RAYMOND HUANG

hello@raymondhuang.com
+1 (425) 443-2437
http://raymondhuang.com

TECHNICAL SKILLS

Backend:	Java & Spring, Ruby & Ruby on Rails, SQL, NoSQL
Frontend:	Angular 8 (TypeScript, HTML, SASS)
AWS:	DynamoDB, S3, SNS, SQS, Redshift, CloudFormation, CloudWatch, IAM, CodeBuild, CodeDeploy, CodePipeline, ElastiCache, EC2, EBS, ELB, CloudFront

WORK EXPERIENCE: AMAZON.COM

SEATTLE, WA, US — Software Developer Engineer I/II — AUGUST, 2014 - MARCH, 2019

I work for the transportation optimization team. I develop service oriented solutions to reduce transportation costs under various constraints employing techniques such as control theory and linear optimization. The services that I work with operate on a scale of 100k+ TPS with SLAs of 250ms.

Resource Optimization Engine

Existing system assigns carriers to individual shipments greedily and optimizes only for business constraints.

- Designed & implemented an optimization engine to solve for all shipments across the network using a linear optimization model that represents the transportation network, developed in conjunction with research scientists, using the proprietary linear optimization software, Xpress
- Computes resource opportunity costs for utilizing transportation capacity to influence carrier assignments
- Reduced memory consumption with GC tuning techniques, string deduplication, in-memory/disk/database caching, and careful management of new objects
- Solves 3 million shipments within 5 minutes
- Realizes yearly transportation cost savings of $200 million in North America

Optimization Engine Aware Simulations Workflow

The legacy simulations were slow and did not take into account the effects of the resource optimization engine.

- Designed & implemented a new simulation workflow leveraging AWS SWF that takes an input of shipments, replays each shipment, processes the shipment responses, sends them to the resource optimization engine, and processes the results
- Horizontally scalable by distributing shipments across hosts, solves 30 million shipments within 10 hours
- Generic API interfaces and components allow for extensibility and ease of on-boarding new use cases
- Worked extensively with the demand forecasting team to onboard, and improved the WAPE of the demand forecast by 5% while also reducing forecasting runtime by 30%

Realtime Replay Client

Required a way to test features and collect metrics in an environment that mimics production traffic.

- Built a client that listens to production messages via AWS SNS and SQS and replays each request against a parallel production-like environment
- Used historical data to estimate when shipments are fulfilled after materializing in a warehouse
- Implemented simplified versions of our clients to improve the accuracy of the replay client, since upstream API calls affect downstream requests
- Enables finance to collect cost savings of features by comparing data between production and the replay client

EDUCATION

WATERLOO, ON, CA — University of Waterloo: Bachelor of Software Engineering — 2009 - 2014