基于 LZO 的快速解压算法解读

王松，房利国，韩炼冰等. 一种快速解压的无损压缩算法 [J]. 通信技术，20

Kingtous

kingtous@hust.edu.cn

2021 年 10 月 26 日

无损压缩算法瓶颈

1. 字符串搜索匹配
    1. 如何找到一个速度、效果动态平衡的字符串匹配结果？

无损压缩发展关系图

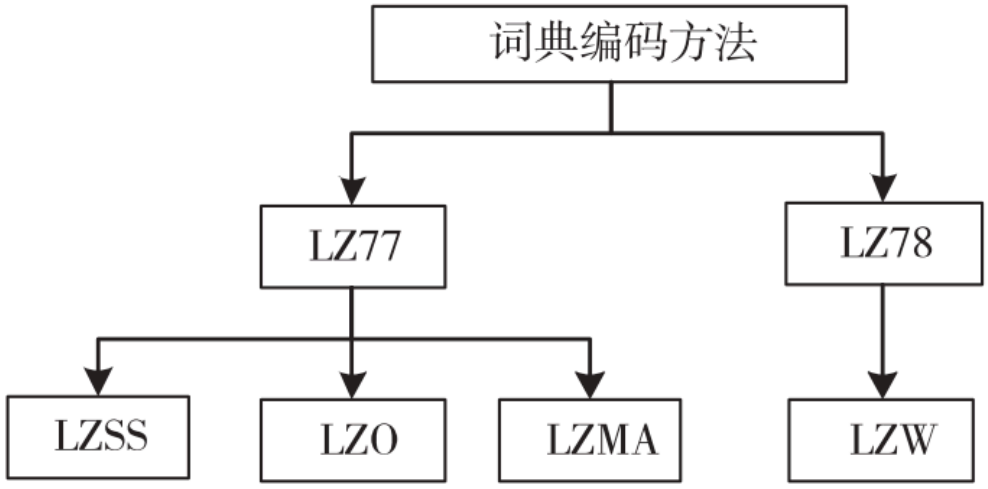


图 1　词典编码方法

1. LZ77
   1. 将已处理的原始数据流作为词典的输入
2. LZ78
   1. 将已处理的数据编入一个字典

无损压缩发展关系图

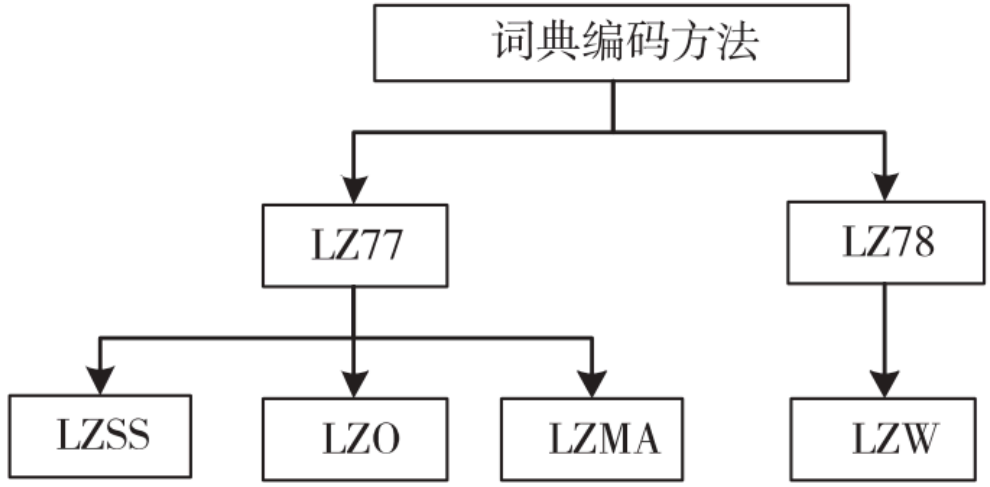


图 1　词典编码方法

对于 LZO 算法，目前最常用的版本为 LZ01X_1。

1. 使用 hash table 来进行数据匹配。
2. 以当前待处理数据的前 4B 数据进行哈希匹配。

LZO 算法的两次匹配

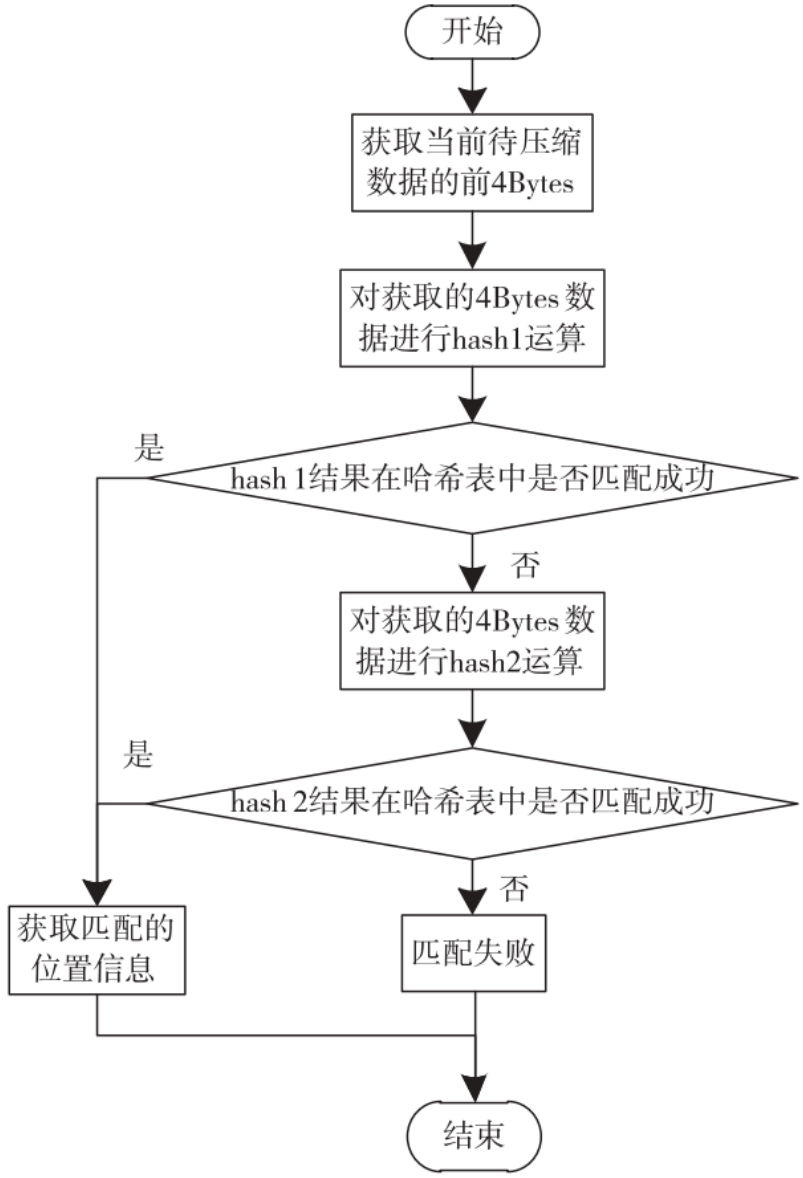


图 2　LZO 算法的两次匹配过程

LZO 算法的压缩格式

匹配长度 4B $\leq$ *len*(*match*) $\leq$8B 时的压缩格式

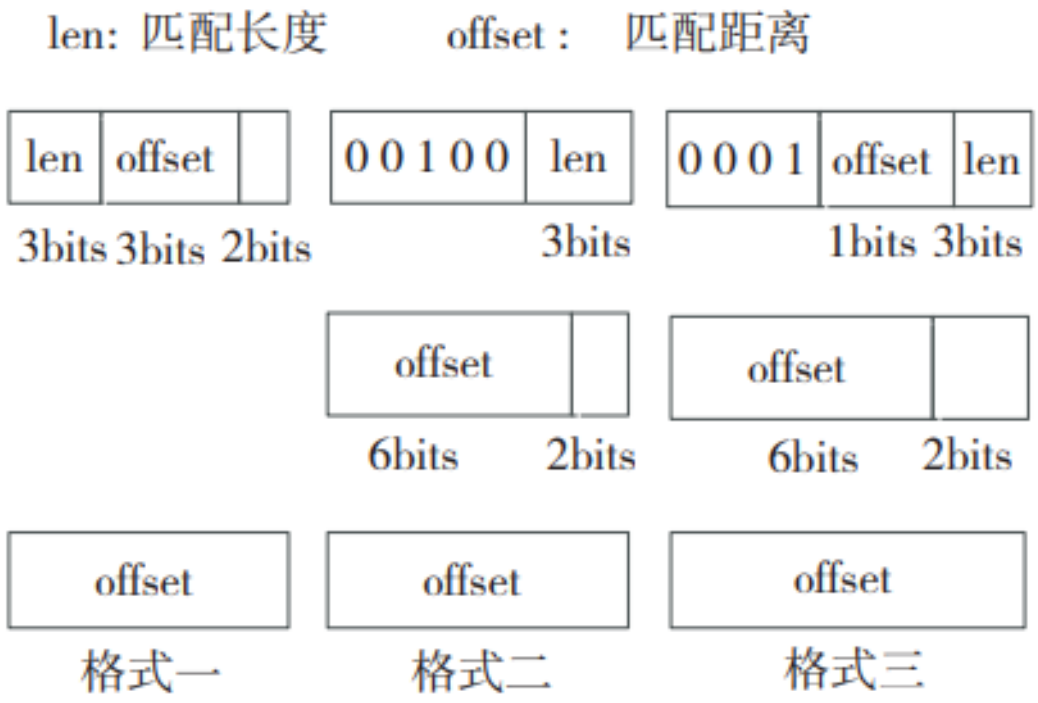


图 3　匹配长度小于等于 8 Bytes 时压缩格式

offset(*match*) 表示匹配距离，则：

1. 格式一：$offset(match) \leq 2KB$
2. 格式二：$2KB < offset(match) \leq 16KB$
3. 格式三：$16KB < offset(match) \leq 48KB$

LZO 算法的压缩格式

匹配长度 >8B 时的压缩格式

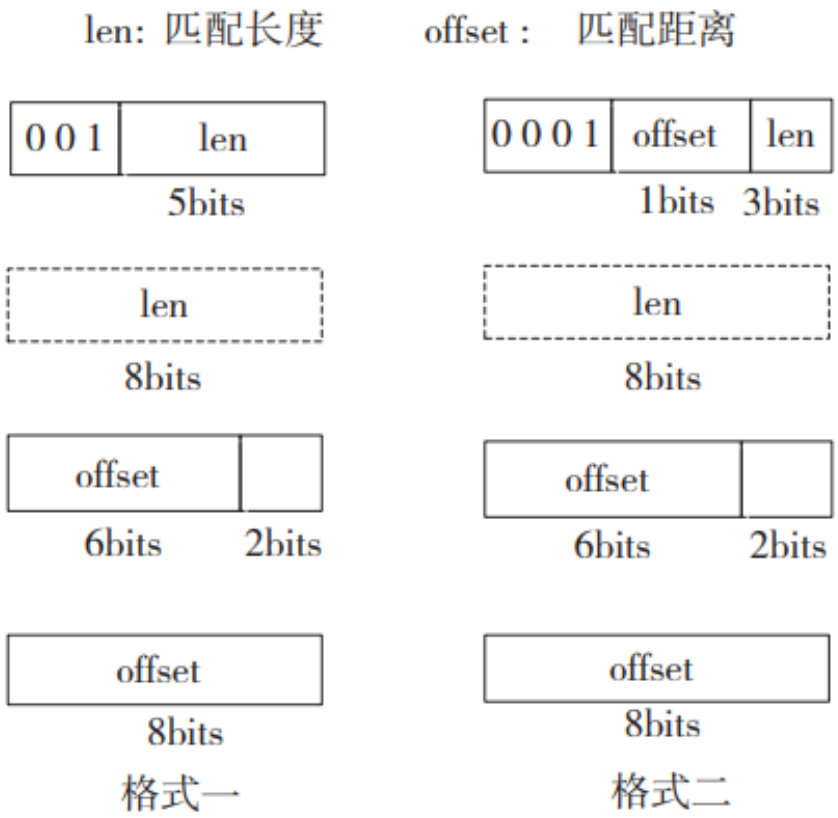


图 4 匹配长度大于 8 Bytes 时压缩格式

offset(*match*) 表示匹配距离，则：

1. 格式一：$offset(match) \leq 16KB$
2. 格式二：$16KB \leq offset(match) \leq 48KB$

LZO 匹配距离综述

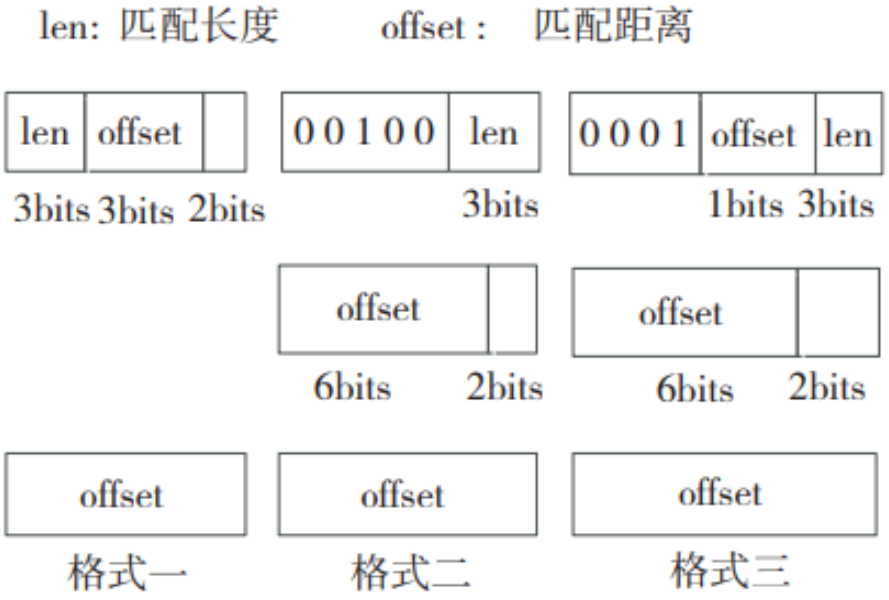


图 3　匹配长度小于等于 8 Bytes 时压缩格式

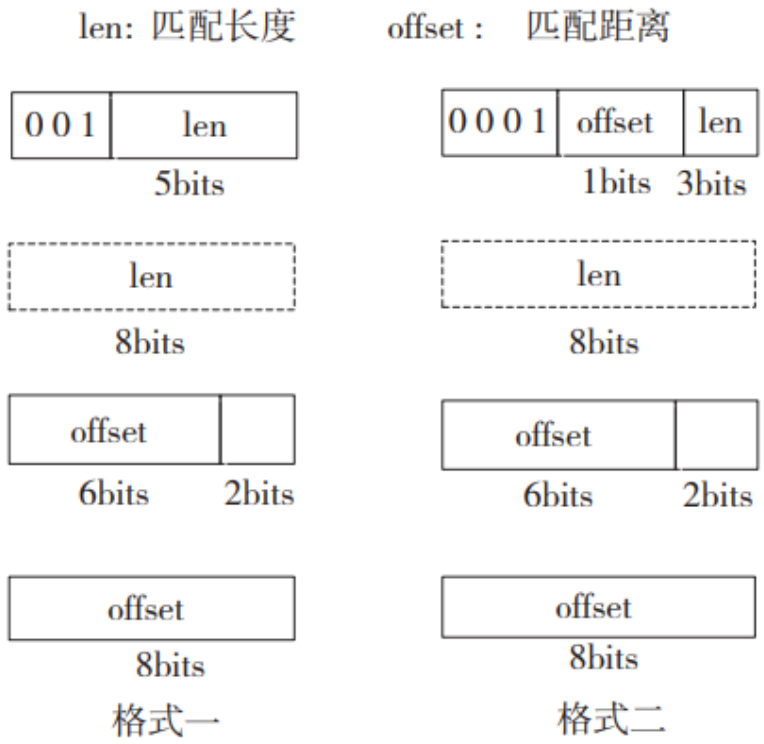
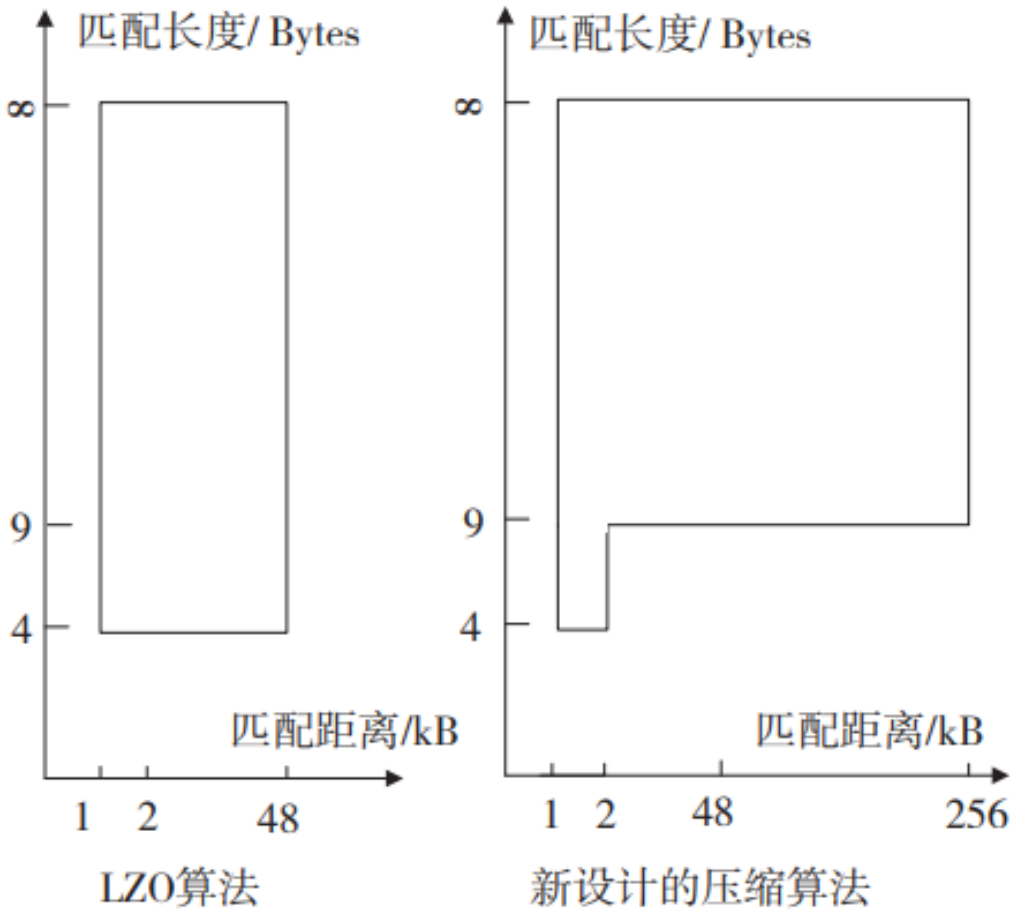


图 4　匹配长度大于 8 Bytes 时压缩格式

定义 sz 为首字节的无符号整数值，则：

1. $sz \geq 64(\mathbf{0}1000000)$ 左图格式一
2. $32(\mathbf{00}100000) \leq sz < 64(\mathbf{0}1000000)$ 左图格式二 or 右图格式一
3. $16(\mathbf{000}10000) \leq sz < 32(\mathbf{00}100000)$ 左图格式三 or 右图格式二
4. $sz < 16(\mathbf{000}10000)$ 未进行压缩的数据

新算法启发



1. 较短数据的压缩处理会增加解压程序的开销
2. 48KB 匹配空间之外，会出现较多的匹配数据无法被压缩

新算法匹配流程

可配置的多页词典查找方式

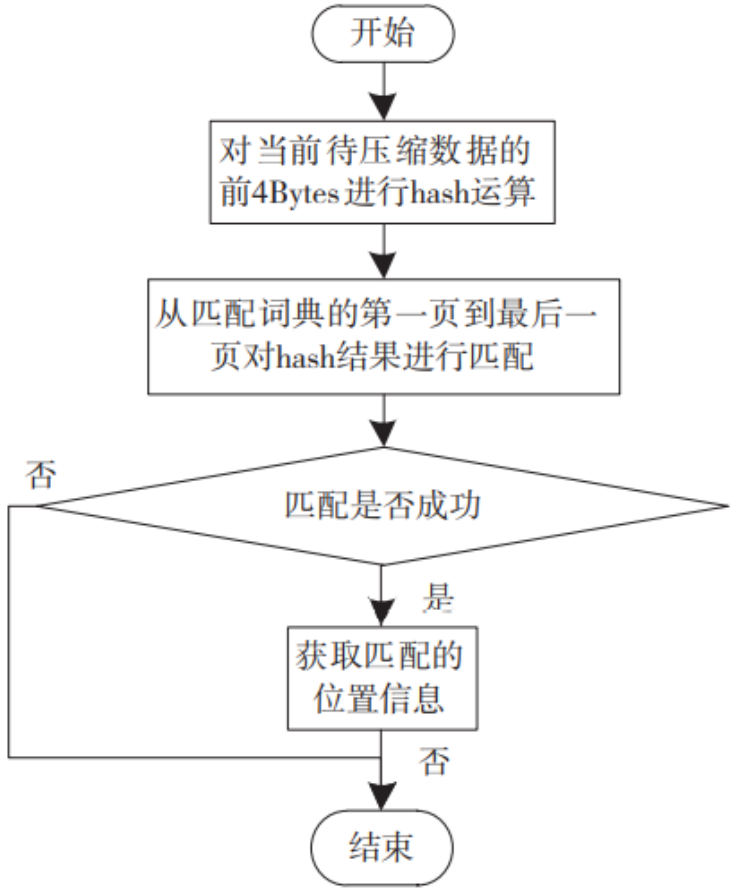


图 6　新算法的匹配过程

1. 分别在每页上进行匹配该字符串出现的可能位置

新算法的压缩格式

匹配距离 <2KB 时的压缩格式

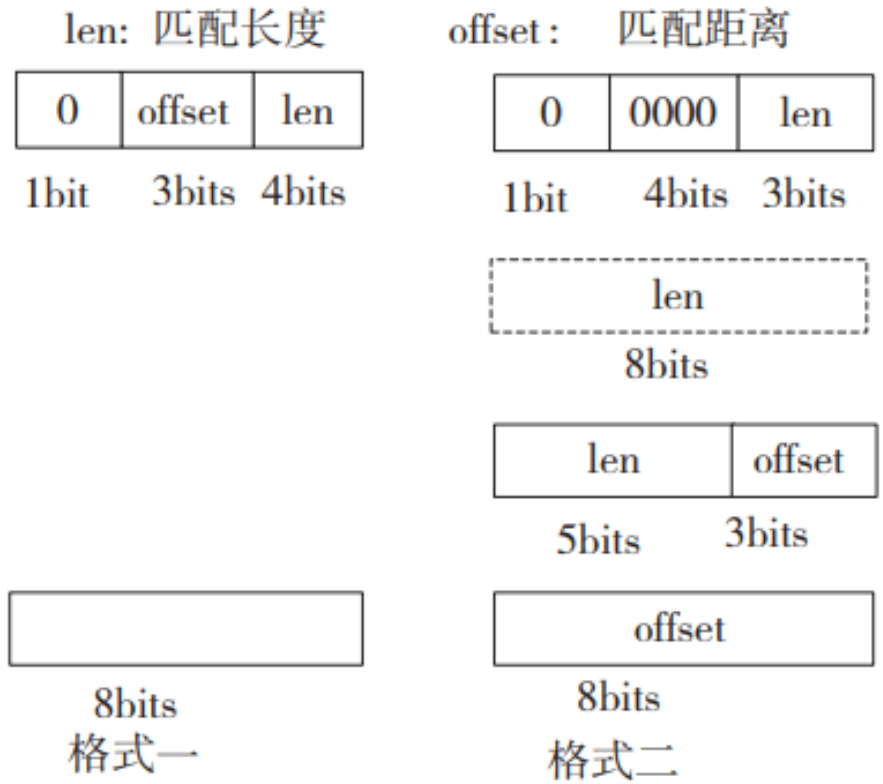


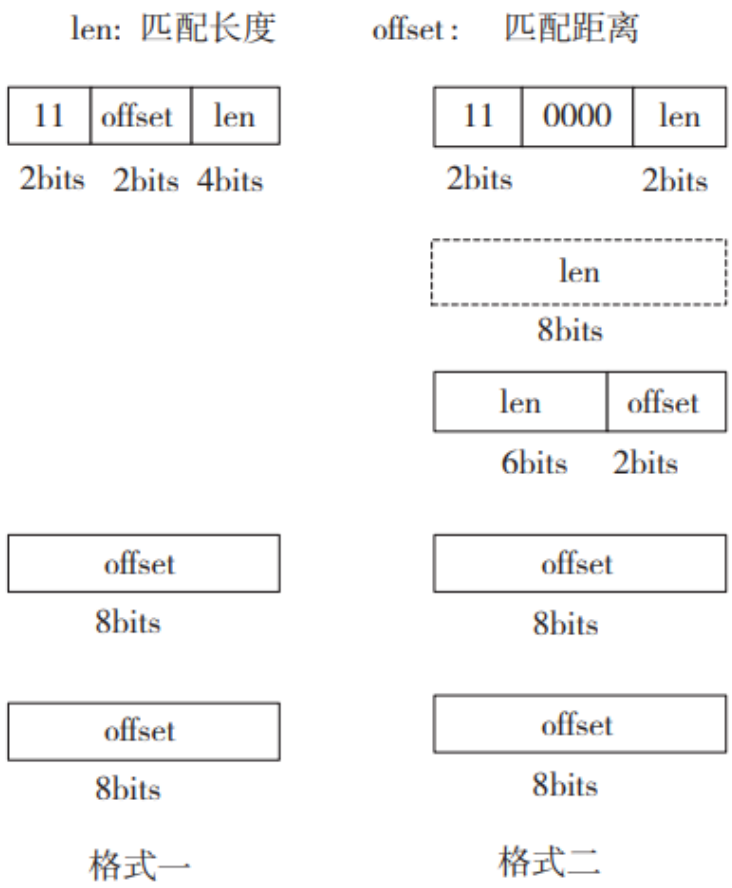
图 7　匹配距离小于 2 kB 时的压缩格式

len(*match*) 表示匹配长度，则：

1. 格式一：$len(match) \leq 17B$
2. 格式二：$len(match) > 17B$

新算法的压缩格式

匹配距离 ≥2KB 时的压缩格式



len(*match*) 表示匹配长度，则：

1. 格式一：$len(match) \leq 18B$
2. 格式二：$len(match) > 18B$

新算法匹配距离综述

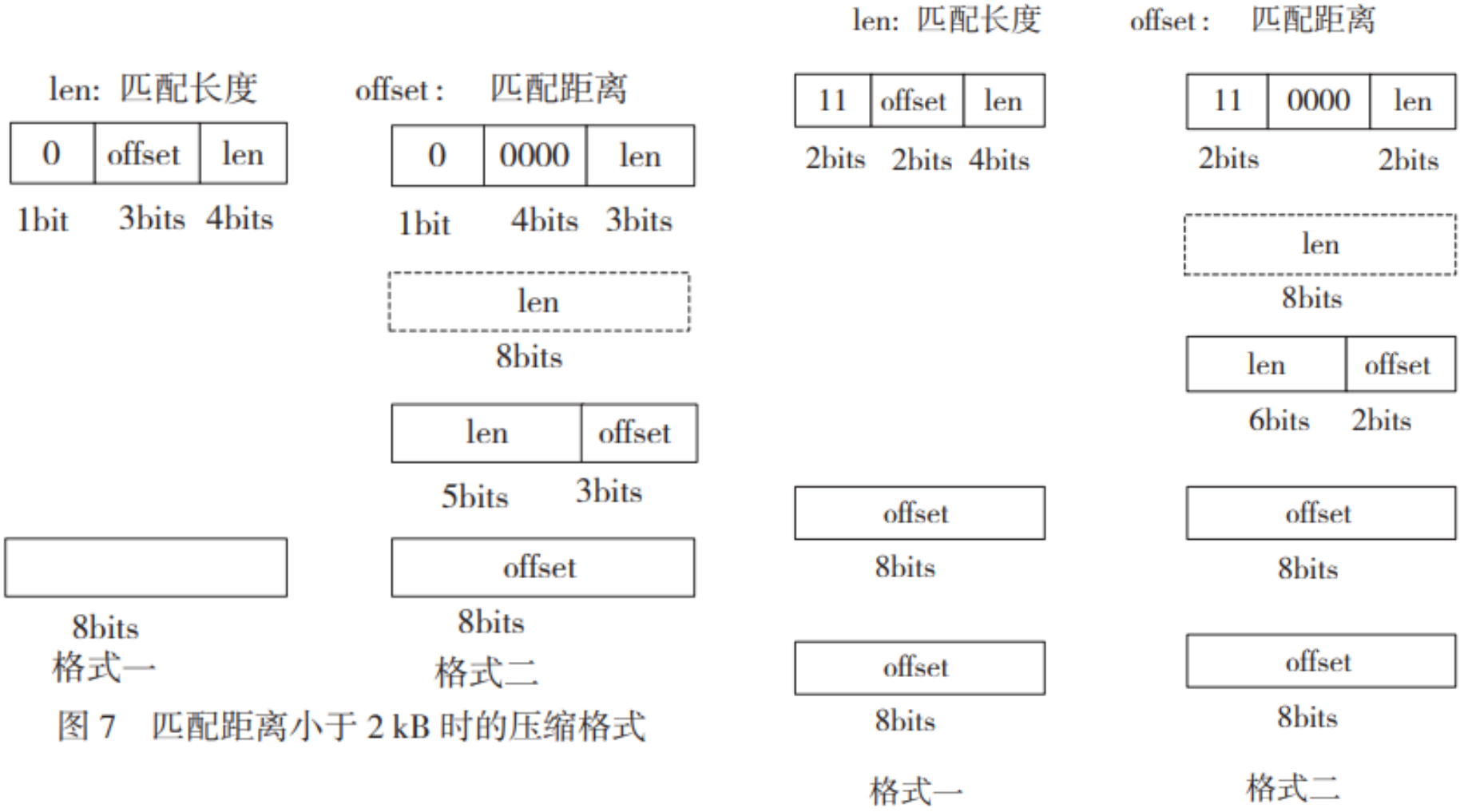


图 7　匹配距离小于 2 kB 时的压缩格式

1. 降低压缩块格式数量
2. 判断简单，有利于提高速度

压缩率比较

表 1　新算法与 LZO 算法压缩率比较

文件名称	文件大小	LZO 算法		新算法	
		压缩块数量	压缩率	压缩块数量	压缩率
文件 1.txt	10 kB	925	0.388	726	0.375
表格 1.xlsx	19 kB	323	0.822	201	0.821
文档 1.docx	24 kB	243	0.884	142	0.883
程序 1.dll	28 kB	1 414	0.256	610	0.281
表格 2.xls	33 kB	2 167	0.294	1 495	0.301
文档 2.doc	37 kB	1 949	0.269	1 081	0.281
表格 3.xlsx	126 kB	877	0.913	536	0.912
程序 2.dll	241 kB	27 958	0.514	15 775	0.561
表格 4.xls	434 kB	29 602	0.247	24 446	0.262
PDF1.pdf	549 kB	3 617	0.870	2 651	0.869
PDF2.pdf	573 kB	4 068	0.919	2 412	0.916
工具 1.exe	717 kB	2 262	0.981	760	0.982
程序 3.lib	1 MB	112 512	0.412	58 152	0.416
程序 4.lib	1.2 MB	86 192	0.306	53 572	0.299
文件 2.nh	1.6 MB	17 903	0.923	3 823	0.908
PPT1.pptx	1.7 MB	10 741	0.483	8 700	0.483
文件 3.kdh	1.8 MB	15 132	0.875	8 569	0.876
工具 2.exe	1.9 MB	200 256	0.613	84 202	0.616
文档 3.doc	2.5 MB	104 161	0.727	31 916	0.512
文件 4.nh	2.6 MB	29 396	0.937	7 733	0.923
工具 3.exe	3.5 MB	6 112	0.998	2 584	0.999
文档 4.docx	5.5 MB	46 581	0.953	26 465	0.950
PDF3.pdf	8.4 MB	64 181	0.831	39 989	0.809
文件 5.nh	11.7 MB	213 168	0.851	108 940	0.852

解压速度比较

表 2　新算法与 LZO 解压速度比较

文件名称	LZO 算法解压时间 /ms	新算法解压时间 /ms	解压速度比较 /（%）
文件 1.txt	0.41	0.39	+4.8
表格 1.xlsx	0.58	0.51	+12.1
文档 1.docx	0.71	0.62	+12.7
程序 1.dll	0.96	0.83	+13.5
表格 2.xls	1.21	1.12	+7.4
文档 2.doc	1.29	1.13	+12.4
表格 3.xlsx	3.56	3.12	+12.3
程序 2.dll	10.35	9.33	+9.85
表格 4.xls	16.11	15.25	+5.3
PDF1.pdf	15.36	13.56	+11.7
PDF2.pdf	16.08	14.07	+12.5
工具 1.exe	19.59	17.05	+12.9
程序 3.lib	43.35	37.39	+13.7
程序 4.lib	45.14	40.56	+10.1
文件 2.nh	48.07	40.65	+15.4
PPT1.pptx	48.25	42.27	+12.3
文件 3.kdh	53.06	46.39	+12.6
工具 2.exe	83.17	67.49	+18.8
文档 3.doc	86.14	68.19	+20.8
文件 4.nh	76.68	65.07	+15.1
工具 3.exe	96.88	84.86	+12.4
文档 4.docx	157.32	137.45	+12.6
PDF3.pdf	237.76	207.73	+12.6
文件 5.nh	350.57	299.57	+14.5

优化总结

1. 放弃一些小块数据压缩处理
2. 在更大的空间上进行搜索匹配