- 唐建法 / TJ
- MongoDB 大中华区首席架构师
- MongoDB 中文社区主席

MONGODB IODP – 智能操作数据平台

mongoDB®

Columbia River Gorge

从帆板到风筝冲浪 – Hood River, Oregon

15年前

今天

帆板

滑行速度：较慢

改变方向：困难

风筝冲浪

滑行速度：快，世界纪录101公里

改变方向：容易

关系模型

Customer ID	First Name	Last Name	City
0	John	Doe	New York
1	Mark	Smith	San Francisco
2	Jay	Black	Newark
3	Meagan	White	London
4	Edward	Daniels	Boston

性能：较难扩展
改变方向（模式）：困难

JSON文档模型

```
{       customer_id : 1,
        first_name : "Mark",
        last_name : "Smith",
        city : "San Francisco",
        phones: [     {
                number :  “1-212-777-1212”  ,
dnc : true,
                type :  “home”
        }]
}
```

性能：易横向扩展
改变方向（模式）：容易
容易开发

目录 Content

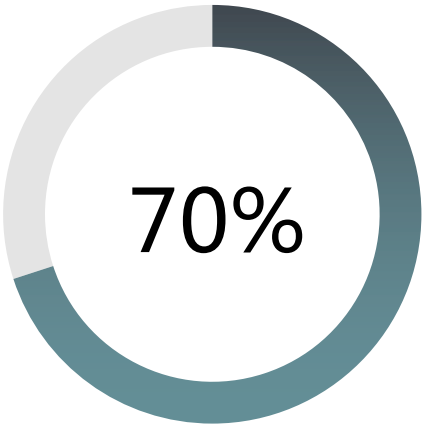
后Hadoop时代的大数据技术思考

01

内容

- 大数据技术分类及衡量标准
- 数据技术比较分析
- 操作型大数据－解决更加实时的场景需求

Hadoop 神话面临严峻挑战
70%的Hadoop部署未实现价值



的大数据分析项目未能体现预期价值，Hadoop体系正在面临严峻挑战

-- Gartner

The Register®
Biting the hand that feeds IT

DATA CENTER | SOFTWARE | SECURITY | TRANSFORMATION | DEVOPS | BUSINESS | PERSONAL TECH | SCIENCE | EMERGENT TECH | BOOTNOTES

Software

IBM leads BigInsights for Hadoop out behind barn. Shots heard

Data analytics platform sunset in December, but enterprise version spared

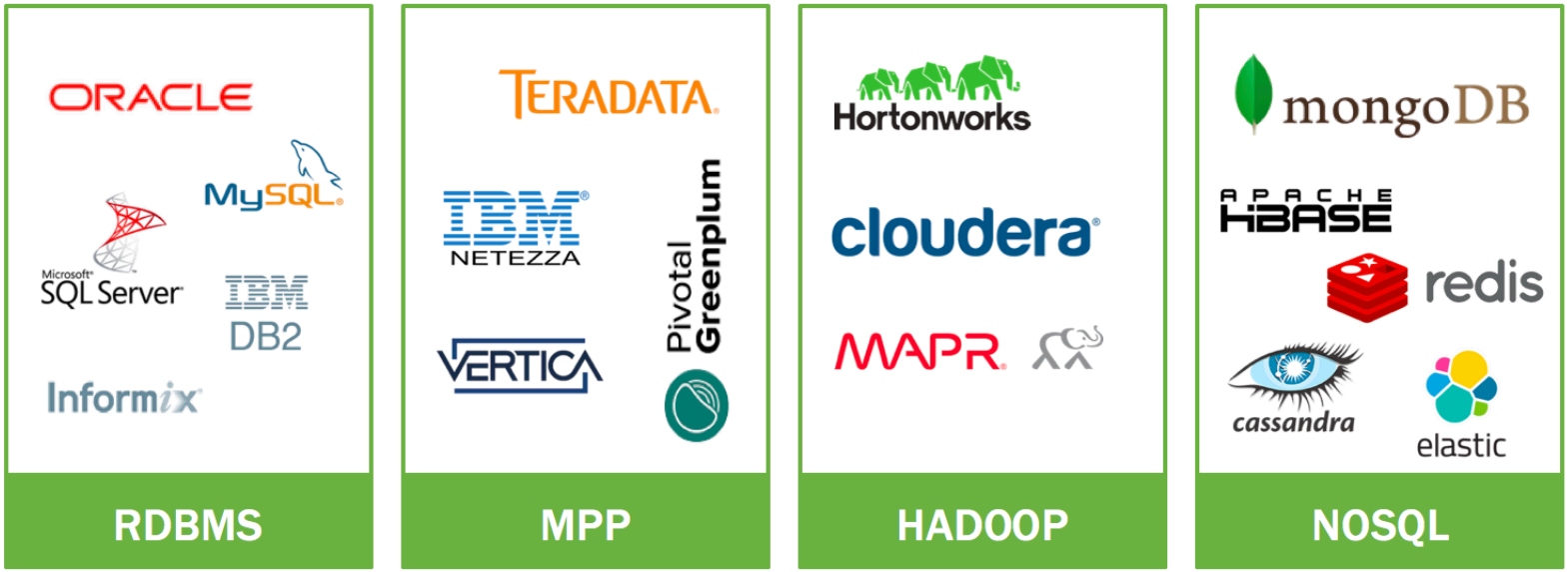
By Rebecca Hill 8 Nov 2017 at 13:29 SHARE ▼

IBM has announced the retirement of the basic plan for its data analytics software platform, BigInsights for Hadoop.

The basic plan of the service will be retired in a month, on December 7 of this year. IBM said the enterprise plan would continue to be available.

IBM 去年宣布其大数据产品BigInsights下架，正式放弃Hadoop

大数据技术分类及代表产品



数据技术的衡量标准
Gartner的3V + ACID

+

能否处理 **足够灵活** 的数据类型？

能否处理的 **足够快**？

能否处理 **足够大** 的数据量？

能否支持 **ACID事务**？

数据库技术类型短板分析

	海量数据 VOLUME	速度与并发 VELOCITY	多结构数据 VARIETY	事务支持 ACID

数据库技术类型短板分析

	海量数据 VOLUME	速度与并发 VELOCITY	多结构数据 VARIETY	事务支持 ACID
RDBMS				

数据库技术类型短板分析

	海量数据 VOLUME	速度与并发 VELOCITY	多结构数据 VARIETY	事务支持 ACID
RDBMS				
MPP				

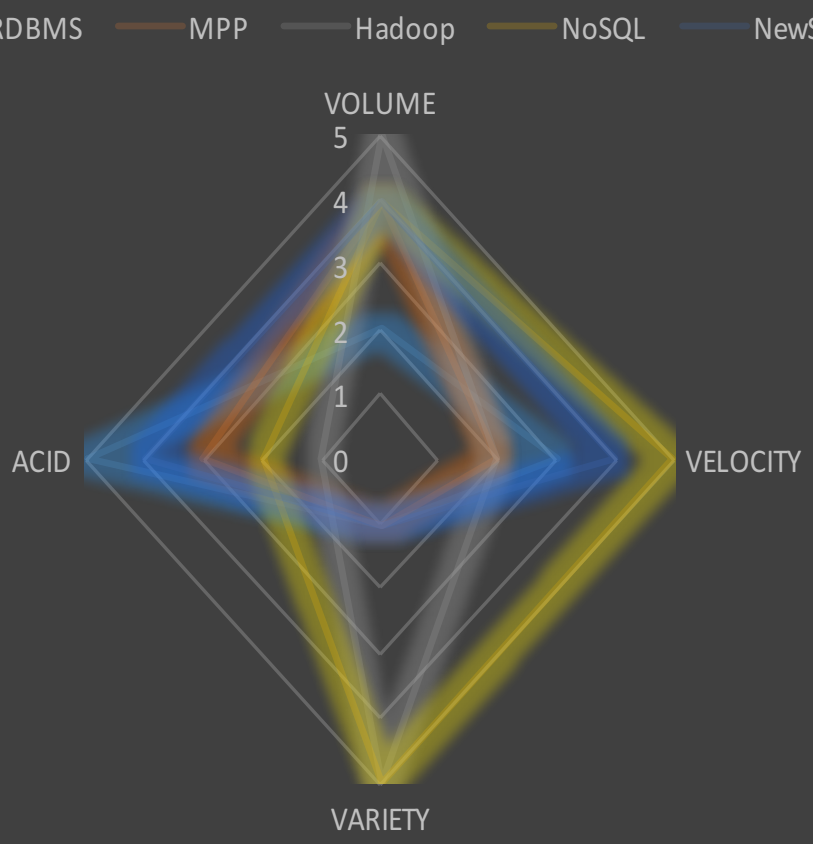
数据库技术类型短板分析

	海量数据 VOLUME	速度与并发 VELOCITY	多结构数据 VARIETY	事务支持 ACID
RDBMS				
MPP				
Hadoop				

数据库技术类型短板分析

	海量数据 VOLUME	速度与并发 VELOCITY	多结构数据 VARIETY	事务支持 ACID
RDBMS				
MPP				
Hadoop				
NoSQL				

数据库场景雷达图



Hadoop 应用场景主要应用于海量数据存储及**离线分析型**应用场景

趋势分析
机器学习
分析报表

以MongoDB为代表的新一代分布式数据库则解决**在线操作型**应用场景

客户360度视图
手机APP
物联网
开发平台
实时风控

MongoDB IODP
智能操作数据平台

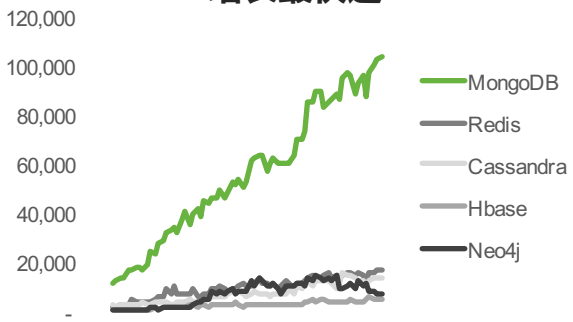
02

内容

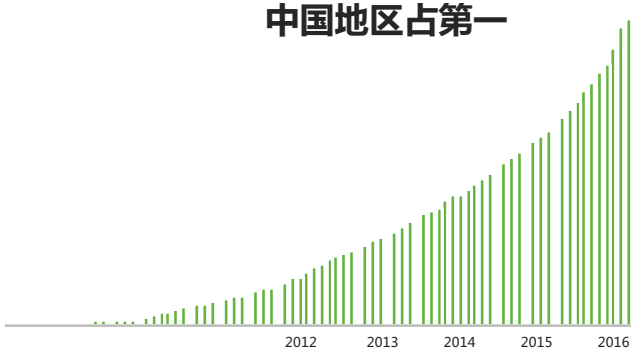
- MongoDB基本介绍
- IODP 智能操作数据平台

MongoDB IODP
智能操作数据平台

Stack Overflow使用热度
增长最快速



全球3000多万下载量
中国地区占第一



26年来第一家
IPO数据库公司

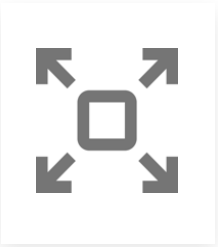
1. 最佳的方式
来管理你的数据

2. 智能化的分布你的数据
在你需要的地方

3. 任意时候任何地点 运
行你的数据库

1. 最佳的方式来管理数据

简单: 以自然的方式来建模，以直观的方式来与数据库交互

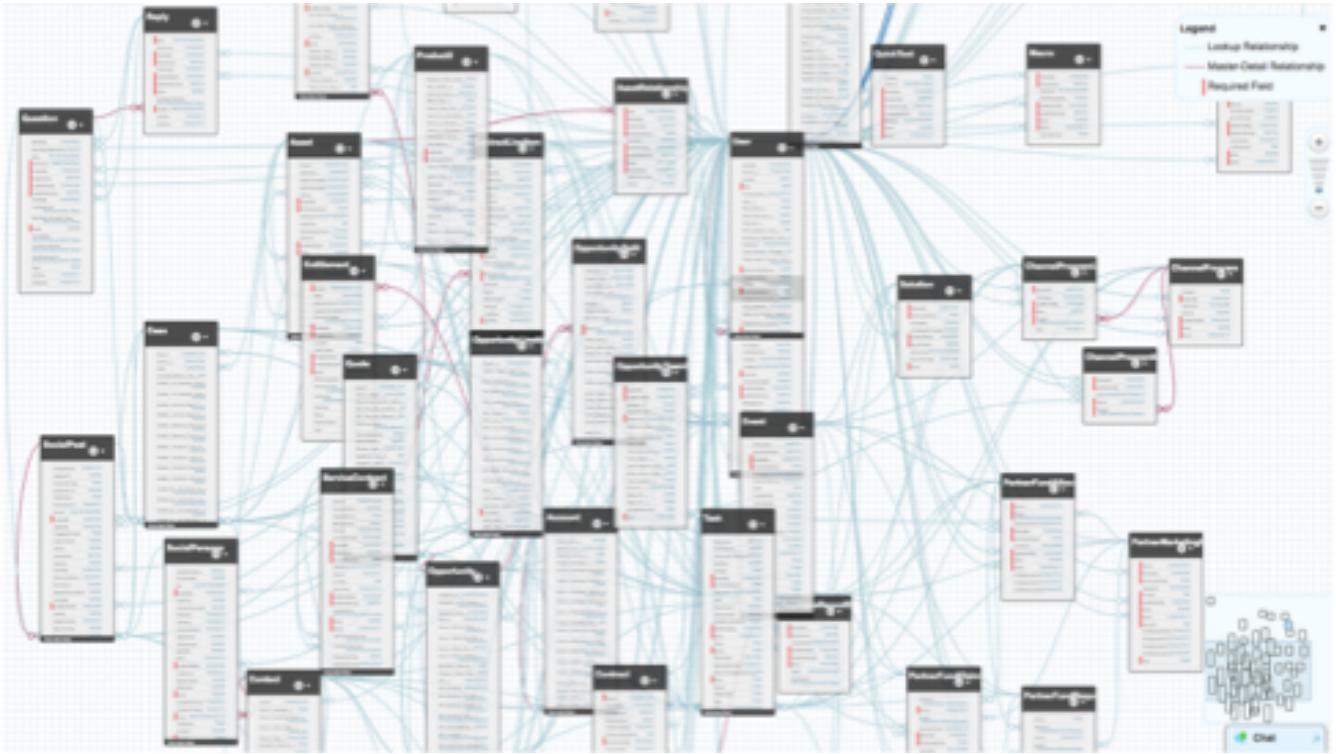


灵活: 弹性模式从容响应需求的频繁变化

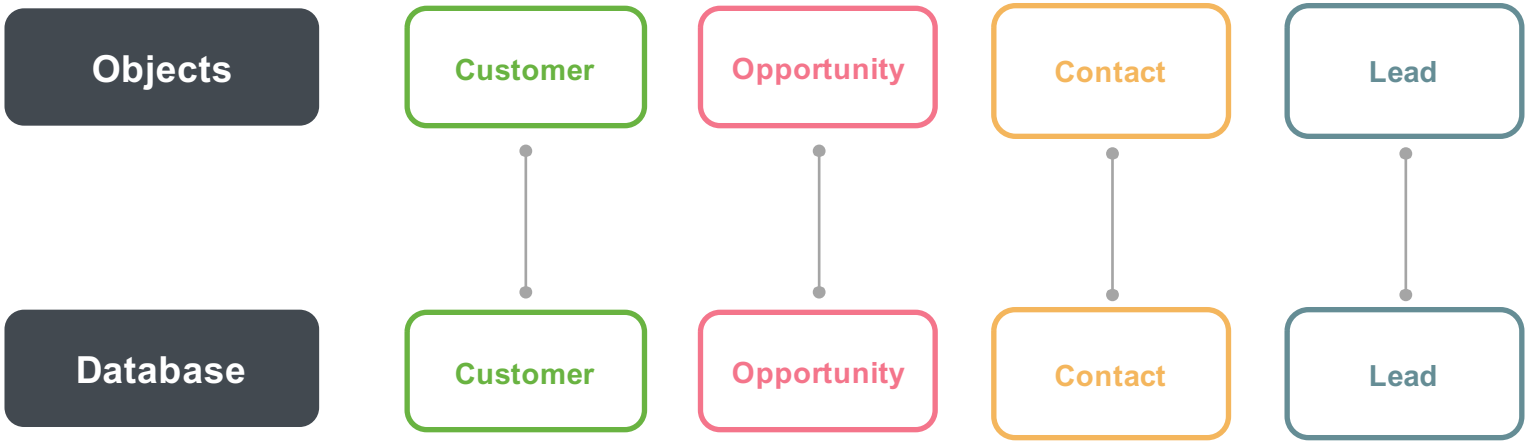
快速开发: 做更多的事，写更少的代码

多模支持: 支持多种类型的数据建模和查询方式

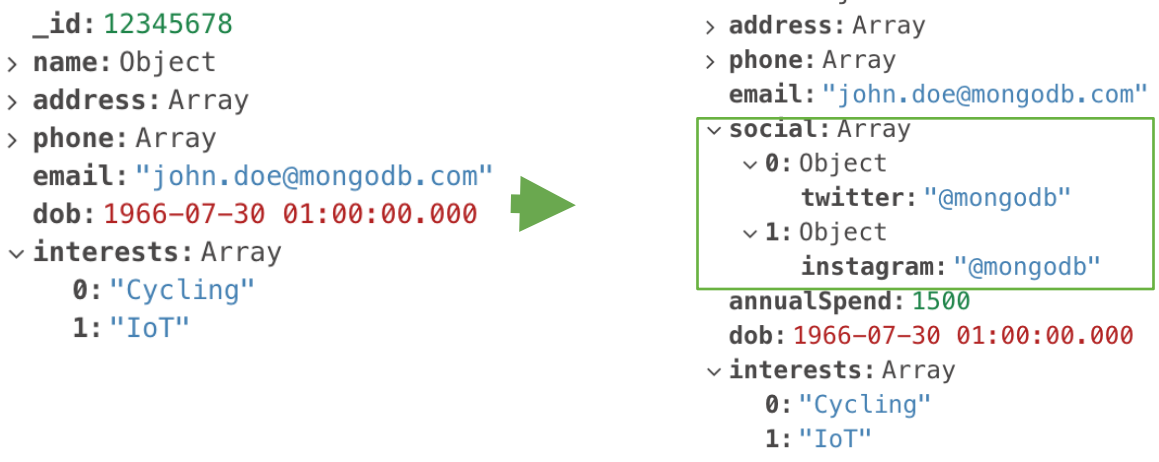
从错综复杂的关系模型



到一目了然的对象模型


Objects
Customer
Opportunity
Contact
Lead
Database
Customer
Opportunity
Contact
Lead

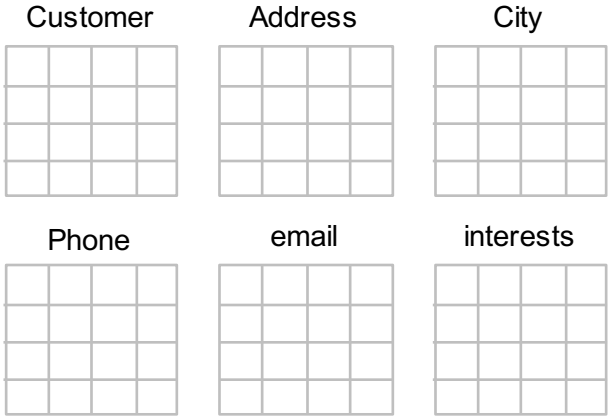
灵活: 快速响应业务变化



可动态增加新字段

- **多形性:** 同一个集合中可以包含不同字段（类型）的文档对象
- **动态性:** 线上修改数据模式，修改是应用与数据库均无须下线
- **数据治理:** 支持使用JSON Schema 来规范数据模式。在保证模式灵活动态的前提下，提供数据治理能力

快速: 最简单快速的开发方式




```
  _id: 12345678
> name: Object
> address: Array
> phone: Array
  email: "john.doe@mongodb.com"
  dob: 1966-07-30 01:00:00.000
v interests: Array
    0: "Cycling"
    1: "IoT"
```

JSON 模型之快速特性:

- 数据库引擎只需要在一个存储区读写
- 反范式、无关联的组织极大优化查询速度
- 程序API自然，开发快速

SQL：插入一个客户相关数据

```
import mysql.connector
from mysql.connector import errorcode

def addUser(connection, user):
  cursor = connection.cursor()

  customerInsert = (
      "INSERT INTO customer (first_name, last_name, email, "
      "DOB, annual_spend) VALUES "
      "(%(first)s, %(last)s, %(email)s, %(dob)s, %
(spend)s)")

  customerData = {
      'first': user['name']['first'],
      'last': user['name']['second'],
      'email': user['email'],
      'dob': user['dob'],
      'spend': user['annualSpend']
  }

  cursor.execute(customerInsert, customerData)
  customerId = cursor.lastrowid

  cityQuery = ("SELECT city_id FROM city WHERE "
      "city = %(city)s")
  for address in user['address']:
      cursor.execute(cityQuery, {'city': address['city']})
      city_id = cursor.fetchone()[0]

      addressInsert = (
          "INSERT INTO address (address, address2, district,"
          " city_id, postal_code, customer_id, location) "
          "VALUES (%(add)s, %(add2)s, %(dist)s, %(city)s, "
          "%(post)s, %(cust)s, %(loc)s)")
```

```
customerData = {
    'first': user['name']['first'],
    'last': user['name']['second'],
    'email': user['email'],
    'dob': user['dob'],
    'spend': user['annualSpend']
}

cursor.execute(customerInsert, customerData)
customerId = cursor.lastrowid

cityQuery = ("SELECT city_id FROM city WHERE "
    "city = %(city)s")
for address in user['address']:
    cursor.execute(cityQuery, {'city': address['city']})
    city_id = cursor.fetchone()[0]

    addressInsert = (
        "INSERT INTO address (address, address2, district,"
        " city_id, postal_code, customer_id, location) "
        "VALUES (%(add)s, %(add2)s, %(dist)s, %(city)s, "
        "%(post)s, %(cust)s, %(loc)s)")
```

```
    addressData = {
        'add': address['number'],
        'add2': address['street'],
        'dist': address['state'],
        'city': city_id,
        'post': address['postalCode'],
        'cust': customerId,
        'loc':  address['location']
    }

    cursor.execute(addressInsert, addressData)

topicQuery = ("SELECT topics_id FROM topics "
    "WHERE subject = %(subj)s")
interestInsert = (
    "INSERT into interests (topic_id, customer_id) "
    "VALUES (%(topic)s, %(cust)s)")

for interest in user['interests']:
    topicId = 0
    topicData = {
        'subj': interest['interest']
    }
```

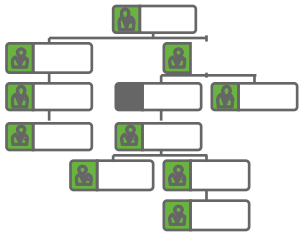
```
    cursor.execute(topicQuery, topicData)
    row = cursor.fetchone()
    if row is None:
        topicInsert = ("INSERT INTO topics (subject) "
            " VALUES (%(subj)s)")
        cursor.execute(topicInsert, topicData)
        topicId = cursor.lastrowid
    else:
        topicId = row[0]

    interestData = {
        'topic': topicId,
        'cust': customerId
    }
    cursor.execute(interestInsert, interestData)

phoneInsert = (
    "INSERT INTO `phone numbers` (customer_id, "
    "phone_number, `Phone number_type`) "
    "VALUES (%(cust)s, %(num)s, %(type)s)")
for phoneNumber in user['phone']:
    phoneData = {
        'cust': customerId,
        'num': phoneNumber['number'],
        'type': phoneNumber['location']
    }
    cursor.execute(phoneInsert, phoneData)

connection.commit()
cursor.close()
return customerId
```

快速: MongoDB 只需要两行代码

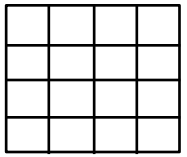
```
def addUser(database, user):
    return database.customers.insert_one(user).inserted_id
```

Code snippet: https://gist.github.com/am-MongoDB/e4f196e85f1946ed66480bce6616a94e

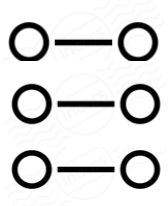
多模支持: 多种数据模型，多种查询方式



JSON



Tabular 表格



键值 Key Value

文本

地理信息

图

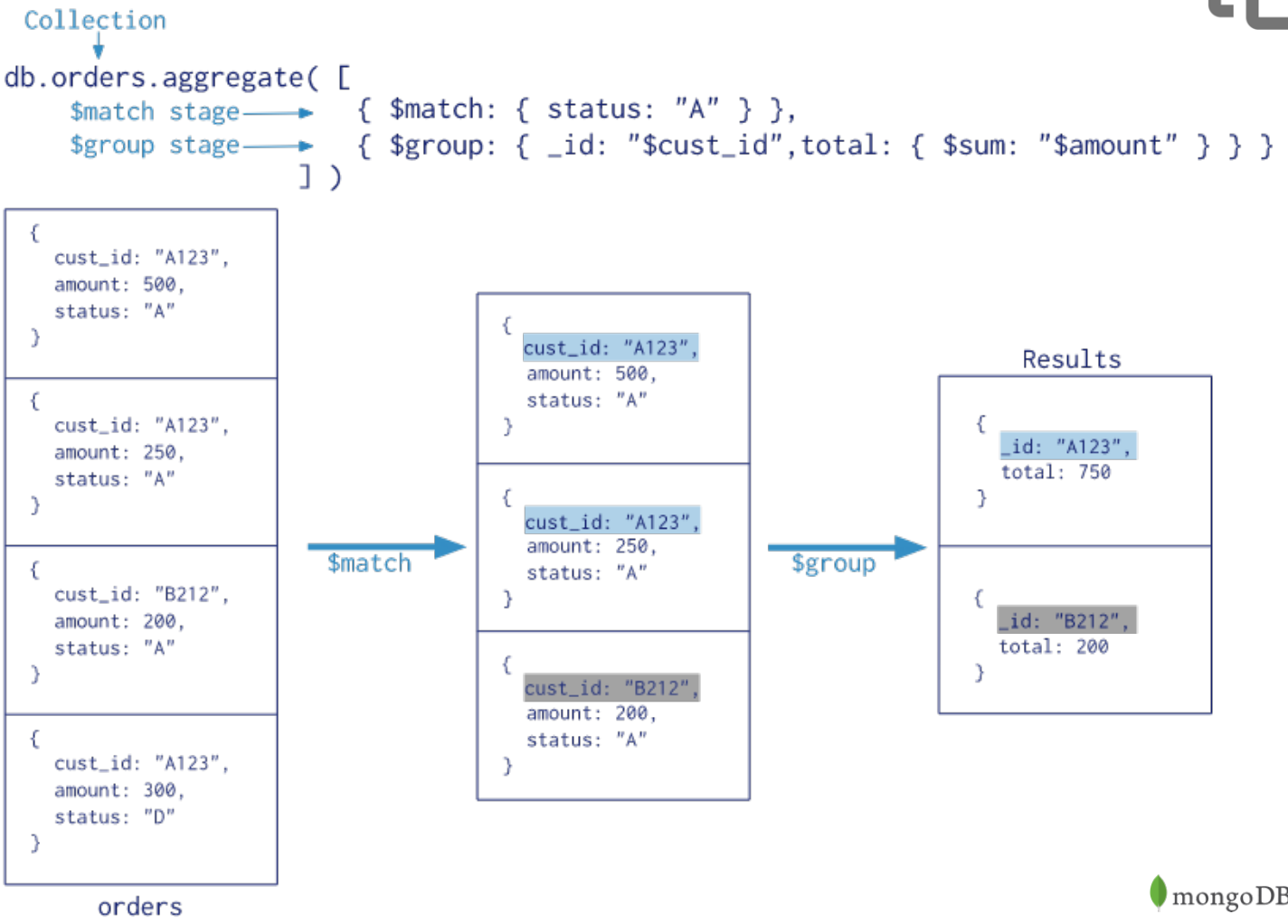
多种查询方式 – 秒杀Hbase、Redis

点查 - 范围 - 地理位置 - 分面搜索 – 聚合 - 关联 – 图

聚合框架

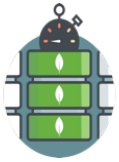
自带强大数据处理能力，用于数据转型，统计、分析等场景

- 多阶段支持
- 类似于Unix管道
- 丰富的表达能力



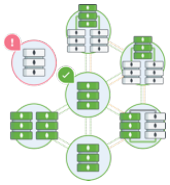
mongoDB.

2. 智能分布数据 - 到你想要的地方



高可用

Built-in **multi-geography** high availability, replication & automated failover



横向扩展

Elastic horizontal scalability - add/remove capacity dynamically **without downtime**

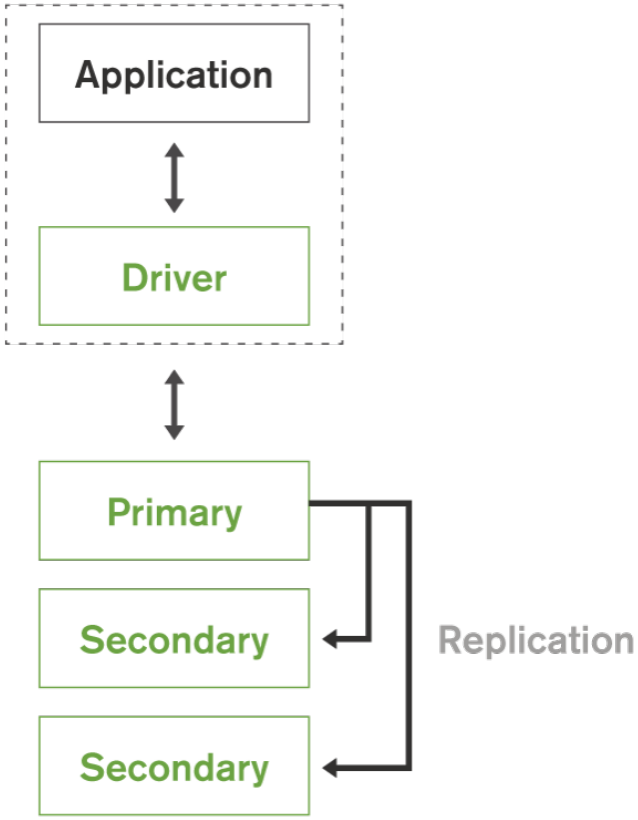
读写分离

Ability to run both **operational & analytics workloads** on same cluster, for timely insight and lower cost

本地读写

Declare **data locality rules** for governance (e.g. data sovereignty), tiers of service & local low latency access

MongoDB 原生复制集 – 99.999%高可用

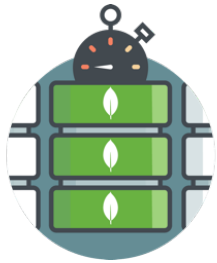


Replica Set – 2 to 50 个成员

自恢复

多中心容灾能力

滚动服务 – 最小化服务终端



智能分布: 全球部署降低访问延迟

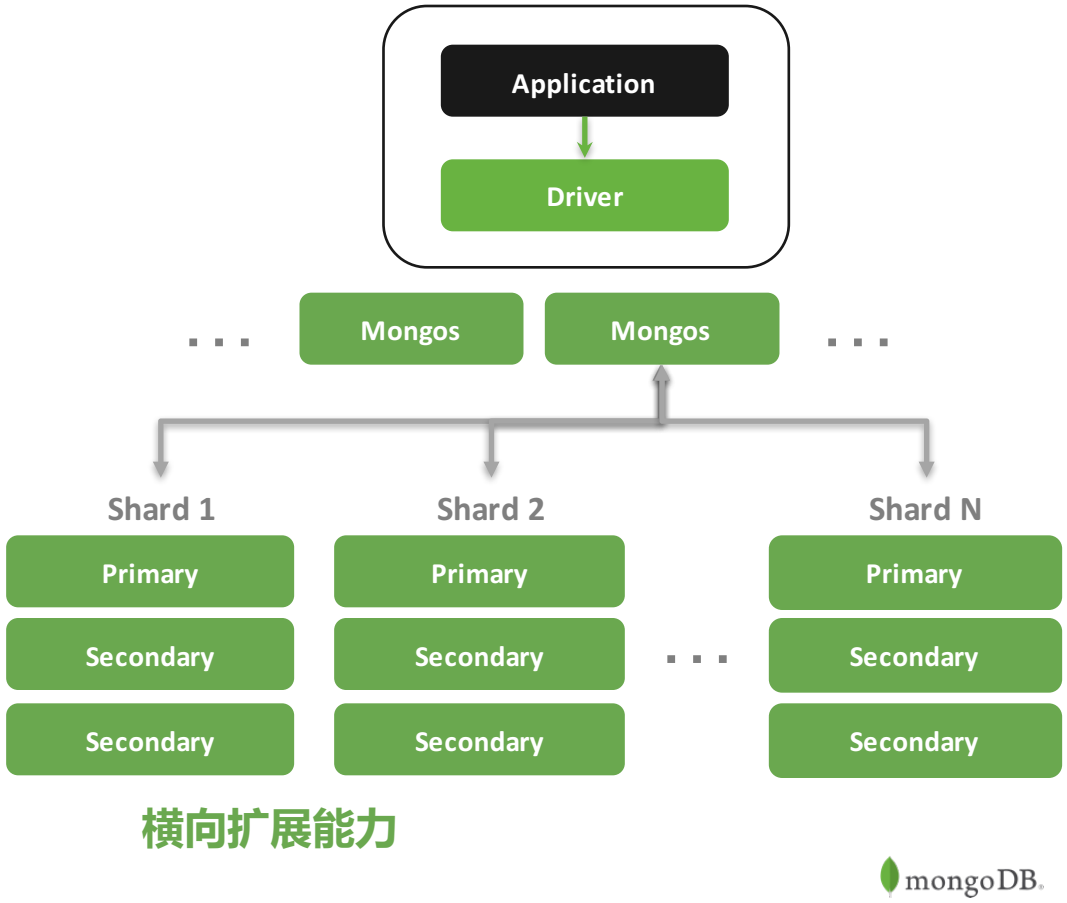
本地读取及写入

- Zone将数据置于不同的节点，为本地访问创建一个副本，让您的应用从本地快速获取数据
  - 每个区域彼此独立的扩展
  - 数据可以根据需要在区中均衡
  - 分区可按地理、设备和应用多种方式

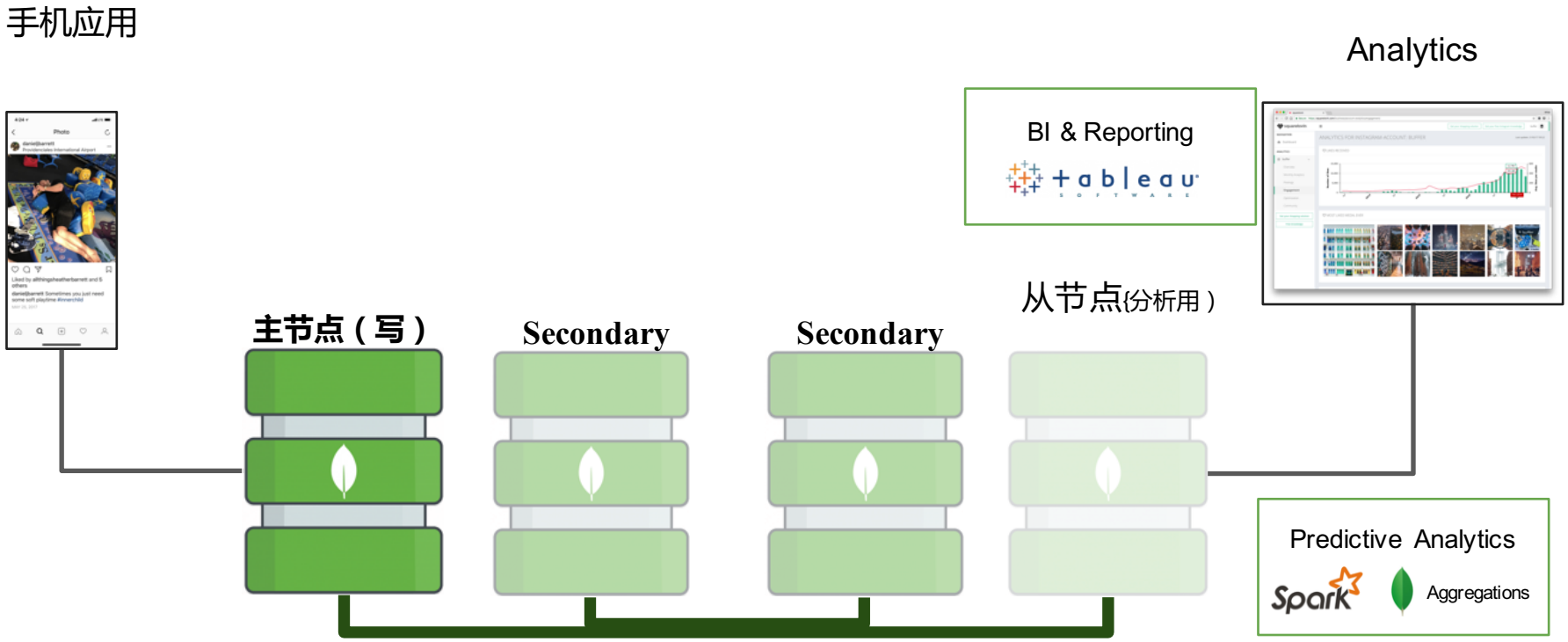
智能分布数据: 横向扩展

自动分片

- 需要的时候无缝扩展
- 应用全透明
- 多种数据分布策略
- 轻松支持TB – PB 数量级



智能分布数据：读写分离，不再ETL



3. 自由的部署，在不同的地点，不同的环境

- 无论何种部署，完全一致的数据库和技术
- 混合云部署优势
- 全球部署
- 防止单一供应商锁定

结语

03

新功能

- 事务支持
- 数据变更流

ACID事务支持（4.0）

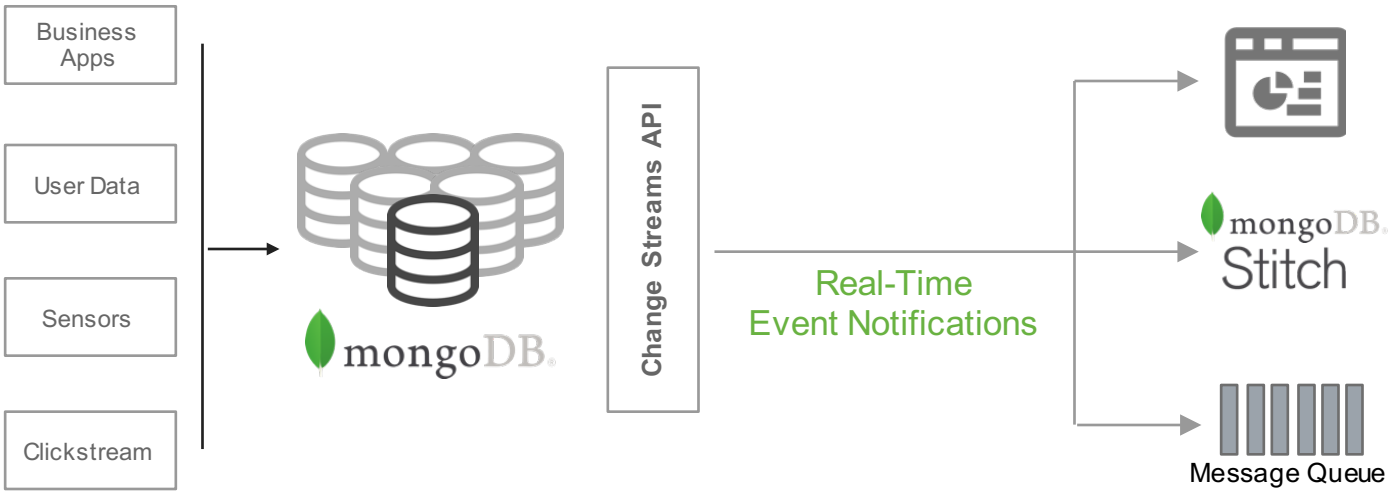
```
with client.start_session() as s:
    s.start_transaction()
    try:
        collection.insert_one(doc1, session=s)
        collection.insert_one(doc2, session=s)
        s.commit_transaction()
    except Exception:
        s.abort_transaction()
```

以熟悉的方式来处理事务

- All or nothing 的保证
- 原生驱动程序支持
- 关系型数据库类似语法
- 使用简单

*Python. Syntax is subject to change

MongoDB变更流 (Change Streams)



类似于Trigger， 快速构建***响应式，实时处理***服务能力

结语

04

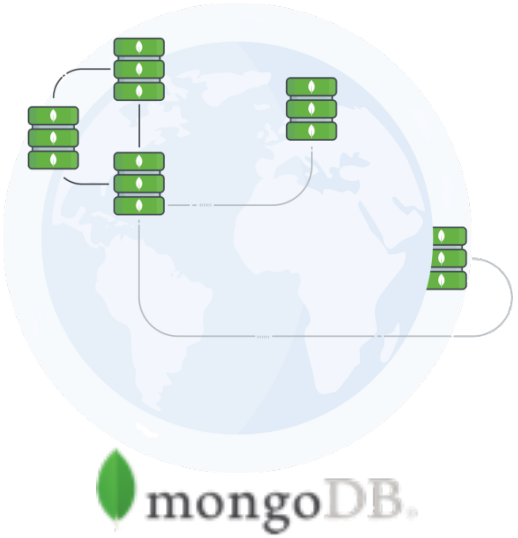
结语

- 参考架构
- 联络我们

MongoDB IODP的独特能力

ACID事务支持
of relational databases

文档模型
最佳的数据管理方式



自由的运行于任何地方

分布式数据库系统


BJ
SH
SZ
HK
China unicom中国联通
太平洋保险 CPIC
CATHAY PACIFIC
中国移动 China Mobile
FOXCONN
PING AN
HUAWEI