**FIGSHARE DATASET AND REPRODUCING THE RESULTS**

Detailed explanation of the dataset used for simulation (from Figshare) and the necessary guidelines for reproducing the results are mentioned below:

**Dataset Overview:**

The simulation uses a dataset obtained from **Figshare**, which contains key parameters for blockchain load-balancing experiments, such as transaction arrival rates, service times, and shard utilization metrics. The dataset provides a realistic representation of blockchain network traffic under varying load conditions. It includes the following attributes:

1. **Transaction Arrival Rates (λ\lambdaλ)**: Simulated using a Poisson distribution to mimic real-world transaction traffic patterns.
2. **Service Times (μ\muμ)**: Represented using an Exponential distribution to model transaction processing times at each node.
3. **Shard Utilization Data**: Captures node idle times, waiting times, and utilization percentages for multiple shards under different traffic scenarios.

**Pre-processing the Dataset**

1. **Data Normalization**: Normalize transaction arrival rates to avoid extreme values that could skew the simulation results. This step ensures consistency and better comparison across all experiments.
2. **Handling Missing Data**: If the dataset contains missing values, use linear interpolation to fill gaps, ensuring continuous transaction arrival rates and service times.
3. **Data Partitioning**: Split the dataset into segments representing different traffic scenarios:
   * **Low Traffic**: Represents off-peak periods.
   * **High Traffic**: Represents peak load periods.

**Simulation Environment Setup:**

To reproduce the results, follow these steps:

1. **Tools and Software**:
   * Use **MATLAB R2021a** for implementing the M/M/k model and calculating performance metrics like waiting time (TwT\_wTw​), idle time (III), and utilization (UUU).
   * Employ **Simio-15** for visualizing load balancing across multiple shards.
2. **System Configuration**:
   * **Hardware**: A system with Intel Core i7 processor (3.6 GHz), 32 GB RAM, running Windows 10 OS.
   * **Shards and Nodes**: Initialize with ms=10m\_s = 10ms​=10 shards and nn=100n\_n = 100nn​=100 nodes distributed dynamically.
3. **Input Parameters**:
   * Arrival Rate (λ\lambdaλ): Varies between 1.4 to 2.2 requests per second.
   * Service Time (μ\muμ): Set at 0.74 queries per second.
   * Utilization Threshold (UmaxU\_{\text{max}}Umax​): Set at 0.9 to prevent overloading any shard.

**Simulation Process:**

1. **Step 1**: Load the dataset into MATLAB or Simio.
2. **Step 2**: Model the transaction arrival rates (λ\lambdaλ) using the Poisson distribution and service times (μ\muμ) using the Exponential distribution.
3. **Step 3**: Implement the M/M/k model to dynamically allocate nodes across shards based on the shard utilization metrics in the dataset.
4. **Step 4**: Run simulations for different traffic scenarios (low and high loads) to measure performance metrics:
   * Waiting Time (TwT\_wTw​).
   * Probability of Waiting (PwP\_wPw​).
   * Idle Time (III).
5. **Step 5**: Use Simio to visualize load distribution across shards and analyze the impact of node reallocation on performance.

**Performance Metrics Calculation:**

1. **Utilization (UUU)**:

U=λk⋅μU = \frac{\lambda}{k \cdot \mu}U=k⋅μλ​

Where kkk is the number of nodes in a shard.

1. **Probability of Waiting (PwP\_wPw​)**: Derived using queuing theory formulas based on shard load.
2. **Waiting Time (TwT\_wTw​)**:

Tw=Pw⋅service time1−UT\_w = \frac{P\_w \cdot \text{service time}}{1 - U}Tw​=1−UPw​⋅service time​

1. **Idle Time (III)**:

I=1−UI = 1 - UI=1−U

**Validation**

* **Compare Results**: Validate the simulation results by comparing them with those published in the paper. Focus on metrics like shard idle times, waiting times, and utilization rates.
* **Repeatability**: Ensure that the dataset and simulation configuration can be shared (e.g., via Figshare or GitHub) for other researchers to reproduce the results.

By following these steps and using the provided dataset, researchers can accurately replicate the simulation results and validate the performance of the proposed load-balancing approach. Let me know if you'd like further assistance with refining this explanation or if additional details are required.Top of Form

Bottom of Form