对 KARP-RABIN 串匹配随机算法的改进
Improvement of KARP-RABIN Randomized Strings-matching Algorithm

何建强
He Jianqiang

(广西民族学院数学与计算机科学系 南宁 530006)
(Dept. of Mathematic and Computer Science, Guangxi University for Nationalities, Nanning, 530006)

摘要 介绍一种 KARP-RABIN 串匹配随机算法中改进的指印函数,以及对指印数值做快速片段比较的方法,减少对正文字符的读取,提高 KR 算法的搜索速度。

关键词 KR 算法 串匹配 指印函数
中图法分类号 TP301.6 A

Asbtract For more quickly searching in the use of the KARP-RABIN randomized strings-matching algorithm, the improved fingerprint function and the method to compare the segments rapidly are introduced.

Key words KR algorithm, strings-matching, fingerprint function

KARP-RABIN 串匹配随机算法的基本思想是先定义一个指印函数,将模式串映射成一个比模式串短得多的指印(二进制位串数据),然后将正文中每一个长度为 m 的子串也映射成为一个比子串短得多的指印(二进制位串数据),算法的匹配比较过程是先比较模式串的指印函数值和正文子串的指印函数值,只有两者相等时才比较模式串与正文子串是否确实匹配。

使用这种方法能以比较数字是否相等来代替费时较多的串比较过程,提高搜索速度的关键是指印函数值易于求出。该算法可以用前一个指印函数值递推求出下一个指印函数值,使用如下递推公式求得:

$$h[i+1] = ((h[i] - xx * \mathrm{asc}(t[i])) * d + \mathrm{asc}(t[i+m])) \bmod q,$$

其中 $h[i]$ 为前一个指印函数的值,$t[i]$ 为正文中第 i 个字符,m 为子串长度,使用该递推函数可方便用于任意长度子串,但计算量稍大[1,2]。

事实上,我们考察一般的串匹配情形,对于一个 32 位二进制的指印值在比较中可能匹配的概率只有 40 亿分之一,亦即我们只要把子串看作一个长的二进制数,从子串中按一定规律抽取二进制位组合作为指印数值一样可以取得相似的效果,而完全不必采用上述如此复杂的

2002-06-28 收稿。

指印递推函数求值。

我们可以编写程序根据模式子串的长度自动判定应该每个字符取几位二进制数，为便于计算机指令处理，可取字符最低的某些位。

设 m 为子串长度，w 为取某字符最低位位数。

计算公式为

$w = (\text{int})(0.99 + 32/m)$，

对于 $m > 32$ 的子串，只取前面32个字符的最低一位作为指印数值。

某些子串可能没有做到每个字符至少取一位，那么能取到多少个字符就只用多少个，因为在串匹配中求指印数值是影响搜索速度的关键，即使只用16字符出现匹配的概率也极低，也就是很少会出现需要检查模式子串和正文子串是否相等的情况，因而简单快捷地获取指印数值可明显提高搜索速度。

例如：

子串 "computer"，可以每个字符取最低 4 位组成指印函数值。子串 "using the Microsoft Foundation classes" 取前32个字符的每个字符最低 1 位组成指印函数值。子串 "Windows resources"，每个字符取最低 2 位组成指印函数值，共取32位，只取前16个字符。

由于可以用逻辑与指令十分方便地获取字节的最低几位，而已知前一个指印函数的值 $h[i]$ 可用一条移位指令和与指令即可求出下一个指印函数值 $h[i+1]$。

设 EBX = $h[i]$ CL = w AL = $t[k]$ （即下一子串的新字符，其中当 $m >= 32$ 时 $k = i + 32$，否则 $k = i + m$）

预先求取二进制位对应的与数 DX：

```
MOV CH,0xFF
SHL CH,CL
NOT CH
```

用前一个指印函数的值 BX 及下个字符 AL 求出下一个指印函数值。

```
AND AL,CH
SHL EBX,CL
OR  EBX,EAX
```

如果取的是字符的最低一位，则只用两条移位指令即可完成递推过程，速度更快：

```
SHR AL,1
SHL EBX,1
```

在指印数值比较时，可以使用片段匹配法提高搜索速度，这样可以只提取正文子串的部分指印数值。

例如：设模式子串的指印数值为

* *

当前正文子串的指印数值为

# #

如果发现不匹配，可不必立即求下一个指印数值，而是将当前正文子串的指印数值左移一位，只做高8位比较，

（下转第160页）

3.2.3 长期重构目标算法

这要求系统维护一张系统重构触发条件表，格式如表3。

表3 系统重构触发条件

节点地址	失效次数	接口传输率过大次数	CPU负载过重次数	总次数
1	3	4	1	8
…	…	…	…	…
255	0	2	8	10

此算法要求每次系统重构算法完成后，根据触发条件重填此表。一段时间后根据此表列出所有信息，给出失效次数节点排序、CPU负载过重节点排序、网络接口传输率过大节点排序、整体最不稳定节点排序。以此作为硬件升级建议。

4 结束语

本文提出了一种分布式系统的重构算法，应用该算法能够使分布式系统达到可重构性。它保证了当分布式系统的某些部分失效时，整个系统能够继续正常运行，同时还保证调整系统负载，使CPU负载和网络通信量相对平衡。本文对于研究分布式系统有很好的参考价值。

参考文献

1 朱海滨. 分布式系统原理与设计. 长沙：国防科技大学出版社，1997. 15.

2 Andrew S. Tanenbaum. 计算机网络. 北京：清华大学出版社，1998. 209.

（责任编辑：黎贞崇）

（上接第156页）

* *

_

* *

_。

如此重复移位、比较，最多移至只剩8个二进制位，平均出现匹配的概率只有1/256。当出现匹配时再取部分低位指印数值做低位比较，这样可以跳过部分字符，提高搜索比较速度。

算法复杂性分析：

如果不考虑在指印匹配时字符匹配比较所需时间，时间复杂性应是$O(n+m)$，n为正文长度。本算法对KR算法中指印函数的改进提高了指印数值的生成速度，片段匹配可跳过部分字符，对搜索速度的贡献约为1倍。

参考文献

1 Richard M. Karp, Michael O. Rabin, Efficient randomized pattern-matching algorithms, IBM Research and Development, 1987, 31(2): 249～260.

2 苏德富，钟 诚. 计算机算法设计与分析. 北京：电子工业出版社，2001.

（责任编辑：蒋汉明）