编程作业 3: 分布式 DBLP 数据查询系统

一、项目结构

```
hw3
├── Client
│   ├── Client.iml
│   └── src
│       ├── AccessServer.java
│       ├── ClientMain.java
```

```
│       └── Initialize.java
├── Logger
│   └── 2153393-hw2-q1.log
├── Server
│   ├── Server.iml
│   └── src
│       ├── Initialize.java
│       ├── Query.java
│       ├── ServerMain.java
│       ├── VirtualServer.java
│       └── localindex
│           ├── DataSet.java
│           ├── IndexInitializer.java
│           ├── IndexQuery.java
│           └── LocalIndex.java
├── hw3.iml
└── mnt
    ├── dblp.xml
    ├── dblp0.xml
    ├── dblp1.xml
    ├── dblp10.xml
    ├── dblp11.xml
    ├── dblp12.xml
    ├── dblp13.xml
    ├── dblp14.xml
    ├── dblp15.xml
    ├── dblp16.xml
    ├── dblp17.xml
    ├── dblp18.xml
    ├── dblp19.xml
    ├── dblp2.xml
    ├── dblp20.xml
    ├── dblp21.xml
    ├── dblp22.xml
    ├── dblp23.xml
    ├── dblp3.xml
    ├── dblp4.xml
    ├── dblp5.xml
    ├── dblp6.xml
    ├── dblp7.xml
    ├── dblp8.xml
    └── dblp9.xml
```

1. Client

- `AccessServer.java`：负责客户端与服务器的交互，主要用于发送查询请求到服务器并接收查询结果。它封装了与服务器的通信逻辑，包括传输作者姓名、年份范围等参数，并返回查询的论文频次。
- `ClientMain.java`：客户端的主程序，负责处理用户交互和操作逻辑，包括初始化 DBLP 数据分布式存储和发送查询请求。支持查询论文频次的功能，具备故障容错能力（查询备份数据）。

- `Initialize.java`：负责初始化客户端环境，包括切分 DBLP.xml 文件为多个小文件，以及将切分后的文件通过网络传输到多个服务器节点，实现分布式数据存储的初始化工作。

---

2. Server

- `Initialize.java`：负责服务器端的文件接收服务，监听指定端口接收客户端上传的 DBLP 数据文件，并将其分类存储为正式文件或备份文件。
- `Query.java`：提供基于文件的查询功能，支持按作者姓名或作者姓名和年份范围查询论文发表频次。通过解析 XML 文件块提取目标信息，统计符合条件的论文数量。
- `ServerMain.java`：服务器主程序，负责启动文件接收服务和查询服务，协调服务器端各模块运行。支持用户选择虚拟机端口并启动相关服务。
- `VirtualServer.java`：负责处理客户端查询请求，监听查询端口并接收查询参数（如作者、年份范围）。支持基于文件的直接查询和索引查询，将结果返回给客户端。

---

3. Server/localindex

- `DataNode.java`：数据存储单元类，用于存储作者姓名、发表年份和论文频次。支持数据的唯一性检查（基于作者和年份），是索引功能的基础单元。
- `DataSet.java`：数据集合类，使用 `HashSet` 管理多个 `DataNode`，支持添加数据、按条件查询论文频次等操作，是本地索引的核心数据结构。
- `IndexInitializer.java`：用于生成本地索引文件，为每个 DBLP XML 文件创建对应的索引目录。通过解析文件内容提取信息，构建高效的索引结构。
- `IndexQuery.java`：基于本地索引文件的查询模块，支持按作者或作者和年份范围查询论文频次。通过加载索引文件快速定位和统计查询结果。
- `LocalIndex.java`：提供索引的生成和查询功能。负责解析 XML 文件、生成索引文件并序列化存储，同时支持通过索引文件查询论文频次，是本地索引的核心实现类。

二、项目环境

- JavaSDK：1.8
- IDE：IntelliJ IDEA
- 一台本地主机(`Mac`)，三台腾讯云服务器(`CentOS 7.6`)
- 服务器之间通过公网IP进行通信。每个服务器上通过运行不同的线程，使用不同的端口来创建出多个虚拟机，在本项目中每一台服务器上运行两台虚拟机
- 本地主机起到客户端作用，负责向云服务器发送查询请求到服务器并接收查询结果

三、环境部署

Server端

- 首先将`Server`文件夹中的全部内容上传至三台云服务器，每台云服务器均配有JavaSDK 1.8，可以将项目正确运行

- 利用进入`Server/src`文件夹下利用`javac -d ../../bin/server localindex/*.java Initialize.java Query.java ServerMain.java VirtualServer.java`命令可以编译java文件，并在根目录下创建`bin`文件夹来存放编译好的class文件

Client端

- 直接打开IntelliJ，然后进入`Client/src`文件夹下运行`ClientMain`即可

四、需求处理

功能需求

> 2.1 给出输入条件：作者名字 author=[Ion Stoica]，返回作者的 DBLP 发表论文总数。
>
> 2.2 给定上述作者信息和年份区间（如: year>2010, 或者 2010< year <2022, 或者 year=2010）返回该作者在对应年份的 DBLP 发表论文总数

- 该部分详情见项目结果部分，均已实现

非功能需求

1. 负载均衡

为了实现存储负载均衡，本项目采用以下方法将原始的 `dblp.xml` 文件切分为 24 个片段 XML 文件，并分配到 6 台虚拟机上，每台虚拟机负责存储 4 个片段：

1. 在本地主机上执行切分程序，该程序会创建 24 个文件流，用于分别写入 24 个片段文件。由于我们要写入的是 XML 文件，为了保持格式正确，选用了 `XMLStreamWriter` 作为文件的写入流。
2. 切分过程中借助 Java 中的 StAX 库进行解析（StAX 的优势是不需要将整个 XML 文件加载到内存，而是逐步解析），逐步读取 `dblp.xml` 文件内容。
3. 每当读取到一个完整的信息块（例如 `<article>...</article>` 或 `<book>...</book>`），会随机选择一个文件流作为输出目标，将该信息块写入对应的文件中。
4. 这一过程持续进行，直到整个 `dblp.xml` 文件被完全读取和分割。

通过多次随机分配，每个文件流中的数据量均匀分布，最终生成的 24 个片段 XML 文件大小基本相等（约为 163MB），从而实现了存储负载的均衡分配。

---

2. 查询容错

在查询客户端始终正常工作的假设下，本项目采用了容错机制，允许至多一半的存储虚拟机（即 N/2 台）发生故障时仍能正确返回查询结果。具体机制如下：

1. **主备存储**
   每台存储虚拟机设置两个存储目录：一个用于存放主 XML 文件，另一个用于存放备份文件。在文件分发过程中，每个 XML 文件片段被发送两次，一次作为主文件发送给一台虚拟机，另一次作为备份文件发送给另一台虚拟机。最终，每个 XML 文件片段的主文件和备份文件会分布在不同的虚拟机上。

2. **查询容错**
   如果查询时某台虚拟机发生故障无法返回结果，查询客户端会前往存储该虚拟机备份文件的另一台虚拟机进行查询，确保结果的完整性。通过主备存储策略，即使有 N/2 台虚拟机发生故障，仍能查询到所有数据。

---

3. 通信模式

本项目的通信模式分为以下两种场景：

1. **数据分发通信**
   在切分 `dblp.xml` 文件后，虚拟机需要将切分后的 24 个片段 XML 文件分发到 6 台负责存储的虚拟机上。此时，切分任务的虚拟机作为客户端，执行分发程序，将生成的 XML 文件片段发送给各个存储虚拟机。存储虚拟机作为服务端，监听文件发送请求并接收和存储这些 XML 片段文件。每个存储虚拟机都会存储 4 个片段文件，实现负载均衡的存储分布。
2. **查询通信**
   查询时，查询客户端需要与 6 个存储虚拟机进行通信。查询客户端会向每个存储虚拟机发送查询请求，传递需要查询的 `author` 和 `year` 信息。每个存储虚拟机会根据收到的查询参数进行本地查询，统计相关论文的频次后，将结果返回给客户端。客户端收到来自 6 台虚拟机的结果后进行汇总，最终得出查询结果。为了提升性能，客户端采用多线程机制，针对每台存储虚拟机分别开启一个线程，从而实现 6 台虚拟机的并行查询，显著缩短了查询时间。

---

4. 查询机制

每台存储虚拟机负责处理其存储的 XML 文件片段，并对这些文件中的数据进行统计，计算论文频次的总和，返回给查询客户端。查询分为两种类型：

1. **按作者查询**
   - 方法一：使用 `grep` 和 `wc` 命令直接搜索每个 XML 文件中 `author` 字段出现的次数，以此统计结果。
   - 方法二：解析 XML 文件，针对每个完整的信息块（如 `<article>...</article>`），判断其中是否包含目标作者。如果包含，则计数器递增。
2. **按作者和年份查询**
   - 解析 XML 文件中的每个信息块，检查块中是否存在符合年份范围的记录，并判断是否包含目标作者。如果同时满足条件，则计数器递增。

解析 XML 文件使用的是 Java 的 StAX 库，可以逐步解析 XML 文件，避免将整个文件加载到内存中，从而高效处理大规模数据。

---

5. 本地索引机制

为了提高查询性能，本项目引入了本地索引机制，其实现如下：

1. **索引数据结构**
   索引基于哈希表实现，使用 Java 的数组和集合（`Set`）组合的结构。整个哈希表由大小为 101 的数组组成，数组中的每个元素是一个集合，用于存储索引数据。以 `author` 和 `year` 作为键值对，通过哈希函数计算键的哈希值，确定其在数组中的位置。数据存储时，如果集合中已存在相同的记录，则仅增加频次值；否则新增记录。

2. **索引生成**
    - 针对每个 XML 文件，逐块解析其内容，提取 `author` 和 `year` 信息。
    - 使用哈希函数计算 `author` 和 `year` 的哈希值，确定数据存储的集合。
    - 在添加数据时检查是否存在重复记录，若有重复则增加频次，避免占用过多内存。
    - 将构建好的哈希表写入磁盘，为每个 XML 文件新建一个索引目录，包含 101 个 `.ser` 文件，每个文件对应一个集合的序列化数据。
3. **查询流程**
    - 在查询时，根据输入的 `author` 和 `year` 信息计算哈希值。为了优化性能，哈希函数仅针对 `author` 进行计算，而 `year` 则通过条件判断过滤。
    - 根据计算得到的哈希值，定位对应的 `.ser` 文件并反序列化加载至内存。
    - 遍历集合，查找满足条件的数据记录并统计频次。

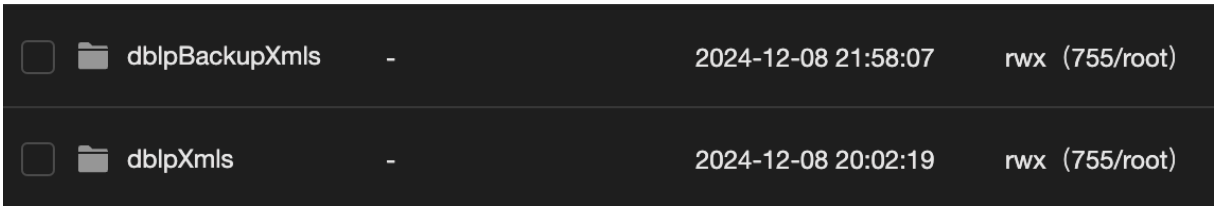
通过本地索引机制，每台存储虚拟机可以快速查询其存储的数据片段，大幅提高查询效率，同时降低内存占用。

五、项目运行

1. 对`dblp.xml`切片并发送至存储虚拟机

Client端

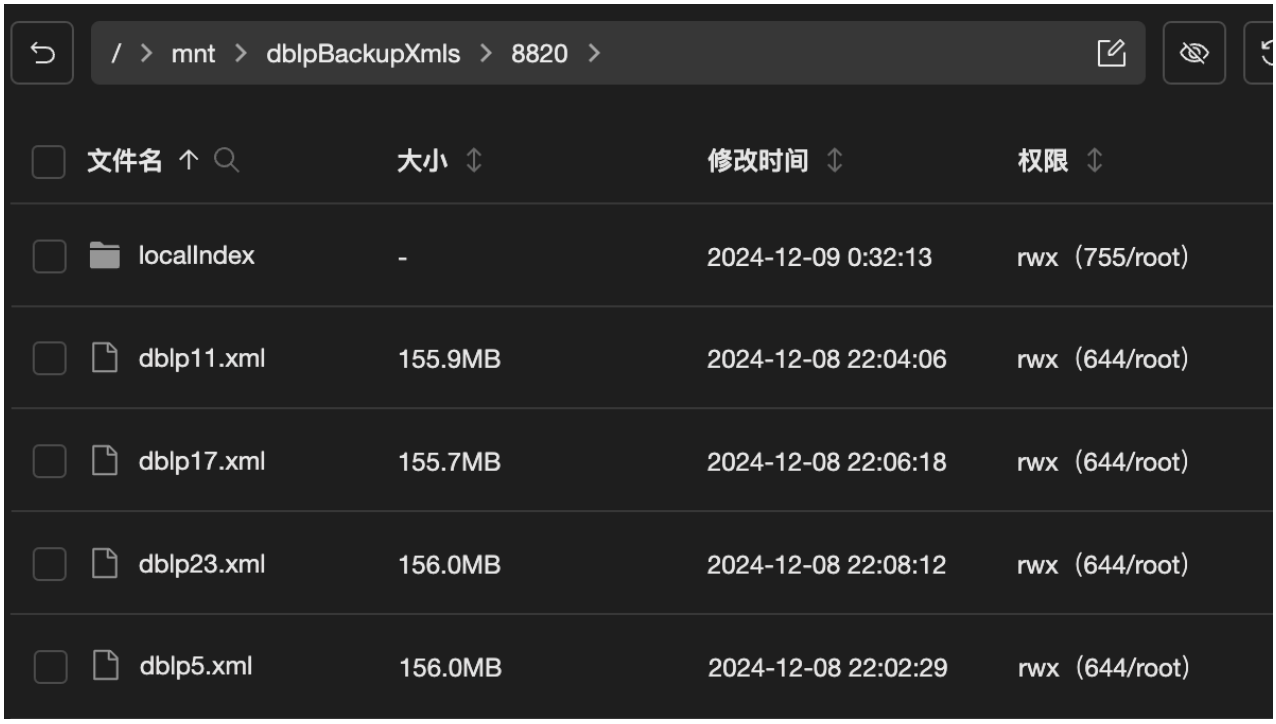
- 首先需要对`dblp.xml`文件切片，运行`ClientMain`后，会选择是否初始化DBLP的分布式存储，如果是首次运行项目，输入yes，即可自动将`dblp.xml`文件切片并开启六个线程发送至各个存储虚拟机，虚拟机中在`mnt`文件夹下会出现两个文件夹，如下图所示，`dblpXmls`文件夹存放的为主块文件，`dblpBackupXmls`文件夹存放的为副本文件



2. 对切片建立索引

Server端

- 随后进入每一个服务器中，进入到`bin`文件夹下，利用`java localindex.IndexInitializer`对切片后的文件建立索引，在索引建立结束以后，会出现如下的文件夹`localindex`，主块和副本均有



- 然后在每一台服务器上利用`tmux`开启两个会话，每一个会话模拟一台虚拟机，进入`bin`文件夹下利用`java ServerMain`命令运行虚拟机，选择端口即可

3. 输入检索作者与年份进行检索

Client端

- 运行`ClientMain`后，会选择是否初始化DBLP的分布式存储，如果是首次运行项目，输入yes；如果先前已经建立索引输入no。

六、项目结果

(一) 正常情况

> 这里要解释一个bug，就是在统计作者和论文的时候会出现将作者的HomePage当作一个论文统计，其格式统计出来以后year会是null，这个在参考论文提过，这就会导致我们实际统计出的论文数目会比实际多1，原因就是将HomePage也当成发表论文了，因此这是一个小bug，后续如果需要修正可以调整`dblp.xml`切分方法
> 论文解释如下：
>
> **Person Records**
>
> Many DBLP authors maintain their own personal "home pages" on the web. It soon became obvious to add links from the DBLP author pages to personal home pages. Home pages were modeled as special web publications:
>
> ```
> <www key="homepages/m/DavidMaier" ...>
>   <author>David Maier</author>
>   <title>Home Page</title>
>   <url>http://web.cecs.pdx.edu/~maier/</url>
> </www>
> ```
>
> The key always starts with `homepages/`, the `author` field specifies the name of the person, the `title` field always has the value "Home Page", and the `url` field contains the location of the home page. Later it became clear that this
>
> 参考论文地址：https://dblp.uni-trier.de/xml/docu/dblpxml.pdf

1. 无年份要求查询

```
/Library/Java/JavaVirtualMachines/jdk-23.jdk/Contents/Home/bin
==========================================================
选择是/否要初始化DBLP文件的分布式存储(输入yes/no): no
==========================================================
请输入作者姓名 (输入exit退出):Zhen Gao
==========================================================
请输入起始年份(输入'*'代表不限制): *
请输入截止年份(输入'*'代表不限制): *
==========================================================
正在查询 (未使用索引) .....
查询成功,  用时: 8.89s
在没有年份限制情况下, 查询到Zhen Gao发表的DBLP论文总数为135
==========================================================
正在查询 (使用索引) .....
查询成功 (使用本地索引) ! 用时: 0.44s
在没有年份限制情况下, 查询到Zhen Gao发表的DBLP论文总数为135
==========================================================
```

2. 无年份下限查询

```
==========================================================
请输入作者姓名 (输入exit退出):Zhen Gao
==========================================================
请输入起始年份(输入'*'代表不限制): *
请输入截止年份(输入'*'代表不限制): 2021
==========================================================
正在查询 (未使用索引) .....
查询成功,  用时: 8.86s
从*年到2021, Zhen Gao发表DBLP论文总数为74
==========================================================
正在查询 (使用索引) .....
查询成功 (使用本地索引) ! 用时: 0.51s
从*年到2021, Zhen Gao发表DBLP论文总数为74
==========================================================
```

3. 无年份上限查询

```
==================================================
请输入作者姓名（输入exit退出):Zhen Gao
==================================================
请输入起始年份(输入'*'代表不限制)： 2022
请输入截止年份(输入'*'代表不限制)： *
==================================================
正在查询（未使用索引）.....
查询成功， 用时：8.80s
从2022年到*，Zhen Gao发表DBLP论文总数为62
==================================================
正在查询（使用索引）.....
查询成功（使用本地索引）！ 用时：0.45s
从2022年到*，Zhen Gao发表DBLP论文总数为62
==================================================
```

4. 指定年份区间查询

- 这里为了验证查询正确性，我利用python脚本将`dblp.xml`转换为`lson`格式并统计每一个作者在每一年的数据，形成如下格式：


```
{"Daniela Florescu":
{"1998":12,"2005":3,"1999":10,"1996":7,"2008":3,"2009":4,"2010":2,"1997":4,"2001":7,"
1995":3,"2000":14,"2003":3,"2004":4,"2013":1,"2002":2,"2006":4,"2019":1,"2007":1,"199
4":1}}
{"Alon Y. Levy":
{"1998":16,"1999":18,"2000":11,"1997":11,"2021":1,"1995":5,"1994":5,"1996":12,"1993":
4,"1992":2,"2001":1}}
{"Dan Suciu":
{"1998":17,"2018":15,"2009":23,"1999":14,"2017":21,"2008":14,"2007":15,"2003":12,"199
7":10,"2011":15,"2024":13,"2023":13,"2022":16,"2001":15,"2021":7,"1994":5,"2000":8,"2
014":20,"2019":12,"2002":10,"2004":12,"2020":24,"2016":18,"2012":19,"2010":14,"2015":
10,"2013":13,"2006":10,"2005":13,"1996":4,"1995":4,"1993":1}}
```


- 此处我们查询的作者是：`Alon Y. Levy`，年份是`1998-1999`，结果应该是`16+18+1=35`（多1的原因本部分开始解释过）

dblp_line.lson


```
{"Arnon Rosenthal":
{"1998":6,"2011":4,"1985":2,"1992":3,"2016":2,"1996":4,"1975":1,"1977":1,"1981":3,"1982":3,"2004":7,"1999":3,"1995":2,"1990":3,"1997":7,"1983":1,"1988":2,"1980":1,"2008":3,"2009":4,"2010":1,"1989":4,"2014":2,"1974":1,"2007":1,"2001":7,"2002":1,"1984":4,"1986":3,"1994":4,"1991":1,"1987":4,"2006":4,"2000":4,"1993":2,"2018":1,"2005":2,"2003":2,"2013":1}}
{"Robin Cover":{"2006":1,"2001":1}}
{"Alin Deutsch":
{"1998":3,"2018":5,"2009":7,"2008":8,"2007":5,"2011":8,"2014":4,"2022":2,"2006":6,"2013":5,"2005":8,"2017":3,"2020":2,"2010":4,"2016":5,"2012":3,"2021":5,"2023":2,"2019":4,"1999":4,"2015":2,"2004":3,"2003":3,"1995":1,"2000":1,"2001":2,"1996":1,"2024":1}}
{"Mary F. Fernandez":
{"1998":5,"2002":2,"1997":5,"2000":2,"1992":1,"1993":1,"2001":2,"1995":2,"1999":1,"1989":1,"2012":1}}
{"Daniela Florescu":
{"1998":12,"2005":3,"1999":10,"1996":7,"2008":3,"2009":4,"2010":2,"1997":4,"2001":7,"1995":3,"2000":14,"2003":3,"2004":4,"2013":1,"2002":2,"2006":4,"2019":1,"2007":1,"1994":1}}
{"Alon Y. Levy":
{"1998":16,"1999":18,"2000":11,"1997":11,"2021":1,"1995":5,"1994":5,"1996":12,"1993":4,"1992":2,"2001":1}}
{"Dan Suciu":
{"1998":17,"2018":15,"2009":23,"1999":14,"2017":21,"2008":14,"2007":15,"2003":12,"1997":10,"2011":15,"2024":13,"2023":13,"2022":16,"2001":15,"2021":7,"1994":5,"2000":8,"2014":20,"2019":12,"2002":10,"2004":12,"2020":24,"2016":18,"2012":19,"2010":14,"2015":10,"2013":13,"2006":10,"2005":13,"1996":4,"1995":4,"1993":1}}
{"Roy T. Fielding":
{"1999":4,"2002":3,"1998":2,"1994":2,"2005":2,"2014":6,"2022":3,"1996":1,"2012":2,"1995":2,"1997":3,"2017":1,"2000":2}}
{"Henrik Frystyk Nielsen":
{"1999":2,"2023":1,"1994":1,"1995":1,"2000":1,"1996":1,"1997":3,"2007":1,"2008":2}}
{"Tim Berners-Lee":
```


- 查询结果

```
==========================================================
请输入作者姓名（输入exit退出):Alon Y. Levy
==========================================================
请输入起始年份(输入'*'代表不限制)： 1998
请输入截止年份(输入'*'代表不限制)： 1999
==========================================================
正在查询（未使用索引）.....
查询成功，  用时：8.84s
从1998年到1999，Alon Y. Levy发表DBLP论文总数为35
==========================================================
正在查询（使用索引）.....
查询成功（使用本地索引）！用时：0.48s
从1998年到1999，Alon Y. Levy发表DBLP论文总数为35
==========================================================
```

(二) 异常测试

1. 任意一台虚拟机故障

```
==================================================
选择是/否要初始化DBLP文件的分布式存储(输入yes/no): no
==================================================
请输入作者姓名 (输入exit退出):Zhen Gao
==================================================
请输入起始年份(输入'*'代表不限制): *
请输入截止年份(输入'*'代表不限制): *
==================================================
正在查询 (未使用索引) .....
==================================================
====================虚拟机2故障====================
====================正在查询备份====================
==================================================
查询成功,  用时: 14.18s
在没有年份限制情况下, 查询到Zhen Gao发表的DBLP论文总数为135
==================================================
正在查询 (使用索引) .....
==================================================
====================虚拟机2故障====================
====================正在查询备份====================
==================================================
查询成功 (使用本地索引) ! 用时: 0.66s
在没有年份限制情况下, 查询到Zhen Gao发表的DBLP论文总数为135
==================================================
```

2. 任意n台虚拟机故障

```
请输入作者姓名（输入exit退出):Zhen Gao
=====================================================
请输入起始年份(输入'*'代表不限制): *
请输入截止年份(输入'*'代表不限制): *
=====================================================
正在查询（未使用索引）.....
=====================================================
====================虚拟机2故障====================
==================正在查询备份====================
=====================================================
=====================================================
====================虚拟机4故障====================
==================正在查询备份====================
=====================================================
查询成功，  用时: 10.49s
在没有年份限制情况下，查询到Zhen Gao发表的DBLP论文总数为135
=====================================================
正在查询（使用索引）.....
=====================================================
====================虚拟机2故障====================
==================正在查询备份====================
=====================================================
=====================================================
====================虚拟机4故障====================
==================正在查询备份====================
=====================================================
查询成功（使用本地索引）！ 用时: 0.87s
在没有年份限制情况下，查询到Zhen Gao发表的DBLP论文总数为135
=====================================================
```

3. 查询不存在的作者

```
==================================================
请输入作者姓名（输入exit退出):ZHEN Gao
==================================================
请输入起始年份(输入'*'代表不限制)： *
请输入截止年份(输入'*'代表不限制)： *
==================================================
正在查询（未使用索引）.....
查询成功，  用时：10.06s
在没有年份限制情况下，查询到ZHEN Gao发表的DBLP论文总数为0
==================================================
正在查询（使用索引）.....
查询成功（使用本地索引）！ 用时：0.62s
在没有年份限制情况下，查询到ZHEN Gao发表的DBLP论文总数为0
==================================================
```

七、项目反思

在本项目中，我们开发了一个分布式客户端系统，用于查询DBLP数据库中作者的发表论文数量，系统通过向多个服务器发送查询请求，并在服务器宕机时自动切换到备份服务器，以确保查询的高可用性和准确性。项目中，索引的构建和与传统文本搜索工具如grep命令的对比是两个关键的反思点。

索引构建

首先，索引的构建在项目中起到了至关重要的作用。通过建立结构化的索引，我们显著提升了查询的速度和效率。索引使得系统能够快速定位特定作者和年份的论文数据，避免了全表扫描，大幅减少了查询时间。此外，分布式索引设计增强了系统的可扩展性，使其能够处理更大规模的数据和更高的并发查询请求。然而，索引的构建也带来了诸多挑战，例如确保各节点间索引的一致性、优化索引的更新效率以及合理管理存储空间。通过实施分布式锁和一致性协议，我们成功解决了索引同步的问题，并通过增量更新和数据结构优化，提高了索引的维护效率。这些经验不仅提升了系统的性能，也为未来的优化和扩展提供了宝贵的参考。

与grep命令对比

另一方面，将我们的系统与传统的grep命令进行对比，进一步凸显了索引构建的重要性。grep作为Unix系统中广泛使用的文本搜索工具，具有强大的模式匹配和快速搜索能力，尤其在处理非结构化的文本数据时表现出色。然而，grep主要适用于单机环境下的简单搜索任务，其查询速度在面对大规模、结构化的数据集时，容易受到文件大小和匹配模式复杂度的影响。相比之下，我们的分布式查询系统通过预先构建索引，能够在常数时间内定位到相关数据，极大地提升了查询速度，尤其适用于需要实时或近实时响应的大规模数据库查询。此外，系统的分布式架构使其具备更高的可扩展性和容错能力，能够在多节点协作下高效处理复杂的查询需求，而grep在这方面则显得力不从心。

综上所述，索引的构建不仅显著提升了系统的查询性能和可扩展性，也让我们深刻认识到在处理大规模、结构化数据时，索引的重要性远超传统的文本搜索工具。同时，通过与grep的对比，我们更加明确了在不同应用场景下选择合适工具和技术的重要性。未来，我们计划进一步优化索引构建算法，增强系统的容错能力，并探索集成更高级的查询功能，以满足更复杂的用户需求。此外，考虑采用成熟的日志框架，将进一步提升日志管理的效率和灵活性。通过此次项目的实施与反思，我们不仅提升了系统的技术水平，也积累了宝贵的经验，为未来的系统设计和优化奠定了坚实的基础。