

УДК 004.4

Е.Г. Толстолужская, Б.В. Паршенцев

*Харьковский национальный университет имени В.Н. Каразина, Харьков*

## ИССЛЕДОВАНИЕ ВОЗМОЖНОСТИ ПАРАЛЛЕЛЬНОЙ ОБРАБОТКИ БОЛЬШИХ ОБЪЕМОВ ДАННЫХ НА ОСНОВЕ "CLOUD COMPUTING"

*Рассматривается задача параллельной обработки данных с помощью "Cloud computing". Для реализации данной задачи создаётся веб сервис, а для хранения данных используется субд MySQL. Параллельная обработка реализуется с помощью кластера Amazon. В данной работе выполнен анализ и создание базы данных с универсальной моделью данных, проведено исследование существующих решений в области "Cloud computing" для обработки этой базы. Представлены формальные модели, используемые в базе данных. Предложенный подход позволяет ускорить обработку больших объёмов данных.*

**Ключевые слова:** анализ, администрирование, базы данных, "Cloud computing".

### Введение

Современный мир можно с уверенностью назвать компьютеризированным, ведь незаменимость ЭВМ в нашем обществе понимает абсолютно каждый. С каждым днем мы наблюдаем рост нашей денежной единицы – информации, который несет за собой потребность правильной структуризации, хранения и ее обработки, одним словом, организации управления. За последние пять лет, несомненно, наблюдается тенденция к усложнению структур данных. Простые виды информации, представленной в форме чисел и символьных строк, не утратив своей значимости, дополняются сегодня многочисленными мультимедийными "документами", графическими образами, временными рядами, процедурными или активными данными [1, 2].

Актуальность проблемы обработки информации связана с резким ростом объёмов самих данных. Несколько лет назад объем создаваемых во всем мире данных превысил 1 зеттабайт – это примерно миллиард целиком заполненных жестких дисков емкостью 1 Тбайт, и уже превышает все доступное на сегодняшний день пространство хранения. Основные причины лавинообразного роста объёмов информации:

1) создавать новую информацию сейчас намного дешевле, чем раньше: стоимость хранения и обработки снизилась в 6 раз с 2005 года;

2) бюджеты на ИТ за то же время увеличились в полтора раза;

3) к 2020 году в 8 раз возрастет количество устройств, которые создают информацию: начиная от смартфонов и камер с более высоким разрешением и заканчивая всевозможными датчиками и умными персональными устройствами дополнительная информация создается в виде производной от уже созданной — в первую очередь, это бэкапы, а также логи, архивы цифрового аудио, видео [3, 4].

В свою очередь, недостаток пространства для хранения объясняется тем, что аппаратные СХД долгое время эволюционировали по принципу быстрее, выше, сильнее – то есть от ленты к дискам большей емкости, более быстрым дискам, накопителям на флеш-памяти, системам из нескольких полок с накопителями разного типа и скорости. А оптимизация СХД «заточивалась» под нужды компаний с большими бюджетами – быстрый сторадж для виртуализации, супер-быстрый сторадж для обработки данных в реальном времени, умный сторадж с оптимизацией под конкретные бизнес-приложения. В то же время про бэкапы, архивы и логи, которые не создают напрямую стоимость для бизнеса и просто занимают место, заказчики как бы забыли, а производители СХД не подумали. Облачная обработка данных (англ. Cloud computing) – это технология обработки данных, в которой программное обеспечение предоставляется пользователю как интернет-сервис. Пользователь имеет доступ к собственным данным, но не может управлять операционной системой и собственно ПО, с которым работает (заботиться об инфраструктуре ему также не нужно). Непосредственно «облаком» называют интернет, который как раз и скрывает многие технические детали.

В основе Cloud Computing лежат несколько подходов. Первый – доступность через интернет. Конечно, бывают и закрытые системы, но, как правило, все можно потрогать через сети (при этом, «наружу» облако предоставляет себя как обычный сервер). Второй важный момент – это виртуализация. Благодаря виртуализации, пользователи получают столько ресурсов, сколько им надо (и, разумеется, сколько могут позволить себе приобрести). Что для этого требуется со стороны сервера и каким образом он может выделить такие ресурсы – все скрыто за стенами виртуальных машин; они могут работать на сотнях и даже тысячах серверов, а зачастую – еще и в разных дата-центрах. Третий мо-

мент: Cloud Computing – это услуга. В Cloud Computing используется схожий подход. Все, что касается Cloud Computing, обычно принято называть словом aaS. Расшифровывается просто – «as a Service», то есть «как сервис», или «в виде сервиса».

SaaS (Software-aaS), или приложения в виде сервисов – вариант, при котором тебе предлагают использовать какое-то конкретное ПО, например, корпоративные системы, в виде сервиса по подписке. Скажем, у предприятия нет возможности или желания хостить внутренний Exchange-сервер для работы почты, календарей и т.п. – и оно может купить его удаленно, с учетом всей необходимой специфики. Часто ли такие сервисы доступны просто в браузере? Пример – Google Docs.

PaaS (Platform-aaS) – в отличие от SaaS, предназначенного больше для конечного пользователя, вариант для разработчиков. В облаке функционирует некоторый набор программ, основных сервисов и библиотек, на основе которых предлагается разрабатывать свои приложения. Самый яркий пример – платформа для создания приложений Google AppEngine. Помимо этого, под PaaS понимают также и отдельные части сложных систем, вроде системы базы данных или коммуникаций.

IaaS (Hardware-aaS) – один из первых терминов, означающих предоставление некоторых базовых «железных» функций и ресурсов в виде сервисов. Но вместо прямой аренды хостинга используется виртуализация. Поэтому, когда речь идет о конкретном железе, понимаются некоторые абстрактные сущности, аналогичные реальному железу (место под хранение, процессорное время в эквиваленте какого-либо реального CPU, пропускная способность).

IaaS (Infrastructure-aaS) – считается, что термин пришел на смену IaaS, подняв его на новый уровень. Для примера – это системы виртуализации, балансировщики нагрузки и тому подобные системы, лежащие в основе построения других систем.

CaaS (Communication-aaS) – подразумевается, что в качестве сервисов предоставляются услуги связи; обычно это IP-телефония, почта и мгновенные коммуникации (чаты, IM).

**Целью данной работы** является создание веб-сервиса для обработки и структуризации данных, которые хранятся в базе с универсальной моделью данных с помощью Cloud computing.

## Разработка базы данных для данного сервиса

Для достижения поставленной задачи – создание веб-сервиса для обработки данных, была разработана база данных. Предметная область базы данных состоит из археологических и архитектурных памятников Черняховской культуры. В базе данных

представлены памятники данной культуры с подробными их описаниями и документами, которые так или иначе связаны с памятниками. В качестве СУБД была использована MySQL. Данная СУБД была выбрана благодаря её характеристикам. Разработку и поддержку MySQL осуществляет корпорация Oracle [5 – 7].

Внутренние характеристики и переносимость. Данная СУБД написана на C и C++ и обладает следующими характеристиками:

1. Протестирована на широком спектре различных компиляторов.
2. Работает на множестве различных платформ.
3. Для обеспечения переносимости использует инструменты GNU – Automake, Autoconf и Libtool.
4. Доступны API-интерфейсы для C, C++, Eiffel, Java, Perl, PHP, Python, Ruby и Tel.
5. Полностью многопоточна с использованием потоков ядра. Может работать в многопроцессорных системах.
6. Обеспечивает транзакционный и нетранзакционный механизмы хранения.
7. Использует очень быстрые дисковые таблицы (MyISAM) со сжатием индексов на основе бинарных деревьев (B-деревьев).
8. Сравнительно простое добавление другого механизма хранения. Это удобно, если требуется добавить SQL-интерфейс к базе данных собственной разработки.
9. Очень быстрая система распределения памяти, основанная на потоках.
10. Очень быстрые соединения, использующие оптимизированные однопроходные мультисоединения.
11. Хранимые в памяти хеш-таблицы, которые используются в качестве временных таблиц.
12. Функции SQL реализованы с использованием высоко оптимизированной библиотеки классов и должны выполняться предельно быстро. Как правило, какого-либо распределения памяти после инициализации запроса не выполняется.

Разработанная авторами база состоит из пяти таблиц: Monuments (памятники) (рис. 1), Author (автор) (рис. 2), Documents (документы) (рис. 3), Articles (статьи) (рис. 4), User (пользователь) (рис. 5). Таблица Monuments описывает памятники архитектуры и их топографическое расположение (рис. 6).

Таблица Author описывает автора находки, год находки и как именно был обнаружен памятник.

Таблица Document описывает документы, принадлежащие памятнику.

Таблица Articles описывает статьи, которые написаны о данном памятнике.

Таблица User описывает пользователей и их права доступа.

Данная таблица позволяет администратору управлять всей базой.

```
CREATE TABLE IF NOT EXISTS
ostrogothia.Monuments(
idMonument INT NOT NULL
AUTO_INCREMENT,
Name VARCHAR(45) NOT NULL,
Type_Monument VARCHAR(45) NOT
NULL,
Nature_research VARCHAR(45) NOT NULL,
Finding VARCHAR(100) NOT NULL,
NI FLOAT NOT NULL,
EI FLOAT NOT NULL,
Hydraulic VARCHAR(200) NOT NULL,
River_order INT NOT NULL,
Topographical VARCHAR(100) NOT NULL,
Character_studies VARCHAR(45),
Region VARCHAR(100) NOT NULL,
AREA VARCHAR(100) NOT NULL,
PRIMARY KEY (idMonument),
ENGINE = InnoDB;
```

Рис. 1. Создание таблицы памятников

```
CREATE TABLE IF NOT EXISTS
ostrogothia.Authors(
idAuthors INT NOT NULL
AUTO_INCREMENT,
NameAuthor VARCHAR(45) NOT NULL,
Action VARCHAR(45) NOT NULL,
Years INT NOT NULL,
Monument BIGINT NOT NULL,
FOREIGN KEY (Monument) REFERENCES
Monuments (idMonument),
PRIMARY KEY (idAuthors) )
ENGINE = InnoDB;
```

Рис. 2. Создание таблицы авторов

```
CREATE TABLE IF NOT EXISTS
ostrogothia.Documents(
idDocuments INT NOT NULL
AUTO_INCREMENT,
Type_File VARCHAR(45) NOT NULL,
Path_File VARCHAR(45) NOT NULL,
Monument BIGINT NOT NULL,
FOREIGN KEY (Monument) REFERENCES
Monuments (idMonument),
PRIMARY KEY (idDocuments) )
ENGINE = InnoDB;
```

Рис. 3. Создание таблицы документов

```
CREATE TABLE IF NOT EXISTS
ostrogothia.Articles(
idArticle INT NOT NULL
AUTO_INCREMENT,
Name VARCHAR(45) NOT NULL,
Years INT NOT NULL,
NameAuthor VARCHAR(45) NOT NULL,
Monument BIGINT NOT NULL,
FOREIGN KEY (Monument) REFERENCES
Monuments (idMonument),
PRIMARY KEY (idArticle) )
ENGINE = InnoDB;
```

Рис. 4. Создание таблицы статей

```
CREATE TABLE IF NOT EXISTS
ostrogothia.Users(
idUser INT NOT NULL
AUTO_INCREMENT,
Name VARCHAR(45) NOT NULL,
Pass VARCHAR(45) NOT NULL,
Email VARCHAR(45) NOT NULL,
PRIMARY KEY (idUser) )
ENGINE = InnoDB;
```

Рис. 5. Создание таблицы пользователей

```
CREATE INDEX nameOfMonument ON
Monuments (Name);
```

Рис. 6. Индексация памятников

В данной базе данных использована индексация по имени памятника. Это позволяет ускорить поиск памятников по их имени.

Также для ускорения поиска авторов, документов и статей аналогичным образом добавлена индексация по имени авторов, названием документов и название статей.

### Анализ и выбор “облачных” технологии

В качестве облачных технологий была выбрана heroku. Heroku была выбрана из-за поддержки различных языков и фреймворков (в данном случае выбран язык Ruby, а фреймворк Rails), лёгкая расширяемость, автомасштабируемость, высокая скорость, одинаковое развёртывание тестовых и рабочих окружений.

Heroku – paas, Платформа как услуга (PaaS, Platform-as-a-Service) – модель, когда потребителю

предоставляется возможность использования облачной инфраструктуры для размещения базового программного обеспечения для последующего размещения на нём новых или существующих приложений (собственных, разработанных на заказ или приобретённых тиражируемых приложений). В состав таких платформ входят инструментальные средства создания, тестирования и выполнения прикладного программного обеспечения – системы управления базами данных, связующее программное обеспечение, среды исполнения языков программирования – предоставляемые облачным провайдером. Контроль и управление основной физической и виртуальной инфраструктурой облака, в том числе сети, серверов, операционных систем, хранения осуществляется облачным провайдером, за исключением разработанных или установленных приложений, а также, по возможности, параметров конфигурации среды (платформы).

На серверах Heroku используются операционные системы Debian или Ubuntu (которая также основана на Debian).

Для реализации возможности параллельной обработки информации был выбран Амазон Cluster GPU instance – HPC инстанс с двумя NVIDIA Tesla “Fermi” M2050 GPU. Спецификация инстанса следующая:

- 22 GB of memory;
- 33.5 EC2 Compute Units (2 x Intel Xeon X5570, quad-core “Nehalem” architecture);
- 2 x NVIDIA Tesla “Fermi” M2050 GPUs;
- 1690 GB of instance storage;
- 64-bit platform;
- I/O Performance: Very High (10 Gigabit Ethernet);
- API name: cg1.4xlarge.

## Выводы

Экспериментально было доказано преимущество использования облачных технологий для обработки больших объёмов данных на примере созданного веб-сервиса.

Последующие исследования целесообразно проводить в направлении ускорения обработки и возможности оптимального времени обработки данных при увеличении самих объёмов данных.

## Список литературы

1. Пелецишин А.М. *Позиціонування сайтів у глобальному інформаційному середовищі* / А.М. Пелецишин. – Львів: Вид-во Національного університету “Львівська політехніка”, 2007. – 258 с.
2. G. David Garson *Cluster Analysis: 2014 Edition (Statistical Associates Blue Book Series 24)* – Kindle Edition, 2014. – 232 p.
3. Alex Vrenios *Linux Cluster Architecture*. – Paperback, 2002. – 254 p.
4. Kevin Faustino *The Rails 4 Way (3rd Edition) (Addison-Wesley Professional Ruby Series)* – Kindle Edition, 2014. – 684 p.
5. Аткинсон Леон. *MySQL. Библиотека профессионала* / Леон Аткинсон. – М.: Вильямс, 2008. – 624 с.
6. Грофф Джеймс. *SQL: полное руководство* / Джеймс Грофф, Пол Вайнберг. – К.: BHV, 2005. – 608 с.
7. Поль Дюбуа. *MySQL, 3-е издание = MySQL, 3ed.* / Поль Дюбуа. – М.: Вильямс, 2006. – 1168 с.

Поступила в редколлегию 15.04.2015

**Рецензент:** д-р техн. наук проф. Г.А. Поляков, Белгородский государственный национальный исследовательский университет, Белгород.

## ДОСЛІДЖЕННЯ МОЖЛИВОСТІ ПАРАЛЕЛЬНОЇ ОБРОБКИ ВЕЛИКИХ ОБ'ЄМІВ ДАНИХ НА ОСНОВІ "CLOUD COMPUTING"

О.Г. Толстолузька, Б.В. Паршенцев

*Розглядається завдання паралельної обробки даних за допомогою "Cloud computing". Для реалізації даного завдання створюється веб-сервер сервіс, а для зберігання даних використовується субд MYSQL. Паралельна обробка реалізується за допомогою кластера Амазон. У даній роботі виконаний аналіз і створення бази даних з універсальною моделлю даних, проведено дослідження існуючих рішень в області "Cloud computing" для обробки цієї бази. Представлені формальні моделі, використовувані в базі даних. Запропонований підхід дозволяє прискорити обробку великих об'ємів даних.*

**Ключові слова:** аналіз, адміністрування, бази даних, "Cloud computing".

## STUDY THE POSSIBILITY OF PARALLEL PROCESSING OF LARGE VOLUMES OF DATA BASED ON "CLOUD COMPUTING"

O.G. Tolstoluzhskaya, B.V. Parshencev

*The problem of parallel processing with the help of "Cloud computing". To implement this task, created a web service, and is used for data storage database MySQL. Parallel processing is realized by using a cluster Amazon. In this paper, the analysis and creation of a database with a universal data model, a study of existing solutions in the field of "Cloud computing" to handle this database. A formal model used in the database. The proposed approach allows to speed up the processing of large amounts of data.*

**Keywords:** analysis, administration, database, "Cloud computing".