Moneyball
Proactive Auto-Scaling in Azure SQL Serverless

Ojas Sethi
Introduction

Azure SQL Serverless

- Among the leading relational database service providers in the cloud
- Auto-scales compute resources on demand
- Pay only for what you utilize
Auto-scaling currently is only ‘reactive’
The Problem

Reactive Auto-Scaling

- Reactive scaling occurs in response to changes in the actual workload
- Relies on triggers or thresholds of performance in real-time
- Could cause a delay of resource availability
- Inefficient when demand is high
- Keeps costs low
Utilizes historical data and patterns to predict future resource requirement

Resources are pre-allocated based on predictions

**Efficient in keeping up with demand**

Reduced latency

Costs can vary
Proactive Auto-Scalability

Challenges

- Large search space of tunable parameters
- Opposing optimization objectives
- Changed resource usage patterns
Moneyball Problem

Proposed Solution

Find a middle-ground somewhere between the contradictory goals of enabling proactive resume, while also reducing the number of short pauses.
Objective 1
Enabling Proactive Resume

- Probabilistic Resume
- Predictive Resume

Source: Poppe et al. 2022
Objective 1
Probabilistic vs Predictive Resume

(a) Proactive resumes: Probabilistic resume (P) vs predictive resume (N)
(b) Benefited databases
(c) Resume cost index

Source: Poppe et al. 2022
Objective 2
Avoid Ineffective Pauses

- Budgeting Algorithms
- Logical Pause-Based Algorithms
Objective 2
Avoid Ineffective Pauses - Budgeting Algorithms

- Greedy Budget
- Predictive Budget

Source: Poppe et al. 2022
Objective 2
Avoid Ineffective Pauses - Logical Pause Based Algorithms

- Greedy Logical Pause
- Predictive Logical Pause

Source: Poppe et al. 2022
Combining The Components

The Model

Source: Poppe et al. 2022
Combining The Components

Results - The Moneyball Problem Space

Source: Poppe et al. 2022
Summary

- Proactive auto-scaling reduces delays in resource availability
- Leverage machine learning methods to predict pause and resume patterns
- Avoid short pauses by logically pausing a database
- Predictive resume & greedy logical pause is the way to go for now
Thank you!