

Thadeus Burgess

Austin, TX
(512) 587-0171
thadeusb@thadeusb.com

Experienced Senior Software Engineer with a robust background in designing and implementing scalable data storage, processing systems, and distributed RESTful services, particularly in the healthcare domain. Proven expertise in leading the development of HIPAA/HiTech compliant web applications and contributing to award-winning projects. Skilled in full-stack development, big data technologies, and agile project management, with a passion for intelligent systems and innovation.

Skills

- Languages & Frameworks: Python, Scala, Node.js, Flask, SQLAlchemy, React, jQuery
- Databases & Storage: PostgreSQL, MongoDB, Redshift, Redis, RabbitMQ, AWS S3, Parquet
- DevOps & Cloud Services: AWS, GCP, Docker, Jenkins, Terraform, Vault, Ansible
- Data Processing & Analysis: Apache Spark, Hadoop Ecosystem, XGBoost, scikit-learn, JupyterHub
- Web Technologies: HTML5/CSS3, Bootstrap, LESS, JSON
- Version Control & Collaboration: Git, Code Review Practices, Leading and mentoring technical teams
- Methodologies: Agile/Scrum, Test-Driven Development, Continuous Integration/Deployment, GitOps
- Patents Received: Establishing a Mesh Network - US 20130279409 A1, Communicating Data in a Mesh Network - US 20130279410 A1

Professional Experience

Staff Software Engineer / Director of Engineering

ClosedLoop.ai

12/2017 - Present

As a senior technical leader and architect, I led significant enhancements in backend systems, data processing, and cloud infrastructure, substantially elevating scalability and performance. My contributions were instrumental in ClosedLoop.ai's triumph in the CMS.gov AI Health Outcomes Challenge and in securing the Best in KLAS award for two consecutive years. My technical proficiency encompassed cloud architecture, API development, big data processing with Apache Spark, infrastructure automation via Terraform, and advanced programming in Python, underpinning the creation of innovative, high-performance platform features.

- As Director of Engineering, successfully balanced executive leadership responsibilities with hands-on technical contributions, driving significant product advancements while fostering a culture of excellence and innovation within the engineering team.
- Led the charge in championing best practices in software development, driving the adoption of standardized coding conventions, continuous integration/deployment strategies, agile methodologies, and rigorous peer review processes.
- Managed a platform that efficiently serviced over 3 million patients daily, incorporating live models operating under strict Service Level Objectives (SLOs). Additionally, the platform supported ad-hoc analytical tasks conducted by the data science team, ensuring flexibility and responsiveness to evolving data needs.
- Established a standard "Product Enhancement Proposal" (PEP) process for introducing new features within the company, drawing inspiration from the Python PEP framework. This initiative fostered a structured approach to innovation, ensuring thorough review and alignment with our strategic goals.

- Developed high-performance Scala APIs leveraging the Lagom framework, Akka Streams, and various cloud services. Focused on efficiency and scalability, these APIs enhanced system responsiveness and throughput, seamlessly integrating with cloud infrastructures for optimal performance.
- Implemented a scalable storage system in Spark, utilizing parquet tables to efficiently store machine learning predictions and SHAP scores. This setup was designed for optimal performance in both data appending and querying, catering to the demands of high-volume, dynamic prediction environments.
- Developed an enterprise-grade service message bus, leveraging Amazon SQS/SNS, complemented by a per-service background job processing system. This architecture fed into a MongoDB-backed priority queue, which was designed to incorporate business logic and deduplication strategies for optimizing work selection. This system empowered users with enhanced control over queue management, ensuring efficient and logical task prioritization.
- Developed a sophisticated routing system designed to intelligently dispatch requests across multiple Spark clusters, tailored to the specific nature and parameters of each job. This strategic approach significantly enhanced system reliability by segregating tasks such as data loading, reporting, analytics, and machine learning into dedicated clusters optimized for those functions. The system also featured advanced priority queueing mechanisms for job selection and management, incorporating automated retry protocols to efficiently handle job failures and ensure smooth operation.
- Designed and implemented a versatile self-service data ingestion and export framework for customers, accommodating a variety of data transfer methods tailored to diverse client requirements. This solution included secure SFTP, AWS S3, JDBC connections, and CURL uploads, providing a flexible and user-friendly API for clients to efficiently send and receive data from the platform.
- Integrated Auth0 and JWT tokens into our product to establish a secure, multi-tenant architecture, enabling distinct customer organizations to operate within a single environment. This implementation was complemented by a comprehensive Role-Based Access Control (RBAC) system, ensuring precise and customizable permission settings for different user roles, enhancing security and user management efficiency.
- Developed a highly efficient and scalable distributed locking system utilizing MongoDB's monotonic write capabilities, ensuring robust data consistency and supporting atomic updates to MongoDB documents. Additionally, I crafted a distributed locking mechanism built upon Redis, specifically designed for managing concurrency in Spark jobs, thereby optimizing performance and reliability in distributed computing environments.
- Developed a versatile caching system designed to temporarily store results in memory, with Amazon S3 as a backup, enhancing data retrieval efficiency. This system was capable of displaying previous results to the user interface during the computation of new data. It seamlessly updated the cache with fresh results once available, accommodating the variable execution times of Spark queries.
- Created a comprehensive model service framework to manage the lifecycle of machine learning models within the platform. This service encompassed defining model metadata such as outcomes and relevant columns, and overseeing the training, testing, and prediction phases. It facilitated the entire workflow from data input through model execution in AWS Batch, to the storage of results within the platform. Additionally, it maintained a detailed record of each model execution, providing valuable data for historical trend analysis, monitoring model performance drift, and assessing patient risk predictions over time.
- Developed optimized Spark code to deliver high efficiency, integrating it with Lagom APIs for dynamic data processing. I engineered a system where every query was handled on an ad-hoc basis by a dedicated, always-on Spark cluster. Innovations included the implementation of custom Catalyst rules and the introduction of an advanced caching system, utilizing in-memory storage with S3 backup. Additionally, I designed a DAG (Directed Acyclic Graph) execution engine to deconstruct large Spark jobs

into manageable segments, significantly enhancing processing speed. This approach enabled the execution of complex dataframes, involving over 2000 joins on datasets with millions of rows and thousands of columns, in under five minutes. Continuous optimization led to achieving sub-second response times for certain queries, marking a significant milestone in query performance and efficiency.

- Led the conceptualization and implementation of a dynamic Python-powered dockerized model inference pipeline, harnessing the power of xgboost and scikit-learn frameworks. Through seamless integration with AWS infrastructure, enabled scalable and efficient execution of machine learning models, effectively compartmentalizing their functionality within the broader platform.
- Designed and developed a comprehensive Python API package, serving as the primary interface for platform users, primarily used through Jupyter Notebooks. This innovative solution was dynamically generated from the Lagom Scala API codebase, ensuring seamless integration and interoperability between the platform and local data science tools.
- Developed the system for code deployment, embodying a build-once, deploy-many architecture. This system featured a comprehensive tagging and version tracking mechanism for build artifacts, enabling efficient rollbacks, hotfixes, and the seamless promotion of stable builds across environments, such as moving validated developments from the development stage to internal testing without the need for artifact rebuilding.
- Contributed to the enhancement and maintenance of the DevOps and infrastructure framework, utilizing Terraform for cloud resource management and Vault for secure credential handling. Actively participated in managing four key environments: development, internal testing, AWS production, and GCP production.
- Led the successful porting of our entire product to Google Cloud Platform (GCP), in addition to its existing deployment on AWS. This strategic move enabled us to offer multi-cloud backend solutions to our customers, catering to diverse compliance requirements and preferences.
- Developed the initial frontend in React, establishing foundational designs that have been retained in the current live application, despite the subsequent formation of a dedicated UI/design team.
- Implemented LLMOps (large language model operations) for monitoring, deploying, and maintaining chat models for use with healthcare.

Solo Software Engineer

ATXware, LLC.

01/2015 - 09/2017

- Led the research and implementation of software solutions for business clients, focusing on rapid application development to create working prototypes that evolved into production-grade applications.
- Developed a web-based compounding pharmacy sales management platform that processed \$98MM in insurance claim transactions over two years.
- Created a platform for medical professionals to utilize call center services for Medicare's annual wellness exams, maintaining strict adherence to HIPAA/HiTech compliance.

Lead Software Engineer

Draker Inc. (Formerly Solar Power Technologies Inc.)

02/2011 - 01/2015

- Architected and implemented cloud-based horizontally scaled data storage, rollup, and analytics platform capable of handling billions of data points on commodity AWS instances.

- Designed and implemented the Intelligent Array analysis algorithms for automatically detecting problems within the solar array. This allows O&M teams to maximize power output and minimize downtime.
- A mixture of custom algorithms and AI systems would work together to determine issues such broken or damaged solar panels, to estimate how dirty the solar array might be and provide recommendations on if the array should be cleaned.
- Designed one of the world's largest scalable mesh network. The network is able to support 5000+ nodes per gateway. Largest node count in production was 3000 nodes. Implemented tools for debugging mesh operations.
- Architected and implemented data acquisition software. It supports collecting data from multiple protocols with a plugin system, such as modbus, proprietary serial communications, TCP/IP, SCADA systems, third party cloud providers, and more. This code runs on linux computers deployed in the field, and on cloud servers. It also supports limited SCADA-like operations.

Software Engineer

Media Research Labs, LLC.

08/2009 - 02/2011

- Responsible for analyzing and implementing software based solutions that increased workplace productivity and employee performance.
- Created a participant management system which handled participant signup, calling and mailing lists, resource scheduling, study management. The software intelligently created call lists of participants based on previous study history, study demographic requirements, and other dynamic variables.
- Created a cross platform (Windows/Linux/Mac) content distribution system to dynamically build playlists and push files asynchronously to all testing computers located in the lab.
- Created a cross platform (Windows/Linux/Mac) gift card tracking system. Using a card reader, the software would keep track of gift cards as they were allocated to participants.
- Created software to automate coding of survey data for research. The software compared text entered by participants about brand recognition and corrected for typos, misspellings, and common misconceptions. This saved the company over \$100,000/year in labor costs.
- Researched and deployed various eye tracking solutions including Tobii, FaceLAB, and MiraMetrix.
- Assisted with study research design & implementation, internal technical support & training, and other technical projects.

Education

Associates in Software Engineering

Texas State Technical College

Graduated Honors; President's List: Fall 2008; Dean's List: Spring 2007, Summer 2008

Material used in courses: Python, C#, VB, .NET, Java, C++, C, MS Access, SQL Server, MySQL, ODBC, Visual Studio, Eclipse, Excel, VMWare