Codes, Graphs & Hyperplanes – Emina Soljanin, Rutgers

... in Distributed Data Access Service

Collaborators in this general area:

M. Aktaş, S. Andersen, A. Johnston, G. Joshi, S. Kadhe, E. Karimi,
F. Kazemi, S. Kurz, G. Matthews, C. Mayer, A. Sprintson, G. Yadgar

Distributed Service Model

There are n nodes providing service to multiple concurrent users, e.g., cloud edge nodes providing streaming, download, computing.

We distinguish between two functional components at each node: one for data storage and the other for service request processing.

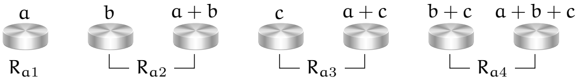
Data Storage Model

Simple Redundant Storage

- k equal size data objects are stored across n nodes $(k \leqslant n)$.
- Data objects are represented as strings of bits.
- All servers have a storage capacity of one data object.
- Each server stores an object or an XOR of two or more objects.

$\Longrightarrow$ A data object can be recovered from multiple sets of coded objects.

Example: Data objects a, b, and c stored across $n = 7$ nodes:

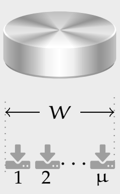


$\Longrightarrow$ a can be recovered from any of the sets $R_{a1}, R_{a2}, R_{a3}, R_{a4}$.

Data Service and Request Models

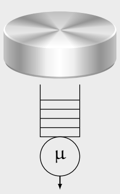
Different practical service models are mathematically equivalent.

We consider the bandwidth and the queuing model:



stores one data object

Server's bandwidth W can accomodate up to μ users.



stores one data object

Users queue for download.

Download is done at rate μ.

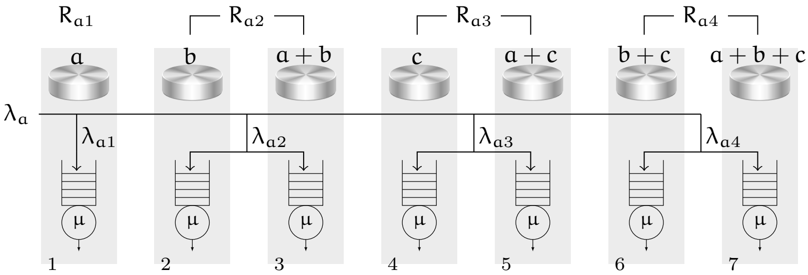
Requests for objects i, $i \in \{1, \ldots, k\}$:

- In the queuing model, requests for object i arrive at rate λ_i.
- In the bandwidth model, the number of requests for object i is λ_i

Distributed Service Model – An Example

λ_a is the request rate (demand) for object a

λ_{aj} is the portion of λ_a assigned to the recovery set R_{aj}, $j \in \{1, 2, 3, 4\}$.



$\{\lambda_{a1}, \lambda_{a2}, \lambda_{a3}, \lambda_{a4}\}$ is a request allocation for λ_a.

Which request vectors $(\lambda_a, \lambda_b, \lambda_c)$ can be serviced by the system?

Service Rate Region

Set of vectors $(\lambda_1, \ldots, \lambda_k)$ that can be served by the system

λ_i is the request rate (demand) for object i, $i = 1, \ldots, k$.
λ_{ij} is the portion of λ_i assigned to the recovery set R_{ij}, $j = 1, \ldots, t_i$.

The request vector $(\lambda_1, \ldots, \lambda_k)$ can be serviced by the system iff there exist λ_{ij} satisfying the following constraints:

1. No server is assigned requests in excess of its service rate:

$$\sum_{i=1}^{k} \sum_{\substack{1 \leqslant j \leqslant t_i \\ \ell \in R_{ij}}} \lambda_{ij} \leqslant \mu \quad \text{for} \quad 1 \leqslant \ell \leqslant n.$$

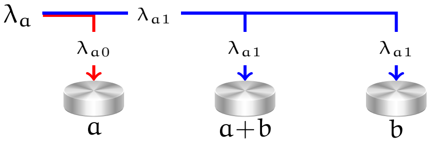
2. All objects' requests are served: $\sum_{j=1}^{t_i} \lambda_{ij} = \lambda_i$ for $1 \leqslant i \leqslant k$

$\{\lambda_{ij} : 1 \leqslant i \leqslant k, 1 \leqslant j \leqslant t_i\}$ as a request allocation for $(\lambda_1, \ldots, \lambda_k)$.

If we require that λ_{ij} be either 0 or μ, we speak of integral service rates.

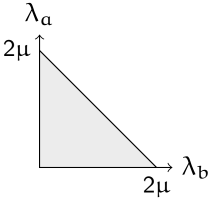
Service Rates for Storage Scheme $[a \;\; b] \rightarrow [a \;\; b \;\; a+b]$

How can requests λ_a be served when $\lambda_b = 0$?

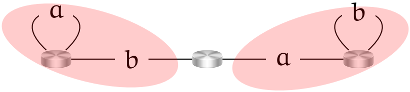


$\implies \lambda_a \leqslant 2\mu$ is achevable.

Service rate region



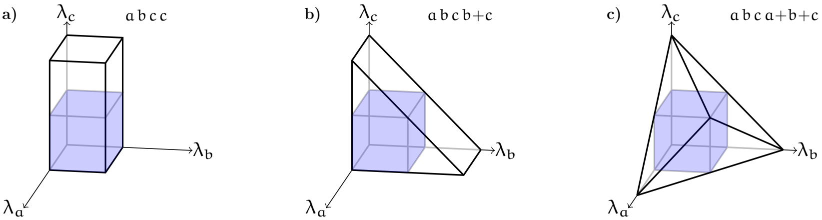
Converse:



$\implies \lambda_a + \lambda_b \leqslant 2\mu$

Three Storage Schemes and Their Service Rates

$k = 3$ data objects stored across $n = 4$ nodes



Many (kinds of) questions are of interest.

"Covering" a Request Region

Requests: $\lambda_a \sim \mathcal{N}^+(4,4)$ and $\lambda_b \sim \mathcal{N}^+(8,8)$ and vice versa.

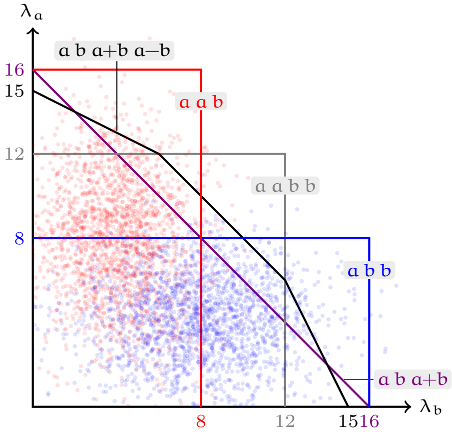
Two systems with equal total service bandwidth, storing $k = 2$ objects.

System 1: $n = 3$ with $\mu = 8$ with codes

$[a, a, b]$ $[a, b, b]$ $[a, b, a{+}b]$

System 2: $n = 4$ with $\mu = 6$ with codes

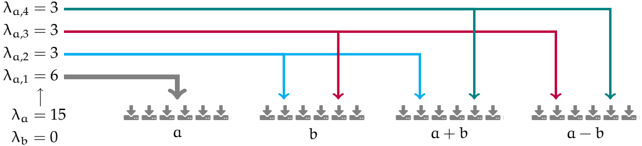
$[a, a, b, b]$ $[a, b, a{+}b, a{-}b]$



Request coverage: 0.7366 for $[a, a, b]$ & $[a, b, b]$, 0.8727 for $[a, b, a{+}b]$

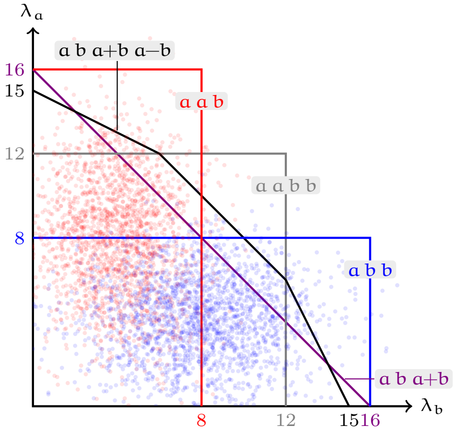
0.9211 for $[a, a, b, b]$, and 0.9434 $[a, b, a{+}b, a{-}b]$.

Service allocation for $(\lambda_a, \lambda_b) = (15, 0)$



Code:

with node capacity $\mu = 6$.



Service Rate Region Problem(s) Formulation

System Model:

- k data objects are stored redundantly across n nodes.
- Data objects are represented as elements of some finite field.
- Each server stores a linear combination of data objects, i.e., a coded object of the same size (same field).
- Requests for object i, $i \in \{1, \ldots, k\}$ arrive to the system at rate λ_i.
- At each node, requests are serviced at rate $\mu = 1$.

SOME OBJECTIVES:

1. Determine the set of rates $(\lambda_1, \ldots, \lambda_k)$ that can be supported by the system implementing some common redundancy scheme.
2. Design a redundancy scheme in order to maximize and/or shape the of region of supported arrival rates under some limited resources.
3. Evaluate the system's performance for a given stochastic model of $(\lambda_1, \ldots, \lambda_k)$ (e.g., probability of supported rates, load imbalance).

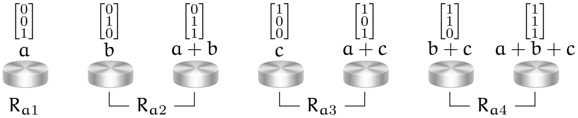
Finding Service Rate Region

Codes and Matrices

We define a code by a $k \times n$ generator matrix G over $\mathbb{F}_q$

$k < n$ & columns of G include all standard bases vectors of $\mathbb{F}_q^k$.

Example: Storage scheme



is defined by matrix $G = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$ in the sense that

$$[a \;\; b \;\; c] \cdot G = [a \;\; b \;\; c \;\; a{+}b \;\; b{+}c \;\; a{+}c \;\; a{+}b{+}c]$$

This redundancy scheme is known as $[7, 3]$ Simplex code.

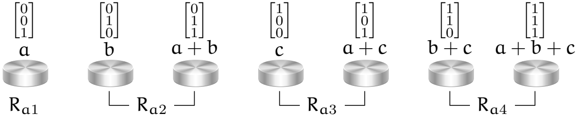
Recovery Sets

Subset R of columns in G is a recovery set of basis vector a if

- $a \in \text{span}(R)$
- $S \subset R \implies a \notin \text{span}(S)$

Example:

Recovery sets of size one and two for a in $G = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$



Coding theorists refer to a, b, c as systematic columns or data symbols.

A Recovery Graph for $[n, k]$ Code

Consider a code with the generator matrix G and size 2 recovery sets.

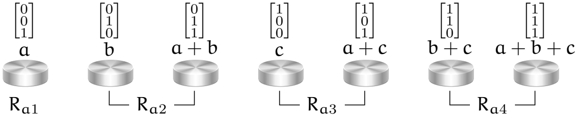
(e.g., simplex and $k = 2$ codes; generalization to any linear code is straightforward)

We define recovery graph Γ as follows:

- Γ has n nodes corresponding to the columns of G, and an additional node is added for each systematic column.
- If two nodes correspond to a recovery set of data symbol x, they are connected by an edge which is given label x.

Example:

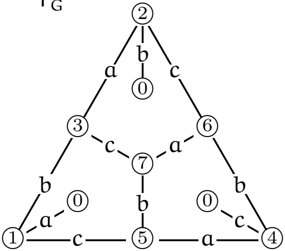
The nodes and some edges in the $[7, 4]$ Simplex code recovery graph

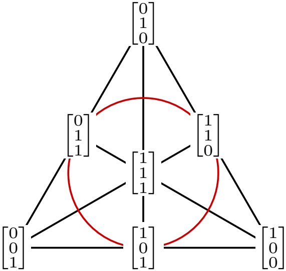


The Recovery Graph for the [7, 3] Simplex Code

$$G = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Γ_G





Matching Notions & Service Rates on Recovery Graphs

A fractional matching of Γ_G assigns non-negative weights to its edges s.t. for each node, the sum of weights of its incident edges does not exceed 1.

An integral matching of Γ_G assigns 0 or 1 weights to its edges s.t. for each node, the sum of weights of its incident edges does not exceed 1.

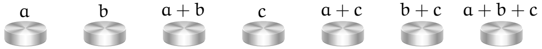
We define λ_x^M, the service rate for data symbol x in matching M, as the sum of the weights that M assigns to x-labeled edges in Γ_G.

Claim: $(\lambda_1, \lambda_2, \ldots, \lambda_k)$ is in the service rate region of G iff there is a matching M in Γ_G s.t. $(\lambda_1, \lambda_2, \ldots, \lambda_k) = (\lambda_1^M, \lambda_2^M, \ldots, \lambda_k^M)$

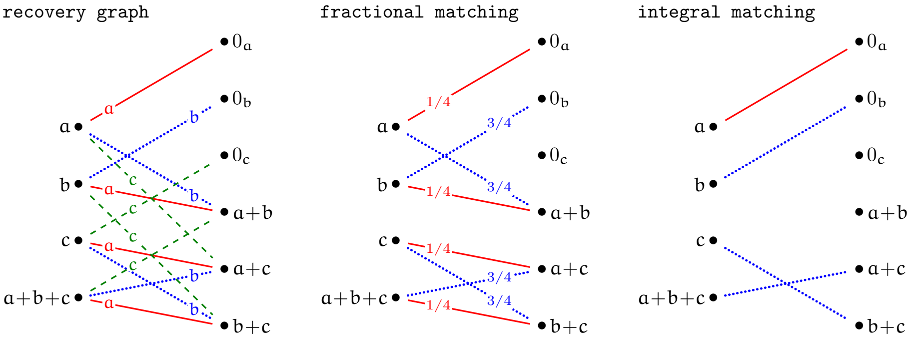
How is this claim helpful in characterize the set of all $(\lambda_1, \lambda_2, \ldots, \lambda_k)$?

Observe that max $\sum_{i=1}^{k} \lambda_i$ is the (fractional) matching number of Γ_G.

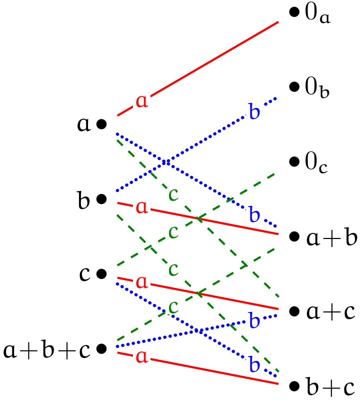
Serving $(\lambda_a, \lambda_b, \lambda_c) = (1, 3, 0)$ with the [7, 3] Simplex Code



Consider two matchings with identical service rates:



Fractional vs. Integral Service



Consider a fractional matching s.t.

- λ_a is the sum of a edge weights.
- λ_b is the sum of b edge weights.
- λ_c is the sum of c edge weights.

$\implies \lambda_a + \lambda_b + \lambda_c \leqslant 4$

Q: If $\lambda_a, \lambda_b, \lambda_c$ are integers, is there always an integral matching with λ_a a-edges, λ_b b-edges, λ_c c-edges? – **a new matching problem**.

A Bound on the Sum of Service Rates

by using well known graph theory results

A vertex cover of a graph Γ is a set of vertices in Γ such that each edge in Γ is incident to at least one vertex in the set.

- Consider a system using an $[n, k]$ code with a generator matrix G.
- Let Γ_G be the recovery graph of G.

$\Longrightarrow$

The sum of rates in any request vector $(\lambda_1, \cdots, \lambda_k)$ that can be served by the system cannot exceed the number of vertices in a vertex cover of Γ_G.

Binary Simplex Codes and their Recovery Graphs

aka Hadamard Codes is CS literature

G_k consist of all distinct nonzero vectors of $\mathbb{F}_2^k$.
$\implies$ Γ_k vertices are labeled by k-bit stings.

Lemma: Structure of the recovery graph Γ_k:

1. Γ_k is bipartite.
2. Each vertex of Γ_k has degree k where each edge is labeled by a different data symbol.
3. The 2^{k-1} vertices of Γ_k that correspond to the odd weight columns of G_k form a minimum vertex cover of Γ_k.

Service Rate Region $[2^k - 1, k]$ Simplex Code

Theorem: Simplex, again!

$\lambda_1, \lambda_2 \ldots, \lambda_k$ can be service rates for the $[2^k - 1, k]$ Simplex code iff $\lambda_1 + \lambda_2 + \cdots + \lambda_k \leqslant 2^{k-1}$.

Proof Sketch for the Achevability:

Rates $\lambda_1, \ldots, \lambda_k$ s.t. $\lambda_1 + \cdots + \lambda_k \leqslant 2^{k-1}$ can be achieved by the fractional matching that assigns weight $\lambda_i / 2^{k-1}$ to each i labeled edge.

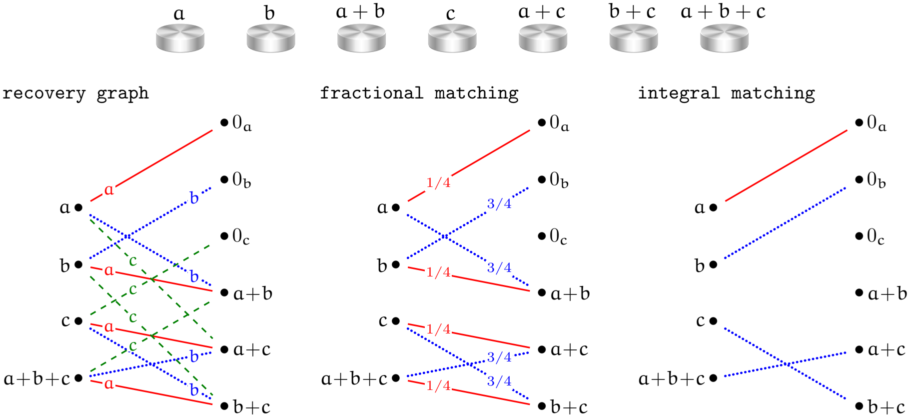
Proof Sketch for the Converse:

For bipartite graphs, the size of the minimum vertex cover (here 2^{k-1}) is equal to the (fractional) matching number.

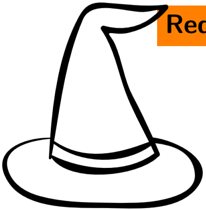
Asynchronous Service Rate Region

Asynchronous Batch Codes by Riet, Skachek, and Thomas

Consider the simplex code and two ways to satisfy demand $(1, 3, 0)$:



Q: If some users leave the system, can others use the freed resources?



Redundancy Scheme Design

We started with a matrix ...

G is a $k \times n$ matrix over $\mathbb{F}_q$

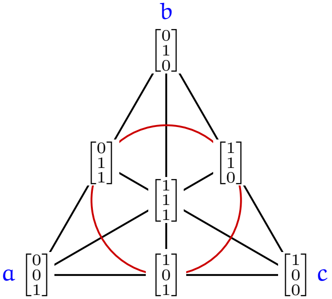
$k < n$ & columns of G include all standard bases vectors of $\mathbb{F}_q^k$.

Columns of G are a multi-set $\mathcal{G}$ of points in $\mathbb{PG}(k-1, q)$

We refer to $\mathcal{G}$ as the ground set of G.

Example:

$$G = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$



For coding theorists, G is a generator matrix of a systematic $[n, k]_q$ code.

A Geometric Bound

Theorem: What is $\mathcal{H}$ to Γ?

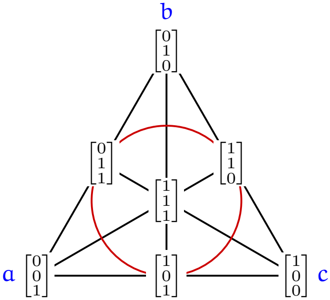
For an $[n, k]_q$ code with ground set $\mathcal{G}$ in $\mathbb{PG}(k-1, q)$ and a vector of achievable rates $(\lambda_1, \lambda_2, \cdots, \lambda_k)$, it holds that

$$\lambda_1 + \lambda_2 + \cdots + \lambda_k \leqslant |\mathcal{G} \setminus \mathcal{H}|$$

where $\mathcal{H}$ is a hyperplane not containing any standard basis vectors.

Example:

$$G = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$



What is $\mathcal{H}$ to Γ?

The points in $\mathcal{G} \cap \mathcal{H}$ correspond to an independent set in Γ.

$\implies$ The points in $\mathcal{G} \setminus \mathcal{H}$ constitute a vertex cover of Γ.

$\implies$ $|\mathcal{G} \setminus \mathcal{H}|$ is an upper bound to $\nu(\Gamma)$ (the matching number of Γ).

$\implies$

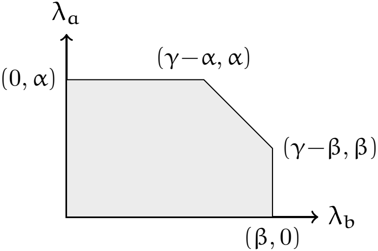
$$\lambda_1 + \lambda_2 + \cdots + \lambda_k \leqslant \nu(\Gamma) \leqslant \mathcal{G}| \setminus \mathcal{H}|$$

How far does this similarity go?

Is there is an encompassing view, e.g., based on matroids?

Covering a Region with Minimal Storage

We need to serve requests in the region $\lambda_a \leqslant \alpha$, $\lambda_b \leqslant \beta$, $\lambda_a + \lambda_b \leqslant \gamma$.



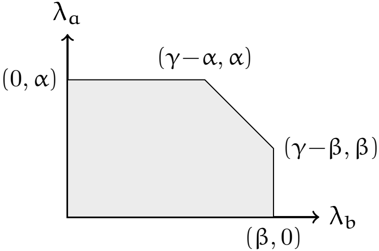
The columns of the generator matrix can only be $\begin{bmatrix}1\\0\end{bmatrix}$, $\begin{bmatrix}0\\1\end{bmatrix}$, and $\begin{bmatrix}1\\1\end{bmatrix}$.

$$\underbrace{\begin{bmatrix}1\\0\end{bmatrix}\ldots\begin{bmatrix}1\\0\end{bmatrix}}_{n_{\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]}} \quad \underbrace{\begin{bmatrix}0\\1\end{bmatrix}\ldots\begin{bmatrix}0\\1\end{bmatrix}}_{n_{\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]}} \quad \underbrace{\begin{bmatrix}1\\1\end{bmatrix}\ldots\begin{bmatrix}1\\1\end{bmatrix}}_{n_{\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]}}$$

Find $n_{\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]}, n_{\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]}, n_{\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]}$ that minimize $n = n_{\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]} + n_{\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]} + n_{\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]}$.

Covering a Region with Minimal Storage

We need to serve requests in the region $\lambda_a \leqslant \alpha$, $\lambda_b \leqslant \beta$, $\lambda_a + \lambda_b \leqslant \gamma$.
What is the minimal number of servers n for a binary storage scheme?



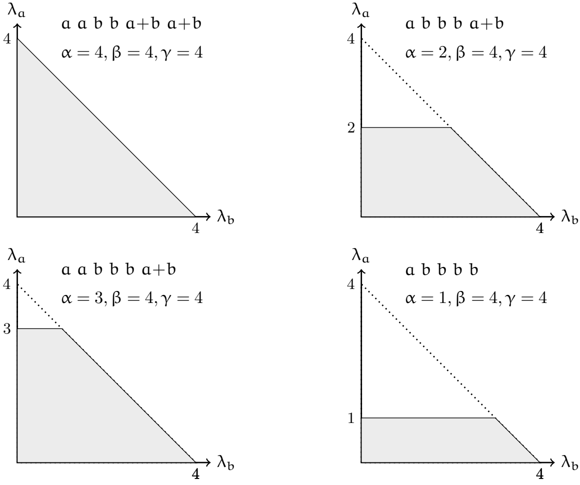
The columns of the generator matrix can only be $\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]$. $\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]$, and $\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]$.
Their multiplicities satisfy the following achievable bounds:

$$n_{\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]} + n_{\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]} \geqslant \alpha, \quad n_{\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]} + n_{\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]} \geqslant \beta, \quad n_{\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]} + n_{\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]} \geqslant \gamma$$

$$\implies \quad n = n_{\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]} + n_{\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]} + n_{\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]} \geqslant (\alpha+\beta+\gamma)/2.$$

Covering a Region with Minimal Storage – Examples

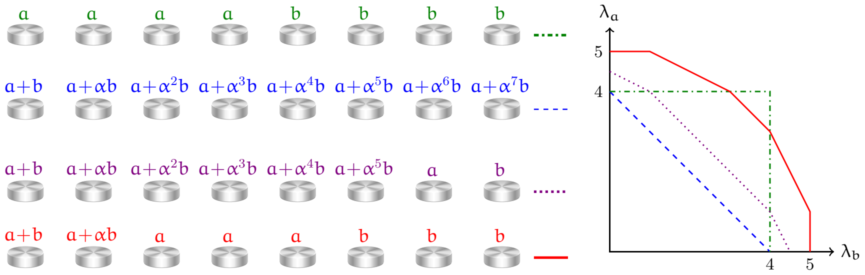
What is the minimal number of servers and the redundancy scheme that satisfy the demand described by $\lambda_a \leqslant \alpha$, $\lambda_b \leqslant \beta$, $\lambda_a + \lambda_b \leqslant \gamma$?



Maximizing Service Rate Region with Fixed Resources

How should we store k objects on n servers?

For $k = 2$, we can have A nodes storing a, B storing b, and C coded nodes.



- Combining coding and replication is beneficial in multiple ways.
- Service rate region depends on the generator matrix of the code.

An Early Theorem for any Two-Object System

$k = 2$, **data objects** a **and** b A nodes storing a, B storing b, & C coded nodes (MDS)

$$\underbrace{\begin{bmatrix}1\\0\end{bmatrix}\dots\begin{bmatrix}1\\0\end{bmatrix}}_{A} \text{ and } \underbrace{\begin{bmatrix}0\\1\end{bmatrix}\dots\begin{bmatrix}0\\1\end{bmatrix}}_{B} \text{ and } \begin{bmatrix}1\\ \alpha^j\end{bmatrix} \quad j = 0, \dots, C-1.$$

Then the service rate region is bounded by

$\lambda_a = 0$, $\lambda_b = 0$, $\lambda_a = \min\{(A+C)\mu,\ (A + \frac{B}{2} + \frac{C}{2})\mu\}$, and

$$L(\lambda_a) = \begin{cases} (B+C)\mu & \text{if } A > C \text{ and } 0 \leqslant \lambda_a \leqslant (A-C)\mu \\ -\frac{1}{2}\lambda_a + (\frac{A}{2} + B + \frac{C}{2})\mu & \text{if } A > C \text{ and } (A-C)\mu < \lambda_a \leqslant A\mu \\ -\frac{1}{2}\lambda_a + (\frac{A}{2} + B + \frac{C}{2})\mu & \text{if } A \leqslant C \text{ and } 0 \leqslant \lambda_a \leqslant A\mu \\ -\lambda_a + (A + B + \frac{C}{2})\mu & \text{if } A\mu < \lambda_a \leqslant (A + \frac{C}{2})\mu \\ -2\lambda_a + (2A + B + C)\mu & \text{if } B > C \text{ and } (A + \frac{C}{2})\mu < \lambda_a \leqslant A + C \\ -2\lambda_a + (2A + B + C)\mu & \text{if } B \leqslant C \text{ and } (A + \frac{C}{2})\mu < \lambda_a \leqslant (A + \frac{B}{2} + \frac{C}{2})\mu. \end{cases}$$

We only have an algorithm for $k = 3$.

Covered Requests, Server Utilization, Load (Im)balance

Requests: $\lambda_a \sim \mathcal{N}^+(4,4)$ and $\lambda_b \sim \mathcal{N}^+(8,8)$ and vice versa.

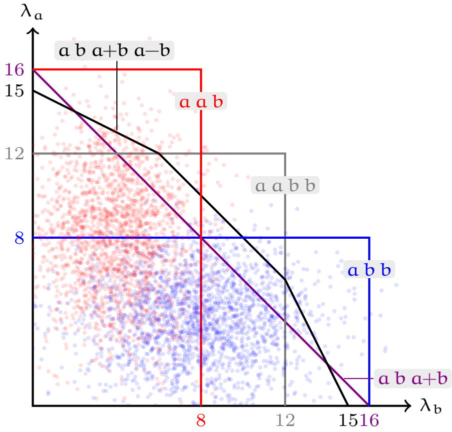
Two systems with equal total service bandwidth, storing $k = 2$ objects.

System 1: $n = 3$ with $\mu = 8$ with codes

$[a, a, b]$ $[a, b, b]$ $[a, b, a{+}b]$

System 2: $n = 4$ with $\mu = 6$ with codes

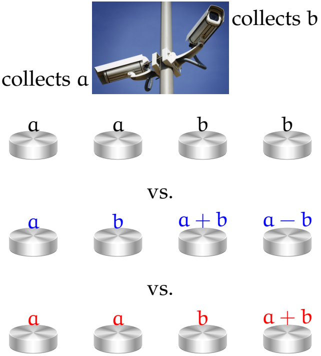
$[a, a, b, b]$ $[a, b, a{+}b, a{-}b]$

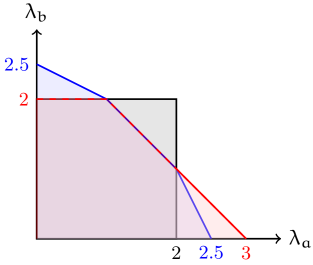


Request coverage: 0.7366 for $[a, a, b]$ & $[a, b, b]$, 0.8727 for $[a, b, a{+}b]$
0.9211 for $[a, a, b, b]$, and 0.9434 $[a, b, a{+}b, a{-}b]$.

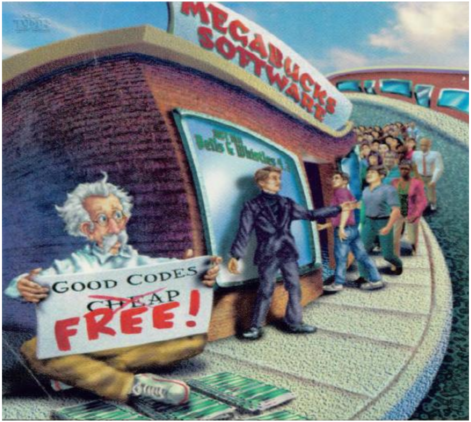
Service Rates of Codes

New applications create new performance metrics for codes, and we need to design new codes and solve new problems.





Needs for Services of Coding Theorists Go On



For more info, see https://arxiv.org/abs/2009.01598.

NSF Award # 2122400: Service Rates of Codes