

# 数据系统基础

2018-08-02

liht@fenbi.com

# 大纲

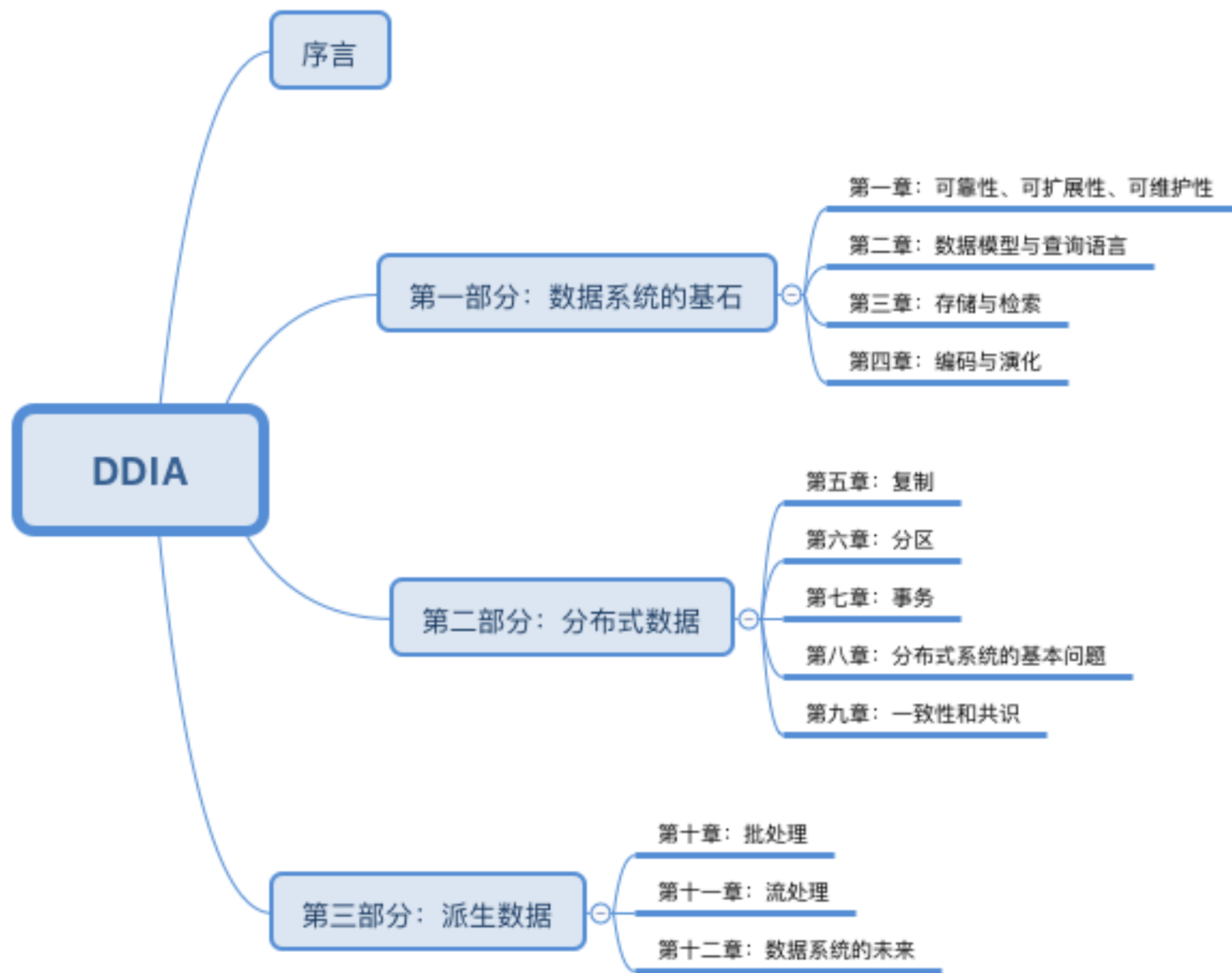
- 序言
- 可靠性、可扩展性、可维护性
- 数据模型与查询语言
- 存储与检索
- Redis

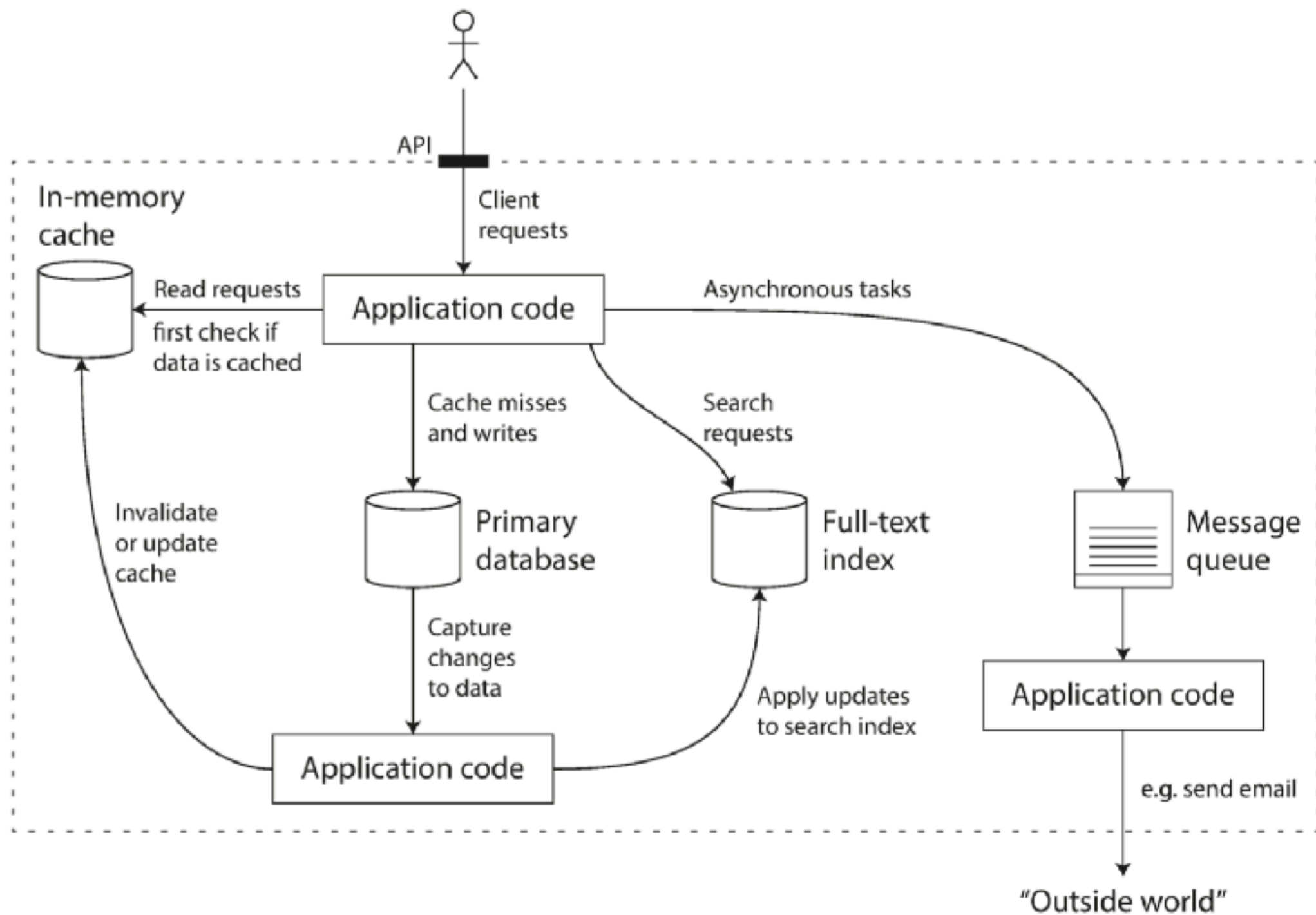
# 序言-1

- 数据存储和处理的skr词汇
  - NoSQL! Big Data! Web-Scale! Sharding! Eventual consistency! ACID! CAP theorem! Cloud services! MapReduce! Real-time!
- 相关方向的进展和变化
  - 互联网公司需要处理更多的数据，创造更有效的工具
  - 敏捷开发要保持数据模型的灵活性
  - 免费和开源软件更受欢迎
  - CPU主频增长变缓，多核处理成为标配
  - 分布式、云服务盛行
  - 服务要求高可用，不接受宕机

# 序言-2

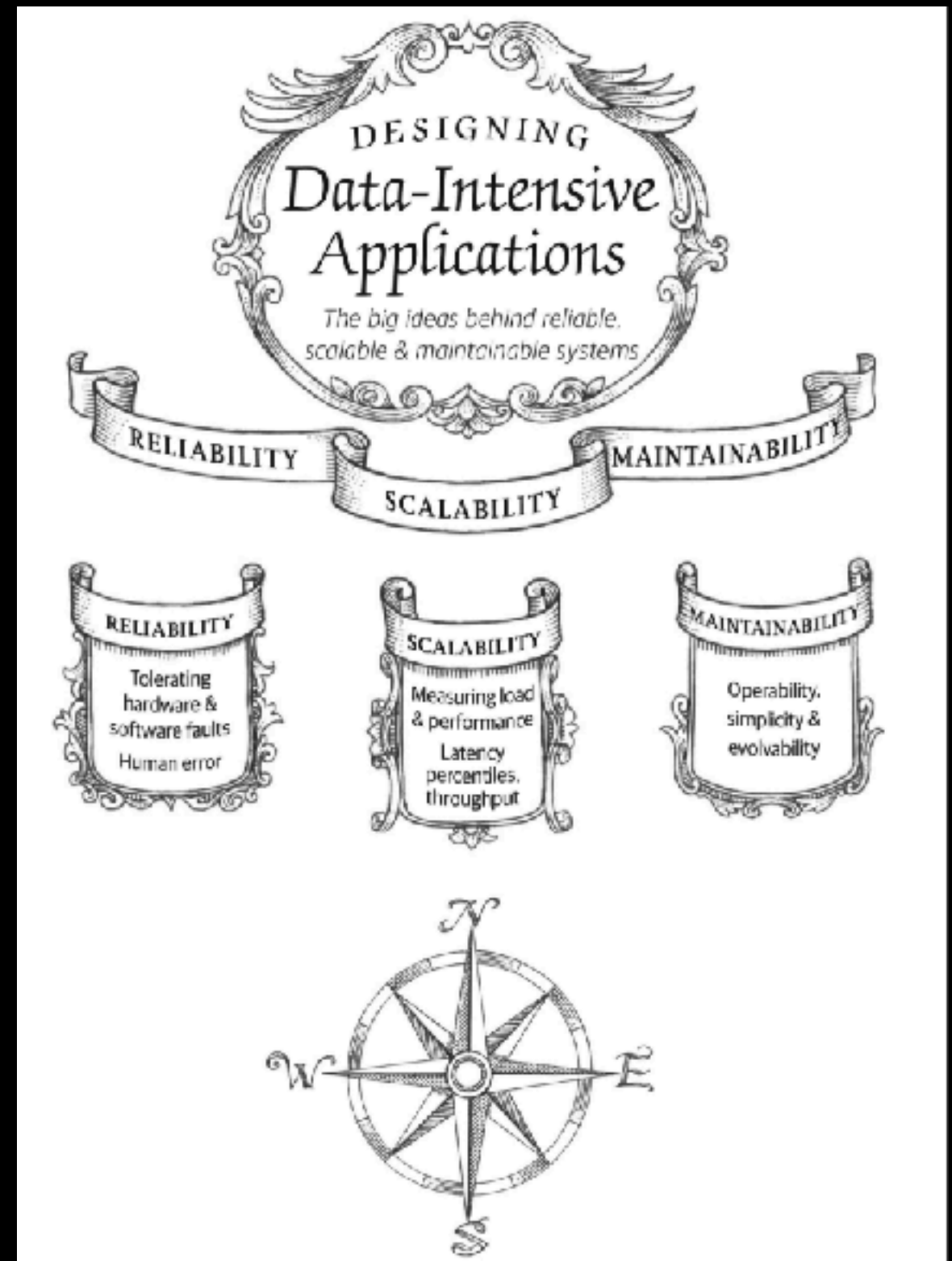
- 数据密集型 VS 计算密集型
- 数据密集型
  - 数据量、数据复杂度、数据变化速度
- 计算密集型
  - 处理器速度
- 找到技术飞速变化背后持续成立的原则
- premature optimaztion





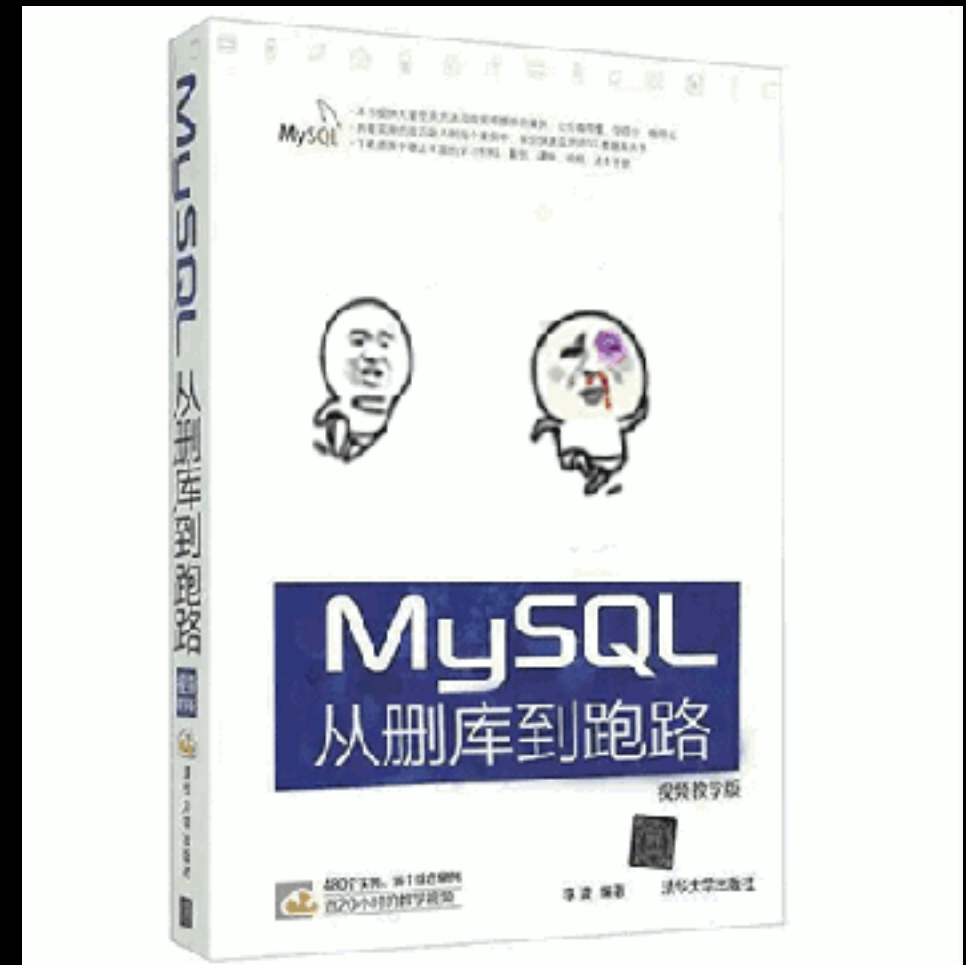
# 可靠性、可扩展性、可维护性

**The Internet was done so well that most people think of it as a natural resource like the Pacific Ocean**



# 讨论 - 可靠性、可扩展性、可维护性

- 基本概念?
- 可靠性
  - 提高可靠性的常用方法
  - 是否一直以最高可靠性为开发目标?
- 可扩展性
  - Twitter的案例讨论
  - 如何描述负载、性能
  - 响应时间avg min max median p99
  - 了解SLI、SLO、SLA <http://www.yunweipai.com/archives/10703.html>
  - 应对负载的方法



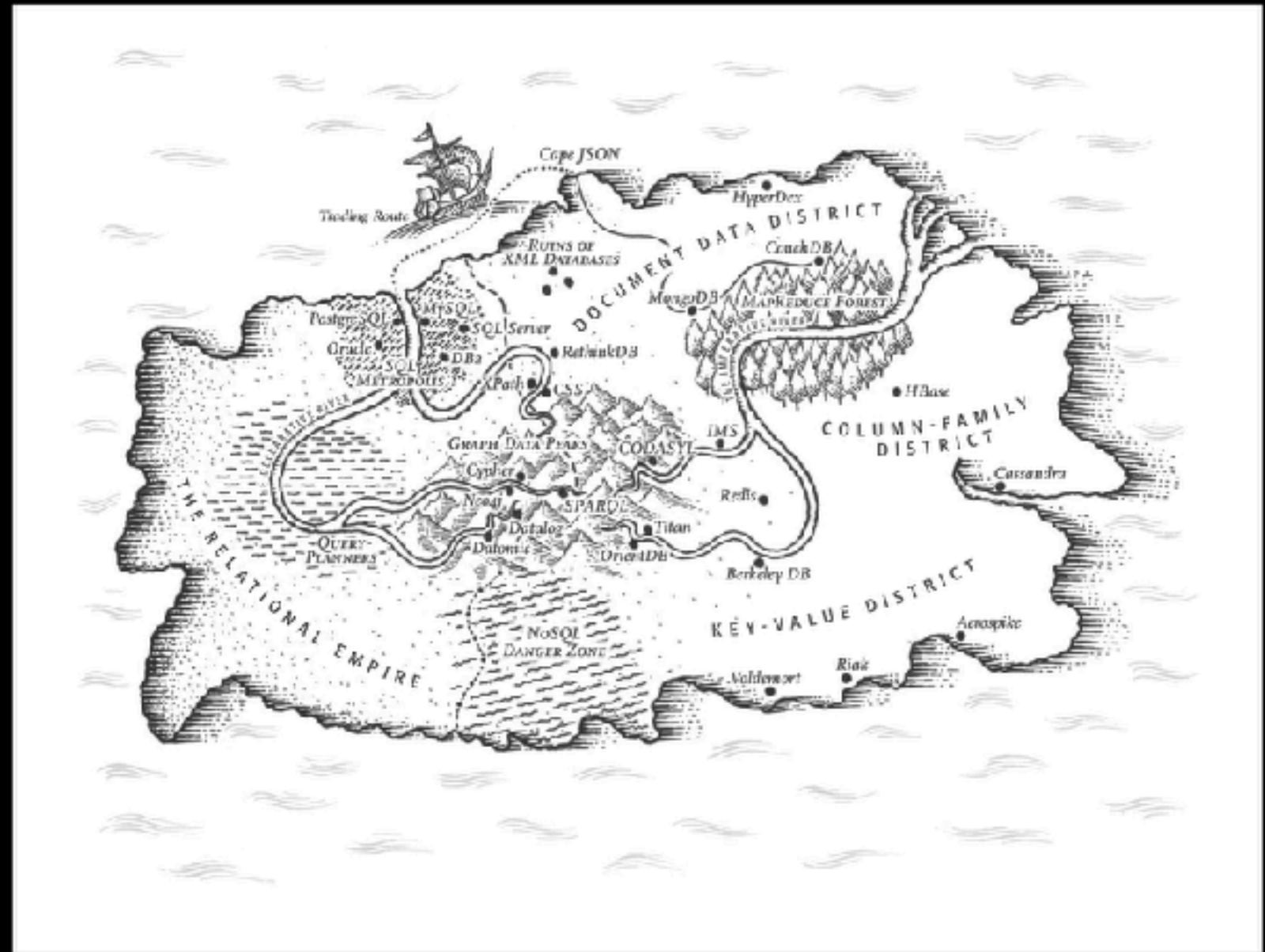


# 小结-可靠性、可扩展性、可维护性

- 可靠性
  - 不挂是不可能不挂的，这辈子不可能不挂，挂了也要继续服务。
- 可扩展性
  - 不变是不可能不变的，这辈子不可能不变，变是为了更好的服务。
- 可维护性
  - 不接锅是不可能不接锅的，这辈子不可能不接锅，结了锅也得正常服务。

# 数据模型与查询语言

**The limits of my  
language mean the limits  
of my world**



# 讨论 - 数据模型与查询语言

- 基本概念：关系模型、文档模型、图数据模型、NoSQL
- 文档模型
  - 文档模型只能保存单一的文档，不能处理相对关系？
  - 文档模型是完全 schemaless 吗？
  - 文档模型能做的关系模型也能做？那文档模型的优势是什么？
- 关系型数据库与文档数据库在今日的对比
  - 哪个数据模型更方便写代码？
  - 文档模型中的架构灵活性
  - 查询的数据局部性
  - 文档和关系数据库的融合

# 讨论 - 数据模型与查询语言

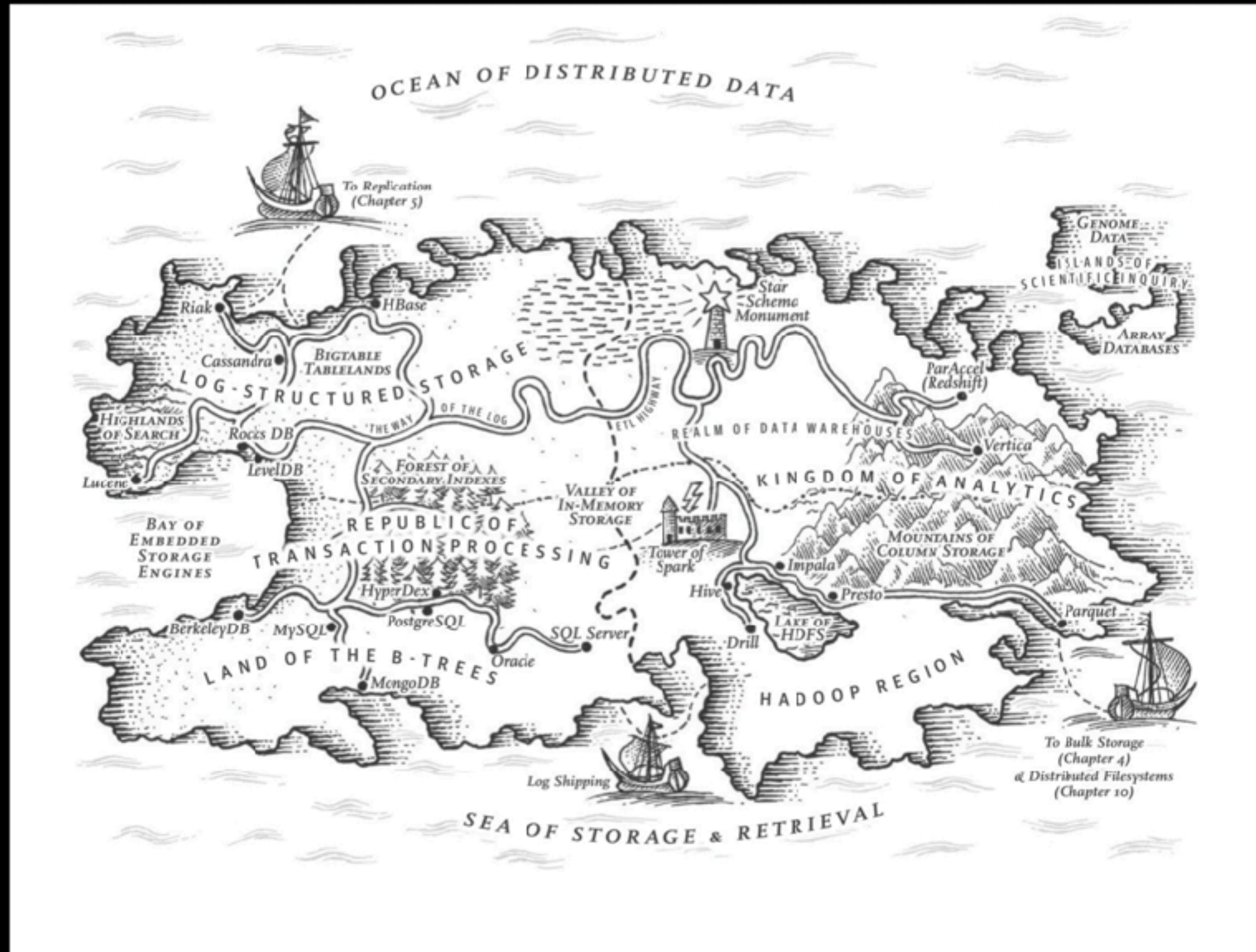
- 数据查询语言
  - 声明式 VS 命令式
  - MapReduce查询
- 图数据模型
  - 一张图是否只能处理一种类型的关联?
  - 属性图 VS 三元组

# 小结-数据模型与查询语言

- 3种模型
  - 关系
  - 文档
  - 图
- 每个数据模型都具有各自的查询语言或框架
  - 声明式
  - 命令式
  - MapReduce

# 存储与检索

If you keep things tidily ordered, you're just too lazy to go searching.



# 存储与检索

wengyk && dengzh && zhaoyang

# 存储与检索

- 驱动数据库的数据结构
  - hash索引
    - 追加日志、分段压缩、删除记录
    - 追加日志 VS 更新日志
  - SSTables和LSM树
    - 合并和压缩的过程
    - 如何崩溃恢复
  - B 树
    - 为什么具有  $n$  个键的B树总是具有  $O(\log n)$  的深度
    - B 树的存储和查询过程
    - WAL 是什么？为什么需要WAL
  - 比较B树和LSM树



# 存储与检索

- 其他存储结构
- 事务处理还是分析？
- 列存储
  - 基本概念
  - 列压缩，位图编码
  - 有排序规则时，列存储是怎么写入的？

属性	事务处理 OLTP	分析系统 OLAP
主要读取模式	查询少量记录，按键读取	在大批量记录上聚合
主要写入模式	随机访问，写入要求低延时	批量导入（ETL），事件流
主要用户	终端用户，通过Web应用	内部数据分析师，决策支持
处理的数据	数据的最新状态（当前时间点）	随时间推移的历史事件
数据集尺寸	GB ~ TB	TB ~ PB

# 小结-存储与检索

- 存储引擎分为两大类
  - 优化事务处理 (OLTP)
  - 优化分析 (OLAP)
- 两大主流学派的存储引擎
  - 日志结构学派
  - 就地更新学派

# Redis

- 用到了我们上面讨论的哪些技术？
  - 数据模型
  - 查询语言
  - 存储
  - 检索

# Redis的过期键删除策略

xingrb

# Redis的数据淘汰机制

# Redis的数据淘汰机制

- volatile-lru: 从已设置过期时间的数据集 (server.db[i].expires) 中挑选最近最少使用的数据淘汰
- volatile-ttl: 从已设置过期时间的数据集 (server.db[i].expires) 中挑选将要过期的数据淘汰
- volatile-random: 从已设置过期时间的数据集 (server.db[i].expires) 中任意选择数据淘汰
- allkeys-lru: 从数据集 (server.db[i].dict) 中挑选最近最少使用的数据淘汰
- allkeys-random: 从数据集 (server.db[i].dict) 中任意选择数据淘汰
- no-eviction (驱逐): 禁止驱逐数据
- volatile-lfu
- allkeys-lfu

# 讨论 - Redis

- 持久化
  - RDB
    - SAVE VS BGSAVE
    - RDB文件结构
  - AOF
    - appendfsync
    - 重写
  - 如何处理过期键
  - COW
- redis.conf
  - <http://download.redis.io/redis-stable/redis.conf>
- 猿辅导线上是怎么配置

```
[maintain@dx-tutor-timeline-redis3-online ~]$ crontab -l
40 0 * * * sh /home/maintain/online/redis_rewrite.sh 6379
10 1 * * 3,6 sh /home/maintain/online/redis_bgsave.sh 6379
```

# 讨论 - Redis

- 为什么说redis是单线程的？服务端单线程怎么和客户端线程池怎么配合？



# 小结 - Redis

- Redis的过期键删除策略
- Redis的数据淘汰机制
- RDB
- AOF

# 下集预告

- 数据库事务的概念和实现
  - AAID
    - 基本概念 ACID BASE
    - 隔离级别
    - 可序列化
  - Redis 事务处理
  - MySQL 事务处理