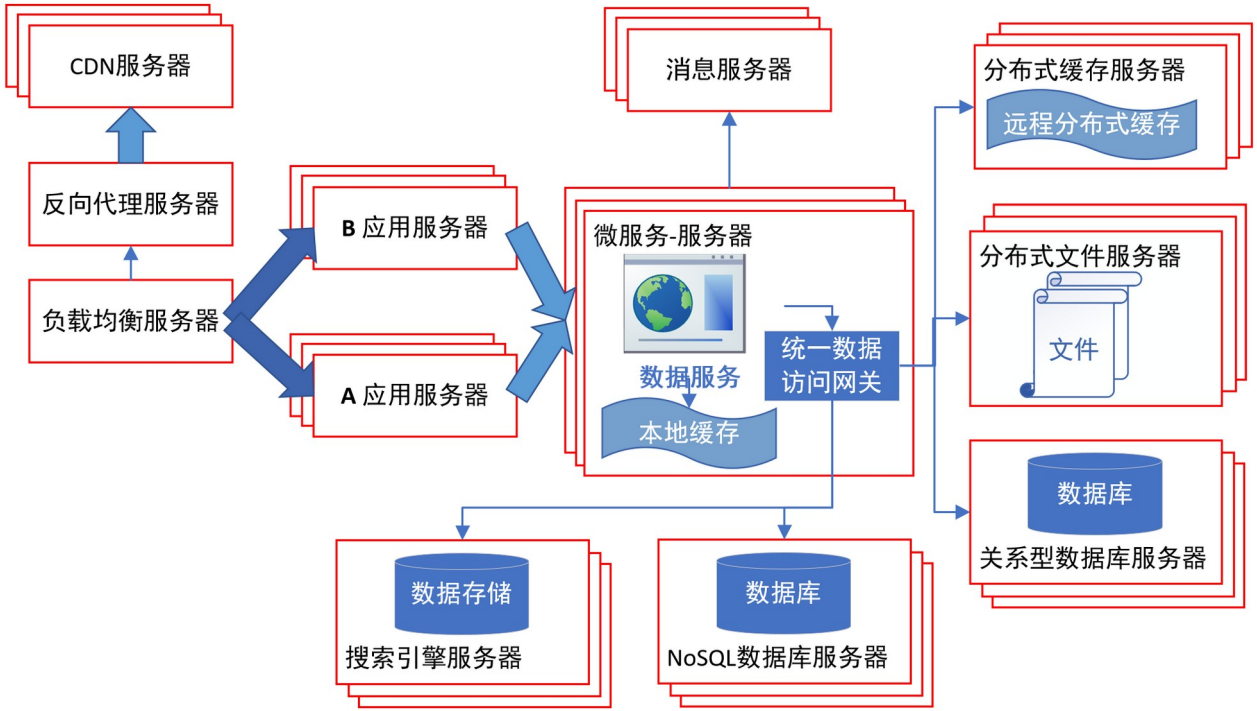
Architecture of Enterprise Applications 0 Overview

Haopeng Chen

***RE**liable, **IN**telligent and **S**calable Systems Group (**REINS**)*
Shanghai Jiao Tong University
Shanghai, China
http://reins.se.sjtu.edu.cn/~chenhp
e-mail: chen-hp@sjtu.edu.cn

- 服务器端程序开发 1 – 架构与实例池管理

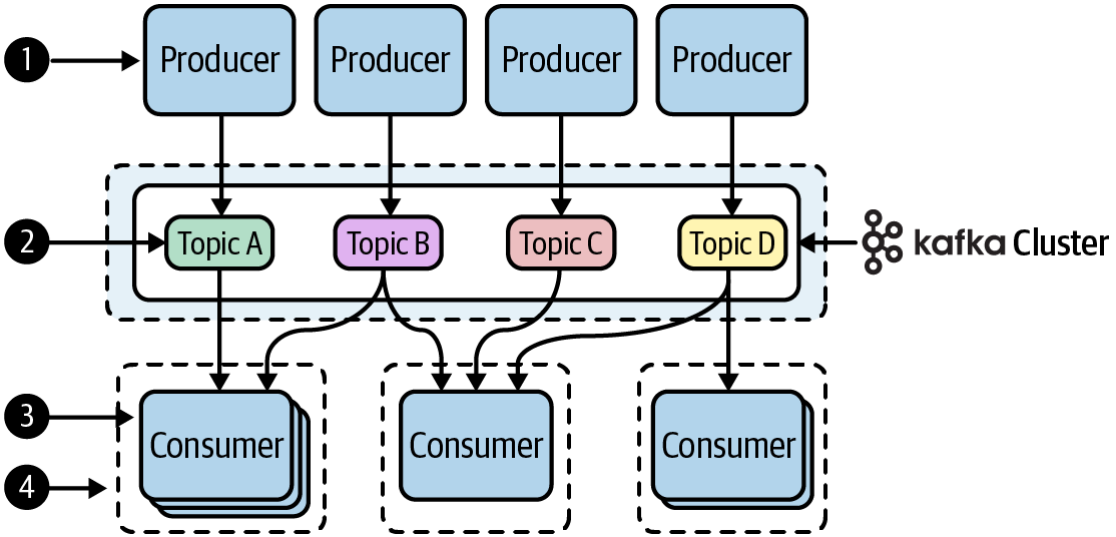


课程内容

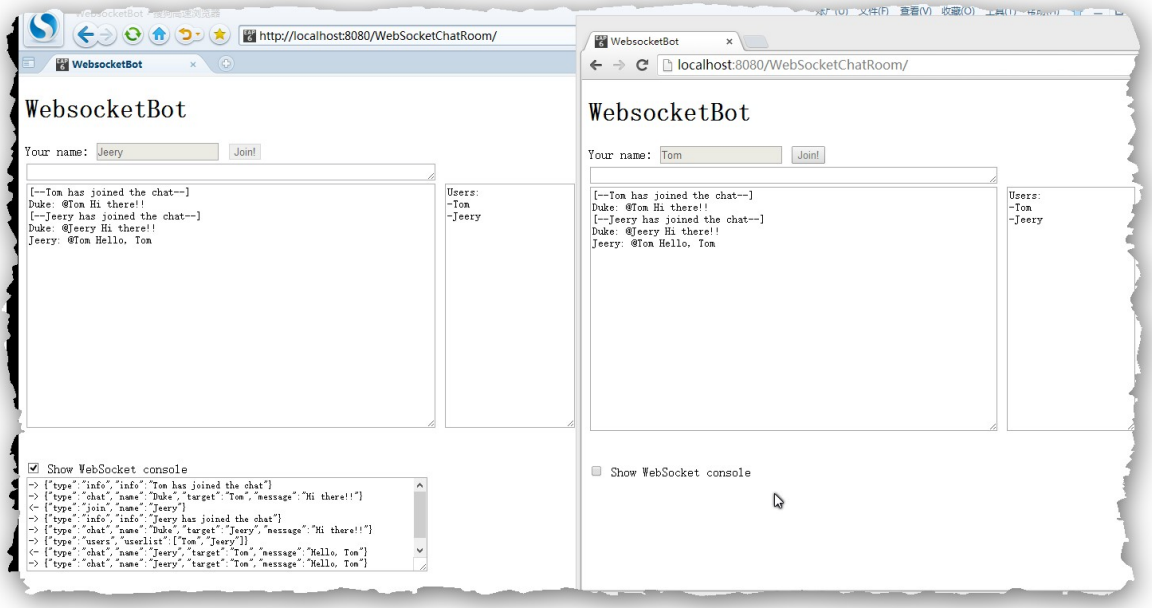
- 服务器端程序开发 2 – 异步通信 1 – JMS

课程内容

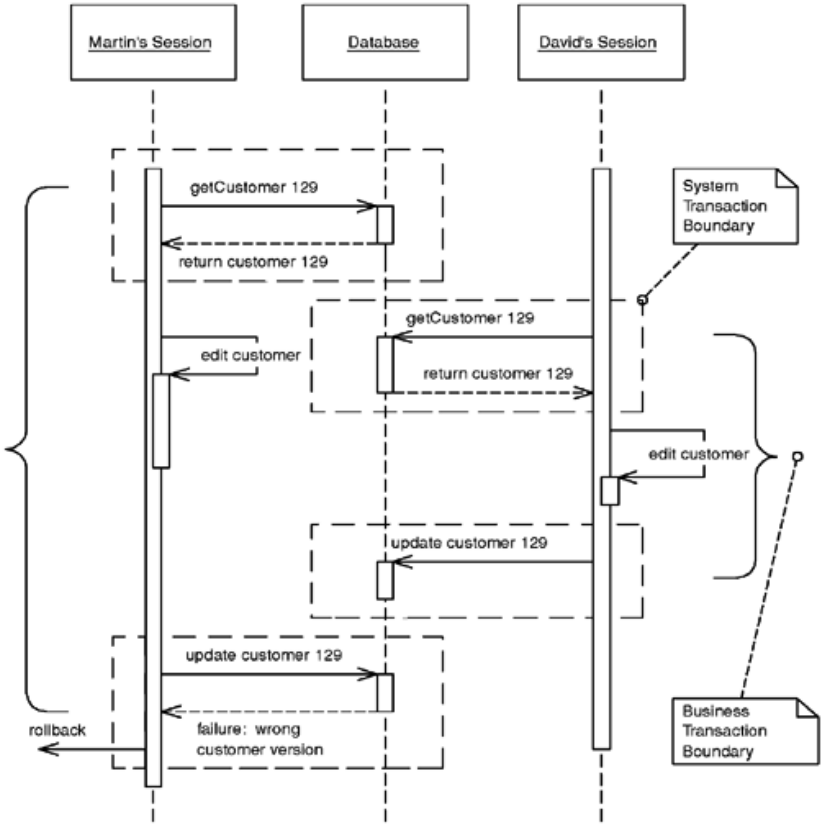
- 服务器端程序开发 2 – 异步通信 1 – Kafka 消息中间件



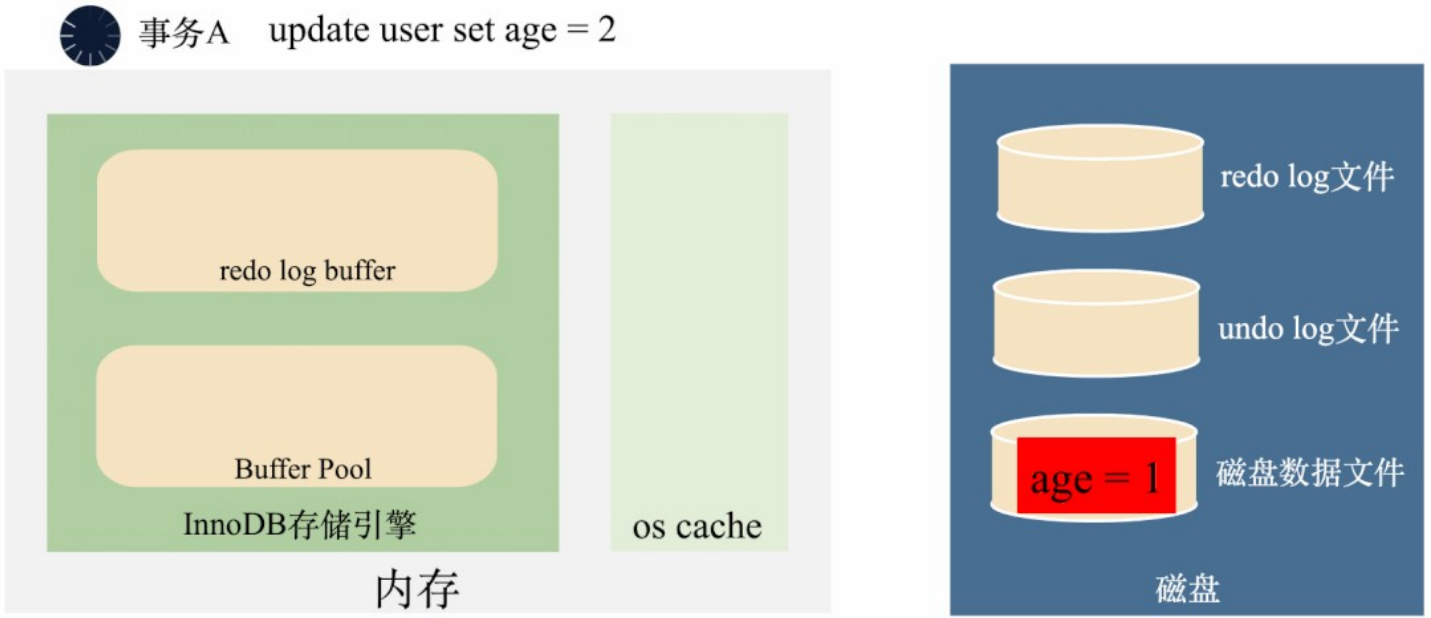
- 服务器端程序开发 3 – 异步通信 2- WebSocket



- 服务器端程序开发 4 – 事务管理

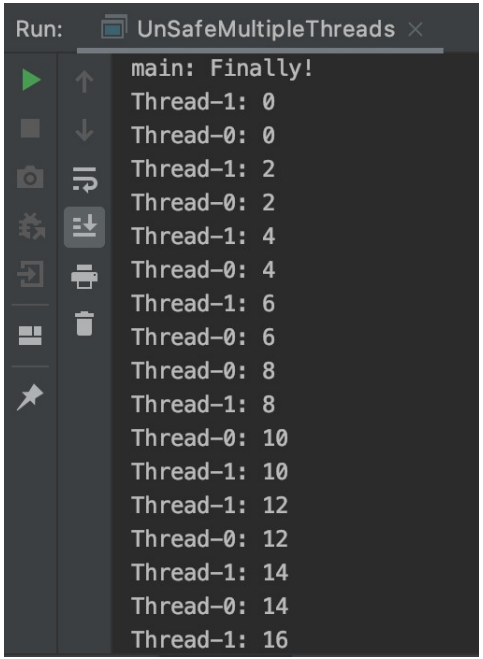


- 服务器端程序开发 5 – 数据库事务管理



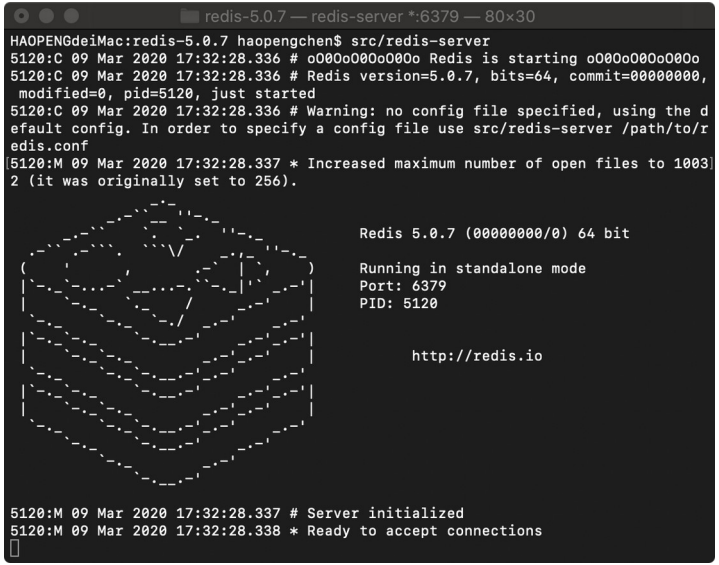
- 服务器端程序开发 6 – Java 多线程编程

```
    public static void main(String args[])
            throws InterruptedException {

        // Delay, in milliseconds before
        // we interrupt MessageLoop
        // thread (default one hour).
        long patience = 1000;// * 60 * 60;


        threadMessage("Starting MessageLoop thread");
        long startTime = System.currentTimeMillis();

        UnSafeMultipleThreads s = new UnSafeMultipleThreads();
        Thread t1 = new Thread(s.new CounterLoop());
        t1.start();
        Thread t2 = new Thread(s.new CounterLoop());
        t2.start();
        threadMessage("Waiting for MessageLoop thread to finish");
    }
}
```

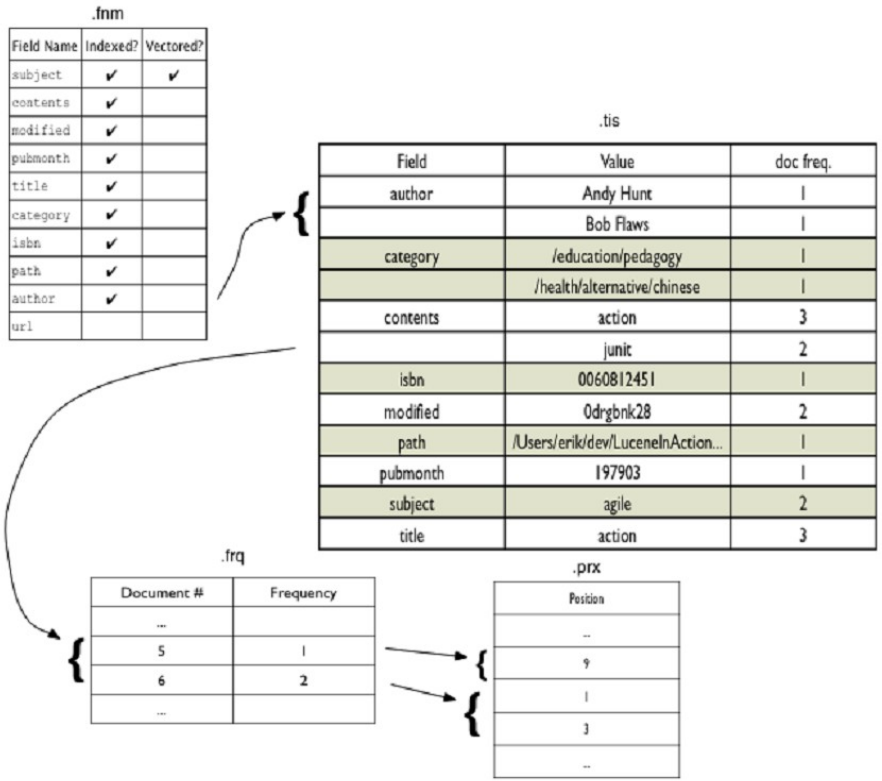


- 服务器端程序开发 7 – 分布式缓存

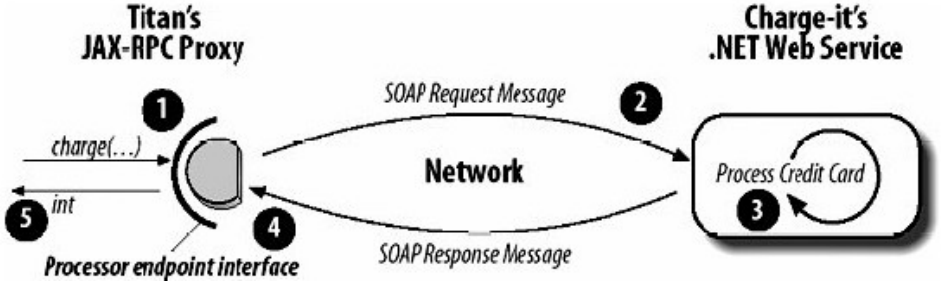
```
@Repository
public class PersonDaoImpl implements PersonDao {
    @Autowired
    private PersonRepository personRepository;
    @Autowired
    RedisUtil redisUtil;

    @Override
    public Person findOne(Integer id) {
        Person person = null;
        Object p = redisUtil.get("user" + id);
        if (p == null) {
            person = personRepository.getOne(id);
            redisUtil.set("user" + id, JSONArray.toJSON(person));
        } else {
            person = JSONArray.parseObject(p.toString(), Person.class);
        }
        return person;
    }
}
```



- 服务器端程序开发 8 – 全文搜索

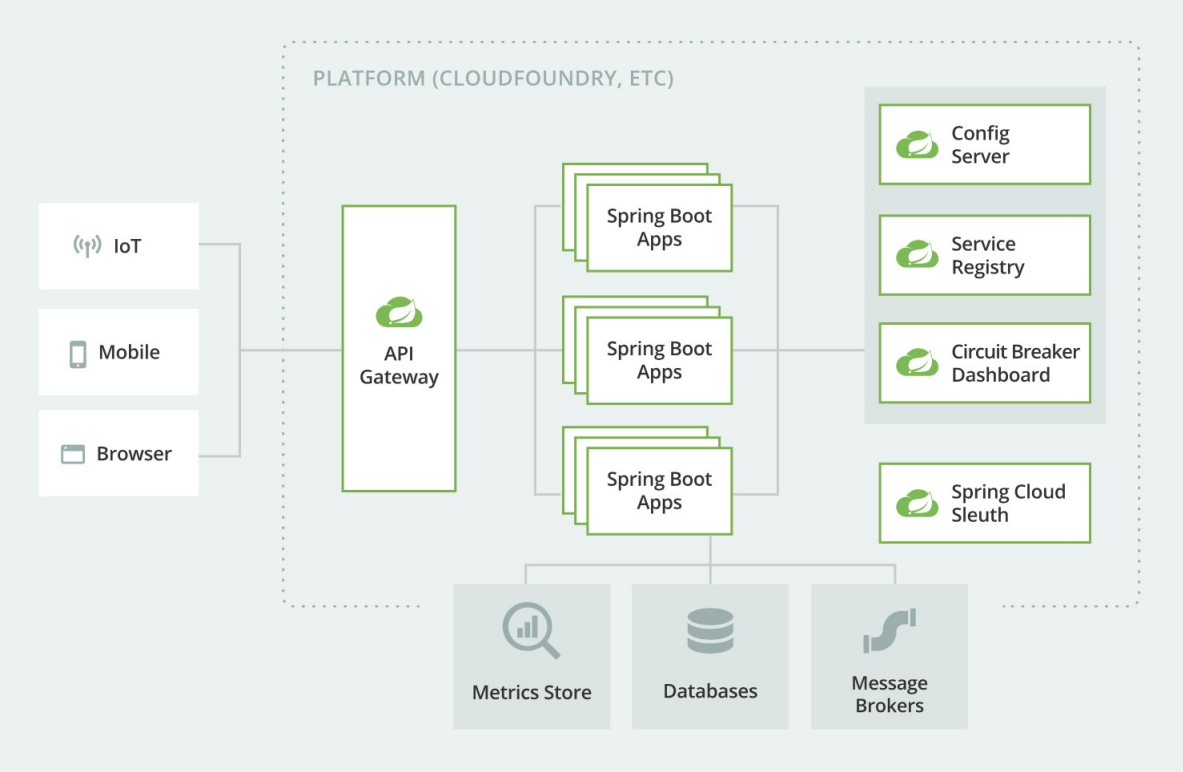


- 服务器端程序开发 9 – Web 服务

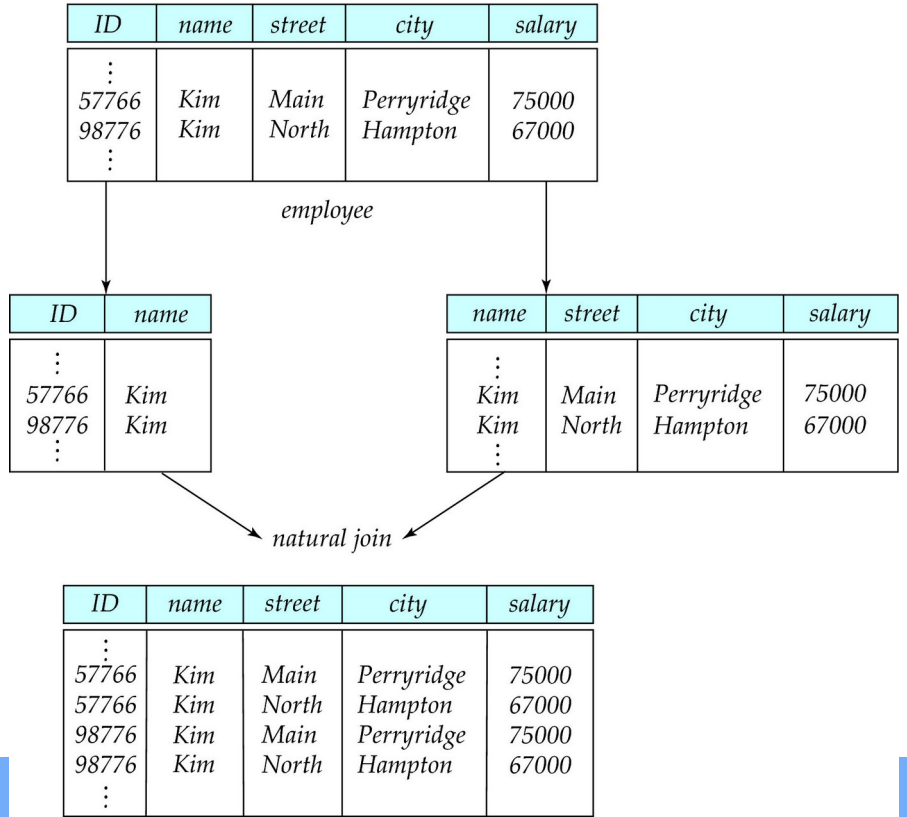



```
@RestController
public class GreetingController {

  private static final String template = "Hello, %s!";
  private final AtomicLong counter = new AtomicLong();

  @GetMapping("/greeting")
  public Greeting greeting(@RequestParam(value = "name", defaultValue = "World") String name)
{
    return new Greeting(counter.incrementAndGet(), String.format(template, name));
  }

  @PostMapping("/greeting")
  public Greeting greetingpost(@RequestParam(value = "name", defaultValue = "Spring") String
name) {
    return new Greeting(counter.incrementAndGet(), String.format(template, name));
  }
}
```

- 服务器端程序开发 10 – 微服务架构



- 数据库进阶 1 – 数据库设计



- 数据库进阶 2 – MySQL 索引与缓存管理

```
shell> mysqld --innodb-buffer-pool-size=8G --innodb-buffer-pool-instances=16
mysql> SELECT @@innodb_buffer_pool_size/1024/1024/1024;
+----------------------------------------------------------+
|      @@innodb_buffer_pool_size/1024/1024/1024 |
+----------------------------------------------------------+
|                                        8.000000000000 |
+----------------------------------------------------------+

shell> mysqld --innodb-buffer-pool-size=9G --innodb-buffer-pool-instances=16
mysql> SELECT @@innodb_buffer_pool_size/1024/1024/1024;
+----------------------------------------------------------+
|      @@innodb_buffer_pool_size/1024/1024/1024 |
+----------------------------------------------------------+
|                                       10.000000000000 |
```

```
CREATE TABLE test (
  id INT NOT NULL,
  last_name CHAR(30) NOT NULL,
  first_name CHAR(30) NOT NULL,
  PRIMARY KEY (id),
  INDEX name (last_name,first_name) );
```

- 数据库进阶 3 – MySQL 备份与恢复

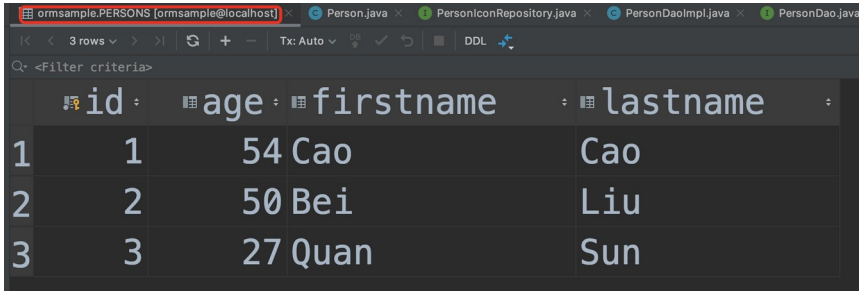
```
shell> mysqldump --single-transaction --flush-logs --master-data=2 \
       --all-databases --delete-master-logs > backup_sunday_1_PM.sql
```

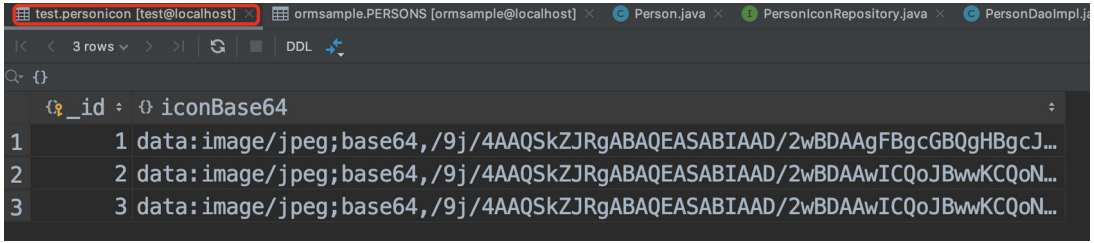
```
-rw-rw---- 1 guilhem  guilhem   1277324 Nov 10 23:59 gbichot2-bin.000001
-rw-rw---- 1 guilhem  guilhem         4 Nov 10 23:59 gbichot2-bin.000002
-rw-rw---- 1 guilhem  guilhem        79 Nov 11 11:06 gbichot2-bin.000003
-rw-rw---- 1 guilhem  guilhem       508 Nov 11 11:08 gbichot2-bin.000004
-rw-rw---- 1 guilhem  guilhem 220047446 Nov 12 16:47 gbichot2-bin.000005
-rw-rw---- 1 guilhem  guilhem    998412 Nov 14 10:08 gbichot2-bin.000006
-rw-rw---- 1 guilhem  guilhem       361 Nov 14 10:07 gbichot2-bin.index
```

- 数据库进阶 4 – MySQL 分区

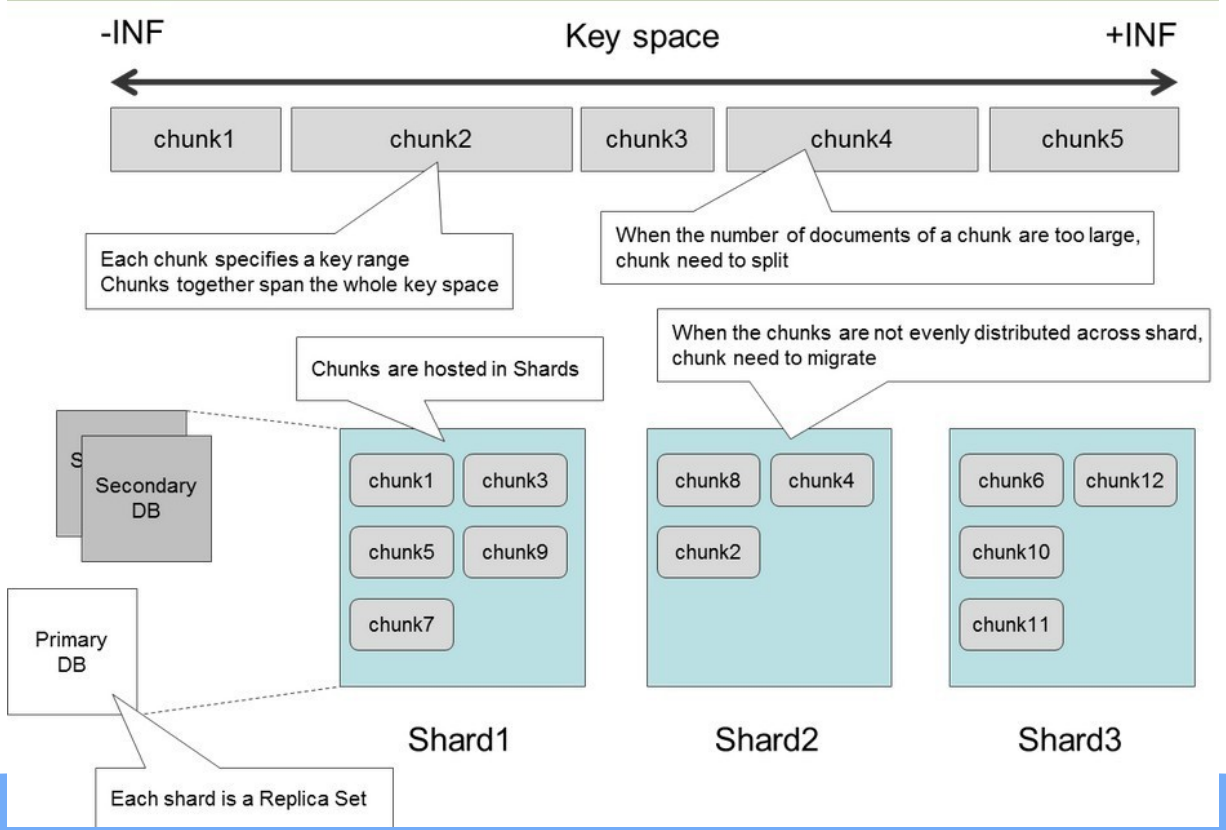
```
CREATE TABLE t1 (
        fname VARCHAR(50) NOT NULL,
        lname VARCHAR(50) NOT NULL,
        region_code TINYINT UNSIGNED NOT
NULL,
        dob DATE NOT NULL
)
PARTITION BY RANGE( region_code ) (
        PARTITION p0 VALUES LESS THAN (64),
        PARTITION p1 VALUES LESS THAN (128),
        PARTITION p2 VALUES LESS THAN (192),
        PARTITION p3 VALUES LESS THAN
MAXVALUE
);

SELECT fname, lname, region_code, dob
        FROM t1
        WHERE region_code > 125 AND
```

- 数据库进阶 5 – NoSQL 数据库 1 – MongoDB



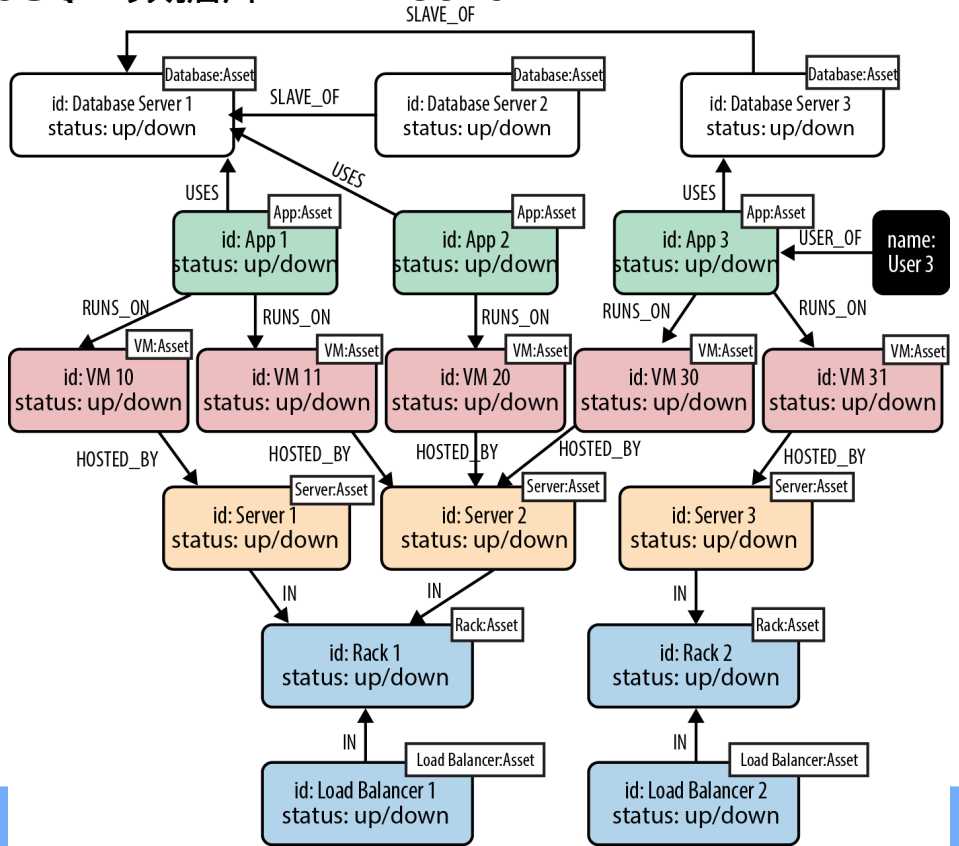


- 数据库进阶 5 – NoSQL 数据库 1 – MongoDB

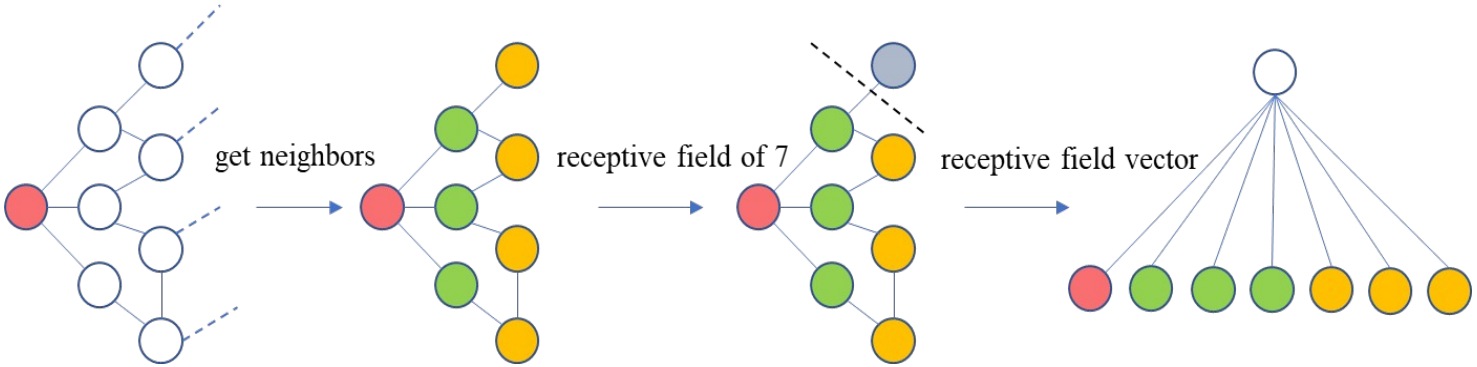


课程内容

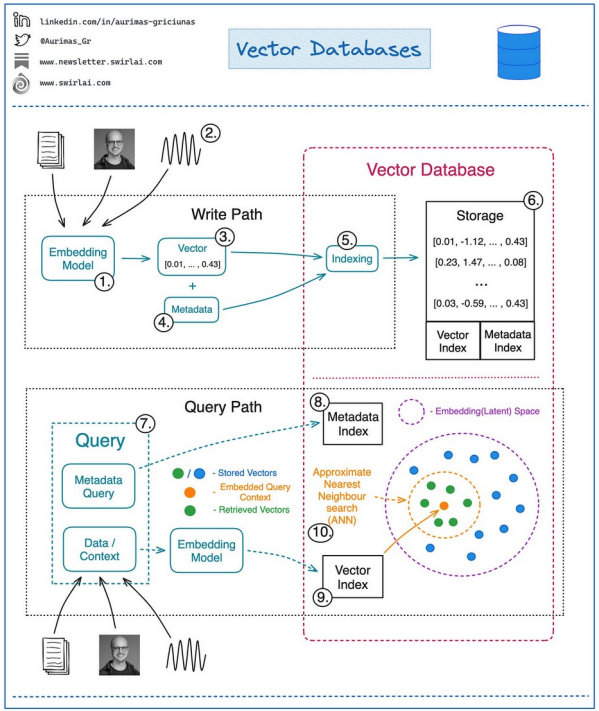
- 数据库进阶 6 – NoSQL 数据库 2 – Neo4J



- 数据库进阶 6 – NoSQL 数据库 2 – Neo4J



- 数据库进阶 7 – NoSQL 数据库 3 – Vector Database & Log Structure Database



From: https://www.newsletter.swirlai.com/p/sai-notes-07-what-is-a-vector-database

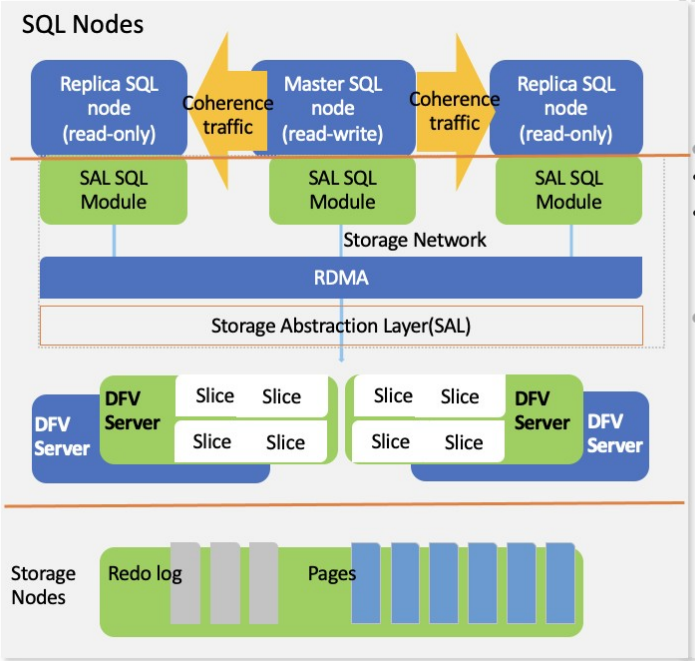
- 数据库进阶 8 – NoSQL 数据库 4 – TSDB

bucket: `my_bucket`

_time	_measurement	location	scientist	_field	_value
2019-08-18T00:00:00Z	census	klamath	anderson	bees	23
2019-08-18T00:00:00Z	census	portland	mullen	ants	30
2019-08-18T00:06:00Z	census	klamath	anderson	bees	28
2019-08-18T00:06:00Z	**census**	**portland**	**mullen**	**ants**	**32**

timestamps | measurement | Tag value | Tag value | Field key | Filed value

- 云数据库 1&2 – 云数据库 openGauss & GaussDB



主节点

- 处理所有的写
- 写 WAL 日志（Redo）

数据分区

- 数据页以 slice 组织，分布在各个 slice 服务
- Slice 冗余保存
- 日志按 slice 分发

01 02 03 04

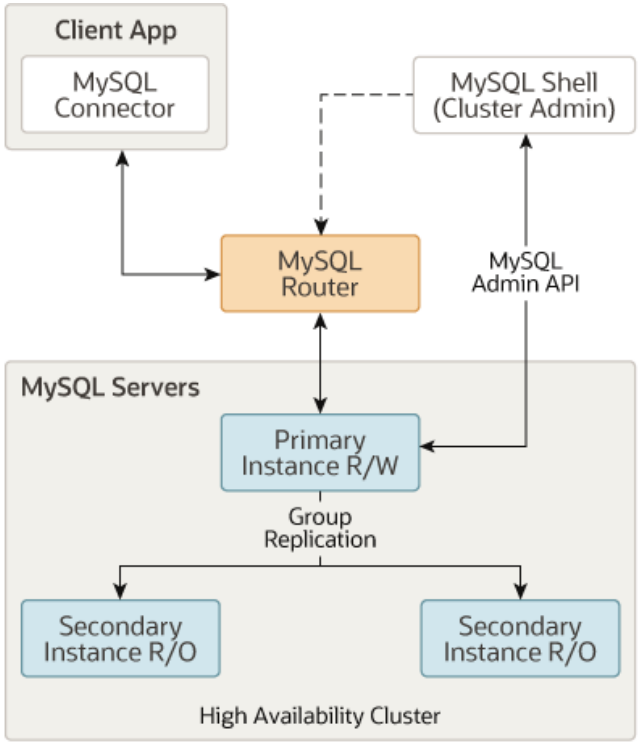
只读实例

- 处理只读请求
- 快速升主
- 快速增/删

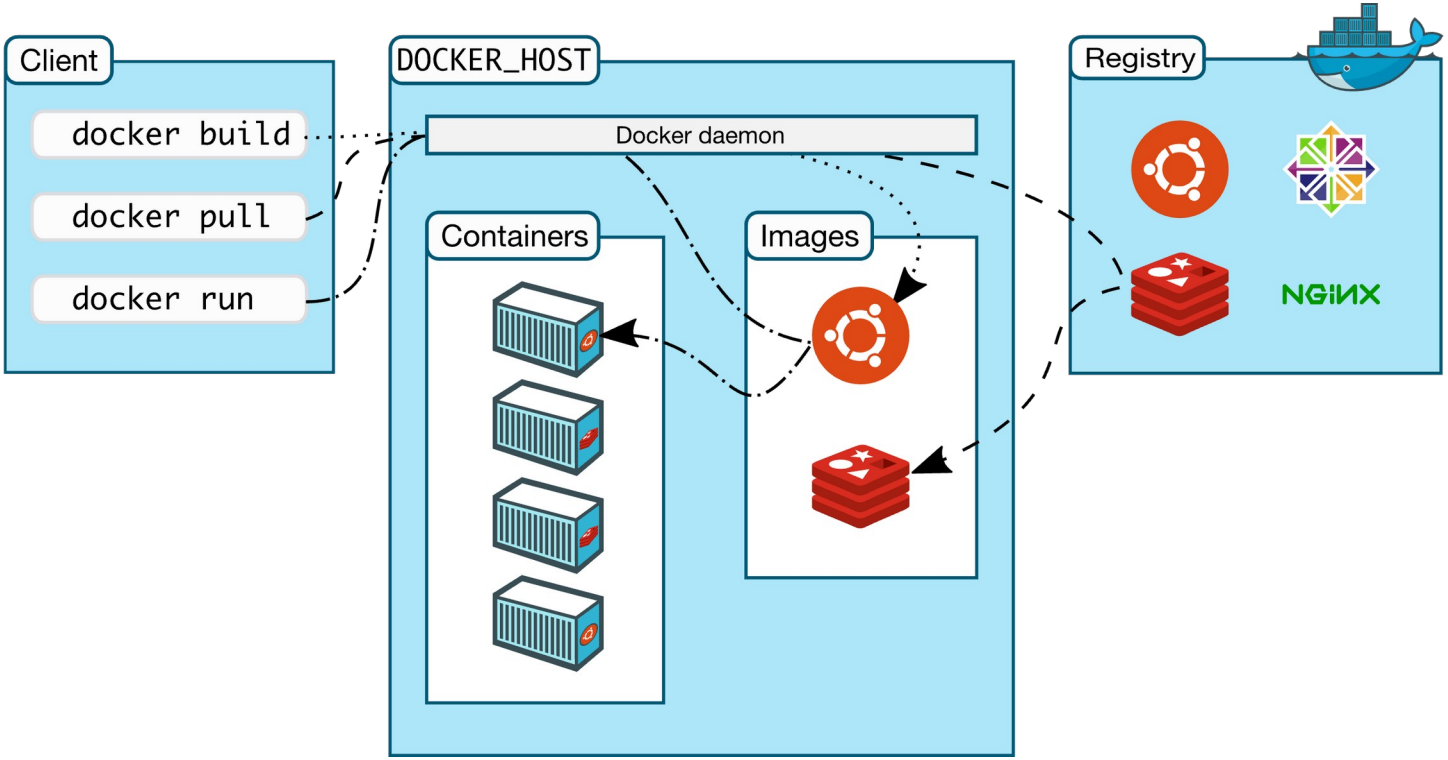
Slice 服务

- 单数据库可以分配多个 Slice，访问隔离
- 存储和处理日志记录
- 维护和重构数据页
- 服务数据页读请求

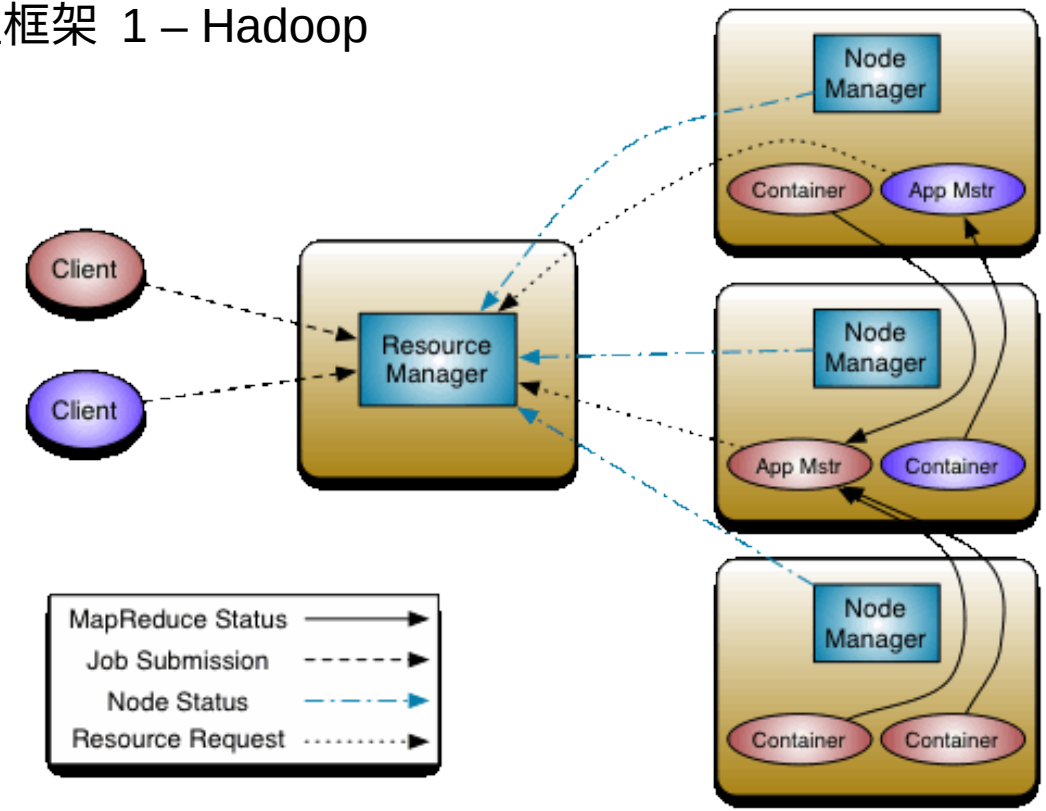
- 系统部署与运维 1 – 集群部署



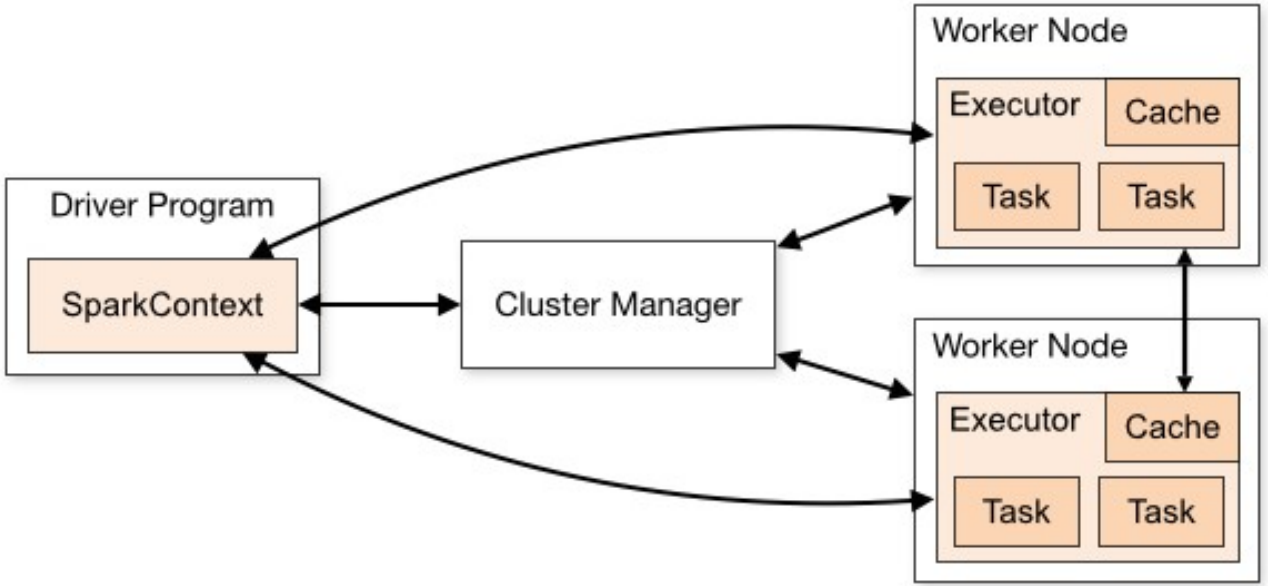
- 系统部署与运维 2 – 容器化



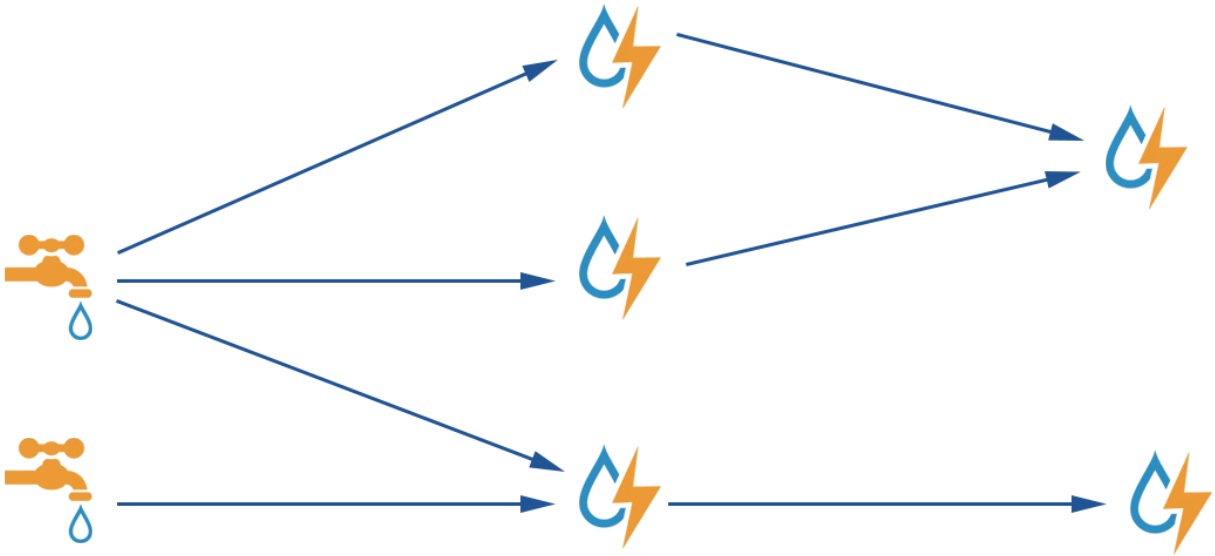
- 分布式并行处理框架 1 – Hadoop



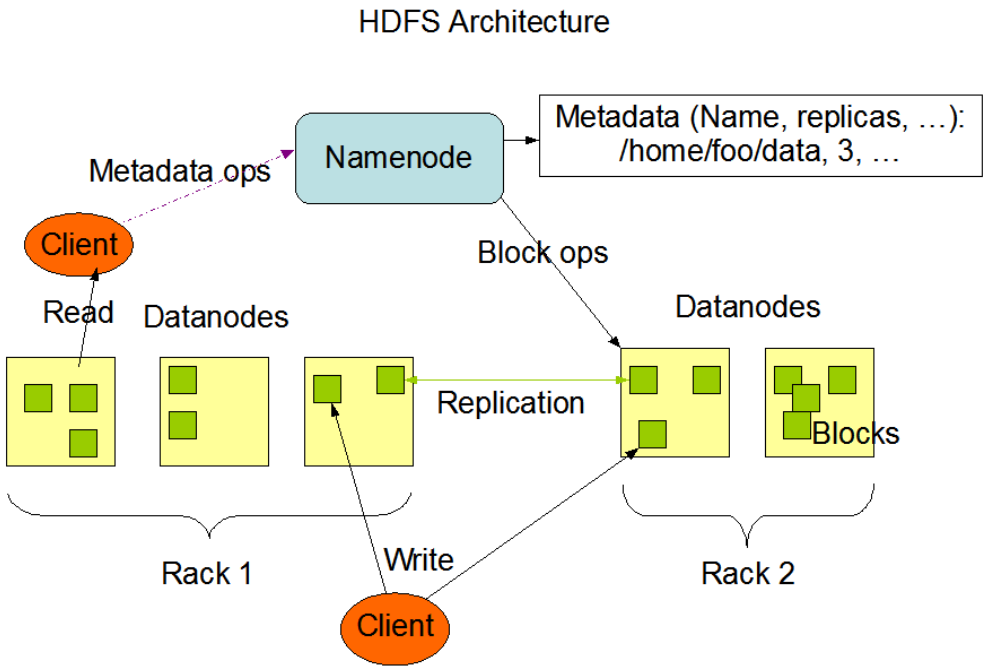
- 分布式并行处理框架 2 – Spark



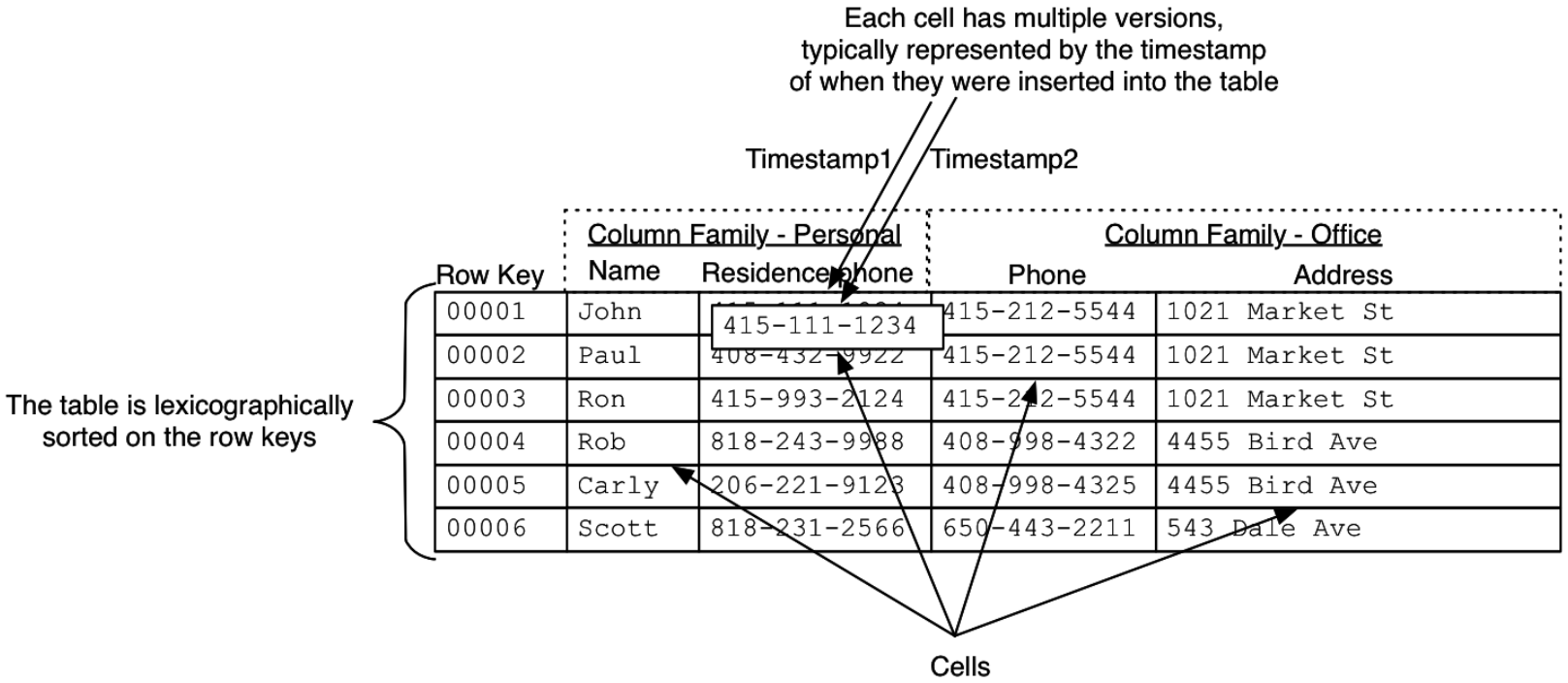
- 分布式并行处理框架 3 – Storm



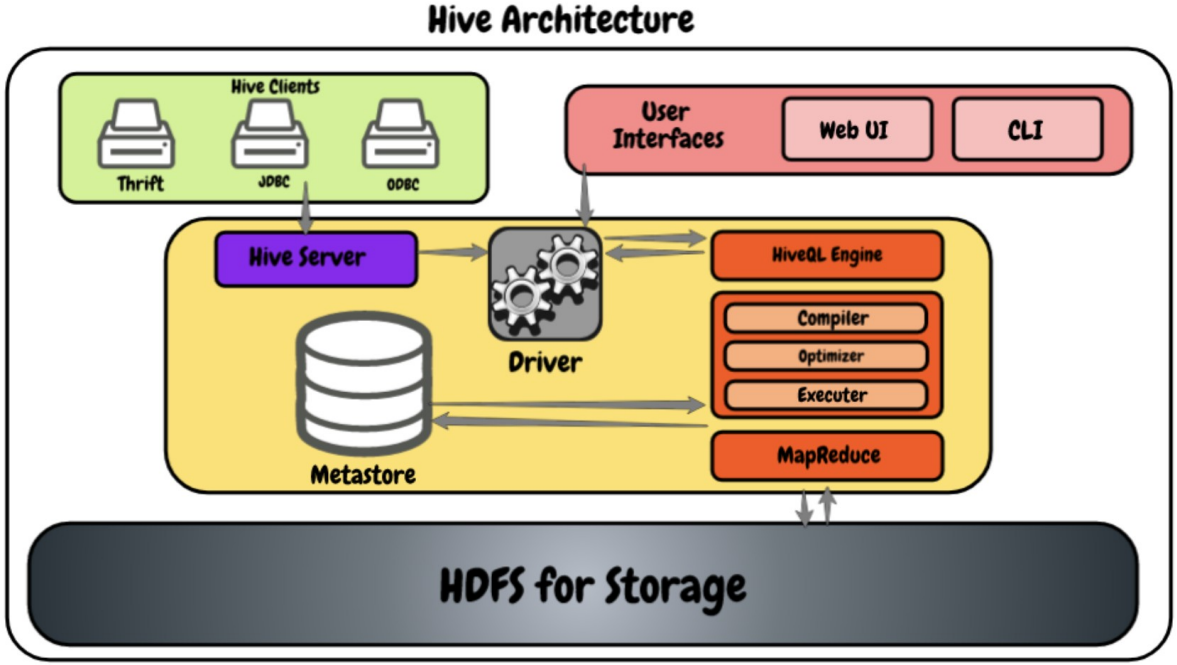
- 分布式并行处理框架 4 – 分布式文件系统



- 分布式并行处理框架 5 – 分布式数据库 HBase

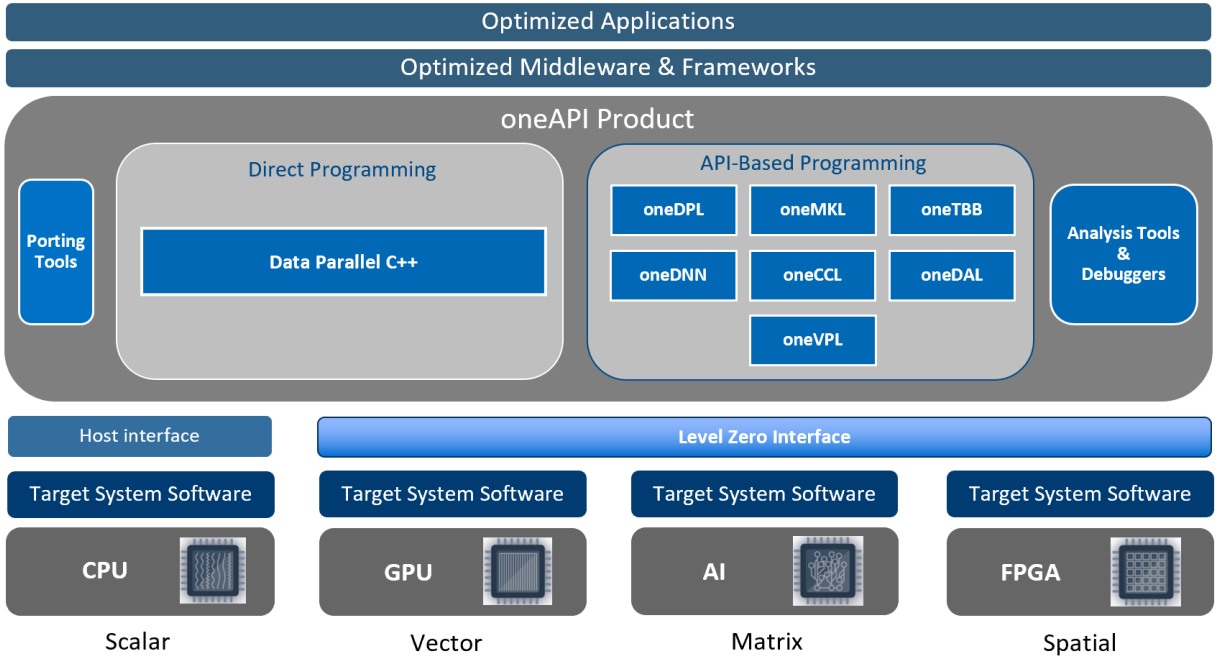


- 分布式并行处理框架 6 – 分布式数据仓库 Hive



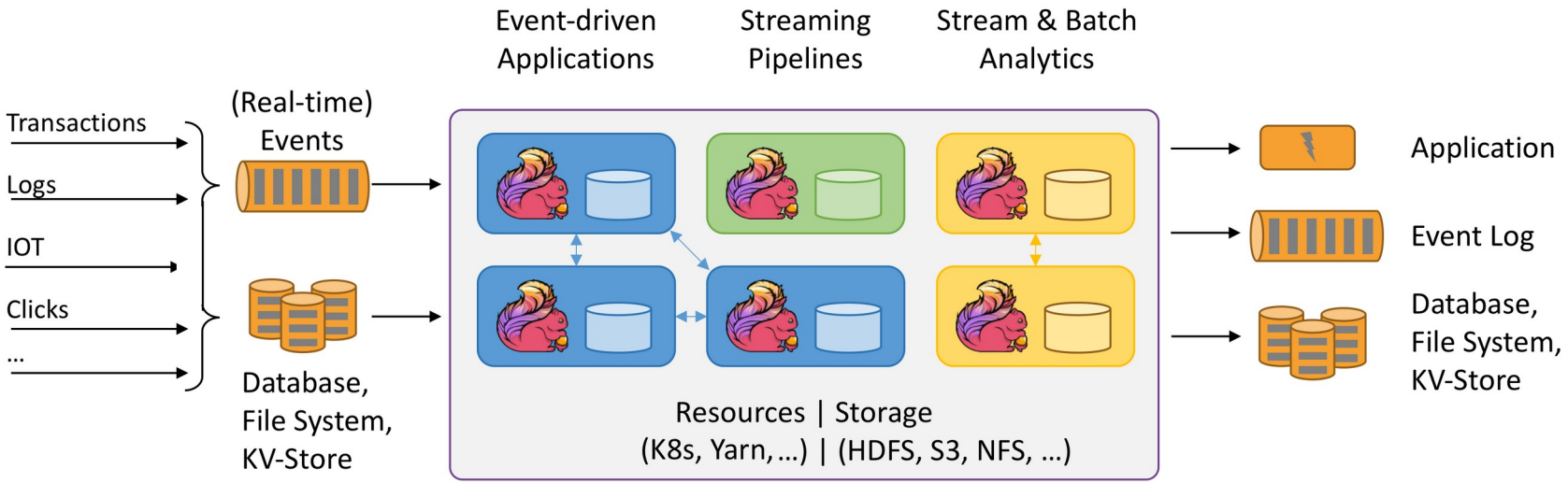
From: https://www.analyticsvidhya.com/blog/2020/10/getting-started-with-apache-hive/

- 分布式并行处理框架 7 – oneAPI



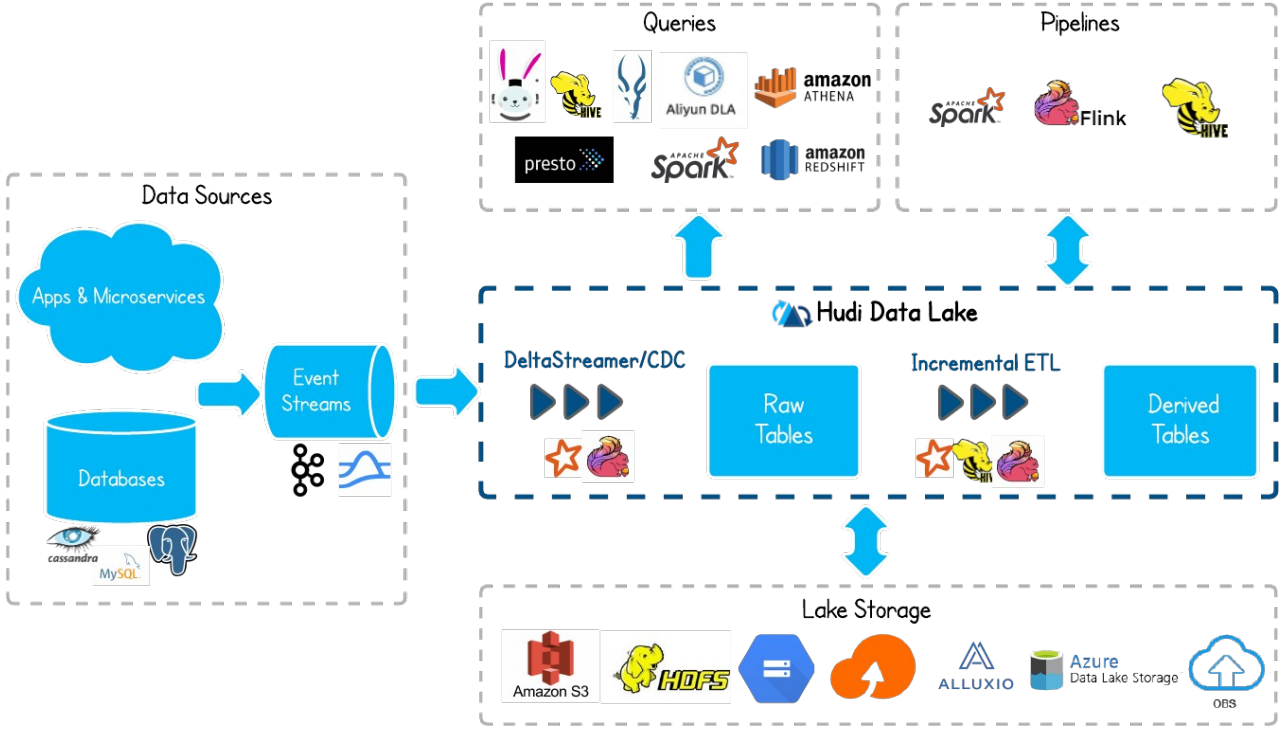
From: https://spec.oneapi.io/level-zero/latest/core/INTRO.html

- 分布式并行处理框架 8 – Flink



From: https://flink.apache.org/

- 分布式并行处理框架 9 – 数据服务 & 数据湖



考核方式

- 期末考试 – 40%
  - 开卷考试 & 全主观题
- 平时作业 – 60%
  - 共 12 次作业，每次 5 分，共 60%
    - 按课程进度安排作业
- 需求
  - 开发 e-Book store 的升级版本
  - E-BookStore 参考实现：
    - 前端： https://github.com/USTC-HuangZA/bookstore-frontend-demo
    - 后端： https://github.com/USTC-HuangZA/bookstore_backend

课程安排

REliable, INtelligent & Scalable Systems

- 上课
  - 周一　16:00-17:40　东中院 2-306
  - 周三　08:00-09:40　东中院 2-306
- 答疑
  - 周三　14:00-16:00　软件大楼 5401
- 助教
  - 曹韫琦　caoyunqi@sjtu.edu.cn
  - 林子宏　923048992@sjtu.edu.cn
  - 陶昱丞　taoyucheng@sjtu.edu.cn
  - 张海达　insular_oasis@sjtu.edu.cn
- Canvas 网址
  - https://oc.sjtu.edu.cn/courses/60117

Thank You!