The Architecture of Mobile Distributed Database Systems

Hanif I. Lumsden

University of Maryland Global Campus

DBST663

Dr. Gonzalez

03/22/2022

Abstract

A mobile distributed database system (MDDS) is a relatively emerging trend in database technology due to the growth of mobile computing technology. Transactions with mobility are processed over a distributed mobile network. This technology created a new opportunity for research and expansion upon storage, consistency, concurrency control, and distribution. The performance, architecture, and protocols of MDDS differ from that of non-MDDS and are highlighted. The architecture analysis will underline the services, performance, heuristic approach, limitations, and application of MDDS.

Keywords: MDDS, distributed, mobility, mobile computing, mobile database, architecture

The Architecture of Mobile Distributed Database Systems

A database that has information located at two or more different sites and is logically connected on a network is a distributed database system (DDS) (Swaroop & Shanker, 2010). This technique is used so two or more users can access and manage a database system without interfering with each other's work (Swaroop & Shanker, 2010). With advances in technology and devices becoming tinier, mobile computing systems and their potential are becoming more prominent (Swaroop & Shanker, 2010). Transportable machine hardware technology development made distributed systems mobile in the 1990s (Baquero, 2006). Mobility is described based on how the users, computers, and data are mobile concerning the environment that utilizes said users and computing systems in that geolocation (Heuer & Lubinski, 1996). Mobility faces the same issues and primary conditions that stationary distributed computing system faces (Heuer & Lubinski, 1996). The review will review the architecture and functionality of mobile distributed database systems and their applications.

Literature Review

Architecture in Mobile Distributed Database Systems

As the name entails, a mobile distributed database system (MDDS) will allow mobile end users access to the database in mobile clients (Mourlin & Farinone, 2019). This is achieved because MDDSs consist of heterogeneous resources and nodes having mobility traits (Ali & Bagchi, 2018). The client-server architecture of stationary databases extends to the client-agent-server architecture of mobile databases (Heuer & Lubinski, 1996). Cloud computing, storage, and energy constraints for heterogeneous resources are imposed on mobile distributed devices (Ali & Bagchi, 2018). Mobile applications are built to access these remote services required by the user for security and architectural reasons (Dumont et al., 2016). Mobile architecture is either

Simple Object Access Protocol (SOAP) based or Representational State Transfer (REST) based. RESTful services are the preferred choice for mobile applications, according to Dumont et al. (2016), as the data in JSON is a better formatting choice for iOS, Android, and Web clients. RESTful services are ideal for “performance, bandwidth efficiency, robustness” (Dumont et al., p. 2) and building services. The following is a table comparing the two architectures for MDDS (Dumont et al., 2016; Walker, 2022):

Table 1

Comparison between SOAP and REST services

SOAP	REST
Protocol with a web-service description language (WSDL) file containing instructions and location.	Architectural style with the following constraints: Client-server, stateless, cacheable, layered system, uniform interface.
Requires a considerable amount of bandwidth and data transfers.	Low amount of bandwidth for requests that contain JSON messages.
Compatible with XML.	Compatible with JSON, .txt, HTML, XML, and more.
Service interface provided by WSDL that is exposed to client applications.	Service interface provided by uniform service locators accessing hardware components.
Not compatible with REST within SOAP.	Compatible with SOAP within REST.

Regarding scalability, MDDSs can cover many mobile devices and clients across geography (Dumont et al., 2016). Another architecture mentioned by Dumont et al. (2016) is Service Oriented Architecture (SOA), such that each service has distinct functions. SOAP is used effectively in SOA architecture.

Performance in Mobile Distributed Database Systems

In large-scale MDDSs, the network topology of these systems is dynamic, and nodes may become unreliable, causing failed communication between server and client due to low bandwidth (Ali & Bagchi, 2018). Due to the unreliability in nodes and the failure to frequent access to the database server from different mobile clients, design and data workflow implementation are difficult, paired with the delay of said workflow as inefficient task assignments to nodes cause unexpected slowdowns (Ali & Bagchi, 2018). Compared to stationary nodes, mobile nodes are unreliable entities that are low in computing and storage power when it comes to workflow execution (Ali & Bagchi, 2018). (Swaroop & Shanker, 2010). When managing large-scale MDDSs, workflow structural properties, resource computation capacity, and mobility must be considered (Ali & Bagchi, 2018). When monitoring mobile cloud computing platforms, the user is confronted with the challenge of troubleshooting the causes of performance degradation in data processing systems (Zvara et al., 2019). Peer-to-peer data sharing techniques improve workflow performance by reducing latencies in data transfers (Ali & Bagchi, 2018). As mentioned in the prior section, RESTful architecture is preferred as it offers efficient performance in bandwidth efficiency and robustness (Dumont et al., 2016).

Mobile computing has dynamically changing communication costs complicating performance optimization (Swaroop & Shanker, 2010). Connection time is a factor in costs for business applications. Due to it costing less energy to receive than to transmit, the transmission and reception have different bearings on costs for the mobile host (Swaroop & Shanker, 2010). Regarding routing and query processing, MDDSs pose conventional problems (Swaroop & Shanker, 2010). Routes between two mobile hosts will potentially change over time, affecting the network level to a location-based degree (Swaroop & Shanker, 2010). Therefore, communication costs determine the query optimization process and query evaluation strategy (Swaroop & Shanker, 2010).

Decomposition of data-intensive applications in the sub-workflow and carefully placed associated datasets can improve overall performance significantly (Dumont et al., 2016). The goal is to minimize communication costs, processing costs, and end-to-end delays to optimize operations (Ali & Bagchi, 2016). Sub-workflow determines the efficiency of MDDSs: large workflows consist of hundreds of interdependent tasks composed of sub-workflows whose complete time determines the overall execution time (Ali & Bagchi, 2016). Caching in mobile computing, which holds data and transactions to prevent loss, alleviates the performance and availability limitations (Swaroop & Shanker, 2010). This is important for the costs put into mobile communications and connectivity costs as it gives incentives to disconnect the hosts for substantial periods (Swaroop & Shanker, 2010).

The processing of big mobile data has a unique set of requirements for batch processing that requires static queries continually processed over dynamic data streaming (Galić et al., 2017). For data streaming applications, the user is confronted with the optimization obstacles such as unknown data distribution, rapidly changing velocity, new outliers, and the necessity for adaptive intervention; it is recommended that the user implement tracing frameworks to execute numerous parallel tasks (Zvara et al., 2019). Zvara et al. (2019), regarding dynamic data streaming solutions, state that mobile systems must be equipped with intelligent logic to adapt the framework and software to changing workloads.

Heuristic Approaches in Mobile Distributed Database Systems

The heuristic methodology is utilized in database management systems to solve problems quicker than classic methods (Singh & Shanker, 2017). Heuristic methods are paired with forwarding and backward validation locks that give better performance control (Singh & Shanker, 2017). Concurrency control is affected by priority heuristic methods in MDDSs (Singh & Shanker, 2017). Singh & Shanker (2017) explain how priority heuristics decide the sequence of a transaction’s execution based on hierarchal feedback and abortion based on the state of the transaction to minimize transaction miss rate. Using read-only transactions, a new heuristic approach is developed where simulations give performance results higher than the conventional procedures: number of locks and mixed-method approach (Singh & Shanker, 2017). The heuristic approach will locate another valid site for execution (Ali & Bagchi, 2018). Compared to non-mobile systems, the wireless medium makes for a degree of uncertainty for priority heuristics in mobile systems in conjunction with the constant communication between the server and mobile clients during transaction execution (Singh & Shanker, 2017).

Discussion

Application of Mobile Distributed Database Systems

Mobile devices such as laptops, notebooks, and multimedia phones can be used as mediums for mobile distributed database systems (Swaroop & Shanker, 2010). Mobile applications typically have extensive multimedia data (Swaroop & Shanker, 2010). Applications such as navigation satellites and stock trading systems operate on mobile distributed database systems (Singh & Shanker, 2017). Devices run on MDDS like mobile things: unmanned aerial vehicles and robots with antenna and sensors (Zhang & Zhou, 2021). Queries are run to navigate satellites from space to collect data; tickets are booked with credit cards and queried by a buyer from a mobile setting in airline services; submitting queries and filtering information in trading systems (Singh & Shanker, 2017). Everyday applications of mobile distributed database systems involve e-commerce with transactions that run on various sites of an online provider; online multimedia and music service providers use unstructured data to optimize services (Singh & Shanker, 2017; Swaroop & Shanker, 2010).

Zhou (2017) ran an experiment comparing stationary database systems and mobile distributed systems when the test table data is more than 1 million and the concurrent number is 100. As a result, the distributed system has a shorter concurrent query time and system response rate indicating that wireless IoT on mobile internet will lead to the development of the logistics industry (Zhou, 2017).

Conclusion

Mobility brings forward a dimension to distributed database querying. Mobile clients can access and utilize distributed database technology through mobile networks and forms of communication. Mobile distributed database systems pose more performance problems than stationary database systems, but they have a lot of growth potential. Technology seeks to become more convenient, which translates to more utilization of mobile computing.

References

Ali, I., & Bagchi, S. (2018). Designing hybrid graph model and algorithmic analysis of workflow decomposition in mobile distributed systems. *Future Generation Computer Systems, 86*, 145-161. https://doi.org/10.1016/j.future.2018.03.012

Baquero, C. (2006). Mobile Distributed Systems I [PDF]. University of Minho. https://gsd.di.uminho.pt/members/cbm/ps/sdm1beam.pdf

Dumont, C., Mourlin, F., & Nel, L. (2016). A mobile distributed system for remote resource access. *Proceedings of the 14th International Conference on Advances in Mobile Computing and Multi Media*. https://doi.org/10.1145/3007120.3007123

Heuer, A., & Lubinski, A. (1996). Database access in mobile environments. Lecture Notes in Computer Science, 544-553. https://doi.org/10.1007/bfb0034709

Galić, Z., Mešković, E., & Osmanović, D. (2016). Distributed processing of big mobility data as spatio-temporal data streams. *GeoInformatica, 21*(2), 263-291. https://doi.org/10.1007/s10707-016-0264-z

Mourlin, F. P., & Farinone, J. (2019). Cloud mobile storage for mobile applications. *International Journal of Interactive Mobile Technologies (iJIM), 13*(03), 13. https://doi.org/10.3991/ijim.v13i03.8086

Singh, P. K., & Shanker, U. (2017). Priority heuristic in mobile distributed real time database using optimistic concurrency control. *2017 23RD Annual International Conference in Advanced Computing and Communications (ADCOM).* https://doi.org/10.1109/adcom.2017.00014

Swaroop, V., & Shanker, U. (2010). Mobile distributed real time database systems: A research challenges. *2010 International Conference on Computer and Communication Technology (ICCCT).* https://doi.org/10.1109/iccct.2010.5640495

Walker, A. (2022, February 19). SOAP vs. REST: Difference between web API services. Guru99. https://www.guru99.com/comparison-between-web-services.html

Zhang, Z., & Zhou, S. (2021). A decentralized strongly secure attribute-based encryption and authentication scheme for distributed internet of mobile things. Computer Networks, 201, 108553. https://doi.org/10.1016/j.comnet.2021.108553

Zhou, B. (2017). A wireless Internet of things architecture based on mobile internet. *International Journal of Online Engineering (iJOE), 13*(10), 132. https://doi.org/10.3991/ijoe.v13i10.7745