Compressing Relations and Indexes

Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft

Speaker : Jonathan Goldstein

Microsoft Research

OVERVIEW

- Motivation
- Compressing relations with fixed size atributes
  - Page level compression
  - File level compression
  - Experimental results
- Compressing indexes
  - Compressing index pages
  - Experimental results
- Related and Future Work
- Conclusions

MOTIVATION

Benefits:

- Improved storage requirements.
- Improved information throughput from disks
- Potentially improved buffer utilization
- Improved fanout for indexing structures.

COMPRESSING RELATIONS(Page Level)

Frame of reference (page level) compression:

Given the points:

{(511,1001),(517,1007),(514,1031)}

subtract (511, 1001) from all tuples and the set in binary becomes:

{(000, 00000), (110, 00110), (011, 11110)}

Only 3 and 5 bits are needed to store these tuples provided that we know (511,1007) was subtracted from them!

COMPRESSING RELATIONS(Page Level)

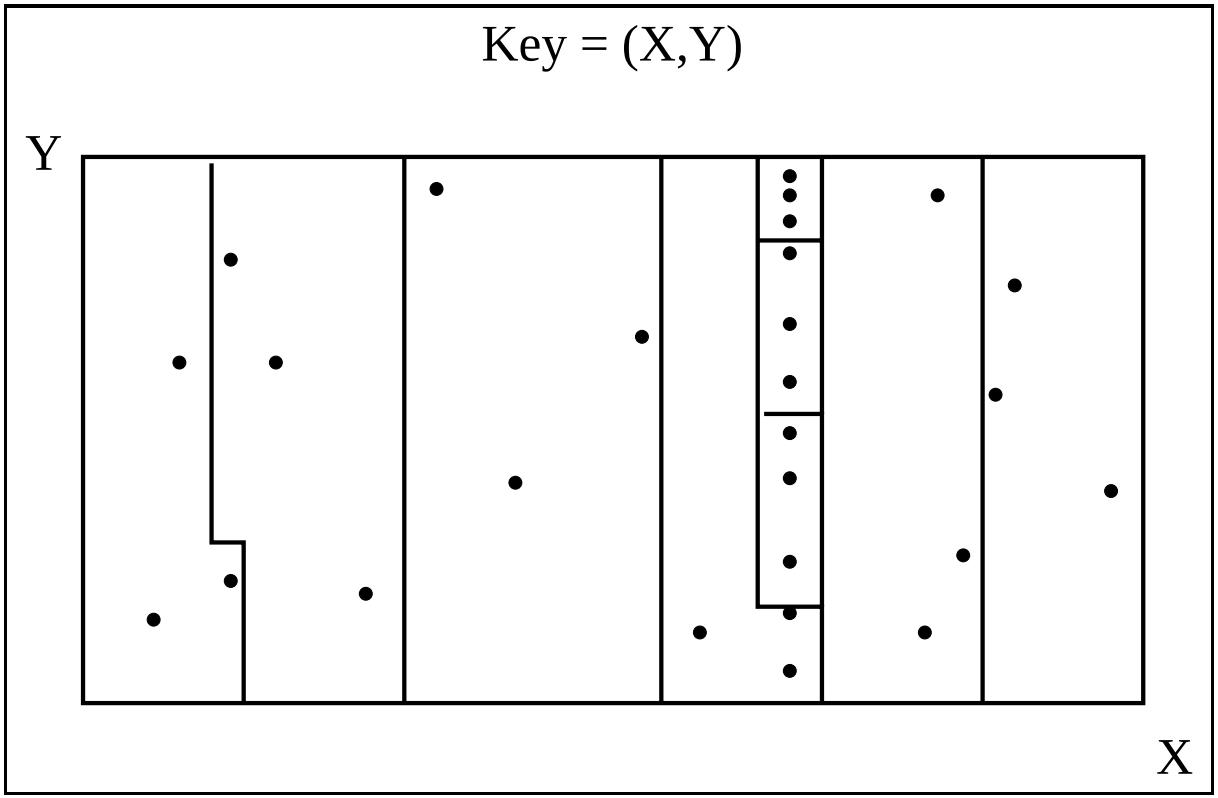
This compression technique can be applied to fixed length data such that:

- Decompression can be done on a per field per tuple granularity
  - Pages can stay in memory compressed.

COMPRESSING RELATIONS(File Level)

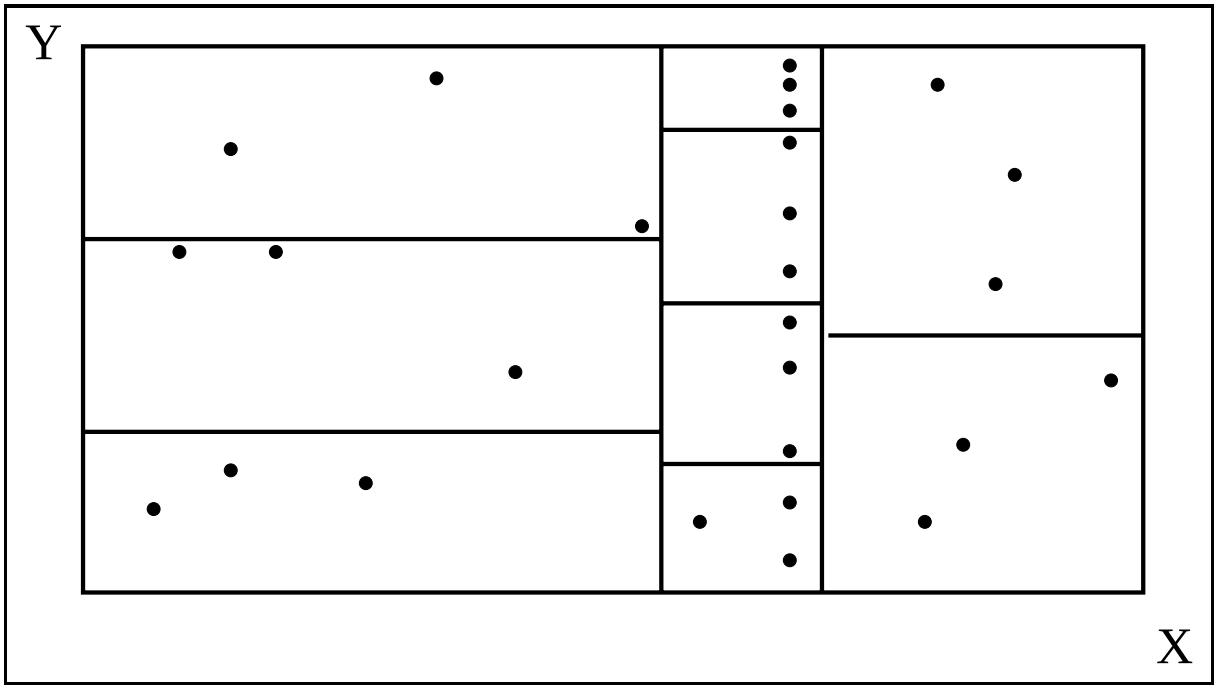
How do we order tuples in a relation to improve compression?

B-Tree tuple grouping on a 2 integer attribute relation:



COMPRESSING RELATIONS(File Level)

R-Tree tuple grouping on a 2 integer attribute relation:



COMPRESSING RELATIONS(File Level)

Thus we can achieve efficient inter-page tuple placement by:

- Using an existing sort order (single or multidimensional).
- Applying B-Tree sort order when there is no pre-existing sort order.

COMPRESSING RELATIONS(Experiments)

Our page level compression implementation had the following properties:

- There was no slot array.
- The page size was 4K.

COMPRESSING RELATIONS(Experiments)

Compression achieved on a catalog company's sales data:

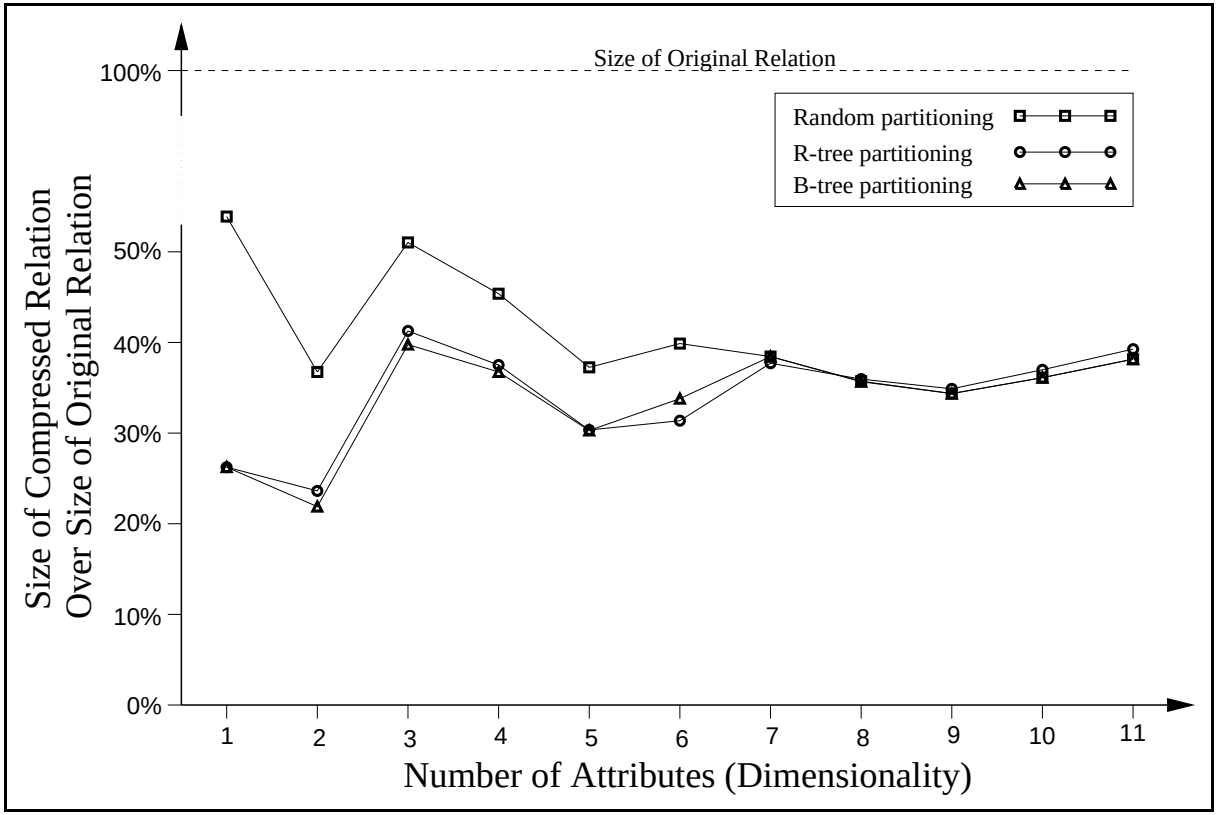


Figure 1: Varying tuple grouping

COMPRESSING RELATIONS(Experiments)

Compression achieved on synthetic data sets using R-Tree ordering:

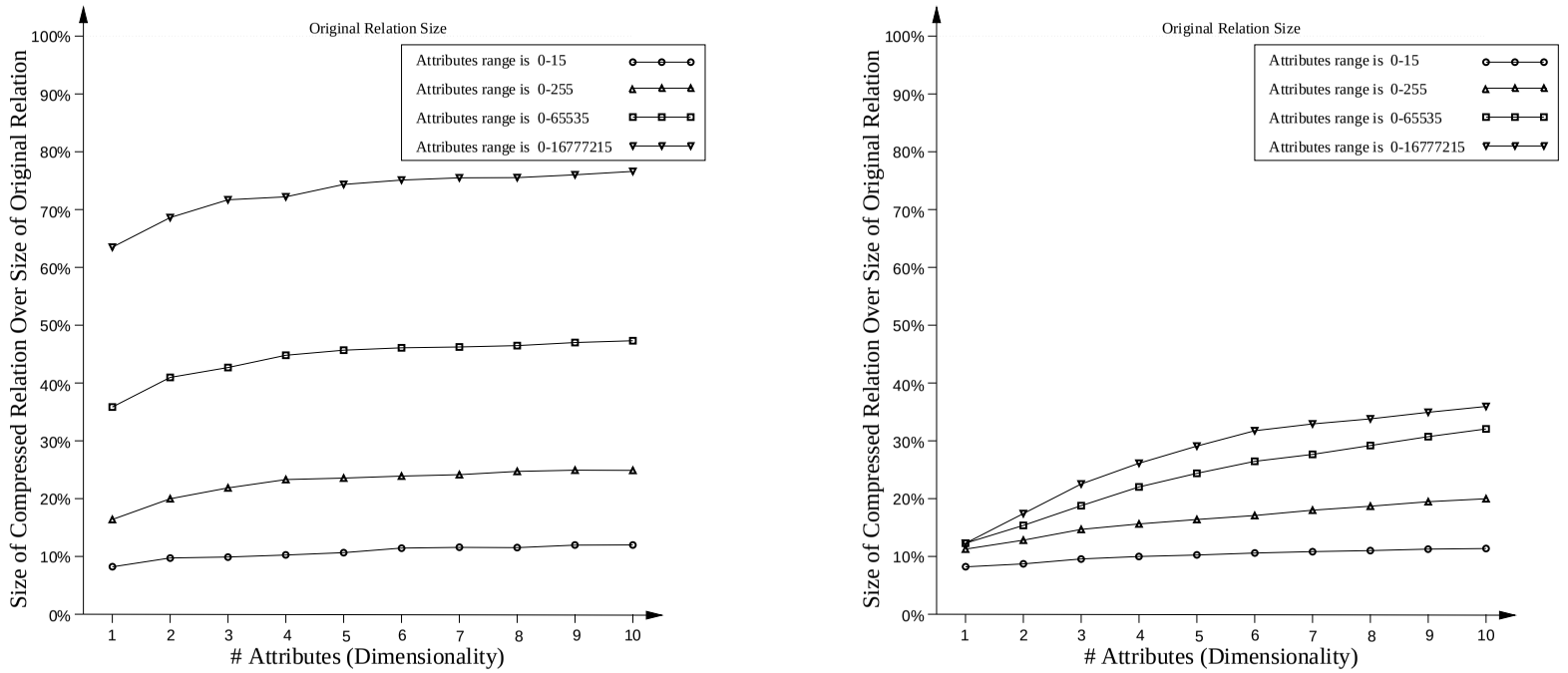


Figure 2: Varying distribution: Uniform(left) and Exponential(right)

COMPRESSING RELATIONS(Experiments)

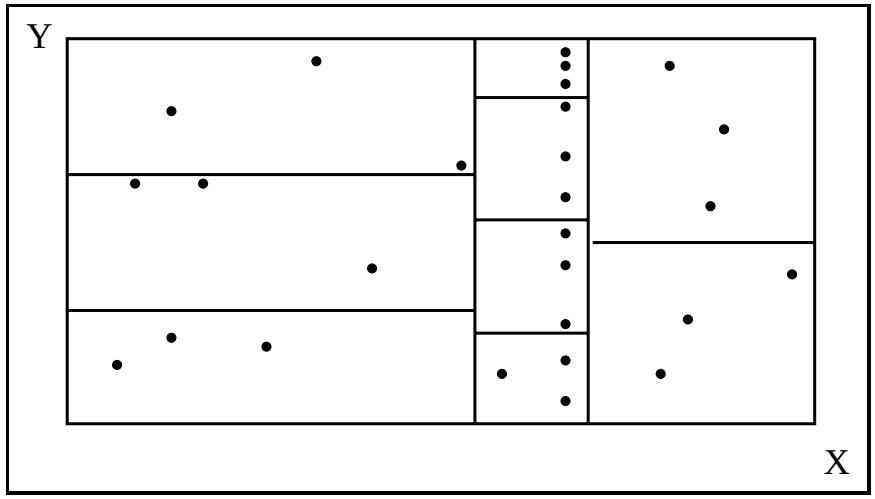
Some interesting data points:

- In memory throughput of compressed page interpretation was 15MB/s (40MB/s compressed)
- A projection of several low cardinality fields from the sales data set compressed to 1/88th of its original size.

COMPRESSING INDEXES(Pages)

Must compress two types of index pages:

- Leaf pages - Compression is identical to page level relational compression.
- Internal pages - Can use lossy compression for greatly enhanced fanout.



COMPRESSING INDEXES(Pages)

Approximating points using a coarse grid:

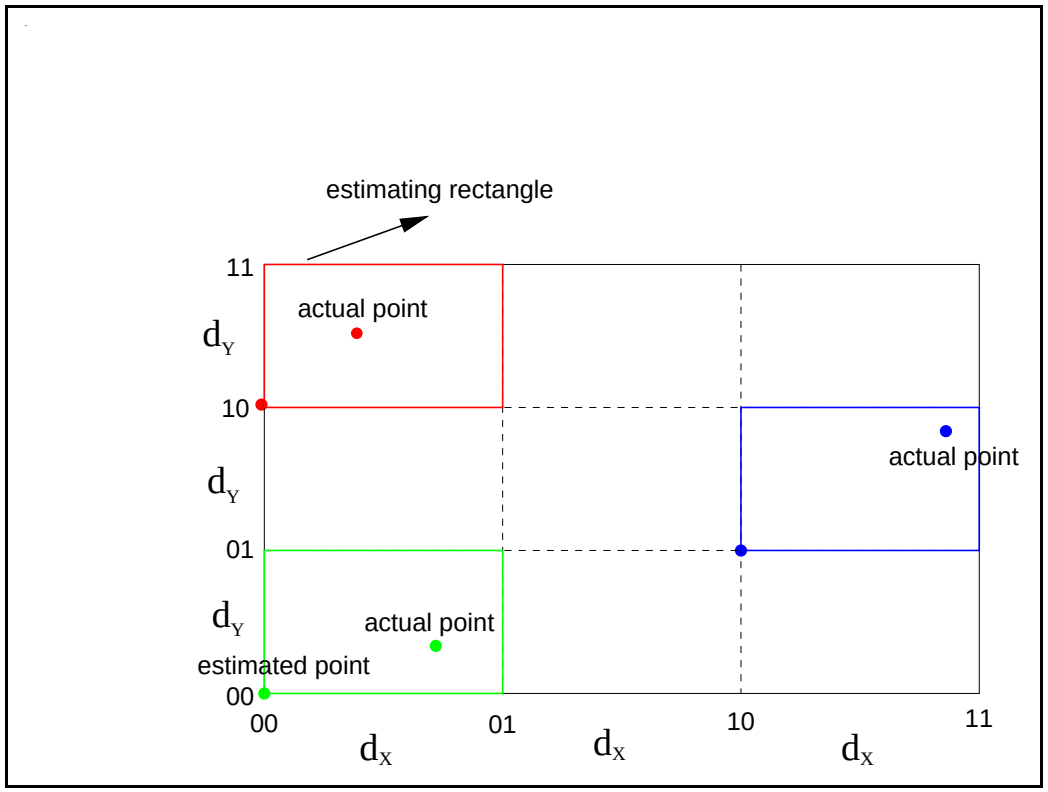


Figure 3: Point approximation in lossy compression.

COMPRESSING INDEXES(Pages)

Using lossy compression to estimate bounding boxes in internal nodes:

- Store the bounding boxes conservatively in a coarse grid
- When the frame of reference changes (on insert) reestimate the bounding boxes using the old estimates.
- Whenever a child is followed, re-estimate the bounding box stored in the parent using FOR information in the child.

COMPRESSING INDEXES(Experiments)

Compressed R-Tree performance experiment:

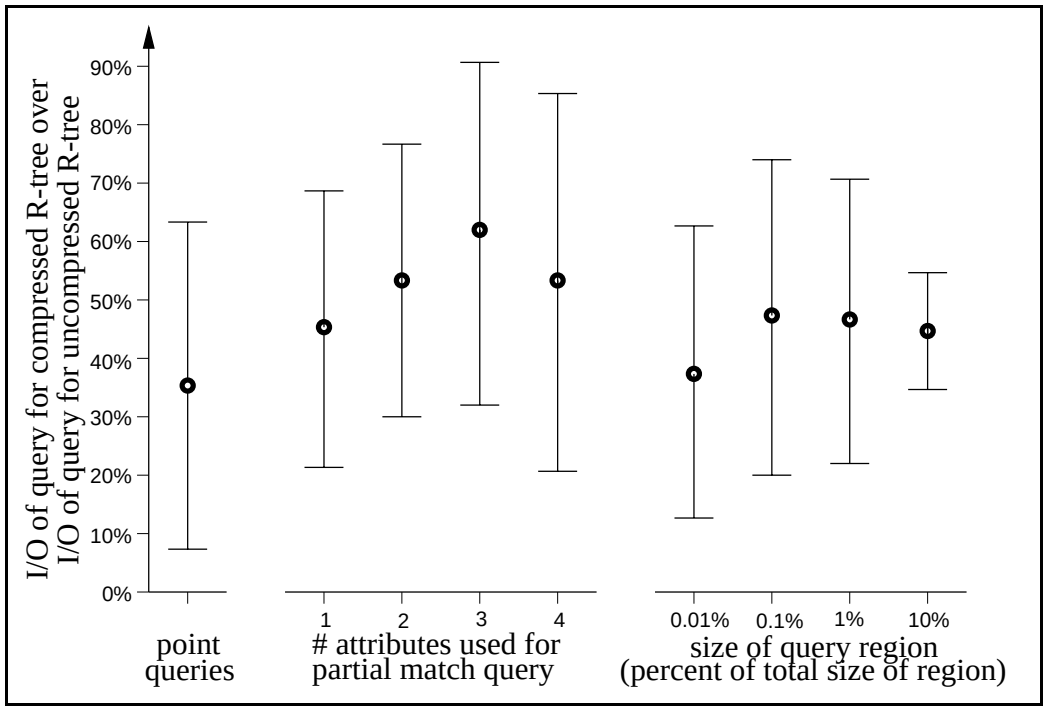


Figure 4: Compressed R-trees on the Sales Dataset.

COMPRESSING INDEXES(Experiments)

Compressed R-Tree performace experiment:

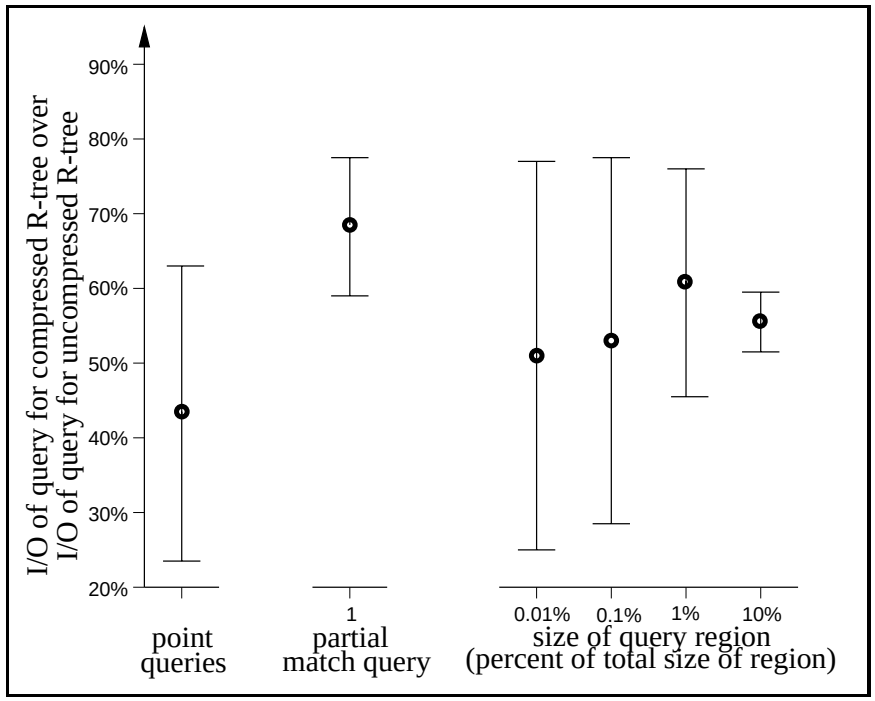


Figure 5: Compressed R-trees on the Tiger Orange County Dataset.

RELATED WORK

- BTree prefix compression used in VSAM, Ingres etc...
- Ng and Ravishankar
- Sybase IQ gzip style compression

FUTURE WORK

- Examine the application of FOR compression to interesting application areas (e.g. OLAP).
- Extend the scheme to gracefully handle variable length data.

CONCLUSIONS

- Developed a page at a time compression technique for fixed length tuple data which allows decompression on a per field per tuple granularity.
- The technique can be applied to both the leaf and internal nodes of R-Trees and B-Trees.
- Established a tie between indexing partitioning and inter-page compression optimization.
- Presented experimental results indicating the current utility of these techniques.