Minor E. Gordon, Ph.D.
Curriculum Vitae

cv256@minorgordon.net   https://github.com/minorg
https://mg4.consulting   Cohoes, NY 12047

Summary

I am a **computer scientist**, **software architect**, and **thinker-doer**. I am passionate about applying technology in ways that help individuals and society. I am a lifelong learner and avid reader, not just in the depth of my field but in the breadth of others, especially history.

Education

2009	**Ph.D., Computer Science**, University of Cambridge
2005	**Diplom Informatiker**, Technische Universität Berlin
2003	**B.Sc., Computer Science**, Oklahoma State University Tulsa

Work Experience

06/2019 –   **Director, Semantic Web Applications** at <u>Rensselaer Polytechnic Institute</u> / Troy, NY

Responsibilities   <u>Technical leadership</u>: mentoring junior staff and students, code reviews, process development
<u>Recruiting</u>: reviewing resumes, coordinating, interviewing for student and staff positions
<u>Software development</u>: design and implementation of technically challenging work streams

07/2012 –   **Independent software consultant** at <u>MG4 Consulting, LLC</u> (self-employed) / Remote

04/2018 – 4/2019   **Software architect and technical lead** at <u>Asemio, LLC</u> / Tulsa, OK
Asemio is a technology consulting company for community data systems. The company works with organizations that are solving complex problems in the criminal justice, education, health, and social sectors.

Responsibilities   <u>Project planning</u>: user story mapping, release design, technical architecture, time estimation
<u>Subject matter expertise</u>: responding to RFPs and other internal and external needs for technical expertise
<u>Technical leadership</u>: mentoring junior developers, code reviews, process development, flower arranging
<u>Recruiting</u>: outreach, reviewing resumes, coordinating, interviewing
<u>Software development</u>: design and implementation of strategic projects

12/2011 – 06/2012   **Software architect and full-stack developer** at **Grokio, LLC** / New York, NY
GrokIO was a business intelligence startup I co-founded.

08/2011 – 12/2011   **Software architect and backend developer** at <u>Birchbox Inc.</u> / New York, NY
Birchbox is a personalized subscription service for women's cosmetics samples.

05/2011 – 08/2011   **Performance engineer** at <u>Chartbeat Inc.</u> / New York, NY
Chartbeat is a real-time web analytics software-as-a-service provider.

10/2010 – 04/2011   **Contract software engineer** at **IBM Research** / Hawthorne, NY

01/2009 – 07/2010   **Programmer / Analyst** at **NEC High Performance Computing Europe GmbH** / Stuttgart, Germany

Academic Projects

6/2019 –   **Explainable artificial intelligence for wireless spectrum allocation** at RPI / Troy, NY
Designed and prototyped a domain-specific language in Scala for representing wireless spectrum access policies as well as requests for spectrum access, then mentored a student in further implementation

9/2019 –   <u>**Open source provenance-aware knowledge store**</u> at RPI / Troy, NY
Designed and implemented a new provenance-aware knowledge store in Java for use in multiple sponsored research projects. Implementation work included: a REST API for manipulating the store and Java and Python

clients for the API, a SPARQL 1.1 endpoint, a public-facing Java library API, Docker images on Dockerhub, Java artifacts on Maven Central, documentation on readthedocs.io, and Continuous Integration with CircleCI.

06/2003 – 12/2008 **Yield: a high-performance C++ application server**
Yield was the vehicle for both my Diplom and Ph.D. research. Its features included:
- Staged, event-driven concurrency
- Non-blocking socket I/O (TCP, SSL, UDP) with efficient polling (epoll, kqueue, event ports)
- Asynchronous file I/O (POSIX AIO, Linux libaio, Windows I/O completion ports)
- File system event notification (inotify, kqueue, OS X volume monitoring, ReadDirectoryChangesW)
- Atomic reference-counted objects (similar to the Boost/TR1 shared_ptr) with efficient use conventions
- Optimized hash tables: cuckoo hash tables, HAT tries, string array hash tables
- Non-blocking data structures: queues, stacks, circular buffers
- HTTP request and response parsing with Ragel state machines based on the ABNF in RFCs 822 and 2616
- URI parsing with a Ragel state machine based on the ABNF in RFC 3986
- HTTP clients and servers with support for pipelining and chunked encoding
- ONC-RPC, XML-RPC, and JSON-RPC clients and servers with efficient parsing and serialization
- Efficient SOAP parser (libxml2, SAX with DOM fallback) and serializer, SOAP clients and servers
- Object marshalling: XDR, XML (libxml2, genx), JSON (yajl)
- Embedded script interpreters and gateways: Python, PHP, Lua, Prolog, CGI, FastCGI, SCGI
- Minimalist xUnit-like unit testing framework (yunit) with one set of unit tests per class

Servers implemented:
- HTTP static file servers, benchmarked with httperf generating a SPECweb99-like workload
- Back end image processing pipelines (decode JPEG, scale, crop, blur, encode JPEG), benchmarked with a custom client and real application traces
- English Wikipedia page lookup and keyword search engine using Berkeley DB and various NLP libraries, benchmarked with httperf with URI logs synthesized from Wikipedia page statistics

Yield's concurrency architecture is based on the concept of *stages* for parallelizing CPU-intensive code (gzipping, parsing, serialization) and offloading blocking calls (disk I/O, DNS lookups). A stage is a *unit of concurrency*: two stages can always run on two different physical processors with minimal synchronization.

Industry Projects

05/2019 – 6/2019 **Data entry web application** for **PSW Applied Research Inc.** / Toronto, ON
Bootstrapped a data entry Semantic Web application using Flask, rdflib, SPARQL, and Virtuoso on the backend, TypeScript, React, and GraphQL on the frontend, and Docker deployments to multiple environments
Documented and handed off the code to a junior developer

04/2018 – 4/2019 **Privacy-preserving record linker** at **Asemio, LLC** / Tulsa, OK
Designed and implemented a privacy-preserving record linker for determining overlap between HIPAA and FERPA protected data sets, using the Sharemind platform for secure multi-party computation, Python libraries and scripts, and C++

Wrote a white paper describing the record linking process and comparing its results to plaintext record linking with deterministic and probabilistic algorithms

Drafted and was awarded a $25,000 grant from a national foundation to fund this work

04/2018 – 4/2019 **Community asset mapping platform** at **Asemio, LLC** / Tulsa, OK
Designed and implemented a web application for tracking community assets such as service organizations, funders, collaboratives, data sharing platforms, et al. The web application was developed with TypeScript+React.js, Python+Flask, and a neo4j graph database. Data was entered by subject matter experts using a bespoke Python-based domain-specific language.

06/2017 – **Motivated Cognition experiment** at **Columbia University** / New York, NY
Motivated Cognition is a psychology research experiment investigating the effects of cognitive bias on our consumption of information online.

Created a complete web application from high-level requirements, using TypeScript, Knockout.js, and webpack

on the frontend; Java with microservices, YAML configuration, and MongoDB persistence on the backend; generated code for models and services in TypeScript and Java; Jenkins continuous integration; and Docker deployment

Translated screens from a graphic designer into pixel-perfect CSS and HTML

Integrated with Mechanical Turk for subject recruitment

04/2017 – 05/2017 **Drug name comparison project** for **PSW Applied Research Inc.** / Toronto, ON
Implemented Kondrak's phonetic alignment and similarity algorithm (ALINE) and a bigram-based orthographic algorithm (BI-SIM) as a PostgreSQL C extension

Built a JavaScript demo application using jQuery and Bootstrap

04/2016 – 04/2018 **Full-stack development** for **Polygon Analytics Ltd.** / Edinburgh, Scotland
Designed and implemented proprietary software in C++ and Python

09/2014 – **Notablist email newsletter search engine** for Notablist, Inc. / New York, NY
Notablist indexes millions of email newsletters from hundreds of thousands of publishers and provides both a search interface and real-time change alerts to users.

Backend prototyping in Python:
• Classified signup responses and mapped form inputs using numpy/scipy/scikit-learn

Backend production rewrite in Java:
• Generated Python JSON-RPC clients, Python command line tools, Java service interfaces and abstract implementations, and Java JSON-RPC servlets from Thrift interface and data structure definitions
• Implemented Guice-injected Java micro-services that encapsulated MongoDB collections; ElasticSearch indices; Redis databases; S3 buckets; screenshotting and signups with Selenium; SpamAssassin checking; logstash queries; Stripe integration; Drip (CRM) integration; MailChimp, SparkPost, and SendGrid transactional email posting; etcd locking; DNS and whois querying and parsing; DMOZ category and Alexa and Quantcast rank lookups
• Designed and implemented a scalable distributed system for processing Common Crawl WARC records and submitting newsletter signups
• Reported metrics to InfluxDB
• Created administrative user interface in Vaadin
• Set up Jenkins continuous integration
• Deployed with Docker+Kubernetes on real hardware

Frontend:
• Assumed responsibility for a three year-old Bootstrap+Backbone.js+Marionette.js code base in JavaScript
• Incrementally migrated code to TypeScript with Knockout.js
• Added Selenium (Java) browser tests

04/2014 – 07/2014 **Development of the Muninn Project** for **PSW Applied Research Inc.** / Toronto, ON
Created a command-line Java application for transcribing handwritten World War I medical records from the Canadian Expeditionary Force (http://blog.muninn-project.org/node/79) using Amazon's Mechanical Turk crowdsourcing service: precut images for groups of lines, redundancy between workers, artificially-introduced mistakes and edit distances to detect cheating, and feedback to workers. The results were published as Linked Open Data on the Canada Open Data Portal.

03/2013 – 07/2013 **Txtmrkt, a marketplace for digital publications** / New York, NY
Designed and implemented a Model-View-Controller-based Java web application with Guice-injected microservices backed by Amazon S3

Authenticated and authorized users with Shiro

Integrated co-branded order placement and payment notification with Amazon's Flexible Payments Service

07/2012 – **TeraScript product suite** for **Tronics Software, LLC** / El Dorado, CA
TeraScribe and TeraScript Server are a visual development environment and server-based runtime for

TeraScript Action Files (TAFs), TeraScript Class Files (TCFs), and the TeraScript Markup Language (TML). TML is a markup-based web application language, similar in spirit to ColdFusion. TAF and TCF are ways of organizing TML.

TeraScript Server 8:
Rewrote TeraScript 7 in Java while maintaining strict backwards compatibility:
• Grammar-based compilers for TML and associated little languages using ANTLR
• Tree interpreters for TML, TAF, and TCF
• Versioned OSGi bundles with Maven and Apache Felix
• Library of standard functions, primitive types, and collections with extensive unit tests

TeraScript Server 7 (formerly Witango Server):
Reorganized, cleaned up, and modernized a 17-year-old C++ code base and addressed numerous bugs and feature requests for the server's first major release in over two years; now in maintenance releases

Eliminated diverging platform-specific build systems (on Win32, Linux, and OS X) in favor of CMake and ported the code base to Win64 using Visual C++ 2012

Designed and implemented a SQL generator that visits dialect-specific trees of SQL-99 constructs

TeraScribe 8:
Reorganized and cleaned up a 10-year-old Java Swing code base in order to add a number of features to the code editor, including autocomplete and syntax highlighting, and refit the data source management interface to use JDBC metadata

Subcontracting:
Debugged production installations of TeraScript Server 7 and 8

Developed cross-platform (Android, iOS, Windows) mobile app prototype with Xamarin Forms and C#

12/2011 – 06/2012   **Haystack business intelligence platform** for **Grokio, LLC** / New York, NY
Designed an agent (Python) and manager (Java with Spring Security+MVC and Guice) architecture for gathering, storing, and querying time series data and metadata

Implemented time series databases from scratch using relational tables and memory-mapped files

Wrote RFC 3986 grammar-based URL and URN parsers using Ragel (Java) and pyparsing (Python)

Generated service interfaces in Java and Python with custom Protocol Buffers compiler plugins

Coded agent plugins in Java and Python for discovering and fetching metric metadata and data:
• Open source monitoring systems: Nagios, Ganglia, collectl, Graphite
• Open source server software: Apache httpd, nginx, MySQL, memcached, MongoDB, JMX
• Platform interfaces: procfs, sysctl, WMI, SNMP
• Third party APIs: Google Analytics, MailChimp, Spring Social, Amazon CloudWatch

Supervised user experience and graphic designers in Ukraine

Implemented a client-side user interface with the Google Web Toolkit, starting from wireframes:
• Architectural best practices: Model-View-Presenter; event bus; activities and places
• Custom widgets: date time range picker, cell table pager, search box, selection tree
• .war deployment to Tomcat with Maven

08/2011 – 12/2011   **Backend development and software architecture** for <u>Birchbox Inc.</u> / New York, NY
Designed, implemented, documented, and successfully deployed warehouse integration code in Java (backend) and PHP (administrative frontend)

Wrote a Thrift compiler and targets in Python+pyparsing to generate Java and PHP from Thrift definitions

Produced ad hoc business reports from MySQL using Java and Jython

05/2011 – 08/2011   **Benchmarking server stacks in order to provision EC2 instances** at <u>Chartbeat Inc.</u> / New York, NY
Synthesized a realistic benchmark for front-line web servers from statistical analyses of nginx access logs

Significantly reduced network bandwidth and latency on production servers with targeted optimizations

Wrote Python scripts for analyzing Ganglia RRDs, replaying HTTP request streams, harnessing httperfs

10/2010 – 04/2011	**File I/O library for <u>X10</u>** at **IBM Research** / Hawthorne, NY

X10 is a type-safe, parallel object-oriented language for high-productivity computing.

Designed, implemented, tested, and documented a new low-overhead, buffer-based file library for X10:
- inspired by POSIX, FUSE, Boost.Filesystem, and Java NIO.2
- X10 and native code for the Java and C++ source-to-source compilation backends
- Scatter/gather I/O, aligned buffers, memory-mapped files, advisory locking

Microbenchmarked the X10 runtime

01/2009 – 07/2010	**<u>LXFS</u>: fast and reliable data storage for computing clusters,** for **NEC HPCE GmbH** / Stuttgart, Germany

Wrote Python and bash scripts for configuring, deploying, and administrating LXFS installations: Lustre and lnet configuration (MGS, MDS, OSS, clients); redundant NFS exports; Nagios, Ganglia, and collectl monitoring; Heartbeat/Linux-HA services for failover; Promise RAID devices; network interfaces (Infiniband, bonded Ethernet)

Finished two major LXFS releases, used in numerous deployments

Debugged and resolved issues in production parallel file systems

01/2009 – 07/2010	**<u>XtreemFS</u>: a distributed and replicated file system for WANs,** for **NEC HPCE GmbH** / Stuttgart, Germany

Designed, implemented, and tested the XtreemFS userspace client:
- FUSE and Dokan (Windows FUSE-like library) interfaces
- Multiple pipelined ONC-RPC streams to a a single server
- Staged, event-driven concurrency for robust performance under load
- Close-to-open file caching with per-file page sizes
- Automatic failover between file replicas on timeout
- Heavily benchmarked under different I/O loads (iobench, dbench, metadata benchmarks)

Finished two major XtreemFS releases, both used in production

Collaborated with academic and industry partners in Europe, Israel, and China

Open Source Projects

11/2015 –	**<u>DressDiscover</u>**: suite of web applications for digital costume collections
7/2016 –	**<u>Pastpy</u>**: Python library for reading PastPerfect (museum database) exports
7/2012 –	**<u>Thryft</u>**: compiler and code generation framework for multilingual web applications

Skills

Languages	English (native); German (fluent)
Programming	Java; Python; TypeScript; C++; C; SQL; C#; JavaScript; Scala
Frameworks	Thrift; Protobufs; Flask; Bootstrap; Knockout.js; React.js
Databases	ElasticSearch; MongoDB; Redis; PostgreSQL; SQL Server
Platforms	Windows; Debian- and RedHat-based Linux; OS X
IDEs and tools	Visual Studio; Eclipse; IntelliJ; Visual Studio Code; Maven; CMake
Revision control	Git; Subversion

Teaching Experience

02/2014 – 04/2014	SQL Fundamentals and Python Fundamentals workshops for Ph.D. students and post-doctoral researchers in the natural sciences at Iowa State University / Ames, IA
01/2002 – 05/2004	Technical supervision of Masters' theses at Oklahoma State University / Tulsa, OK
01/2002 – 03/2002	Mathematics tutoring for 6th and 7th graders at Tulsa Public Schools / Tulsa, OK

Research Experience

2005 – 2008   **Doctoral student** / University of Cambridge, Cambridge, UK
Supervisor: Jean Bacon
Examiners: Steven Hand, University of Cambridge; James Larus, Microsoft Research
Research focus: high-performance web servers
Dissertation: Stage scheduling for CPU-intensive servers
Funding: EPSRC grant for the EDSAC 21 project

2003 – 2005   **Student research assistant /** Zuse Institut Berlin, Berlin, Germany
Supervisors: Alexander Reinefeld, Hans-Ulrich Heiß
Research focus: high-performance web servers

2002 – 2003   **Student research assistant** / Oklahoma State University, Tulsa, OK
Supervisor: Marcin Paprzycki
Research focus: software agents, Semantic Web technologies
Diplomarbeit: Staged design for highly concurrent web servers
Funding: full academic scholarship from the State of Oklahoma

Selected Publications

Gordon, M. (2010). Stage scheduling for CPU-intensive servers (No. UCAM-CL-TR-781). University of Cambridge, Computer Laboratory.

Gordon, M. (2007). Twisted in knots [a review of *Twisted Network Programming Essentials* by Abe Fettig]. IEEE Distributed Systems Online, (3), 5.

Gordon, M. (2006). Small-scale peer-to-peer overlays. ACM SIGOPS Operating Systems Review, 40(3), 45-48.

Hupfeld, F., & **Gordon, M.** (2006, November). Using distributed consistent branching for efficient reconciliation of mobile workspaces. In 2006 International Conference on Collaborative Computing: Networking, Applications and Worksharing (pp. 1-9). IEEE.

Gordon, M. (2005). An introduction to network programming the Python way [a review of *Foundations of Python Network Programming* by John Goerzen]. IEEE Distributed Systems Online, 6(10).

Gawinecki, M., **Gordon, M.**, Kaczmarek, P., & Paprzycki, M. (2005). The problem of agent-client communication on the Internet. Scalable Computing: Practice and Experience, 6(1).

Gordon, M., & Paprzycki, M. (2005, July). Designing agent based travel support system. In The 4th International Symposium on Parallel and Distributed Computing (ISPDC'05) (pp. 207-216). IEEE.

Gordon, M. (2004). A Textbook for the Semantic Web [a review of *Knowledge Representation* by John Sowa]. IEEE Distributed Systems Online, (1), 6.

Gordon, M. (2004). An introduction to RDF technologies: Too little, too soon [a review of *Practical RDF* by Shelley Powers]. IEEE Distributed Systems Online, (8), 5.

Paprzycki, M., Gilbert, A., & **Gordon, M.** (2002, October). Knowledge representation in the agent-based travel support system. In International Conference on Advances in Information Systems (pp. 232-241). Springer, Berlin, Heidelberg.

Angryk, R., Galant, V., **Gordon, M.**, & Paprzycki, M. (2002, June). Travel Support System: An Agent Based Fram ework. In International Conference on Internet Computing (pp. 719-725).