УДК 004.912-004.75

А.Н. Волокита, И.А. Чесниший, А.В. Федосенко, Н.И. Мищеряков

ОБЛАЧНЫЙ СЕРВИС ДОБАВЛЕНИЯ ЦИФРОВЫХ ВОДЯНЫХ ЗНАКОВ

Аннотация: разработано программное обеспечение – сервис, предоставляющий пользователям вычислительные ресурсы в облаке для добавления цифровых водяных знаков на документ в формате изображения, выполнены экспериментальные исследования и анализ основных характеристик предложенной распределенной системы.

Ключевые слова: cloud computing, цифровой водяной знак, вейвлет-преобразование, стеганография.

Цифровые водяные знаки как средство защиты информации

Цифровые водяные знаки (ЦВЗ) получили свое название по аналогии с водяными знаками на банкнотах и других ценных бумагах и применяются для защиты авторских прав мультимедийных файлов. ЦВЗ делятся на два типа — видимые и невидимые. Невидимые ЦВЗ внедряются в цифровые данные, чтобы пользователю было трудно их выявить, а видимые ЦВЗ обычно представляют собой информацию, которая идентифицирует автора, например текст или логотип [1].

На рис. 1 представлено исходное изображение и его водяной знак, который при наложении не будет виден пользователю.

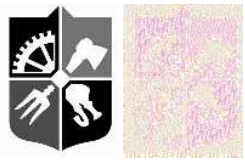

Рис. 1 – Пример невидимого водяного знака

ЦВЗ применяются для защиты от несанкционированного копирования цифровых данных, а также как средство защиты документов с фотографиями, например паспортов или водительских удостоверений. При этом изменение водяного знака свидетельствует о подделке всего документа. Невидимые ЦВЗ встраиваются в компьютерные файлы, не заметны для пользователя и позволяют обнаружить несанкционированное изменение файла [2].

Алгоритмы добавления водяных знаков

Современные алгоритмы создания водяных знаков делятся на две группы: пространственные и частотные. Наибольшее применение находят частотные алгоритмы, использующие дискретное вейвлет-преобразование (ДВП). [3,4]

Дискретное вейвлетное преобразование (Discrete Wavelet Transformation) обрабатывает каждую строку и столбец изображения с помощью частотного фильтра, при этом разбивает изображение на высокочастотные и низкочастотные области [5,6].

После одного этапа ДВП обрабатываемый фрагмент делится на четыре сегмента:

- LL – низкие частоты по строкам и столбцам;
- HL – высокие частоты по строкам и низкие по столбцам;
- LH – низкие частоты по строкам и высокие по столбцам;
- HH – высокие частоты по строкам и столбцам.

На каждом следующем этапе обрабатывается только низкочастотная область (LL), так как в высокочастотных областях обычно не содержится важной информации.

Выделенная архитектура сервиса

Мультитенантность - это возможность изолированно обслуживать пользователей из разных организацией (независимых подписчиков) в рамках одного сервиса. Выделенная архитектура (single-tenant) предполагает, что для каждого пользователя предоставляется собственная инфраструктура, в том числе логические или физические сервера. Мультитенантная архитектура является кардинальным способом снижения стоимости вычислительных ресурсов и хранилища для SaaS решений за счет минимально необходимого количества используемых ресурсов (разделяемой инфраструктуры) и их максимальной загрузки. [7]

Разработанный сервис **Waterificator** построен на основе выделенной архитектуры, что обусловлено следующими причинами:

Изолированность. Пользователям выделяются отдельные ресурсы, таким образом перекрыт потенциальный канал утечки данных.

Отказоустойчивость. Мультитенатное приложение более уязвимо к сбою экземпляра, чем выделенное приложение. Сбой выделенного развертывания влияет только на клиента, который использует данное приложение, тогда как сбой мультитенантного приложения затронет всех клиентов.

Проверка подлинности и авторизация. При выделенной модели данные механизмы защиты информации реализуются без дополнительных ограничений.

Мониторинг. Мониторинг одного развертывания приложения проще, чем мониторинг нескольких развертываний.

В перспективе, при существенном наращивании мощностей сервиса Waterificator, целесообразней использовать мультитенантную модель, однако при малых количествах пользователей достаточно выделенной архитектуры.

Структура и принцип работы сервиса

Для пользователя сервис представляется прозрачным. Все действия и команды происходят через главный узел, пользователь видит только один интерфейс (одну машину) через которую происходит взаимодействие. Схематически структура сервиса изображена на рис. 2.

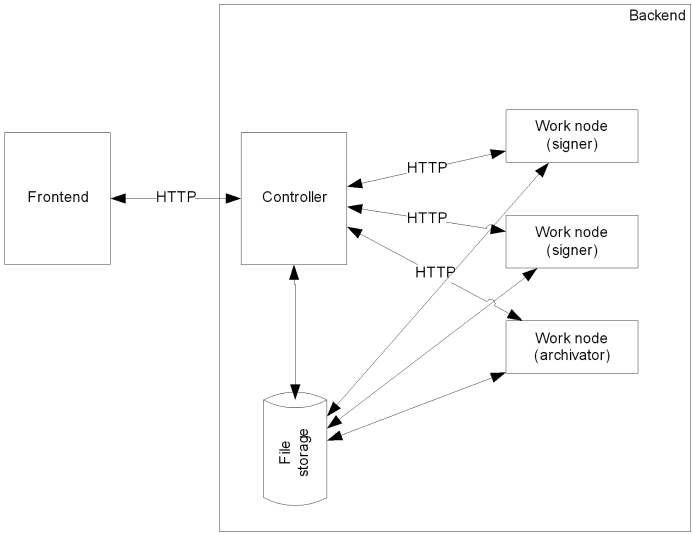

Рис. 2 – Схематически структура сервиса Waterificator

Программное обеспечение представляет собой онлайн сервис, написанный на языке программирования высокого уровня C#.

Схема взаимодействия классов изображена на рис. 3.

На рис. 4 представлен пример работы сервиса (на изображение наложен видимый пользователю водяной знак).

Анализ результатов работы сервиса

Для демонстрации результатов работы была выполнена серия тестов. В качестве рабочих узлов выступали 3 ПК с двухядерными

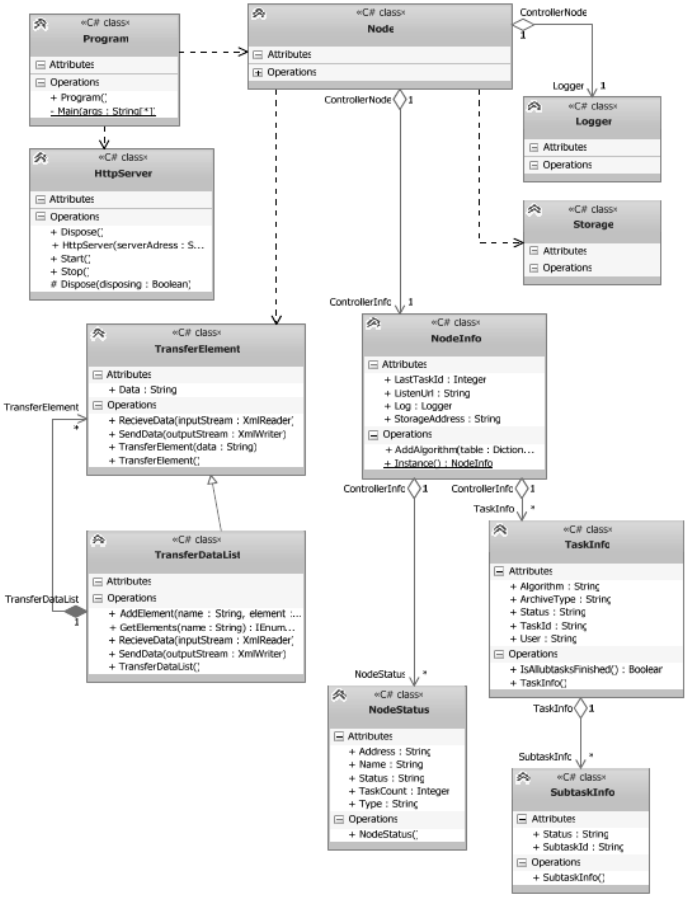

Рис. 3 – Схема взаимодействия классов на языке UML

процессорами Intel I5 с тактовой частотой 2.4 ГГц. В качестве входных данных были взяты 4 файла размером приблизительно 0.5 МБ, 1 МБ, 2 МБ, 4 МБ. Данные приведены в таблице 1.

На рис. 5 изображены графики зависимости времени работы от количества обрабатываемых файлов. Статистика собрана для файлов разных размеров и для разного количества вычислительных узлов.

Как видно из графиков, размер файлов, практически не влияет на характер кривизны изменения времени работы. Т.е. для всех видов файлов общий вид характеристик почти одинаков. Также видно, что кривые зависимости времени работы для 2-х и 3-х узлов

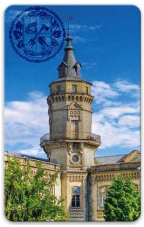

Рис. 4 – Пример работы сервиса

Таблица 1

Сводная таблица результатов работы сервиса

Кол-во узлов	Кол-во файлов	Время, с			
1	1	20,791	38,175	75,123	141,056
	2	48,912	92,984	175,381	335,762
	4	143,008	280,063	552,145	1082,472
	8	202,402	402,408	794,806	1584,612
	16	423,016	840,230	1664,060	3145,812
2	1	20,780	38,140	75,140	140,256
	2	22,212	40,514	78,058	144,753
	4	51,065	95,789	179,021	340,641
	8	108,843	212,476	405,884	801,151
	16	225,447	441,566	875,457	1750,474
3	1	20,157	38,204	75,099	141,125
	2	22,158	39,874	78,124	144,340
	4	50,876	94,315	176,206	337,452
	8	99,012	204,527	399,457	794,365
	16	218,705	437,579	863,012	1737,648
		0.5 МБ	1 МБ	2 МБ	4 МБ

почти не отличаются, что связано с неоптимальным распределением задач и обусловлено тем, что значительная часть времени уходит на упаковку файлов и распределение задач, а также с тем, что узлы получали и отправляли фоновый трафик и обрабатывали фоновые процессы. Почти прямая форма кривой для 2-х узлов говорит о хорошем распараллеливании (время на пересылку данных между узлами мало по сравнению с полезной работой узлов).

Выводы

При проектировании программного обеспечения была выбрана выделенная модель архитектуры сервиса. Это решение обусловлено простотой реализации и поддержки сервиса и гарантирует сохранность данных одних пользователей от других. Были выполнены тесты производительности при разных значениях входных параметров и на разном количестве вычислительных узлов. Ре-

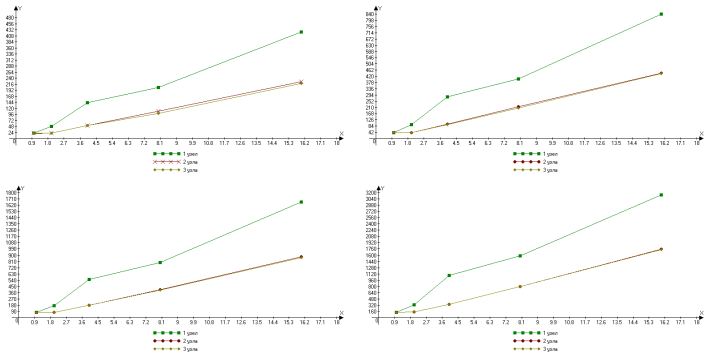

Рис. 5 – Графики времени работы сервиса для файлов размером 0.5МБ, 1МБ, 2МБ и 4МБ

зультаты моделирования показывают, что при увеличении количества узлов необходимо выполнять оптимизацию распределения задач на вычислительные узлы, что может быть исследовано в следующих публикациях.

Последняя версия программы находится по адресу https://github.com/deswars/waterificator/

Список использованных источников

1. *Аграновский А.В., Девянин П.Н., Хади Р.А.* Основы компьютерной стеганографии. – М.: Радио и связь, 2003
2. *Грибунин В.Г., Оков И.Н., Туринцев И.В.* Цифровая стеганография. – М.: СОЛОН-Пресс, 2002.
3. Мультитенантная архитектура для SaaS приложений / Блог компании Microsoft. – Режим доступа: http://habrahabr.ru/company/microsoft/blog/145027/ – Дата доступа : 01.02.2014 – Название с экрана.
4. *Eggers J., Girod B.* Informed Watermarking. Boston, MA: Kluwer, 2002.
5. *Johnson N.F., Duric Z., Jajodia S.* Information Hiding. Steganography and Watermarking – Attacks and Countermeasures. Boston, MA: Kluwer, 2001.
6. *Katzenbeisser S., Petitcolas F.A.* Information Hiding Techniques for Steganography and Digital Watermarking. Norwood, MA: Artech House, 2000.
7. *Furth B., Kirovski D.*, “Multimedia Encryption and Authentications: Techniques and Applications” – USA: Auerbach Publications., 2006.

Отримано 15.10.2015 р.