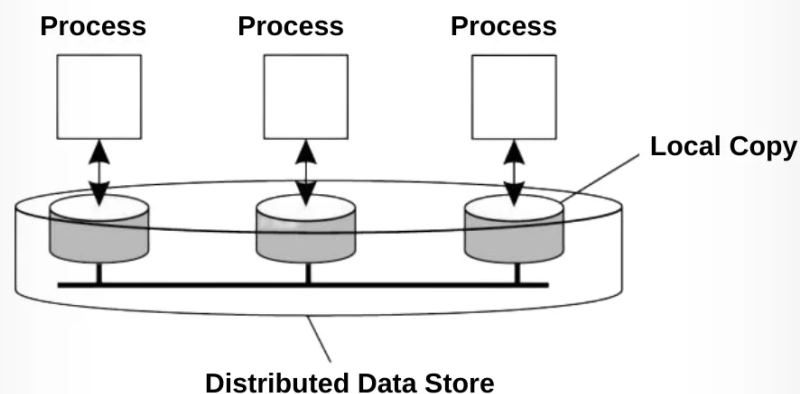


Data Centric Consistency Model in Distributed Systems

🏠 > Artificial Intelligence > Data Centric Consistency Model in Distributed Systems

Data Centric Consistency Model



In Data Science, distributed system is a collect of independent components located in different machines sharing messages to achieve common goals. Consistency Model is a set of rules that governs the behaviour within a distributed system.

The consistency model in distributed system plays a key role in ensuring that the system is consisten and dependable in the face of computing facilities. There are different types of consistency model and each model has its own beneficial features.

The following blog will work as a guide for you to know about Data Centric Consistency Model in distributed systems.

Table of Contents



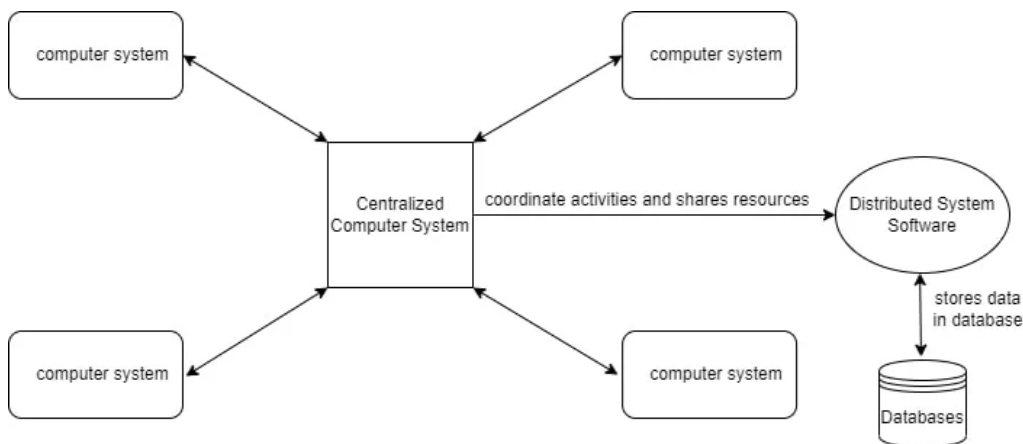
1. What is Data Centric Consistency Model?
2. Types of Data Centric Consistency Model:
 1. Strong Consistency:
 2. Eventual Consistency:
 3. Causal Consistency:
 4. Read-Your-Writes Consistency:

- 5. Monotonic Reads and Writes:
- 3. Difference Between Data Centric and Client Centric Consistency Model:
- 4. Conclusion
- 5. FAQs
 - 1. What is a data-centric system?
 - 2. What are the various consistency models in distributed systems?
 - 3. What is a client centric model?
 - 4. What are the different types of data models?
 - 5. Distinguish any four points between network model and hierarchical model?

What is Data Centric Consistency Model?

The Data Centric Consistency Model is a set of principles and protocols used in distributed systems to define how data is accessed and updated, ensuring that all nodes in the system agree on the state of shared data.

It primarily focuses on the consistency guarantees provided to clients regarding the data they access and manipulate. In a distributed system, data is stored and managed across multiple nodes that communicate and collaborate to provide services and respond to client requests.



However, due to factors like network delays, node failures, and concurrent data updates, achieving consistency in such an environment can be challenging.

Types of Data Centric Consistency Model:

Data-centric consistency models are a set of principles and protocols that define how data is accessed and updated in a distributed system, ensuring that all nodes agree on the state of shared data.

These models primarily focus on the consistency guarantees provided to clients regarding the data they access and manipulate. Let's explore the most common data-centric consistency models in detail:

Strong Consistency:

- Strong consistency is the most stringent data-centric consistency model. It guarantees that all read and write operations on the data will appear to be executed in a single, global order.
- In other words, a read operation will always return the most recent write. This model ensures that all nodes in the system have the same view of the data at any given moment.
- To achieve strong consistency, distributed systems typically use techniques such as locking, serialization, or consensus protocols (e.g., Paxos or Raft). These mechanisms introduce additional latency and coordination overhead, which can impact system performance and availability. However, they ensure that the system behaves as if it were a single, centralized database.

Eventual Consistency:

- Eventual consistency is a more relaxed data-centric consistency model compared to strong consistency. It allows updates to be propagated asynchronously across nodes and promises that, given enough time and absence of new updates, all replicas will eventually converge to the same state.
- Under eventual consistency, clients might observe temporary inconsistencies or divergent views of the data during the convergence process. However, once the system stabilizes, all replicas will have the same data. This model enhances system availability and performance since it does not impose strict coordination requirements on every write operation.
- Eventual consistency is commonly used in scenarios where immediate synchronization is not required, such as content distribution systems, social media platforms, or non-critical data stores.

Causal Consistency:

- Causal consistency is a more refined data-centric consistency model that preserves causality between related operations. It ensures that if one operation causally affects another, the latter will observe the effects of the former.
- For example, if a user posts a message and then later requests to view their posts, causal consistency ensures that the user will always see their own posts. However, it does not guarantee that the user will see posts from other users that were made after the request was initiated.
- Causal consistency allows for greater parallelism and performance improvements compared to strong consistency while still maintaining a partial ordering of operations.

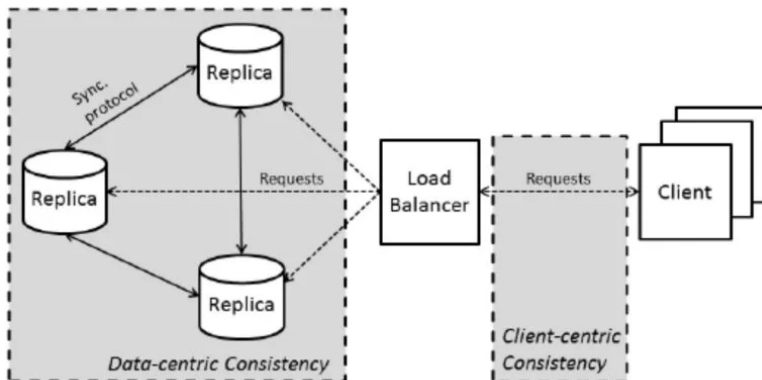
Read-Your-Writes Consistency:

- Read-your-writes consistency is a data-centric consistency model that guarantees that if a client performs a write operation, any subsequent read operation from the same client will always see the updated value. In other words, the client will always see its own writes.
- This model is useful for maintaining a consistent view of the data from the client's perspective and is commonly applied in scenarios like session management, user preferences, or personal data updates.

Monotonic Reads and Writes:

- Monotonic reads and writes are data-centric consistency models that ensure specific ordering guarantees for read and write operations, respectively.
- Monotonic reads guarantee that if a client reads a particular value, it will never observe a previous value in subsequent reads. In other words, the value read by a client will only increase or remain the same over time.
- Monotonic writes ensure that writes from a client are serialized in a way that preserves their order. This means that if a client issues two write operations, the system will ensure that these writes are applied in the same order they were issued.
- These consistency models offer more relaxed guarantees compared to strong consistency but provide useful properties for applications where certain ordering requirements are essential.

Difference Between Data Centric and Client Centric Consistency Model:



Data Centric Consistency Model	Client Centric Consistency Model
Typical models are like Sequential Consistency or Casual Consistency	you can find monotonic reads or read your writes as typical models
guarantees that the results of read and write operations performed can be replicated to nearby stores immediately.	it does not handle simultaneous updates but to maintain a consistent view different replicas can be accessed for different client processes.
Does not guarantee that client can read latest data and may read out of date data.	the focus is on creating a positive customer experience.
restricts data storage for saving ine processing sequence of data.	Believes that clients are the only reason that a business exist and ensures to keep the client satisfied.

Conclusion

From the above blog, you have been able to gain knowledge on Data Centric Consistency Model in a distributed system. As you come to know how data Centric Models are different from Client Centric Consistency Models, you learn the importance of both. Moreover, the Data Centric Model restricts the data storage on what can be stored therefore, ensuring complete privacy and authenticity of data.

FAQs

What is a data-centric system?

A data-centric system is a type of information system or software architecture where the primary focus and organization revolve around data and its management.

What are the various consistency models in distributed systems?

The various types of Consistency Models include the following:

1. Strong Consistency
2. Pipelined Random Access Memory or FIFO Consistency Model
3. Sequential Consistency Model
4. Casual Consistency Model
5. Weak Consistency Model
6. Processor Consistency Model

What is a client centric model?

Client centric model refers to the way in which data store presents the data models and values to individual clients defining how and when the client can access data across different replicas.

What are the different types of data models?

- Strong Consistency
- Casual Consistency
- Eventual Consistency
- Read your writes Consistency
- Monotonic Read and Writes

Distinguish any four points between network model and hierarchical model?

Following are the points that distinguishes network model with hierarchical model:

1. Data Structure:

1. **Hierarchical Model:** In the hierarchical model, data is organized in a tree-like structure with a single root and various levels of child nodes branching out from the root. Each node can have multiple children, but each child can have only one parent, creating a parent-child relationship between nodes.
2. **Network Model:** The network model, on the other hand, represents data as a more flexible graph-like structure. It allows each record to have multiple parent and child relationships, creating a network of interconnected records. This enables more complex and diverse relationships between data elements.

2. Relationship Representation:

1. **Hierarchical Model:** In the hierarchical model, relationships between data elements are represented through a parent-child hierarchy. Each child has only one parent, making it suitable for representing one-to-many relationships.
2. **Network Model:** The network model represents relationships through sets, where a record can be a member of multiple sets. This allows the representation of many-to-many relationships, making it more suitable for complex data structures.

3. Flexibility:

1. **Hierarchical Model:** The hierarchical model provides a simple and straightforward representation of data with a fixed, one-to-many parent-child relationship. While it works well for certain types of data, it may be less flexible in handling more complex data structures with multiple relationships.
2. **Network Model:** The network model offers greater flexibility in representing complex data structures and relationships. With the ability to have multiple parent and child relationships for each record, it can handle a wider range of data scenarios.

4. Querying Capabilities:

1. **Hierarchical Model:** In the hierarchical model, queries typically follow a top-down approach, starting from the root and traversing down the tree to retrieve data. While it efficiently retrieves hierarchical data, it may become challenging to navigate the tree for more complex queries involving multiple relationships.
2. **Network Model:** The network model allows for more versatile queries due to its ability to represent many-to-many relationships. It enables more efficient retrieval of interconnected data, making it suitable for applications that require complex querying capabilities.

🕒 August 4, 2023

🏠 Artificial Intelligence

💡 causal consistency model

💡 consistency models in distributed systems

💡 data centric consistency model

💡 difference between data centric and client-centric consistency model

💡 strict consistency model

💡 weak consistency model

[ABOUT THE AUTHOR](#)

[LATEST POSTS](#)

Aishwarya Kurre

I work as a Data Science Ops at Pickl.ai and am an avid learner. Having experience in the field of data science, I believe that I have enough knowledge of data science. I also wrote a research paper and took a great interest in writing blogs, which improved my skills in data science. My research in data science pushes me to write unique content in this field. I enjoy reading books related to data science.

Radical ML

🕒 [AUGUST 31, 2022](#)

What is Cloud Computing ? Types, Facts, Benefits – A Complete Guide

🕒 [MARCH 1, 2023](#)

Statistical Analysis- Types, Methods & Examples

🕒 [MARCH 3, 2023](#)

How to Become an Artificial Intelligence (AI) Engineer?

🕒 [NOVEMBER 25, 2022](#)

Enter a keyword...



Recent Posts

[Explore data effortlessly with Python Libraries for \(Partial\) EDA: Unleashing the Power of Data Exploration](#)

🕒 DECEMBER 11, 2023

[Data Intelligence empowers informed decisions](#)

🕒 DECEMBER 4, 2023

[Consistency protocols in distributed systems ensure data coherence & reliability](#)

🕒 DECEMBER 4, 2023

[How can AI and data protection work together: unveiling the truth](#)

🕒 NOVEMBER 23, 2023

[AI's photo recovery for digital cameras](#)

🕒 NOVEMBER 22, 2023

Categories

Artificial Intelligence	48
Big Data	5
Business Intelligence	9
Career Path	68
Case Study	1
ChatGPT	3
Cheat Sheets for Data Scientists	1
Cloud Computing	6
Data Analysts	39
Data Engineering	4
Data Forecasting	2
Data Governance	4
Data Science	106
Data Visualization	1
Machine Learning	62
Microsoft Excel	4
News	1
Programming Language	6
Python	8
Python Programming	30
SQL	8
Uncategorized	10

Follow us on social media



COMPANY

[About Us](#)

[Success Stories](#)

[Verify Certificate](#)

[Contact Us](#)

COMMUNITY

[YouTube](#)

[Discord](#)

[Blog](#)

[Events](#)

OUR COURSES

[Foundation Course in Data Science \(For College Students\)](#)

[Foundation Course in Data Science \(For Professionals\)](#)

[Data Science Course for Kids](#)

[ML 1o1 – Introduction to Machine Learning](#)

[Job Assistance Program](#)

[Job Guarantee Program](#)

WORK WITH US

[Become an Instructor](#)

[Career](#)

© Pickl.AI 2023. All rights reserved

[Refund Policy](#)

[Cookie Notice](#)

[Privacy Policy](#)

[Terms of Use](#)