Tutorial

Distributed Systems (IN2259)

Sample Solution: Exercises on CAP Theorem and Erasure Coding

Exercise 1 CAP Theorem

For each of the following scenarios, answer what kind of system is the most adequate for the situation (CA, CP, AP). Justify your answer.

(a) A global monitoring system for air traffic controllers. Each Air Traffic Control (ATC) client can request information about any flight. When an anomaly occurs, clients are required to be informed without delay in order to avert disasters. The air traffic controllers are directly connected to a dedicated optical fiber network with a Mean Time Between Failures (MTBF) of 500 years and six-nines availability.

Solution: CA System: Since the network is extremely robust, we can assume no network partitions will occur. Because the system will be handling mission-critical data, it must operate with maximal consistency and availability.

(b) An online shopping website which is specialized in Christmas gifts. It only sells items once at the beginning of each month, at a heavy discount. Items are only guaranteed to be in time for Christmas, no matter when you order.

Solution: AP System: Since it is a normal online site, maintaining P is essential. Between C and A, availability is more important since the website only goes online once a month, so it cannot afford to be unavailable during that time. Because there is a long leeway for fulfilling orders, inconsistent operations can be resolved after the fact. For instance, the system could accept an arbitrary amount of orders, close the sale, then verify which orders can be fulfilled given the amount of stock. Only those orders will be charged, the rest will be canceled.

(c) The local branch of a bank wishes to record all operations performed at its ATMs. Because of security issues, this information cannot leave the physical boundaries of the office (e.g., through a network out of the bank, or with physical media).

Solution: CA System: Since this is a local application, the ATM could rely on a single mainframe computer to log all transactions. This would satisfy the privacy requirements and avoid all network communications. In this case, partitions cannot occur in a centralized system, therefore CA is the appropriate choice.

(d) A promotional offer in a restaurant chain allows customers to use a raffle app on their phone to obtain a prize if their bill exceeds a minimum amount. The grand prize is a new car. The app is controlled by the restaurant company and the promotion operates under a strict global budget which cannot be exceeded. If there are any technical issues with the app, a rain check ticket is given to the customers so that they can come back later to check and claim their prize.

Solution: CP System: Since this is a restaurant chain with a global budget, a distributed system is appropriate. P must therefore be maintained, and the choice is between A and C. Because there is a backup mechanism if the app fails (a rain check), C is the appropriate choice, so that the company doesn't have to rescind the prizes if an inconsistent state occurs. For instance, if multiple cars are handed out when there is only one grand prize. Rescinding a prize would be a PR nightmare. Thus, consistency is important.

Exercise 2 Reed Solomon Error Correction

A data center makes use of 6 storage servers to store a set of data. The storage manager breaks down the data into shards and stores them across all storage servers. Furthermore, the storage manager utilizes Reed-Solomon encoding for generating and storing parity shards for original data, which are used to recover the corrupted data.

(a) Divide the text "Reed Solomon Err" into four shards and create the parity shards by using the following encoding matrix. (**Hint:** Use the ASCII Table to convert the string to decimal values. `http://www.asciitable.com/`)

$$\begin{bmatrix} 01 & 00 & 00 & 00 \\ 00 & 01 & 00 & 00 \\ 00 & 00 & 01 & 00 \\ 00 & 00 & 00 & 01 \\ 27 & 28 & 18 & 20 \\ 28 & 27 & 20 & 18 \end{bmatrix}$$

Solution:
We break the string into four shards and store them in a matrix, where each row present one shard.

$$\begin{bmatrix} R & e & e & d \\ SPACE & S & o & l \\ o & m & o & n \\ SPACE & E & r & r \end{bmatrix}$$

We use the ASCII Table to convert each entry to its corresponding decimal ASCII value.

$$\begin{bmatrix} 82 & 101 & 101 & 100 \\ 32 & 83 & 111 & 108 \\ 111 & 109 & 111 & 110 \\ 32 & 69 & 114 & 114 \end{bmatrix}$$

We multiply the encoding matrix with the converted matrix to calculate the parity matrix. The two last rows of parity matrix are the parity shards which are stored in the storage servers.

$$\begin{bmatrix} 01 & 00 & 00 & 00 \\ 00 & 01 & 00 & 00 \\ 00 & 00 & 01 & 00 \\ 00 & 00 & 00 & 01 \\ 27 & 28 & 18 & 20 \\ 28 & 27 & 20 & 18 \end{bmatrix} * \begin{bmatrix} 82 & 101 & 101 & 100 \\ 32 & 83 & 111 & 108 \\ 111 & 109 & 111 & 110 \\ 32 & 69 & 114 & 114 \end{bmatrix} = \begin{bmatrix} 82 & 101 & 101 & 100 \\ 32 & 83 & 111 & 108 \\ 111 & 109 & 111 & 110 \\ 32 & 69 & 114 & 114 \\ \mathbf{5748} & \mathbf{8393} & \mathbf{10113} & \mathbf{9984} \\ \mathbf{5956} & \mathbf{8491} & \mathbf{10097} & \mathbf{9968} \end{bmatrix}.$$

(b) Imagine that the first two shards of the original data are corrupted due to storage server failures. Recover the original data according to the encoding matrix and the stored parity shards.

Solution:
Because the first two shards are corrupted, we need to calculate the inverse matrix of the last four rows of the encoding matrix.

$$\begin{bmatrix} 00 & 00 & 01 & 00 \\ 00 & 00 & 00 & 01 \\ 27 & 28 & 18 & 20 \\ 28 & 27 & 20 & 18 \end{bmatrix}^{-1} = \begin{bmatrix} -74/55 & 36/55 & -27/55 & 28/55 \\ 36/55 & -74/55 & 28/55 & -27/55 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

And finally, for recovering the original data, we multiply the calculated inversed matrix with the four last rows of the parity matrix we calculated in the previous part.

$$\begin{bmatrix} -74/55 & 36/55 & -27/55 & 28/55 \\ 36/55 & -74/55 & 28/55 & -27/55 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 111 & 109 & 111 & 110 \\ 32 & 69 & 114 & 114 \\ 5748 & 8393 & 10113 & 9984 \\ 5956 & 8491 & 10097 & 9968 \end{bmatrix} = \begin{bmatrix} 82 & 101 & 101 & 100 \\ 32 & 83 & 111 & 108 \\ 111 & 109 & 111 & 110 \\ 32 & 69 & 114 & 114 \end{bmatrix}.$$

And finally, we use the ASCII Table to convert back the matrix to the original string.

$$\begin{bmatrix} R & e & e & d \\ SPACE & S & o & l \\ o & m & o & n \\ SPACE & E & r & r \end{bmatrix}$$