

银河麒麟服务器操作系统 V4

mongodb 软件适配手册



KYLIN
银河麒麟

天津麒麟信息技术有限公司

2019年6月

目 录

目 录.....	I
1 概述.....	2
1.1 系统概述.....	2
1.2 环境概述.....	2
1.3 MONGODB 软件简介.....	2
1.4 MONGODB 特点.....	2
1.5 MONGODB 使用原理.....	3
1.6 MONGODB 数据逻辑结构.....	3
2 MONGODB 软件适配.....	4
3 MONGODB 常见操作.....	4

1 概述

1.1 系统概述

银河麒麟服务器操作系统主要面向军队综合电子信息系统、金融系统以及电力系统等国家关键行业的服务器应用领域，突出高安全性、高可用性、高效数据处理、虚拟化等关键技术优势，针对关键业务构建的丰富高效、安全可靠的功能特性，兼容适配长城、联想、浪潮、华为、曙光等国内主流厂商的服务器整机产品，以及达梦、金仓、神通、南大通用等主要国产数据库和中创、金蝶、东方通等国产中间件，满足虚拟化、云计算和大数据时代，服务器业务对操作系统在性能、安全性及可扩展性等方面的需求，是一款具有高安全、高可用、高可靠、高性能的自主可控服务器操作系统。

1.2 环境概述

服务器型号	长城信安擎天 DF720 服务器
CPU 类型	飞腾 2000+处理器
操作系统版本	Kylin-4.0.2-server-sp2-2000-19050910.Z1
内核版本	4.4.131
mongodb 版本	2.6.10

1.3 mongodb 软件简介

MongoDB 是一个基于分布式文件存储的数据库。由 C++语言编写。旨在为 WEB 应用提供可扩展的高性能数据存储解决方案。

MongoDB 是一个介于关系数据库和非关系数据库之间的产品，是非关系数据库当中功能最丰富，最像关系数据库的。它支持的数据结构非常松散，是类似 json 的 bson 格式，因此可以存储比较复杂的数据类型。Mongo 最大的特点是它支持的查询语言非常强大，其语法有点类似于面向对象的查询语言，几乎可以实现类似关系数据库单表查询的绝大部分功能，而且还支持对数据建立索引。

1.4 Mongoddb 特点

- 面向集合存储，易存储对象类型的数据；
- 模式自由；
- 支持动态查询；
- 支持完全索引，包含内部对象；
- 支持查询；
- 支持复制和故障恢复；

- 使用高效的二进制数据存储，包括大型对象（如视频等）；
- 自动处理碎片，以支持云计算层次的扩展性；
- 支持 RUBY, PYTHON, JAVA, C++, PHP, C#等多种语言；
- 文件存储格式为 BSON（一种 JSON 的扩展）；
- 可通过网络访问。

1.5 Mongodb 使用原理

所谓“面向集合”（Collection-Oriented），意思是数据被分组存储在数据集中，被称为一个集合（Collection）。每个集合在数据库中都有一个唯一的标识名，并且可以包含无限数目的文档。集合的概念类似关系型数据库（RDBMS）里的表（table），不同的是它不需要定义任何模式（schema）。Nytro MegaRAID 技术中的闪存高速缓存算法，能够快速识别数据库内大数据集中的热数据，提供一致的性能改进。

模式自由（schema-free），意味着对于存储在 mongodb 数据库中的文件，我们不需要知道它的任何结构定义。如果需要的话，你完全可以把不同结构的文件存储在同一个数据库里。

存储在集合中的文档，被存储为键-值对的形式。键用于唯一标识一个文档，为字符串类型，而值则可以是各种复杂的文件类型。这种存储形式称为 BSON（Binary Serialized Document Format）。

MongoDB 已经在多个站点部署，其主要场景如下：

1) 网站实时数据处理。它非常适合实时的插入、更新与查询，并具备网站实时数据存储所需的复制及高度伸缩性。

2) 缓存。由于性能很高，它适合作为信息基础设施的缓存层。在系统重启之后，由它搭建的持久化缓存层可以避免下层的数据源过载。

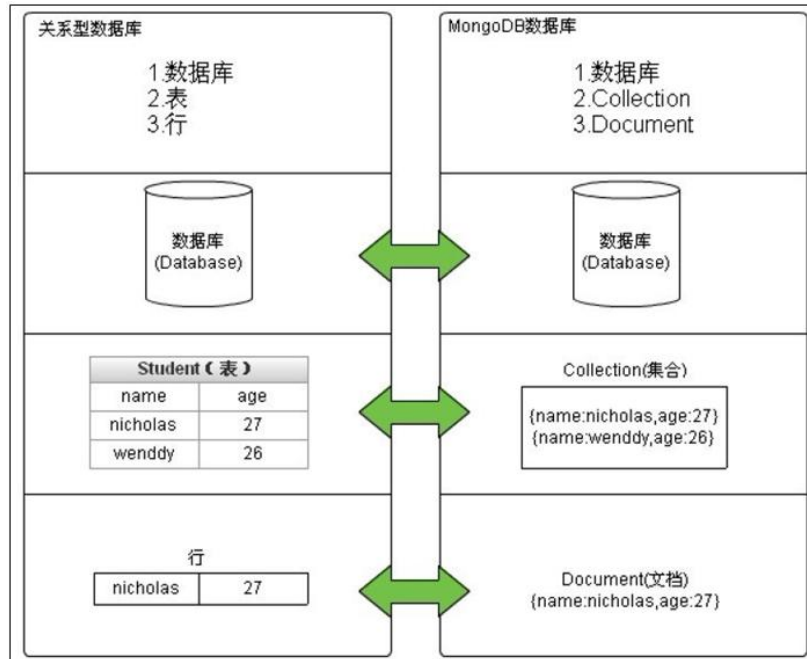
3) 高伸缩性的场景。非常适合由数十或数百台服务器组成的数据库，它的路线图中已经包含对 MapReduce 引擎的内置支持。

不适用的场景如下：

- 1) 要求高度事务性的系统。
- 2) 传统的商业智能应用。
- 3) 复杂的跨文档（表）级联查询。

1.6 Mongodb 数据逻辑结构

MongoDB 数据逻辑结构分为数据库（database）、集合（collection）、文档（document）三层。



- 一个 mongod 实例中允许创建多个数据库。
- 一个数据库中允许创建多个集合（集合相当于关系型数据库的表）。
- 一个集合则是由若干个文档构成（文档相当于关系型数据库的行，是 MongoDB 中数据的基本单元）。

2 mongodb 软件适配

1) 安装 mongodb

首先，请配置好银河麒麟服务器系统软件安装源。

```
$ apt install mongodb
```

2) 查看服务状态

```
$ service mongod status
```

```
root@sp2-arm64-hargrove-01:/opt# service mongodb status
● mongodb.service - An object/document-oriented database
   Loaded: loaded (/lib/systemd/system/mongodb.service; enabled; vendor preset: enabled)
   Active: active (running) since 三 2019-06-12 09:24:36 CST; 1min 49s ago
     Docs: man:mongod(1)
    Main PID: 5713 (mongod)
      Tasks: 9 (limit: 512)
   CGroup: /system.slice/mongodb.service
           └─5713 /usr/bin/mongod --config /etc/mongodb.conf

6月 12 09:24:36 sp2-arm64-hargrove-01 systemd[1]: Started An object/document-oriented database.
```

3 Mongodb 常见操作

1) 登录客户端

```
$ mongo
```

2) 创建数据库及 Collection

```
> use Test
switched to db Test
> db.createCollection('collection1')
{ "ok" : 1 }
```

3) 查看数据库

```
> show dbs
Test    0.078GB
admin   (empty)
local   0.078GB
```

4) 数据插入

```
> db.student.insert({_id:1, sname: 'zhangsan', sage: 20})
WriteResult({ "nInserted" : 1 })
```

5) 更新数据

```
> db.student.save({_id:2, sname: 'lisi', sage: 25})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

6) 查看数据

```
> db.student.find()
{ "_id" : 1, "sname" : "zhangsan", "sage" : 20 }
{ "_id" : 2, "sname" : "lisi", "sage" : 25 }
{ "_id" : ObjectId("5afce712cdb7d29a97a9fc8e"), "sname" : "lisi", "sage" : 21 }
{ "_id" : ObjectId("5afce712cdb7d29a97a9fc8f"), "sname" : "wangwu", "sage" : 22 }
```

7) 查询指定列数据

```
> db.student.find({}, {sname:1, sage:1})
{ "_id" : 1, "sname" : "zhangsan", "sage" : 20 }
{ "_id" : 2, "sname" : "lisi", "sage" : 25 }
{ "_id" : ObjectId("5afce712cdb7d29a97a9fc8e"), "sname" : "lisi", "sage" : 21 }
{ "_id" : ObjectId("5afce712cdb7d29a97a9fc8f"), "sname" : "wangwu", "sage" : 22 }
```

8) and 与条件查询

```
> db.student.find({sname: 'zhangsan', sage: 20})
{ "_id" : 1, "sname" : "zhangsan", "sage" : 20 }
```

9) or 条件查询

```
> db.student.find({$or: [{sage: 20}, {sage: 25}]})
{ "_id" : 1, "sname" : "zhangsan", "sage" : 20 }
```

```
{ "_id" : 2, "sname" : "lisi", "sage" : 25 }
```

10) 修改数据

最后一个 `true` 表示修改查询到的所有结果，`false` 表示只修改查询到的第一个结果。

```
> db.student.update({sname: 'lisi'}, {$set: {sage: 30}}, false, true)
WriteResult({ "nMatched" : 2, "nUpserted" : 0, "nModified" : 2 })
```

11) 删除字段

```
> db.student.remove({sname: 'zhangsan'})
WriteResult({ "nRemoved" : 1 })
```

12) 删除文档

```
> db.collection1.remove({})
WriteResult({ "nRemoved" : 0 })
```

13) 删除当前数据库

```
> db.dropDatabase()
{ "dropped" : "Test", "ok" : 1 }
```

14) 退出

```
> exit
```