# Introduction

The concept of distributed database is one that I am quite familiar with. For my work, I am currently in the process of improving my application’s database performance, and one of my ideas is through distributed databases.

# A distributed database is a collection of several different databases distributed among multiple computers. Discuss why a company would want a distributed database.

A company might consider using a distributed database for a few reasons: data availability, security, scalability, and performance. “Access to data is a critical feature of an efficient, progressive and ultimately self-correcting scientific ecosystem” (Altman, et al., p. 1). Within the growing Internet of Everything, seemingly exponential amounts of data are being uploaded and distributed every day. People have access to so many resources with the stroke of keypad. Along with this unfathomed amount of data comes the expectation of real-time result. We want our data now. Companies, such as Netflix, are aware of our need for data and have therefore adopted the distributed database model. Distributed databases store data in separate locations or on separate databases or computers (Megha, p. 1). This decoupling of data provides a few advantages. Since data is decoupled, the users have to sort through less data. This reduces the query time to receive the data, enhancing performance. Also, by storing different data in different regions, users in those regions will receive their data more quickly since they do not have to sort through an entire superset of data. This is similar to database partitioning, which is distributing data, but within a single database. I have use database partitioning to achieve performance increases within my role as a software engineer. Partitioning creates database tables with subsets of data based on the specified schema. This does two things: reduces the amount of data and blocks unauthorized access. These can be translated of you distributed databases. A company might choose a distributed database to block users from having access to an entire superset of data that they do not need access to. If each of these databases are distributed geographically and the company has multiple locations, this helps the companies work more efficiently by seeing only the applicable data for that location. Distributed database architecture also works well for scalability. If a company were to add a new location, it would a smaller database server to store only the new location’s data, instead of storing the entire company’s data. Only storing local data does have a disadvantage; outside data cannot easily be shared, which is why data replication must be considered between the distributed databases.

# Productivity increases as rapid response times are achieved. Discuss what is considered an acceptable system response time for interactive applications.

Interactive response time plays a key role in user productivity, but only in data-entry tasks (Corl, Martin, p. 3). Tasks that require problem solving or personal computations are not generally affected by a slightly longer response time. This is due to people needing to understand their tasks before completing them, which makes sense. I can only work as fast as my brain allows me. Now, one could teach his or her brain to “work faster,” in which case, the argument holds that rapid response time increase productivity.

The question of acceptable system response times is a rather open-ended question, simply due to the diverse systems that users interact with. Users might interact with a web interface, API, database, standalone application, etc. The system might be simple and perform one or two tasks, or it might perform ten or eleven tasks and return a larger amount of data. Ecommerce websites require a latency budget less than 100ms (Brotherton & Dietz, p. 4). For an interactive web application, I would consider an acceptable response time to be somewhere between 500ms to 1000ms. I base this on the amount of data needed to be loaded on the page. For an API, I would consider an acceptable response time to be between 100ms to 400ms. The applications I create require response times less than 200ms. Sometimes this is difficult due to the amount of data, so I must look within the application to determine how to reduce the response times. This is an instance where distributed databases would benefit the system, by reducing the amount of data the application sorts through. With regard to manipulating an interactive application, the framework that the application is built using also comes into play. When comparing Anguler and Blazor web development frameworks, Nilsson (2021) found that Angular was the more favorable framework due to overall performance.

# A fully centralized data processing facility is centralized in many senses of the word. Discuss the key characteristics of centralized data processing facilities.

Centralized data processing refers to data processing performed by one or more computers located in a central facility. All the data is stored and processed in one location (Foroughi, ch. 3). Centralized processing can be beneficial for a few reasons: simplicity, security, and cost. Since all of the necessary hardware for the data processing is localized, a company or user does not need to pay for multiple facilities. This can also have negative impacts if the data set is enormous. As part of my work, I helped move data from a centralized facility into a Cloud database service. This was done primarily for cost. But there is another benefit of a centralized data processing facility, which is security. Since there is only one facility and one way to access the stored data, the data is much more secure than if it could be accessed over the Internet. Especially with more and more access to more and more information every day, data security is a major issue, and a centralized data facility helps to mitigate that risk. There are, however, downfalls to a centralized facility. It can be susceptible to centralized attacks, either physical or digital. Centralized facilities do create a good environment for maintaining data integrity. Since all the data resides in one place, there is no need for data replication. Data cannot become out of sync in different locations. In a single location, scalability is easy because new computer systems can be added more easily (Akhtar, p. 2). Lastly, a centralized data processing facility is simpler than a distributed facility. This may seem obvious, but it is worth noting because simplicity is a key factor in system design. Data centers, due to their nature, are comprised of mostly the same types of equipment: routers, switches, firewalls, cables, modems, and hard drives. All computers on premises are connected through a LAN network. One or many computers or hard drives act as a server for storing all the data. This decoupling is great because if one hard drive breaks, only the data on that hard drive is affected. When whole server needs to be down, either for maintenance or processing, this is a problem for system users. They are not able to access their data. I face this problem at work; some of the applications I rely on are only available during certain hours because the data centers turn off availability to network traffic. They do this for maintenance and batch processing, but it is inconvenient. Having a singular LAN data center does allow for increased performance, depending on the application, since all of the data is immediately present.

# Equipment and communication redundancies are common in today's data centers. Discuss the major types of equipment and communication redundancies found in today's data centers.

“Redundancy is exhibited as the diversity of pathways and is critical for a system’s capacity to adapt under changing environmental conditions arising from shocks and disturbances (Fath, et al., p. 2). Modern systems must have the ability to backup and route data in many ways to account both for network traffic and system disturbances. A few common practices exist to ensure the redundancies and fault tolerances of today’s data centers. Brotherton and Dietz (2014) point out a simplistic seven tier site redundancy plan. Tier seven describes and automated failover process and site recovery, which is simply pointing two a backup connection. This requires the data center to have duplicate hard drives, routers, switches, etc., which is a very costly endeavor. One benefit, though, is near zero data loss. Failover process centers are great for Cloud applications, as data is easily replicated across multiple regions. Tier six of the model uses disk and tape storage mirroring. Essentially, the data is only stored in one place, with many pointers to that data. This decouples the hard drives from the switches in the even that a switch or router fails. Tier five incorporates hot sites with two-phase commit. Data can be rolled back to a previous working version without and negative effect. Very little data, if any, will be lost. Tier four incorporates point-in-time copies, which copies data on two active data centers. When a user writes new data, that data is automatically committed at two different sites. Again, this requires two complete data centers. Tiers three through one pertain to localized and lengthy backups. An interesting note is that the higher tiers are much more redundant due implementing multiple data centers. Along with the necessary hardware within the data centers, equipment cooling total electricity consumption are factors to consider. The former can be handled by implementing multiple fans and cooling systems to ensure that hardware is not running too hot. Electricity consumption is based on multiple factors such as location and size. Backblaze is a data storage provider with over 130,000 hard drives in each of its data centers (Amram, et al., p. 1). Obviously, this will be more expensive to run and maintain that a data center with only 50,000 hard drives. Backblaze uses a system modeling technique to predict when a hard drive might need to be replaced. This forethought creates a redundant system by being proactive when system hardware breaks.

# Bibliography

Akhtar, Z. (2021, June 28). *What is centralized database? | functions, advantages & disadvantages*. DatabaseTown. https://databasetown.com/centralized-database-functions-advantages/

Amram, M., Dunn, J., Toledano, J. J., & Zhuo, Y. D. (2021). Interpretable predictive maintenance for hard drives. *Machine Learning with Applications*, *5*, 100042. https://doi.org/10.1016/j.mlwa.2021.100042

Brotherton, H., & Deitz, J. (2014, May). *Data center site redundancy*. Purdue University. https://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=1026&context=i3r2

el Khaled, Z., & Mcheick, H. (2019). Case studies of communications systems during harsh environments: A review of approaches, weaknesses, and limitations to improve quality of service. *International Journal of Distributed Sensor Networks*, *15*(2), 155014771982996. https://doi.org/10.1177/1550147719829960

Foroughi, A. (2018). *Distributed Data Processing* [Slides]. Usi.Edu. https://www.usi.edu/business/aforough/cis367notesf2003/c367ch3.htm

Hardwicke, T. E., Mathur, M. B., MacDonald, K., Nilsonne, G., Banks, G. C., Kidwell, M. C., Hofelich Mohr, A., Clayton, E., Yoon, E. J., Henry Tessler, M., Lenne, R. L., Altman, S., Long, B., & Frank, M. C. (2018). Data availability, reusability, and analytic reproducibility: evaluating the impact of a mandatory open data policy at the journal *Cognition*. *Royal Society Open Science*, *5*(8), 180448. https://doi.org/10.1098/rsos.180448

Kharrazi, A., Yu, Y., Jacob, A., Vora, N., & Fath, B. D. (2020). Redundancy, Diversity, and Modularity in Network Resilience: Applications for International Trade and Implications for Public Policy. *Current Research in Environmental Sustainability*, *2*, 100006. https://doi.org/10.1016/j.crsust.2020.06.001

MARTIN, G. L., & CORL, K. G. (1986). System response time effects on user productivity. *Behaviour & Information Technology*, *5*(1), 3–13. https://doi.org/10.1080/01449298608914494

Nilsson, S. & Dalarna University, School of Information and Engineering. (2021). *Evaluation of the blazor and angular frameworks performance for web applications* (No. 1). Dalarna University, School of Information and Engineering. https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1578257&dswid=5258

Tomar1, P., & Megha, M. (2014). An overview of distributed databases. *International Journal of Information and Computation Technology*, *4*(2), 207–214. https://www.ripublication.com/irph/ijict\_spl/ijictv4n2spl\_15.pdf