In Amazon DynamoDB, **Read Capacity Units (RCUs)** and **Write Capacity Units (WCUs)** represent the throughput capacity allocated for reading and writing data to a DynamoDB table. DynamoDB allows you to provision the amount of capacity required for your workloads, ensuring predictable performance by charging based on the number of RCUs and WCUs you allocate. Here's a detailed breakdown of RCUs and WCUs:

1. Read Capacity Unit (RCU)

- **Definition**: A **Read Capacity Unit (RCU)** is a unit of throughput for read operations in DynamoDB.
- **One RCU allows**:
    - **Strongly Consistent Reads**: One RCU allows you to perform **one strongly consistent read per second** for an item up to **4 KB** in size.
    - **Eventually Consistent Reads**: One RCU allows you to perform **two eventually consistent reads per second** for an item up to **4 KB** in size.
    - **Transactional Reads**: For transactional operations (which ensure ACID compliance), one RCU allows **one transactional read per second** for an item up to **4 KB** in size.
- **How It Works**: If an item is larger than 4 KB, more RCUs are consumed. For example:
    - A 6 KB item would require **2 RCUs** for a strongly consistent read or **1 RCU** for an eventually consistent read.
- **Use Case**: RCUs are important for applications with frequent read operations. The number of RCUs you provision must be sufficient to handle the read load, ensuring the table can meet query or scan requests without throttling.

2. Write Capacity Unit (WCU)

- **Definition**: A **Write Capacity Unit (WCU)** is a unit of throughput for write operations in DynamoDB.
- **One WCU allows**:
    - **One write per second** for an item up to **1 KB** in size.
    - **Transactional Writes**: For transactional operations (which ensure ACID compliance), one WCU allows **one transactional write per second** for an item up to **1 KB** in size.
- **How It Works**: If an item is larger than 1 KB, more WCUs are required. For example:
    - A 2.5 KB item would require **3 WCUs** for a write operation (since it's rounded up to the next KB).
- **Use Case**: WCUs are important for applications with frequent write operations, including inserts, updates, and deletes. Sufficient WCUs must be provisioned to handle the write throughput, ensuring smooth write performance without throttling.

3. Understanding 5 RCU and 5 WCU

- **5 RCU**:
    - For **strongly consistent reads**, 5 RCUs allow **5 reads per second** for items up to **4 KB** each.
    - For **eventually consistent reads**, 5 RCUs allow **10 reads per second** for items up to **4 KB** each.
    - If the item size exceeds 4 KB, more RCUs will be consumed. For example, with a 6 KB item, 5 RCUs allow:
        - 2 reads per second for strongly consistent reads (each 6 KB item consumes 2 RCUs).
        - 5 reads per second for eventually consistent reads (each 6 KB item consumes 1 RCU).
- **5 WCU**:
    - 5 WCUs allow **5 writes per second** for items up to **1 KB** each.
    - If the item size exceeds 1 KB, more WCUs are consumed. For example, with a 2 KB item, 5 WCUs allow:
        - 2 writes per second (since each 2 KB item consumes 2 WCUs).

4. Provisioned vs. On-Demand Capacity Modes

- **Provisioned Capacity**: In this mode, users specify the exact number of RCUs and WCUs they need. This is suitable for predictable workloads where traffic patterns and usage are stable. If your workload exceeds the provisioned capacity, requests may be throttled unless you're using auto-scaling.
- **On-Demand Capacity**: DynamoDB automatically scales to accommodate your workload without requiring you to provision RCUs or WCUs. This is ideal for unpredictable or variable workloads where it's hard to estimate capacity needs upfront. Users pay only for the read and write operations they perform.

5. Throttling

- If the number of read or write requests exceeds the provisioned RCUs or WCUs, DynamoDB throttles those operations, returning a `ProvisionedThroughputExceededException`. To avoid this, ensure that RCUs and WCUs are provisioned adequately for your workload, or use auto-scaling.

6. Best Practices

- **Right-Size Provisioning**: Monitor your application’s workload using CloudWatch metrics to understand the read and write patterns, and provision RCUs and WCUs accordingly to avoid over-provisioning (which increases costs) or under-provisioning (which leads to throttling).

- **Use Eventually Consistent Reads When Possible**: If your application can tolerate slightly stale data, using eventually consistent reads can double the read throughput for the same number of RCUs, reducing costs.
- **Auto-Scaling**: Use DynamoDB's auto-scaling feature to automatically adjust RCUs and WCUs based on actual demand. This helps optimize performance and cost without the need for manual intervention.
- **Estimate Data Size**: Accurately estimating item sizes helps calculate the required RCUs and WCUs, ensuring you're not under- or over-provisioning resources.
- **Monitor and Optimize**: Use CloudWatch to monitor DynamoDB metrics like read and write throughput, throttled requests, and consumed capacity to optimize RCUs and WCUs as needed.