清华大学

流量大数据检索

位图索引编码机制研究

温禹豪

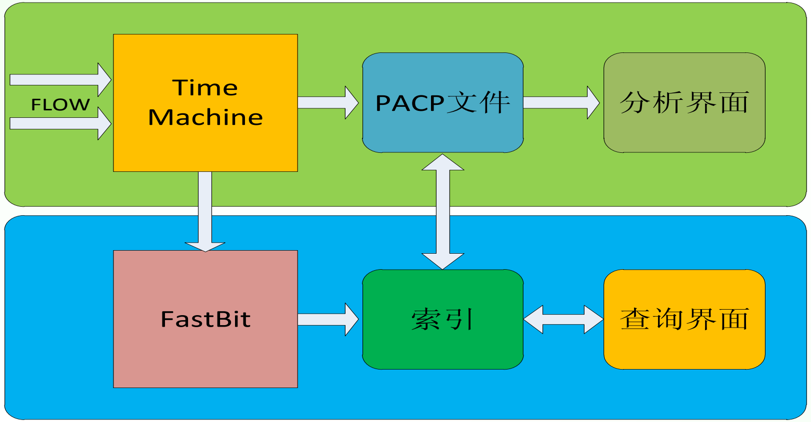
2014年03月22日

网流归档与查询系统-TIFAflow

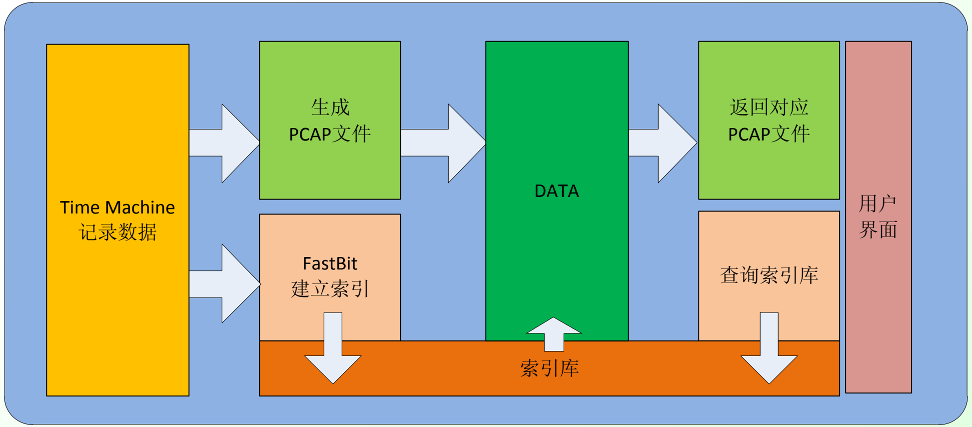
- 研究问题：网络安全事件难以追溯定位，如披露的美国安全局的网络攻击事件
- 研究挑战：
  - 1)骨干链路速率高，流量大，存储速度慢
  - 2)索引空间消耗大，查询速度慢

J. Li et al., TIFA: Enabling Real-Time Querying and Storage of Massive Stream Data. Proc. of International Conference on Networking and Distributed Computing (ICNDC), 2011.

- 研究创新：1)基于流粒度的存储与查询；2)位图索引编码算法



TIFA 系统结构

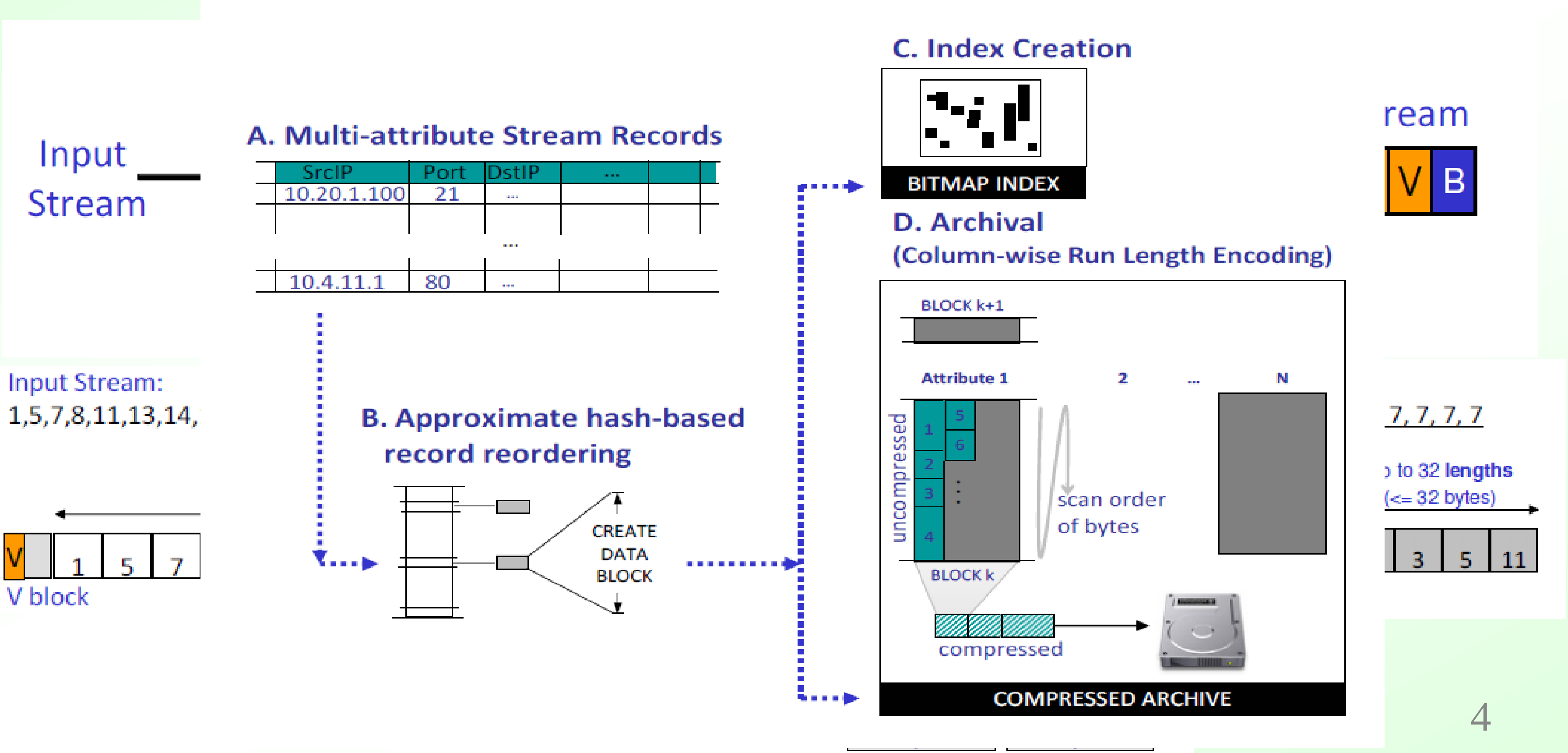


工作流程

位图索引及其特点

row ID	X	bitmap index b_0 =0	b_1 =1	b_2 =2	b_3 =3
1	0	1	0	0	0
2	1	0	1	0	0
3	3	0	0	0	1
4	2	0	0	1	0
5	3	0	0	0	1
6	3	0	0	0	1
7	1	0	1	0	0
8	3	0	0	0	1

RasterZip压缩与查询



算法全称

- **WAH: Word-Aligned Hybrid**
- **PLWAH: Position List Word-Aligned Hybrid**
- **COMPAX: COMPressed Adaptive indeX format**
- **SECOMPAX: Scope Extended COMPressed Adaptive indeX format**
- **ICX: Improved CompaX**
- **MASC: MAximized Stride with Carrier**

常用术语

- Chunk: 31比特为单位的块
- Word: 32比特字
- Fill： 一个word/chunk全部为0或1
- Literal: 一个word/chunk不全为0或1

经典编码算法1-WAH(Word-Aligned Hybrid)

- 每31bits 进行一次压缩，全0为0-fill，全1为1-fill，否则为literal.
- 相邻的一串0-fill压成一个0-fill,相邻的一串1-fill压成1个1-fill.

128 bits	1*1,20*0,3*1,79*0,25*1			
31-bit groups	1,20*0,3*1,7*0	62*0	10*0,21*1	4*1
literal (hex)	40000380	00000000 00000000	001FFFFF	0000000F
WAH (hex)	40000380	80000002	001FFFFF	0000000F

uncompressed (in 31-bit groups)					
A	40000380	00000000	00000000	001FFFFF	0000000F
B	7FFFFFFF	7FFFFFFF	7C0001E0	3FE00000	00000003
C	40000380	00000000	00000000	00000000	00000003
compressed					
A	40000380	80000002		001FFFFF	0000000F
B	C0000002		7C0001E0	3FE00000	00000003
C	40000380	80000003			00000003

经典算法2-PLWAH(Position List Word-Aligned Hybrid)

- **WAH的基础上：piggyback(携带)**
- **Piggyback:如果0-fill序列下一个literal中只有一个1，那么将这个1 piggyback到前一个fill word中.**

经典算法2-PLWAH

Uncompressed bitmap organized in groups of 31 bits:

```
0000000000 0000000000 0000000000 0
0000000000 0000000001 0000000000 0
0000000000 0000000000 0000000000 0
0000000000 0000000000 0000000000 0
0000000100 0000000000 0000000000 0
0000000000 0000000100 0000000000 0
```

31 bits

11 bits

Merging consecutive homogenous groups:

```
0000000000 0000000000 0000000000 0
0000000000 0000000001 0000000000 0
0000000000 0000000000 0000000000 0
0000000000 0000000000 0000000000 0
0000000100 0000000000 0000000000 0
0000000000 0000000100 0000000000 0
```

2 groups merged

Encoding 32 bits fill words:

1 0 0000000000 0000000000 0000000001	0 Fill word, counter = 1
0 0000000000 0000000001 0000000000 0	Literal word
1 0 0000000000 0000000000 0000000010	0 Fill word, counter = 2
0 0000000100 0000000000 0000000000 0	Literal word
0 0000000000 0000000100 0000000000 0	Literal word

Encoding sparse 32 bits literal words:

1 0 1010000000 0000000000 0000000001	0 Fill word, cnt = 1, pos = 20
1 0 0100000000 0000000000 0000000010	0 Fill word, cnt = 2, pos = 8
0 0000000000 0000000100 0000000000 0	Literal word

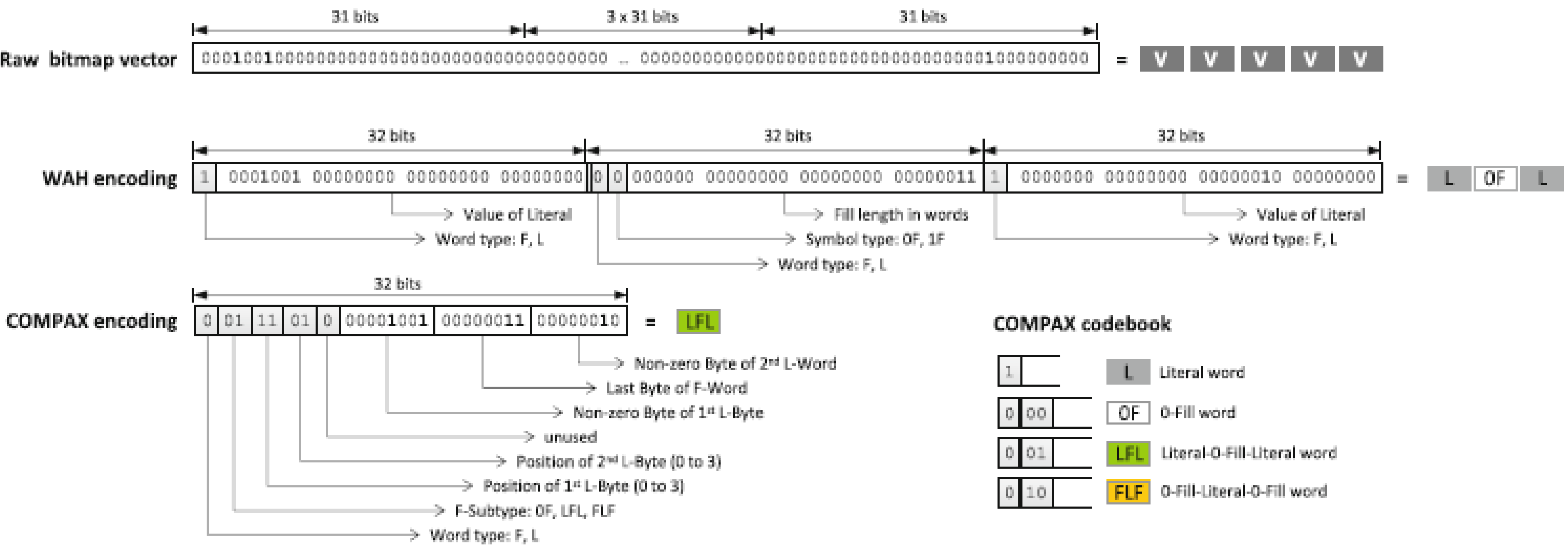
经典算法2-PLWAH

- **优点：减少WAH中literal出现的数量，节省空间**
- **进一步改进：**
- **1.piggyback最多可以带5个(position)**
- **2.Adaptive Counter (相当于将2个或更多fill word的counter合并)**

经典算法3-COMPAX(COMPressed Adaptive indeX format)

- 在WAH的基础上进行改进
- 增添了码本，加入了LFL(literal-fill-literal)以及FLF(fill-literal-fill)
- LFL：两个L都只有一个dirty byte（dirty byte: 相对全0 chunk,所有的非零比特在同一个byte内），F限定为0-fill
- FLF：L只有一个dirty byte

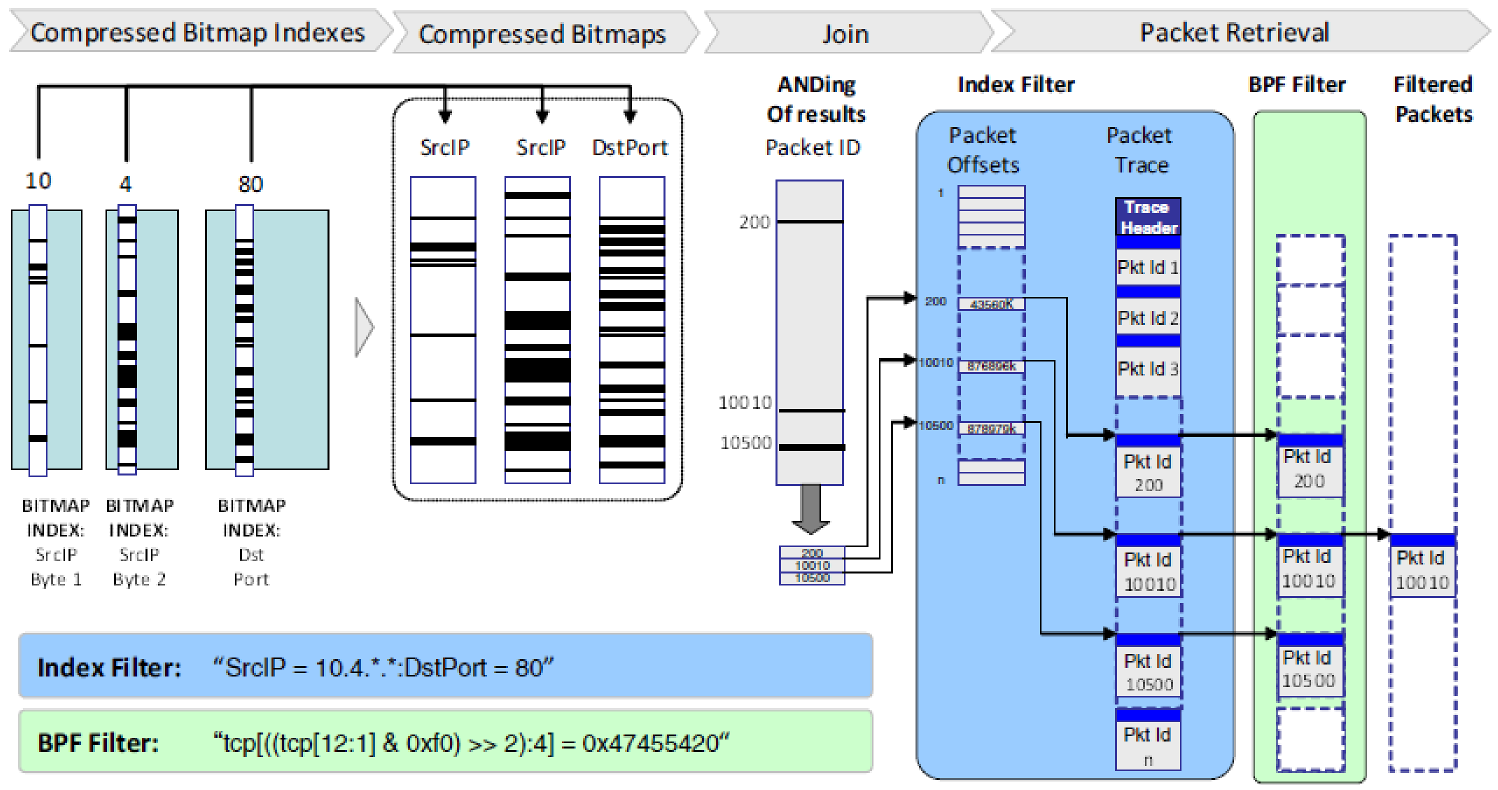
经典算法3-COMPAX



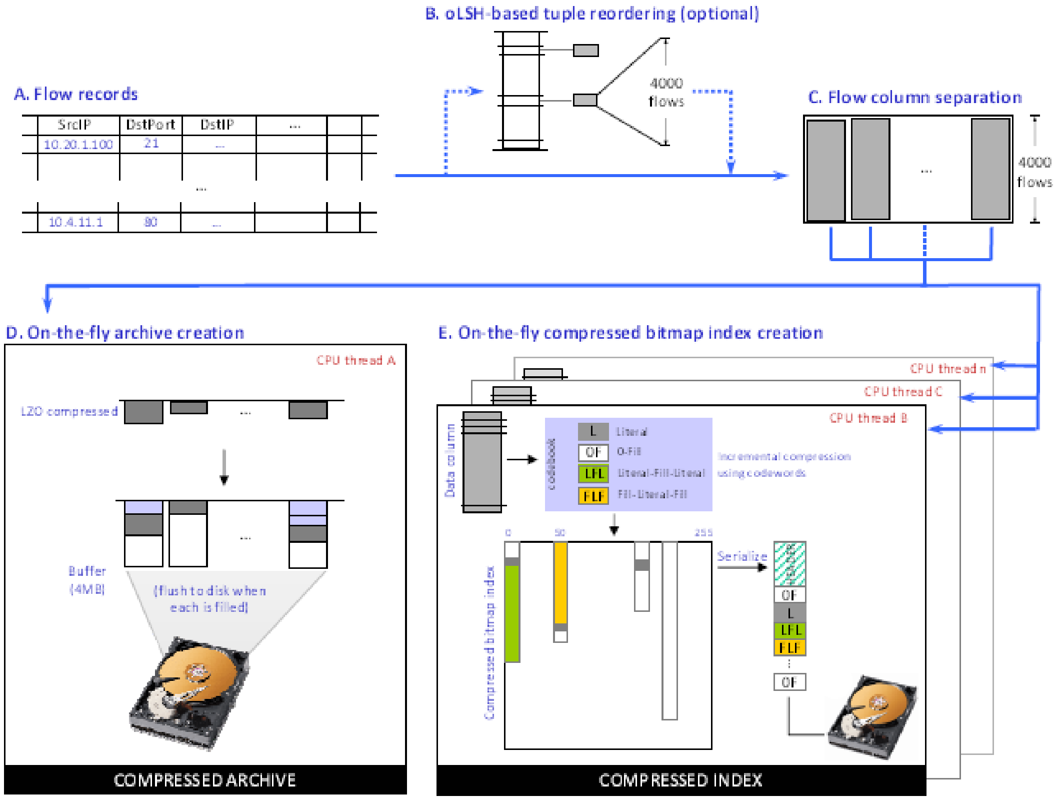
经典算法3-COMPAX

- 优点：增加编码类型，提高压缩率
- 进一步改进：
- **COMPAX2:增加1-fill以及LFL中literal-1 fill-literal类型**

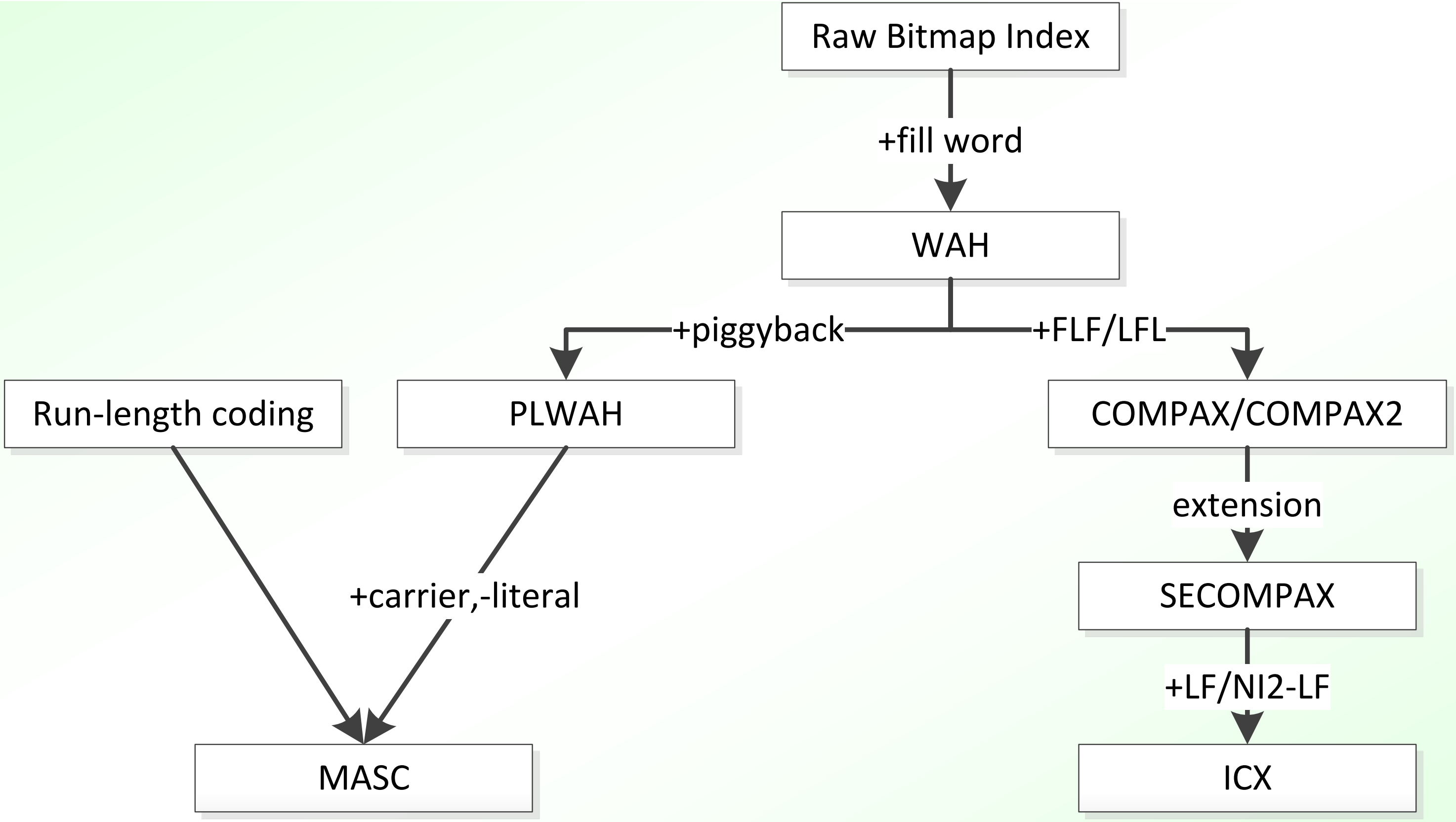
COMPAX应用-PcapIndex



COMPAX应用：NET-FLi



算法之间关联-roadmap



新算法-SECOMPAX(Scope Extended COMPressed Adaptive indeX format)

- 以COMPAX改进版——COMPAX2为基础
- 1.明确定义带dirty byte的成为nearly identical literal (NI-L)，并改变dirty byte定义，原来是和全零chunk比所有非零位均在一个byte中，增添和全一chunk比所有非一位均在一个byte中的dirty byte类型，即0-NI-L与1-NI-L
- 2.扩展LFL和FLF类型
- LFL：原来只有0-NI-L + F + 0-NI-L类型，增添其余三种
- FLF：原来只有0F + 0-NI-L + 0F与1F+0-NI-L+1F 两种，增添其余六种

新算法-SECOMPAX

Origin sequence	0000000 00000000 00111010 00000000 00...0(3*31) 1011010 00000000 00000000 00000000					
WAH encoding	1	0000000 00000000 00111010 00000000	0	0000000 00000000 00000000 00000011	1	1011010 00000000 00000000 00000000
COMPAX encoding	001	01110	00111010 00000011 01011010			
SECOMPAX encoding	001	00111	00111010 00000011 01011010			

Origin sequence	00...0(7*31) 1111111 11001001 11111111 11111111 11...1(3*31)					
WAH encoding	0	0000000 00000000 00000000 00000111	1	1111111 11001001 11111111 11111111	0	0000000 00000000 00000000 00000011
COMPAX encoding	000	00000 00000000 00000000 00000111	1	1111111 11001001 11111111 11111111	011	00000 00000000 00000000 00000011
SECOMPAX encoding	011	01110	00000111 11001001 00000011			

Origin sequence	1111111 11111111 11000111 11111111 00…0(3*31 bits) 0011111 11111111 11111111 11111111					
WAH encoding	1	1111111 11111111 11000111 11111111	0	0000000 00000000 00000000 00000011	1	0011111 11111111 11111111 11111111
COMPAX encoding	1	1111111 11111111 11000111 11111111	000	00000 00000000 00000000 00000011	1	0011111 11111111 11111111 11111111
SECOMPAX encoding	001	10111	11000111 00000011 10011111			

- 公认最好的算法是COMPAX算法
- 在普通情况下，两者差异不大
- 在位图中“1”出现比例占多，出现NI近F的情况下，SECOMPAX算法是COMPAX算法的***3***倍

新算法-SECOMPAX

1			Literal Word
0	11		FLF
0	00	0	0-Fill word
0	00	1	1-Fill word
0	10	0	LFL(First Literal word is almost 0-Fill while second literal word is almost 1-Fill)
0	10	1	LFL(First Literal word is almost 1-Fill while second literal word is almost 0-fill)
0	01	0	LFL(both literal words are 0-fill)
0	01	1	LFL(both literal words are 1-fill)

新算法-SECOMPAX

- 优点：将COMPAX提出的LFL与FLF概念进一步推广，相对COMPAX有更出色的压缩率
- 在0和1局部数量相近时效果更加明显
- 进一步改进：ICX

新算法-ICX(Improved COMPAX)

- 在SECOMPAX基础上进一步改进
- 增添两个新类型：LF与NI2-LF
- LF：一个nearly identical的literal + F，相当于对LFL情况补充
- NI2-LF：带2个dirty byte的literal + F

新算法-ICX(Improved COMPAX)

LF:

Origin sequence	1111111 11111111 11000111 11111111 00...0(3*31 bits) 0011111 11111111 00011111 11111111					
WAH encoding	1	1111111 11111111 11000111 11111111	0	0000000 00000000 00000000 00000011	1	0011111 11111111 00011111 11111111
COMPAX encoding	1	1111111 11111111 11000111 11111111	000	00000 00000000 00000000 00000011	1	0011111 11111111 00011111 11111111
ICX encoding	00001	101	11000111 00000000 00000011	1	0011111 11111111 00011111 11111111	

NI2-LF：

Origin sequence	1111111 11001111 11000111 11111111 11...1(11*31 bits)			
WAH encoding	1	1111111 11001111 11000111 11111111	0	1000000 00000000 00000000 00001011
COMPAX encoding	1	1111111 11001111 11000111 11111111	011	00000 00000000 00000000 00001011
ICX encoding	0001	1011	11000111 11001111 10001011	

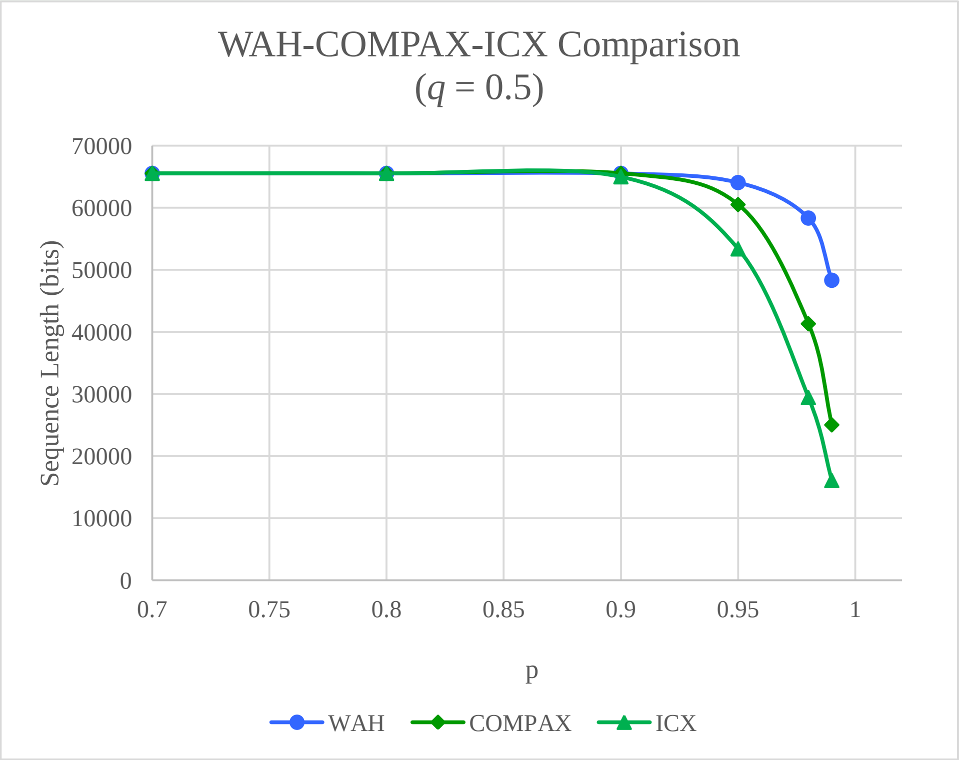
新算法-ICX(Improved COMPAX)

- 码本：

1	Literal Word
0 11	FLF
0 10 0	LFL(First Literal word is almost 0-Fill while second literal word is almost 1-Fill)
0 10 1	LFL(First Literal word is almost 1-Fill while second literal word is almost 0-fill)
0 01 0	LFL(both literal words are almost 0-fill)
0 01 1	LFL(both literal words are almost 1-fill)
0 00 1	NI2-LF(Literal word has two dirty bytes)
0 00 0 1	LF (Literal word is almost 0-fill/1-fill)
0 00 0 0	Fill Word

新算法-ICX(Improved COMPAX)

- 数据集:
- 概率生成01比特串
- p:
- q:



新算法-ICX(Improved COMPAX)

- 优点：在COMPAX&SECOMPAX基础上继续扩展编码方式
- 在0/1局部数量相近（在同一数量级）且分布不完全规律时效果明显

新算法-MASC(MAximized Stride with Carrier)

- 受PLWAH以及排序后位图实际排布启发。。。
- 改变编码方式，不再以chunk为单位，转而寻求最大编码长度，注重连续的0/1比特（与游程编码类似）
- 保留0-fill和1-fill概念，但是counter进行变动，能将非整数chunk的连续0/1也编码进来
- 对0-fill 增加carrier，最多可携带连续30个1.

新算法-MASC-原理介绍

0 0 000000 00000000 00000000 001 01110

1 0 000000 00000000 00000000 001 00101

0 1 00100 0 00000000 00000000 010 11001

0 0 000000 00000000 00000000 001 01110

0000000000 0000000000 0000000000 0

0000000000 0001111000 0000000000 0

1 0 00000 0 00000000 00000000 00000001

0 0000000 00000011 11111111 11111111

0 1111111 11111111 11110000 00000000

1 0 00000 0 00000000 00000000 00000010

0 0000000 00000011 11000000 00000000

1 0 00000 0 00000000 00000000 00000001

Type = 0, Chunk = 1, Additional = 0

Type = 0, Chunk = 1, Additional = 13

Type = 1, Chunk = 0, Additional = 18

Type = 1, Chunk = 0, Additional = 37

Type = 0, Chunk = 0, Additional = 12

Type = 1, Chunk = 0, Additional = 4

Type = 0, Chunk = 0, Additional = 14

Type = 0, Chunk = 2, Additional = 12

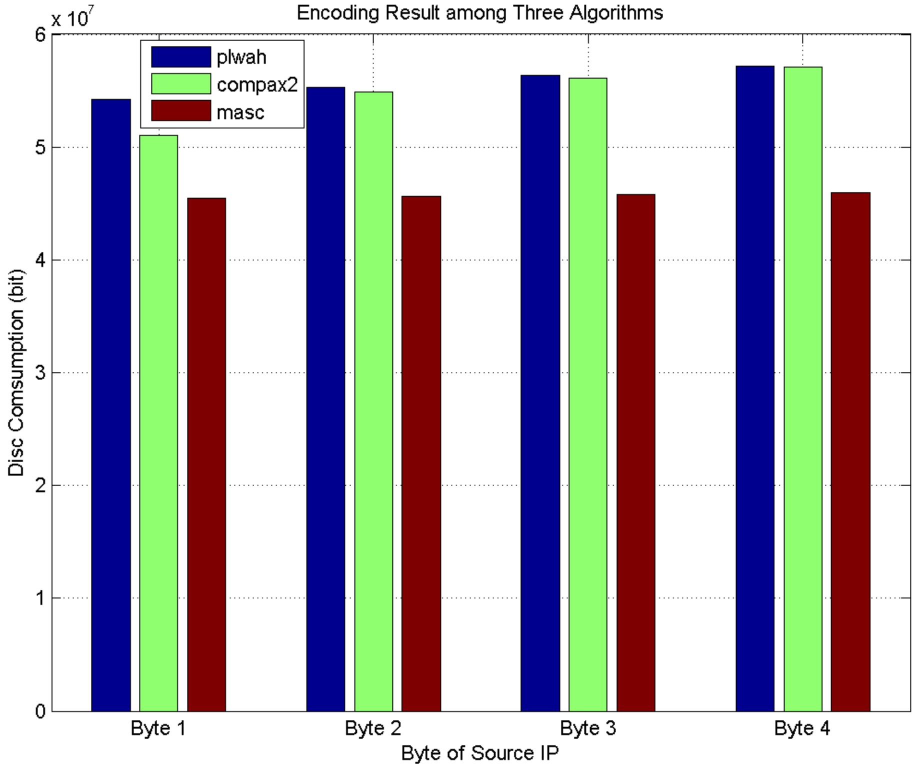
Type = 0, Chunk = 1, Additional = 14

Type = 0, Chunk = 2, Additional = 25

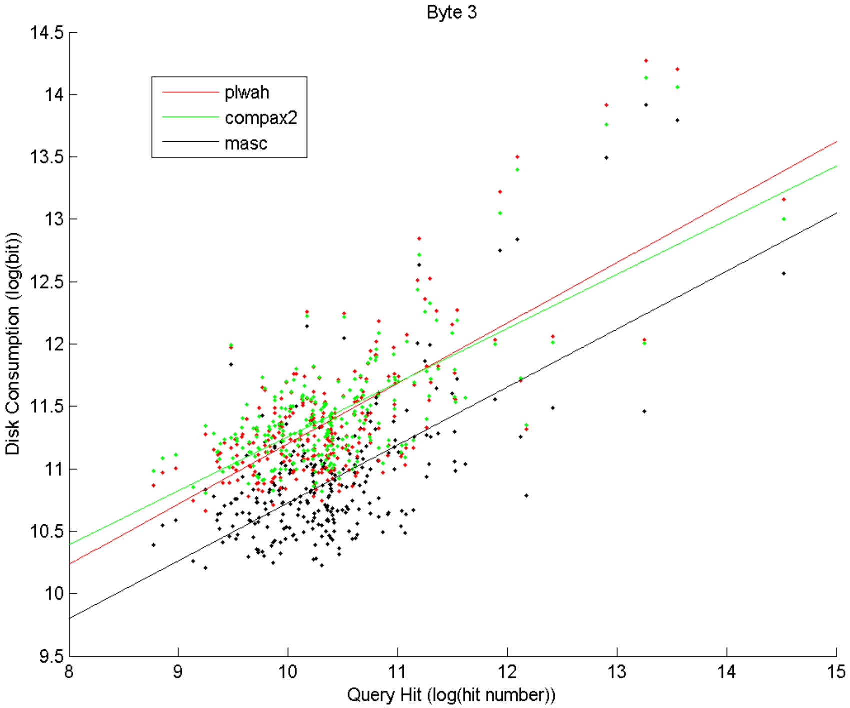
Type = 1, Chunk = 1, Additional = 6

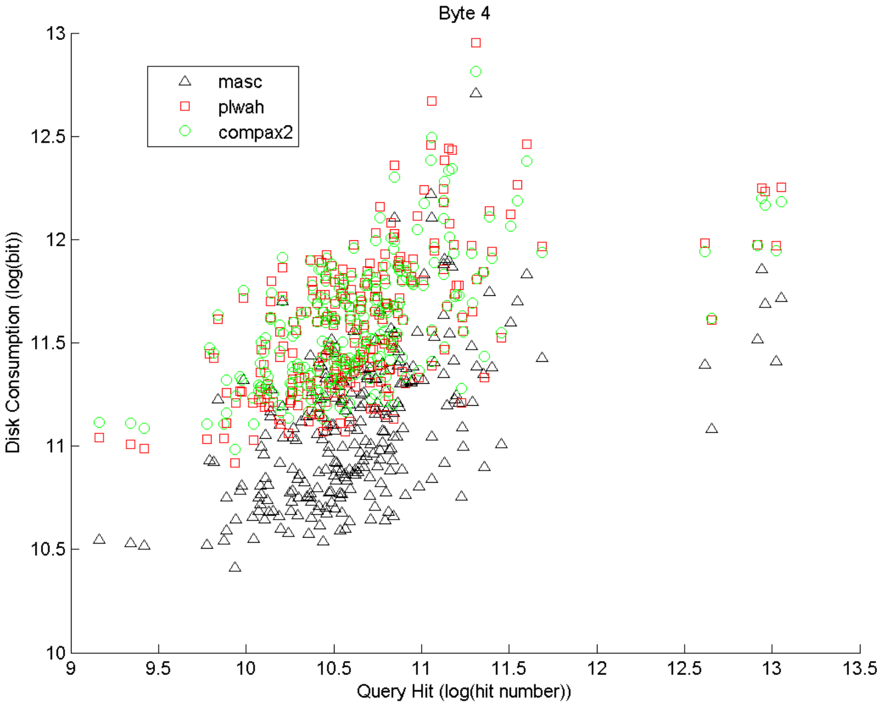
新算法-MASC-实验评估

- 性能比较：
- 18.07%优于PLWAH
- 16.59%优于COMPAX2
- 数据集：CAIDA-2013



Source IP 4字节图

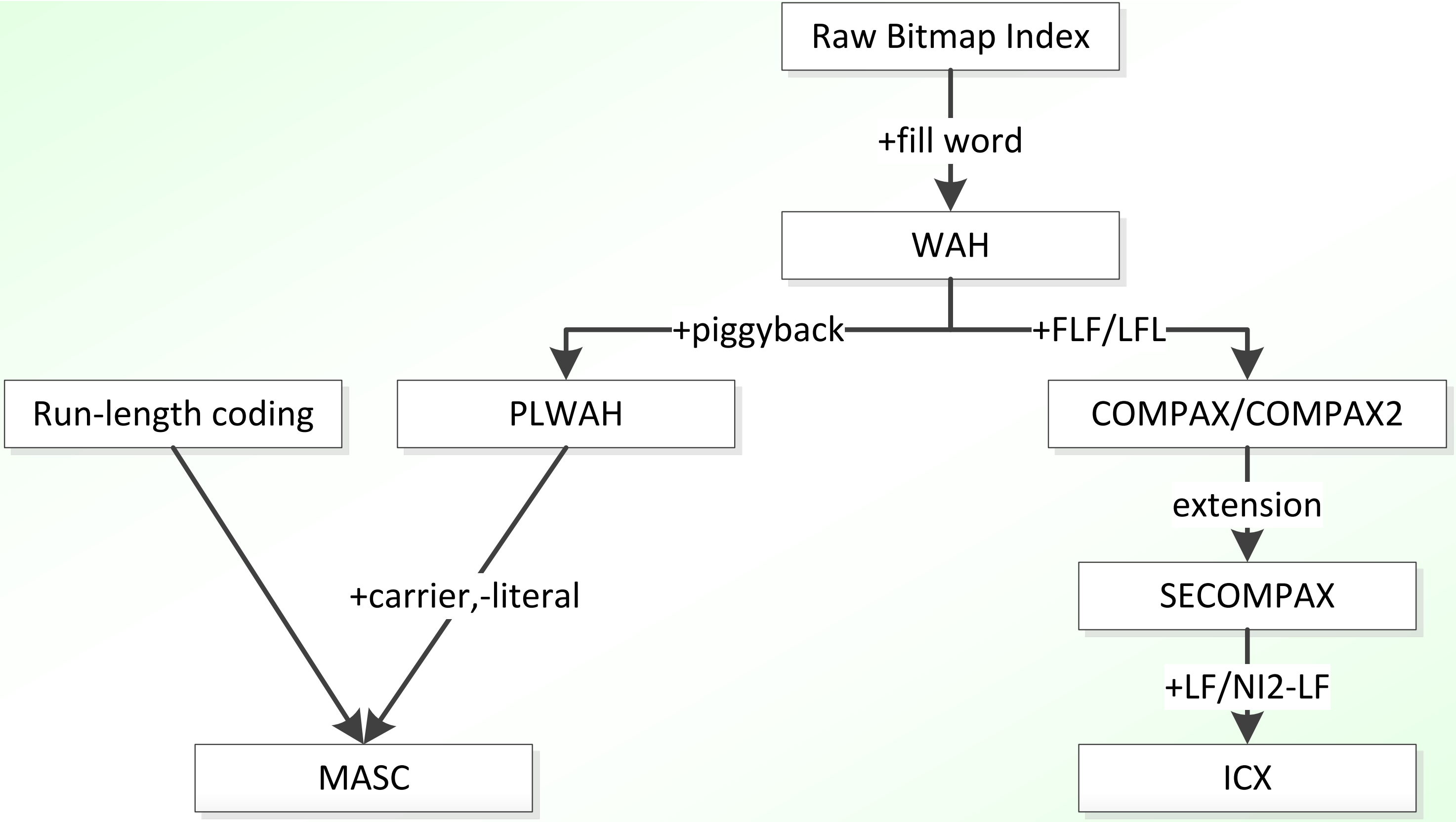




新算法-MASC-创新点

- 优点：专注于对于连续0、1比特的压缩，优化并最终去掉了literal的概念，码本简洁且压缩效果提升明显
- 改进：增加查询表以弥补查询速度可能的缺陷

算法之间关联-roadmap



经典文献

- [1] Jeffrey Dean. "Challenges in building large-scale information retrieval systems: invited talk." In Proceedings of the Second ACM International Conference on Web Search and Data Mining, pp. 1-1. ACM, 2009.
- [2] Wu, Ming-Chuan, Alejandro P. Buchmann, and P. Larson. Encoded Bitmap Indexes and Their Use for Data Warehouse Optimization. Shaker, 2001.
- [3] Wu, Kesheng, Ekow J. Otoo, and Arie Shoshani. "Optimizing bitmap indices with efficient compression." ACM Transactions on Database Systems (TODS) 31, no. 1 (2006): 1-38.
- [4] F. Deli`ege and T. B. Pedersen. Position list word aligned hybrid: optimizing space and performance for compressed bitmaps. Proc. of the 13th Int. Conf. on Extending Database Technology, EDBT ‘10, 2010.
- [5] Fusco, Francesco, Michail Vlachos, and Marc Ph Stoecklin. "Real-time creation of bitmap indexes on streaming network data." The VLDB Journal-The International Journal on Very Large Data Bases 21, no. 3 (2012): 287-307.
- [6] Fusco, Francesco, Michail Vlachos, and Xenofontas Dimitropoulos. "RasterZip: compressing network monitoring data with support for partial decompression." Proceedings of the 2012 ACM conference on Internet measurement conference. ACM, 2012.
- [7] Fusco, F., Dimitropoulos, X., Vlachos, M., & Deri, L. (2012). pcapIndex: an index for network packet traces with legacy compatibility. ACM SIGCOMM Computer Communication Review, 42(1), 47-53.

研究小结

- 论文投稿
  - SECOMPAX: A bitmap index compression algorithm for Internet traffic archival
  - ICX: a new bitmap index compression scheme for archival Internet traffic
  - MASC: a bitmap index encoding algorithm for fast data retrieval
- 专利申请
  - 最大步进携带位图索引编码的方法
  - 一种新的位图索引编码的方法
  - 一种位图索引编码机制
  - 基于倒排列表网流索引压缩查询机制

谢谢！