

第 1 章

MongoDB 简介

MongoDB 是目前非常流行的文档数据库之一，它既能够作为完全托管的云服务来使用，也可以部署到私有服务器上自行维护。它的出现使数据的存储、获取和分析领域带来了里程碑式的变革。对于当今的互联网从业者来说，了解并能够灵活使用 MongoDB 非常重要。本章将详细介绍 MongoDB 数据库相关的基础知识，使读者对该数据库有一个全面的了解，并且能够快速认识 MongoDB 的特点以及与其他数据库的区别。

本章主要涉及的知识点包括：

- 什么是 MongoDB
- MongoDB 简史
- MongoDB 的特性
- MongoDB 的应用场景
- MongoDB 涉及的概念
- MongoDB 的数据类型

1.1 MongoDB 概述

随着大数据时代的到来，数据呈爆炸式增长，这使得传统的关系数据库面临越来越大的挑战。高性能、可扩展的数据库变得越来越重要。在这样的场景下，非关系数据库应运而生，并且越来越受欢迎。MongoDB 便是非关系数据库的典型代表。MongoDB 是一个介于关系数据库与非关系数据库之间的产品，是非关系数据库中功能最丰富、最像关系数据库的 NoSQL 数据库。

MongoDB 是一款为 Web 应用和互联网基础设施设计的数据库管理系统。它使用 C++ 语言编写，是一个可扩展、开源、表结构自由的数据库，旨在为 Web 应用提供可扩展的高性能数据存储解决方案。它支持的数据格式是 BSON，一种类似于 JSON 的二进制形式的存储格式，即 Binary JSON，这种格式与 JSON 一样能够支持内嵌的文档对象和数组对象。

MongoDB 的官网上写着这样一句话：“Loved by developers, trusted by enterprises（被开发者所钟爱，被企业所信任）”。MongoDB 凭借其出色的数据存储功能，已成为当前非关系数据库领域的佼佼者。根据 DB-Engines 网站目前最新的统计（2024 年 11 月），前 20 个数据库各项指标的排名如图 1.1 所示。可以看到，MongoDB 在数据库领域总排名第 5，仅次于 Oracle、MySQL 等关系数据库，在 NoSQL 数据库领域排名首位。

423 systems in ranking, December 2024									
Rank			DBMS	Database Model	Score				
Dec 2024	Nov 2024	Dec 2023			Dec 2024	Nov 2024	Dec 2023		
1.	1.	1.	Oracle +	Relational, Multi-model	1263.79	-53.22	+6.38		
2.	2.	2.	MySQL +	Relational, Multi-model	1003.76	-14.04	-122.88		
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	805.69	+5.88	-98.14		
4.	4.	4.	PostgreSQL +	Relational, Multi-model	666.37	+12.04	+15.47		
5.	5.	5.	MongoDB +	Document, Multi-model	400.39	-0.54	-18.76		
6.	6.	6.	Redis +	Key-value, Multi-model	150.27	+1.63	-8.08		
7.	7.	↑10.	Snowflake +	Relational	147.36	+4.87	+27.48		
8.	8.	↓7.	Elasticsearch	Multi-model	132.32	+0.68	-5.43		
9.	9.	↓8.	IBM Db2	Relational, Multi-model	122.78	+1.04	-11.81		
10.	10.	↑11.	SQLite	Relational	101.72	+2.24	-16.23		
11.	11.	↑12.	Apache Cassandra +	Wide column, Multi-model	97.94	+0.22	-14.26		
12.	12.	↓9.	Microsoft Access	Relational	90.82	-0.49	-30.93		
13.	↑14.	↑17.	Databricks +	Multi-model	87.68	+1.22	+7.37		
14.	↓13.	14.	Splunk	Search engine	85.36	-3.11	-10.94		
15.	15.	↓13.	MariaDB +	Relational, Multi-model	83.77	+1.07	-16.66		
16.	16.	↓15.	Microsoft Azure SQL Database	Relational, Multi-model	76.37	-0.16	-6.67		
17.	17.	↓16.	Amazon DynamoDB +	Multi-model	72.73	+0.33	-9.68		
18.	18.	18.	Apache Hive	Relational	53.09	+1.59	-16.31		
19.	19.	19.	Google BigQuery +	Relational	52.29	+1.71	-9.88		
20.	↑21.	↑22.	Neo4j +	Graph	43.07	+0.37	-6.92		

图 1.1 DBMS Ranking

综合 MongoDB 官网提供的信息、权威机构的评价以及开发者总结的博客，可以大概整理出 MongoDB 属性的相关信息，如表 1.1 所示。表 1.1 中描述的概念或属性会在后面的章节中进行讲解。

表 1.1 MongoDB系统属性

参数名称	描 述
首选数据模型	文档存储
其他数据模型	空间存储、搜索引擎、时间序列存储
DB-Engines 排名	总排名第 5、文档存储型第一
官网地址	www.mongodb.com
首次发行版本时间	2009
当前最新版本	2023 年 3 月发行的 6.0.5 版本
开源与否	开源
是否支持独立服务部署	支持
DBaaS（数据库即服务）	MongoDB Atlas、ScaleGrid for MongoDB Database
编写语言	C++
支持的平台	Linux、OS X、Solaris、Windows
数据模式	自由模式
标准的数据类型	string, integer, double, decimal, boolean, date, object_id, geospatial

(续表)

参数名称	描 述
第二索引	支持
SQL 支持	通过 MongoDB Atlas SQL 接口进行只读 SQL 查询
API 以及其他接口方法	GraphQL、HTTP REST、Prisma
支持的编程语言	市面上主流的编程语言基本都支持
服务端脚本语言	JavaScript
触发器	仅在 MongoDB Atlas 中支持
分区方法	Sharding
复制集方法	可通过 MongoDB Atlas 实现
MapReduce	支持
分布式中的数据一致性方案	默认为即时一致性，可配置最终一致性
事务	多文档事务
并发	支持
数据持久化	支持
内存存储	支持
外键限制	无
用户权限控制	支持

此外，MongoDB 作为开源项目，对开发者的支持非常友好。MongoDB 官方提供了众多学习及解决问题的途径，包括官方的 API 文档、开发者中心、社区以及 MongoDB 学校。这些途径使得开发者在资料搜集以及问题解决方面节省了非常多的时间，并且能够得到非常权威的解答。MongoDB 官方提供的资源地址如表 1.2 所示。

表1.2 MongoDB官方资源地址

名 称	说 明	地 址	描 述
Documation	开发文档	https://docs.mongodb.com/manual/	网站式的开发文档、操作手册，包含 MongoDB 介绍、安装使用以及各项功能说明，支持搜索查询
Developer Center	开发者中心	https://www.mongodb.com/developer/	开发者中心，汇聚各种开发语言的教程，支持按语言分类搜索，支持常见问题解答
MongoDB University	MongoDB 学校	https://learn.mongodb.com	MongoDB 学校，可以根据不同重点内容进行学习，支持按语言搜索，包含 MongoDB 各方面的课程
Developer Community	开发者社区	https://www.mongodb.com/community/forums	开发者社区，这里汇聚了使用各种开发语言的开发者，涵盖在使用 MongoDB 时各种各样的问题及解决方法，支持按照类别或标签进行搜索

MongoDB 还公布了有史以来所有 BUG 的报告，使用者可以跟踪 BUG 列表查看所有的问题记录，有针对性地选择适用自己的版本，避开缺陷。

不仅如此，MongoDB 还为完善及开发 MongoDB 数据库服务端的开发者提供了交流论坛：

<https://community.mongodb.com/c/server-dev>。

从目前的市场占有率来看，MongoDB 的应用已经渗透到各个领域。以下是一些主要的应用场景。

- 内容管理：MongoDB 可以用于存储网站的文章、博客、图片、视频等内容。例如，一个新闻网站可以使用 MongoDB 来存储新闻文章、评论等内容，每篇文章及其相关的内容作为一个文档，这样可以轻松地扩展以处理大量的新闻和媒体数据。
- 社交网络：MongoDB 可以用于存储社交网络用户的信息、帖子、评论等数据。例如，一个社交媒体平台可以使用 MongoDB 存储用户发表的帖子、评论等数据，由于 MongoDB 支持嵌套文档，可以将用户的活动流存储在其个人资料文档中，一次查询即得所有，使得实时推送和社交互动更高效。
- 物流信息：使用 MongoDB 存储订单信息，在订单状态转变的过程中，同时存储订单的流转信息，这样使用时就可以快速获得订单相关的所有信息。
- 物联网：MongoDB 可以用于存储物联网设备的数据，如智能设备信息、传感器数据、位置信息等。例如，IoT（Internet of Things，物联网）设备可以生成大量数据，如温度、湿度、位置、流量等，同时可以存储设备汇报的日志信息，MongoDB 可以将这些数据用于分析，从而达到监控设备状态、执行预测维护以及生成分析报告的目的。
- 游戏：MongoDB 可以用于存储游戏用户的数据，如角色信息、装备信息、积分等。例如，在线游戏需要快速更新和响应用户的操作，MongoDB 可以用于存储用户配置、游戏状态等数据，以实现实时互动。
- 地理空间数据：地图应用程序和位置服务可以使用 MongoDB 来存储地理空间数据，如地点、地区边界和路线。这些数据可以用于位置搜索、导航和地理信息分析。
- 大数据处理：MongoDB 可以用于存储大数据并提供灵活的查询和分析能力，用于分析行业动态、用户习惯等。例如，一家电子商务公司可以使用 MongoDB 来存储销售数据、用户行为和产品信息，通过 MongoDB 的聚合功能可以实时分析销售趋势、用户偏好和库存情况。
- 日志和事件存储：大型服务器集群运行过程中会生成大量的日志和事件数据，MongoDB 可以用于存储和检索这些数据以进行系统监控、性能分析和故障排除。

1.2 MongoDB 的发展历史

MongoDB 自诞生至今已有 16 年的历史。2007 年，10gen 公司在开发一个 PaaS 项目时，设计了一款新的数据库技术。该技术旨在解决传统关系数据库在云计算环境中难以管理的问题，以实现 Web 应用的托管和自动扩展。开发者对这项新的数据库技术表现出极大的兴趣，因此 10gen 公司决定将更多精力投入该数据库的研发中，从而促成了 MongoDB 的诞生。经过一段时间的研发和测试，MongoDB 于 2009 年正式发布，并推出了 MongoDB 1.0 版本。此后，MongoDB 进入了一个快速发展的阶段。在最初的短短 6 年时间里，MongoDB 取得了其他产品可能需要 10 年甚至 20 年才能达到的发展成果。

在接下来的几年中，MongoDB 不断推出新的版本，逐步增加了许多功能和特性。2010 年 8 月，

MongoDB 发布了 1.6 版本，引入了一些主要特性，如用于水平伸缩的分片、具备自动故障转移能力的副本集，以及对 IPv6 的支持。2012 年 5 月 23 日，MongoDB 发布了 2.1 版本，采用全新的架构，提供了诸多增强特性。

2012 年 8 月，MongoDB 发布了 2.2 版本，引入了聚合管道，可以将多个数据处理步骤组合成一个操作链。

2013 年，MongoDB 推出第一款商业版本 MongoDB Enterprise Advanced。这一版本的推出不仅进一步巩固了 MongoDB 在开源数据库市场中的地位，还标志着 MongoDB 正式进入商业数据库市场。

此后，MongoDB 不断推出新的版本，逐步完善数据库的功能和性能。发展至今，MongoDB 已经迭代了 7 个大版本，几百个小版本，版本迭代时间线如图 1.2 所示。

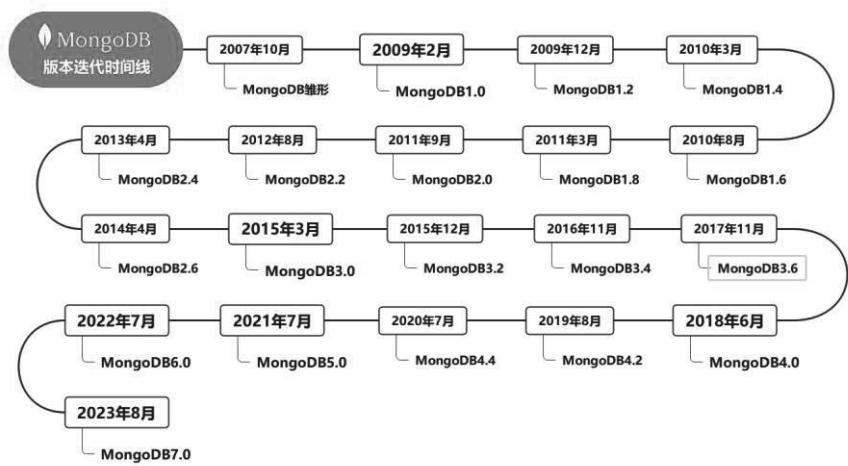


图 1.2 MongoDB 版本迭代时间线

从设计之初，MongoDB 便以开源形式向外公开，这也是越来越多的人选择 MongoDB 并将其用于实际生产环境的主要原因。MongoDB 源码在 GitHub 的地址为 <https://github.com/mongodb/mongo>。在 GitHub 上，目前我们依然能够看到创始人之一 Dwight 在 2007 年 10 月第一次提交的 MongoDB 代码，如图 1.3 所示。

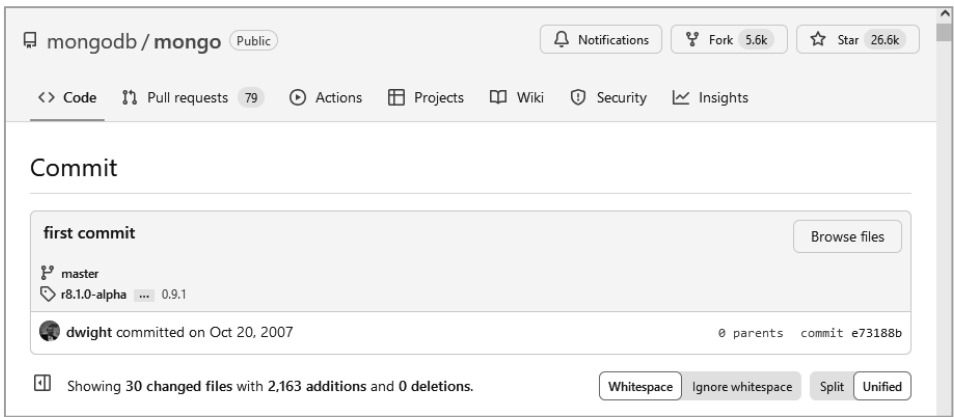


图 1.3 MongoDB 的开源代码

在该开源项目上，我们可以看到 MongoDB 多达 128 个分支，每个分支对应特定的版本或功能特性。目前，一直在维护的分支对应版本 4.2、4.4、5.0、6.0、7.0、7.1 以及其他一些特定功能的分支，如图 1.4 所示。

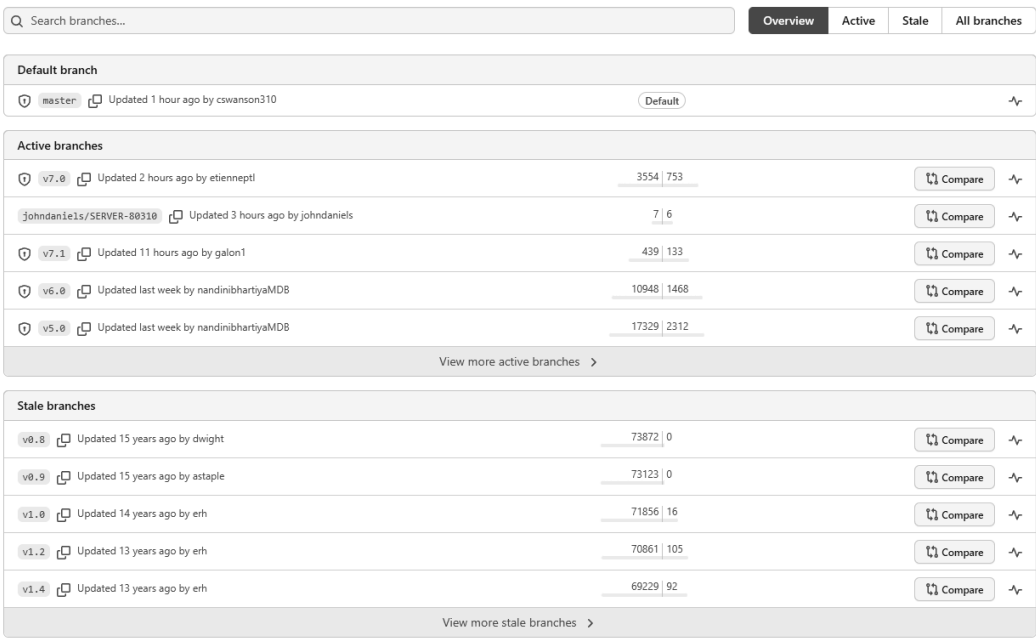


图 1.4 MongoDB 分支概览

目前还在维护的部分版本特性如表 1.3 所示。

表1.3 MongoDB部分版本特性

版本名称	新特性描述
4.2	引入分布式事务，增强在集群环境下的数据操作，数据安全性和可扩展性的改进，以及一些新的功能和改进，全面提高了部署规模及数据安全性
4.4	主要包括兼容性改进、性能改进、数据可用性改进等，可以帮助开发者实现更快、更加高效的性能。例如，支持插入和更新操作符的增强功能，支持使用正则表达式进行查询，改变了备库获取增量更新的方式，增加了对身份验证和访问控制的更多选项和支持，增强了复制集的稳定性和可用性等
5.0	主要支持原生时间序列数据的全生命周期管理，涵盖从数据采集、存储、查询、实时分析到可视化的各个环节，甚至包括在线归档和数据老化后的自动失效处理。此外，系统还增加了许多新功能，例如对时间序列数据的压缩存储，以及对实时聚合和索引的支持等。这些功能进一步扩展了其在物联网、金融分析、物流等领域的应用场景
6.0	增强的时间序列集合，引入了分片来改善数据分布，引入了列式压缩来改善存储空间占用，还引入了密集化和间隙填充来支持部分数据点缺失情况下的时序分析，提供了更多的索引类型，提供了强大的自动分片功能

(续表)

版本名称	新特性描述
7.0及以上	优化慢查询的日志消息，默认启用并发存储引擎事务，增加复合通配符索引，新增区块迁移分片统计信息以及一些查询、删除与分片方面的改进
8.0	- 性能提升：在性能优化方面取得了显著进展，查询速度提高了 30%以上，吞吐量提高了 36%。内存使用量和查询时间减少，数据复制期间的更新吞吐量提高了 59%，并发写入速度提高了 20%。 - 安全性增强：引入了可查询加密，允许客户在客户端加密敏感数据，并在加密状态下进行查询处理，增强了数据的安全性。 - 可扩展性提升：新的分片功能使得跨分片分发数据的速度提高了 50 倍，入门成本降低了 50%，使得数据库更容易进行水平扩展，并且成本更低。 - 用户体验优化：提供了更好的控制权，优化数据库性能，以应对高峰需求和意外事件，确保应用程序的平稳运行

MongoDB 的版本号命名分为两个阶段：5.0 之前的版本和 5.0（含）之后的版本。在 5.0 之前的版本，例如 4.4 及更早的版本，MongoDB 采用的是生产/开发版本控制机制。版本号的格式为 X.Y.Z，其中 X.Y 表示发行版本序列号或开发版本序列号，Z 表示补丁号或修订号，即对之前版本中发现的漏洞进行修复的补丁。例如，版本号 4.4.7 中的 4.4 是发行版本序列号，而 7 是该发行版本的修订号。

- 如果 Y 是偶数，则 X.Y 是发行版本序列号。例如，4.0、4.2 是一个发行版本序列号。发行版本通常比较稳定，可用于生产环境。
- 如果 Y 是奇数，则 X.Y 是开发版本序列号。例如，4.1、4.3 是一个开发版本序列号。开发版本应该仅用于测试，不能用于生产环境。

因此，在图 1.2 所示的发布时间线中只能看到 4.0、4.2、4.4 版本。

通常发行版本序列号的改变（如 4.0 变成 4.2）标志着引入了新的特性，这些新特性通常无法向后兼容。补丁（修订号）发布的改变（例如 4.0.12 到 4.0.14）通常标志着 BUG 的修复，并且可以向后兼容。

从 MongoDB 5.0 开始，版本号的前两个数字位（X.Y）表示的意义发生了变化，并且开始分为两个不同的发布系列：主版本（Major Releases）和快速发布版本（Rapid Releases）。除此之外，还有补丁发布版本（Patch Releases）、候选发布版本（Release Candidates）。其中补丁发布版本的版本号与之前的版本号格式相同。

- 主版本：例如 5.0、6.0、7.0、8.0。
- 快速发布版本：例如 5.1、5.2。
- 补丁发布版本：例如 5.0.1、5.2.1。
- 候选发布版本：例如 5.0.0-rc0。

主版本大约每年发布一次，会引入一些新特性和改进。主版本支持 MongoDB Atlas（云平台）和本地部署。目前（到 2025 年年初）MongoDB 最新的稳定版本为 8.0。

快速发布版本大约每个季度发布一次，也会引入一些新特性和改进。快速发布版本仅在 MongoDB Atlas 中支持，不支持本地部署。同时，快速发布版本不适用于部署工具 MongoDB Ops Manager。

补丁发布版本是根据主版本和快速发布版本的需要来发布的，通常包括 BUG 修复和小的改进。例如：

```
5.0.1 (主版本的补丁版本)
5.2.1 (快速发布版本的补丁版本)
```

在新的主版本和快速发布版发布之前，候选发布版本可用于早期测试。候选发布版本代表即将发布的版本，该版本足够稳定，可以用于测试，但不适合生产部署。

例如：

```
5.0.0-rc0
5.0.0-rc1
5.1.2-rc5
```

关于版本部分，MongoDB 官方有一句重要提示，始终升级到所使用的发布系列的最新稳定补丁发布版。也就是说，如果使用的是 4.x 版本，那么始终要保持更新到 4.x 对应的最新的稳定的补丁版本。

1.3 MongoDB 的优缺点

1.3.1 MongoDB 的优点

MongoDB 拥有诸多优点，涉及多个方面。

1) 模式自由

MongoDB 放弃了复杂的关系模型，转而采用对象模型，并采用文档存储形式。这种存储方式易于存储、易于查询且效率高，非常适合敏捷式的快速开发。与关系数据库不同，MongoDB 的文档无须预先定义 Schema。在关系数据库中，每张表都有严格定义的 Schema，规定了列和类型，这种模式使得扩展变得非常困难，有时甚至会因为一小部分数据的变更而需要修改整个表结构。

在 MongoDB 中，同一个集合可以包含具有不同字段（类型）的文档对象，并且支持使用 JSON Schema 来规范数据模式。理论上，集合中的每个文档都可以拥有完全不同的结构。但在实践中，一个集合中的文档通常相对统一。例如，在一个订单集合中，每个文档通常都包含订单号、商品内容、创建时间、购买人、价格等字段。

MongoDB 在保证模式的灵活性和动态性的同时，还提供了强大的数据治理能力。

2) 动态查询

MongoDB 支持动态查询，查询语言丰富，能够满足大多数应用的需求，并且支持丰富的查询语法和灵活的条件查询，包括范围查询、正则表达式、地理空间查询等。这使得开发人员可以轻松构建复杂的查询来满足各种需求。

3) 可扩展性

MongoDB 被用在一些规模庞大的环境中，FourSquare 和 Craigslist 网站都在使用它。通过分片技术，MongoDB 能够实现数据的水平扩展，以应对大规模数据的存储和高并发访问。如此优秀的扩展能力，使得它能够快速响应业务变化。

4) 复制和故障修复

MongoDB 支持复制功能，可以在多个服务器之间复制数据。虽然这是一种数据冗余的存储方式，但是可以提高数据的安全性、可用性和可靠性。同时，MongoDB 还支持故障恢复功能。例如，当主节点出现故障时，系统可以自动切换到从节点，以保证数据库的可用性、完整性和可靠性。

5) 支持多种存储引擎

MongoDB 支持多种存储引擎，如 WiredTiger、MMAPv1，这使得 MongoDB 能够适应不同的应用场景和需求。从 MongoDB 3.2 版本开始支持 WiredTiger 存储引擎。与此同时，WiredTiger 成为 MongoDB 的默认存储引擎，不再使用 MMAPv1 作为默认存储引擎。WiredTiger 存储引擎通过使用不同的数据结构，在兼顾磁盘 I/O 操作的同时，维持了获取数据的高速度。

除上述两种存储引擎外，MongoDB 还提供了 InMemory 存储引擎。该引擎将数据仅存储在内存中，而将少量的元数据和诊断日志存储到硬盘文件中。由于无须进行磁盘 I/O 操作，使用 InMemory 存储引擎可以显著降低数据查询的延迟，从而快速获取所需的数据。然而，这种存储方式并不适合大规模数据存储，因为一旦服务器宕机，内存中的数据将会丢失。

6) 支持多语言驱动程序

MongoDB 支持多种语言的驱动程序，包括 Java、Python、PHP、C#、JavaScript 等，这使得开发者可以使用自己熟悉的编程语言来访问和使用 MongoDB。

1.3.2 MongoDB 的缺点

正如许多技术解决方案一样，MongoDB 在提供强大功能的同时，也存在一些潜在的局限性。以下是 MongoDB 的一些主要缺点。

1) 高存储空间消耗

MongoDB 通常会占用更多的磁盘空间，因为它为每个文档存储键和其他一些元数据，同时在集合和数据库级别维护索引。

2) 较高的内存消耗

MongoDB 倾向于使用更多的内存来提供更高的性能，尤其是在索引和聚合操作中，这可能导致更高的内存消耗。

3) 查询性能与复杂性

虽然 MongoDB 可以快速执行简单的查询，但对于复杂的查询和聚合操作，可能需要编写更复杂的聚合管道或使用 MapReduce。这相比关系数据库的 SQL 查询，可能需要更多的开发工作。

4) 数据一致性

MongoDB 是一个面向可扩展性和分布式架构的数据库，因此在数据一致性方面进行了一些权

衡。MongoDB 在默认配置下提供的是最终一致性，这意味着在复制和分片环境中，不同副本之间的数据同步可能存在一定的延迟。

5) 生态系统

与一些传统关系数据库相比，MongoDB 的生态系统相对较新，支持和工具生态系统相对有限。这意味着在使用特定工具或框架时，可能会遇到一些限制或缺乏成熟的解决方案。

此外，MongoDB 还存在以下问题：在集群分片中的数据分布不均匀、持续插入大量数据时写入性能会有较大波动、单机可靠性较差等。

1.4 常用概念

在 MongoDB 中，一个数据库包含多个集合，一个集合中包含多个文档。这使得 MongoDB 在结构上与关系数据库类似，但又在某些方面保持了灵活性。例如，在关系数据库中，数据按照表格的形式进行存储和组织，而在 MongoDB 中，数据则以集合的形式进行存储和组织。另外，MongoDB 是面向集合的存储，它支持的数据结构非常松散，是类似 JSON 的 BSON 格式。这使得 MongoDB 在处理大量数据时具有很高的效率。

- **数据库 (Database)**：在 MongoDB 中，数据库是一个逻辑容器，用于存储数据。每个数据库都有自己的集合和权限。MongoDB 支持多个数据库，并且每个数据库可以放置在不同的文件中。
- **集合 (Collection)**：集合是 MongoDB 中一组文档的集合。它类似于关系数据库中的表，但不需要在插入数据之前定义结构。集合中的文档可以有不同的字段和数据类型，这使得 MongoDB 非常灵活，可以根据需要存储各种类型的数据。
- **文档 (Document)**：文档是 MongoDB 中的基本数据单元，类似于关系数据库中的行。文档以 BSON (Binary JSON) 格式存储，这是一种类似于 JSON 的二进制格式，可以存储复杂的数据结构，包括数组、嵌套文档等。
- **BSON**：BSON 是一种二进制格式的存储和交换数据的方式，是 JSON 的一种扩展，用于在网络传输过程中存储和读取数据。
- **_id**：在 MongoDB 中，每个文档都有一个唯一的 _id 字段作为主键，用于唯一标识该文档。这个字段可以是用户自定义的，也可以是 MongoDB 自动生成的。
- **索引 (Index)**：MongoDB 支持通用的二级索引，适用于多种查询场景，并提供唯一索引、复合索引、地理空间索引以及全文索引。索引可以提高查询性能，使得数据的检索更加快速和高效。
- **聚合 (Aggregation)**：MongoDB 支持聚合管道 (Aggregation Pipeline)，可以用于数据的分析和处理。通过对数据进行分组、过滤、排序等操作，可以得到各种统计结果和数据摘要。
- **复制 (Replication)**：MongoDB 支持数据复制功能，可以用于数据的备份和故障恢复。通过将数据复制到多个服务器上，可以提高数据的可靠性和可用性。
- **分片 (Sharding)**：MongoDB 支持数据分片功能，可以将数据分布到多个服务器上存储

和处理。这可以提高系统的扩展性和性能，使得 MongoDB 能够处理大规模的数据集。

传统数据库中有些概念与上述概念会出现重复，但意义可能不同。表 1.4 列出了传统关系数据库与 MongoDB 在数据库术语方面的对比。

表1.4 数据库中的概念对比

传统关系数据库概念	MongoDB 概念	含 义
数据库（Database）	数据库（Database）	包含多个表格或集合，每个表格或集合包含一组相似的数据
表（Table）	集合（Collection）	用于存放同一业务数据的集合
行（Row）	文档（Document）	一条数据记录
列（Column）	字段（Field）	每个字段或列都有一个名称，用于标识该字段或列
索引（Index）	索引（Index）	对数据库表中一列或多列的值进行排序的一种结构，使用索引可以快速访问数据库表中的特定信息
主键（Primary Key）	ID（字段）	唯一地标识表中的每一行或文档
视图（View）	视图（View）	一个虚拟表，其内容由查询定义
表连接（Table Joins）	-	表连接，MongoDB 中不支持

1.5 数据类型

MongoDB 支持多种数据类型，常见的数据类型如表 1.5 所示。

表1.5 MongoDB中常见的数据类型

数据类型	描 述
String	字符串类型，是最常用的数据类型。不过在 MongoDB 中，只有 UTF-8 编码的字符串才是合法的
Integer	整型，用于存储数值。根据使用服务器的不同，整型可以分为 32 位或 64 位两种
Boolean	布尔型，用于存储布尔类型的值（true/false）
Double	双精度浮点型，用于存储浮点型（小数）数据
Min/Max keys	将一个值与 BSON 元素的最低值和最高值进行对比
Array	数组类型，用于将数组、列表或多个值存储在一个键中
Timestamp	时间戳，记录文档修改或添加的具体时间
Object	用于内嵌文档
Null	用于创建空值
Symbol	符号，该数据类型与字符串类型类似。不同的是，它一般用于采用特殊符号类型的语言
Date	日期时间，用 UNIX 时间格式来存储当前日期或时间。可以创建 Date 对象并将 date、month、year 的值传递给 Date 对象来指定自己的日期时间
Object ID	对象 ID，用于创建文档的 ID
Binary Data	二进制数据，用于存储二进制数据
Code	代码类型，用于在文档中存储 JavaScript 代码
Regular expression	正则表达式类型，用于存储正则表达式

其中有几个数据类型，是在传统数据库中不常见的，例如 Object ID、Object 等。Object ID 类似

于关系数据库中的主键 ID。在 MongoDB 中，Object ID 由 12 字节的字符组成，其中：

- 前 4 字节表示当前的 UNIX 时间戳。
- 之后的 3 字节是当前设备的机器标识码。
- 再之后的 2 字节是 MongoDB 服务器的进程 ID。
- 最后 3 字节为计数器，计数器的起始值可以随机获得。

63b05c84	a817c7	55a9	3afab2
↑	↑	↑	↑
时间戳	机器码	进程 ID	计数器

下面的示例展示了一个简单的文档结构：

```
{
  "id": ObjectId("63b05c84a817c755a93afab2"),
  "at": NumberLong("1672502404422"),
  "title": "《活着》",
  "title english": "To Live",
  "type": "长篇小说",
  "author": "余华",
  "publish": ISODate("1992-12-20T00:00:00.000Z"),
  "words": NumberInt("132000"),
  "awards": [
    {
      "title": "意大利“格林扎纳·卡佛”文学奖",
      "award_time": "1998 年 7 月"
    },
    {
      "title": "华文冰心文学奖",
      "award time": "2002 年"
    }
  ]
}
```

其中 `_id` 是一个 12 字节的十六进制数字，可确保每个文档的唯一性，其结构如上述 Object ID 的组成所述。`publish` 是一个标准化日期类型，类型名称为 `ISODate`。`at` 是一个长整型数字类型，类型名称为 `NumberLong`。`words` 是一个整型数字类型，类型名称为 `NumberInt`。其余字段为字符串类型，类型名称为 `String`（可省略）。

1.6 本章小结

本章主要介绍了 MongoDB 数据库的概述、发展历史、优缺点、常用概念和数据类型，并简述了 MongoDB 的特性及其在不同应用场景中的使用情况。通过阅读本章内容，读者将对 MongoDB 有一个整体的了解，能够理解 MongoDB 的出现背景、适用的业务场景以及它所支持的数据类型。本章内容较为基础，建议读者全面掌握。

第 2 章

MongoDB 的安装

第 1 章讲解了 MongoDB 的简介，相信读者对 MongoDB 已经有了基本的了解。理论知识只有结合实践，才能被更透彻地理解，也才能更高效地记忆。在开始实践之前，需要先准备工具，因此从本章开始，我们将介绍 MongoDB 的安装与使用。

MongoDB 有多个版本，包括企业版、社区版以及 Atlas 版本。本书选择免费的社区版进行演示。本章以 MongoDB 8.0 版本为例，挑选了服务器中常见的两种操作系统——Windows 和 Linux 进行演示，详细讲解 MongoDB 的安装过程和初步使用方法。如果需要使用 MongoDB 的其他版本，可以在官网左上角进行版本切换。

本章主要涉及的知识点包括：

- 在 Windows 系统上安装 MongoDB
- 在 Linux 系统上安装 MongoDB
- 在其他操作系统上安装 MongoDB
- 安装过程中常见的问题

2.1 在 Windows 系统上安装 MongoDB

在 Windows 系统上安装 MongoDB，首先需要确定自己的 Windows 系统版本以及 MongoDB 社区版所支持的系统版本。这里使用的是 Windows 10 操作系统和 MongoDB 8.0.4 版本。由于 MongoDB 6 版本之后不再默认安装 Mongo Shell，因此本节分为两部分：安装 MongoDB 和安装 MongoDB Shell。

2.1.1 安装 MongoDB

在 Windows 平台下，MongoDB 提供了两种安装形式：直接运行 MSI 安装包和 `msiexec.exe` 命令行工具，默认直接安装 MSI 安装包，`msiexec.exe` 命令行形式更倾向于使用自动化无人值守形式部署的系统管理员。

步骤 01 打开 MongoDB 官网的下载页面，地址为 <https://www.mongodb.com/try/download/community>。

单击 **Select Package** 显示版本选择界面。在 **Version** 下拉列表中选择版本 8.0.4，在 **Platform** 下拉列表中选择 Windows X64 操作系统，安装包格式选择 `msi`，单击 **Download** 按钮，开始

下载，如图 2.1 所示。

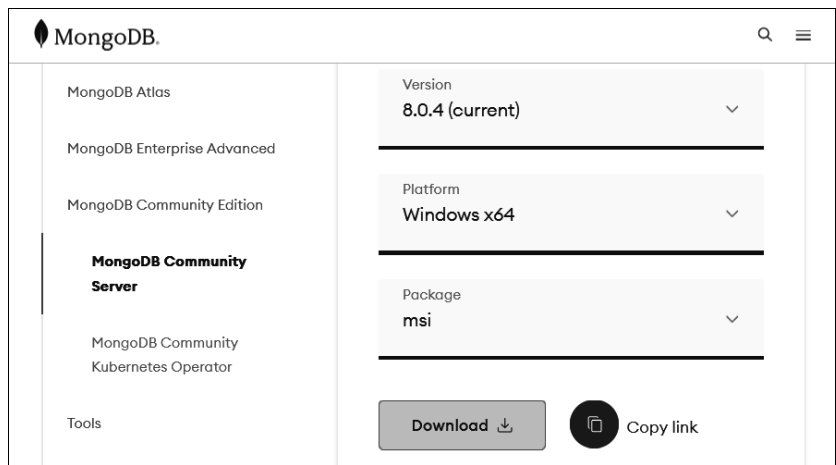


图 2.1 MongoDB 下载页面

步骤 02 下载完成之后，双击.msi 安装包文件开始安装。在安装过程中会弹出安装向导界面，指导使用者安装 MongoDB 以及 MongoDB 的可视化工具 MongoDB Compass。安装包运行后，界面如图 2.2 所示。



图 2.2 开始安装 MongoDB

步骤 03 安装向导提示当前安装的 MongoDB 版本信息，如图 2.3 所示，单击 Next 按钮。

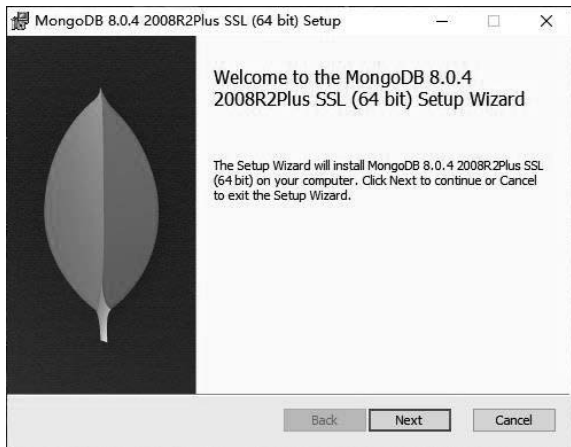


图 2.3 进入 MongoDB 安装向导

步骤 04 接受当前的终端用户安装协议，选中同意协议的复选框（I accept the terms in the License Agreement），单击 Next 按钮，如图 2.4 所示。

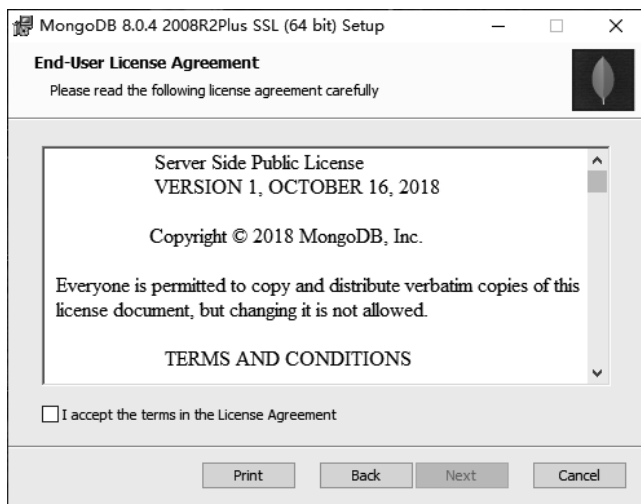


图 2.4 MongoDB 安装协议

步骤 05 选择安装类型，MongoDB 支持完整安装和个性化安装。个性化安装方式支持选择所需要的安装组件，同时也支持自定义 MongoDB 的安装路径。官方推荐的是完整安装。对于初学者或对 MongoDB 使用不熟练的使用者，建议默认完整安装。此方式会将 MongoDB 安装在路径 C:\Program Files\MongoDB\Server\8.0 下。选择 Complete 安装类型后，单击 Next 按钮，如图 2.5 所示。

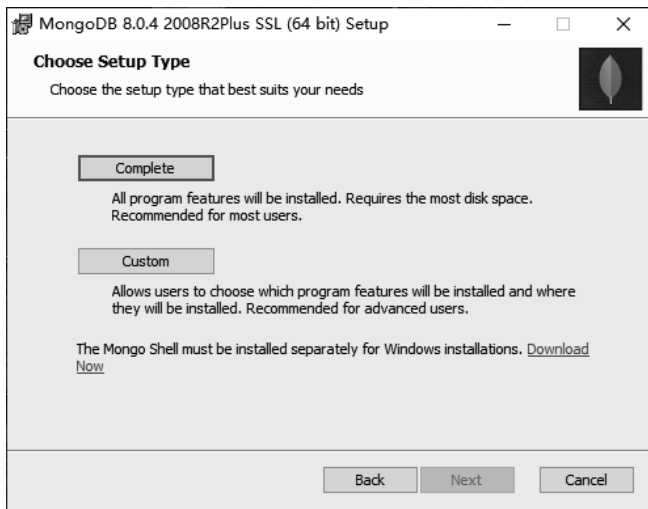


图 2.5 选择安装类型

步骤 06 配置服务，从 4.0 版本开始，MongoDB 可以配置为一项 Windows 服务。在这个步骤中，可以选择将 MongoDB 作为 Windows 的一个系统服务，如图 2.6 所示。在这个界面中，可以配置服务的名称，一般默认为 MongoDB，同时配置 MongoDB 的数据目录和日志目录。配置

完成后，单击 Next 按钮。如果这里不将 MongoDB 配置为 Windows 的服务，可以手动启动，具体可以查看后面关于手动启动 MongoDB 实例的内容。

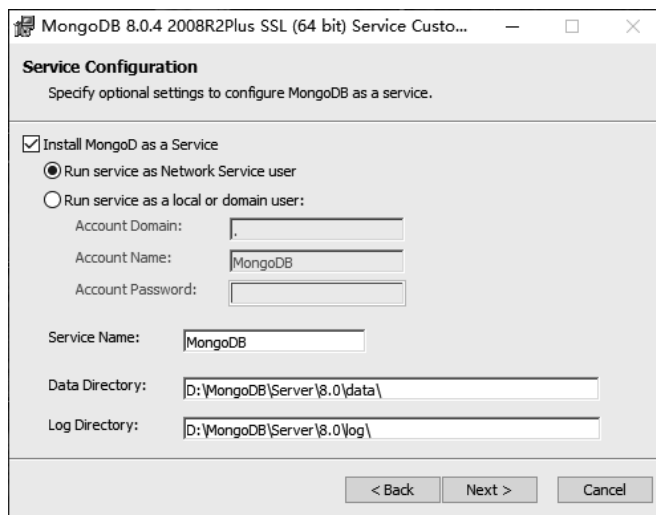


图 2.6 配置 MongoDB 服务

步骤 07 安装可视化工具 MongoDB Compass。这里为可选项，既可以安装，也可以不安装。这里选中进行安装，单击 Next 按钮进入下一步，如图 2.7 所示。

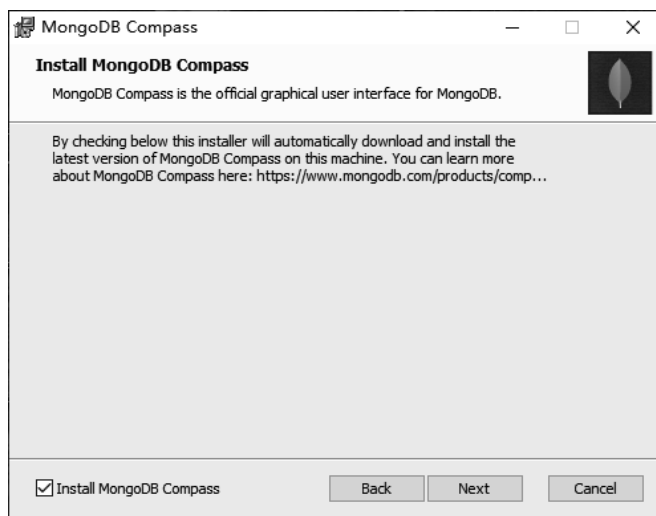


图 2.7 选择安装可视化工具

步骤 08 进入安装进程。上述所有配置完成，单击 Install 按钮进入安装进程，如图 2.8 所示。

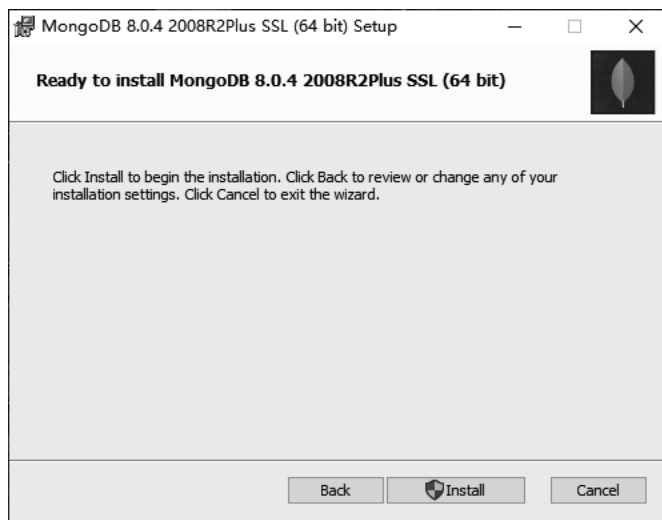


图 2.8 进入安装进程

步骤 09 等待安装进程进行。整个过程可能会持续较长时间，与所使用的计算机配置有关，不同配置耗费的时间有所不同，耐心等待即可，如图 2.9 所示。

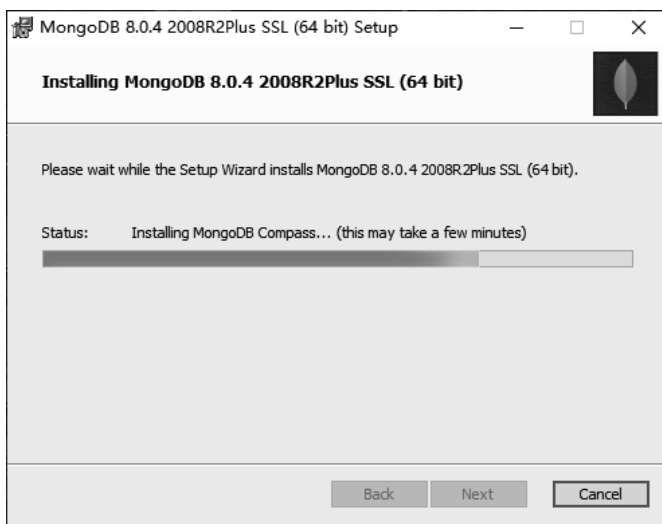


图 2.9 安装进行中

步骤 10 安装进程结束，完成 MongoDB 和 MongoDB Compass 的安装。单击 Finish 按钮，退出安装进程，如图 2.10 所示。

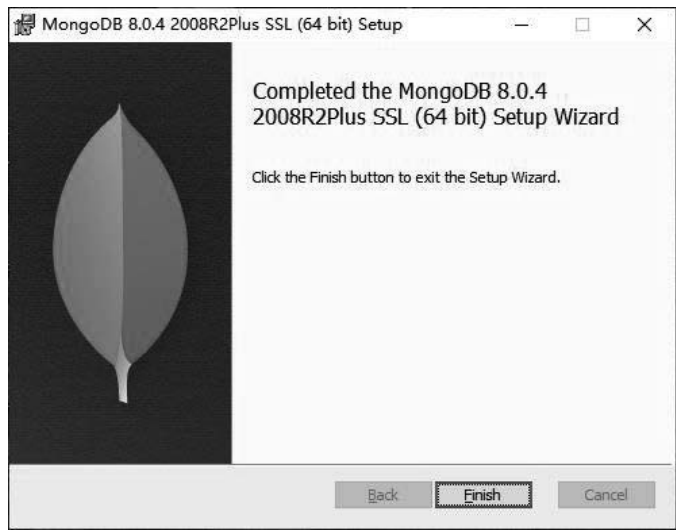


图 2.10 安装完成

如果将 MongoDB 作为服务安装，那么安装结束后，会自动开启服务。要查看 MongoDB 服务的状态，可以打开系统服务界面，按 Ctrl + Shift + Esc 组合键，打开任务管理器，切换到“服务”选项卡，或者直接通过 Windows 系统的搜索找到“服务”界面，在“服务”界面中找到 MongoDB 服务，查看服务是否正在运行。在服务名称上右击，可以选择开启或停止该服务，如图 2.11 所示。



图 2.11 “服务”界面

或者打开浏览器访问 <https://localhost:27017>，如图 2.12 所示。



图 2.12 在浏览器中访问 MongoDB

同样，也可以通过命令查看 MongoDB 服务的状态，开启或停止 MongoDB 服务。命令如下：

```
#查看 MongoDB 服务的状态
sc query MongoDB
#停止 MongoDB 服务
net stop MongoDB
```

```
#开启 MongoDB 服务
net start MongoDB
```

执行结果如图 2.13 所示。

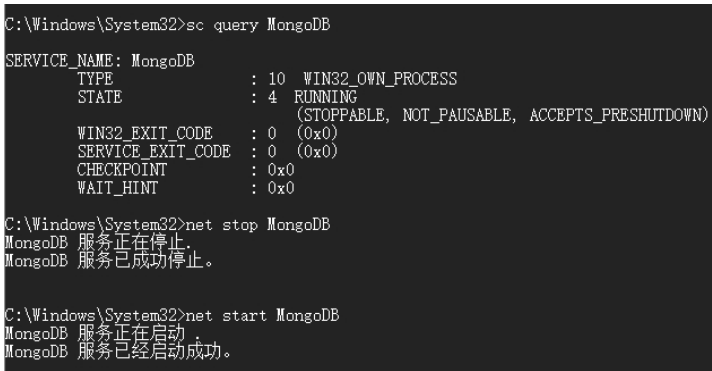


图 2.13 查看 MongoDB 服务的状态

如果不作为服务安装，那么需要手动启动 MongoDB 实例，这里需要用到 `mongod` 命令。该命令在 MongoDB 安装目录下的 `bin` 文件夹（`D:\MongoDB\Server\8.0\bin`）下。启动时，需要配置数据目录，例如 `D:\MongoDB\data`，如图 2.14 所示。



图 2.14 mongod 启动服务

MongoDB 启动后，该数据目录会生成一些数据库文件，如图 2.15 所示。

名称	修改日期	类型
db	2025/1/9 0:08	文件夹
diagnostic.data	2025/1/9 0:44	文件夹
journal	2025/1/9 0:44	文件夹
log	2024/5/16 9:53	文件夹
_mdb_catalog.wt	2025/1/9 0:44	WT 文件
collection-0-2921174767861492997.wt	2025/1/9 0:44	WT 文件
collection-2-2921174767861492997.wt	2025/1/9 0:44	WT 文件
collection-4-2921174767861492997.wt	2025/1/9 0:44	WT 文件
index-1-2921174767861492997.wt	2025/1/9 0:44	WT 文件
index-3-2921174767861492997.wt	2025/1/9 0:44	WT 文件
index-5-2921174767861492997.wt	2025/1/9 0:44	WT 文件
index-6-2921174767861492997.wt	2025/1/9 0:44	WT 文件
mongod.lock	2025/1/9 0:44	LOCK 文件
sizeStorer.wt	2025/1/9 0:44	WT 文件

图 2.15 MongoDB 的 data 目录

在 `bin` 目录下，存在一些关键文件，如图 2.16 所示。其中，`mongod.exe` 用来启动 MongoDB 服务，`mongos.exe` 用来管理分片集群。

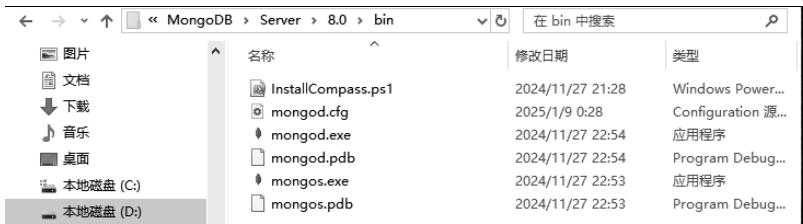


图 2.16 MongoDB 的 bin 目录

2.1.2 配置环境变量

为了方便使用 mongod 以及其他命令，可以将 MongoDB 的安装路径加入环境变量。在配置了环境变量后，以上命令就可以省略路径，只保留命令本身的名称。打开环境变量配置界面，检查是否已自动创建环境变量，如果没有，则进行手动创建。首先创建 MONGO_HOME 的系统变量，如图 2.17 所示。



图 2.17 配置 MongoDB 环境变量

然后编辑环境变量 Path，在最后加入一行内容，内容为：

```
%MONGO_HOME%\bin
```

最终结果如图 2.18 所示。这样就完成了环境变量的配置。之后使用相关命令时，不需要再进入命令所在目录。

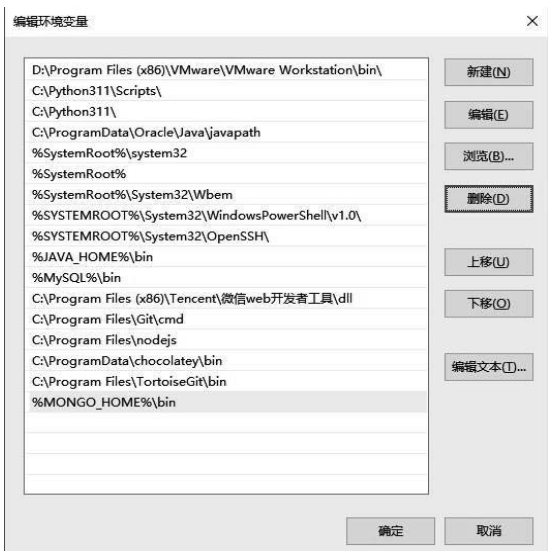


图 2.18 环境变量

2.1.3 其他安装形式

除前面介绍的安装方式外，MongoDB 还支持使用命令行工具 `msiexec.exe` 来安装。这种安装形式一般用在无人值守自动部署中。

首先，打开 `cmd` 命令窗口，使用如下命令：

```
msiexec.exe /l*v mdbinstall.log /qb /i mongodb-windows-x86_64-8.0.4-signed.msi
```

默认安装路径为 `C:\Program Files\MongoDB\Server\6.0\bin`，可以使用 `INSTALLLOCATION` 配置项来修改安装路径：

```
msiexec.exe /l*v mdbinstall.log /qb /i mongodb-windows-x86_64-8.0.4-signed.msi  
INSTALLLOCATION="C:\MongoDB\Server\6.0"
```

如果同步安装可视化工具，可以使用如下命令：

```
msiexec.exe /l*v mdbinstall.log /qb /i mongodb-windows-x86_64-8.0.4-signed.msi  
SHOULD_INSTALL_COMPASS="0"
```

其他设置和操作与双击运行 `.msi` 安装包的形式相同。

除以上形式外，还支持 `.zip` 免安装形式。下载 `.zip` 文件后，解压到相应目录，根据以上操作配置环境变量即可。

2.1.4 常见问题和注意事项

1. 服务无法启动

安装完成后，可能会出现服务无法启动的情况，如图 2.19 所示。

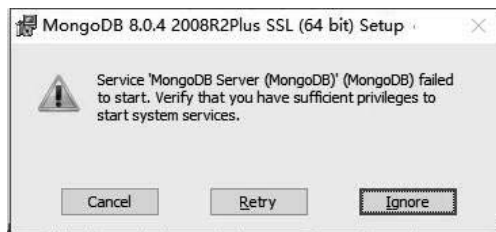


图 2.19 服务启动错误

首先检查是否存在 MongoDB 服务，如果存在，那么尝试手动启动；如果启动失败，那么先移除服务，再尝试手动添加服务；如果不存在 MongoDB 服务，可以手动添加 MongoDB 服务。移除 MongoDB 服务的命令如下：

```
sc delete MongoDB
```

添加服务的命令如下：

```
mongod -install -f "C:\Program Files\MongoDB\Server\8.0\bin\mongod.cfg"
```

手动添加 MongoDB 服务后，参考前述关于手动启动或停止 MongoDB 服务的描述，使用界面或命令开启 MongoDB 服务。

2. 单点发行

MongoDB 的自动版本更新须在同一大版本下，例如 4.2.1 到 4.2.2 可以自动更新，但 4.0 到 4.2 这种全量更新，需要重新安装。

3. 平台支持

MongoDB 8.0 社区版目前支持以下版本的 64 位 Windows 操作系统：

- Windows Server 2019
- Windows Server 2016
- Windows 7/10/11

4. 虚拟化

如果要使用 VirtualBox 在 Windows 系统上安装 MongoDB，则需要禁用 Hyper-V。

2.2 在 Linux 系统上安装 MongoDB

MongoDB 官网提供了多个 Linux 版本操作系统的安装指南，例如 Red Hat、CentOS、Ubuntu、Debian 和 SUSE 等版本的操作系统。限于篇幅，这里不再一一描述。感兴趣的读者可到官网进行查看，地址为 <https://www.mongodb.com/docs/manual/administration/install-on-linux/>。Red Hat 是商业版的 Linux 系统，与 CentOS 基本相似，所以它们的安装方式以及支持的 MongoDB 版本基本类似。

下面以 CentOS 7 为例，讲解 MongoDB 的安装方法。此安装教程同样适用于 Red Hat Enterprise Linux 7（RHEL 7）、Oracle Linux、Rocky Linux 和 AlmaLinux。

与 2.1 节讲解的在 Windows 系统上安装 MongoDB 一样，这里选择的 MongoDB 版本为 8.0.4，该数据库版本支持的操作系统版本如下：

- RHEL（Red Hat Enterprise Linux）
- CentOS
- Oracle Linux
- Rocky Linux
- AlmaLinux 9（Starting in MongoDB 8.0.4）
- AlmaLinux 8

2.2.1 使用 yum 方式安装 MongoDB

在 CentOS 系统上安装 MongoDB 有多种方式，可以直接使用在线 yum 方式安装，无须提前下载安装包，也可以直接到 <https://repo.mongodb.org/yum/redhat/> 页面下载 rpm 文件安装，或者在官网 <https://www.mongodb.com/try/download/community> 页面下载 tgz 文件进行安装。下面演示如何使用 yum 方式进行在线安装。

首先创建 repo 文件，文件目录为 `/etc/yum.repos.d/mongodb-org-8.0.repo`，文件内容如下：

```
[mongodb-org-8.0]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/7/mongodb-org/8.0/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-8.0.asc
```

repo 文件内容如图 2.20 所示。

```
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/7/mongodb-org/8.0/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-8.0.asc
```

图 2.20 repo 文件内容

如果使用的是 Red Hat Enterprise Linux 8, 那么 baseurl 需要换成 https://repo.mongodb.org/yum/redhat/8/mongodb-org/8.0/x86_64/, 以此类推。

创建完成后, 执行安装命令, 如下所示:

```
sudo yum install -y mongodb-org
```

结果如图 2.21 所示。

```
Dependencies Resolved
=====
Package                               Arch              Version           Repository        Size
=====
Installing:
mongodb-org                           x86_64            6.0.12-1.el7      mongodb-org-6.0   6.2 k
mongodb-org-database                  x86_64            6.0.12-1.el7      mongodb-org-6.0   6.3 k
mongodb-org-mongos                     x86_64            6.0.12-1.el7      mongodb-org-6.0   22 M
mongodb-org-server                     x86_64            6.0.12-1.el7      mongodb-org-6.0   31 M
mongodb-org-tools                      x86_64            6.0.12-1.el7      mongodb-org-6.0   6.2 k
=====
Transaction Summary
=====
Install 5 Packages
Total download size: 53 M
Installed size: 233 M
Downloading packages:
(1/5): mongodb-org-database-6.0.12-1.el7.x86_64.rpm | 6.3 kB 00:00:00
(2/5): mongodb-org-6.0.12-1.el7.x86_64.rpm | 6.2 kB 00:00:00
(3/5): mongodb-org-mongos-6.0.12-1.el7.x86_64.rpm | 22 MB 00:00:04
(4/5): mongodb-org-server-6.0.12-1.el7.x86_64.rpm | 6.2 kB 00:00:00
(5/5): mongodb-org-tools-6.0.12-1.el7.x86_64.rpm | 31 MB 00:00:06
-----
Total                                     7.0 MB/s | 53 MB 00:00:07
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : mongodb-org-mongos-6.0.12-1.el7.x86_64 | 1/5
  Installing : mongodb-org-tools-6.0.12-1.el7.x86_64 | 2/5
  Installing : mongodb-org-server-6.0.12-1.el7.x86_64 | 3/5
Created symlink from /etc/systemd/system/multi-user.target.wants/mongod.service to /usr/lib/systemd/system/mongod.service.
  Installing : mongodb-org-database-6.0.12-1.el7.x86_64 | 4/5
  Installing : mongodb-org-6.0.12-1.el7.x86_64 | 5/5
  Verifying : mongodb-org-server-6.0.12-1.el7.x86_64 | 1/5
  Verifying : mongodb-org-database-6.0.12-1.el7.x86_64 | 2/5
  Verifying : mongodb-org-tools-6.0.12-1.el7.x86_64 | 3/5
  Verifying : mongodb-org-mongos-6.0.12-1.el7.x86_64 | 4/5
  Verifying : mongodb-org-6.0.12-1.el7.x86_64 | 5/5
Installed:
mongodb-org.x86_64 0:6.0.12-1.el7      mongodb-org-database.x86_64 0:6.0.12-1.el7      mongodb-org-mongos.x86_64 0:6.0.12-1.el7
mongodb-org-server.x86_64 0:6.0.12-1.el7      mongodb-org-tools.x86_64 0:6.0.12-1.el7
Complete!
[root@localhost ~]#
```

图 2.21 安装完成界面

如果要指定 MongoDB 6.0 下的某个版本, 可以执行如下语句:

```
sudo yum install -y mongodb-org-6.0.10 mongodb-org-database-6.0.10
```

```
mongodb-org-server-6.0.10 mongodb-org-mongos-6.0.10 mongodb-org-tools-6.0.10
```

结果如图 2.22 所示。

```
[root@localhost ~]# sudo yum install -y mongodb-org-6.0.10 mongodb-org-database-6.0.10 mongodb-org-server-6.0.10 mongodb-org-mongos-6.0.10 mongodb-org-tools-6.0.10
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirrors.ustc.edu.cn
 * extras: mirrors.ustc.edu.cn
 * updates: mirrors.ustc.edu.cn
Package matching mongodb-org-6.0.10-1.el7.x86_64 already installed. Checking for update.
Package matching mongodb-org-database-6.0.10-1.el7.x86_64 already installed. Checking for update.
Package matching mongodb-org-server-6.0.10-1.el7.x86_64 already installed. Checking for update.
Package matching mongodb-org-mongos-6.0.10-1.el7.x86_64 already installed. Checking for update.
Package matching mongodb-org-tools-6.0.10-1.el7.x86_64 already installed. Checking for update.
```

图 2.22 指定安装版本

默认情况下，MongoDB 使用 `mongod` 的 Linux 系统用户账户，使用如下默认的数据目录和日志目录，安装过程中会自动创建上述目录。

```
/var/lib/mongo (the data directory)
/var/log/mongodb (the log directory)
```

如果要修改该目录，可以修改 `mongod.conf` 文件的配置。

- `storage.dbPath`: 配置数据目录，例如 `/some/data/directory`。
- `systemLog.path`: 配置日志目录，例如 `/some/log/directory/mongod.log`。

创建完成后，确保当前登录系统的用户账号有权限操作这些文件和目录，如果没有的话，可以使用如下命令配置：

```
sudo chown -R mongod:mongod /some/data/directory
```

安装完成后，要运行和管理 `mongod` 进程，需要使用操作系统的内置初始化系统启动或停止该进程。新版本的 Linux 使用 `systemd`（使用命令 `systemctl`），而旧版本的 Linux 使用 `System V init`（使用命令 `service`），这里的 CentOS 7 使用的是 `systemctl`。

如果无法确定应该使用哪个命令，请在 Linux 下运行以下命令来确定：

```
ps --no-headers -o comm 1
```

如果结果为 `systemd`，则使用 `systemctl` 命令；如果结果为 `init`，则使用 `service` 命令。作者执行命令的结果如图 2.23 所示。

```
[root@localhost ~]# ps --no-headers -o comm 1
systemd
```

图 2.23 确定命令

确定命令后，使用命令开启 MongoDB 服务。命令如下：

```
sudo systemctl start mongod
```

验证 MongoDB 是否已成功启动。通过以下命令进行检查：

```
sudo systemctl status mongod
```

执行结果如图 2.24 所示。从图中可以看到 MongoDB 当前的状态、启动的时间、配置文件的目录以及启动服务所使用的 `sh` 命令脚本文件。

```

[root@localhost ~]# sudo systemctl start mongod
[root@localhost ~]# sudo systemctl status mongod
* mongod.service - MongoDB Database Server
   Loaded: loaded (/usr/lib/systemd/system/mongod.service; enabled; vendor preset: disabled)
   Active: active (running) since Fri 2023-12-22 18:58:07 CST; 49s ago
     Docs: https://docs.mongodb.org/manual
   Main PID: 1491 (mongod)
   CGroup: /system.slice/mongod.service
           └─1491 /usr/bin/mongod -f /etc/mongod.conf

Dec 22 18:58:07 localhost.localdomain systemd[1]: Started MongoDB Database Server.
Dec 22 18:58:07 localhost.localdomain mongod[1491]: {"t":{"$date":"2023-12-22T02:58:07.728Z"},"s":"I", "c":"","CONTROL", "id":"7404500", "ctx":"","msg"... false"}
Hint: Some lines were ellipsized, use -l to show in full.
[root@localhost ~]#

```

图 2.24 查看 MongoDB 服务状态

如果希望 MongoDB 随操作系统自动重启，可以使用如下命令：

```
sudo systemctl enable mongod
```

停止 MongoDB 服务，可以使用如下命令：

```
sudo systemctl stop mongod
```

重新启动 MongoDB，可以使用如下命令：

```
sudo systemctl restart mongod
```

MongoDB 启动后，会生成相关的日志文件，此文件可用来跟踪错误或重要消息的进程状态，日志文件目录为 `/var/log/mongodb/mongod.log`。

完成以上操作后，确保 MongoDB 处于开启状态，开始使用 `mongo` 连接数据库。与 Windows 不同，Linux 下会直接安装 `mongo` 脚本，而不需要再次手动安装。

Mongo 脚本工具命令为 `mongo`，执行结果如图 2.25 所示。

```

[root@localhost ~]# mongo
MongoDB shell version: 6.0.12
connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongo-2.1.1
Using MongoDB: 6.0.12
Using Mongosh: 2.1.1

For mongosh info see: https://docs.mongodb.com/mongosh-shell/

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

-----
The server generated these startup warnings when booting
2023-12-22T18:58:08.237+08:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2023-12-22T18:58:08.237+08:00: /sys/kernel/mm/transparent_hugepage/enabled is 'always'. We suggest setting it to 'never'
2023-12-22T18:58:08.237+08:00: /sys/kernel/mm/transparent_hugepage/defrag is 'always'. We suggest setting it to 'never'
2023-12-22T18:58:08.237+08:00: vm.max_map_count is too low
-----

Deprecation warnings:
- Using mongo on the current operating system is deprecated, and support may be removed in a future release.
See https://www.mongodb.com/docs/mongosh-shell/install/#supported-operating-systems for documentation on supported platforms.
test>

```

图 2.25 mongo 连接数据库

至此，就可以使用数据库命令操作数据库了。例如显示当前的数据库，如图 2.26 所示。

```

test> show databases
admin    40.00 KiB
config  12.00 KiB
local    40.00 KiB
test>

```

图 2.26 显示当前数据库

2.2.2 卸载 MongoDB

要从系统中完全删除 MongoDB，必须删除 MongoDB 应用程序本身、配置文件以及包含以下

内容的任何目录、数据和日志。本小节内容将指导读者完成必要的步骤。

下面的卸载过程将完全删除 MongoDB 及其配置和所有数据库。此过程是不可逆的，因此在继续操作之前，请确保备份所有的配置和数据。

1. 停止 MongoDB

停止 mongod 服务可以通过以下命令完成：

```
sudo service mongod stop
```

2. 删除包

删除以前安装的所有 MongoDB 包：

```
sudo yum erase $(rpm -qa | grep mongodb-org)
```

3. 删除数据目录

删除 MongoDB 数据库和日志文件：

```
sudo rm -r /var/log/mongodb sudo rm -r /var/lib/mongo
```

2.2.3 使用其他方式安装 MongoDB

MongoDB 同样支持以 tgz 压缩包形式进行安装。首先使用如下命令安装依赖包：

```
sudo yum install libcurl openssl xz-libs
```

然后到官网下载 tgz 压缩包，下载完成后解压缩，命令如下：

```
tar -zxvf mongodb-linux-*8.0.4.tgz
```

将 MongoDB 的 bin 目录配置到环境变量的 Path 中。支持如下两种方式：

- 复制 MongoDB 的 bin 目录下的文件到 path 变量中已配置的某一路径中，例如 `sudo cp /path/to/the/mongodb-directory/bin/* /usr/local/bin/`。
- 在 MongoDB 的 bin 目录和 path 变量中已配置的某一路径之间创建系统链接映射，例如 `sudo ln -s /path/to/the/mongodb-directory/bin/* /usr/local/bin/`。

这种方式安装的 MongoDB 的默认数据目录和日志目录与 yum 方式一样。首先创建目录：

```
sudo mkdir -p /var/lib/mongo
sudo mkdir -p /var/log/mongodb
```

因为 MongoDB 默认使用的系统用户为 mongod，所以同样需要为 group 为 mongod、用户名为 mongod 的用户进行目录授权：

```
sudo chown -R mongod:mongod /var/lib/mongo
sudo chown -R mongod:mongod /var/log/mongodb
```

如果不使用默认目录，则修改方式与 yum 方式相同。

2.2.4 常见错误和注意事项

1. 允许访问 cgroup

当前的 SELinux 策略不允许 MongoDB 进程访问/sys/fs/cgroup，这样 MongoDB 就无法确定系统上的可用内存。如果你打算强制运行 SELinux，则需要进行以下调整。

首先，确保你的系统已安装软件包 checkpolicy：

```
sudo yum install checkpolicy
```

其次，创建自定义策略文件 mongodb_cgroup_memory.te：

```
cat > mongodb_cgroup_memory.te <<EOFmodule mongodb_cgroup_memory 1.0;require
{ type cgroup t; type mongod t; class dir search; class file { getattr open
read };}#===== mongod t =====allow mongod t cgroup_t:dir
search;allow mongod_t cgroup_t:file { getattr open read };EOF
```

创建后，编译和加载自定义策略模块，运行以下 3 个命令：

```
checkmodule -M -m -o mongodb_cgroup_memory.mod
mongodb_cgroup_memory.tesemodule package -o mongodb_cgroup_memory.pp -m
mongodb_cgroup_memory.modsudo semodule -i mongodb_cgroup_memory.pp
```

MongoDB 进程现在能够以 SELinux 模式强制访问文件了。

2. 系统 ulimit 值

大多数类 UNIX 系统会限制进程所使用的系统资源数，MongoDB 4.4 之后，这个值低于 64 000 会影响 MongoDB 系统的运行，这个数值建议设置到 64 000。

3. 以 yum 方式安装 MongoDB 的自动更新问题

以 yum 方式安装的 MongoDB 会自动更新，如果想要关闭自动更新功能，可以修改/etc/yum.conf，在文件中加入如下内容：

```
exclude=mongodb-org,mongodb-org-database,mongodb-org-server,mongodb-mongosh
,mongodb-org-mongos,mongodb-org-tools
```

2.3 本章小结

本章主要介绍了在 Windows 和 Linux 平台下安装 MongoDB 社区免费版数据库的过程，详细讲解了各个平台下的多种安装方式，并涵盖详细的步骤和操作流程。同时，本章还列出了安装过程中常见的问题及需要注意的事项。读者需要根据本章内容多次动手实践，才能更好地掌握安装过程，并有助于更透彻地理解 MongoDB 数据库的整体架构。

对于其他操作系统，例如 macOS 或 Docker 环境下的安装，读者可以参考 MongoDB 官方网站的相关文档，地址为：<https://www.mongodb.com/docs/manual/installation/>。请注意，由于网络原因，您可能无法成功访问该链接。如果需要访问该网页，请检查链接的合法性，并尝试重新加载网页。如果问题持续存在，建议稍后再试，或者访问 MongoDB 的官方文档页面查找相关信息。