Cloud Optimization: Gurobi & Cloud Foundry
Philip Glebow
Director
Pricing Architecture
Gap Inc.
Introduction

• Topic: Optimization using Gurobi and Cloud Foundry
• Assumptions made
• Not an endorsement
Overview

• What is Optimization?
• The Business Problem
• Why Cloud & Cloud Foundry?
• Integration Overview
• Organizational Issues
• Q&A
• Presenter Biography
• References
What is Optimization?

• Example: The Diet Problem
  – Select foods that meet nutritional requirements at the minimum cost

• Example: Facility Location
  – A warehouse supplies supermarkets
    • Reducing the distance to a market from a warehouse saves money
    • Opening a warehouse is costly though
  – What’s the optimal tradeoff between delivery and the cost of building new facilities?
Business Problem

• Localized Promotional Pricing
• A non-trivial optimization problem
• Initial results indicate we can meet the timing objective using 90 c4.4xlarge instances using the Gurobi Instant Cloud Product
Why Cloud?

• We can vary...
  – Cost
  – Performance
  – Hardware

• The cloud solution allows us to build an elastic, high-performance solution that can scale

• We don’t need to build and maintain a servers that sit idle for a large period of time each day
What is Cloud Foundry?

• An open-source Platform-As-A-Service (PaaS) which allows developers to easily build, deploy, run and scale applications

• Works best with 12-factor applications
  – See http://12factor.net for more detail

• See https://www.cloudfoundry.org for more detail
Advantages of Cloud Foundry

• Quick & easy to deploy
  – Simplification of CI/CD
• Quick to scale
• Infrastructure simplification
• Useful patterns
Integration Overview
Tech Stack

- Web Browser
- REST Services
- Spring Boot
- Spring Actuator
- Java
- MongoDB
- Rabbit MQ
- Gurobi
- Cloud Foundry
Integration Overview

• The optimization problem is expressed in Java using Gurobi’s API
• Spring Boot is the development framework
• RabbitMQ is used for concurrency, resilience and notification
• A set of REST services provide
  – Data to the model
  – Process control
  – A way to retrieve results
• A web-based UI allows users to accept or reject recommendations
• System is deployed to Cloud Foundry
• It uses a variety of data services
  – Data virtualization technology
  – NoSQL database to store results
Sequence Diagram: Create Cloud

1: Start
2: Create Infrastructure
3: Launch
4: MachineResponse
5: Store Machine Responses in Cyclic Iterator
6: [loop][nullLicenses]
7: Poll machines for completion
8: [loop][until Ready]
Sequence Diagram: Build Messages
Sequence Diagram: Process Messages
Organizational Issues

• Java is not the preferred language for data science / optimization
• Agile Development in a Waterfall Context
Biography of the Presenter

Philip Glebow is a software architect at Gap Inc. and is the product architect for pricing in the planning domain. Prior to joining Gap Inc., Mr. Glebow worked as a software architect in financial services for Blackrock, Inc. and Barclays. He began his career as a consultant where he worked on complex custom software solutions in the Information, Communications and Entertainment practice at BearingPoint (previously KPMG Consulting, Inc.). Mr. Glebow holds a Bachelor of Science in Computer Engineering from Cal Poly San Luis Obispo and a Masters of Science in Bioinformatics from the Johns Hopkins University. He may be reached via LinkedIn at www.linkedin.com/in/philipglebow or via email at pglebow@gmail.com.
References

• https://en.wikipedia.org/wiki/Linear_programming
• Gurobi - http://www.gurobi.com
• Interactive Facility Location Demo - http://examples.gurobi.com/facility-location
• Spring Boot - http://projects.spring.io/spring-boot
• Cloud Foundry - https://www.cloudfoundry.org
• The Twelve-Factor App - http://12factor.net
• Rabbit MQ - https://www.rabbitmq.com
• MongoDB - https://www.mongodb.org
Thank you!