Web Graph compression

Outline

- Web Graph definition;
- Literature review:
  - Connective Server;
  - Link Data;
  - WebGraph Framework;
  - Extra literature.
- Compression results comparison;
- Final considerations.

Web Graph

A web graph relative to a set of URLs is a directed graph having those URLs as the set of nodes. An arc $u \rightarrow v$ is identified for each hyperlink from a URL *u* towards a URL *v*.

– URLs that do not appear either as sources or in more than *T* (4) pages are ignored;

–The URLs are normalized by converting hostnames to lower case, cannonicalizes port number, re-introducing them where they need, and adding a trailing slash to all URLs that do not have it.

Main features of Web Graphs

Locality: usually most of the hyperlinks are local, i.e, they point to other URLs on the same host. The literature reports that on average 80% of the hyperlinks are local.

Consecutivity: links within a same page are likely to be consecutive respecting to the lexicographic order.

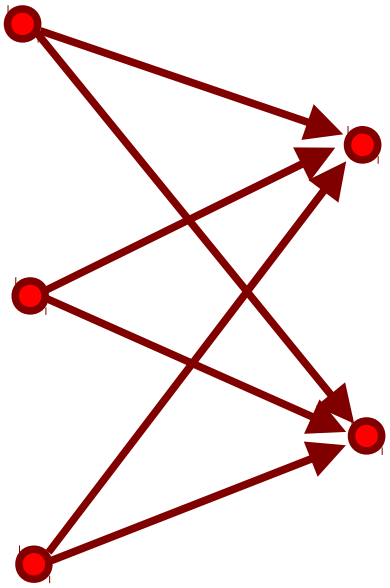
Ex: http://my.sample.com/whitepaper/nodeA.html

http://my.sample.com/whitepaper/nodeB.html

Main features of WebGraphs

Similarity: Pages on the same host tend to have many hyperlinks pointing to the same pages.

Consecutivity is the dual distance-one similarity.



Level of Compression

- Node Compression

- Adjacency Lists:

  Forward and backward representations.

  Many graphs computations, search algorithms (e.g. HITS) and queries require access to backlinks.

Literature

Connectivity Server (1998) – *Digital Systems Reseach Center and Stanford University* – K. Bharat, A. Broder, M. Henzinger, P. Kumar, S. Venkatasubramanian;

Link Database (2001) - *Compaq Systems Research Center* – K. Randall, R. Stata, R. Wickremesinghe, J. Wiener;

WebGraph Framework (2002) – *Universita degli Studi di Milano* – P. Boldi, S. Vigna.

S-Node Representation (2002), *Database Group and Stanford University* – Sriram Raghvan and Hector Garcia-Molina

Connectivity Server

- Tool for web graphs visualisation, analysis (connectivity, ranking pages) and URLs compression.
- Used by Alta Vista;
- Links represented by an outgoing and an incoming adjacency lists;

- Composed of:

URL Database: URL, fingerprint, URL-id;

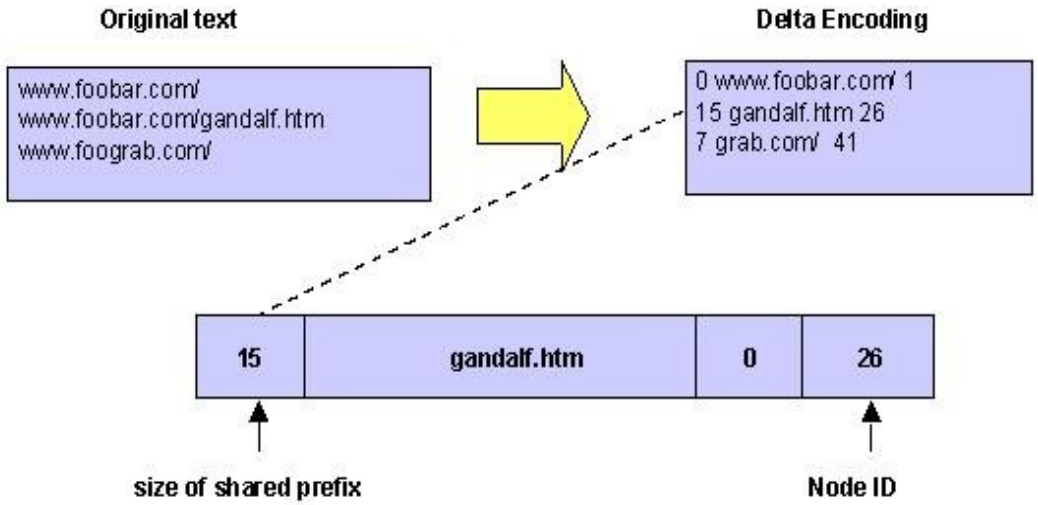
Host Database: group of URLs based on the hostname portion;

Link Database: URL, outlinks, inlinks.

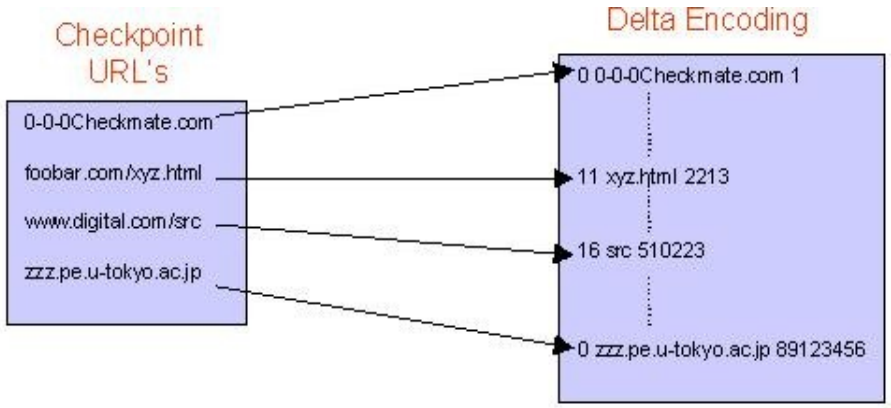
Connectivity Server: URL compression

URLs are sorted lexicographically and stored as a delta encoded entry (70% reduction).

URLs delta encoding



Indexing the delta enconding



Link1: first version of Link Database

No adj. lists compression: simple representation of outgoing and incoming adjacency lists of links.

Avg. inlink size: 34 bits

Avg. outlink size: 24 bits

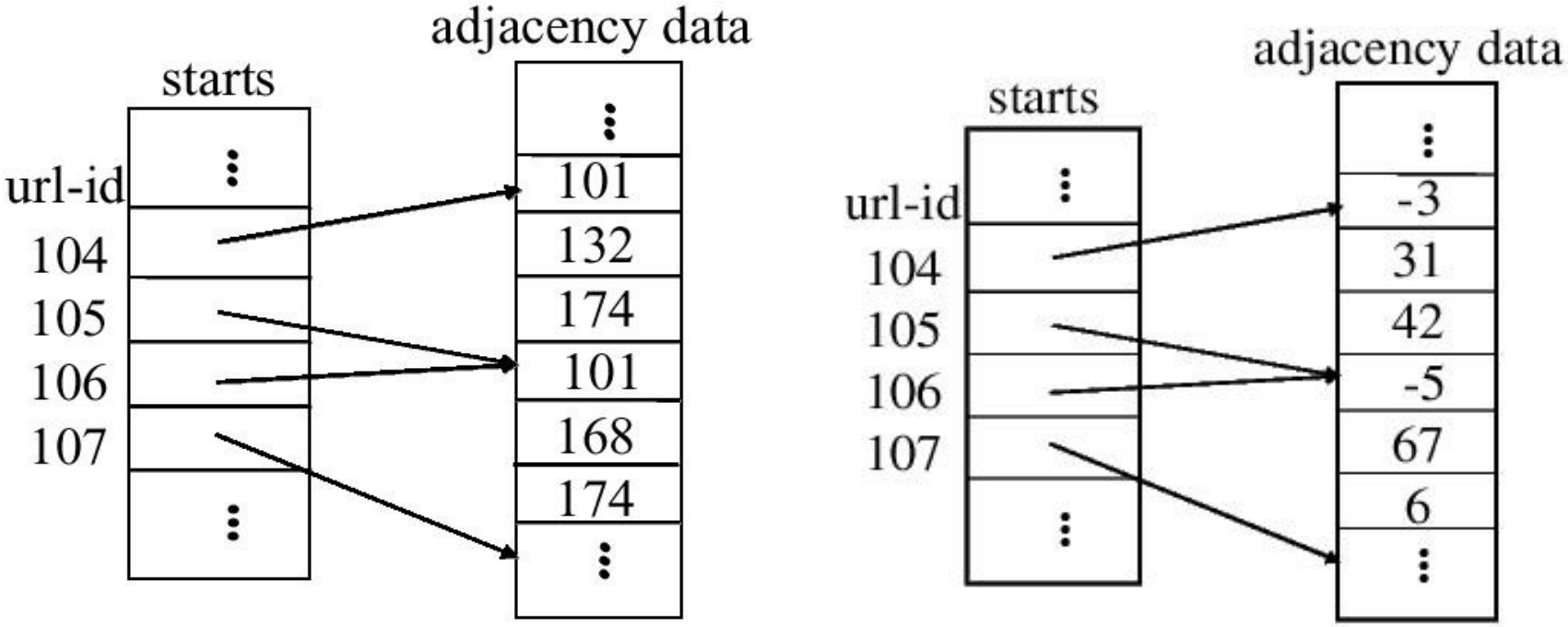
Link2: second version of Link Database

Single list compression and starts compression

Avg. inlink size: 8.9 bits

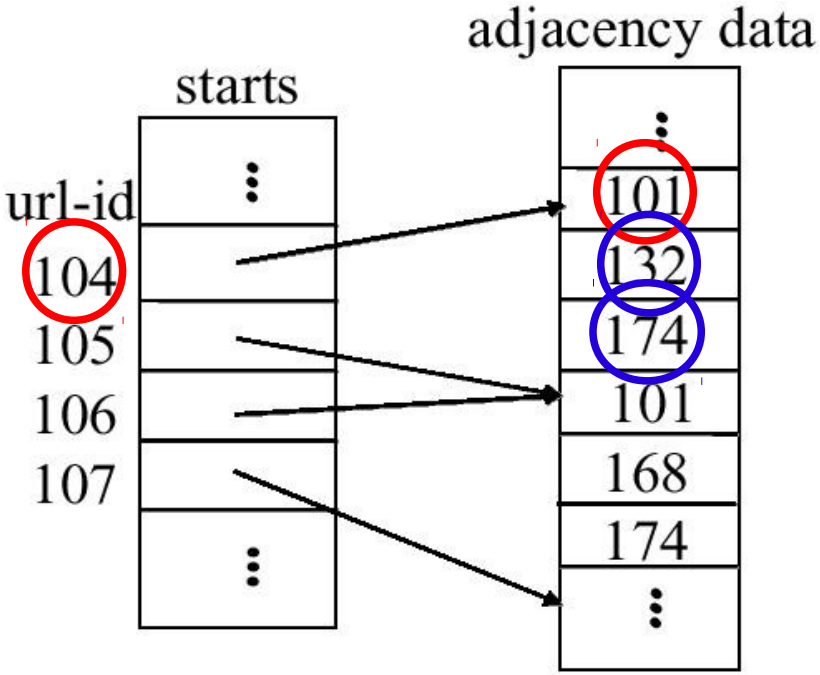
Avg. outlink size: 11.03 bits

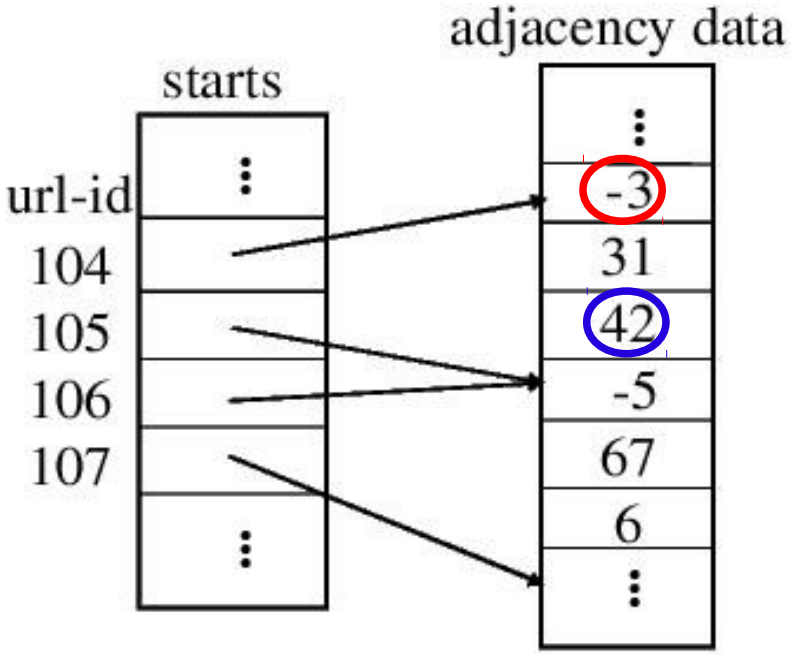
Delta Encoding of the Adjacency Lists



Each array element is 32 bits long.

Delta Encoding of the Adjacency Lists





-3 = 101-104 (first item)

42 = 174-132 (other items)

Nybble Code

- The low-order bit of each nybble indicates whether or not there are more nybbles in the string
- The least-significant data bit encodes the sign.
- The remaining bits provide an unsigned number

28 = 0111 1000

-28 = 1111 0010

Starts array compression

- The URLs are divided into three partitions based on their degree;
- Elements of *starts* are indices to nybbles;
- The literature reports that 74% of the entries are in the low-degree partition.

Starts array compression

Entry range	Partition	# bits
$Z(x) > 254$	High-degree partition	32
$254 \leq Z(x) \leq 24$	medium-degree partition	$(32+P*16)/P$
$Z(x) < 24$	low-degree partition	$(32+P*8)/P$

$Z(x) = max$ (indegree(x), outdegree(x))

P = the number of pages in each block.

Link3: third version of Link Database

Interlist compression with representative list

Avg. inlink size: 5.66 bits

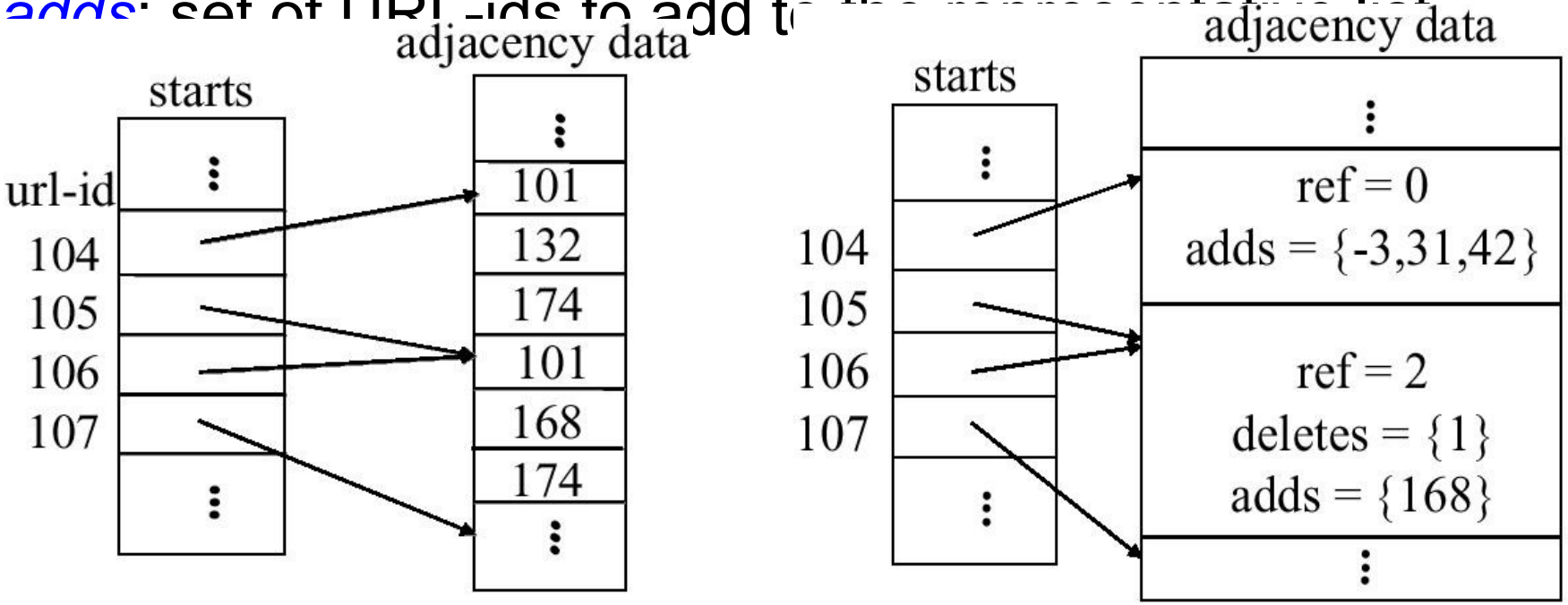
Avg. outlink size: 5.61 bits

Interlist Compression

ref : relative index of the representative adjacency list;

deletes: set of URL-ids to delete from the representative list;

adds: set of URL-ids to add to the representative list



LimitSelect-K-L: chooses the best representative adjacency list from among the previus *K* (8) URL-ids' adjacency lists and only allows chains of fewer than *L* (4) hops.

-codes (WebGraph Framework)

Interlist compression with representative list

Avg. inlink size: 3.08 bits

Avg. outlink size: 2.89 bits

Compressing Gaps

Uncompressed adjacency list

Node	Outdegree	Successors
...	...	...
15	11	13, 15, 16, 17, 18, 19, 23, 24, 203, 315, 1034
16	10	15, 16, 17, 22, 23, 24, 315, 316, 317, 3041
17	0	
18	5	13, 15, 16, 17, 50
...	...	...

Adjacency list with compressed gaps.

Node	Outdegree	Successors
...	...	...
15	11	3, 1, 0, 0, 0, 0, 3, 0, 178, 111, 718
16	10	1, 0, 0, 4, 0, 0, 290, 0, 0, 2723
17	0	
18	5	9, 1, 0, 0, 32
...	...	...

Successor list $S(x) = \{s_1\text{-}x, s_2\text{-}s_1\text{-}1, ..., s_k\text{-}s_{k-1}\text{-}1\}$

For negative entries: $\nu(x) = \begin{cases} 2x & \text{if } x \geq 0 \\ 2|x| - 1 & \text{if } x < 0 \end{cases}$

Using copy lists

Uncompressed adjacency list

Node	Outdegree	Successors
...	...	...
15	11	13, 15, 16, 17, 18, 19, 23, 24, 203, 315, 1034
16	10	15, 16, 17, 22, 23, 24, 315, 316, 317, 3041
17	0	
18	5	13, 15, 16, 17, 50
...	...	...

Adjacency list with copy lists.

Node	Outd.	Ref.	Copy list	Extra nodes
...	...	...	...	...
15	11	0		13, 15, 16, 17, 18, 19, 23, 24, 203, 315, 1034
16	10	1	01110011010	22, 316, 317, 3041
17	0			
18	5	3	11110000000	50
...	...	...	...	...

- Each bit on the copy list informs whether the corresponding successor of *y* is also a successor of *x*;
- The reference list index *ref.* is chosen as the value between 0 and *W* (window size) that gives the best compression.

Using copy blocks

Adjacency list with copy lists.

Node	Outd.	Ref.	Copy list	Extra nodes
...	...	...	...	...
15	11	0		13, 15, 16, 17, 18, 19, 23, 24, 203, 315, 1034
16	10	1	01110011010	22, 316, 317, 3041
17	0			
18	5	3	11110000000	50
...	...	...	...	...

Adjacency list with copy blocks.

Node	Outd.	Ref.	# blocks	Copy blocks	Extra nodes
...	...	...	...	...	...
15	11	0			13, 15, 16, 17, 18, 19, 23, 24 203, 315, 1034
16	10	1	7	0, 0, 2, 1, 1, 0, 0	22, 316, 317, 3041
17	0				
18	5	3	1	4	50
...	...	...	...	...	...

- The last block is omitted;
- The first copy block is 0 if the copy list starts with 0;
- The length is decremented by one for all blocks except the first one.

Compressing intervals

Adjacency list with copy lists.

Node	Outd.	Ref.	Copy list	Extra nodes
...	...	...	...	...
15	11	0		13, 15, 16, 17, 18, 19, 23, 24, 203, 315, 1034
16	10	1	01110011010	22, 316, 317, 3041
17	0			
18	5	3	11110000000	50
...	...	...	...	...

Adjacency list with intervals.

Node	Outd.	Ref.	# blocks	Copy blocks	# intervals	Left extremes	Length	Residuals
...	...	...	...	...	...	...	...	...
15	11	0			2	0, 2	3, 0	5, 189, 111, 718
16	10	1	7	0, 0, 2, 1, 1, 0, 0	1	600	0	12, 3018
17	0							
18	5	3	1	4	0			50
...	...	...	...	...	...	...	...	...

- Intervals: represented by their left extreme and lenght;
- Intervals length: are decremented by the threshold L_{min};
- Residuals: compressed using differences.

Compressing intervals

Adjacency list with copy lists.

Node	Outd.	Ref.	Copy list	Extra nodes
...	...	...	...	...
15	11	0		13, 15, 16, 17, 18, 19, 23, 24, 203, 315, 1034
16	10	1	01110011010	22, 316, 317, 3041
17	0			
18	5	3	11110000000	50
...	...	...	...	...

Adjacency list with intervals.

Node	Outd.	Ref.	# blocks	Copy blocks	# intervals	Left extremes	Length	Residuals
...	...	...	...	...	...	...	...	...
15	11	0			2	0, 2	3, 0	5, 189, 111, 718
16	10	1	7	0, 0, 2, 1, 1, 0, 0	1	600	0	12, 3018
17	0							
18	5	3	1	4	0			50
...	...	...	...	...	...	...	...	...

0 = (15-15)*2

600 = (316-16)*2

5 = |13-15|*2-1

3018 = 3041-22-1

50 = ?

Compression comparison

	Inlink	Outlink	RanTime	SeqTime	#Mpage	#Mlink	Database
Huff.	15.2	15.4	98	112	320		WebBase
Link1	34	24	72	13	61	1000	Web Crawler Mercat
Link2	8.9	11.03	109	47	61	1000	Web Crawler Mercat
Link3	5.66	5.61	336	248	61	1000	Web Crawler Mercat
ζ-codes	3.25	2.18		206	18.5	300	.uk domain
s-Node	5.07	5.63	702	298	900		WebBase

Using different computers and compilers.

Extra Literature

Toward compressing Web graphs (2001), *University of Massachusetts, Harvard University*, M. Adler, M. Mitzenmacher – theorical study of Web-graph compression.

Compressing the graph structure of the web (2001), *Polytechnic University (NY)*, T. Suel, J. Yuan – only consider outlinks and stores 14 bits per link.

Representing web graphs (2002), *Stanford University*, S. Raghan, H. Garcia-Molina – propose a new representation for Web graphs that reduce time access.

Conclusions

The compression techniques are specialized for Web Graphs.

The average link size decreases with the increase of the graph.

The average link access time increases with the increase of the graph.

The κkϱζ-codes seems to have the best trade-off between avg. bit size and access time.