PHP 如何在两个大文件中找出相同的记录?

作者：ClassmateLiny
原文链接：https://ld246.com/article/1621676297551
来源网站：链滴

引言

给定a，b两个文件, 分别有x，y行数据, 其中(x, y均大于10亿), 机器内存限制100M，该如何找出其中同的记录?<!--more-->

思路

- 处理该问题的困难主要是无法将这海量数据一次性读内内存中.
- 一次性读不进内存中，那么是否可以考虑多次呢？如果可以，那么多次读入要怎么计算相同的值呢?
- 我们可以用分治思想， 大而化小。相同字符串的值hash过后是相等的, 那么我们可以考虑使用hash模, 将记录分散到n个文件中。这个n怎么取呢？ PHP 100M内存，数组大约可以存100w的数据, 那么a,b记录都只有10亿行来算, n至少要大于200。
- 此时有200个文件，相同的记录肯定在同一个文件中，并且每个文件都可以全部读进内存。那么可依次找出这200个文件中各自相同的记录，然后输出到同一个文件中，得到的最终结果就是a, b两个件中相同的记录。
- 找一个小文件中相同的记录很简单了吧，将每行记录作为hash表的key, 统计key的出现次数>=2就以了。

实操

10亿各文件太大了，实操浪费时间，达到实践目的即可。

问题规模缩小为：1M内存限制, a, b各有10w行记录, 内存限制可以用PHP的ini_set('memory_limit', 'M');来限制。

生成测试文件

生成随机数用于填充文件:

```
/**
 * 生成随机数填充文件
 * Author: ClassmateLin
 * Email: classmatelin.site@gmail.com
 * Site: https://classmatelin-1258942535.cos-website.ap-hongkong.myqcloud.com
 * @param string $filename 输出文件名
 * @param int $batch 按多少批次生成数据
 * @param int $batchSize 每批数据的大小
 */
function generate(string $filename, int $batch=1000, int $batchSize=10000)
{
    for ($i=0; $i<$batch; $i++) {
        $str = '';
        for ($j=0; $j<$batchSize; $j++) {
            $str .= rand($batch, $batchSize) . PHP_EOL; // 生成随机数
        }
        file_put_contents($filename, $str, FILE_APPEND);  // 追加模式写入文件
    }
}

generate('a.txt', 10);
generate('b.txt', 10);
```

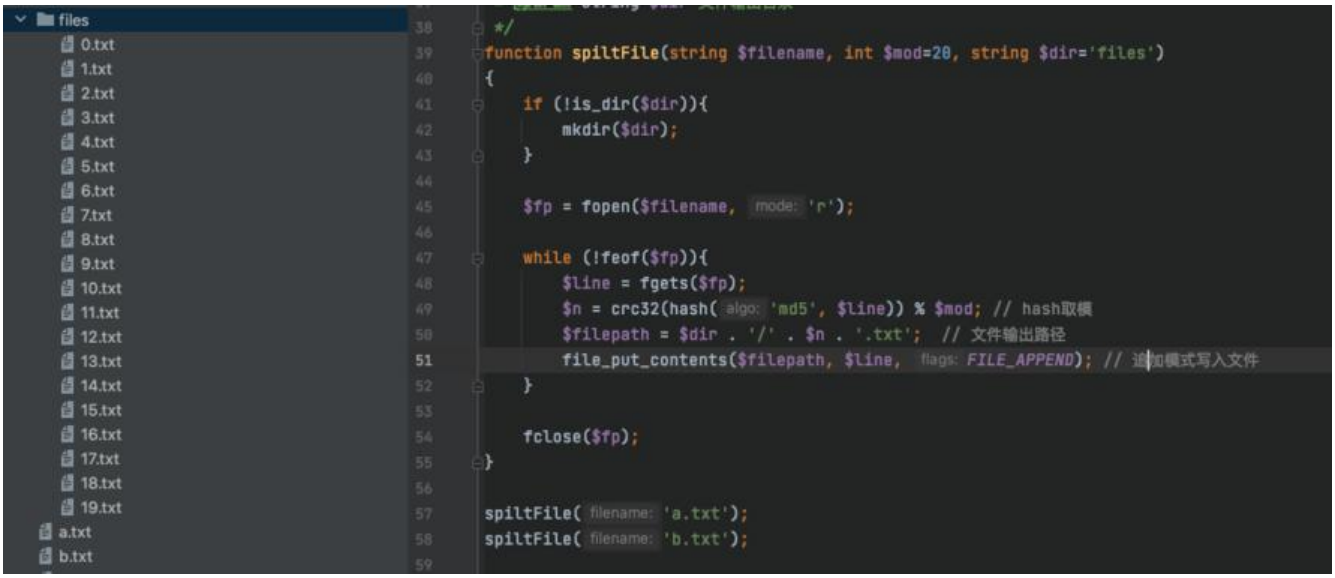
分割文件

- 将 a.txt, b.txt通过hash取模的方式分割到n个文件中.

```
/**
 * 用hash取模方式将文件分散到n个文件中
 * Author: ClassmateLin
 * Email: classmatelin.site@gmail.com
 * Site: https://classmatelin-1258942535.cos-website.ap-hongkong.myqcloud.com
 * @param string $filename 输入文件名
 * @param int $mod 按mod取模
 * @param string $dir 文件输出目录
 */
function spiltFile(string $filename, int $mod=20, string $dir='files')
{
    if (!is_dir($dir)){
        mkdir($dir);
    }

    $fp = fopen($filename, 'r');

    while (!feof($fp)){
        $line = fgets($fp);
        $n = crc32(hash('md5', $line)) % $mod; // hash取模
        $filepath = $dir . '/' . $n . '.txt';  // 文件输出路径
        file_put_contents($filepath, $line, FILE_APPEND); // 追加模式写入文件
    }

    fclose($fp);
```

```
}

spiltFile('a.txt');
spiltFile('b.txt');
```

- 执行 splitFile函数, 得到如下图files目录的20个文件。



查找重复记录

现在需要查找20个文件中相同的记录, 其实也就是找一个文件中的相同记录，操作个20次。

- 找一个文件中的相同记录:

```
/**
 * 查找一个文件中相同的记录输出到指定文件中
 * Author: ClassmateLin
 * Email: classmatelin.site@gmail.com
 * Site: https://classmatelin-1258942535.cos-website.ap-hongkong.myqcloud.com
 * @param string $inputFilename 输入文件路径
 * @param string $outputFilename 输出文件路径
 */
function search(string $inputFilename, $outputFilename='output.txt')
{
    $table = [];
    $fp = fopen($inputFilename, 'r');

    while (!feof($fp))
    {
        $line = fgets($fp);
        !isset($table[$line]) ? $table[$line] = 1 : $table[$line]++; // 未设置的值设1，否则自增
    }

    fclose($fp);

    foreach ($table as $line => $count)
    {
        if ($count >= 2){ // 出现大于2次的则是相同的记录，输出到指定文件中
```

```
            file_put_contents($outputFilename, $line, FILE_APPEND);
        }
    }
}
```

● 找出所有文件相同记录:

```
/**
 * 从给定目录下文件中分别找出相同记录输出到指定文件中
 * Author: ClassmateLin
 * Email: classmatelin.site@gmail.com
 * Site: https://classmatelin-1258942535.cos-website.ap-hongkong.myqcloud.com
 * @param string $dirs 指定目录
 * @param string $outputFilename 输出文件路径
 */
function searchAll($dirs='files', $outputFilename='output.txt')
{
    $files = scandir($dirs);

    foreach ($files as $file)
    {
        $filepath = $dirs . '/' . $file;
        if (is_file($filepath)){
            search($filepath, $outputFilename);
        }
    }
}
```

● 到这里已经解决了大文件处理的空间问题，那么时间问题该如何处理? 单机可通过利用CPU的多核处理，不够的话通过多台服务器处理。

完整代码

```
<?php
ini_set('memory_limit', '1M'); // 内存限制1M

/**
 * 生成随机数填充文件
 * Author: ClassmateLin
 * Email: classmatelin.site@gmail.com
 * Site: https://classmatelin-1258942535.cos-website.ap-hongkong.myqcloud.com
 * @param string $filename 输出文件名
 * @param int $batch 按多少批次生成数据
 * @param int $batchSize 每批数据的大小
 */
function generate(string $filename, int $batch=1000, int $batchSize=10000)
{
    for ($i=0; $i<$batch; $i++) {
        $str = '';
        for ($j=0; $j<$batchSize; $j++) {
            $str .= rand($batch, $batchSize) . PHP_EOL; // 生成随机数
        }
        file_put_contents($filename, $str, FILE_APPEND);  // 追加模式写入文件
    }
```

```
}



/**
 * 用hash取模方式将文件分散到n个文件中
 * Author: ClassmateLin
 * Email: classmatelin.site@gmail.com
 * Site: https://classmatelin-1258942535.cos-website.ap-hongkong.myqcloud.com
 * @param string $filename 输入文件名
 * @param int $mod 按mod取模
 * @param string $dir 文件输出目录
 */
function spiltFile(string $filename, int $mod=20, string $dir='files')
{
    if (!is_dir($dir)){
        mkdir($dir);
    }

    $fp = fopen($filename, 'r');

    while (!feof($fp)){
        $line = fgets($fp);
        $n = crc32(hash('md5', $line)) % $mod; // hash取模
        $filepath = $dir . '/' . $n . '.txt';  // 文件输出路径
        file_put_contents($filepath, $line, FILE_APPEND); // 追加模式写入文件
    }

    fclose($fp);
}



/**
 * 查找一个文件中相同的记录输出到指定文件中
 * Author: ClassmateLin
 * Email: classmatelin.site@gmail.com
 * Site: https://classmatelin-1258942535.cos-website.ap-hongkong.myqcloud.com
 * @param string $inputFilename 输入文件路径
 * @param string $outputFilename 输出文件路径
 */
function search(string $inputFilename, $outputFilename='output.txt')
{
    $table = [];
    $fp = fopen($inputFilename, 'r');

    while (!feof($fp))
    {
        $line = fgets($fp);
        !isset($table[$line]) ? $table[$line] = 1 : $table[$line]++; // 未设置的值设1，否则自增
    }
```

```
    fclose($fp);

    foreach ($table as $line => $count)
    {
        if ($count >= 2){ // 出现大于2次的则是相同的记录，输出到指定文件中
            file_put_contents($outputFilename, $line, FILE_APPEND);
        }
    }
}

/**
 * 从给定目录下文件中分别找出相同记录输出到指定文件中
 * Author: ClassmateLin
 * Email: classmatelin.site@gmail.com
 * Site: https://classmatelin-1258942535.cos-website.ap-hongkong.myqcloud.com
 * @param string $dirs 指定目录
 * @param string $outputFilename 输出文件路径
 */
function searchAll($dirs='files', $outputFilename='output.txt')
{
    $files = scandir($dirs);

    foreach ($files as $file)
    {
        $filepath = $dirs . '/' . $file;
        if (is_file($filepath)){
            search($filepath, $outputFilename);
        }
    }
}

// 生成文件
generate('a.txt', 10);
generate('b.txt', 10);

// 分割文件
spiltFile('a.txt');
spiltFile('b.txt');

// 查找记录
searchAll('files', 'output.txt');
```