GPU Acceleration of Fuzzy Tags
Natalie Jadue and Michael Kaminski

PROJECT GOAL

Improving Computational Efficiency

• Jobvite runs Fuzzy Tags computations on the cloud
  • Amazon Web Services (AWS) charges hourly rates for cloud computing
  • Faster computation will yield immediate cost savings

• Goal: Reduce computation time
  • We are NOT looking to

Frequently used data remains resident on the GPU

Data transfers between CPU and GPU memory can be slow

Better performance expected when run on Jobvite

Faster computation will yield immediate cost savings

https://www.jobvite.com/

We are NOT looking to

Created a working version of Fuzzy Tags running computations on the GPU using

Amazon Web Services (AWS) charges

Jobvite:

Implemented simplified Fuzzy Tags computations in CuPy

rates for cloud computing

Model doesn’t use gender, age, ethnicity, or other traits that could indicate these

its implementation

No need for hard coded memory copies in the naïve implementation

NVIDIA CUDA Documentation:

"Tags" refers to labels on documents

https://cupy.dev/

Initial Benchmark Tests

Produced benchmark test showing GPU accelerated computation in CuPy to be 130x faster than CPU computations

We benchmarked runtimes in a mock use case by using Fuzzy Tags as a search engine for Stack Overflow questions

Skills focused

• Can help reduce bias in the hiring process
  • Model doesn’t use gender, age, ethnicity, or other traits that could indicate these

Why CuPy?

• Python library aiming to create a GPU accelerated NumPy
  • Most functions synonymous with NumPy
  • Easy to convert the Fuzzy Tags NumPy implementation into CuPy
  • Supports CUDA Unified Memory
  • Memory manager handles data transfers via a page fault system
  • No need for hard coded memory copies in the naïve implementation
  • Frequently used data remains resident on the GPU
  • Supports manual data transfers between CPU and GPU
  • Enables fine-tuning performance achieved by CUDA Unified Memory

ACKNOWLEDGEMENTS AND REFERENCES

Future Work

• Implement portions of Fuzzy Tags in Numba
  • Numba is a "just-in-time" compiler which compiles slow Python code into fast C code at runtime

• Incorporate Basic Linear Algebra Subprograms (BLAS) to make code more efficient
  • Specialized implementations of basic operations taking advantage of hardware specifics to improve performance

• Investigate optimal sorting of Fuzzy Tags input data
  • Grouping pieces of data which are commonly used in tandem can reduce total data transfers between the CPU and GPU

TOOLS

Why CuPy?

• Python library aiming to create a GPU accelerated NumPy
  • Most functions synonymous with NumPy
  • Easy to convert the Fuzzy Tags NumPy implementation into CuPy
  • Supports CUDA Unified Memory
  • Memory manager handles data transfers via a page fault system
  • No need for hard coded memory copies in the naïve implementation
  • Frequently used data remains resident on the GPU
  • Supports manual data transfers between CPU and GPU
  • Enables fine-tuning performance achieved by CUDA Unified Memory

DATASET

• Jobvite data is private, Stack Overflow data is public but similar in structure

Challenge:

• Jobvite data is private, Stack Overflow data is public but similar in structure

Conclusions and Future Directions

Results

• Created a working version of Fuzzy Tags running computations on the GPU using CuPy and CUDA Unified Memory
  • Implemented additional optimizations to the GPU accelerated version

• Compared runtime of original CPU NumPy implementation with the GPU CuPy implementations
  • Used Fuzzy Tags as a search engine for questions from stackoverflow.com
  • Jobvite data is private, Stack Overflow data is public but similar in structure

• Read in search queries and returned the question most likely to match the search

The GPU version is more than 3.5x faster in the Stack Overflow benchmark

• At this performance level, the speed of computations do not cover the additional hardware costs for AWS instances with a GPU

• Better performance expected when run on Jobvite-owned data

• Larger computations and additional data structure will help the GPU performance

REFERENCES

• Tensorflow: https://www.tensorflow.org/
  • CuPy: https://cupy.dev/
  • NVIDIA CUDA Documentation: https://docs.nvidia.com/cuda/
  • Jobvite: https://www.jobvite.com/

CONCLUSIONS AND FUTURE DIRECTIONS

IMPLEMENTATION AND OPTIMIZATIONS

• Implemented portions of Fuzzy Tags in Numba
  • Amazon Web Services (AWS) charges hourly rates for cloud computing
  • Faster computation will yield immediate cost savings

• Goal: Reduce computation time
  • We are NOT looking to

Frequently used data remains resident on the GPU

Data transfers between CPU and GPU memory can be slow

Better performance expected when run on Jobvite

Faster computation will yield immediate cost savings

https://www.jobvite.com/

We are NOT looking to

Created a working version of Fuzzy Tags running computations on the GPU using

Amazon Web Services (AWS) charges

Jobvite:

Implemented simplified Fuzzy Tags computations in CuPy

rates for cloud computing

Model doesn’t use gender, age, ethnicity, or other traits that could indicate these

its implementation

No need for hard coded memory copies in the naïve implementation

NVIDIA CUDA Documentation:

"Tags" refers to labels on documents

https://cupy.dev/

Initial Benchmark Tests

Produced benchmark test showing GPU accelerated computation in CuPy to be 130x faster than CPU computations

We benchmarked runtimes in a mock use case by using Fuzzy Tags as a search engine for Stack Overflow questions

Skills focused

• Can help reduce bias in the hiring process
  • Model doesn’t use gender, age, ethnicity, or other traits that could indicate these

Why CuPy?

• Python library aiming to create a GPU accelerated NumPy
  • Most functions synonymous with NumPy
  • Easy to convert the Fuzzy Tags NumPy implementation into CuPy
  • Supports CUDA Unified Memory
  • Memory manager handles data transfers via a page fault system
  • No need for hard coded memory copies in the naïve implementation
  • Frequently used data remains resident on the GPU
  • Supports manual data transfers between CPU and GPU
  • Enables fine-tuning performance achieved by CUDA Unified Memory

ACKNOWLEDGEMENTS AND REFERENCES

Future Work

• Implement portions of Fuzzy Tags in Numba
  • Numba is a “just-in-time” compiler which compiles slow Python code into fast C code at runtime

• Incorporate Basic Linear Algebra Subprograms (BLAS) to make code more efficient
  • Specialized implementations of basic operations taking advantage of hardware specifics to improve performance

• Investigate optimal sorting of Fuzzy Tags input data
  • Grouping pieces of data which are commonly used in tandem can reduce total data transfers between the CPU and GPU

TOOLS

Why CuPy?

• Python library aiming to create a GPU accelerated NumPy
  • Most functions synonymous with NumPy
  • Easy to convert the Fuzzy Tags NumPy implementation into CuPy
  • Supports CUDA Unified Memory
  • Memory manager handles data transfers via a page fault system
  • No need for hard coded memory copies in the naïve implementation
  • Frequently used data remains resident on the GPU
  • Supports manual data transfers between CPU and GPU
  • Enables fine-tuning performance achieved by CUDA Unified Memory

DATASET

• Jobvite data is private, Stack Overflow data is public but similar in structure

Challenge:

• Jobvite data is private, Stack Overflow data is public but similar in structure

Conclusions and Future Directions

Results

• Created a working version of Fuzzy Tags running computations on the GPU using CuPy and CUDA Unified Memory
  • Implemented additional optimizations to the GPU accelerated version

• Compared runtime of original CPU NumPy implementation with the GPU CuPy implementations
  • Used Fuzzy Tags as a search engine for questions from stackoverflow.com
  • Jobvite data is private, Stack Overflow data is public but similar in structure

• Read in search queries and returned the question most likely to match the search

The GPU version is more than 3.5x faster in the Stack Overflow benchmark

• At this performance level, the speed of computations do not cover the additional hardware costs for AWS instances with a GPU

• Better performance expected when run on Jobvite-owned data

• Larger computations and additional data structure will help the GPU performance

REFERENCES

• Tensorflow: https://www.tensorflow.org/
  • CuPy: https://cupy.dev/
  • NVIDIA CUDA Documentation: https://docs.nvidia.com/cuda/
  • Jobvite: https://www.jobvite.com/

The Data Mine Corporate Partners Symposium 2021