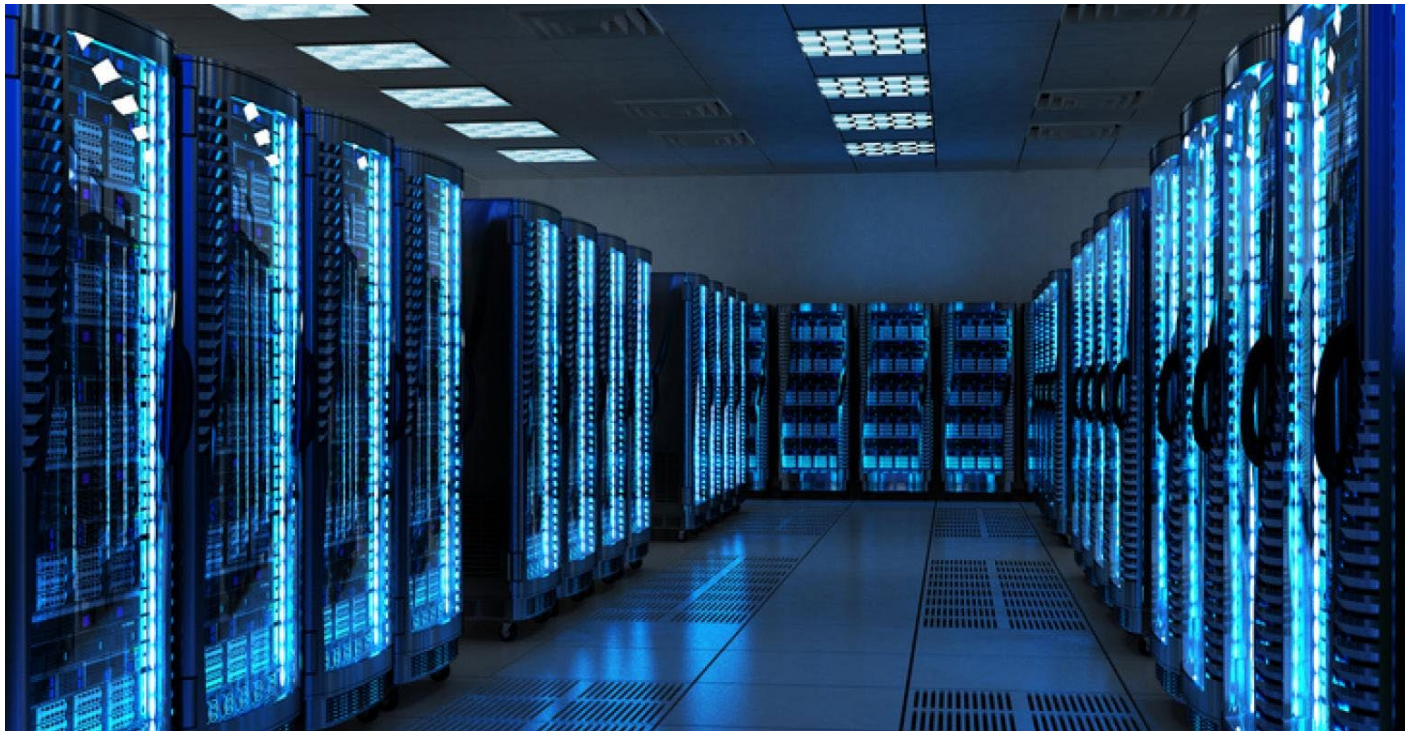
Data Structures
Programming Project #1

Data Center

- A data center consists of multiple severs
- The servers are connected by switches in a local area network



Servers in Data Centers

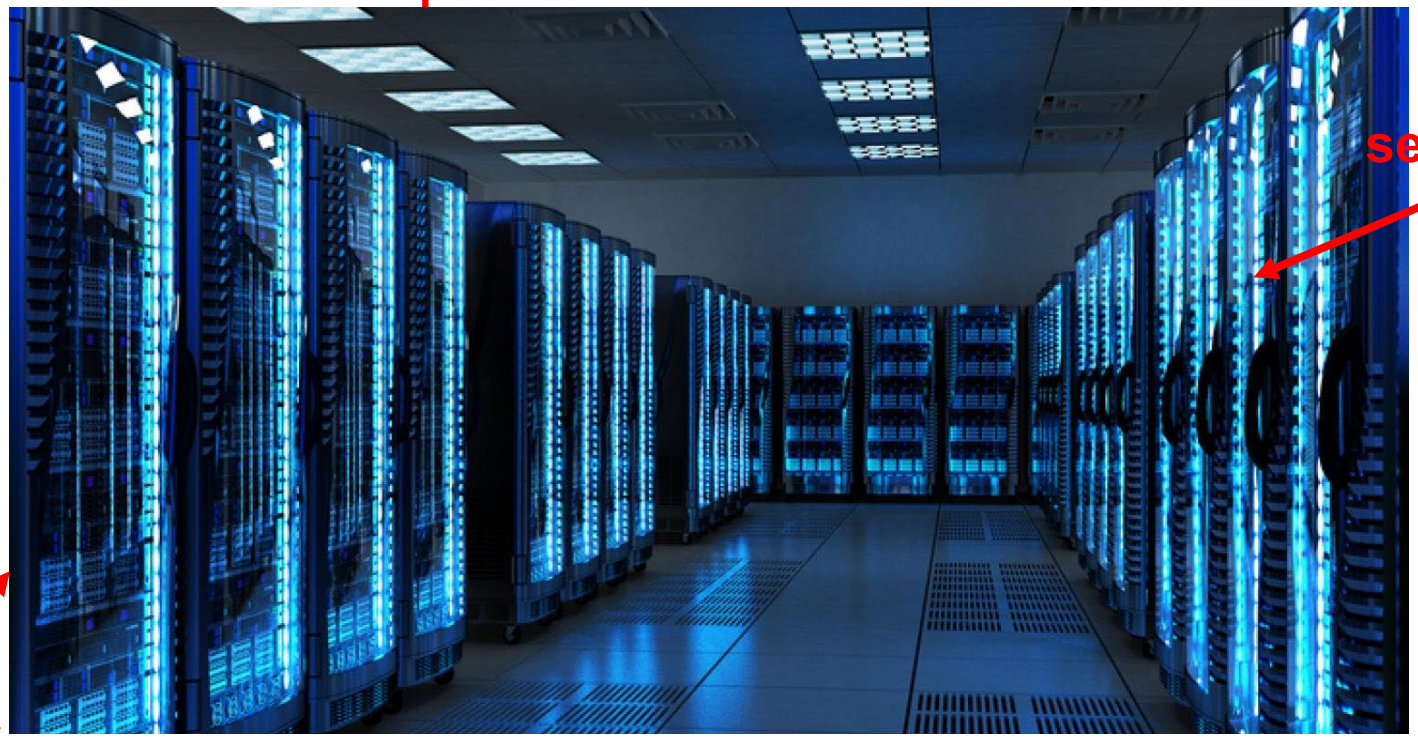
- Rack servers and blade serve
- Pros and cons

http://techgenix.com/tower-vs-rack-vs-blade-servers/

https://www.racksolutions.com/news/data-center-optimization/blade-server-vs-rack-server/

Data Availability

- System upgrade, power failure, system crash,…
- Data cannot be stored in one server only
- We need backup schemes to maintain availability

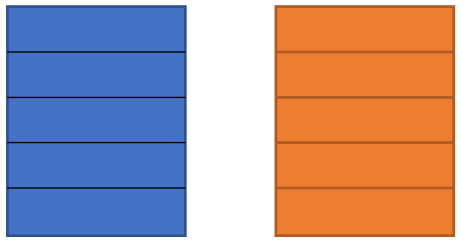


Data Availability

- Backup is necessary to prevent data loss
- A simple way is to make copies (mirroring)
- Availability rate = 90%
- Make 3 copies → $1 - (1 - 0.9)^3 = 99.9\%$
- Redundancy rate = $2/1 = 200\%$

Data Availability – Erasure Code

- Erasure code can mitigate the storage cost
- Availability rate = 90%
- Divide data into n chunks
- Make $(n + m)$ chunks
- Any n chunks can reconstruct the data
- Make $(5 + 5)$ chunks
- $1 - \sum_{i=6}^{10} C_i^{10}(0.1)^i(1 - 0.9)^{10-i}$
  $\approx 1 - 0.00015 = 99.9985\%$
- Redundancy rate $= 5/5 = 100\%$

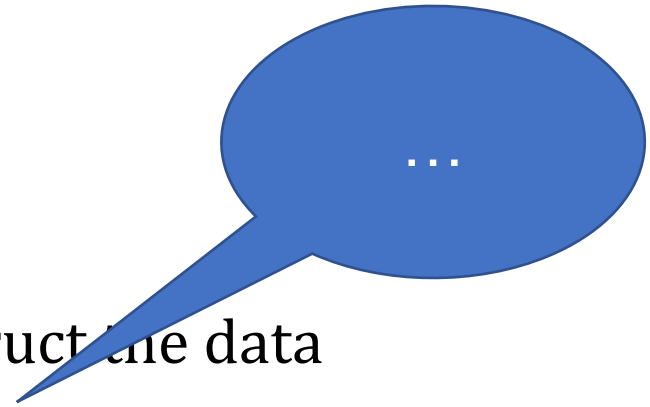


Programming Project #1: Design an erasure-code backup scheme

- Input:
  - Data, n, m
- Procedure:
  - Divide data into n chunks
  - Generate m chunks for recovery
- Output:
  - $(n + m)$ chunks
  - Ensure any n chunks can reconstruct the data
- The grade is inversely proportional to the redundancy rate

Programming Project #1: Design an erasure-code backup scheme

- Input:
  - Data, n, m
- Procedure:
  - Divide data into n chunks
  - Generate m chunks for recovery
- Output:
  - $(n + m)$ chunks
  - Ensure any n chunks can reconstruct the data
- The grade is inversely proportional to the redundancy rate



Programming Project #1: Design an erasure-code backup scheme

- Input:
  - Data, n, m
- Procedure:
  - Divide data into n chunks
  - Generate m chunks for recovery
- Output:
  - $(n + m)$ chunks
  - Ensure any n chunks can reconstruct the data
- The grade is inversely proportional to the redundancy rate

Programming Project #1: Design an erasure-code backup scheme

- Input:
  - Data, n, m
- Procedure:
  - Divide data into n chunks
  - Generate m chunks for recovery
- Output:
  - $(n + m)$ chunks
  - Ensure any n chunks can reconstruct the data
- The grade is inversely proportional to the redundancy rate

Programming Project #1: Design an erasure-code backup scheme

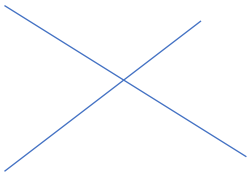
- Input:
  - Data, n, m
- Procedure:
  - Divide data into n chunks
  - Generate m chunks for recovery
- Output:
  - $(n + m)$ chunks
  - Ensure any n chunks can reconstruct the data
- The grade is inversely proportional to the redundancy rate

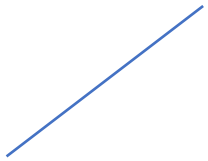
Programming Project #1: Design an erasure-code backup scheme

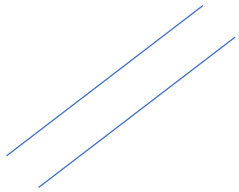
- Input:
  - Data, n, m
- Procedure:
  - Divide data into n chunks
  - Generate m chunks for recovery
- Output:
  - $(n + m)$ chunks
  - Ensure any n chunks can reconstruct the data
- The grade is inversely proportional to the redundancy rate

The Idea from System of Linear Equations

- n linear equations with n variables
- If the equations are linear independent, the variables can be solved uniquely
- $\begin{cases} x + y = 4 \\ x + 2y = 6 \end{cases}$
- $\begin{cases} x + y = 4 \\ 2x + 2y = 8 \end{cases}$
- $\begin{cases} x + y = 4 \\ 2x + 2y = 9 \end{cases}$







A Simple Erasure-Code Backup Scheme Encode

- Data is divided into n chunks (i.e., variables)
- Data $= 22, n = 2$
- $\begin{cases} x = 2 \\ y = 2 \end{cases}$

- Generate m chunks (i.e., use additional equations)
- $m = 2$
- $\begin{cases} 2x + y = 6 \\ x + y = 4 \end{cases}$

How to Generate the m Equations

- Generate m chunks (i.e., use additional equations)
- Use the Vandermonde matrix to generate

$$V = \begin{bmatrix} 1 & \alpha_1 & \alpha_1^2 & \dots & \alpha_1^{n-1} \\ 1 & \alpha_2 & \alpha_2^2 & \dots & \alpha_2^{n-1} \\ 1 & \alpha_3 & \alpha_3^2 & \dots & \alpha_3^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_m & \alpha_m^2 & \dots & \alpha_m^{n-1} \end{bmatrix}$$

How to Generate the m Equations

- Generate m chunks (i.e., use additional equations)
- Use the Vandermonde matrix to generate

$$V = \begin{bmatrix} 1 & 1 & 1^2 & \dots & 1^{n-1} \\ 1 & 2 & 2^2 & \dots & 2^{n-1} \\ 1 & 3 & 3^2 & \dots & 3^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & m & m^2 & \dots & m^{n-1} \end{bmatrix}$$

- $n = 2, m = 2$
- ??

How to Generate the m Equations

- Generate m chunks (i.e., use additional equations)
- Use the Vandermonde matrix to generate

$$V = \begin{bmatrix} 1 & 1 & 1^2 & \dots & 1^{n-1} \\ 1 & 2 & 2^2 & \dots & 2^{n-1} \\ 1 & 3 & 3^2 & \dots & 3^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & m & m^2 & \dots & m^{n-1} \end{bmatrix}$$

- $n = 2, m = 2$
- ??

How to Generate the m Equations

- Generate m chunks (i.e., use additional equations)
- Use the Vandermonde matrix to generate

$$V = \begin{bmatrix} 1 & 1 & 1^2 & \dots & 1^{n-1} \\ 1 & 2 & 2^2 & \dots & 2^{n-1} \\ 1 & 3 & 3^2 & \dots & 3^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & m & m^2 & \dots & m^{n-1} \end{bmatrix}$$

- $n = 2, m = 2$
- $\begin{cases} x + y = 4 \\ x + 2y = 6 \end{cases}$

A Simple Erasure-Code Backup Scheme Decode

- Get n chunks from $(n + m)$ chunks
- $n = 2$
- $\begin{cases} x + y = 4 \\ x + 2y = 6 \end{cases}$

- Reconstruct the data
- Use Gauss-Jordan Elimination or Inverse Matrix
- $\begin{cases} x = 2 \\ y = 2 \end{cases}$
- The data $= 22$

Programming Project #1: Implement a well-known erasure-code backup scheme

- Input:
  - Mode: $0 =$ Encode
  - Encode: data, n, m
- Encode procedure:
  - Divide data into n chunks
  - Generate m chunks for recovery
- Output:
  - $(n + m)$ chunks

- Input:
  - Mode: $1 =$ Decode
  - Decode: n equtions
- Encode procedure:
  - Solve the n equations to reconstruct the data
- Output:
  - The reconstructed data

Input Sample: use scanf

Format:

Mode n m
InputString

Exampe:

0 2 2
22

Format:

Mode n
n_Equations

Exampe:

1 2
1 1 4
1 2 6

Output Sample: use printf

Format:

$(n + m)$_Equations

Example:

1 0 2
0 1 2
1 1 4
1 2 6

Format:

Data

Example:

22

Note

- Superb deadline: 10/19 Tue
- Deadline: 10/26 Tue
- Pass the test of our online judge platform
- Submit your code to E-course2
- Demonstrate your code remotely with TA
- **C Source code (i.e., only .c)**
- Show a good programming style

Further Reading

- An ensemble of replication and erasure codes for cloud file systems, IEEE INFOCOM 2013
- Fast Erasure Coding for Data Storage: A Comprehensive Study of the Acceleration Techniques, ACM FAST 2019
- …