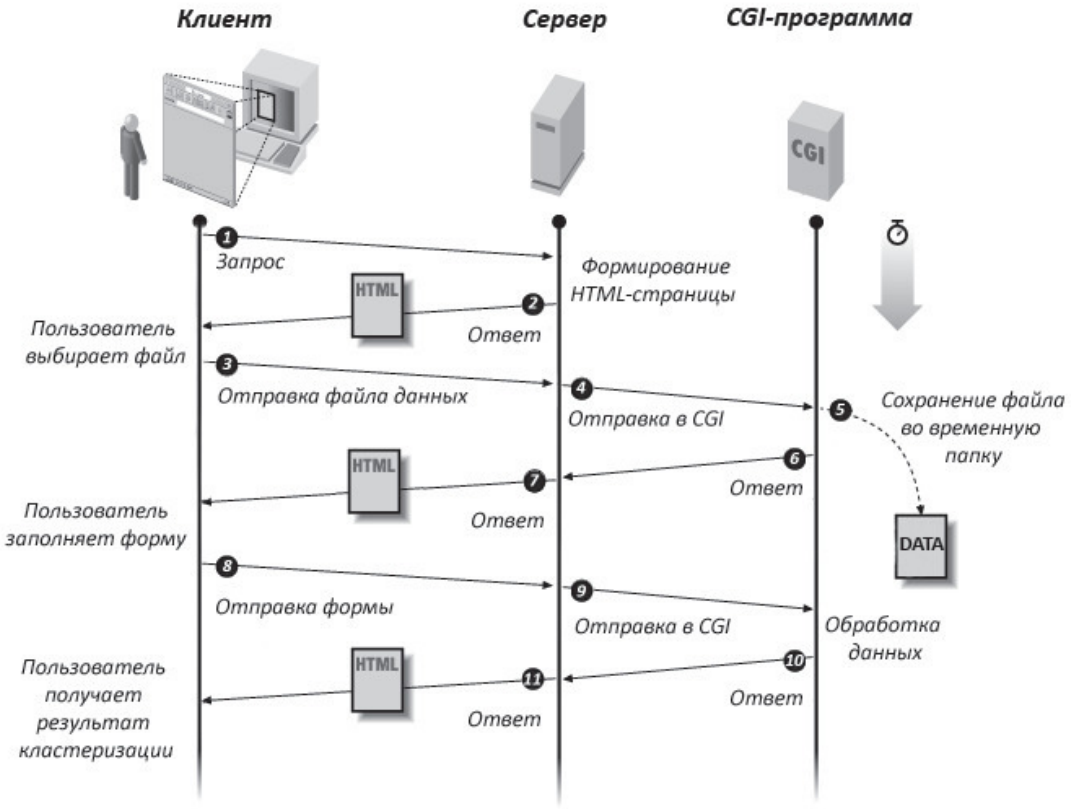

Рис. 3. Общая схема взаимодействия

1. Пользователь заходит на сайт и посылает запрос на получение html-страницы.

2. Сервер выдает html-страницу, содержащую форму выбора файла на компьютере пользователя.
3. Пользователь выбирает файл и нажимает кнопку ‘submit’.
4. Сервер вызывает CGI-скрипт, отвечающий за загрузку файла на сервер
5. CGI-скрипт (модуль загрузки пользовательского файла) сохраняет файл во временную директорию на сервере.
6. CGI-скрипт отправляет в ответ форму ввода параметров кластеризации.
7. Сервер перенаправляет html-страницу клиенту.
8. Пользователь отправляет форму с заполненными значениями параметров.
9. Сервер вызывает CGI-скрипт и передает ему параметры.
10. CGI-программа производит кластеризацию данных по основному алгоритму и возвращает результаты кластеризации в виде html-страницы.
11. Сервер перенаправляет html-страницу клиенту.

Таким образом, использование среды разработки C++ совместно с технологией CGI позволило производить сложные вычисления согласно алгоритмам кластеризации данных на стороне сервера и дало пользователям возможность загружать файлы на сервер, задавать входные параметры и получать результаты вычислений.

Список литературы

1. Нейский И.М., Филиппович А.Ю. Методика адаптивной кластеризации фактографических данных на основе интеграции алгоритмов MST и Fuzzy C-means // Известия высших учебных заведений. Проблемы полиграфии и издательского дела. М.: Изд-во МГУП, 2009. №3. С. 48-61.
2. Нейский И.М., Филиппович А.Ю. Разработка тарифной политики для клиентов брокерского обслуживания на базе методов адаптивной кластеризации // Прикладная информатика, №1 (31). 2011. С. 3-11.
3. Anil K Jain, R.C. Dubes. Algorithms for Clustering Data. Prentice Hall, New Jersey, 1988.
4. Кевин Мельтцер, Брент Михальски. Разработка CGI-приложений на Perl. М.:«Вильямс», 2001.
5. *C++* Web Programming (CGI Programming) // Tutorialspoint. Режим доступа: http://www.tutorialspoint.com/cplusplus/cpp_web_programming.htm (дата обращения: 20.02.14).
6. David Cutting. HOW-TO Write a CGI Program in C/C++ // PurplePixie.org 2005. Режим доступа: http://www.purplepixie.org/cgi/howto.php (дата обращения: 20.02.14)

7. CGI Programming in C++ // Codecall.net Tutorial forum. 2012.Режим доступа: http://forum.codecall.net/topic/72818-cgi-programming-in-c/ (дата обращения: 20.02.14).