Michael Maniscalco

email: michael@michael-maniscalco.com
github: github.com/buildingcpp *(c++ software architecture projects)*
github: github.com/michaelmaniscalco *(compression and algorithm projects)*
linkedin: linkedin.com/in/michael-a-maniscalco
website: michael-maniscalco.com

Expert C++ developer and software architect with a proven track record of creating and building best in class solutions across many domains in computer science.

- **FinTech software – Architect and Developer:**

  Software architect and principal developer with Lime Financial.
  Developed software for low latency trading in Linux environment.
  Designed, developed and documented next generation, low latency, distributed market data product.
  Well versed in networking, kernel bypass, multicast etc.

- **Algorithms and Computer Science:**

  - **Concurrency and Multi-Threading:**
  Work Contracts - a novel lock free/wait free, threading and asynchronous task management with significantly better latency and throughput than conventional solutions using lock free tasks queues.
  [https://github.com/buildingcpp/system/tree/main/src/library/system/work_contract/README.md]

  - Algorithms:
  MSufSort - Best in class suffix array construction algorithm. Published as "*An Efficient, Versatile Approach to Suffix Sorting" ACM Journal of Experimental Algorithmics Volume 12, Article 1.2*
  [http://www.michael-maniscalco.com/download/10.1.1.184.56.pdf]

  - Data Compression:
  M99 – Invented high performance BWT compressor: *[github.com/michaelmaniscalco/m99]*
  M03 – Invented "context aware" BWT. Best in class BWT algorithm. *[github.com/michaelmaniscalco/m03]*
  RLE-EXP – Exponential RLE *[www.michael-maniscalco.com/download/10.1.1.12.5317.pdf]*

- **Tools:**

  - **Ultra low latency instrumentation/logging/data visualization:**
  Glimpse - a highly scalable binary logging and instrumentation for C++ coupled with a suite of strong visualization tools. Can achieve sub-nanosecond throughput level in highly parallel environments.
  [source private. Available upon request]

Lime Financial

Architect and Principal Developer — *December 2020 - present*
Director of Engineering / Trade and Execution Services Division — *November 2017 - February 2019*
Principal Software Engineer and Lead Architect — *July 2015 - February 2019*

Primary Roles and Responsibilities:

Architected and implemented next generation low latency market data product:

- Highly deterministic sub-microsecond low latency performance.
- Responsible for all design and architectural decisions, authored the entire code base, all documentation and all early client interaction and support.
- Low latency networking, multicast, kernel bypass etc.
- Custom ultra-low latency (low single digit nanoseconds) instrumentation library for performance analysis.
- Highly asynchronous, virtually 100% lock free/wait free architecture for receiving exchange market feeds and producing normalized proprietary output multicast for consumption by various services across a distributed architecture.
- Robust, lightweight, low latency library for receiving the normalized proprietary market data multicast and distributing to local processes via shared memory.
- Libraries using same lock free/wait free architecture for converting normalized multicast into order books in client process address space with extremely deterministic low latency performance.
- High availability, distributed architecture with suite of specialized services in a distributed architecture to manage low latency managed/non display multicast market data, book snapshots, multicast recordings and packet retransmits, reference data, etc.
- Extremely robust, trivially easy to use, messaging library using template meta programming, compile time hashing and message routing for ultra low latency message parsing and marshaling. TMP based pattern for easily defining new protocols and the messages of those protocols. Lock free multi-producer message streams.
- Director for C++ engineering division (July 2015-February 2019) - Market Data and Trading Server products as well as continuing to serve as hands on architect and developer.

Hydrolix

Principal Software Developer — *November 2019 - October 2020*

Primary Roles and Responsibilities:

Principal software developer for stealth mode start up.

- **M99b integer compression algorithm:**
  Invented a variation of my M99 compression algorithm, employing SIMD, to achieve high compression and high speed integer compression. Implemented many filters to improve compression based on apriori knowledge of the input data (sorted, strictly sorted, etc).

- **Metrics agent:**
  Developed process which collects metrics streamed from the main process. Forward metrics from there via HTTP requests to external consumer.

Saleae

Software Architect — *February 2019 - August 2019*

Primary Roles and Responsibilities:

Developing features for next generation of company's “Logic” application. Built module for on the fly storage and retrieval of recording of live streams of data (form hardware) for use in the “Logic” protocol analyzer software.

Viasat/Intelligent Compression Technologies

Senior Developer - Acceleration and Research Technologies Division — *January 2002 - July 2015*

Primary Roles and Responsibilities:

Senior Engineer responsible for algorithm development including all compression related software, various proprietary hashes, string pattern matching algorithms, language parsers, etc. I was generally responsible for any aspect of the division's products which have either high demands in time or space and for any features which require specific coding optimizations to meet such demands.

- **Data Compression Suite:**
  Authored the company's compression engines.

- **Delta Compression Algorithm and Patent:**
  Invented and authored the company's massive scale delta compression algorithm. A system of identifying similar (not necessarily duplicate) data from a massive repository with high speed and applying it as reference data to achieve dynamic delta compression. This compression engine serves as a core piece of the company's WAN accelerator. See “Patents” below for details.

- **"Associative" HTTP Prefetching:**
  Invented a neural networking approach for predictive HTTP object prefetching for use in HTTP acceleration over high latency networks. This work is capable of identifying previously recorded HTTP experiences from a massive database which might contain similarities to the current experience and then blending these data to produce predictions of future HTTP requests and responses.

- **Video Predictive Block Compression Algorithm:**
  Authored the company's 'predictive block delta' compression algorithm which is currently used for video over HTTP acceleration with high latency connections.

Portfolio: github, algorithms, research & inventions

Contributions to computer science, open source projects, independent research, compression and sorting algorithms

- **Work Contract library:**
  github.com/buildingcpp/system

  A simple, low latency, lock free (almost wait free) threading and asynchronous task management system. Benchmarks show that this unique approach to asynchronous tasks can vastly outperform existing methods such as task queues etc and is especially well suited for use in ultra low latency systems.

- **Asynchronous Network Library:**
  github.com/buildingcpp/network

  A simple, easy to use network library designed to showcase “work contracts”. Supports async send and receive for TCP, UDP and UDP multicast.

- **Glimpse - ultra low latency instrumentation/logging/data visualization:**
  [github private - available on request]

  Ultra high performance “type rich” application instrumentation and graphical analysis tools. This software can instrument C++ applications with incredibly low overhead (8ns-20ns per sample) and provide unbelievably rich, streaming, object oriented, real time instrumentation data which can be sampled, visualized, and mined by powerful visualization tools.

- **M99 - High performance BWT compressor:**
  github.com/michaelmaniscalco/m99

  I am the inventor of the M99 entropy encoding algorithm. Originally developed in 1999 as an entropy coder for the Burrows/Wheeler transform, this algorithm is a wavelet based entropy encoder specialized for encoding data which contains locally skewed symbol probabilities. It is a very simple, extremely fast, low memory encoding scheme which is highly effective on the right types of data (such as BWT data).

- **M03 - First and only “context aware” BWT compression algorithm:**
  github.com/michaelmaniscalco/m03

  I am the inventor of the M03 context based BWT compression algorithm. M03 is a progressive encoding scheme that achieves the highest compression of any generic Burrows/Wheeler based compressor. This is the only algorithm to date which can encode the Burrows Wheeler Transform with respect to the contexts contained in the original pre-transformed data. It is a fast, low memory compressor and has appeared in the paper "Post BWT Stages of the Burrows/Wheeler Compression Algorithm" by Dr Jeurgen Abel.

- **MSufSort - suffix array construction algorithm:**
  github.com/michaelmaniscalco/msufsort

  MSufSort represented a large amount of my non-professional programming time. Over the years I have invented many specialized algorithms which have preserved MSufSort as the state of the art. When it was first introduced, MSufSort was 2-3x faster than the previous state of the art. The algorithm is described in the paper "An Efficient, Versatile Approach to Suffix Sorting", ACM Journal of Experimental

Algorithmics Volume 12, Article 1.2 as well as in the paper “Faster Lightweight Suffix Array Construction” and is cited in numerous academic papers and journals . The most recent version of MSufSort (v4 alpha) achieves highly parallel suffix array construction which easily outperforms any existing suffix array construction solution by great margins. The work, however, remains incomplete and in alpha state.

- **RLE-EXP - Exponential run length encoding:**
  www.michael-maniscalco.com/download/10.1.1.12.5317.pdf

  Inventor of the RLE-EXP (exponential run length encoding algorithm). Since its first appearance in 2001 this simple enhancement on basic run length encoding has become a defacto standard encoding stage for many modern BWT compressors. RLE-EXP also appears in Dr Jeurgen Abel's paper "Improvements to the Burrows-Wheeler Compression Algorithm: After BWT Stages"

Publications

- **An Efficient, Versatile Approach to Suffix Sorting:**
  Maniscalco & Puglisi - ACM Journal of Experimental Algorithmics Volume 12, Article 1.2
  http://www.michael-maniscalco.com/download/10.1.1.184.56.pdf

- **Faster Lightweight Suffix Array Construction:**
  Maniscalco & Puglisi - 17th Australasian Workshop on Combinatorial Algorithms (AWOCA'06)
  http://www.michael-maniscalco.com/download/10.1.1.183.9182.pdf

Patents

- **Methods and systems for utilizing delta coding in acceleration proxy servers:**
  Patent #US8010705B1

  This patent covers the delta compression algorithms used in the Viasat WAN accelerator product. The basic algorithm is capable of identifying sources which are similar (not necessarily identical) with exceptionally high speed and accuracy. Similar sources are then used as reference dictionaries to achieve extremely high compression ratios.

- **Selective prefetch scanning:**
  Patent #US9407717B1

  This patent covers a method for scanning HTML and similar response data and predicting subsequent HTTP request produced by the browser which renders the data. These predictions are used to "pre-fetch" these HTTP requests and then position the response data closer to the requester in order to reduce page load times over high latency networks.