

Resume of Dimitar Gueorguiev, PhD

Address and contact info

111 Brigham St, Unit 14A

Hudson, MA 01749

Contact phone: (978) 243-7400

Email: dimitar_pg13@hotmail.com

Linkedin: <https://www.linkedin.com/pub/dimitar-gueorguiev/7/6bb/a>

Github: <https://github.com/dimitarpg13/>

Professional Summary

Dimitar Gueorguiev is a Data Scientist / Software Engineer with over 18 years of experience in algorithm design and mathematical modeling, software development with object-oriented design, including multi-process and multi-threaded application design, numerical methods, performance analysis and testing. He has in-depth experience in parallel processing of large datasets, numerical algorithms.

- Object oriented design
- Python, C++ v11, v14, Boost, STL
- Multi-process, multi-threaded application design
- Windows, Linux, MacOS, Unix (Solaris, HP-UX, Dell Unix, AIX), Yellow Dog Linux for PS3, FreeBSD
- Performance analysis and tuning
- Version control and continuous integration (git, github, bitbucket, svn, cvs)
- Numerical Methods, Simulation & Modeling
- Relational Databases and key-value stores (PostgreSQL, Redis, RocksDB, SQLite3)
- Compilers: clang, MSVC, gcc

Education

Ph.D., Aerospace & Mechanical Engineering, Boston University, 2001, Best Student Paper Award at the 136th meeting of Acoustical Society of America; John J. and Helen Carey Fitzgerald Award for outstanding research in the Department of Aerospace and Mechanical Engineering, College of Engineering, Boston University, 1999

B.S. Applied Mathematics and Informatics, Technical University of Sofia, 1995

M.S., Mechanical Engineering, Technical University of Sofia, 1994

Professional Experience

Nike, Boston, MA

September 2019 - present

Lead Data Scientist

Since January 2020 performed complete refactoring and worked on introducing new objectives and features of the adversarial reinforcement learning algorithm for fulfillment optimization. For the purpose used various optimization techniques, reformulated the algorithm as Mixed Integer Problem by linearization of the original objective function which introduced multi-fold speedup of the core algorithm. The algorithm is executed in AWS EKS service for product fulfillment at Nike and serves for fulfilling Nike Digital orders throughout North America. Worked on introducing new ML Demand prediction service into the Fulfillment Optimization AWS solution and incorporated its predictions into the Fulfillment optimization engine. Integrated external REST API services with the Fulfillment Optimization Engine thereby providing live inputs and asynchronously via lambda functions to the core

algorithms of the engine. Worked on enhancing the *Celect* Machine Learning pipelines with new features such as metadata logging service to database and entity creation parser. Researched various algorithms for optimal inventory placement and simulation framework for predicting peak inventory positions. At present, researches various algorithms for Root Cause Analysis using Bayesian inference and Probabilistic Temporal Logic. Looking into improving the learning component of the algorithm by various means including reformulating the parameters in the Markovian algorithmic state and abandoning the Markov property using instead a non-Markovian cache of algorithmic states.

Technology: Python, pandas, numpy, networkx, pyspark/graphx, scikit-learn, or-tools, gurobi, coin-or/cbc, AWS EKS, AWS Lambda, Postgres, Redis

Celect Inc (acquired by Nike), Boston, MA

February 2019 – August 2019

Senior Data Science Engineer

From February 2019 worked on the development of a new frequency domain forecaster which predicts future trends and seasonal/cyclical events in datasets from retail/fashion industry. For the purpose have developed a super-resolution algorithm which poses the problem as a convex optimization problem in frequency domain and solves it using FFT relaxation technique.

Technology: Linux, MacOS, C++ v11, or-tools, Python, scikit-learn, numpy, pandas

Qlik Tech, Newton, MA

October 2015 – January 2019

Senior Software Engineer

From December 2017 until 2019 worked on productizing the Qlik Big Data Index (*BDI*) engine which already passed the prototyping phase. Worked on the indexing algorithms and moving of column indexlets within the *BDI* engine and preparing them for consumption by the *BDI* Query Executor. The *BDI* engine reads the symbols from the data sources in Parquet format and stores the compressed symbols using *FarmHash* and *Roaring* compression in *RocksDB* instance configured in a cluster environment. The *BDI* engine is a proprietary distributed in-memory column store, running in Linux cluster which augments the standard Qlik Engine in processing very large Qlik applications in AWS Cloud settings and on premise.

From June 2017 to November 2017 worked on *Data Engineering* stack which is the current Qlik Big Data processing pipeline using Docker swarm with overlay network deployed in AWS. Implemented Cloud-ready monitoring and logging service which exports Qlik Engine performance metrics and logging information to *Qlik Cloud Gen 3* running on AWS.

From October 2016 to May 2017 worked on Centralized Logging service storing metrics from the Qlik Engine and other services into a Postgres DB instance. For the purpose employed Log4Net library executing in multithreaded and multi-process environment using .NET thread pools.

From November 2015 to September 2016 worked on defining Qlik Engine performance metrics and exporting them from the *Qlik Engine* to the *Qlik Repository service*. For the purpose, instrumented the *Qlik Engine* scheduler to measure with configurable degree of accuracy the time *Qix Requests* spend executing on a thread, waiting in a wait queue, and waiting on monitors. Added an important new engine feature which exports log entries from the Qlik Engine with the id of every object which has been clicked on by the user which would be used to reconstruct usage patterns and compile valuable for the customer data.

Technology: Linux, MacOS, Windows; C++ v14, C++ v11, C#, JavaScript, Perl, Python

EMC Corp, Hopkinton, MA
Principal Software Engineer

February 2001 – October 2015

From June 2013 to October 2015 participated in the design and implementation of *WorkloadPlanner*. *WorkloadPlanner* is a simulation module integrated in the *Unisphere for VMAX* software product. *WorkloadPlanner* simulates the performance of the target array by calculating its component utilizations and back-end response times. Designed the component simulation models of CPUs / physical and logical cores, Infiniband Fabric, I/O boards, disks and I/O ports which developed and validated in Java. Responsible for the design, implementation and validation of a simulation model for the proprietary EMC *Symmetrix Fully Automated Storage Tiering (FAST)* feature, which was fully implemented, validated and unit tested in Java. The *FAST* engine decides how to move storage extents within the available storage pools of disks at runtime such that the characteristics of the workload residing on each storage pool match the performance profile of the disks on the same pool thereby satisfying the Service Level Objective (SLO) and Service Level Expectation (SLE) constraints. The *WorkloadPlanner FAST* model is a constrained optimization solver written in Java, used to decide if the incoming workload can be accommodated on the existing target array pools with respect to performance and capacity without violating the storage group SLOs and the storage pool SLE response times and is used in order to estimate the available headroom on the storage array back end.

From July 2009 to May 2013 participated in the research & development of *TierAdvisor*. *TierAdvisor* was a multi-threaded, object-oriented, software application, written in C# and C++, which simulated the performance impact of Fully Automated Storage Tiering (*FAST*) on a generic EMC storage system in terms of disk utilization, response time as well as relative price. *TierAdvisor* presents tiered view of the storage system and explores alternative tiered configurations based on user defined tier mixtures inside customized Storage Policies. The new storage system (Target) uses the workload and configuration information of existing (Source) EMC *Symmetrix* and *Unified* storage arrays. For the purpose *TierAdvisor* collected and analyzed workload and configuration data of the EMC *Symmetrix* and *Unified* data storage arrays. Redesigned the *TierAdvisor's WorkloadProcessor* module such that Virtual Provisioning pool stats and allocation levels for device extents mapped to multitude of device pools are accounted and estimated by fast heuristic algorithm rendering very good estimation of the former. Researched various analytical techniques such as the *Wavelet Transform Modulus Maxima* and *Histogram Method* as means of obtaining approximation for sub-LUN skew for various workload mixtures. Worked on the next gen of *TierAdvisor* compute engine which was *SQLite* extension module simulating the work of the *FAST* subsystem residing entirely inside *SQLite* database.

From January 2003 to August 2009 participated in the design, implementation, maintenance and support of *SymmMerge*. *SymmMerge* was multithreaded performance simulation tool with object-oriented design, written in C++ and later rewritten in C#. *SymmMerge* estimates the performance of EMC *Symmetrix* storage array based on supplied configuration information and workload trace data using queueing network utilization model. Participated in every aspect of the object-oriented design & implementation of *SymmMerge* – enhancement and tuning of the utilization algorithms, workload processing, the algorithms for target generation, parser system, UI enhancements, Unit tests update, third party library source code maintenance. Extended the existing disk model to simulate realistic disk seek times using the disk layout configuration and greedy algorithm for distributing the IOs on the disk hyper volumes.

From February 2001 to January 2003 participated in development of various tools facilitating and driving storage performance tests written in C (*Bowtie*, *BuildVolumeMap*) and drivers (*iodriver*), as well as writing shell scripts driving the tests.

Technology:

Windows, Unix, Linux: C++98, C, C#, Java, Matlab

Dept. of Aerospace and Mechanical Engineering, Boston University, Boston, MA

September 1998 – January 2001

Graduate Research Assistant

Developed high frequency FEM model for the vibrations of MEMS resonator with high Q factor, using 2D solid elements. The studied resonators had single- and double-periodic structures which create regions with localized energy in certain frequency ranges thereby yielding high Q factor. The 2D mesh of finite elements was determined by using the quarter wavelength rule. Using these FEM models there were developed a series of realistic designs with various lengths and different geometries and were manufactured using the existing MUMPs technology at Boston University. The measured resonances, stop and pass bands have been found with agreement with the predictions by the FEM models.

Technology:

Solaris, Windows; Bourne shell, Matlab, COMSOL, CALFEM

Publications

Analysis of Floquet wave generation and propagation in a plate with multiple arrays of line attachments, Dimitar Gueorguiev, James G. McDaniel, Pierre DuPont, and Leopold Felsen, Journal of Sound and Vibration, 234(5),819-840

Simplified dispersion relations for Floquet waves in a plate with multiple arrays of attachments, Dimitar Gueorguiev, James G. McDaniel, Pierre DuPont, 1999 ASME Design Engineering Technical Conferences September 12-15, Las Vegas, Nevada

Additional Training

DSP for Wireless Communication, Fall 2009, IEEE Boston

Debuggers for Modern Applications – Performance and Static Analysis, Sept 25, 2018, CppCon

Parallelism in Modern C++: From CPU to GPU with Michael Wong, Sept 21-22, 2019, CppCon

Additional Information

As a hobby enjoys tinkering with NVidia Jetson platforms (TX1/TX2) using Cuda, coding and debugging machine learning algorithms with it.

Topics I am currently interested in:

Reinforcement Learning/Game Theory:

https://github.com/dimitarpg13/reinforcement_learning_and_game_theory

Root Cause Analysis and Model verification:

https://github.com/dimitarpg13/root_cause_analysis_and_model_checking

Bayesian Inference/Bayesian Networks: https://github.com/dimitarpg13/learning_bayesian_networks

Python related stuff: <https://github.com/dimitarpg13/UnderstandingPythonEcosystem>

Tensorflow source: <https://github.com/dimitarpg13/InsideTensorflow2Source>

Graph Theory and Dynamic Programming:

https://github.com/dimitarpg13/graphs_and_dynamic_programming

AI concepts: <https://github.com/dimitarpg13/aiconcepts/blob/master/docs/SemanticSimulation.pdf>