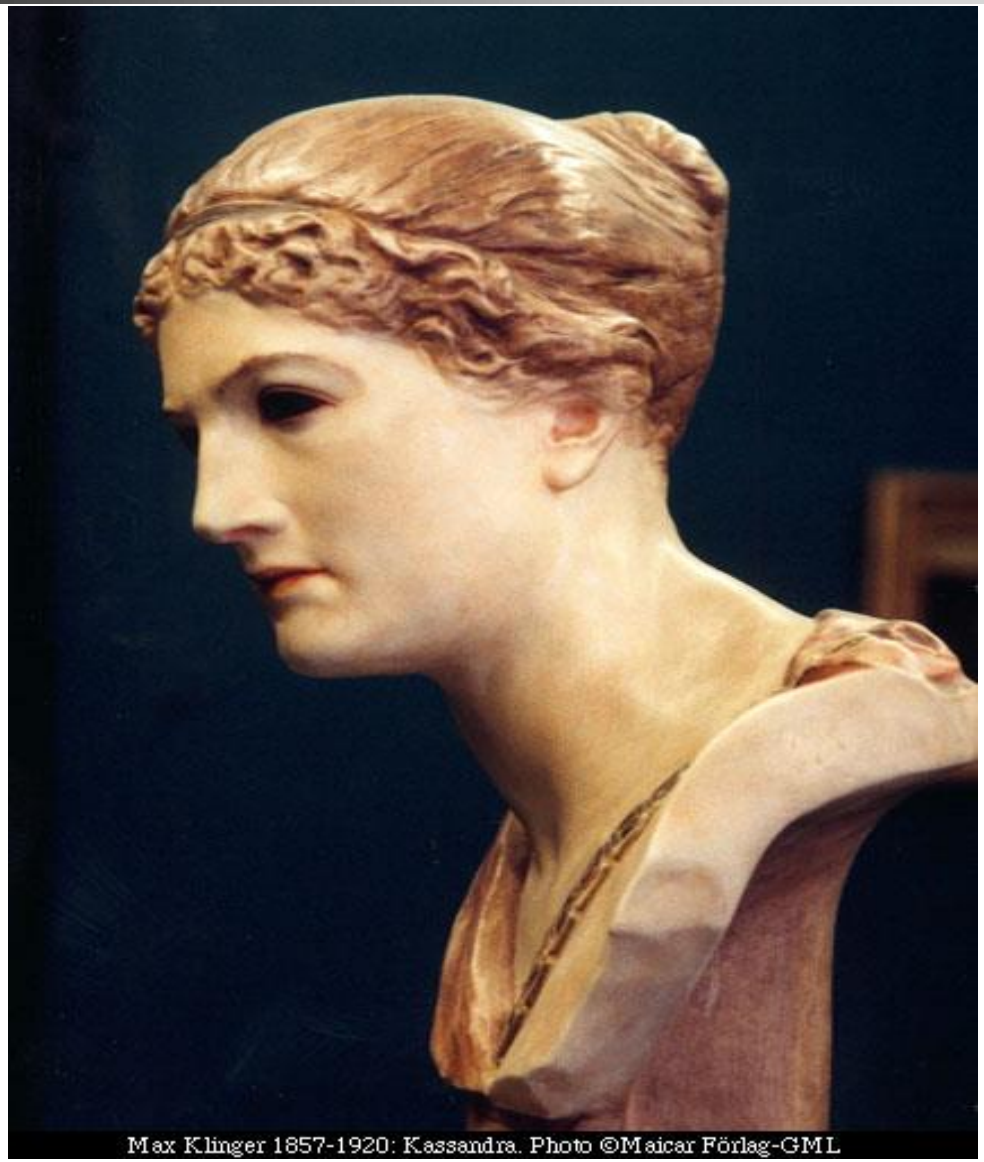
VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui

Cassandra Consistency Levels

Lecturer* : *Dr. Pavle Mogin

SWEN 432
Advanced Database Design and Implementation

Cassandra The Fortune Teller



Plan for Consistency Leveles

- Configuring Data Consistency
  - Prolog
  - Write and Read Consistency Levels
- Write Requests
  - Examples of Write Consistency Levels
- Read Requests
  - Examples of Read Consistency Levels

Prolog

- Consistency refers to how up-to-date and synchronized a data object is on all of its replicas
- Cassandra extends the concept of eventual consistency by offering tuneable consistency
  - For any given read or write operation, the client application decides how consistent the requested data should be
  - Available consistency ranges from eventual, via strong to even strict (the highest)
  - As the consistency is higher, the throughput is lower
  - The consistency can be set globally (on a cluster or data centre basis), or per individual operation basis
  - The default consistency level is eventual
  - On individual operation basis, consistency level is defined using cqlsh command `CONSISTENCY`

```
cqlsh> consistency <level>;
```

Write Requests

- A client can ask any node in a cluster to coordinate a write
  - Each node has information about all other nodes
- The coordinator sends a write request to *all* replicas that own the row being written
- As long as all replica nodes are up and available, they will get the write
- If a replica misses a write, Cassandra will make the row consistent later using one of its built-in repair mechanisms:
  - Hinted handoff,
  - Read repair, or
  - Anti-entropy node repair

Write Consistency Levels

- The write consistency level specifies the number of replicas on which a write must succeed before the coordinator returns an acknowledgment to the client application
- Success means that the data was written to the commit log and the memtable
- The write consistency levels:
  - ANY, ALL, QUORUM, EACH_QUORUM, LOCAL_QUORUM, LOCAL_SERIAL, SERIAL, LOCAL_ONE, ONE, TWO, and THREE
- Even at consistency level ONE or LOCAL_QUORUM, the write is still sent to all replicas for the written key, even to replicas in other data centres

Write Consistency Level ANY

- ANY
  - Description:
    - A write must be written to at least one node
    - If all replica nodes for the given row key are down, the write can still succeed after a hinted handoff has been written
    - If all replica nodes are down at write time, an ANY write is not readable until the replica nodes for that row have recovered
  - Features:
    - Provides low latency and a guarantee that a write never fails
    - Delivers the lowest consistency and highest availability compared to other levels
- Example:

```
blogs> consistency any;
Consistency level set to ANY.
```

Write Consistency Level ALL

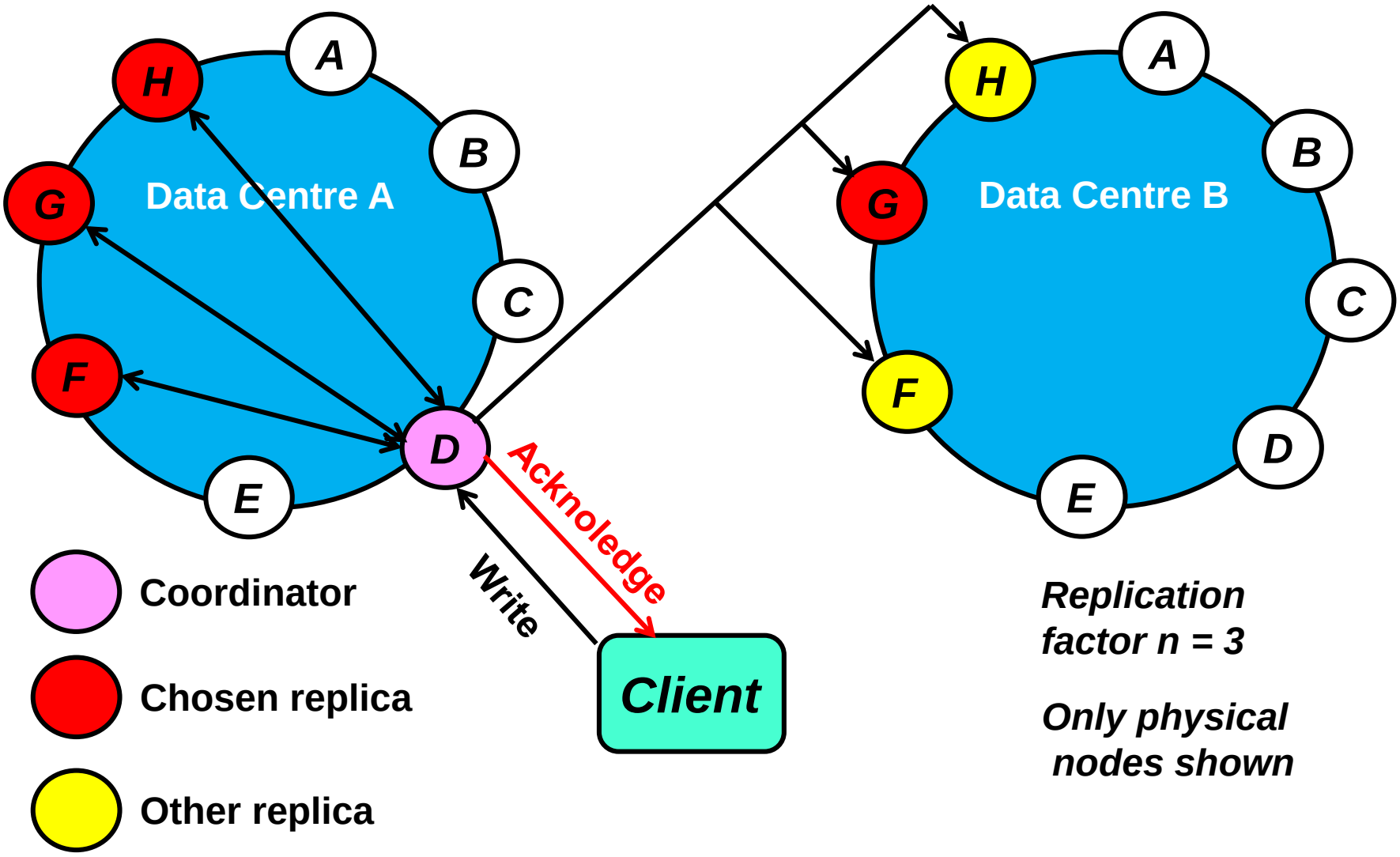
- ALL
  - Description:
    - A write must be written to the commit log and memtable on all replica nodes for that row in the cluster
  - Features:
    - Provides the strict (highest) consistency and the lowest availability of any other level
- Example:

```
blogs> consistency all;
Consistency level set to ALL.
```

Consistency QUORUM and EACH_QUORUM

- QUORUM
  - Description:
    - A write must be written to the commit log and memtable on a quorum of replica nodes (regardless of the number of data centres)
  - Features:
    - A prerequisite for the strong consistency
- EACH_QUORUM
  - Description:
    - A write must be written to the commit log and memory table on a quorum of replica nodes in *all* data centres
  - Features:
    - Used in multiple data centre clusters to provide for the strong consistency level in each data centre

Two DC Cluster and QUORUM



Consistency Level LOCAL_QUORUM

- LOCAL_QUORUM
  - Description:
    - A write must be written to the commit log and memtable on a quorum of replica nodes in the same data center as the coordinator node
  - Features:
    - A prerequisite for strong consistency
    - Used in multiple data centre clusters with a rack-aware replica placement strategy (Network Topology Strategy) and a properly configured snitch
    - Fails when using Simple Strategy
    - Used to maintain consistency locally (within a single data centre)
    - Avoids latency of inter-data centre communication

```
blogs> consistency local_quorum;
Consistency level set to LOCAL_QUORUM.
```

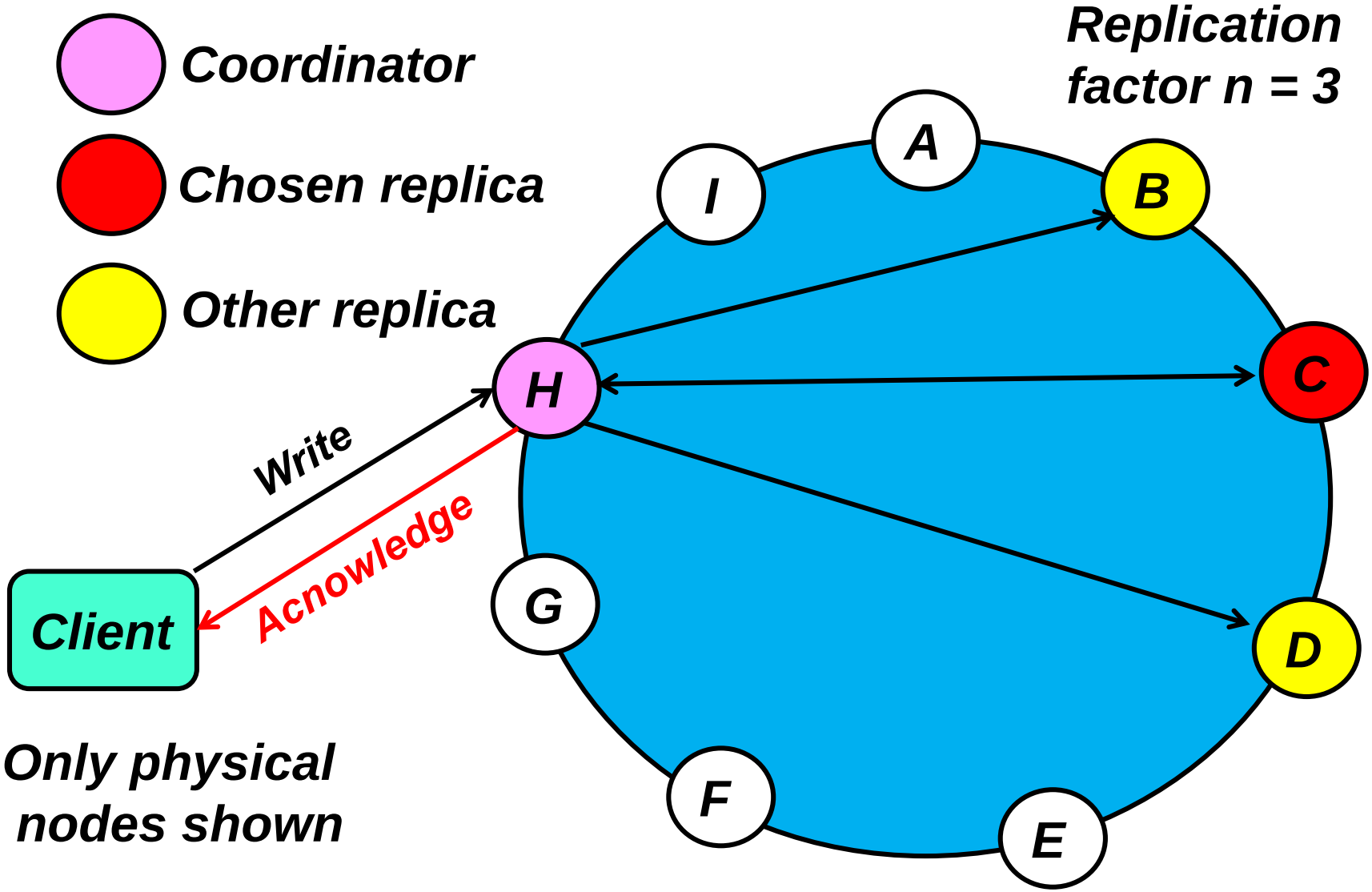
Consistency LOCAL_SERIAL and SERIAL

- LOCAL_SERIAL
  - Description:
    - A write must be written conditionally to the commit log and memory table on a quorum of replica nodes in the same data centre
  - Features:
    - Used to achieve linearizable consistency for lightweight transactions by preventing unconditional updates
- SERIAL
  - Description:
    - A write must be written conditionally to the commit log and memory table on a quorum of replica nodes
  - Features:
    - Same as for LOCAL_SERIAL

Consistency ONE and LOCAL_ONE

- ONE
  - Description:
    - A write must be written to the commit log and memory table of at least one replica node
  - Features:
    - Guaranties the eventual consistency only (unless the ALL consistency level is used for reading)
    - The replica node closest to the coordinator node that received the request, serves the request (unless the dynamic snitch determines that the node is performing poorly and routes it elsewhere)
- LOCAL_ONE
  - Description:
    - A write must be sent to all, and successfully acknowledged by at least one replica node in the local data centre
  - Features:
    - Avoids the latency induced by inter data centre traffic

A Single DC Cluster and ONE



Types of Read Requests

- There are two types of read requests that a coordinator node can send to a replica:
  - A direct read request
  - A background read repair request
- The number of replicas contacted by a direct read request is determined by the consistency level specified by the client
- Background read repair requests are sent to any additional replicas that did not receive a direct request
- Read repair requests ensure that the requested row is made consistent on all replicas

The Direct Read Request

- A client can send a read request to any node in the cluster
  - Each node contains information about all the other nodes in the cluster
  - The contacted node becomes the coordinator for the read request
- The coordinator node first contacts the replicas specified by the consistency level
- The coordinator sends these requests to the replicas that are currently responding the fastest
  - The nodes contacted respond with the requested data
  - The replica that has the most recent data (based on the timestamp) is used by the coordinator to forward the result back to the client

Read Path Details (1)

- The coordinator node uses various Cassandra mechanisms to execute a read request:
  - The `ReadExecutor` determines the replicas (endpoints) to read from by processing the row (partition) key with the `ReplicationStrategy` for the keyspace
  - Endpoints are filtered to contain only those that are currently up/alive
  - If there are not enough live endpoints to meet the consistency level, an `UnavailableException` reponse is returned
  - Endpoints are sorted by "proximity"
  - With a `SimpleSnitch`, proximity directly corresponds to proximity on the token ring
  - With implementations based on `PropertyFileSnitch`, endpoints in the same rack, or if not available then in the same data center are always considered "closer"

Read Path Details (2)

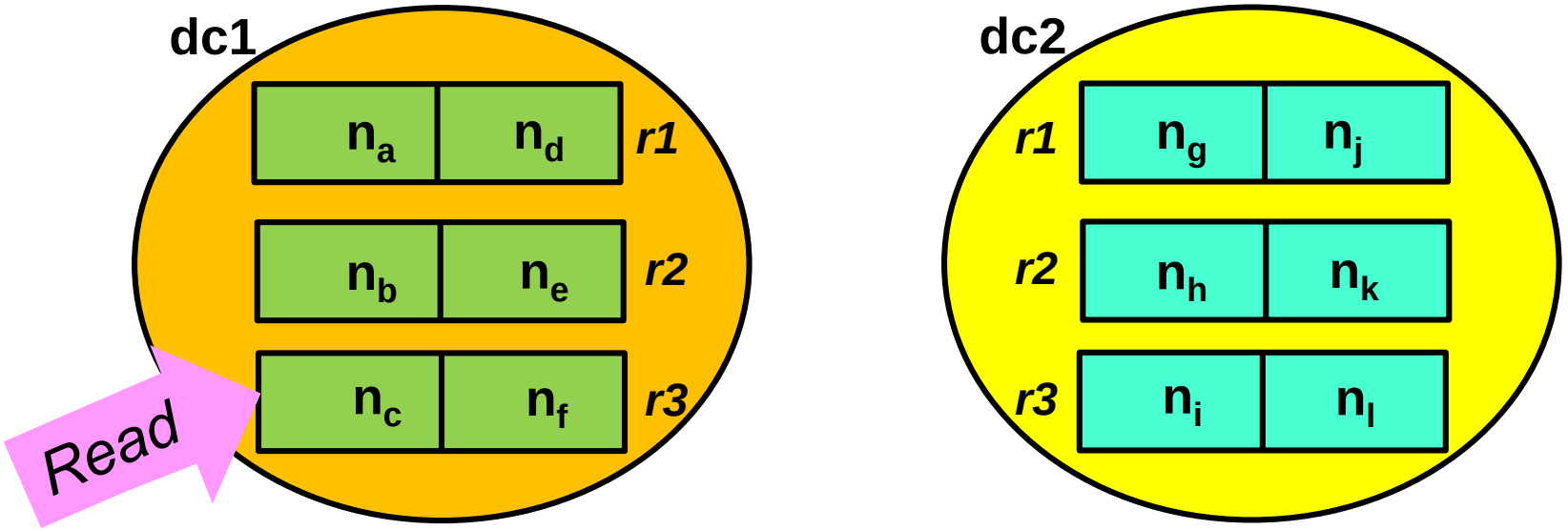
- A further filtering of the “closest” node is done:
  - By using information from a file called `DynamicSnitch` that contains data about the performance of other nodes
    - An effort to avoid routing more traffic to slower endpoints
- The closest node (as determined by proximity sorting discussed) will be sent a command to perform an actual data read
  - As required by consistency level, additional nodes may be sent digest commands, asking them to perform the read locally but send back the digest only
  - For example, at replication factor 3 a read at consistency level QUORUM would require one digest read in addition to the data read request sent to the closest node
  - A node asked for a digest read, performs a normal read locally, but sends only a hash of data read over the network

Read Path Details (3)

- On the data node a read request is processed in the way described in the Cassandra Storage Engine lecture by reading from the memTable and SSTables and returning the result to the coordinator node
- Back on the coordinator node, responses from replicas are handled:
  - If a replica fails to respond before a configurable timeout, a `ReadTimeoutException` is raised
  - If responses (data and digests) do not match, a full data read is performed against the contacted replicas in order to guarantee that the most recent data is returned
- Once retries are complete and digest mismatches resolved, the coordinator responds with the final result to the client

A Node Proximity Example

- Assume a cluster spans two data centers and has six nodes in each.
- Nodes are stored in racks according to the picture below.



- The node n_c receives a client request to read data under consistency level ONE. Data to read are replicated on nodes n_d, n_e, and n_f in dc1 and on nodes n_j, n_k, and n_l in dc2.

- Which node is the coordinator node n_c going to send a direct read request, assuming all nodes are performing equally good?

Read Consistency Levels

- The read consistency level specifies how many replicas must respond to a read request before returning data to the client application
- The read consistency levels:
  - ALL, QUORUM, EACH_QUORUM, LOCAL_QUORUM, LOCAL_SERIAL, SERIAL, ONE, LOCAL_ONE, TWO, and THREE
- Cassandra checks the specified number of replicas for the most recent data, based on the timestamp, to satisfy the read request

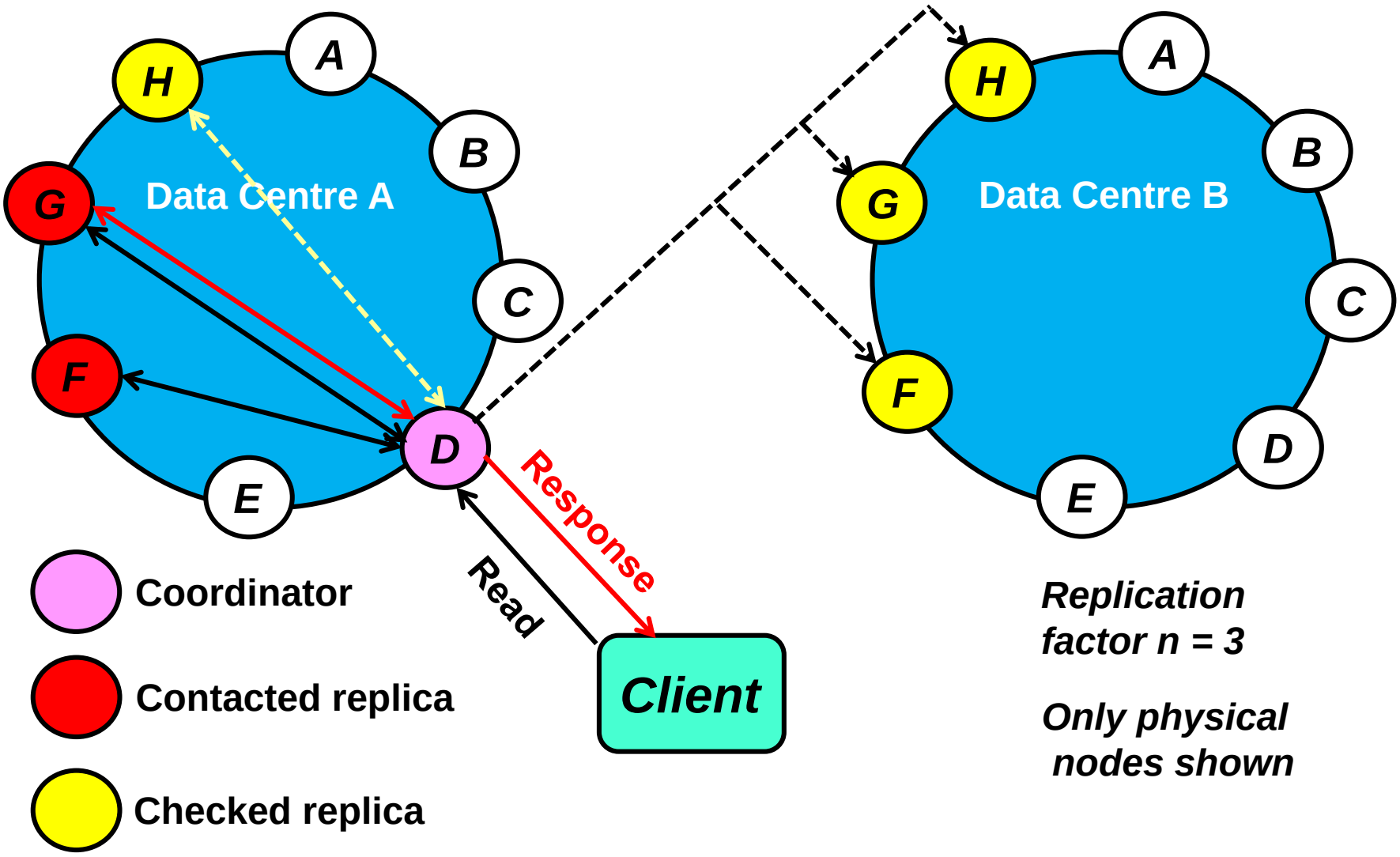
Read Consistency Level ALL

- ALL
  - Description:
    - Returns the record with the most recent timestamp after all replicas have responded
    - The read operation will fail if a replica does not respond
  - Features:
    - Provides the highest (strict) consistency of all levels and the lowest availability of all levels (since the latency is the greatest)

Read Consistency LOCAL_QUORUM

- LOCAL_QUORUM
  - Description:
    - Returns the record with the most recent timestamp once a quorum of replicas in the current data centre as the coordinator node has reported
  - Features:
    - A prerequisite for the strong consistency
    - Used in multiple data centre clusters with a rack-aware replica placement strategy (Network Topology Strategy) and a properly configured snitch
    - Fails when using Simple Strategy
    - Avoids latency of inter-data centre communication

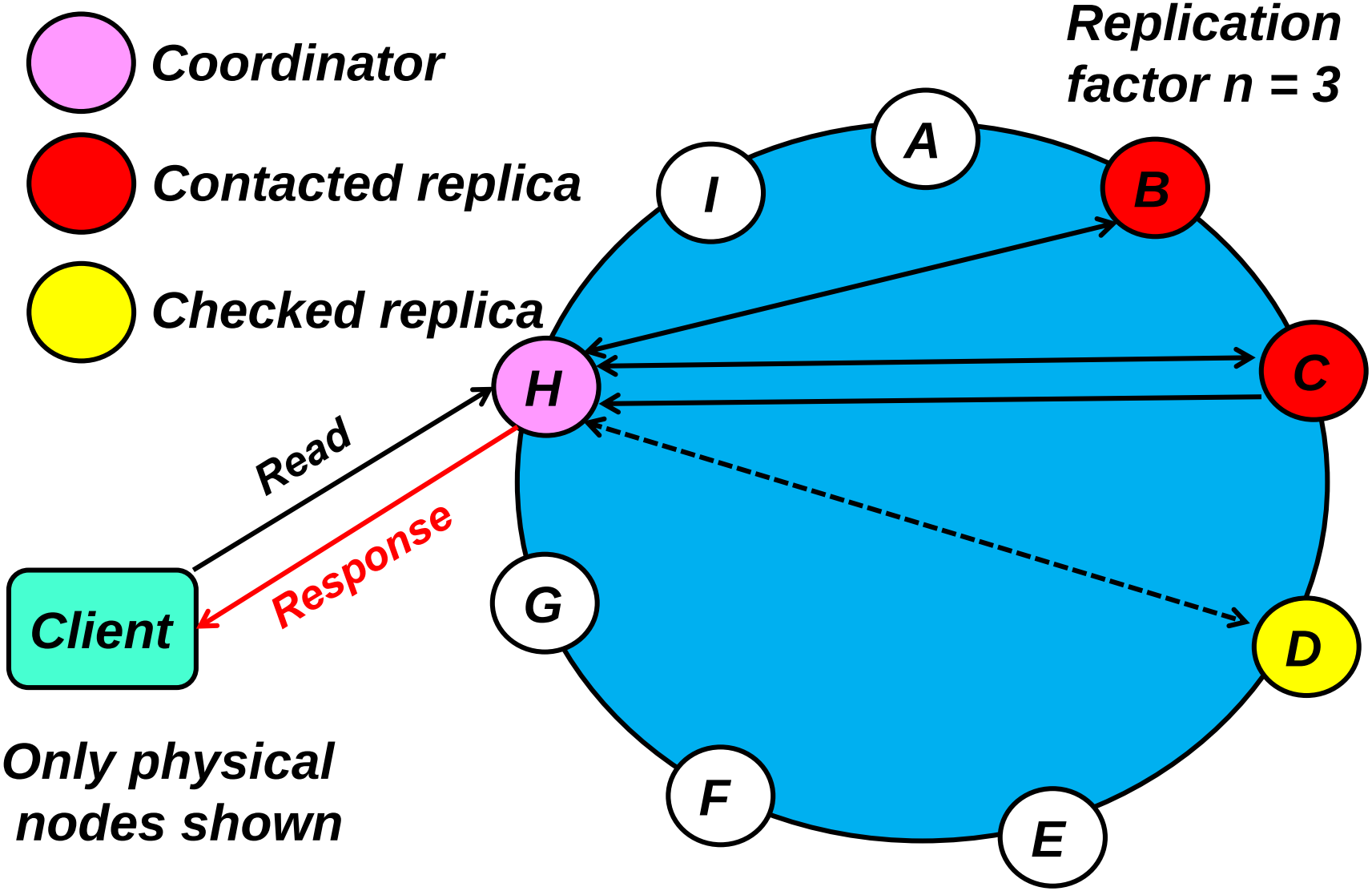
Two DC Cluster and LOCAL_QUORUM



Consistency QUORUM and EACH_QUORUM

- QUORUM
  - Description:
    - Returns the record with the most recent timestamp after a quorum of replicas has responded regardless of data centre
  - Features:
    - A prerequisite for the strong consistency
- EACH_QUORUM
  - Description:
    - Returns the record with the most recent timestamp once a quorum of replicas in each data centre of the cluster has responded
  - Features:
    - Used with Network Topology Strategy
    - Fails with Simple Strategy

A Single DC Cluster and QOURUM



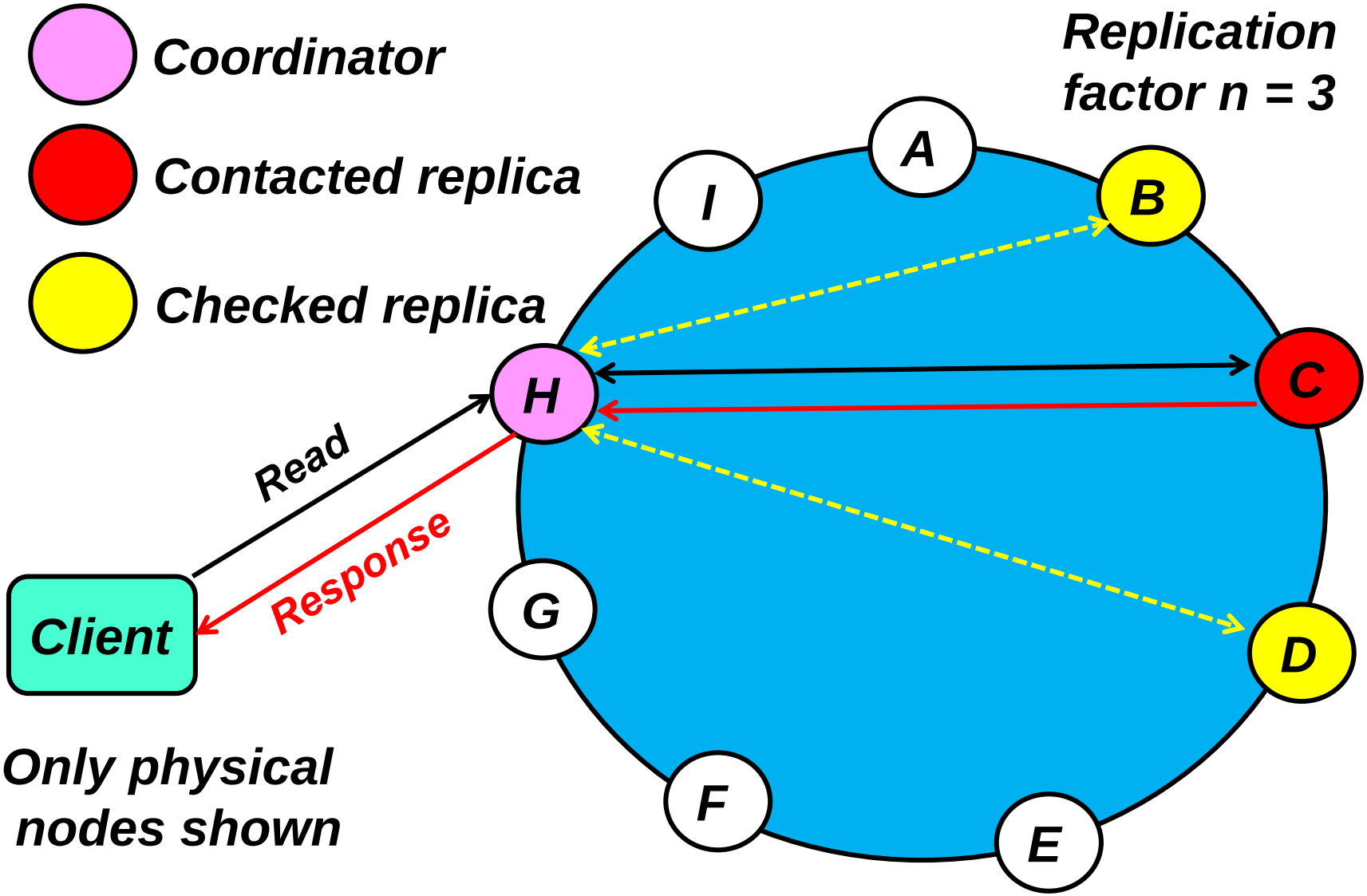
Consistency SERIAL and LOCAL_SERIAL

- SERIAL
  - Description:
    - Allows reading the current (and possibly uncommitted) state of data without proposing a new addition or update
    - If a SERIAL read finds an uncommitted transaction in progress, it will commit the transaction as part of the read.
  - Features:
    - Used with Light Weight Transactions
- LOCAL_SERIAL
  - The same as SERIAL only confined to the data centre of the coordinator node

Consistency ONE and LOCAL_ONE

- ONE
  - Description:
    - Returns a response from the closest replica, as determined by the snitch
    - By default, a read repair runs in the background to make the other replicas consistent
  - Features:
    - Provides the highest availability of all the levels if you can tolerate a comparatively high probability of stale data being read
    - The replicas contacted for reads may not always have the most recent write
- LOCAL_ONE
  - Description:
    - Returns a response from the closest replica, as determined by the snitch, but only if the replica is in the local data centre
    - Features:
    - Avoids the latency induced by inter data centre traffic

A Single DC Cluster and ONE



QUORUM in Multi DC Clusters

- The quorum *q* in a multi data centre cluster is

$$q = \lfloor sum\_of\_rep \quad lication\_f \ actors/2 \rfloor + 1$$

- Example:
  - Assume a three data centre cluster where each data centre has a replication factor of $n = 3$, then:
    - Under the consistency level QOURUM, $q = 5$ (regardless of data centre),
    - Under the consistency level LOCAL_QOURUM, $q = 2$,
    - Under the consistency level EACH_QOURUM, $q_i = 2$, for i in {*1*, *2*, *3*}, (at least *2* nodes from each data centre has to respond, so, *6* in total)

Simple Consistency Examples

- Eventual Consistency

  use blogs;

  insert…;

  read…;

  – Why is the consistency eventual?

- Strong Consistency

  use blogs;

  consistency quorum;

  insert…;

  read…;

  – Why is the consistency strong?

Question for You

- Question for you:
  - Assume someone issues the following CQL statements:

    use blogs;

    consistency quorum;

    insert…:

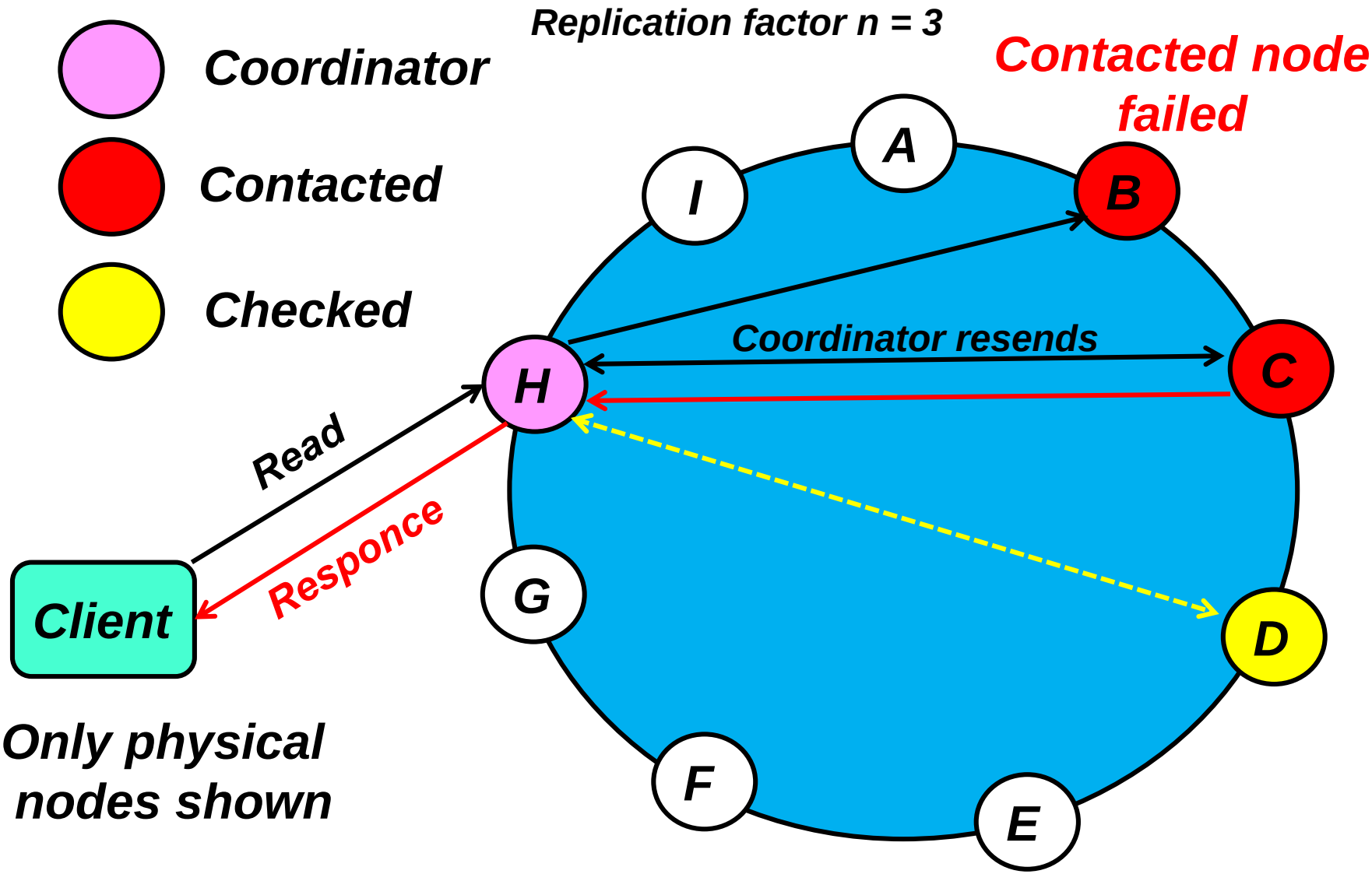
    consistency one;

    read…;
- What is the consistency level of data read?
  - Strong?
  - Eventual?
- Why eventual?

The Rapid Read Protection

- Rapid read protection (RRP) allows Cassandra to still deliver read requests when the originally selected replica nodes are either down or taking too long to respond
- If the table has been configured with the `speculative_retry` property, the coordinator node for the read request will retry the request with another replica node if the original replica node exceeds a configurable timeout value to complete the read request

RRP, CONSISTENCY ONE



Summary

- A client can issue a write or read request to any node in a cluster, since each node has information about partitions of all other nodes in the cluster
  - The contacted node becomes the coordinator
- Write consistency levels are:
  - ANY, ALL, QUORUM, EACH_QUORUM, LOCAL_QUORUM, LOCAL_SERIAL, SERIAL, LOCAL_ONE, ONE, TWO, and THREE
  - The coordinator sends a write request to all replicas
  - The write consistency level determines how many replicas have to acknowledge
- Read consistency levels are:
  - ALL, QUORUM, EACH_QUORUM, LOCAL_QUORUM, LOCAL_SERIAL, SERIAL, ONE, LOCAL_ONE, TWO, and THREE
  - The consistency level determines to how many replicas the coordinator sends the direct read request
- The consistency level of a request is defined by the client