

Введение

Курс «Базы данных»

Цесько Вадим Александрович
<http://incubos.org>
@incubos

Computer Science Center

16 сентября 2013 г.

Содержание

- 1 О лекторе
- 2 Организация курса
- 3 Содержание курса
- 4 Определения и примеры
- 5 Классификация БД
- 6 Домашнее задание

О лекторе

- 5 лет в Digitek Labs¹
 - Параллельные системы цифровой обработки сигналов (гидроакустика)
 - Статический анализ кода программных систем (C, JavaScript)
- 2.5 года в Яндекс
 - Распределённые высоконагруженные системы: Яндекс.Авто, Яндекс.Недвижимость и др.
- 3 года преподавания² в СПбГПУ
 - Базы данных
 - Методы анализа и обеспечения качества ПО
 - Параллельные вычисления

¹<http://digiteklabs.ru>

²<http://kspt.ftk.spbstu.ru/people/tsesko/>

Формат

- Каждый понедельник с 20:00 до 21:30 лекция, с 21:30 до 22:00 ответы на вопросы
- 14-16 лекций, в т. ч. приглашённые докладчики во второй половине курса

Курсовая работа

- Критерий получения зачёта
- Своё хранилище данных
- New feature request — каждые 2 недели, лимит на реализацию — 2 недели
- Проект на <http://bitbucket.org> (Hg) / <http://github.com> (Git) + Issue Tracking
- Scala/Java + ScalaTest/JUnit + Maven/SBT
- README + INSTALL
- В конце — отчёт (2-10 стр. с картинками)
- Команда 1-3 человека

Материалы

- Курс «Базы данных» предыдущего года³
- Ссылки на книги, статьи и блоги в каждой лекции — для **самостоятельного** изучения

³Илья Тетерин, <http://fluffypulser.ru/static/dbcourse.2012/>

Цели

- Принципы построения систем хранения (и обработки) данных
- Проблемы и варианты их решения
- Промышленные примеры реализации принципов
- Навыки оценки решений

О чём мы будем говорить

- Классификации баз данных
- Big Data: Scalability, Availability, CAP
- SQL vs NoSQL
- Транзакции
- Производительность
- Примеры: Redis, Cassandra, MongoDB, HBase, etc.
- Проектирование систем хранения и обработки

О чём мы (почти) НЕ будем говорить

- Реляционная модель данных
- Теория нормальных форм
- SQL

О чём мы возможно будем говорить

- Примеры решения конкретных задач:
 - Хранение картинок
 - Полнотекстовый поиск
- Встраиваемые БД
- Графовые БД

База данных

A database

is an organized collection of data^a.

^a<http://en.wikipedia.org/wiki/Database>

- Телефонная книга
- Книжная библиотека
- Коллекция фотографий
- Товарный чек
- Социальная сеть
- Файловая система
- Географическая карта
- ...

Операции над данными

CRUD

Create, read, update, delete^a.

^ahttp:

[//en.wikipedia.org/wiki/Create,_read,_update_and_delete](http://en.wikipedia.org/wiki/Create,_read,_update_and_delete)

Например

Operation	SQL	HTTP
Create	INSERT	POST
Read (Retrieve)	SELECT	GET
Update (Modify)	UPDATE	PUT / PATCH
Delete (Destroy)	DELETE	DELETE

Бумажная телефонная книга

- Create
 - Открыть страницу по первой букве фамилии
 - Вписать в конец страницы ФИО и телефон
- Read
 - Открыть страницу по первой букве фамилии
 - Просматривать записи по порядку до совпадения
- Update
 - Открыть страницу по первой букве фамилии
 - Просматривать записи по порядку до совпадения
 - Зачеркнуть ФИО и телефон
 - Вписать в конец страницы ФИО и телефон
- Delete
 - Открыть страницу по первой букве фамилии
 - Просматривать записи по порядку до совпадения
 - Зачеркнуть ФИО и телефон

Про «скорость» телефонной книги

- Create — «быстрая» вставка
- Read, Update, Delete — всё сводится к поиску записи

Инвариант

В телефонной книге есть 0 или 1 незачёркнутая запись для любой ФИО

Если хотим оценить точнее, то нужно кое-что вспомнить...

$O(n)$ и $o(n)$

Формальное определение^a

^ahttp://en.wikipedia.org/wiki/Big_O_notation

$$f(n) \in O(g(n)) \Leftrightarrow \exists k > 0 \exists n_0 \forall n > n_0 f(n) \leq g(n) \cdot k$$
$$f(n) \in o(g(n)) \Leftrightarrow \forall k > 0 \exists n_0 \forall n > n_0 f(n) \leq g(n) \cdot k$$

$o(n)$: Сравнение

Компьютер мощностью 1 Gops:

$o()$	1 Gops	1 Tops
$o(n)$	1 секунда	16 минут
$o(\log_2(n))$	30 микросекунд	40 микросекунд
$o(n^2)$	32 года	32 миллиона лет
$o(n \cdot \log_2(n))$	30 секунд	11 часов
$o(k \cdot n)$	k с.	$1000k$ с.

$o(n)$: Примеры

- $o(n)$ — последовательный перебор
- $o(n^2)$ — сравнение всех со всеми
- $o(\log_2(n))$ — бинарный поиск
- $o(n \cdot \log_2(n))$ — сортировка Quick Sort
- $o(k \cdot n)$ — сортировка Radix Sort

Вернёмся к телефонной книге

Пусть в книге 100 записей.

- Если совсем не сортировать, то поиск за
 - $100/2 = 50$ операций в среднем
- Если всё полностью отсортировано, то поиск за
 - $\log_2(100) \approx 6.64$ операции
- А в бумажной телефонной книге
 - $\log_2(28) + 100/28/2 \approx 6.59$ операции

Ещё есть ограничения физического мира

1	L1 cache reference	0.5 ns	
2	Branch mispredict	5 ns	
3	L2 cache reference	7 ns	
4	Mutex lock/unlock	25 ns	
5	Main memory reference	100 ns	
6	Compress 1K bytes with Zippy	3,000 ns	= 3 us
7	Send 2K bytes over 1 Gbps network	20,000 ns	= 20 us
8	SSD random read	150,000 ns	= 150 us
9	Read 1 MB sequentially from memory	250,000 ns	= 250 us
10	Round trip within same datacenter	500,000 ns	= 0.5 ms
11	Read 1 MB sequentially from SSD*	1,000,000 ns	= 1 ms
12	Disk seek	10,000,000 ns	= 10 ms
13	Read 1 MB sequentially from disk	20,000,000 ns	= 20 ms
14	Send packet CA->Netherlands->CA	150,000,000 ns	= 150 ms

45

⁴<https://gist.github.com/jboner/2841832>

⁵<http://elizarov.livejournal.com/27425.html>

Disclaimer

- Общепризнанных устоявшихся классификаций нет⁶
- Классификации неполны по определению
- Многие хранилища данных являются гибридными или настраиваемыми
- Любые противоречия случайны и на совести автора

⁶http://en.wikipedia.org/wiki/Database#Database_type_examples

Data Model

- Hierarchical
- Relational
- Graph
- Object
- Document
- Key-value
- Columnar
- Deductive
- Spatial
- ...

Deployment

- Embedded (in-process)
- Separate
- Distributed
- ...

Storage

- In-memory
- On-disk
- Data warehousing
- ...

Usage patterns

- Messaging
- Counters
- Logging + Batch processing
- OLTP
- Entity enrichment
- Versioning
- ...

И это ещё не всё

- Query Language
- Consistency
- Availability & Replication
- Partition tolerance
- Transactions
- Backups
- Security
- ...

План минимум

- Телефонная книга: nickname (Ключ), ФИО + телефон (Значение)
- CRUD: `add("ivanov ...)`, `get("ivanov")`, `delete("ivanov")`
- Доступ через консоль или HTTP или TCP
- В памяти на List или Map
- Персистентно на диске

План максимум: key-value storage

- IMDB dataset⁷
- Quora: Where can I get large datasets open to the public?⁸

⁷<http://www.imdb.com/interfaces>

⁸<http://www.quora.com/Data/>

Where-can-I-find-large-datasets-open-to-the-public

Подсистемы

- API
- Ключ (`string`, `byte []`, ...)
- Запись/документ (`string`, `byte []`, JSON, ...)
- Парсер аргументов
- Сериализация/десериализация (JSON, XML, ...)
- **Надёжное** хранение (`flush()`, `commit log`, `snapshot`, ...)

Вопросы?

- <http://incubos.org/contacts/>
- Общие вопросы — в Twitter: @incubos
- Вопросы по лекциям — в комментариях:
<http://incubos.org/blog/>
- Частные вопросы — в почту
vadim.tsesko@gmail.com
- Не забудьте про ДЗ