# 工业大数据

非结构化数据库管理系

**MBBench-U** 

# 修订历史

版本	日期	主要作者	
1.0	13/4/2018	程梦倩	初稿
1.1	26/4/2018	葛昊	增加、修正内容
1.2	26/6/2018	程梦倩	增加、修正内容
1.3	27/6/2018	葛昊	增加、修正内容
1.4	16/9/2018	葛昊、程梦倩	增加案例内容
1.41	17/9/2018	葛昊、程梦倩	增加、修正内容

# 目录

1. MBBe	ench-U 概述	1
1.1	MBBench-U 介绍	1
1.2	MBBench-U 功能特点	1
1.3	手册介绍目标及读者人群	2
2. 负载:	场景及应用	2
2.1	工业大数据背景	2
2.2	负载描述	3
	2.2.1 负载一: 数据传输	3
	2.2.2 负载二: 单属性查询	3
	2.2.3 负载三: 多属性查询	4
	2.2.4 负载四: 版本管理	4
	2.2.5 负载五: 排序	5
	2.2.6 负载六: 统计值查询	5
	2.2.7 负载七: 异常文件检测	6
	2.2.8 负载八: 热点文件查询	6
3. MBB	ench-U 安装	7
3.1	运行环境	7
3.2	安装环境	7
	3.2.1 Java 安装说明	7
	3.2.2 Tomcat 安装说明	7
	3.2.3 Maven 安装说明	7
3.3	组件步骤	8
4. 操作	指南	9
4.1	数据生成	9
	4.1.1 初始化生成	9
	4.1.2 历史记录生成	12
4.2	负载测试	13
	4.2.1 数据传输负载测试	14
	4.2.2 简单查询负载测试	14
	4.2.3 复杂查询负载测试	15
4.3	结果显示	16
	4.3.1 数据传输负载结果	17
	4.3.2 简单查询负载结果	18
	4.3.3 复杂查询负载结果	19
4.4	结果对比	19
	开发	
5.1	关于结果验证接口的使用	27
	5.1.1 版本管理结果检测	27
	5.1.2 异常文件结果检测	27
	5.1.3 条件查询结果检测	28
	5.1.4 统计值计算查询结果检测	29
	5.1.5 排序负载结果检测	29

6.	故障排除与错误信息3	30
	6.1 组件过程问题3	30
7.	附录一:测试结果文档详细说明3	30
8.	附录二:测试流程案例3	36
	8.1 MongoDB 测试流程3	36
	8.1.1 MongoDB 环境说明与启动3	36
	8.1.2 系统设置	37
	8.1.3 海量文件生成	38
	8.1.4 数据库连接	40
	8.1.5 数据上传负载测试4	11
	8.1.6 数据下载负载测试4	43
	8.1.7 简单查询负载测试4	45
	8.1.8 复杂查询负载测试4	16
	8.1.9 结果查看4	17
	8.2 Elasticsearch 测试流程	49
	8.2.1 Elasticsearch 环境说明与启动	19
	8.2.2 系统设置5	51
	8.2.3 海量文件生成	51
	8.2.4 数据库连接5	51
	8.2.5 数据上传负载测试5	52
	8.2.6 数据下载负载测试5	52
	8.2.7 简单查询负载测试5	52
	8.2.8 复杂查询负载测试	
	8.2.9 结果查看	

# 1. MBBench-U 概述

### 1.1 MBBench-U 介绍

"工业大数据非结构化数据库系统评测工具——MBBench-U"是一款针对高端制造业的数据特征与特点,制定非结构化数据管理系统的测试工具。该测试工具以"仿真实验结果文件"为数据原型,以"海量小文件存储与管理"为主题,以高端制造业对非结构化数据管理的需要为应用场景。

系统评测工具由数据生成、负载测试和结果显示、结果对比四个部分组成,其中数据生成部分可作为一个独立应用,便于用户快速生成海量小文件及相应元数据文件。负载测试部分不仅选取了通用的非结构化功能测试,还根据工业领域中的制造业在非结构化数据的典型应用,设计相应的负载功能。结果显示部分将相应数据以表格、图形显示,使用户对结果内容有更直观、清晰的理解,并在结果对比部分对不同测试数据库的评测结果进行全方面的对比,给出评测对比结果报告。

# 1.2 MBBench-U 功能特点

MBBench-U是一款针对工业大数据非结构化数据库的基准评测工具,旨在为用户提供最大限度的方便、快捷、高效的数据库评测。基于这一目标,MBBench-U系统拥有简单易用的操作,多种类的主流数据库评测,快捷的海量文件生成,全面精准的负载测试以及多样化的结果展示五大特点。

### 1. 简单易用的操作

MBBench-U 界面友好,简单易用,文件生成、负载测试、结果显示等操作都实现一键式执行,让不同知识背景的用户都可以无障碍使用本系统。

### 2. 多种类的主流数据库评测

MBBench-U 支持多种当下主流的非结构化数据库的评测,如: MongoDB、CouchBase、ElasticSearch 等。用户还可以进行二次开发,只需实现特定的方法类,就可以对新的数据库进行评测。

### 3. 快捷的海量文件生成

MBBench-U根据工业大数据的特点,实现了海量小文件的数据的快速生成,海量文件可用作非结构化数据库及文件管理系统的评测导入数据。用户可以在较短的时间内,生成一定规模的小型不可读文件以及该文件的元数据文件,其中元数据文件的格式包含当今主流的非结构化数据库的元数据格式,用户无需更正便可将相应格式的元数据文件导入到数据库内。

### 4. 全面精准的负载测试

MBBench-U包含全面精准的负载测试,负载设计考量工业领域中的制造业在非结构化数据的典型应用,与实际需求接轨。MBBench-U的负载测试主要分为三大类:数据传输负载测试、简单查询负载测试和复杂查询负载测试,其中还包括版本管理测试、异常文件检测测试、热点文件查询测试等具有特色的负载测试,满足用户不同层面的评测需求。

### 5. 多样化的结果展示

MBBench-U为用户提供了多样化的负载测试结果展示,不仅对结果做文档形式的保存,还将结果以图形、表格等形式展示出来,使结果一目了然,方便用户的整体把握。此外,MBBench-U还提供结果的版本管理以及结果的对比展示,提供评测结果对比报告。

# 1.3 手册介绍目标及读者人群

手册主要介绍工业大数据非结构化数据库系统评测工具——MBBench-U 如何进行安装、操作及二次开发。用户也可登录 MBBench-U 系统,到帮助文档界面查看 MBBench-U 的安装、使用等相关信息。手册针对的读者是使用 MBBench-U 的数据库评测人员以及普通开发者。

# 2. 负载场景及应用

# 2.1 工业大数据背景

工业大数据是指在工业领域中,围绕典型的智能制造模式,从客户需求到销售、订单、计划、研发、设计、工艺、制造、采购、供应、库存、发货和交付、售后服务、运维、报废或回收再制造等整个产品全生命周期各个环节所产生的各

类数据及相关技术和应用的总称。工业实际应用中,一次仿真实验很容易产生 100GB 或者几百 GB 的文件,这些仿真文件个数很多,很容易到上亿,甚至百亿,但是单个文件的容量很小。MBBench-U 系统从工业大数据管理的背景出发,针对高端制造业的数据特征与特点,以"仿真实验结果文件"为数据原型,以"海量小文件存储与管理"为主题,以高端制造业对非结构化数据管理的需要为应用场景,制定评测基准和测试工具。

# 2.2 负载描述

# 2.2.1 负载一: 数据传输

### 1. 负载应用

数据传输负载用于衡量数据库的上传和下载能力。在实际应用中,对于大批量工业数据的传输效率的测试十分重要。

### 2. 负载设计

将已生成好的仿真结果文件及元信息批量上传到数据库内, 衡量数据库上传能力。之后将数据库的数据批量下载到本地, 衡量数据库的下载能力。

### 3. 结果衡量

数据传输负载的结果会显示数据文件和元数据文件的上传下载速度、执行全部的用时以及单个文件的实时速度和每秒实时传输文件的个数。

# 2.2.2 负载二: 单属性查询

### 1. 负载应用

单属性查询属于文件管理系统中的基本操作,主要用于对文件中某一属性的 数值进行查询操作。在工业实际应用中,查看单个属性数值是最基础并且经常使 用的操作。

### 2. 负载设计

单属性查询共分为三条语句:一是对单个属性的单个数值进行查询;二是对单个属性的多个离散数值进行查询;三是对单个属性的范围数值(连续型数据)进行查询。查询的内容涵盖文件类型、地区、仿真时间等。

### 3. 结果衡量

单属性查询负载的结果会显示查询文件的数量、查询速度、用时、查询结果 正确率,以及下载单个文件的实时速度。

# 2.2.3 负载三:多属性查询

### 1. 负载应用

多属性查询主要针对文件中多个属性的数值进行查询操作。在实际应用中, 很多情况下需要获取多个属性的特定数据进行查看。

### 2. 负载设计

多属性查询共分为四条语句:一是对多属性的单个数值进行查询;二是对多属性的多个离散数值进行查询;三是对多属性的范围数值进行查询;四是对多属性的多种类型数值进行查询。查询内容包括文件类型、地区、塔架高度、扇叶长度等多种属性。

### 3. 结果衡量

多属性查询负载的结果会显示查询文件的数量、查询速度、用时、查询结果 正确率,以及下载单个文件的实时速度。

# 2.2.4 负载四: 版本管理

### 1. 负载应用

在实际应用中,一个数据文件会存在多版本的情况,版本管理是要对同一文件的不同历史版本进行保存,并且能够针对每个文件的所有版本做一些基本管理,例如对不同版本做时间上的排序等。

### 2. 负载设计

版本管理实现对每个文件所有版本的检测,并统计记录每个文件的版本数。

#### 3. 结果衡量

版本管理的结果显示文件的总数、查询速度、查询用时以及统计版本数的正确率,并返回每个文件对应的版本数。

# 2.2.5 负载五: 排序

### 1. 负载应用

排序操作属于文件系统管理中的通用操作,也是在工业上经常使用的操作之一。在工业实际应用中,向数据库内上传文件时文件的顺序不一定严格按照仿真生成时间存储,特备是在一个文件有多个版本的情况下文件是乱序存放的,因此在下载仿真文件时,需要按照一定顺序对文件排序。

### 2. 负载设计

排序负载总共分为三条语句,针对的数据范围为已上传的全部数据或部分数据。语句针对塔架高度、文件名称、仿真时间,分别对应数值排序、字符串排序和时间排序,以升序或降序执行排序操作。

### 3. 结果衡量

排序负载的结果会显示是否可执行排序操作、当前排序的数据量、排序速度以及排序产生结果的用时。

### 2.2.6 负载六:统计值查询

### 1. 负载应用

在工业数据中,很多数据都是批量生产的,比如风速数据,一个仿真结果文件中可能会生成上百个风速数据,对此类批量数据的统计值进行计算可以把批量数据刻画得更加简明。

### 2. 负载设计

统计值计算负载考量三个方面,分别是最大值、最小值与平均值计算。计算 的数据是每个仿真文件的风速数据,以数组的方式存储在数据库内。

### 3. 结果衡量

统计值计算负载的结果会显示当前计算的文件数量、计算速度、执行全部的 用时以及正确率。

# 2.2.7 负载七: 异常文件检测

### 1. 负载应用

在工业仿真实验中,由于参数及执行函数的设计,会出现小批量的异常数据,例如对于风速来说,风速的变化过于剧烈或者风速数据出现负数等都是不正常的。 对此,要求数据库系统能够对上传的文件进行异常数据检测,并对异常文件进行记录,例如对异常文件进行加标签的操作,或者对异常文件定期汇总等。

### 2. 负载设计

在评测系统中,异常文件检测的负载设计主要围绕异常文件的产生及标记。 这里拟选取风速来作为异常数据制造属性,通过加大瞬变风因子的倍数从而制造 异常数据。而对于异常文件的标记,默认在数据生成时会让用户填写异常文件的 占比,并在生成数据时,将靠后的文件用异常数据填充。因此异常文件的数量即 为整体文件数量乘以异常文件占比的比值。

### 3. 结果衡量

异常文件检测负载是针对全体文件的监测,显示待检测的文件数量、检测的速度、执行全部的用时以及检测的准确率和召回率。

# 2.2.8 负载八: 热点文件查询

### 1. 负载应用

在工业上实际管理仿真数据时,有时会经常性地访问某些文件中的特定数据,如某时期的或者某一类型的数据,这类文件称为热点文件。如果数据库管理系统支持对热点文件的检索,自动地针对经常访问的文件做特殊存储,将会很大程度上减少文件查询的时间,提高工作效率。

### 2. 负载设计

针对热点文件的构造,拟定采用直接构造法。如对于某一类的查询在一段时间内以相同的时间间隔连续执行若干次,统计每次查询的用时。

### 3. 结果衡量

热点文件查询负载将比对每次查询的用时,评断其执行时间趋势。若为下降

的则表明该数据库管理系统支持热点文件查询; 否则为不支持。

# 3. MBBench-U 安装

# 3.1 运行环境

本评测系统是基于 Windows7 开发,采用 Eclipse 作为开发工具,以 Maven 的 WebApp 为程序框架搭建的工业大数据非结构化数据库系统评测工具。安装 MBBench-U 前需要保证 Windows 操作系统,设备上要确认配有 Java 运行环境, JDK 版本在 1.8 及以上,并且需要安装、配置 Maven 和 Tomcat。硬件要求: CPU 要求 Core I5 2300 及以上的处理器,内存至少 4GB,硬盘空间(包括系统环境配置及文件生成和下载)至少 500GB。

# 3.2 安装环境

# 3.2.1 Java 安装说明

本评测系统是基于 Java 8 开发,使用者需要保证运行系统内安装 Java 8 及以上版本。具体安装方式与环境配置请使用者在官网查询。

# 3.2.2 Tomcat 安装说明

本系统在使用中采用 Java Web 进行图形界面操作,需要将系统程序在Tomcat 环境下运行。使用者需要保证运行系统安装 Tomcat,最好是 Tomcat7及以上版本。具体安装方式与环境配置请使用者在官网查询。

说明:请保证 Tomcat 的安装路径中不存在空格,否则系统运行中会出现文件路径获取错误的情况。

# 3.2.3 Maven 安装说明

本系统的安装过程需要使用 Maven 作为工程组件,用于自动构建运行程序并便于开发者对本系统进行二次开发。使用者需要保证运行系统安装 Maven,最好是 Maven 3 版本。

由于系统内有一些 Java 8 的新特性语句,因此需要修改 Maven 的配置文件,将 Java 的编译版本改成 1.8,并且编译编码设定为"UTF-8",否则程序不

## 3.3 组件步骤

从 GitHub 上下载 MBBench-U 的压缩软件包 MBBench-U.zip, 压缩包包含了 MBBench-U 系统运行所需要的所有必要组件,解压后得到 MBBench-U 文件。打开命令行,在 MBBench-U 文件的根目录下运行 Maven 程序: mvn package,出现"BUILD SUCCESS"表示运行成功。

```
INFO] --- maven-war-plugin:2.2:war (default-war) @ MBBench-U ---
INFO] Packaging webapp
INFO] Assembling webapp [MBBench-U] in [D:\GitHub\MBBenchU\target\MBBench-U]
INFO] Processing war project
INFO] Copying webapp resources [D:\GitHub\MBBenchU\src\main\webapp]
INFO] Webapp assembled in [335 msecs]
INFO] Webapp assembled in [335 msecs]
INFO] WEB-INF\web.xml already added, skipping
INFO] WEB-INF\web.xml already added, skipping
INFO] Total time: 6.639 s
INFO] Total time: 6.639 s
INFO] Total time: 6.639 s
INFO] Finished at: 2018-04-10T20:32:13+08:00
```

图 3.1 运行 Maven 成功

成功运行 Maven 后会在 MBBench-U 文件的根目录下生成 target 文件,找到 target 文件目录下的 MBBench-U.war 文件,将其复制到 Tomcat 的 webapps 文件下,之后启动 Tomcat。打开浏览器,在网址栏中输入

http://localhost:8080/MBBench-U(注意:8080 为 Tomcat 的默认端口号,若您的设备中修改了端口号,此处应该是您设备中 Tomcat 的端口号),按下回车,显示 MBBench-U系统的系统说明界面,如图 3.2 所示,表示 MBBench-U 安装成功。



图 3.2 MBBench-U 系统说明界面

# 4. 操作指南

# 4.1 数据生成

进行数据生成之前先选择待评测的数据库,在系统说明界面进行设置。点击 选择想要评测的数据库名称,然后输入负载结果存放路径,点击"确认信息", 界面右上角出现"设置成功"提示则设置成功。

MBBench-U 的数据生成方式总共有两种,一种是初始化生成:通过修改 XML 配置文件及在海量文件生成界面上填写相应参数生成初始数据;另一种是历史记录生成:根据历史参数配置记录文件生成更新数据。

# 4.1.1 初始化生成

修改 XML 配置文件。

用户可以在配置文件 peizhi.xml 中设置产生数据的个数、相关风速的时间, 并添加或修改地区的名称、该地区的基本风速、最大渐变风风速等数值。 peizhi.xml 的路径为: MBBenchU>target>classes>conf。修改示例如图 4.1 所示。

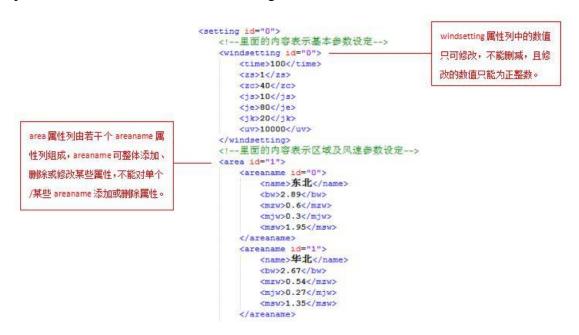


图 4.1 XML 配置文件

XML 配置文件说明: windsetting 属性列内,从上到下的数值依次表示: 数据个数、阵风开始时间、阵风持续周期、渐变风渐变开始时间、渐变风渐变结束时间、渐变风持续时间及当前输送电线的电压值。areaname 属性列内,从上到下的数值依次表示: 区域名称、区域平均风速、区域阵风风速峰值、区域渐变风风速峰值和区域瞬间风速峰值。

在海量文件生成界面上设置参数。

首先对文件存储路径、设备数量、仿真文件参数等基本参数进行设置。输入文件存储路径,并选择元数据文件的类型,有三种文件类型供用户选择,分别是TXT、JSON和XML,至少选择其中一种作为元数据仿真分析结果文件的数据文件。然后输入仿真设备数量,建议设置的数值不要过小(建议>=100),增加文件属性的多样。设置每一个仿真设备所产生的文件数量及更新文件数量,注意更新文件数量不可超过文件数量,后面每次在生成更新数量时也是如此。输入异常文件占比,表示生成的数据文件中有异常数据的文件所占比例。最后输入仿真设备的三个参数范围设置,分别是塔架高度、扇叶长度和扇叶宽度。这里面的数值都是正整数,并且两个正整数之间用"一"分割。图 4.2 是设置基本参数的一种示例。



图 4.2 基本参数设置示例

之后是仿真文件大小的设置。在"设置文件分配数量"下的文本框内输入一个正整数,表示文件大小的分配数量,点击"生成设置文件"按钮弹出设置栏,设置栏内依次输入文件大小及所占比例,这些数字均可为非负且非零的浮点数。图 4.3 是设置仿真文件大小的一种示例。

### 仿真文件大小设置



图 4.3 仿真文件大小设置示例

接着进行仿真时间的设置。仿真日期及时间的设置必须为符合实际意义的正整数。图 4.4 是设置仿真时间的一种示例。

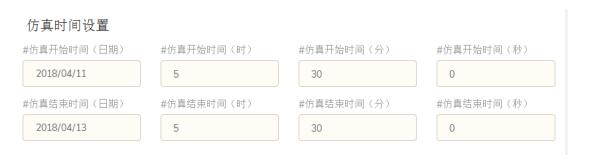


图 4.4 仿真时间设置示例

至此,生成文件的参数设置完成。点击"刷新信息"按钮,在右侧的"生成文件信息"框架内会显示当前生成文件的数量及预估文件大小,示例如图 4.5 所

示。



图 4.5 生成文件信息示例

设置的信息确认无误后,点击"生成文件"按钮,文件即可生成,图 4.6 是文件生成过程示例。



图 4.6 文件生成过程示例

# 4.1.2 历史记录生成

每一次配置数据文件时都会在"HistorySettings"目录下存储生成记录,包括仿真设备数量、文件数量、文件大小及分布情况等。用户可在海量文件生成界面下,点击"使用已有的配置"下的"使用"按钮,即可获取最新的五次配置信息的历史记录,如图 4.7 所示。用户可选择一个历史版本,并选择按照此版本的配置信息生成新的文件。其中"按照原设置文件数量继续生成"选项表示生成文件数量的参数使用历史版本中的设置;"按照新设置的文件数量生成"选项表示用户可重新设置生成文件的数量,设置示例如图 4.8。

#### 过去的配置信息

当前已有的仿真配置文件

Simulation Device Num: 100
Simulation Start Time: 2018-03-27 0:0:0
Simulation End Time: 2018-03-28 0:0:0
Each Wind Based Varables: 100 10000 1.0 40.0 10.0 80.0 20.0
The 1 of Device Varables: 3 49 78 33 XiNan 1.65 0.34 0.17 0.1
The 2 of Device Varables: 5 47 75 26 QingZangGaoYuan 2.51 0.48 0.24 0.3
The 3 of Device Varables: 5 43 75 36 QingZangGaoYuan 2.51 0.48 0.24 0.3
The 4 of Device Varables: 0 53 76 38 DongBei 2.89 0.6 0.3 0.5
The 5 of Device Varables: 3 48 75 38 XiNan 1.65 0.34 0.17 0.1

选择数据生成内容

● 按照原设置文件数量继续生成 ● 按照新设置的文件数量生成



图 4.7 已有配置文件信息

选择数据生成内容

◎ 按照原设置文件数量继续生成 ◎ 按照新设置的文件数量生成

生成文件数量

1000 文件路径为前面已设置的内容 不可超过文件内的文件生成设置数

图 4.8 按照新设置文件数量生成示例

设置完成后点击"确认信息",弹出信息确认框,如图 4.9 所示。信息确认 无误后就可以生成文件了,点击"生成文件"按钮,完成文件生成即可。



图 4.9 信息确认

# 4.2 负载测试

进行负载测试之前务必要先连接数据库,在数据库连接界面进行操作。根据用户数据库的实际情况填写网址、端口号、数据库名称和集合名称。注意数据库名称和集合名称不要使用中文。之后点击"连接数据库",界面右上角出现"数据库连接成功"提示表明连接成功。

数据库连接成功后,就可以进行负载测试了。MBBench-U的负载测试有三

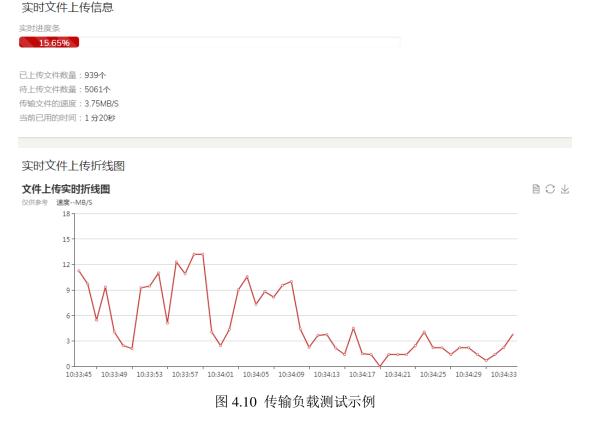
种:数据传输负载测试、简单查询负载测试和复杂查询负载测试。

# 4.2.1 数据传输负载测试

进行数据传输负载测试之前务必要新建传输结果文件夹,点击"创建文件夹"按钥即可。

注意:每一次进行数据传输负载测试时都要重新创建新的传输结果文件夹, 否则可能会导致传输测试结果出现混乱,影响结果查看!

之后依次进行数据上传、元数据上传、数据下载、元数据下载。在运行过程中可通过实时传输信息与实时传输折线图来查看传输情况。示例如图 4.10 所示。



4.2.2 简单查询负载测试

进行查询(简单查询/复杂查询)负载测试之前务必要新建负载结果文件夹, 点击界面右上侧的"创建文件夹"按钮即可。

注意:每次进行新一阶段的查询负载测试之前,需要创建新的负载结果文件夹,否则会导致负载测试结果出现重写问题,进而造成结果无法记录与显示。

简单查询负载测试包括单属性查询、多属性查询和数据文件版本管理。其中

单属性查询分单个数值查询、有限元数值查询、范围数值查询三个独立的单条查询;多属性查询分单个数值查询、有限元数值查询、范围数值查询、多类型数值查询四个独立的单条查询。负载测试时只需点击想要执行的查询负载下的"开始查询"按钮,相应的负载即可进行。图 4.11 以单属性查询为例展示了简单查询的执行过程。

单属性查询测试

单属性查询共分三个查询:单个数值查询,有限元数值查询,范围数值查询。						
1. 针对单一属性的单个数值查询:查询	1. 针对单一属性的单个数值查询:查询属性:文件类型;查询值:BLADE					
2. 针对单一属性的有限元数值查询:查询	间属性:地区;查询值:XiBei、XiNan、	QingZangGaoYuan				
3. 针对单一属性的范围数值查询:查询	属性:仿真时间;查询值:请设定					
#仿真开始时间(日期)						
请确认数据库已连接成功,如果未连接或者连接失败,请返回数据库连接重新操作						
开始宣词 Single Attribute Query is Doing, Please wait						

图 4.11 单属性查询测试

若动态显示栏显示"Single attribute query has been finished"则表示此查询负载测试完成。

# 4.2.3 复杂查询负载测试

复杂查询负载测试包括排序负载测试、最值及均值负载测试、异常文件检测和热点文件查询。其中排序负载测试分数值型排序、字符串型排序、时间型排序三个独立的单条排序负载;最值及均值负载测试分最大值查询、最小值查询、平均值查询三个独立的单条查询负载。负载测试时只需点击想要执行的查询负载下的"开始查询"按钮,相应的负载即可进行。图 4.12 以最值及均值负载测试为例展示了复杂查询的执行过程。

#### 最值及均值负载测试



图 4.12 最值及均值负载测试

若动态显示栏显示"Satistics query has been finished"则表示此负载测试完成。

需要注意的是,热点文件查询中需要提前设置查询次数和每次查询的间隔时间,查询次数建议不能过少,每次查询间隔时间单位是毫秒,数据类型为整型。图 4.13 是热点文件查询设置示例。

热点文件查询负载测试



图 4.13 热点文件查询设置示例

# 4.3 结果显示

结果显示部分包括系统信息显示以及负载结果显示两大部分,负载结果有数据传输负载结果、简单查询结果、复杂查询结果三部分。在结果查看界面可以看到系统信息,展示了 JDK 信息、系统基本信息、CPU 信息、磁盘信息等设备硬软件资源信息,作为用户的一种参考。点击界面右上角的"负载结果"可以查看负载结果。图 4.14 为传输结果显示的版本管理界面,前面表示运行负载的日期,后面的数字表示这一天内不同的负载结果,按照执行时间排序。



图 4.14 传输结果版本管理

在简单查询结果信息与负载结果信息界面上点击"查询测试版本"出现不同版本的数据传输下对应的不同版本的查询测试,如图 4.15 所示。每一个查询结果都放在相应传输负载测试下,对于未进行查询负载测试的不进行结果的输出。用户可以查看所有历史版本的查询负载测试结果。



图 4.15 查询结果版本管理

# 4.3.1 数据传输负载结果

点击结果显示界面右上角"负载结果"下的"数据传输负载结果"转到数据传输结果信息界面,用户可以查看数据库基本信息,如图 4.16 所示,以及传输测试基本信息,如图 4.17 所示。



图 4.16 数据库基本信息

传输	传输测试基本信息					
编号	号	传输名称	传输用时	传输速度		
1		上传数据文件	1H:5M:33S	9.356718550982016Mb/s		
2		上传元数据文件	4M:12S	190.4762per/s		
3		下载数据文件	23M:9S	26.493861814983635MB/s		
4		下執元数据文件	2M:10S	369.23077per/s		

图 4.17 传输测试基本信息

为了方便用户对整个传输过程有整体的把握,系统对单位时间(秒)的传输速度进行记录,并以折线图的形式显示,其中文件传输的速度单位为 MB/s,文档传输的速度单位为 per/s。图 4.18 以文件上传结果为例展示了数据传输结果折线图。为了方便用户对整个传输情况做整体把握,系统对传输速度数据做了统计计算的处理,分别以 5、10、20、50、100 个单位时间点做统计平均数处理为一个点,用户可选择不同阈值的单位点观察数据传输情况。



图 4.18 数据传输结果折线图示例

数据传输折线图具有局部展示的功能,拉动折线图下方的坐标尺,可随意调节折线图中显示的数据范围,方便用户观察。图 4.19 是对图 4.18 的折线图进行显示范围调节后的一个示例。

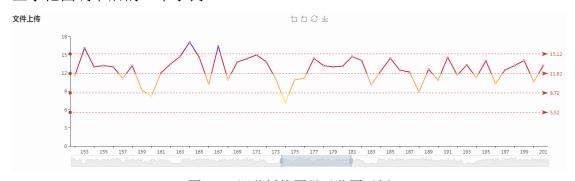


图 4.19 调节折线图显示范围示例

# 4.3.2 简单查询负载结果

简单查询负载包括单属性查询、多属性查询和版本管理查询,图 4.20 是以单属性查询结果信息为例的一个简单查询结果示例。结果查看部分有局部刷新的功能,用户首先选择想要查看的结果版本,然后点击结果信息右下角的"刷新信息"就可获得相对应的结果,局部刷新功能可支持单类负载测试结果的刷新显示,用于当用户只做了部分负载测试时的结果查看。

系统会将简单查询的查询结果进行下载保存,点击结果信息右下角的"数据下载结果查看"可查看下载速度的折线图。折线图的操作与 4.3.1 中相同。

单属	性查询结果信息				
编号	查询类型	查询条数[条]	查询速度[条/毫秒]	查询用时[毫秒]	查询正确率
1	单一属性单个数值查询 查词属性:文件类型;查询值:Blade	2000	0.7823834	2556.2915	100.00%
2	单一属性离散数值查询 查询属性:地区;查询值:XiBei、XiNan、QingZangGaoYuan	2760	7.1870008	384.02667	100.00%
3	单一属性连续数值查询 查询属性: 仿真时间; 查询值: 2018-06-21 0:0:02018-06-21 6:0:0	708	1.2112309	584.52936	
数据	下载结果查看				刷新信息

图 4.20 简单查询结果示例

# 4.3.3 复杂查询负载结果

复杂查询负载包括排序、统计值计算、异常文件检测和热点文件查询。其中排序、统计值计算和异常文件检测的结果显示与简单查询相同,具体操作参见4.3.2。热点文件查询以每次查询用时作为负载测试结果,并用折线图展示其执行时间趋势,图 4.21 是热点文件查询结果的一个示例。

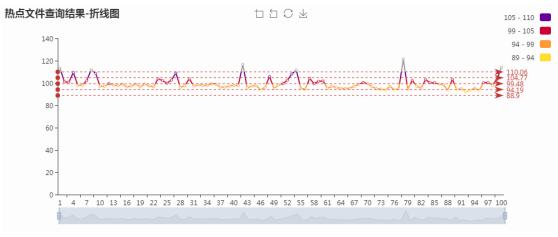


图 4.21 热点文件查询用时折线图

# 4.4 结果对比

结果对比主要是对不同数据库的评测结果进行对比,得出有一定结论性的结果对比报告。

注意:进行结果对比之前请确认是否正确填写负载结果存在路径,否则,在"系统设置"界面设置。

首先需要用户手动填写测试数据库的基本运行环境,包括机器数量以及机器的硬件参数。图 4.22 是设置运行环境的示例。



图 4.22 运行环境设置示例

之后设置对比结果信息,包括测试数据库名称以及传输和查询测试的版本, 对比结果信息设置示例如图 4.23 所示。

本页面主要用于结果对比的选择,请按照步骤与过程正确选择相应的结果内容 如果没有内容,请返回系统设置页面输入结果日志



图 4.23 结果对比信息设置示例

所有信息设置好后,点击"生成结果对比报告"按钮即可生成结果对比的报告。结果对比报告包括基本运行环境、数据库存储信息、传输结果信息以及查询结果信息等几个部分的对比结果,以表格、图形等多种形式展示,并且在每个部分的结果对比后附有结论性的总结,方便用户直观地了解评测结果。

基本信息对比如下图 4.24 所示:



基本运行环境





# 基本信息对比



图 4.24 基本信息对比显示

传输结果对比如下图 4.25、4.26 所示:

### 传输结果对比【表格】

编号	对比内容	数据库—ElasticSearch	数据库二MongoDB
1	文件上传	Time: 48 : 17 Speed: 1.3432507549518957Mb/s	Time: 6 : 57 Speed: 9.331888338358853Mb/s
2	文档上传	Time: 15 : 53 Speed: 7.8698845per/s	Time: 57 Speed: 131.57895per/s
3	文件下载	Time: 5 : 39 Speed: 11.479048487007793MB/s	Time: 2 : 14 Speed: 29.04027938131076MB/s
4	文档下载	Time: 30 Speed: 250.0per/s	Time: 10 Speed: 750.0per/s

#### - 数据传输对比结果总结 -

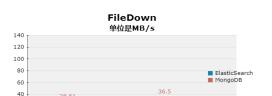
在文件上传上,MongoDB的传输速度比ElasticSearch的传输速度快2480秒,相对效率而言,MongoDB是ElasticSearch的14.39%在文档上传上,MongoDB的传输速度比ElasticSearch的传输速度快896秒,相对效率而言,MongoDB是ElasticSearch的59.98%在文件下载上,MongoDB的传输速度比ElasticSearch的传输速度快205秒,相对效率而言,MongoDB是ElasticSearch的39.53%在文档下载上,MongoDB的传输速度比ElasticSearch的39.53%在文档下载上,MongoDB的传输速度比ElasticSearch的传输速度快20秒,相对效率而言,MongoDB是ElasticSearch的633.33%

图 4.25 传输结果表格对比显示

### 传输结果对比【图表】

20

#### 文件上传对比图 FileLoad 单位是MB/s 140 120 100 80 ■ ElasticSearch ■ MongoDB 60 40 20



文件下载对比图

#### 文件上传对比结果总结

在文件上传的平均速度上, MongoDB数据库的传 输速度比ElasticSearch数据库的传输速度快,在文件上传效率上MongoDB数据库是ElasticSearch数 据库的670.64% 在文件上传的速度稳定性上,MongoDB数据库的

相较于ElasticSearch数据库来说更不稳定,其浮动比率MongoDB数据库是ElasticSearch数据库的

149.68% 在文件上传的速度峰值差上,MongoDB数据库的 相较于ElasticSearch数据库来说波峰与波谷的差距更大,MongoDB数据库是ElasticSearch数据库差 距的457.69%

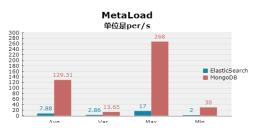
#### 文件下载对比结果总结

在文件下载的平均速度上, MongoDB数据库的传 输速度比ElasticSearch数据库的传输速度快,在文件下载效率上MongoDB数据库是ElasticSearch数

据库的210.29% 在文件下载的速度稳定性上,ElasticSearch数据库的相较于MongoDB数据库来说更不稳定,其浮动比率ElasticSearch数据库BMongoDB数据库的

129.35% 在文件上传的速度峰值差上,MongoDB数据库的 相较于ElasticSearch数据库来说波峰与波谷的差距更大, MongoDB数据库是ElasticSearch数据库差 距的160.45%

### 文档上传对比图



### 文档下载对比图

#### MetaDown 单位是per/s



#### 文档上传对比结果总结

在文档上传的平均速度上, MongoDB数据库的传 输速度比ElasticSearch数据库的传输速度快,在文档上传效率上MongoDB数据库是ElasticSearch数

据库的1641.38% 在文档上传的速度稳定性上,MongoDB数据库的 相较于ElasticSearch数据库来说更不稳定,其浮动比率MongoDB数据库是ElasticSearch数据库的

在字MongoDB数据库是HasticSearch数据库的477.20%在文件上传的速度峰值差上,MongoDB数据库的相较于ElasticSearch数据库来说波峰与波谷的差距更大,MongoDB数据库包含证式Search数据库差

#### 文档下载对比结果总结

在文档下载的平均速度上, MongoDB数据库的传 输速度比ElasticSearch数据库的传输速度快,在文档下载效率上MongoDB数据库是ElasticSearch数

指序的281.82% 据库的281.82% 在文档下载的速度稳定性上,MongoDB数据库的 相较于ElasticSearch数据库来说更不稳定,其浮动 比率MongoDB数据库是ElasticSearch数据库的

138.67% 在文件上传的速度峰值差上,MongoDB数据库的相较于ElasticSearch数据库来说波峰与波谷的差距 更大, MongoDB数据库是ElasticSearch数据库差

图 4.26 传输结果图形对比显示

单属性多属性查询结果对比如下图 4.27 所示:

### 单、多属性查询结果对比

	查询条数【条】		查询速度【条/MS】		查询用时【MS】	
编 查询内容 号	数据库一 ElasticSearch	数据库二 MongoDB	数据库一 ElasticSearch	数据库二 MongoDB	数据库— ElasticSearch	数据库二 MongoDB
单一属性单个数值查 间	2500	2500	4.10496	4.8467383	609.01935	515.8108
单一属性离散数值查 询	3750	3750	3.758501	10.755094	997.73816	348.672
单一属性连续数值查 询	3756	3756	5.80269	10.887711	647.286	344.9761
多属性单个数值查询	400	400	6.9754844	5.3682475	57.34369	74.512215
多属性离散数值查询	1300	1300	3.0655406	5.5889764	424.06876	232.60072
多属性的连续数值查询	6150	6150	3.1797225	10.690778	1934.131	575.26215
多属性的多类型数值查询	1250	1250	6.7618136	11.022437	184.86165	113.405045
	单一属性单个数值查 词 单一属性离散数值查 词 单一属性连续数值查 词 多属性单个数值查询 多属性的连续数值查询 多属性的连续数值查	查询内容     数据库一-ElasticSearch       单一属性单个数值查     2500       单一属性离散数值查     3750       单一属性连续数值查     3756       多属性单个数值查询     400       多属性的连续数值查     1300       多属性的连续数值查     6150       多属性的多类型数值     1250	查询内容     数据库 ElasticSearch     数据库 MongoDB       单-属性单个数值查 词     2500     2500       单-属性离散数值查 词     3750     3750       學-属性连续数值查 词     3756     3756       多属性单个数值查词     400     400       多属性的连续数值查 词     6150     6150       多属性的多类型数值     1250     1250	查询内容     数据库ElasticSearch     数据库ElasticSearch       単一属性单个数值查询     2500     2500     4.10496       単一属性常数值查询     3750     3750     3.758501       単一属性连续数值查询     3756     3756     5.80269       多属性单个数值查询     400     400     6.9754844       多属性的连续数值查询     1300     1300     3.0655406       多属性的连续数值查询     6150     6150     3.1797225       多属性的多类型数值     1250     6.7618136	查询内容     数据库—- ElasticSearch     数据库—- MongoDB     数据库—- ElasticSearch     数据库—- MongoDB       单一属性单个数值查 询     2500     2500     4.10496     4.8467383       单一属性离散数值查 询     3750     3750     3.758501     10.755094       单一属性连续数值查 询     3756     3756     5.80269     10.887711       多属性单个数值查询     400     400     6.9754844     5.3682475       多属性的连续数值查询     1300     1300     3.0655406     5.5889764       多属性的连续数值查询     6150     6150     3.1797225     10.690778       多属性的多类型数值     1250     6.7618136     11.022437	查询内容     数据库——ElasticSearch     数据库——ElasticSearch     数据库——ElasticSearch     数据库——ElasticSearch       单一属性单个数值查询     2500     2500     4.10496     4.8467383     609.01935       单一属性有散数值查询     3750     3750     3.758501     10.755094     997.73816       单一属性连续数值查询     3756     3756     5.80269     10.887711     647.286       多属性单个数值查询     400     6.9754844     5.3682475     57.34369       多属性单次值查询     1300     1300     3.0655406     5.5889764     424.06876       多属性的连续数值查询     6150     6150     3.1797225     10.690778     1934.131       多属性的多类型数值     1250     6.7618136     11.022437     184.86165

#### 单、多属性查询结果对比总结

在单属性单数值查询负载上,MongoDB数据库的查询速度比ElasticSearch数据库要快,是其查询速度的118.07% 在单属性离散数值查询负载上,MongoDB数据库的查询速度比ElasticSearch数据库要快,是其查询速度的286.15% 在单属性连续数值查询负载上,MongoDB数据库的查询速度比ElasticSearch数据库要快,是其查询速度的187.63% 在多属性单数值查询负载上,ElasticSearch数据库的查询速度比MongoDB数据库要快,是其查询速度的129.94% 在多属性离散数值查询负载上,MongoDB数据库的查询速度比ElasticSearch数据库要快,是其查询速度的182.32% 在多属性连续数值查询负载上,MongoDB数据库的查询速度比ElasticSearch数据库要快,是其查询速度的336.22% 在多属性多类型数值查询负载上,MongoDB数据库的查询速度比ElasticSearch数据库要快,是其查询速度的

# 图 4.27 单、多属性查询结果对比显示版本查询结果对比如下图 4.28 所示:

### 版本查询结果对比

编号	对比内容	数据库一ElasticSearch	数据库二MongoDB
1	是否支持	null	null
2	查询条数	0	0
3	响应时间	0	0
4	查询速度	0	0
5	查询正确率	0	0

版本管理结果对比总结 -

ElasticSearch数据库不支持版本管理负载。MongoDB数据库不支持版本管理负载。

图 4.28 版本查询结果对比显示

排序、统计值计算查询结果对比如下图 4.29 所示:

排序	统计值	计算查询结果	741FV
JHE/J/	シルレコロ	刀开旦四和不	ンフレロ

编	查询条数【条】				【条/MS】   查询用时【MS		5]	
<del>细</del> 号		数据库— ElasticSearch	数据库二 MongoDB	数据库— ElasticSearch	数据库二 MongoDB			
1	数值排序	7500	7500	4.5253267	3.9571471	1657.3389	1895.3048	
2	字符串排序	7500	7500	4.184906	8.892284	1792.1549	843.42786	
3	时间排序	7500	7500	5.740929	9.832279	1306.4087	762.79364	
4	平均值计算	null	2500	null	0.37461597	null	6673.501	
5	最小值计算	null	2500	null	20.66538	null	120.97527	
6	最大值计算	null	2500	null	24.916601	null	100.33471	

- 排序、统计值计算结果对比总结 -

在数值排序负载上,ElasticSearch数据库的查询速度比MongoDB数据库要快,是其查询速度的114.36%在字符串排序负载上,MongoDB数据库的查询速度比ElasticSearch数据库要快,是其查询速度的212.48%在日期排序负载上,MongoDB数据库的查询速度比ElasticSearch数据库要快,是其查询速度的171.27%ElasticSearch数据库不支持平均值计算负载。ElasticSearch数据库不支持最小值计算负载。ElasticSearch数据库不支持最小值计算负载。

# 图 4.29 排序、统计值计算查询结果对比显示 异常文件检测结果对比如下图 4.30 所示:

### 异常文件检测结果对比

编号	对比内容	数据库—ElasticSearch	数据库二MongoDB
1	是否支持	null	null
2	查询条数	null	null
3	响应时间	null	null
4	查询速度	null	null
5	找到异常文件[条]	null	null
6	实际异常文件[条]	null	null
7	正确率	null	null

—— 异常文件检测策结果对比总结 ————————

ElasticSearch数据库不支持异常文件检测负载。MongoDB数据库不支持异常文件检测负载。

图 4.30 异常文件检测结果对比显示 热点文件查询结果对比如下图 4.31 所示:

#### 数据库二-MongoDB 数据库一-ElasticSearch 热点文件查询结果-折线图 热点文件查询结果-折线图 1104 110 L 1323 - 1511 1135 - 1323 720 - 771 947 - 1135 669 - 720 759 - 947 × 1134.98 400 5 9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69 73 77 81 85 89 93 97 5 9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69 73 77 81 85 89 93 97

### 热点文件结果对比

图 4.31 热点文件查询结果对比显示

# 5. 二次开发

MBBench-U 支持用户进行二次开发,用户可以按照我们指定的操作步骤完成对新的数据库的评测。首先在打开 MBBench-U 的项目文件 MBBench-U 文件夹,找到 pom.xml 文件,向其中添加新的数据库所需 jar 包的相关信息。然后找到 dbfunction 文件夹(MBBench-U>src>main>java>cn>ruc>edu>dbfunction),在 dbfunction 文件夹下添加新的数据库的执行类文件,如: xx.java。该 Java 类文件需要继承 basecore 文件夹下的 Java 文件 UnStrDBBase.java(MBBench-U>src>main>java>cn>ruc>edu>basecore),并且实现给定的所有函数方法,具体实现参考已有类文件。

最后在系统说明页面 sysinfo.html (MBBench-U>target>MBBench-U) 中添加新数据库的有关信息(注意: 在基本设置中添加新数据的名称信息时, value 值必须与该数据库的 Java 类文件名相同)。图 5.1 是 sysinfo.html 中基本设置的部分代码示例。

图 5.1 基本设置部分代码示例

# 5.1 关于结果验证接口的使用

本测试工具会对测试的结果进行验证,因此需要在二次开发过程中进行相应接口的引用与实现。当然,若用户觉得无需验证结果可跳过这部分内容。

结果验证接口名称为"CheckResult",其内置了三个需要实现的验证方法,分别是条件查询结果验证、统计值计算结果验证和排序结果验证。另外,在"UnstrDBbase"文件内置了关于版本管理结果验证和异常文件结果验证的方法,用户直接继承该类、使用相应方法即可。下面分别介绍每个方法的说明及使用事例。

# 5.1.1 版本管理结果检测

- (1) 方法: public int[] CopyCheck(String datapath, HashMap<String, Integer> resulthm)
- (2) 参数说明: datapath 为生成数据文件的总路径; resulthm 为查询结果的 key-value 对,其中 key 为文件的名称, value 为文件的个数。
- (3)返回结果:返回一个数组,其中内容包括检测到的不同文件数量、正确的数量和错误的数量。
  - (4) 实例:

# 5.1.2 异常文件结果检测

- (1) 方法: public int[] AbnormalCheck(String datapath, ArrayList<String> abnormalFileName)
- (2) 参数说明: datapath 为生成数据文件的总路径; abnormalFileName 为异

常文件列表,内容由"文件名称" + ":" + "文件版本"。

- (3)返回结果:返回一个数组,其中内容包括检测到的异常文件数量、真正异常的文件数量和正确的异常文件数量。
  - (4) 实例:

```
ArrayList<String> resulthm = new ArrayList<>();
  for (int i = 0; i < filelist.size(); i++) {
          DBObject obj = (DBObject ) filelist.get(i);
          String file = (String) obj.get("FILENAME");
          int copy = (Integer ) obj.get("CopyRight");
          resulthm.add(file + " : " + copy);
        }
    int[] result = AbnormalCheck(datapath, resulthm);</pre>
```

# 5.1.3 条件查询结果检测

- (1) 方法: public int CheckQuery(Iterator〈DBObject〉 result,
  Map〈String, Object〉 queryMap)
- (2) 参数说明: result 为查询结果的迭代器; queryMap 为查询条件的组合, 其中 key 表示查询数据的属性名称, value 表示查询相对应属性的条件 value 的 类型有如下:
  - (1) ArrayList:表示该属性的数值符合列表内的任一数值。
  - (2) 字符串带 "-": 表示数值的范围查询。
  - (3) 单个数值对象:表示该属性的数值符合这个值。

map 内的每一个 key-value 对之间的关系为"和"关系, 表示结果需要满足其内的所有条件。

- (3) 返回结果:返回一个数值,表示其结果符合条件的数量。
- (4) 实例:

```
ArrayList<String> fileList = new ArrayList<>();
ArrayList<String> queryContentList = new ArrayList<>();
queryContentList.add("XiBei");
queryContentList.add("XiNan");
queryContentList.add("QingZangGaoYuan");
Map<String, Object> queryMap = new HashMap<>();
queryMap.put("AREANAME", queryContentList);
int rightCount = this.CheckQuery(dbCursor.iterator(), queryMap);
```

### 5.1.4 统计值计算查询结果检测

- (1) 方法: public int CheckStatistics(Iterator〈DBObject〉 result, String statisticsType, String queryContent, String contentName)
- (2) 参数说明: result 为查询结果的迭代器; statisticsType 为统计值计算的内容,如平均值、最大值和最小值; queryContent 为数据内针对统计值计算的属性: contentName 表示结果内对计算结果的属性名称。
  - (3) 返回结果:返回一个数值,表示其结果符合条件的数量。
  - (4) 实例:

```
DBObject unwind = new BasicDBObject ("$unwind", "$WINDSPEED");

DBObject groupFields = new BasicDBObject ("_id", "$_id");

groupFields.put ("AvgSpeed", new BasicDBObject ("$avg", "$WINDSPEED"));

DBObject group = new BasicDBObject ("$group", groupFields);

AggregationOutput output = dbCollection.aggregate (unwind, group);

count = output.results().toString().split(", ").length/2;

int rightCount = CheckStatistics (output.results().iterator(), "AVG",
"WINDSPEED", "AvgSpeed");
```

# 5.1.5 排序负载结果检测

- (1) 方法: public String CheckSorted(Iterator (DBObject) result, String query, String type, String sortWay)
- (2) 参数说明: result 为查询结果的迭代器; query 为针对数据的排序属性 名称; type 为排序属性的数据类型; sortWay 表示排序的方式, 其中 "<" 表示升序, ">"表示降序。
  - (3) 返回结果:返回一个字符串,表示其结果是否符合该顺序。
  - (4) 实例:

String checkSorted = CheckSorted (dbCursor.iterator (),

"SIMULATIONTIME", "String", "<");

# 6. 故障排除与错误信息

# 6.1 组件过程问题

当使用 Maven 进行编译的时候,由于里面采用的一些包在镜像仓库内可能 没有,会导致运行过程中出现图 6.1 的情况:

图 6.1 运行 Maven 出错

请到 MBBench-U 文件下找到"构建项目时可能会用到的包"文件夹,将里面的 jar 包放到 Maven 本地仓库相关路径下,然后再执行即可运行成功。

# 7. 附录一: 测试结果文档详细说明

本系统负载测试的所有结果均可在系统中结果查看界面显示,并且所有结果数据都以文档形式保存,存储路径为系统设置界面"负载结果存放路径"中设置的路径,不同数据库的结果数据存放在以数据库名称命名的文件夹中,之后按照负载测试的时间存放结果数据。结果数据存放路径示意图如下图 7.1 所示:



图 7.1 结果日志存储路径显示

负载测试结果包括两部分:传输负载测试和查询负载测试。传输负载测试共有7个结果文档,查询负载测试共有23个结果文档,其中单属性、多属性查询

的文件下载结果文档有7个,以"down"开头命名;查询负载结果文档有16个,以"query"开头命名,包括:单属性查询(3个)、多属性查询(4个)、排序查询(3个)、统计值查询(3个)、版本查询、异常文件查询、热点文件查询。负载测试结果文档示意图如下图7.2 所示:

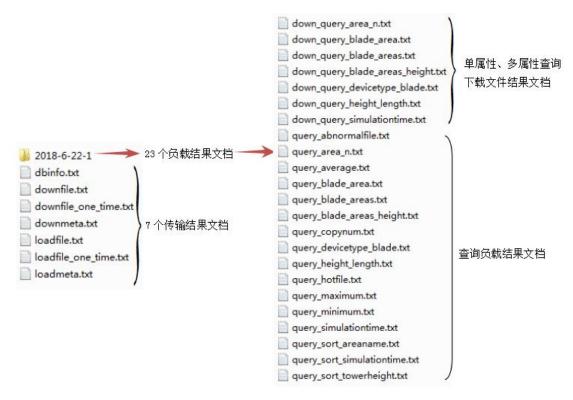


图 7.2 结果日志内部逻辑图

查询负载测试结果文档以 ket-value 对的形式保存结果数据, key 表示结果数据的类别, value 表示此类别的具体数值, ket-value 对之间以";"分隔,如:count;2000 表示查询数量为 2000。查询负载测试结果文档的详细说明以及结果文档与系统"结果查看"界面中结果展示的对应情况如下:

- 1. loadfile\_one\_time.txt: 保存二进制文件上传中每个文件上传的平均速度,单位: MB/s。
- 2. downfile\_one\_time.txt:保存二进制文件下载中每个文件下载的平均速度,单位:MB/s。
- 3. dbinfo.txt: 保存数据库基本信息以及传输测试基本信息,对应系统中结果 查看的部分如下图 7.3 所示:



图 7.3 基本信息显示

4. loadfile.txt: 保存二进制文件上传中每秒钟上传文件的速度,单位: MB/s,对应系统中结果查看的部分如下图 7.4 所示:



图 7.4 文件上传速度折线图

5. loadmeta.txt:保存元数据文档上传中每秒钟上传文档的个数,单位:per/s,对应系统中结果查看的部分如下图 7.5 所示:



图 7.5 文档上传速度折线图

6. downfile.txt:保存二进制文件下载中每秒钟下载文件的速度,单位:MB/s,对应系统中结果查看的部分如下图 7.6 所示:



图 7.6 文件下载速度折线图

7. downmeta.txt:保存元数据文档下载中每秒钟下载文档的个数,单位:per/s,对应系统中结果查看的部分如下图 7.7 所示:



图 7.7 文档下载速度折线图

- 8. 关于条件查询结果,文件有如下七个:
  - (1) query\_devicetype\_blade.txt
  - (2) query\_area\_n.txt
  - (3) query\_simulationtime.txt
  - (4) query\_blade\_area.txt
  - (5) query\_blade\_areas.txt
  - (6) query\_height\_length.txt
  - (7) query\_blade\_areas\_height.txt

以上的文件分别保存查询负载一到查询负载七的结果数据,具体包括count(查询数量)、time(查询用时)、speed(查询速度)、rate(查询正确率),其中查询负载三无 rate(查询正确率)。query\_blade\_areas\_height.txt 对应系统中结果查看的部分如下图 7.8 和图 7.9 所示:

#### 单属性查询结果信息

编号	查询类型	查询条数[条]	查询速度[条/毫秒]	查询用时[毫秒]	查询正确率
1	单一属性单个数值查询 查询属性:文件类型;查询值:Blade	2000	0.7823834	2556.2915	100.00%
2	单一属性离散数值查询 查询属性:地区;查询值:XiBei、XiNan、QingZangGaoYuan	2760	7.1870008	384.02667	100.00%
3	单一属性连续数值查询 查询属性:仿真时间;查询值: 2018-06-21 0:0:02018-06-21 6:0:0	708	1.2112309	584.52936	

图 7.8 单属性查询结果信息

#### 多属性查询结果信息

编 号	直询类型	查询条数[条]	查询速度[条/毫秒]	查询用时[毫秒]	查询正确率
1	多属性单个数值查询 查词属性:地区及文件类型; 查词值:查词地区为"DongBei",文件类型为"Blade"	480	5.986981	80.173965	100.00%
2	多属性离散元数值查询 查词属性:地区及文件类型; 查词值:查词地区为"DongBei"及"HuaBei",文件类型为"Blade"	840	6.3326836	132.64519	100.00%
3	多属性的连续数值查询 查询属性:塔架高度和扇叶长度; 查询值:塔架高度和扇叶长度在合理范围内	1860	6.2205276	299.01	100.00%
4	多属性的多类型数值查询 查词属性:塔架高度、地区和文件类型; 查询值:塔架高度在合理范围内且查询地区为"DongBei"及"HuaDong",文件类型 为"Blade"	540	7.4258647	72.7188	100.00%

图 7.9 多属性查询结果信息

9. 关于版本管理负载结果文件, query\_copynum.txt: 保存版本查询结果数据, 具体包括: count(查询数量)、speed(查询速度)、time(查询用时)、rate(查询正确率),对应系统中结果查看的部分如下图 7.10 所示:

#### 版本查询结果信息

编号	查询类型	查询条数[条]	查询速度[条/毫秒]	查询用时[毫秒]	查询正确率
1	版本管理查询查询 所有文件的版本数量; 查询值:遍历整个版本	null	null	null	null%

#### 图 7.10 版本查询结果信息

- 10. 关于排序结果文件,有如下3个:
  - (1) query\_sort\_towerheight.txt
  - (2) query\_sort\_areaname.txt
  - (3) query\_sort\_simulationtime.txt
- 以上文件保存排序查询结果数据,具体包括: ability(是否可执行)、count

(查询数量)、speed(查询速度)、time(查询用时),对应系统中结果查看的部分如下图 7.11 所示:

排序查询结果信息

编号	查询类型	是否可执行	查询条数[条]	查询速度[条/毫秒]	查询用时[毫秒]
1	数值型排序 所有设备类型为"BLADE"的文件中,按"TOWERHEIGHT"字段倒序查询。	true	2000	4.71203	424.44553
2	字符串型排序 所有设备类型为"BLADE"的文件中,按"AREANAME"字段正序查询。	true	2000	7.6300406	262.1218
3	时间型排序 所有设备类型为"BLADE"的文件中,按"SIMULATIONTIME"字段正序查询。	true	2000	6.5107594	307.18384

图 7.11 排序查询结果信息

- 11. 统计值计算结果文件,有如下3个:
  - (1) query\_average.txt
  - (2) query\_minimum.txt
  - (3) query\_maximum.txt

以上文件保存统计值查询结果数据,具体包括: count(查询数量)、speed(查询速度)、time(查询用时)、rate(查询正确率),对应系统中结果查看的部分如下图 7.12 所示:

#### 最值及平均值查询结果信息

编号	查询类型	查询条数[条]	查询速度[条/毫秒]	查询用时[毫秒]	查询正确率
1	平均值查询 查询每个文件"WINDSPEED"数组中所有数值的平均值。	2000	9.553402	209.34952	100.00%
2	最大值查询 查询每个文件"WINDSPEED"数组中所有数值的最大值。	2000	25.095003	79.69714	100.00%
3	最小值查询 查询每个文件"WINDSPEED"数组中所有数值的最小值。	2000	28.764774	69.52949	100.00%

刷新信息

#### 图 7.12 统计值计算查询结果信息

12. 关于异常文件检测负载结果文件, query\_abnormalfile.txt: 保存异常文件检测结果数据,具体包括: count (待检测文件数量)、speed (查询速度)、time (查询用时)、findcount (找到的异常文件数量)、truecount (实际异常文件数量)、rate (查询正确率),对应系统中结果查看的部分如下图 7.13 所示:

异常文件检测结果信息

查询类型	待检测文件总	查询速度	查询用时	找到的异常文	实际的异常文	查询正
	数[条]	[条/毫秒]	[毫秒]	件[条]	件[条]	确率
异常文件检测 检测生成的风速数据,以正常风速方差的最大值的2倍 为其准检测显堂文件。	null	null	null	null	null	null%

图 7.13 异常文件检测结果信息

13. 关于热点文件查询检测负载结果文件,down\_query\_hotfile.txt: 保存热点文件查询结果数据,包括 n 次相同查询操作的每次查询用时,单位: ms,对应系统中结果查看的部分如下图 7.14 所示:

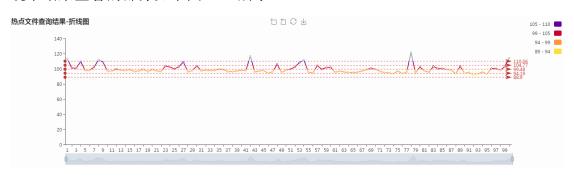


图 7.14 热点文件查询结果信息

#### 8. 附录二: 测试流程案例

#### 8.1 MongoDB 测试流程

#### 8.1.1 MongoDB 环境说明与启动

本部分以 MongoDB 为测试对象,说明如何运行测试系统对 MongoDB 进行评测及结果查看。例子中的 MongoDB 是安装在 Windows 系统下的,在单机上进行运行。

打开 MongoDB, 在其 bin 路径下运行 mongod, 启动 MongoDB 至所存放的数据库存储路径。如下图所示:

```
Microsoft Windows [版本 6.1.7601]
版权所有(c)2009 Microsoft Corporation。保留所有权利。
C:\Users\Daniel>D:
D:\>cd "Program Files\MongoDB\Server\3.4\bin"
D:\Program Files\MongoDB\Server\3.4\bin>mongod --dbpath D:\MongoDB_Data
```

图 8.1 MongoDB 启动命令

打开 MongoDB 图形操作软件 Robomongo,连接数据库,可得到数据库内的相应信息,如下图所示:

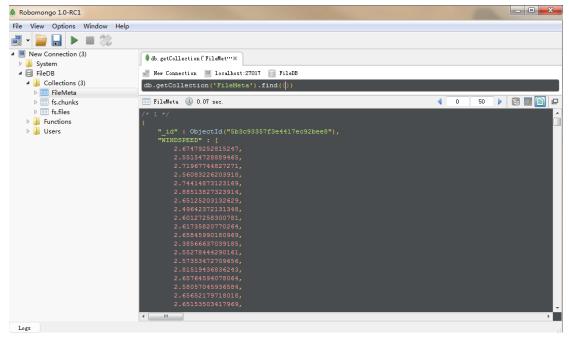


图 8.2 MongoDB 数据库信息

#### 8.1.2 系统设置

首先进行 MBBench-U 的安装,安装步骤参照第三章节。之后进入到 MBBench-U 的主界面,在主界面上进行待测数据库选择,如图所示:



图 8.3 数据库选择区

随后,在负载结果存放路径上输入结果日志存储路径,之后点击"确认信息" 按钮,完成系统设置,如图所示:



图 8.4 系统设置结果反馈

#### 8.1.3 海量文件生成

点击"海量文件生成"按钮,进入海量文件生成界面。其中,在基本参数设置区域内,依次输入文件存储总路径,选择想要生成的元数据文件类型,随后输入参数信息,如:输入模拟的设备数量、每个设备生成的文件数量、每个设备生成的更新文件数量、异常文件占比;之后输入模拟生成设备的基本信息范围,如: 塔架高度区间、扇叶长度区间和扇叶宽度区间,两个数值之间用"-"分割,如下图所示:



图 8.5 海量文件生成--基本参数设置

之后是仿真文件大小的设置,首先输入文件的分配数量,如"3",之后点击 "生成设置控件"按钮。在每一个控件设置区域内,输入最大生成文件大小及该 文件大小下生成的文件数量占比。文件大小可以使用小数,但是占比建议不要使 用小数。设置好后,如下图所示:

## 仿真文件大小设置 #设置文件分配数量 生成设置文件控件 3 生成设置控件 第1个设置 3.8 第2个设置 2.5 第3个设置 0.6 70

图 8.6 海量文件生成-仿真文件大小设置

最后,在仿真时间设置区域,设置开始仿真时间与结束仿真时间,用于确定生成文件的时间范围。如下图所示:



图 8.7 海量文件生成-仿真时间设置

至此,数据集生成设置就进行完成了,在运行数据生成前,可以点击"刷新信息"按钮,获取生成文件总数量及大致上的存储容量,如下图所示:



图 8.8 海量文件生成-生成文件信息

点击"生成文件"按钮,进行文件生成。文件生成进度会实时动态显示,方 便用户及时获取生成状态。如下图所示:

# 实时生成信息 实时进度条 54.49% 存储总文件的总路径为: F:/predatagen 总共二进制文件数量为: 33000个 生成二进制文件数量为: 17983个 当前剩余的文件数量为: 15017个 当前生成文件的速度为: 77个秒 当前生成文件的速度为: 61.90MB/秒 当前已经消耗的时间为: 6分21 秒 当前预计结束的时间为: 3分15 秒

图 8.9 海量文件生成-实时生成文件信息

#### 8.1.4 数据库连接

点击"数据库连接"按钮,进入数据库连接界面。在界面的标题上,可以看到所选待测数据库的名称。在输入参数区域依次输入数据库的连接地址、连接端口号、文件数据存放名称和元数据存放名称,如图所示:



图 8.10 数据库连接操作

输入完成后,点击"连接数据库"按钮,得到信息反馈。注意在点击按钮前确认数据库已经开启。如下图所示:

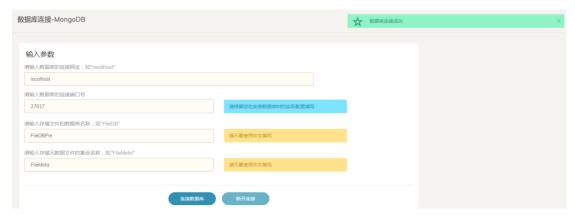


图 8.11 数据库连接信息反馈

#### 8.1.5 数据上传负载测试

点击"数据上传负载测试",进入数据上传负载测试界面。界面上方同样会显示待测数据库的名称。首先需要用户在创建新传输结果文件夹区域下点击"创建文件夹"按钮,用于保存传输负载结果日志文件。随后在确认信息区域下确认上传文件路径、上传文件存储名称和上传元数据存储名称,用户也可点击"修改信息"按钮,对信息进行修改。如下图所示:



图 8.12 数据上传负载测试信息

当信息确认无误后,点击"数据上传",即可运行二进制文件上传负载测试。 上传情况会通过信息记录和直线图的方式实时显示出来,如下图所示:



图 8.13 二进制文件上传负载实时信息

二进制文件上传完成后,点击"元数据上传"按钮,运行元数据上传负载测试,如下图所示:

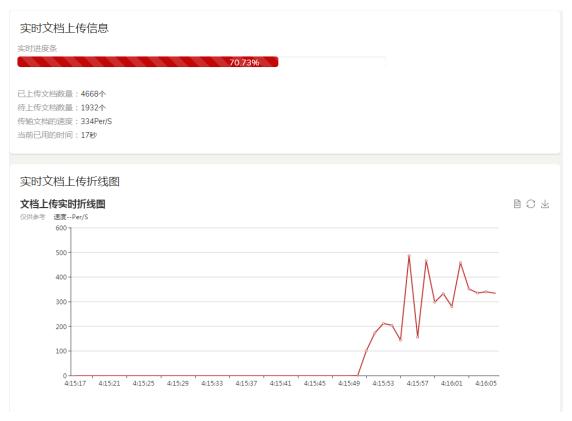


图 8.14 元数据上传负载实时信息

#### 8.1.6 数据下载负载测试

点击"数据下载负载测试",进入数据下载负载测试界面。由于之前已经进行结果日志存储路径设置,因此这里不需要再次设置。首先用户可在下载文件信息区域查看当前数据库内的文件数量和文件总量。如下图所示:



图 8.15 数据下载信息

然后在数据下载测试区域内输入文件下载路径,之后点击"数据下载",即可运行二进制文件下载负载测试,如下图所示:



图 8.16 二进制文件下载负载实时信息

二进制文件下载完成后,点击"元数据下载"按钮,即可运行元数据下载负载测试,如下图所示:

### 字的文档下载信息 実時进度条 75.89% PT 歌文件粉量:5009へ 传育歌文件粉量:1591へ 传輸文件的速度:317Per/S 当前已用的时间:17秒 字 时文档下载が线圏 文性下載文明が线圏 QUIS参考 速度-Per/S 500 400 400 425:20 4:25:24 4:25:28 4:25:32 4:25:36 4:25:40 4:25:44 4:25:48 4:25:52 4:25:56 4:26:00 4:26:04 4:26:08 444

#### 图 8.17 元数据下载负载实时信息

#### 8.1.7 简单查询负载测试

点击"简单查询负载测试"按钮,进入简单查询负载测试界面。首先需要在创建新负载结果文件下点击"创建文件夹",用于保存查询负载结果日志文件。然后在其下方的设置并确认信息区域,设置查询文件下载路径并核查文件存储名称和元数据存储名称是否正确。如下图所示:



图 8.18 查询负载信息确认

之后,在确认数据生成记录区域内选择相对应的数据生成信息,用于随机生成查询条件和查询数值。数据生成记录可在页面的数据生成记录区域内进行查看。如下图所示:



图 8.19 数据生成记录确认

随后,在单属性查询测试区域内,点击"开始查询"按钮,即可运行单属性查询负载。如下图所示:



图 8.20 单属性查询测试执行

当查询下方出现"Single attribute query has been finished"字样时,表示查询 负载执行完毕,可运行之后的负载测试。

多属性查询测试和单属性查询测试流程类似,在此不做过多叙述。

在数据文件版本管理测试上,需要输入生成文件存储总路径,可以从数据生成记录里的"Data Storage Path"中获取。随后点击"开始查询",即可运行。如下图所示:



图 8.21 数据文件版本管理测试执行

#### 8.1.8 复杂查询负载测试

复杂查询负载测试包含排序负载测试、最值及均值负载测试、异常文件检测 负载测试和热点文件查询负载测试。由于在简单查询负载测试界面上已经创建的 结果日志保存文件并选择了数据生成文件,因此在本界面中无需再次设置。

排序负载测试和最值及均值负载测试与前面的简单查询负载测试相同,在此不再叙述。

异常文件检测负载测试与前面的数据文件版本管理测试相同,在此不再叙述。 对于热点文件查询负载测试,需要设定查询次数和每次查询的间隔时间。需 要注意,间隔时间是以毫秒为单位的,并且输入的只能是正整数。如下图所示:



图 8.22 热点文件查询负载测试执行

#### 8.1.9 结果查看

点击"结果查看"按钮,进入到结果查看界面。首先会显示当前系统信息,包含 JDK 信息、基本信息、内存信息等。之后,点开界面右上方的"负载结果",选择相应的负载查看内容,进入具体结果查看界面,如下图所示:

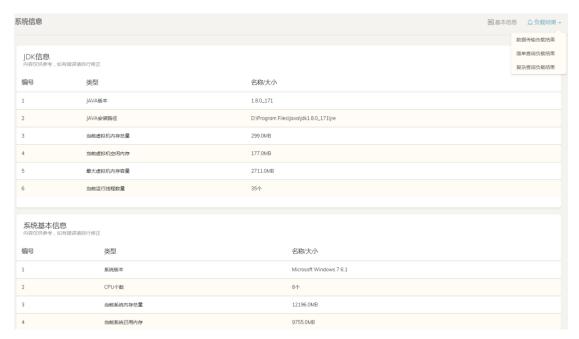


图 8.23 系统信息显示

首先是传输结果信息,用户可通过选择传输测试版本来查看相应的传输负载 测试结果。具体内容如下图所示:



图 8.24 传输结果基本信息

在内容下方,具有各个传输记录的折线图显示,用户可以选择不同的时间间隔单位,如每五秒为一平均的结果显示或者是每一秒的速度结果显示。显示示例如下图所示:



图 8.25 文件上传折线图

折线图下方的横向伸缩线可放大或者缩小显示内容,并可动态调控显示位置。 其次是简单查询负载信息,用户可通过选择查询测试版本来查看相应的传输 负载测试结果。具体内容如下图所示:



图 8.26 简单查询基本信息显示

其中,对于进行数据下载的负载,可通过点击"数据下载结果查看"按钮来显示其数据下载折线图,如下图所示:



图 8.27 下载结果折线图显示

最后是复杂查询负载结果界面,其内容与格式同简单查询负载结果界面相同,在此不再叙述。

以上是使用 MBBench-U 对 MongoDB 进行负载测试的流程,如有问题可参考界面的帮助文档或者与我们联系。

#### 8.2 Elasticsearch 测试流程

#### 8.2.1 Elasticsearch 环境说明与启动

本系统是安装在 Windows 环境下的,在本地运行,因此 ElasticSearch 也在 Windows 环境下安装。首先需要安装 jdk,ElasticSearch 是基于 lucence 开发的,在运行时需要 java jdk 的支持,本系统中测试的是 ElasticSearch6.2.3 版本,需要依赖于 jdk1.8 或更高版本。jdk 的安装在此不做说明。之后下载 ElasticSearch6.2.3,并将文件解压到指定目录,假设解压到 D 盘根目录下。打开命令行,进入到 D:\elasticsearch-6.2.3\bin 目录下执行 elasticsearch.bat,如下图所示:



图 8.28 ElasticSearch 启动

如此就启动了一个 ElasticSearch 节点,启动后会显示如下图所示的信息提示:

图 8.29 ElasticSearch 启动成功

在浏览器中打开 http://localhost:9200/, 若显示如下信息, 表示 ElasticSearch 启动成功:

```
{
    "name" : "kwNyY6Z",
    "cluster_name" : "elasticsearch",
    "cluster_uuid" : "b7u5BluuQb-pu_40Juv3yg",
    "version" : {
        "number" : "6.2.3",
        "build_hash" : "c59ff00",
        "build_date" : "2018-03-13T10:06:29.741383Z",
        "build_snapshot" : false,
        "lucene_version" : "7.2.1",
        "minimum_wire_compatibility_version" : "5.6.0",
        "minimum_index_compatibility_version" : "5.0.0"
},
        "tagline" : "You Know, for Search"
}
```

图 8.30 ElasticSearch 启动成功显示信息

此时默认的集群名为 elasticsearch, 节点名为 node-estest。集群名、节点名等参数可以在 D:/elasticsearch-6.2.3/config/elasticsearch.yml 中进行修改。

安装 ElasticSearch-head 插件可用来浏览、监控 ElasticSearch 的状态,与 ElasticSearch 进行交互。ElasticSearch-head 的安装在此不做说明。ElasticSearch 通过 ElasticSearch-head 的前端呈现如下图所示:

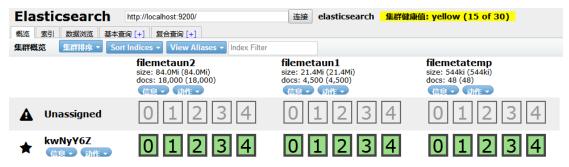


图 8.31 ElasticSearch 前端呈现

#### 8.2.2 系统设置

打开 MBBench-U 系统,在"系统设置"界面点击所要评测的数据库,下拉菜单选择"ElasticSearch",界面下方随之显示 ElasticSearch 的简介,之后在"负载结果存放路径"一栏填写负载结果的存放总路径,点击"确认信息"的按钮,系统右上方弹出"设置成功"的提示,表示测试 ElasticSearch 的系统设置成功。操作示例如下图所示:

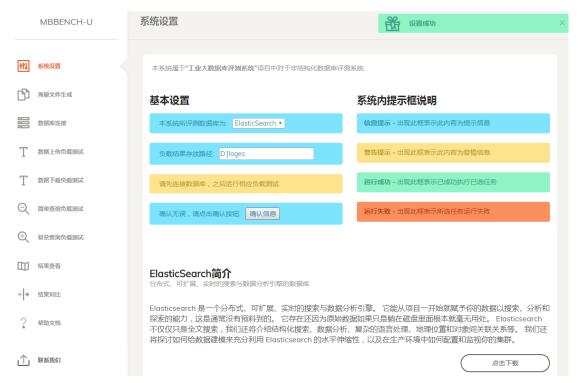


图 8.32 ElasticSearch 系统设置

#### 8.2.3 海量文件生成

海量文件生成用于生成测试 ElasticSearch 的数据。此部分的操作参照 MongoDB 测试流程中"海量文件生成"部分的说明,在此不再叙述。

#### 8.2.4 数据库连接

在"数据库连接"界面进行 ElasticSearch 的连接,分别输入连接网址(这里为本地 IP: localhost),端口号(9300)以及分别用于存储二进制文件和元数据文件的索引(类比关系型数据库中数据库的概念): filedb 和 filemeta。注意

elasticsearch 中的索引名称需要小写。之后点击"连接数据库"按钮,待界面右上方出现"数据库连接成功"的提示表示 ElasticSearch 连接成功。操作示例如下图所示:

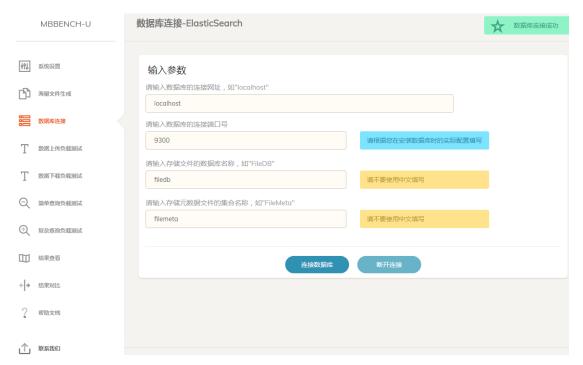


图 8.33 ElasticSearch 数据库连接

#### 8.2.5 数据上传负载测试

数据上传负载测试的操作流程与 MongoDB 相同,此部分参照 MongoDB 测试流程中"数据上传负载测试"部分的说明,在此不再叙述。

#### 8.2.6 数据下载负载测试

数据下载负载测试的操作流程与 MongoDB 相同,此部分参照 MongoDB 测试流程中"数据下载负载测试"部分的说明,在此不再叙述。

#### 8.2.7 简单查询负载测试

简单查询负载测试的操作流程与 MongoDB 相同,此部分参照 MongoDB 测试流程中"简单查询负载测试"部分的说明,在此不再叙述。

#### 8.2.8 复杂查询负载测试

复杂查询负载测试的操作流程与 MongoDB 相同,此部分参照 MongoDB 测试流程中"复杂查询负载测试"部分的说明,在此不再叙述。

#### 8.2.9 结果查看

点击"结果查看"按钮,进入到结果查看界面。首先显示的是当前系统信息, 包含 JDK 信息、基本信息、内存信息等。系统信息界面如图 8.34 所示。

统信息							M 基本信息	△负载结果
JDK信 内容仅供	<b>記</b> 供参考,如有错误请自	行修正						
编号	类型	!			名称/大小			
1	JAVA	版本			1.8.0_161			
2	JAVA	安装路径			D:\Java\jdk1.8.0_1	61\jre		
3	当前。	虚拟机内存总量			1424.0MB			
4	当前原	<b>显拟机空闲内存</b>			800.0MB			
5	最大点	虚拟机内存容量			3627.0MB			
5	当前江	运行线程数量			84个			
	基本信息 <sup>供参考,如有错误请自</sup>	行修正						
偏号	类型	1		名和	<b>尔大小</b>			
1	系统	版本		Micr	osoft Windows 7 6	.1		
2	CPU-	个数		8个				
3	当前	系统内存总量		163	14.0MB			
4	当前	系統已用内存		150	33.0MB			
5	当前	系统空闲内存		128	0.0MB			
CPU <b>(</b> 内容仅使	<b>言息</b> 供参考,如有错误请自	行修正						
扁号	CPU <b>总量</b> MH:	z CPU生产	一商	CPU类别			CPU缓存数	全
L	3408	Intel		Core(TM) i7-6	700 CPU @ 3.40GH	lz	-1	
2	3408	Intel		Core(TM) i7-6	700 CPU @ 3.40GH	łz	-1	
3	3408	Intel		Core(TM) i7-6	700 CPU @ 3.40GH	lz	-1	
1	3408	Intel		Core(TM) i7-6	700 CPU @ 3.40GH	lz	-1	
5	3408	Intel		Core(TM) i7-6	700 CPU @ 3.40GH	lz	-1	
5	3408	Intel		Core(TM) i7-6	700 CPU @ 3.40GH	lz	-1	
7	3408	Intel		Core(TM) i7-6	700 CPU @ 3.40GH	lz	-1	
3	3408	Intel		Core(TM) i7-6	700 CPU @ 3.40GH	lz	-1	
磁盘信	<b>言息</b> 共参考,如有错误请自	行修正						
扁号	磁盘符号	总共容量(GB)	剩余容量	k(GB)	已用占比	读速度(MB/s)	写速度(1	MB/s)
L	C 盘	97.00	23.00		76.0	676.16113	657.5342	
2	D盘	809.00	668.00		18.0	211.64453	547.7314	5

图 8.34 系统信息显示

界面右上方的"负载结果"中有数据传输负载结果、简单查询负载结果以及

复杂查询负载结果。点击"数据传输负载结果"可以查看数据库的基本存储信息 以及数据上传、下载的结果,用户可通过选择传输测试版本来查看特定的传输负 载结果,如下图所示:

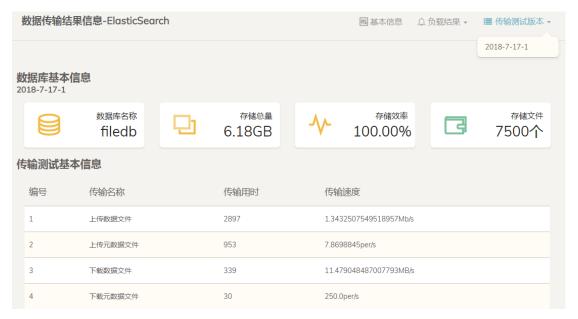


图 8.35 传输结果基本信息

二进制文件和元数据文档的上传下载结果分别以折线图的形式展示, 折线图中可以选择不同的时间间隔单位, 如每五秒为一平均的结果显示或者是每一秒的速度结果显示。以文件上传结果为例, 文件上传结果的折线图如下图所示:



图 8.36 文件上传折线图

折线图下方的横向伸缩线可放大或者缩小显示内容,并可动态调控显示位置。 查询(简单查询、复杂查询)负载结果显示中,包括查询条数、查询速度、 查询用时等结果信息组成的表格,图 8.37 是以简单查询为例的结果表格。在查 询负载结果显示中,可以选择不同的查询测试版本来查看某一特定的查询负载结果信息,点击某一测试版本,之后在特定查询结果信息栏的右下角点击"刷新信息"即可。

简单查试	<b>向负载结果</b>		加 基本信息	△负载结果▼	置查询测试版本、
单属	性查询结果信息				
编号	查询类型	查询条数[条]	查询速度[条/毫秒]	查询用时[毫秒]	查询正确率
1	单一属性单个数值查询 查询属性: DEVICETYPE:BLADE	2500	4.10496	609.01935	100.00%
2	单一属性离散数值查询 查询属性: AREANAME:[XiBei, QingZangGaoYuan, DongBei]	3750	3.758501	997.73816	100.00%
3	单一属性连续数值查询 查询属性: 2018-07-10 04:02:532018-07-10 16:02:53	3756	5.80269	647.286	100.00%
隐藏	数据下载结果				刷新信息

图 8.37 简单查询基本信息显示

对于进行数据下载的负载,可通过点击"数据下载结果查看"按钮来显示其数据下载折线图,如下图所示:



图 8.38 下载结果折线图显示

复杂查询负载结果的展示形式、内容与简单查询大致相同,复杂查询结果显示如下图所示:



图 8.39 复杂查询结果显示