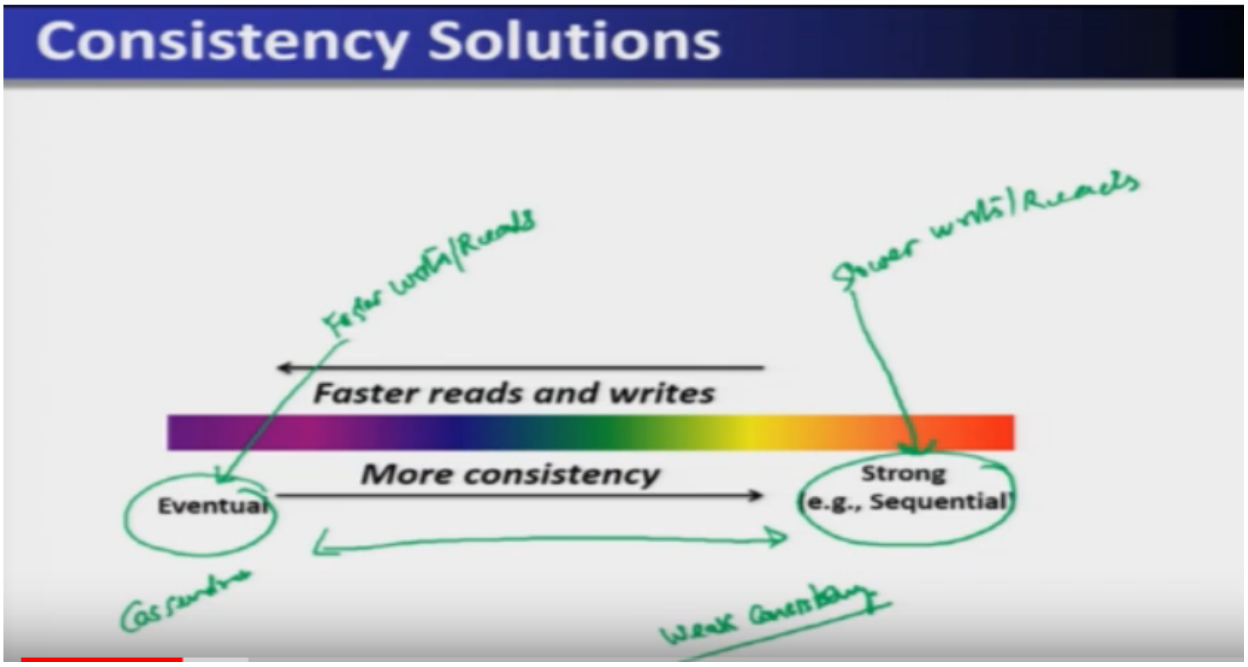
Lecture 15
Consistency Solutions

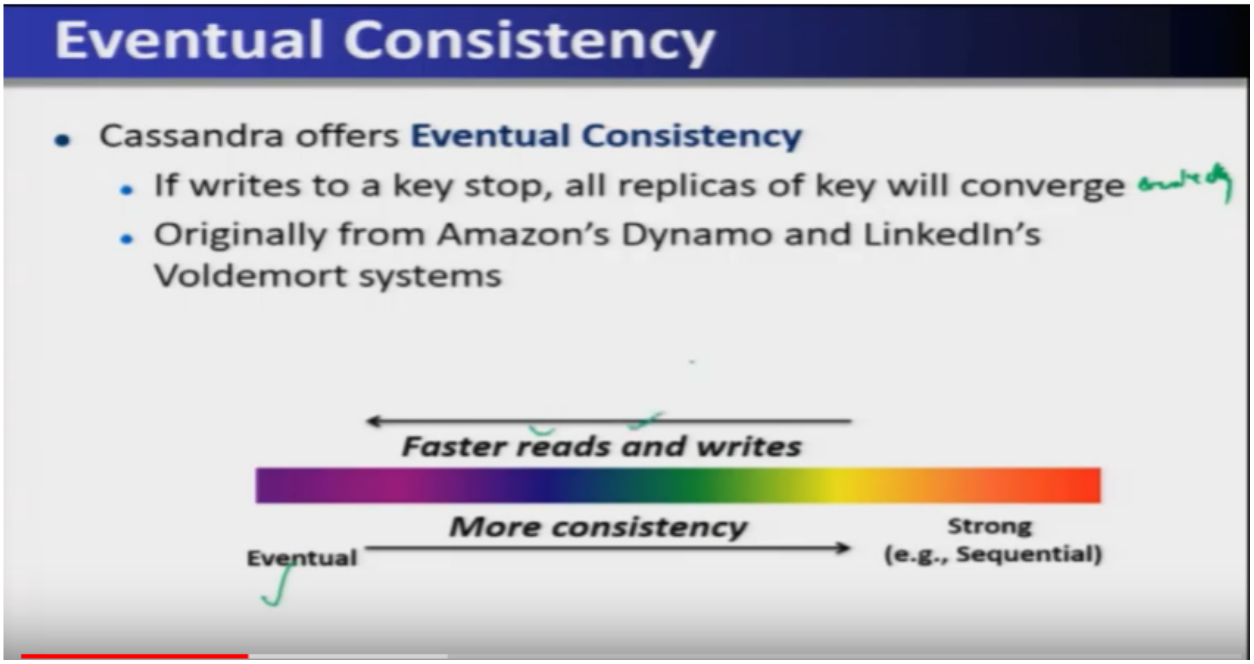
Consistency solutions

Refer slide time :(0:15)



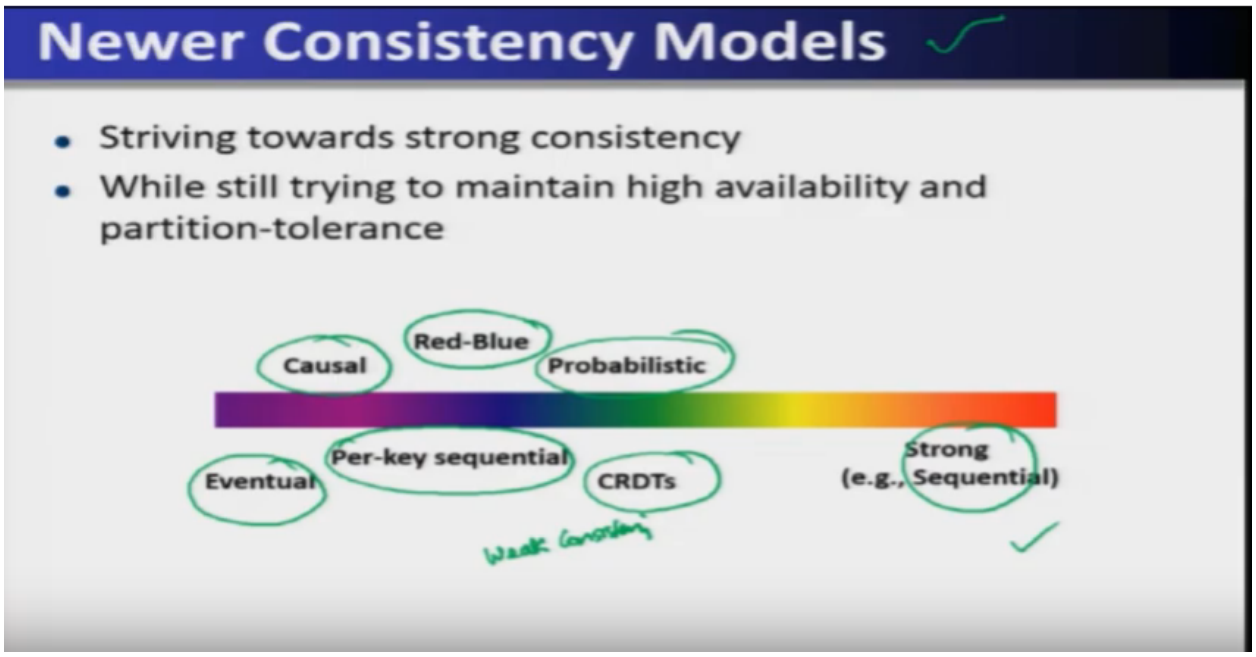
So, we will see the spectrum of different consistency solutions, which are available, under the weak consistency model. And we have seen that eventual consistency, which is supported by the Cassandra, is that one extreme level and strong consistency, at the other extreme level, in between what are the other consistency, models are there. So, before that we, we have to understand, when to use these consistency solutions, now if you want to do a faster read and writes, obviously the eventual consistency, is the good enough for the faster, faster writes and reads, similarly if we require, the achieve the high consistent high, consistency or strong consistency, then we will be having a slower, writes and reads. Now in between, we will see the other forms of consistency and the performance varies, from strong consistency, to the weak consistency that means we have to ensure, what kind of reads and write whether it is faster or it is slower, read and writes are required.

Refer slide time :(01:50)



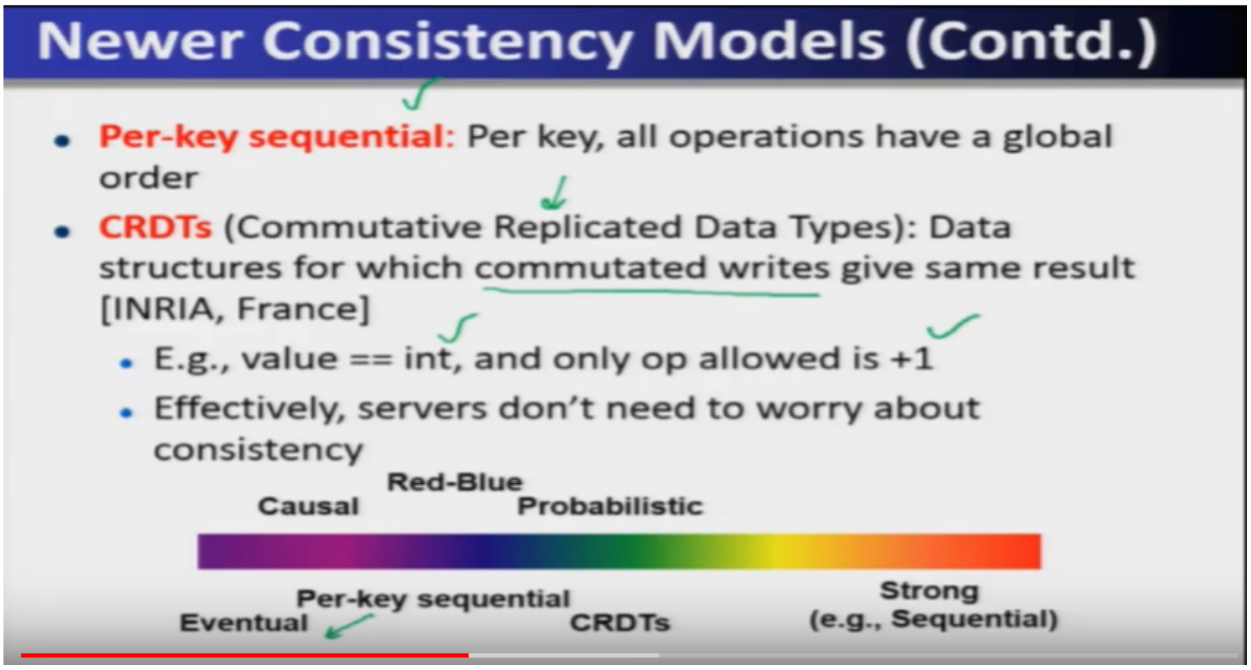
So, eventual consistency Cassandra, offers eventual consistency if, if writes to a key stops all the replicas of a key will converge eventually, originally this eventual consistency, was inspired by, inspired from a vengeance dynamo and LinkedIn's voldemort system, in into the Cassandra system into the Cassandra's design. So, here we have to see that strong eventual consistency she, supports faster reads and write operations.

Refer slide time :(02:34)



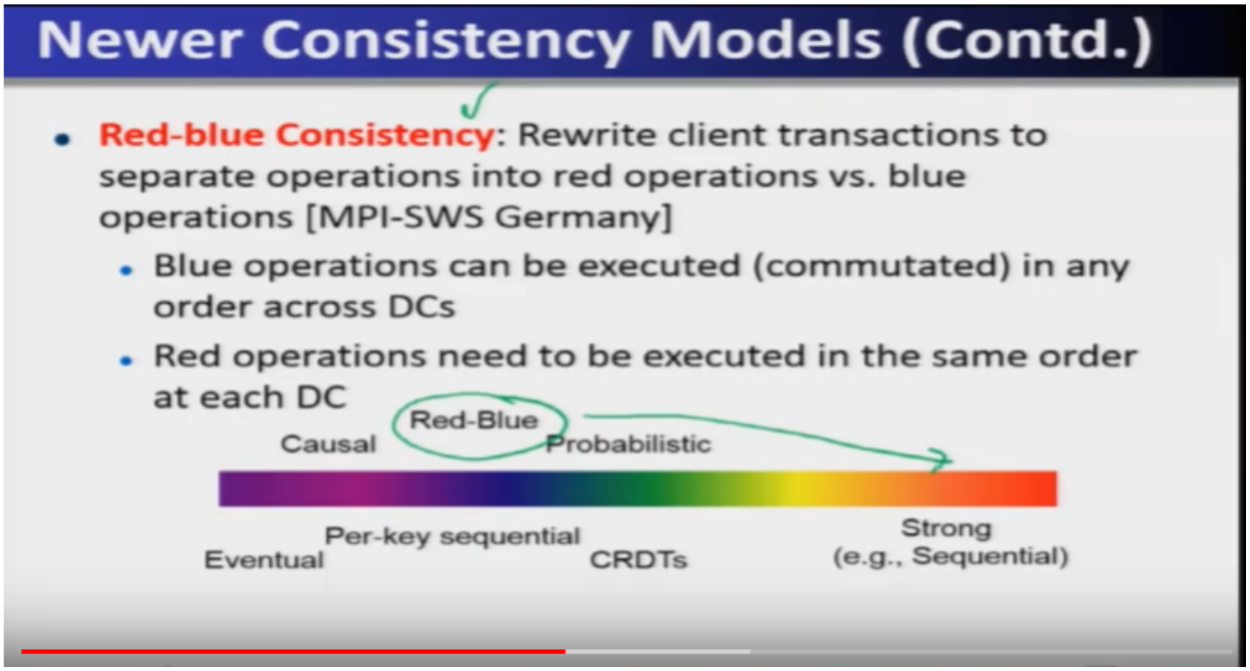
And let us see between the, the eventual consistency and between, the strong consistency, what are the other weak form of consistency, consistency available they are called,' Newer Consistency', model. So, is striving towards a strong consistency is, very much needed, but different applications, but while still trying to maintain high availability and a partition tolerance, forces to go for some weak consistency models. So, there are different models, which are shown over here, they are called,' Per Key Sequential', consistency model then, then red blue consistency model, then CRDTs and causal consistency model, probabilistic let us see this newer consistency model, how they are now trading the consistency and giving the,

Refer slide time :(03:37)



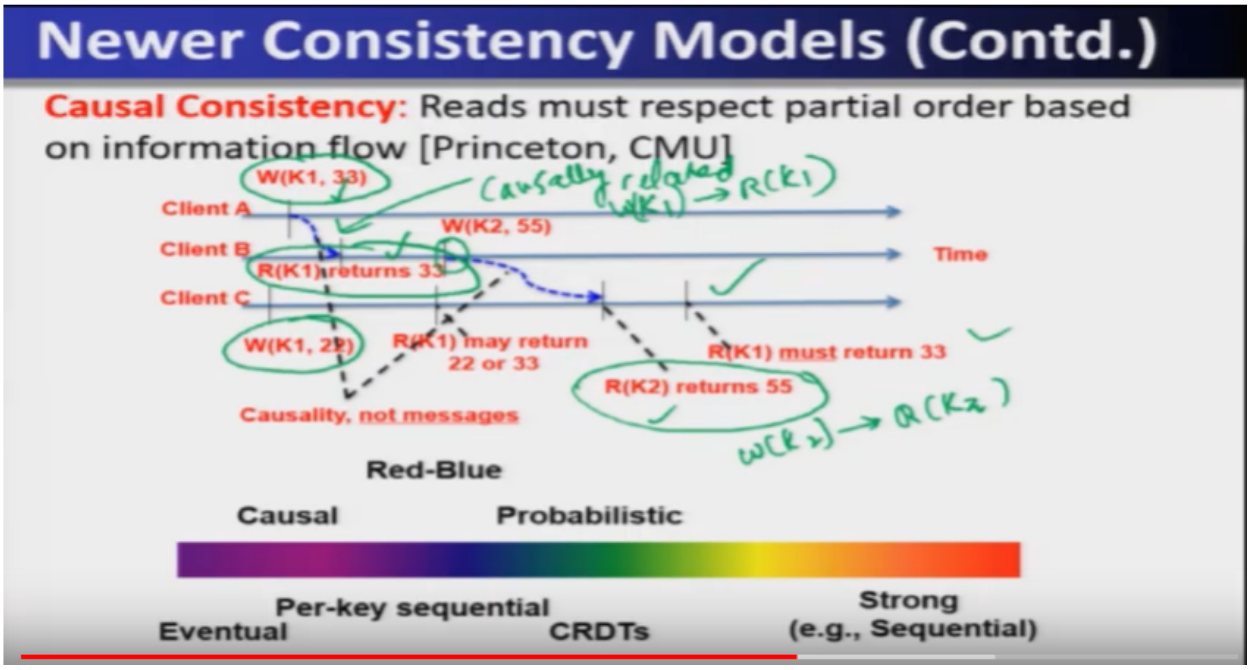
the performance required by the applications. So, as far as the, per key sequential consistency model, since that perky all the operations, have the global order. So, this ensures a performance, better than the eventual consistency, because it ensures the, the strong consistency notion, per key bases at global level. Now another type of consistency is called ,'CRDTs', that is commutative replicated data types, here the data structure for which the commutated writes, gives the same result this was given by the INRIA, France for example, if it is if the value is integer and only operation allowed is plus one. So, whether the to writes, whether one has happened before the other it doesn't makes, any difference why because it is only doing the plus operation, plus one operation, increment operation effectively server don't need have to worry about, the consistency if it is the commutative, operations commutating, write operations in CRDTs.

Refer slide time :(04:57)



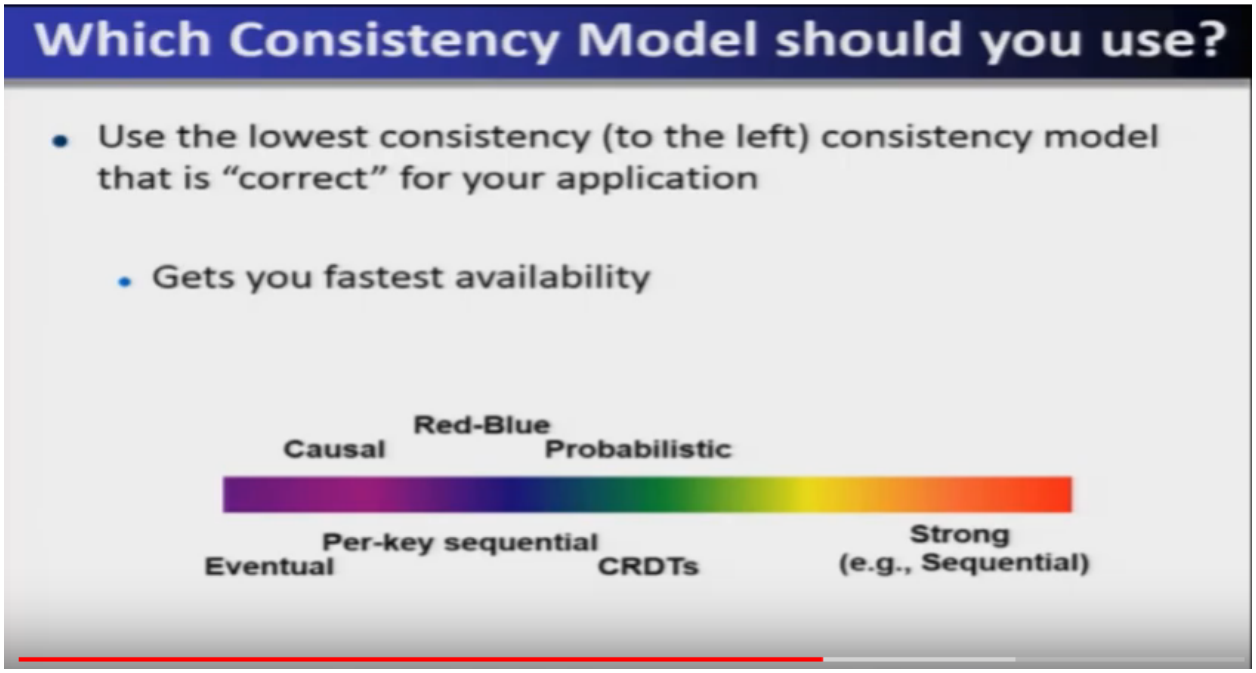
Now all the operations cannot be commutative. So, therefore in the NER in the, in the red-blue consistency, it will divide into two different type of operations, as per commutative and the other operations. So, it will rewrite the client's transaction two separate, operations into the red operations, that is vs. blue operations, blue operations can be executed, commutated in any order across datacenter, whereas the red operations, need to be executed in the same order in the data center. So, this classification, of these operations into the blue and red ensures, the red-blue consistency and therefore will make this particular, approach that is the red-blue consistency a much faster than eventual means, it will ensure a strong, much better consistency level, compared to the eventual consistency.

Refer slide time :(06:03)



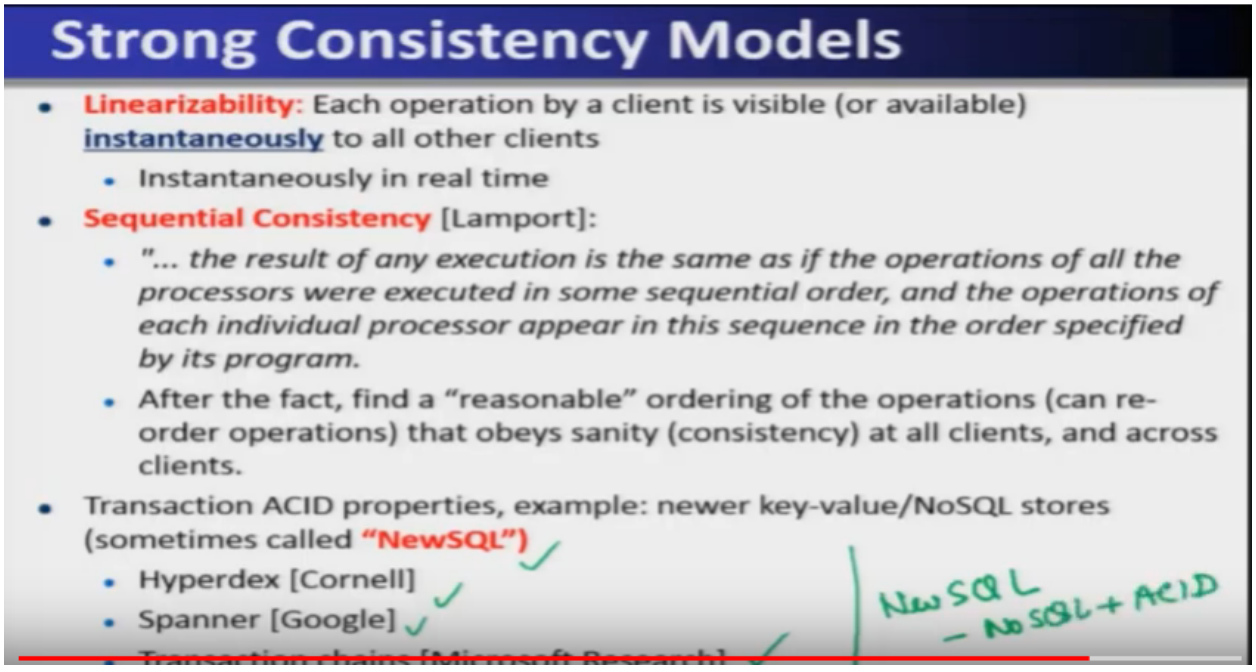
Now causal consistency, causal consistency deals, with the another weak form of consistency, which says that the deed must respect the partial order, based on the information flow, for example here we have seen that the client a writes, the key 1 as 33 whereas, the client B will read this value of K, after some point of time k1. So, it should return the value 33 so, obviously this right of a client 1, is causally related. So, client this key writing of key k1, is causally related to the read operation of K, key one and this there is no message exchange, but internally we have to see this kind of causal dependency, exist similarly, you see that there is a causal flow similarly, between client B, which reads between client B and client C, which reads this value of K key 2, which is written by the client B, at this point of time also are causally related. So, that means right of K key2, of cake of a key K 2, is happened before the read of key K 2, in this case. So, this particular read of K 2, returns the value, which is written by the client B on key K 2 and similarly when, the client C performs, a read of K 1 it must return using this causal path, the value 33which, is written by the client a not, the other value, which is the old value, which is written s 22. So, the causality where the messages, are not exchanged even then the causality is protected, then it is called,' Causal Consistency'. So, this kind of consistency also is a very fast, notion of providing read and write, although they are ensuring some, some strong notion of consistency.

Refer slide time :(08:34)



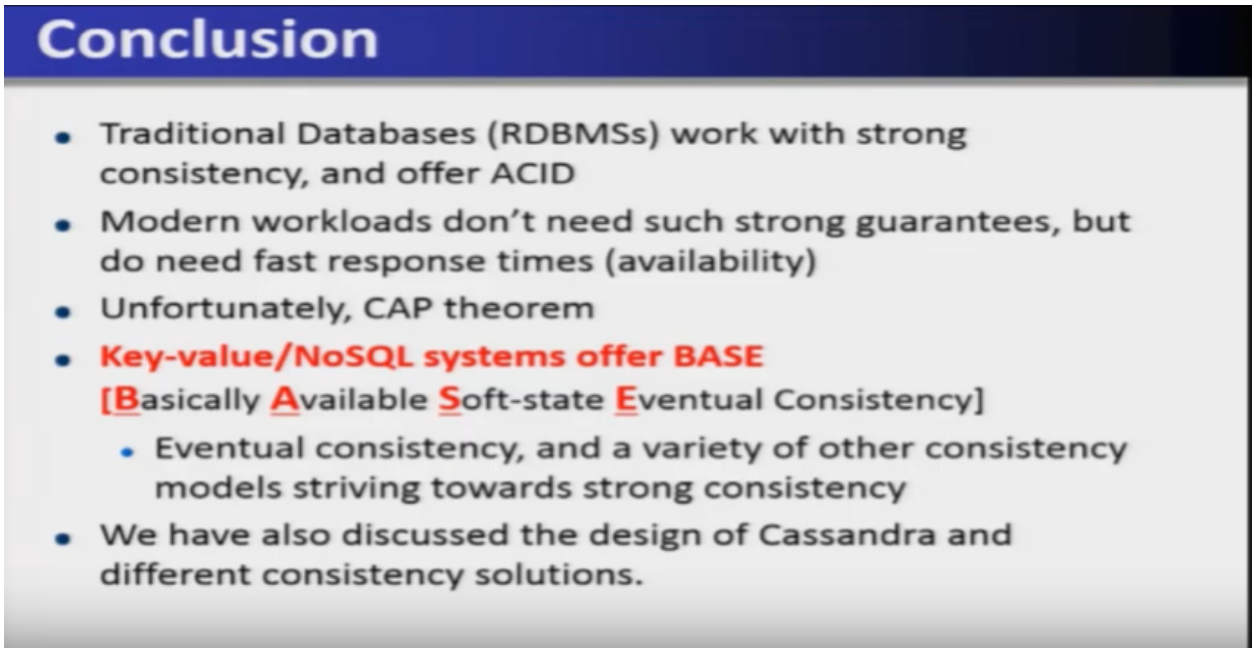
So, which model, of consistency should be used, souse thus the lowest consistency to the left, that is the consistency model that is the correct for your application, gets you the faster availability in this particular scenario.

Refer slide time :(08:49)



Now what is the strong consistency morals, let us quickly review them. So, linearizability is the strong consistency model, where each operation, by the client is visible, to all other it just like storing the data, on a single system, where whenever one data is modified it is available to all the clients instantaneously. And in real time, if that is there then it is called,' Linearizability', is very difficult to achieve in the distributed systems, another form of strong consistency, is called,' Sequential Consistency', which says that the result of any execution, is the same as if the operations of all the processors, were executed in some sequential order and operations of each individual processor appeared, in this sequence in the order is specified by, by its program, meaning to say that although it is not linearizability approach, but the sequential consistency, says that some order, which is which is to be executed in some sequential order, across all the clients, then if it is unsure then it is called,' Sequential Consistency', it is given by the lamp or so, after the fact find reasonable ordering of operations, can be reorder operations that obeys the consistency, at all the clients across the clients. So, this is again, again another form of strong constraint that is a sequential consistency that that means all the clients will see the same, kind of the operations, by all the clients called sequential consistency. Now transactions, which are basically following the acid properties and whether the newer, key value, stores that is no sequel store sometimes called,' New Sequel', which also ensures the acid transaction properties. So, the nowadays, the application which required, also the acid property to be followed, besides all the properties, which we have seen in the no sequel stores, if both are in addition, to that acid is also required and they are called a,' New SQL Systems'. So, for example new SQL is now a days is also available, in the form of high products given by the Cardinal and the Spain or a database management system, using by the Google and transaction chain by the Microsoft research and they call it as new SQL which supports, besides whatever is there in no SQL, plus acid properties, to support the transactions.

Refer slide time :(11:53)



So, in conclusion, we have to see that the traditional, RDBMS work with a strong consistency model and offer the acid properties and whereas the modern workloads don't need, such a strong guarantees, but do need fast response that is availability, unfortunately the cap theorem comes in to an effect, which says that out of consistency availability and partition, you have to choose two of them. That we have seen, in this part of the discussion. So, no sequel system which are also called, 'Key Value Store', offers the base property that is called basically available, soft state eventual consistency. So, eventual consistency and a variety of other consistency models are striving, towards a strong consistency, we have also discussed the design, details of Cassandra system and also covered different consistency models, which are the newer consistency models, which are available, for the newer kind of systems. Thank you.