Zip-Ada

Recent developments in Zip-Ada – part 2

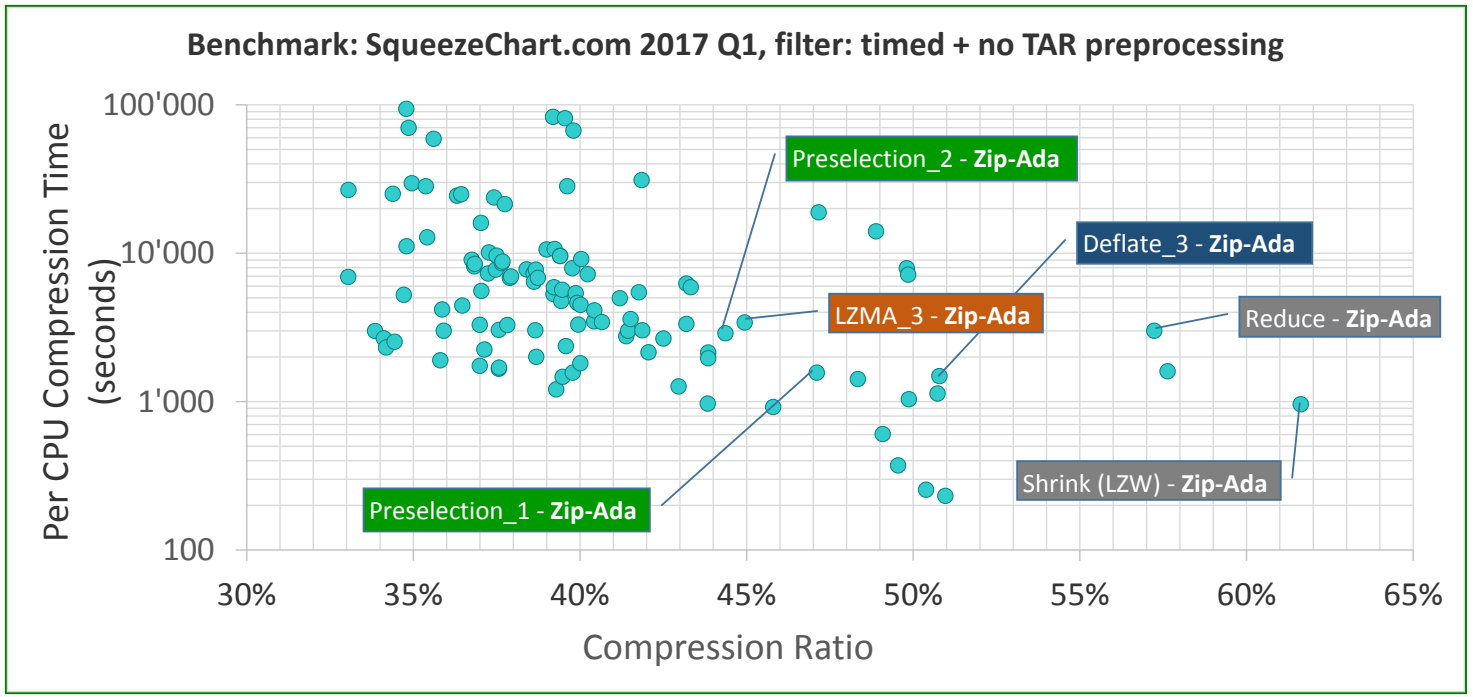
Part 1: Overview; new Deflate compression algorithm

Part 2: New LZMA compression algorithm

Dr Gautier de Montmollin

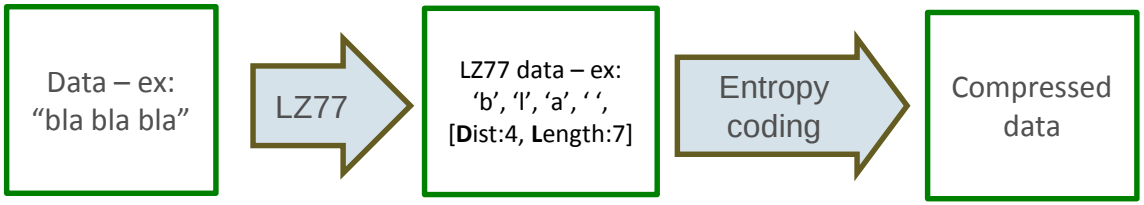
Swiss Ada Event 2017, Rapperswil, September 21, 2017

Zip-Ada compression family (as in v.52)



The LZMA format

- Invented by Igor Pavlov (~1998), shipped the 7-Zip software.
- Included as an “official” Zip compression format by PKWare (ref. #4).
- Combines **LZ77** (font-end) and **range encoding** (entropy back-end).



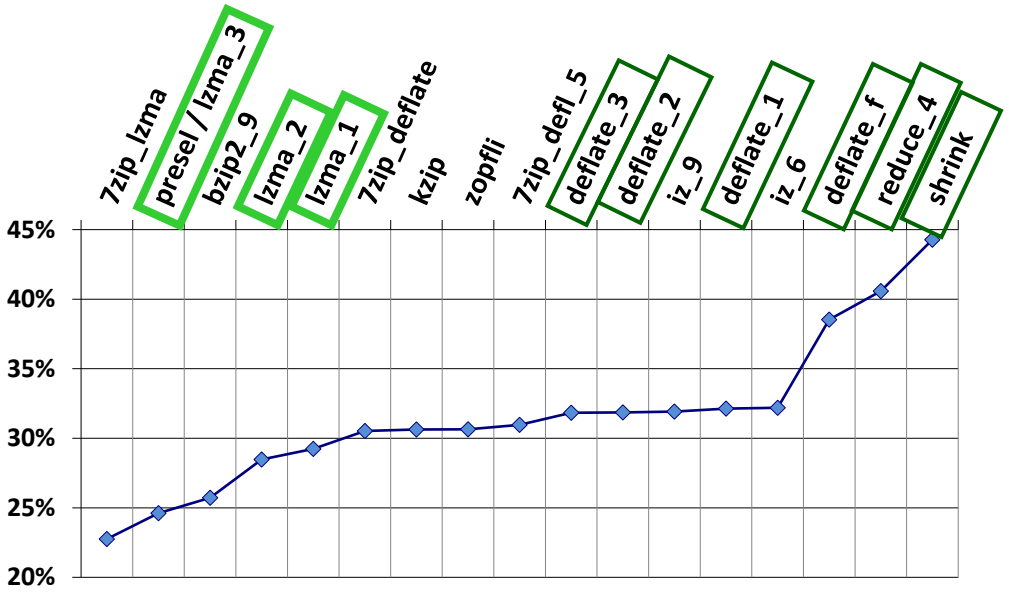
- **Adaptive** back-end (compression model is adapted with the flow of LZ77 data).
- Single compression block.
- First serious documentation, specification and reference decoder: 2013. NB: range encoding (ref. #3) was covered by an IBM patent from 1978 to 2003.

The LZMA format – Zip-Ada implementation – 2016

- Common generic **LZ77** (can be used standalone). *Info-Zip/zlib's* LZ77 is used for our LZMA_1 .. LZMA_2 (quick). A *7-Zip's* LZ77 is used for our LZMA_3.
- **No LZ77 at all** is used for our LZMA_0 – best on certain data!
- Straightforward back-end encoder (300 LOC !) for LZMA_1; more sophisticated encoder for LZMA_2 .. LZMA_3.

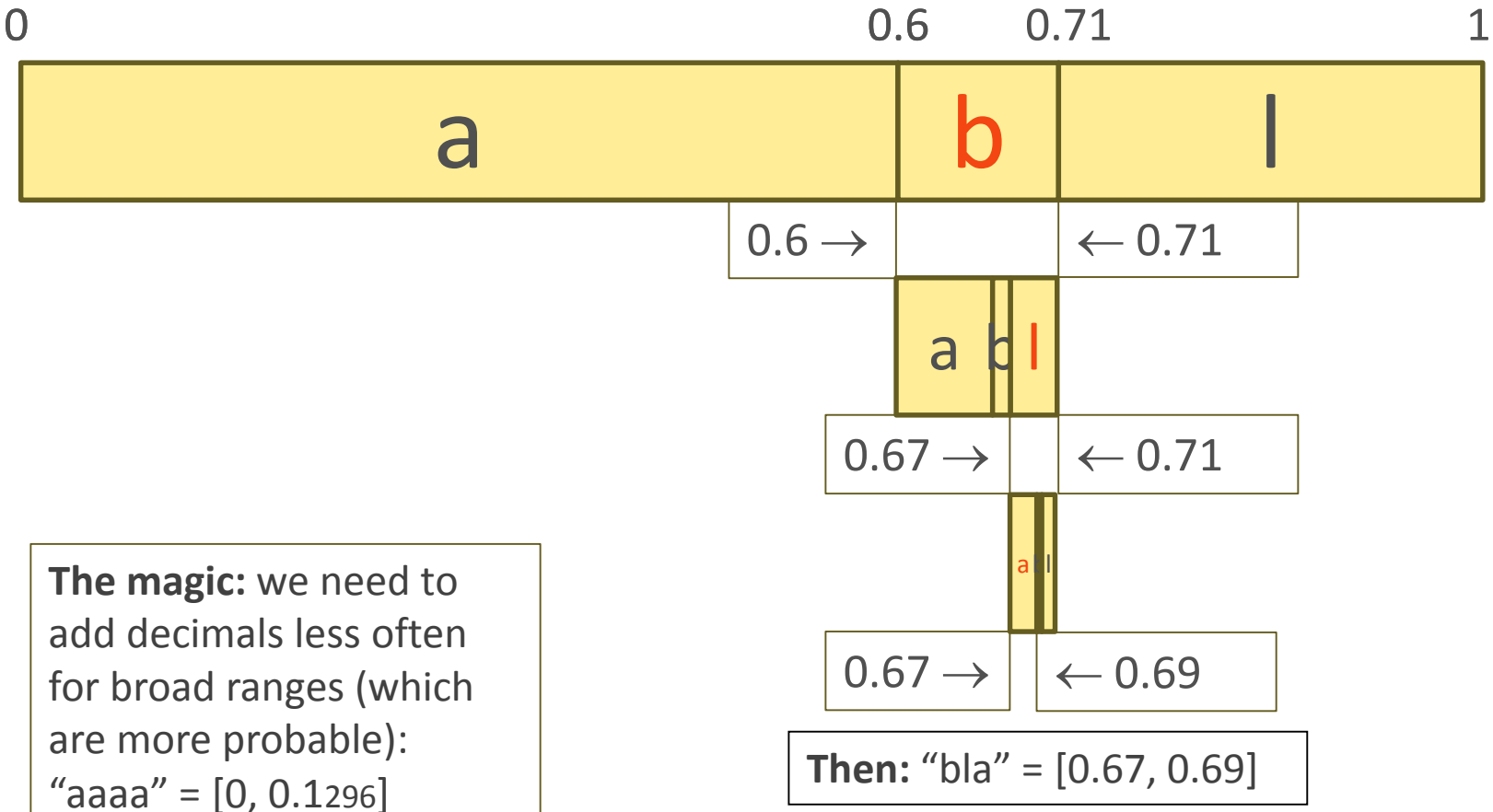
Silesia corpus			
Date / Size	% compr	Name	Deflate bench
48'240'494	22.8%	7zip_lzma	-25.44%
52'169'187	24.6%	presel / lzma_3	-19.37%
54'509'539	25.7%	bzip2_9	-15.75%
60'346'016	28.5%	lzma_2	-6.73%
61'970'916	29.2%	lzma_1	-4.22%
64'698'142	30.5%	**7zip_deflate**	**0.00%**
64'921'533	30.6%	kzip	+0.35%
64'949'384	30.6%	zopfli	+0.39%
65'636'076	31.0%	7zip_defl_5	+1.45%
67'462'614	31.8%	deflate_3	+4.27%
67'506'579	31.9%	deflate_2	+4.34%
67'634'472	31.9%	iz_9	+4.54%
68'110'939	32.1%	deflate_1	+5.27%
68'230'447	32.2%	iz_6	+5.46%
81'667'070	38.5%	deflate_f	+26.23%
85'991'264	40.6%	reduce_4	+32.91%
93'826'501	44.3%	shrink	+45.02%
211'938'580	100.0%	original data	

Green = Zip-Ada



Range encoding

Example with a restricted alphabet: a, b, l. Widths are proportional to average frequencies in the English language. We want to encode "bla".

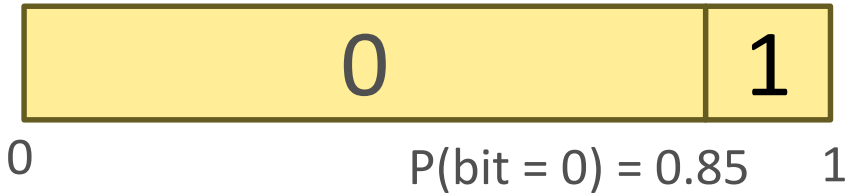


The magic: we need to add decimals less often for broad ranges (which are more probable): "aaaa" = [0, 0.1296]

Then: "bla" = [0.67, 0.69]

Range encoding *in LZMA*

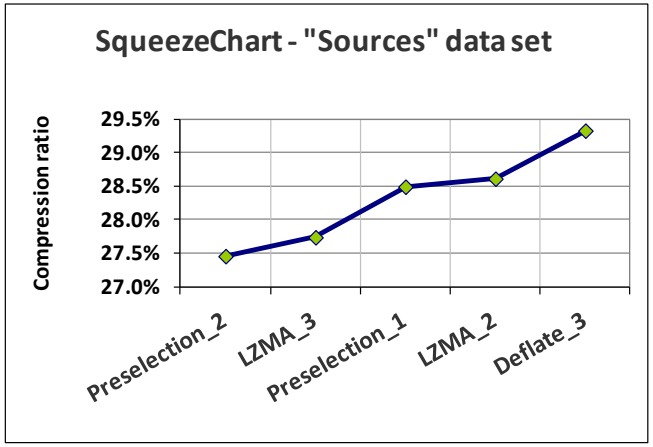
- Only 0's and 1's. The interval may be *expanded* after a bit output.

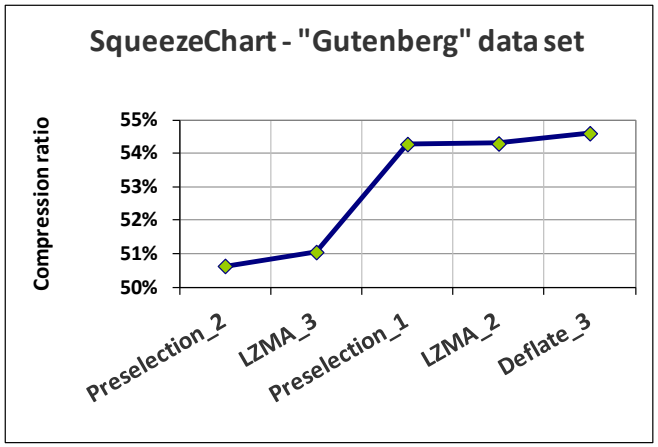


- Many **contextual probability sets** used. Here, for literals:
  - **Previous bits in a byte** (end effect: 256 subintervals, one prob. for each byte value)
  - **Value of previous byte** (Markov predictor)
  - **Position of the byte** modulo up to 16 (good for structured data or Unicode text)
  - ⇒ each bit uses *one* of **8,388,608** probabilities (max configuration) !
- Default, neutral probability is 0.5, then adapted with a factor (~1.03) on each output.
- Max probability ~0.985: in the best case, compressed output is ~**0.03 bit per uncompressed bit** – that, only the for "MA" part, it is on top of the "LZ" compression !

"Preselection" algorithm-picking method

- LZMA needs some warm-up phase to have its huge probability model adapted to data – it works better on large, homogeneous data.
- Indeed, Deflate usually beats LZMA on data smaller than 9000 bytes (empirical threshold).
-  idea: select Deflate for small data, LZMA for large ones.





- Special cases: see Zip.Compress' body (LZ77 choices, 225 LZMA configs, …)

References

1. **Zip-Ada** web site **http://unzip-ada.sf.net/**

2. **AZip** web site **http://azip.sf.net/** (AZip is GUI archive manager using Zip-Ada)

3. **Squeeze Chart:** large and varied corpus: 5 GB; 21,532 files; web site: **http://www.squeezechart.com/**

4. **Range encoding: an algorithm for removing redundancy from a digitized message**, G. N. N. Martin, Video & Data Recording Conference, Southampton, UK, July 24-27, 1979.

5. **Zip file format specification:** **https://support.pkware.com/display/PKZIP/APPNOTE**