Portfolio: "Textkompressionsalgorithmen in Datenbanken - Eine Forschungsfeldübersicht"

Related Work:

<https://web.archive.org/web/20200620124923/https://www.percona.com/blog/2016/03/09/evaluating-database-compression-methods/> + <https://web.archive.org/web/20200229161007/https://www.percona.com/blog/2016/04/13/evaluating-database-compression-methods-update/>

<https://www.rroij.com/open-access/a-short-survey-of-data-compression-techniques-for-column-oriented-databases-43-46.php?aid=38816>

<https://openproceedings.org/2017/conf/edbt/paper-146.pdf>

<https://arxiv.org/pdf/2506.10000>

Grundlagen:

* Verlustfreie vs. Verlustbehaftete Kompression
* Prinzipien: Entropie, Redundanz, Kodierung
* Überblick über gängige Kompressionsalgorithmen
  + LZ-Familie
  + Huffman
  + RLE, Snappy,…
* Datenbanken: relationale, spaltenorientierte, zeilenorientierten, NoSQL/Key-Value Stores

Übersicht über aktuelle Ansätze:

* FSST: <https://dl.acm.org/doi/10.14778/3407790.3407851>
* Dictionary-based Order-preserving String Compression for Main Memory Column Stores: <https://dl.acm.org/doi/10.1145/1559845.1559877>
* Fast & Strong: The Case of Compressed String Dictionaries on Modern CPUs: <https://dl.acm.org/doi/10.1145/3329785.3329924>
* ALPHAZIP : NEURAL NETWORK-ENHANCED LOSSLESS TEXT COMPRESSION: <https://arxiv.org/abs/2409.15046>

Optimierungen genannter Ansätze:

* GSST: Parallel string decompression at 191 GB/s on GPU: <https://dl.acm.org/doi/10.1145/3719330.3721228>
* Auf der Suche nach mehr Quellen

Evaluation:

* Compression Rate
* Encoding/decoding Speed
* Komplexe/einfache Implementierung
* Evtl auch: Energieeffizienz: <https://link.springer.com/article/10.1007/s00607-025-01469-0>