Tylar Simone
017510582
Due: 07/10/2020

CECS 327 Assignment 1

1. Name two advantages and two disadvantages of distributed systems over centralized systems.

Two advantages of utilizing a distributed system over a centralized system are lower latency due to geographical spread of each node and independence of each node allowing the system to continue operating if one of more nodes lose connectivity. However, the disadvantages of utilizing distributed systems in lieu of centralized systems are the issues related to not having a common clock between nodes which leads to difficulty in ordering events and transactions and the high level of maintenance required to efficiently operate distributed systems.

2. Describe what is meant by a scalable system.

A scalable system, in terms of distributed systems, is a system's ability to adjust to an ever changing set of criteria. These systems are measured along these three criteria: Size, Geographical and Administrative. Size scalability refers to the system's ability to increase the number of users or resources without a noticeable loss of performance. Geographical scalability refers to the system's ability to seamlessly integrate communication between users or resources over long geographic distances. Finally, administrative scalability refers to the system's manageability even if it spans across several independent organizations.

3. Executing nested transactions requires some form of coordination. Explain what a coordinator should be doing in a scenario like this.

Executing nested transactions potentially leads to several subtransactions all executing in parallel with each other. A coordinator must be able to logically divide a single transaction such that it's subtransactions utilize and manipulate its own private data that can be returned to its initial state if its parent transaction is aborted. In other words, the original set of data of the initial transaction must be able to return to its initial state even after one or more subtransactions have already been completed.

4. What is the role of middleware in a distributed system?

The role of middleware in a distributed system is to manage communication, security, accounting, recovery and resources through various applications across a networked environment.

5. Explain what is meant by a virtual organization and give an example on how such organizations could be implemented.

A virtual organization is a collection of individuals and/or entire organizations that share resources and information in a collaborative effort for some common goal. These organizations interact through a variety of communicative methods such as the internet, telecommunications, emails, etc. to exchange information between its members. A recent example of a virtual organization coalescing for a common goal would be the amalgamation of Covid19 related deaths and infections across all 50 states to gain a better understanding of where the largest outbreaks are located and where to send shared resources (ideally, in practice maybe not) to better combat this pandemic.

6. What is the difference between multiprocessor and multicomputer?

A multiprocessor is an organization of multiple CPUs in which each CPU has access to the same physical memory. In contrast, a multi computer system is an organization of multiple computers connected through a network with each computer having access to its own physical memory rather than a shared memory.

7. Why is it sometimes so hard to hide the occurrence and recovery from failures in a distributed system?

A distributed system can have difficulty distinguishing the difference between a failed process and an unexpectedly slow one. This is especially apparent when a user attempts to access a website but the browser times out during the loading process. The entire web server could be down, or could just be experiencing a high volume of requests.

8. In class, I discussed why having transparency isn't always a good thing. Do you agree with me? Explain your reasoning.

I agree that high transparency isn't always a good thing for the simple fact that more transparency causes debugging issues. If a piece of software isn't functioning correctly, I would like to know the exact reason, or at least be able to easily conclude the reason, that the issue exists.

9. Scalability in distributed systems can be achieved by applying different techniques. What are these techniques?

Distributed systems can achieve scalability by implementing the following techniques:

- Hide communication latencies by utilizing asynchronous communication and having a separate handler for incoming responses.

- Splitting components into smaller parts and subsequently spreading the parts across the system in a technique called partitioning and distribution.
- Relplicating components across a distributed system to minimize performance degradation.
- Allowing the client of a resource to make a copy of that resource in a process called Caching.

10. When a transaction is aborted, the computing system has to be restored to its previous state as if the transaction never happened. This is actually a lie. Give an example where resetting the computing system is practically impossible.

A common example of this occurs at my office when utilizing the file management software, Egnyte. Egnyte allows users to access files across multiple platforms. Using either the desktop application or the web based application exclusively results in normal operations regarding file management and manipulation. The issues arise when the same file or files are accessed or manipulated when both applications, web and desktop, are used. The inherent lag between the desktop application and the cloud based server that houses the files creates a fairly large window of opportunity where the same root file can be edited by both of the applications. When the file is terminated, Egnyte's system cannot differentiate between the two new "master" files and decides that the last one to be closed is now the conflict copy or simply shows a save error when attempting to save the file under the same name. The refresh rate between the web based application and the desktop application used to be several minutes or, in some cases, hours long, causing multiple conflict copies of the same file to be present after a day of work, creating extreme confusion between users about which is the correct file to be using in the future. This is a rather long explanation to this question, but the original document is lost as soon as it is opened more than once during the "refresh" period and a series of conflict copies are created.