Reducing Memory Latency Using Data Compression

Neil Lindquist

September 21st, 2018

Compressed Sparse Row Format

Problem Setup

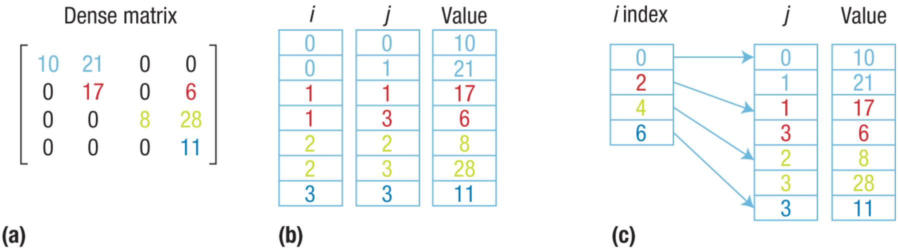

Image Credit: Buono, et al, Optimizing Sparse Linear Algebra for Large-Scale Graph Analytics

Conjugate Gradient

Problem Setup

1: **while** *distance*$(\mathbf{A}\vec{x}, \vec{b}) \geq$ *tolerance* **do**
2: Adjust $\vec{x}$ to reduce $\mathbf{A}\vec{x} - \vec{b}$
3: **end while**

High level pseudocode of the Conjugate Gradient method to solve $\mathbf{A}\vec{x} = \vec{b}$ for $\vec{x}$ given a matrix $\mathbf{A}$ and a vector $\vec{b}$ [1]

Main Loops

Problem Setup

```
for i from 0 to numRows(A) do
    sum ← 0
    vals ← rowValues(A, i)
    inds ← rowIndices(A, i)
    for j from 0 to nnzInRow(A, i) do
        sum ← sum + vals[j] · x⃗[inds[j]]
    end for
    y⃗[i] ← sum
end for
```

Pseudocode of the sparse matrix-vector product $\vec{y} \leftarrow \mathbf{A}\vec{x}$, given a sparse matrix $\mathbf{A}$, and vectors $\vec{y}, \vec{x}$ [2]

Single Precision Floating Point Numbers

- Same access restrictions as the double precision version
- Can easily be applied to only certain vectors
  - For example, by using C++'s template system
- Fixed compression ratio of 1:2
  - Worse when mixing precisions

Squeeze (SZ) Compression

- Utilizes local patterns in the data
- Stores each value as the mode used to (de)compress the data [3]
- The modes used depended on the type of data
- Compression ratio highly dependent on how well the data matches the patterns

Elias Gamma Coding

Compression Methods

- Compresses integers greater than zero by storing the number of bits needed [4]
- To compress a positive integer x:
  - Let $N = \mathit{ceil}(\log_2(x)) = \mathit{numBits}(x) - 1$
  - Store N 0's, then the $N+1$ bits of x
- Compression ratio depends on the size of the differences

Further Compression Methods

- Other, individual compression methods were tried too
  - Including huffman coding and ZFP compression
- Compression methods can be combined
  - Only combined compressions performed well in practice

References

[1] Y. Saad.

Iterative Methods for Sparse Linear Systems

[2] J. Dongarra, M. Heroux, and P. Luszczek.

HPCG Benchmark: A New Metric for Ranking High Performance Computing Systems

[3] S. Di and F. Cappello.

Fast Error-Bounded Lossy HPC Data Compression with SZ

[4] P. Elias.

Universal Codeword Sets and Representations of the Integers