MongoDB IODP操作数据平台及4.0 新功能

ABOUT ME

TJ Tang 唐建法

Mongoing中文社区主席

MongoDB大中华区首席架构师

AGENDA

- 关于MongoDB
- MongoDB 4.0新功能概览
- 超越服务器计算: Serverless & Mobile

从风帆到风筝的变迁 – Hood River, Oregon

15年前

今天

风帆

滑行速度：较慢

携带及组装：笨重、复杂

改变方向：困难

风筝冲浪

滑行速度：快，世界纪录101公里

携带及组装： 一个高尔夫球包

改变方向：容易

关系型

80年代

严谨的数据结构

单机式架构

MongoDB

21世纪

灵活的数据结构

分布式架构

MongoDB – 非关系型数据库之王者

40+ Million Unique Downloads

1,000,000+ Online Education Students

50,000+ MongoDB User Group Members

5th Most Popular Database *

29 Offices around the World

1000+ MongoDB, Inc. Employees

6,000+ Paying Customers

50+ Fortune 100 Customers

nent systems according to their popularity.

智能数据操作平台

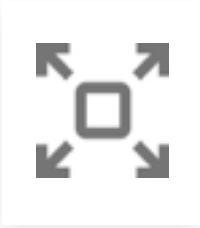
最佳的数据管理方式

智能将数据放在需要的地方

自由的在任何地方运行

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

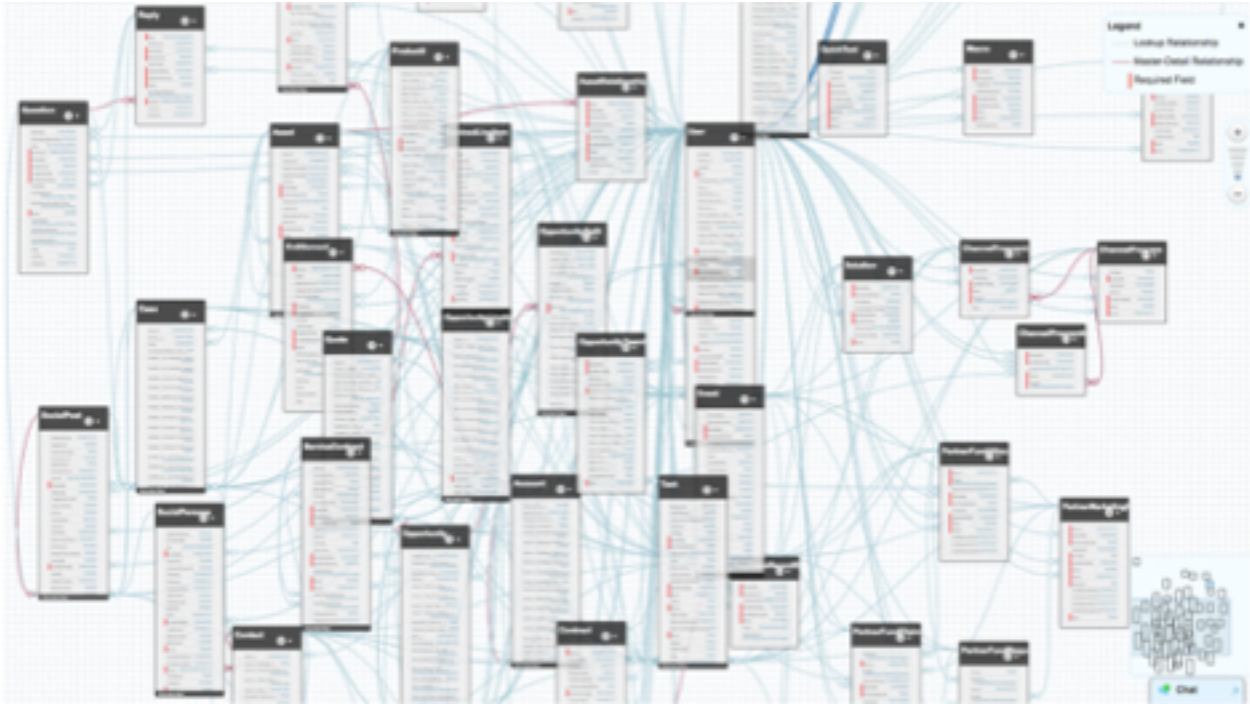


灵活: 弹性模式从容响应需求的频繁变化

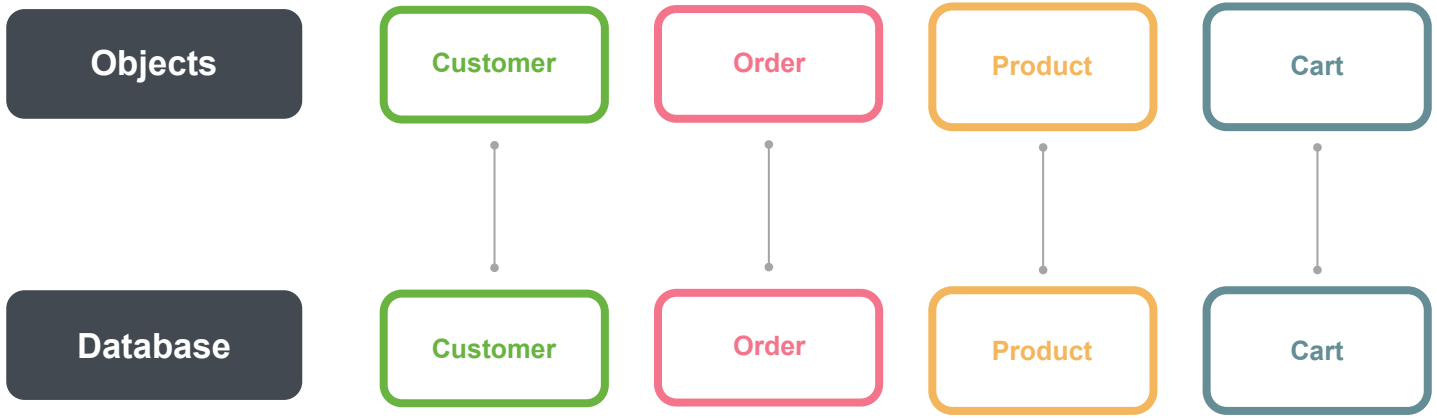
快速开发: 做更多的事，写更少的代码

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

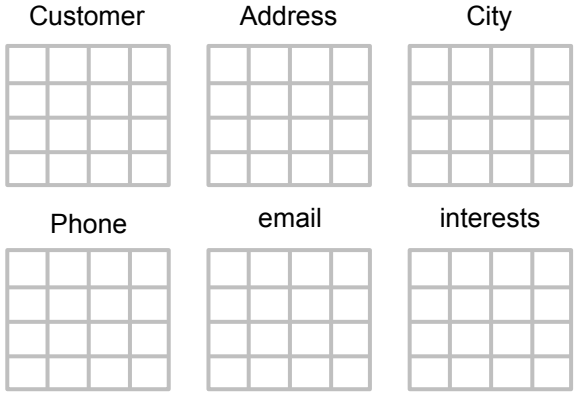

Objects
Customer
Order
Product
Cart
Database
Customer
Order
Product
Cart

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"
```

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) "
                                  )s, %(city)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
```

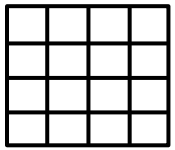
快速: MongoDB 只需要两行代码

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

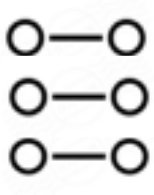
Code snippet: https://gist.github.com/am-MongoDB/e4f196e85f1946ed66480bce6616a94e

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

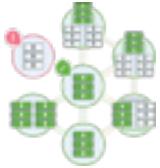
多种查询方式 – 秒杀Hbase、Redis、Cassandra

点查 - 范围 - 地理位置 - 分面搜索 – 聚合 - 关联 – 图

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**

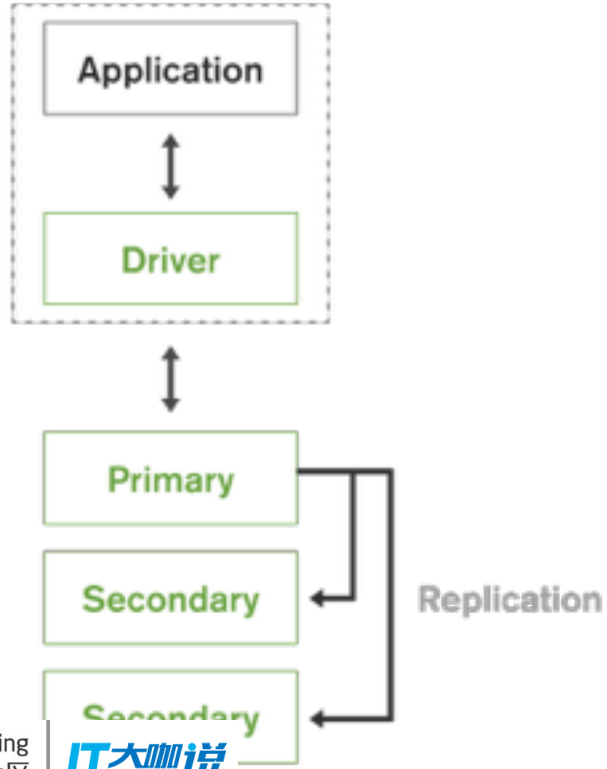
读写分离

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

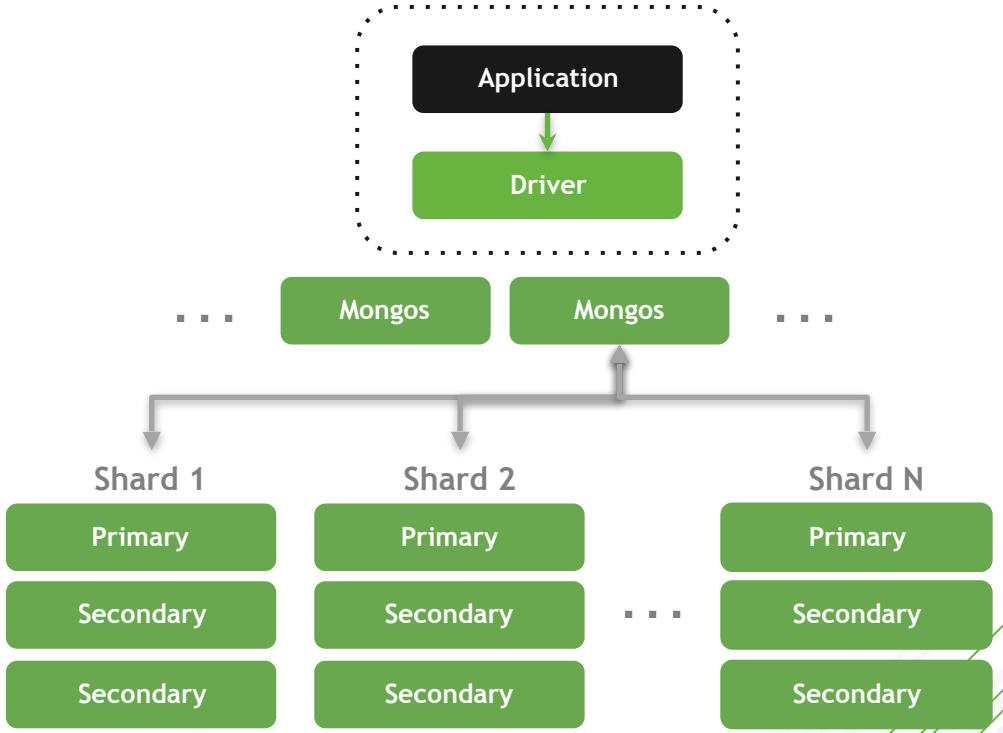
本地读取及写入

- 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

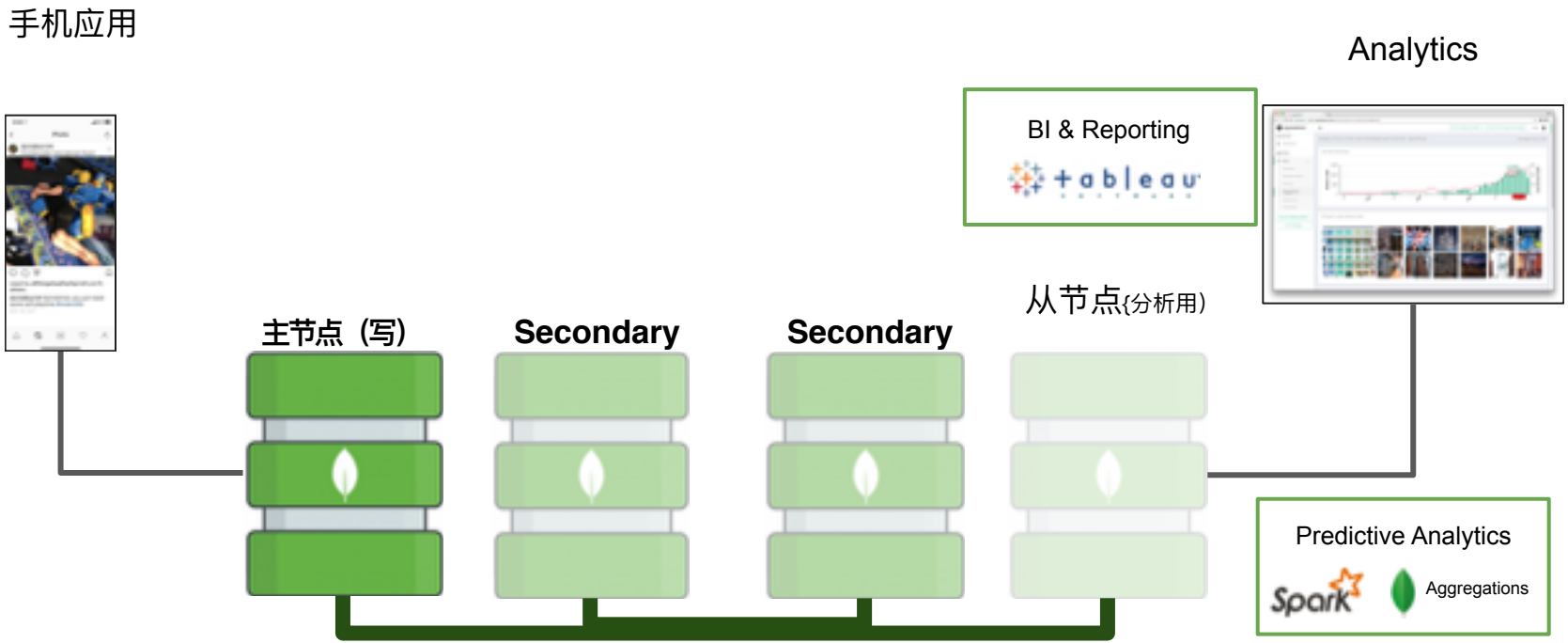
自动分片

- 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



横向扩展能力

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

Atlas: 来自MongoDB 官方的云托管服务

aws

Azure

Google Cloud Platform

Database as a service

全网统一的数据库

Consistent experience across AWS, Azure, and GCP

全球可用

Deploy in over 50 regions worldwide

Create globally distributed databases with a few clicks

多云策略

Exploit the benefits of AWS, Azure, or GCP services on your data

防止供应商锁定

Easily migrate data between cloud providers

MongoDB 4.0 新功能

mongoDB 4.0

智能数据操作平台

最佳的数据管理方式

- 多文档ACID 事务
- 类型转换支持
- 图形化Aggregation构造器
- MongoDB Charts

智能地将数据放在需要的地方

- 数据性能提升40%
- 从节点非阻塞读
- SHA-2 认证支持

自由的在任何地方运行

- Atlas全球集群
- Atlas企业版安全和合规
- 免费云监控
- Kubernetes & OpenShift集成

mongoDB 4.0

智能数据操作平台

最佳的数据管理方式

智能地将数据放在需要的地方

自由的在任何地方运行

超越服务器计算

MongoDB Stitch | MongoDB Mobile

MongoDB: 已经被各个行业应用在各种业务场景中

Single View

eCommerce

Internet of Things

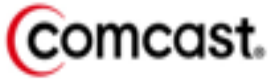
Digital Transformation

Mobile

Travel Graph &
Recommendation System

Product Catalog

Drug Sequencing



Database as a Service

Analytics

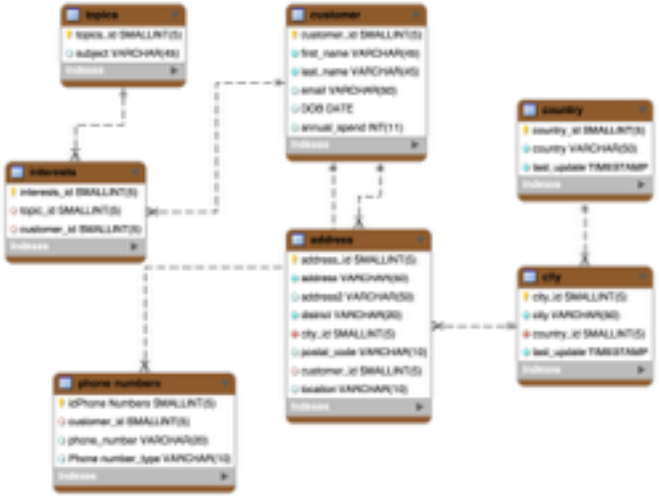
Artificial Intelligence

Gaming

...尽管没有多文档事务支持

数据模型和事务

不同的数据采用不同的方式



表结构(关系型) 数据库

相关的数据被放到不同的多个表中，
跨记录的事务是必须的

```
  _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"
```

文档数据库

相关的数据通常在一个单独的文档中，
事务在单文档层次

数据模型和事务

对很多应用而言，单文档事务支持，

就可以满足业务的数据一致性需求

Document Database

需要多文档事务的场景举例

金融

跨账户的存取款

零售

插入订单，减少库存

电信

插入一个话单，更新计费信息

MongoDB 多文档事务支持

很像关系性数据库的事务

- 相似的语法
- 任何应用都可使用
- 支持一个或多个集合中的多个文档

ACID一致性保证

- 要么全成功，要么全失败
- 对不要求事务的操作无性能影响

发布计划

- MongoDB 4.0: 支持复制集环境
- MongoDB 4.2: 支持分片环境

MongoDB 4.0开始支持多文档的事务

针对多文档，MongoDB 4.0开始支持事务，其使用方式和传统数据库一致，使用非常简单

```
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()
```

和关系数据库的事务操作一样

- 语法基本一样
- 任何应用都可以使用
- 一个表或多个表中的多个文档都适用

ACID 保证

- 要么全成功，要么全失败

版本要求

- MongoDB 4.0, 支持复制集，已经发布
- MongoDB 4.2: 支持分片，预计19年发布

和传统数据库类似的形式

```
db.start_transaction()
cursor.execute(orderInsert, orderData)
cursor.execute(stockUpdate, stockData)
db.commit()
```

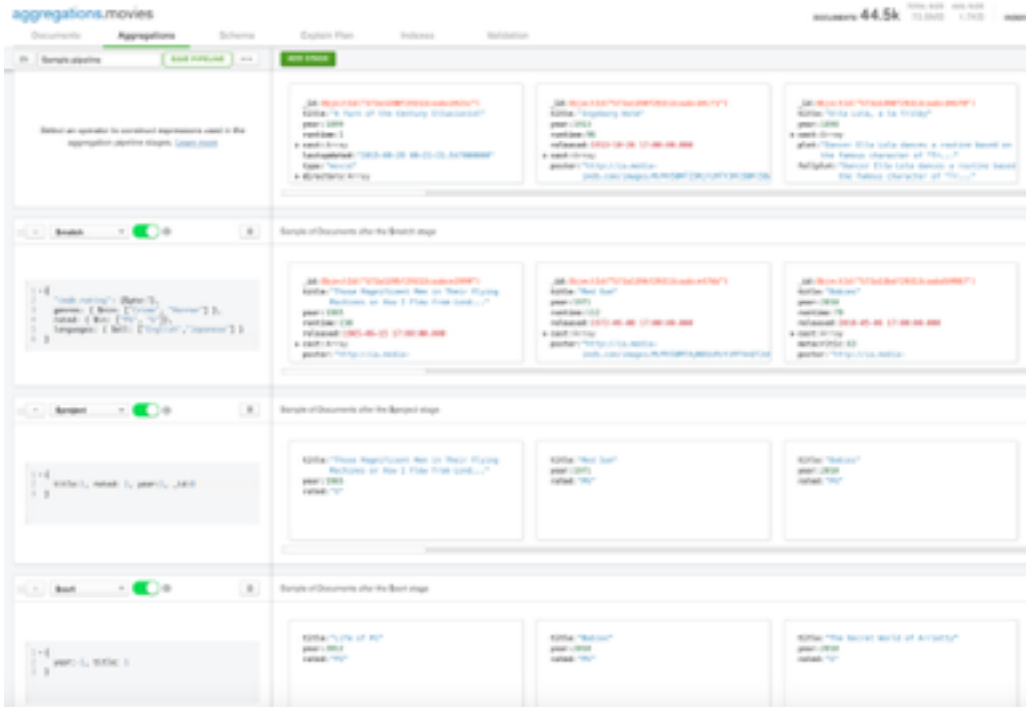

```
s.start_transaction()
orders.insert_one(order, session=s)
stock.update_one(item, stockUpdate,
s.commit_transaction()
```

*Python. Syntax is subject to change

MongoDB Compass

- MongoDB Compass图形化工具:
  - **可视化的展现和浏览** 你的数据结构
  - **插入和修改**文档
  - 构建查询，查看**执行计划和索引**
  - 控制数据模型，**构建数据模型规则**
- Compass社区版支持社区版MongoDB服务器
- Compass企业版支持 :
  - 高级安全功能
  - 数据模型分析
  - 实时性能面板
  - 构建模型规则

聚合管道构建器



- 构建聚合操作更加简单直观
- 代码提示和自动补全
- 自动管道执行结果预览
- 拖拉改变管道执行顺序
- 支持保存和导入聚合操作语句

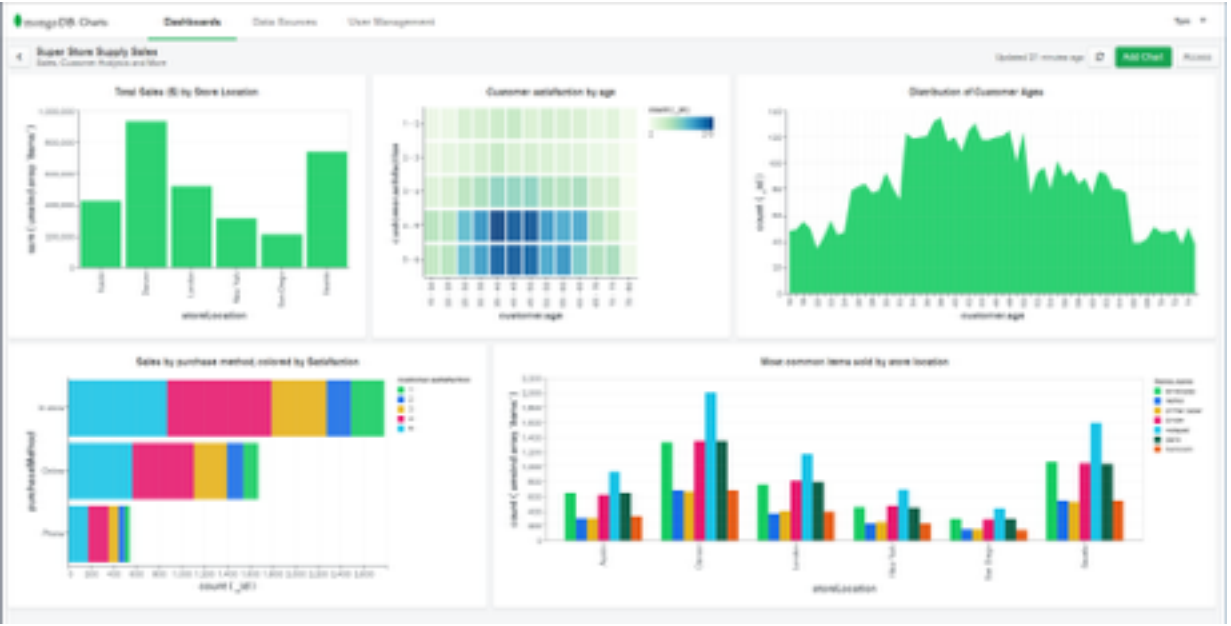
Introduced in MongoDB Compass 1.14

支持直接生成语言代码 (Beta版)

- 查询语句构建：支持导出代码
- 支持复杂的多参数查询和聚合管道
- 开始支持： C#/.NET, Java, JavaScript, Python, and mongo shell 语法
  - 未来将支持更多语言

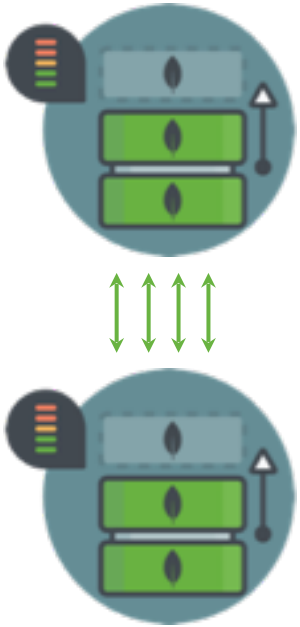
Introduced in MongoDB Compass 1.15

MongoDB Charts (Beta)



- 最简单最快速的MongoDB数据展现方法
- 支持MongoDB的数据模型
- 支持从自有环境到云的各种MongoDB环境

灵活扩展：数据迁移性能提升40%



- 分片数据均衡器负责维护集群中数据的均衡分布
- 4.0版本数据均衡迁移的速度将有40%的提升
- 增加和减少节点的速度将更快
- 帮助客户更经济的快速的应对容量和压力需求变化

Atlas

实现 **敏捷，** 并

降低成本

Self-service and elastic

Global and highly available

Secure by default

Comprehensive monitoring

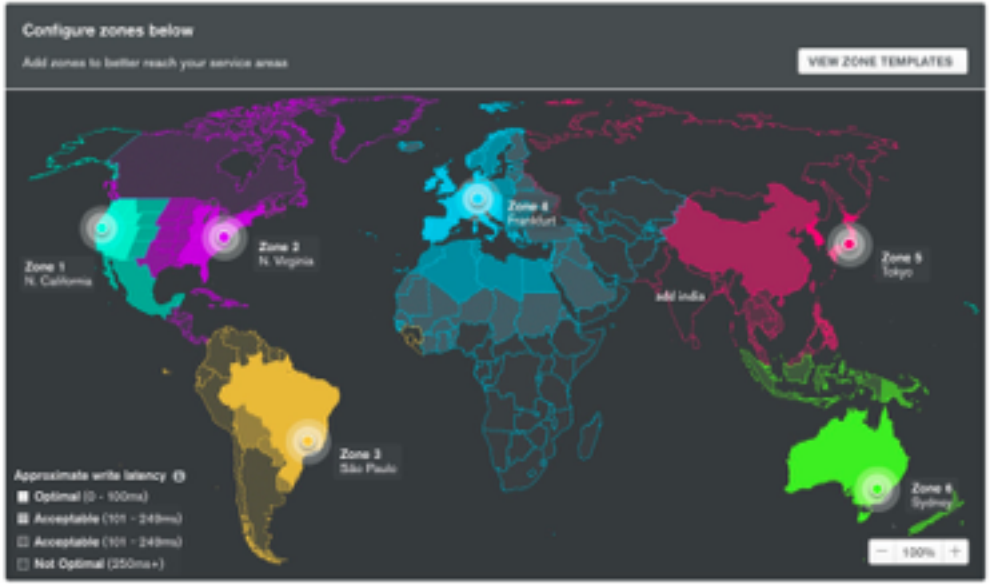
Managed backup

Cloud agnostic

Atlas全球集群

Atlas将帮助您将数据自动部署到全球的每个角落

- 为您的分布式应用，提供毫秒级的数据本地读写
- 确保指定区域的数据保存在指定的区域中
- 可以快速的用预定义模板部署，或者通过图形化的接口自己来选择云区域部署



MongoDB 监控云服务

免费服务，无需安装Agent, 查看健康和性能状态

超越服务器计算

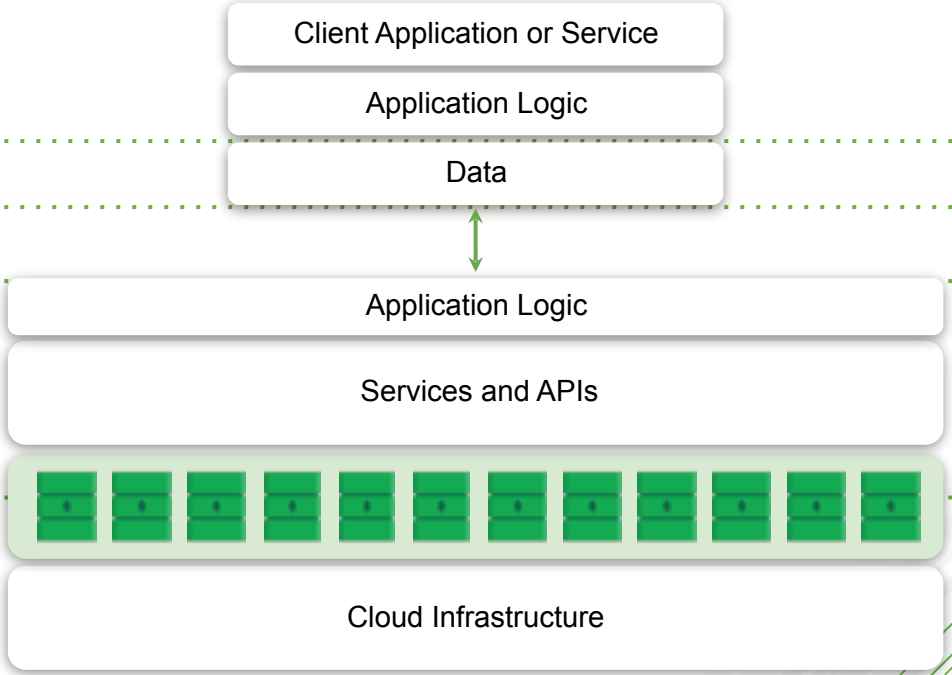
MongoDB提供更广泛的智能数据操作平台为您加速

利用智能操作平台，**生产力将提升3到5倍**

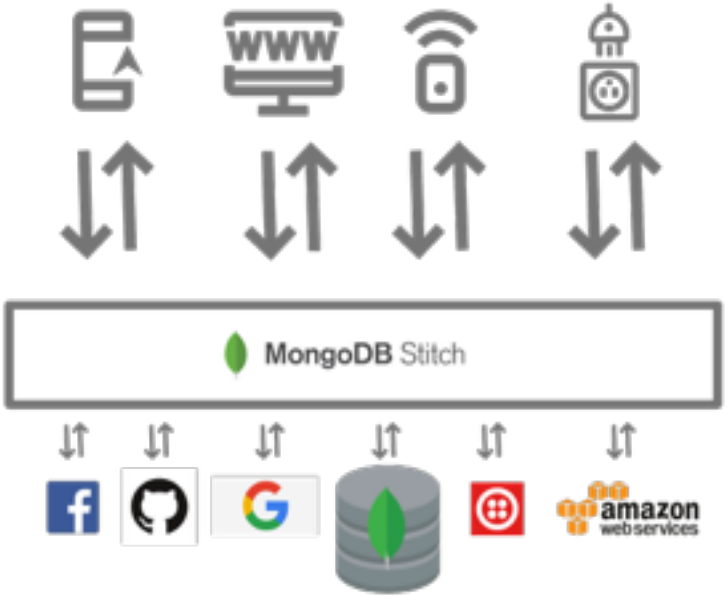
MongoDB Mobile 您设备中的MongoDB

MongoDB Stitch
无服务器平台帮助开发人员专注与创新，而非重复构建和集成服务和重复编写基础代码

MongoDB Atlas
快速部署，动态扩展，全球跨区域和云提供商部署



MongoDB Stitch 无服务器（Serverless）架构平台



无需编写上千行代码，无需管理服务器，为应用开发提供简单、安全的数据和服务

让你更快的发布应用，同时免于维护之苦

MongoDB Stitch 无服务器平台 – 服务

Stitch QueryAnywhere

将 MongoDB强大的查询能力直接带到前端

iOS, Android, Web, IoT

Stitch Functions

集成微服务+服务器端逻辑+云服务

通过定制API，构建完整应用、或者数据及服务

Stitch Triggers

实时让应用对数据库变化做出反应

(即将发布)

Stitch Mobile Sync

自动同步手机和后台数据库的数据

(即将发布)

Streamlines app development with simple, secure access to data and services from the client with thousands of lines less code to write and no infrastructure to manage – getting your apps to market faster while reducing operational costs.

聚焦你能创建的核心价值的地方

业务逻辑代码

服务集成和数据访问控制

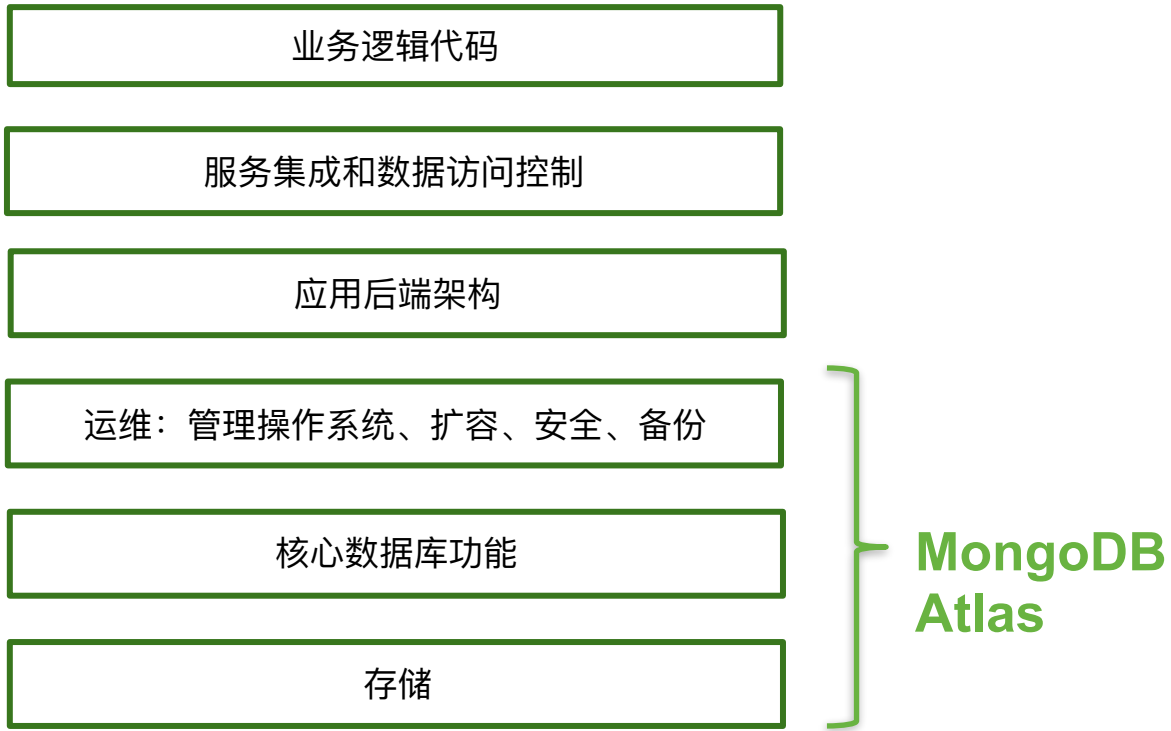
应用后端架构

运维：管理操作系统、扩容、安全、备份

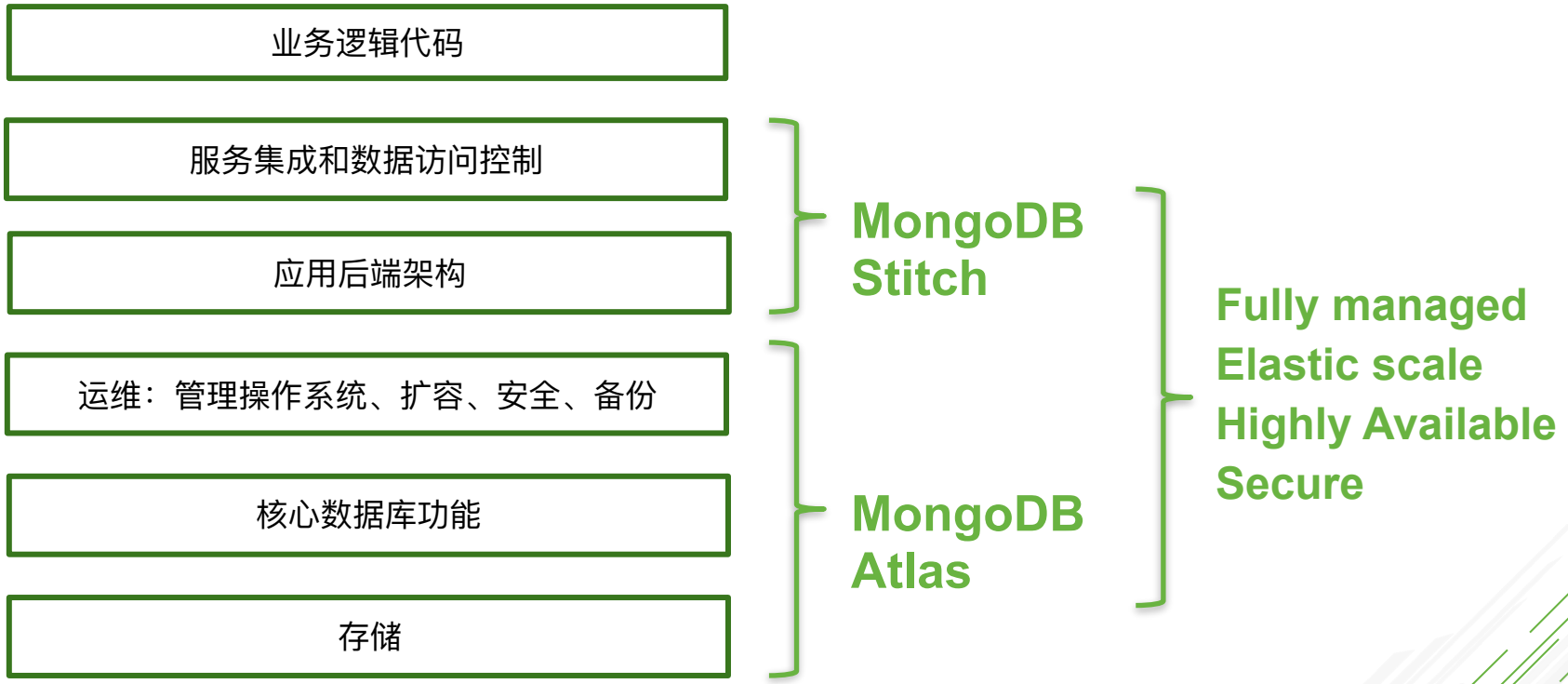
核心数据库功能

存储

聚焦你能创建的核心价值的地方



聚焦你能创建的核心价值的地方



聚焦你能创建的核心价值的地方

New Online Course
M040: New Features and Tools in MongoDB 4.0

- 免费▪ 在线 ▪ 3周课程
- 现在开放注册 - https://university.mongodb.com/courses/M040/about

开始行动

- 下载白皮书 Guide to What's New
- 查看产品说明 4.0 Release Notes
- 部署使用 MongoDB Atlas

Q&A