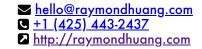
RAYMOND HUANG



TECHNICAL SKILLS

Languages, frameworks: Java, Spring, Ruby, Ruby on Rails, JavaScript, TypeScript, Angular 1/2

AWS (proficient): DynamoDB, S3, SNS, SQS, Redshift, CloudFormation, CloudWatch, IAM

AWS (familiar): CodeBuild, CodeDeploy, CodePipeline, ElastiCache, EC2, EBS, ELB, CloudFront

WORK EXPERIENCE: AMAZON.COM

SEATTLE, WA, US Software Developer Engineer II

2014 - NOW

2009 - 2014

I work for the <u>transportation optimization team</u>. I develop service oriented solutions to reduce transportation costs under various constraints employing techniques such as control theory and linear optimization.

Resource Optimization Engine

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

- Built an optimization engine to solve for all shipments across the network using a linear optimization model that
 models 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
- Realizes yearly transportation cost savings of \$200 million in North America

Optimization Engine Aware Simulations Workflow

Legacy simulation system was slow, and did not take into account the effects of the resource optimization engine.

- Built 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
- 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 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