

Coding for Service Capacity Problems

Austin Allen ¹ Dr. Emina Soljanin ²

¹Carnegie Mellon University

²Rutgers University

June 3rd, 2019

An Analogy

- Assume that you really like ice cream
- It's a hot day and Busch Student Center has pints of ice cream
- Busch has four freezers
- Freezer 1 (10 pints chocolate)
- Freezer 2 (10 pints chocolate)
- Freezer 3 (10 pints vanilla)
- Freezer 4 (10 pints vanilla)

An Analogy (cont.)

- Freezers can only hold 10 pints each
- Some people will be lucky, others won't
- This will lead to angry customers!

An Observation

- There is an issue with this system
- What if one flavor is more popular than the other?
- Is there a way to account for this demand?

What's the purpose?

There is always demand for storage systems that are reliable and available (latency and service capacity) to handle large amounts of data:

- Amazon (could lose sales)
- Google (page views can drop)
- Gaming (ping could be really high)
- Virtual Reality (not realistic enough)

Service Capacity Problem

Consider the following model:

- We want to store k files f_1, f_2, \dots, f_k redundantly across n servers
- Each file is of the same size and each server stores the same amount of data
- Requests to download file f_i arrive at a rate of λ_i

This problem is similar to the first ice cream analogy!

Coding for Service Capacity Problems

Rather than figuring out the rates that the system can handle, we can consider the following problem:

- Suppose there are k files that we want to store
- We know the arrival rates $\lambda_1, \dots, \lambda_k$ for each file
- What is the best coding scheme to use to store the data?

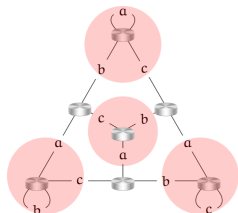
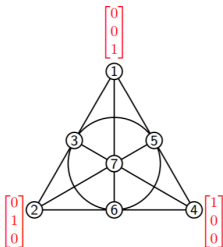
This problem is similar to the second ice cream analogy!

Our goal and how we might solve it

Goal: Given the service capacity region of some distributed storage system, what is the most "efficient" redundancy scheme we can apply?

Suggestions:

- Use Batch Codes
- Use private information retrieval (PIR) codes
- Use simplex codes
- Maybe the problem is already solved!



Acknowledgements

A special thanks to the National Science Foundation for supporting this research through the NSF grant CCF-1852215.

