CSC547 Homework Assignment #2

Akruti Sinha (@asinha6) || Priya Andurkar (@pandurk)

Problem 1:
Problem 2.3. Storage. Nutanix (https://www.nutanix.com/ is a manufacturer of storage products; they have a local, RTP presence and they hire NCSU graduates frequently. It might be a good idea to know the company a little better...

1. Find the model numbers of a few (hardware) storage devices with capacities over 1TB.
2. In https://www.nutanix.com/sg/solutions, they classify Storage as “Files Storage”, “Objects Storage”, and “Volumes Block Storage”. Describe, in your own words, these types of storage.

Solution 1:
Solution 2.3

1. Nutanix offers the following hardware storage solutions:

   For more information, please see the following link:
   https://www.nutanix.com/products/hardware-platforms/specsheet

   - NX-1065-G9: This most recent addition has a maximum storage capacity of 43.68 TB, 1024 GB of memory, and 24 cores per CPU socket powered by Dual Intel Sapphire Rapids CPUs. Storage configurations include SSD+HDD and SSD+HDD SED. It excels in backup and disaster recovery use cases, as well as remote office/branch office scenarios.
   - NX-3035-G9: The NX-3035-G9 is a newcomer with strong Dual Intel Sapphire Rapids CPUs, 2048 GB of memory, and 28 cores per CPU socket. It has a maximum storage capacity of 87.36 TB and is available in a range of storage options including All SSD, NVMe+SSD, All SSD SED, SSD+HDD, and SSD+HDD SED. This adaptable solution excels in applications such as IT applications, databases (DB), private clouds, server virtualization, backup and disaster recovery, and virtual desktop infrastructure (VDI).
   - NX-3060-G9: The Nutanix NX-3060-G9 is a new addition to the Nutanix range, with Dual Intel Sapphire Rapids CPUs, 1536 GB of memory, and 20 cores per CPU socket. It has a maximum storage capacity of 46.08 TB and storage types such as All SSD, NVMe+SSD, and All SSD SED. This paradigm is ideal for Databases (DB), IT Apps, Server Virtualization, Virtual Desktop Infrastructure (VDI), End-User Computing, Remote Office/Branch Office, and Private Cloud deployments.
   - NX-8150N-G8: This model has a storage capacity of 184 TB, 4096 GB of memory, and 40 cores. It stores data using self-encrypting and NVMe SSDs, making it suitable for applications such as business essential operations, disaster recovery, and big data analytics.
   - NX-3170N-G8: This option offers 92 TB of storage, 2048 GB of memory, and 32 cores. It, like the previous option, uses self-encrypting and NVMe SSDs for storage. Its main applications are private cloud deployments and end-user

computing environments.

2. Storage is classified into three forms on the Nutanix solutions page, https://www.nutanix.com/sg/solutions: "File Storage," "Object Storage," and "Block Storage." Here is our explanation:
    - File Storage:
        - "File Storage" is presented by Nutanix as a way for organizing and storing files in a structured manner, similar to what is found in operating systems such as Windows or MacOS. Files are arranged into folders, and each file can be identified uniquely by its path.
        - Files are spread across nodes in an existing cluster or on a specialized storage cluster in a cloud-based file storage system. These storage devices can be used by multiple customers to store their data.
        - The cloud service provider has the ability to increase or decrease storage capacity based on their individual needs.
        - Because of its familiarity, this storage format is a popular choice for many users, even if it may not be the most efficient solution for managing massive volumes of unstructured data.
    - Object Storage:
        - Object storage is primarily intended for storing large amounts of unstructured data and is commonly used for archival and backup purposes.
        - This storage type's objects contain the actual data to be stored, as well as metadata and a unique identifier. Unlike file storage, objects are stored in a big pool of data rather than in folders. For fault tolerance, large groups of objects are joined together and replicated at multiple places.
        - Users can access objects via an API request, using GET requests for data retrieval, PUT/POST requests for data uploads, and DELETE requests for object removal.
        - Among the three storage types—file-based, block, and object storage—object storage has the most scalability, making it an excellent choice for managing large data volumes.
    - Block Storage:
        - Block storage entails dividing data into uniform-sized chunks, each with its own unique identifier. These portions are then kept in different locations and recombined when the complete file is needed.
        - To ensure fault tolerance, multiple copies of the same chunk are kept in separate locations. Because of this redundancy, different pathways can be used to reassemble the original file, increasing retrieval speed.
        - Block storage is frequently used to support containerized applications and to establish virtual machine file systems.

Problem 2:
Problem 2.7 Consider https://www.nfl.com/, the official website of the NFL.

1. Describe as many types of requests as you can, for this website.
2. For each type, mention very clearly, who can potentially create such requests.
3. For one type only, can you hypothesize about the “pattern” of such requests (e.g., it is uniform, may have peaks, etc.)?

4. For one type only, can you guesstimate how much time a request would take to be processed? State your assumptions clearly.

Solution 2:
Solution 2.7

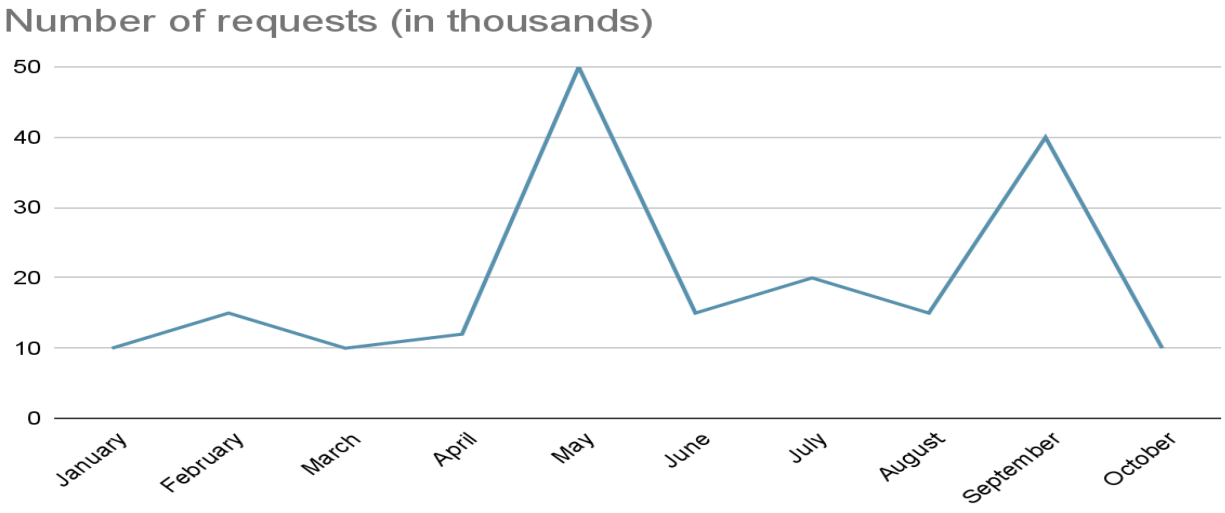
1. The official website for NFL is a huge platform with a wide range of information available to end users and various other systems which consume their data. There are multiple types of requests which it can serve and subsequently provide data for. Following are some categories of the types of requests:
    a. User sign-up and authentication – This request helps user to create accounts and login subsequently
    b. Real time score updates – This request helps in displaying the real time score updates of ongoing matches
    c. Live streaming matches – This request enables users to be able to stream and watch ongoing matches on their devices
    d. Provision of statistical information – This request provides various statistics of data available for past games, players, leagues and more
    e. Managing user accounts – This request deals with managing the services specifically for a given user such as their subscriptions, purchases, etc
    f. Fantasy football – This request provides users with ways to play fantasy football which and manage teams based on statistics of games, leagues, rosters, etc
    g. Ticket booking and management – This request allows user to be able to book tickets and manage their bookings
    h. Merchandise retail – This request helps with displaying shopping catalog, placing and fulfilling orders
    i. Security – This request enables securing user transactions, protecting their personal data, and protecting them against potential fraud on their platform
    j. Latest news – This request provides latest news available on players, matches, and more
    k. Customer support – This request helps user get support for their queries and feedback
    l. Searching requests – This request enables user to lookup teams, players, and other details
    m. Multiple device access requests – This request helps user load the website content on multiple devices

2. These above mentioned requests can be created by multiple agents:
    a. User sign-up and authentication – Individual end-users
    b. Real time score updates – Individual end-users, third-party applications integrating NFL content APIs and showing match updates
    c. Live streaming matches – Individual end-users
    d. Provision of statistical information – Individual end-users, third-party applications integrating NFL content for statistical analysis
    e. Managing user accounts – Individual end-users, NFL website to give tailor recommendations based on user details and activity
    f. Fantasy football – Individual end-users
    g. Ticket booking and management – Individual end-users, third-party applications integrating NFL content and enabling ticket booking

h. Merchandise retail – Individual end-users, third-party applications which also integrate with NFL catalogs APIs for e-commerce
i. Security – These requests would not be directly invoked by end-user but rather, any activity on user's end such as logging in, managing subscriptions, changing personal information would in turn call these requests implicitly
j. Latest news – Individual end-users, third-party applications which integrate with NFL news board APIs
k. Customer support – Individual end-users needing help with account, website, content, and more
l. Searching requests – Individual end-users, automated systems showing results as found within NFL API responses
m. Multiple device access requests – Individual end-users trying to access the website content from multiple devices such as mobile phones, tablets, laptops etc

3. We believe that the pattern of usage of the Merchandise Retail requests would mostly be uniform, however it would see peaks around game seasons, major matches, seasonal sale offers, and more. End-users are likely to shop on their website throughout the year for personal use, for gifting, and other reasons. However, in the above mentioned scenarios, the traffic on the website would increase significantly for placing orders.

   Following is a hypothetical example of what the trend of incoming requests could look like for the online retail for nfl.com. The y-axis represents the number of requests and the x-axis represents the months. So let's say that there is a major game season in May and a huge 80% discount sale in September, those are the timelines that would see the most requests coming in as people would shop the most. Throughout other months, the traffic is uniform without much difference due to frequent shoppers or browsers of the catalog.



4. For this question, we tried to guesstimate the time it would take to fulfill a request for ticket booking and management. The time taken to fulfill any request is dependent on multiple factors such as, load on the servers, user's network connectivity, efficiency of

the algorithm, physical distance of the server from the user, size of the request and response data, and many more.

For this guesstimate, we assumed the following:

1. User is booking tickets for an unpopular game, hence the load on the server is minimal
2. User is in close geographical proximity of the NFL server
3. User has good network connectivity

Actions	Assumed time taken
User authentication	50 milliseconds (0.05 seconds)
Communication within server to send details of request to process it after successful authentication	30 milliseconds (0.03 seconds)
Communicating with banking client API to confirm payment receipt	700 milliseconds (0.7 seconds)
Saving the newly booked ticket details for the particular user	200 milliseconds (0.2 seconds)
Retrieving all booking details for the particular user to send as a response	600 milliseconds (0.6 seconds)
Transmission time over the network	100 milliseconds (0.1 seconds)

Hence the total time taken to fulfill the request to book the tickets and to provide data to users to manage their bookings it would take 1.18 seconds

Problem 3:
Problem 2.14 Tenants in a Facebook datacenter. Consider the set of Facebook users. Supply at least three different criteria for partitioning this set into tenants. Describe how you could identify traffic entering a datacenter as belonging to a specific tenant. Discuss, if possible, advantages and disadvantages of your proposal.

Solution 2.14

Here are three possible criteria for categorizing Facebook users into tenants, as well as ways to identify traffic that belongs to a given tenant:

1. **Geographic Location:**
    a. **Criteria**: Users could be divided into tenants based on where they live in the world. Users are classified according to their geographical location or region (for example, North America, Europe, or Asia).
    b. **Identifying Traffic for this tenant:** This tenant-specific traffic is easy to identify. Facebook may identify traffic by examining incoming request IP addresses. GeoIP databases can assist in determining a user's location.
    c. **Advantages**:
        i. By serving consumers from datacenters closer to their location, latency is reduced.
        ii. Users in different regions might be given localized content and services.
        iii. Another advantage is that it complies with data sovereignty requirements.
        iv. Performance improvements for users in densely populated areas.
    d. **Disadvantages**:
        i. Increased complexity in data center infrastructure and operations.
        ii. Users who are concerned about Facebook tracking their whereabouts may have privacy issues.
        iii. As people interact with friends all around the world, their actual relationships or preferences may differ.
        iv. Complicated routing and infrastructure management.
2. **User Activity or Usage Patterns:**
    a. **Criteria:** Users can be classified according to their activity or usage patterns, such as regular posters, gamers, advertising, or business accounts. In other words, users could be classified as tenants depending on their Facebook behavior. Users, for example, could be classified as tenants based on their interests, the types of material they consume, or the frequency with which they interact.
    b. **Identifying Traffic for this tenant:** This tenant's traffic can be identified in a variety of ways. The user's cookie ID or other unique identifier can be used to identify traffic from this tenant. To classify users into distinct usage patterns, analyze user behavior, API queries, and content can also be used.
    c. **Advantages:**
        i. Facebook can optimize resources for various types of users.
        ii. Users can receive more customized advertising and content recommendations.
        iii. Facebook can better understand its users' needs and interests.
        iv. It can provide better monetization opportunities to the business users and strategies to the marketing teams.
    d. **Disadvantages:**

i. Users who are concerned about Facebook tracking their activity may have privacy issues.
ii. If Facebook employs tenant partitioning to limit the amount of resources that particular groups of users have access to, there is a possibility of discrimination.
iii. It can be difficult to precisely categorize users based on their behavior.
iv. Users with multiple activities have limited flexibility.
v. Policy enforcement for each specific category can be difficult.

3. **Privacy Level:**
   a. **Criteria:** Users should be classified according to the sensitivity of their data, such as public profiles, private chats, or financial activities. In other words, based on their privacy choices, Users could be classified into tenants. Users who have chosen extra privacy, for example, could be assigned to a different tenant.
   b. **Identifying Traffic for this tenant:** The user's privacy settings can identify traffic from this tenancy. Alternatively, analyze the type of data accessed or transmitted during queries and categorize users accordingly.
   c. **Advantages:**
      i. Data security and privacy restrictions for sensitive accounts can been enhanced.
      ii. Compliance with privacy requirements (for example, the GDPR).
   d. **Disadvantages:**
      i. If consumers are concerned about their data being isolated from other users, they may be less reluctant to share information with Facebook.
      ii. Processes for tagging and categorizing data can be complex.
      iii. Potential performance costs associated with rigorous data classification

BONUS QUESTIONS

Problem 4:
Problem 2.12. Workloads in an AWS datacenter. Find out articles for the types of workloads Amazon hosts in their data centers. Describe, in your own words, the main features of these workloads

Solution 4:
Solution 2.12:

We know that in the context of cloud computing, the term workload refers to capabilities, applications, and services that consume cloud computing and cloud storage resources. From the below provided reference [1], we found out about some such workloads running in the AWS data centers:

- Compute - An example of compute workload in AWS would be the Amazon Elastic Cloud Compute. As per reference [2], it is a "*web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers.*" It provides multiple important features to their users. Some of those features include:
  - Auto Scaling
  - Instances based on required optimizations

    - Choices of Operating Systems
    - High Availability
    - Security

  The traffic within these workloads can vary based on the customer applications. For some e-commerce website being hosted on one such instance, there may be uniform traffic throughout the year but there may also be intermittent spikes around the time of huge sale offers. In our example of an e-commerce website, we can further say that this workload requires multiple processors, however for simpler compute applications a single processor can also be used. These workloads may use both compute and storage servers which are dependent on the need of the application. In our example we can say that the common units of measurement include the number of requests.

- Storage - An example of a Storage workload is Amazon Elastic Block Store. As per reference [2], it *"provides persistent block storage volumes"*. Some of its features are as follows:
    - Highly durable
    - Highly available
    - Scalability of volumes
    - Encryption
    - High performance
    - Easy integration across other services

  The traffic on EBS workloads can be higher for periodic data loads and data-backup processes, and can be uniform for regular database queries. These make use of storage servers only, however it can be integrated with other workloads which use compute servers and may use multiple processors for the same.These workloads can be measured with the amount of data being transferred, or the number of read and write operations. The time taken by these workloads are largely dependent on the associated EC2 instances and their performance.

- Analytics - An example of an Analytical workload is Amazon Athena. As per reference [2], it is an *"interactive query service that makes it easy to analyze data in Amazon S3 using standard SQL"*. It is very useful for querying large datasets and performing analysis. Following are some key features:
    - Serverless
    - Integration with Business Intelligence tools
    - Similar to SQL
    - Integration with storage workloads
    - On-demand analysis on semi-structured data

  The workload on Athena can be both bursty and uniform, depending on the frequency of data queries being fired. For relaxed calls, the workload may be uniform but irregularity in the intervals of calls may cause a burst in activity. It makes use of multiple processors in the background to optimize and parallelize the processing of the query being fired. This workload is primarily a compute service hence does not use any storage server of its own. To measure the workload, we don't use the generic number of requests as in case of compute workloads, we make use of the queries itself. A request on Athena can be served and response time may vary based on the complexity of the queries, amount of data being looked up.

- Machine Learning - An example of a Machine Learning workload in AWS is Amazon Forecast. As per reference [2], it is a *"fully managed service that uses ML to deliver highly accurate forecasts"*. It helps in providing business intelligence by providing accurate forecasts based on historical data. Following are some of its features:
    - Data importing and pre-processing
    - Implements effective time series forecasting
    - Calculates accuracy metrics
    - Integration with other AWS services
    - Compliant with security regulations
    - Fully managed service

  The forecast workload also has varying traffic, based on the client forecasting requests. These workloads are auto scaling workloads and hence the underlying infrastructure and the number of processors being used may change as per the need of the customer forecast request. It is a compute intensive workload and hence it uses compute servers, however it is dependent on other storage workloads for provision of input data and exporting results. The unit of measurement for this workload would essentially be the forecasting task. Based on the size of the historical data input, the time on completion of a particular task may also vary.

References

[1] https://aws.amazon.com/products/?aws-products-all.sort-by=item.additionalFields.productNameLowercase&aws-products-all.sort-order=asc&awsf.re%3AInvent=all&awsf.Free%20Tier%20Type=*all&awsf.tech-category=tech-category%23compute

[2] https://docs.aws.amazon.com/whitepapers/latest/aws-overview/introduction.html

[3] https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Storage.html

Problem 5: Suggest your own problem on a theme related to Topic 2 and provide a solution. Indicate the level of (perceived) difficulty.

Solution 5:

A. In problem number three, we discussed some of the drawbacks of categorizing tenants into numerous groups. To overcome the said drawbacks, how would we create an equitable resource allocation system for a multi-tenant datacenter where different tenants share computing resources with different requirements and different priorities?

Solution: In order to address the issue of equitable resource distribution in a multi-tenant datacenter:

- Resource Profiling: Compile detailed statistics on tenant resource utilization.
- Resource Pools: Create segregated resource pools for CPU, memory, storage, and so on.

- Allocation Policies: Create dynamic policies that take priority and past usage into account.
- Dynamic Allocation: Implement real-time modifications and load balance
- Resource Reservation: Allow tenants to reserve resources for peak demand.
- Monitoring: entails continuously tracking resource utilization and setting up alarms.
- Scaling: Scale resources automatically based on demand.
- Quality of Service and Fairness: Ensure equitable allocation and prioritize key applications.
- Security: Tenant security is ensured by strong isolation and RBAC.
- Feedback Loop: Collect tenant feedback to improve policy.
- Reporting: Provide visibility into consumption and estimate resource requirements.
- Reclamation: is the process of reclaiming and redistributing idle resources.
- Continuous Optimization: entails reviewing and optimizing allocation policies on a regular basis.