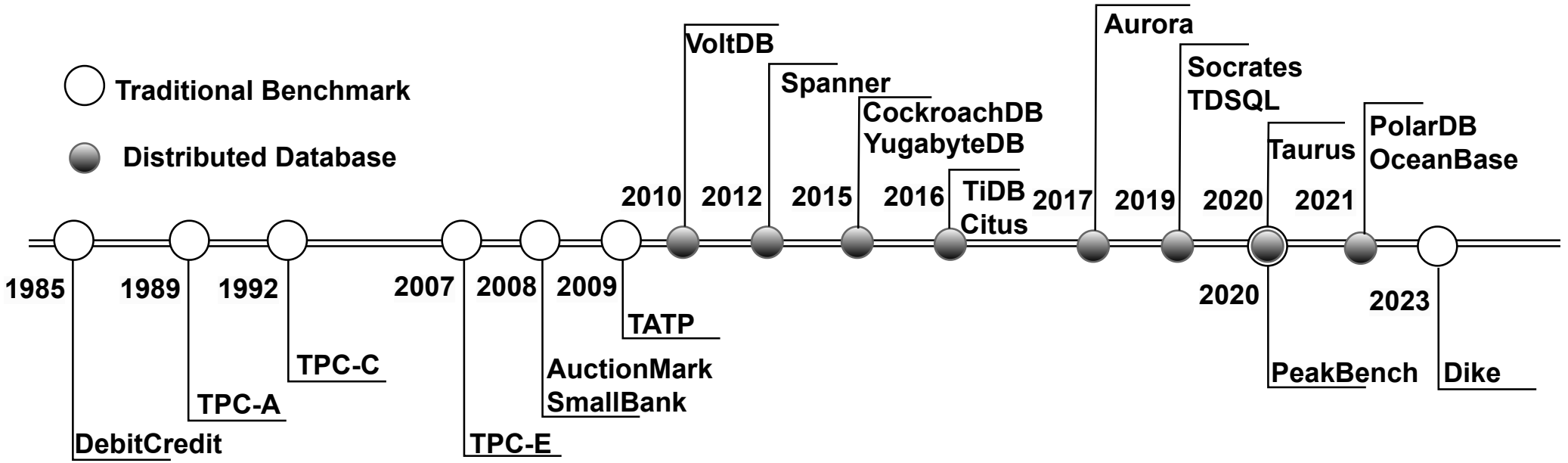
Benchmarking Distributed Transactional Database Systems

Speaker: Lingyang Zeng

Bench 2024

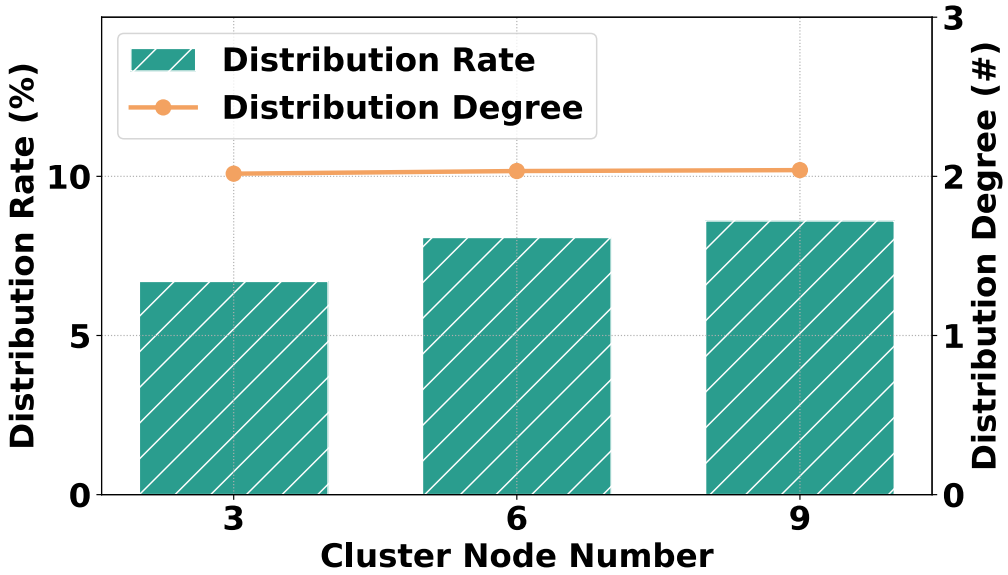
Limitations of Traditional Benchmark



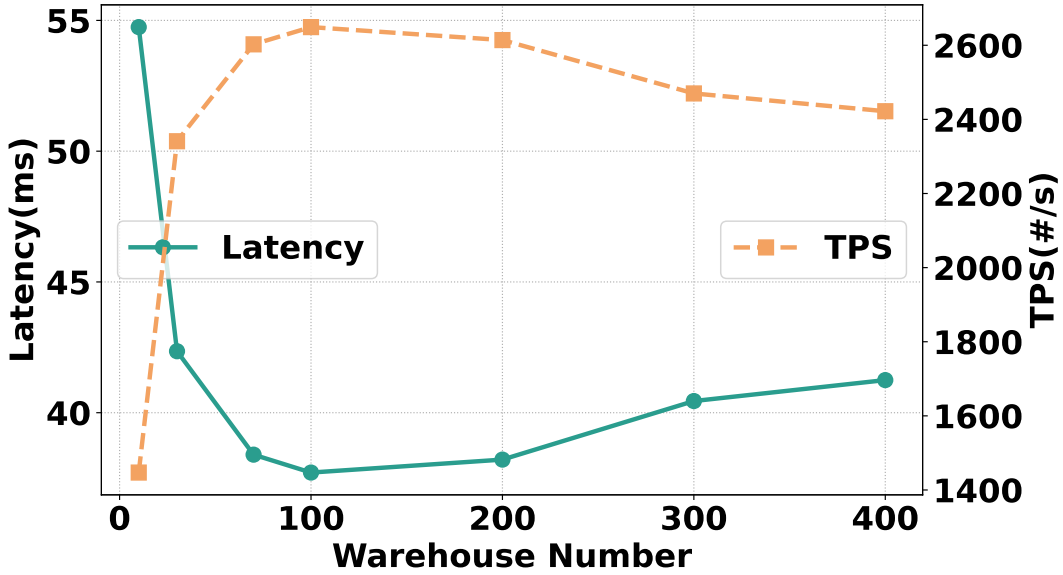
- **Benchmark development has fallen behind database progress**
  - Lack of focus on distributed scenarios
  - Lack of evaluation for database techniques
  - Lack of quantitative control

Take TPC-C as an Example …

- Not distributed-oriented
- Lack of quantitative control methods

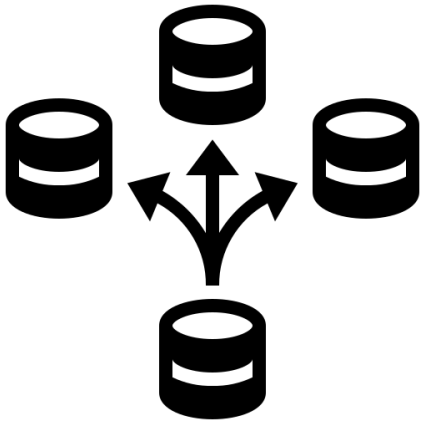


Distribution transaction generation



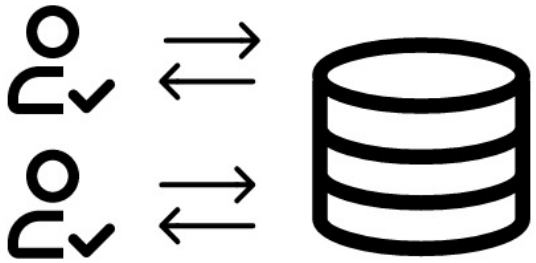
Impact of data scale

Distributed Transactional Database System



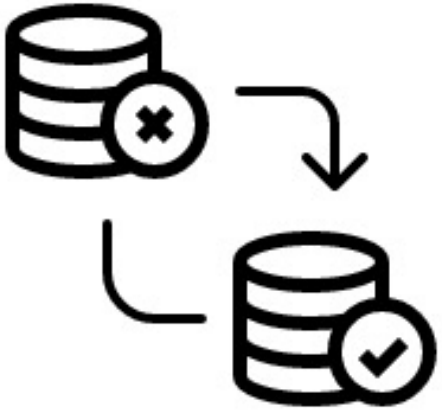
Scalability

- Distributed transaction
- Dynamic data scheduling



Consistency

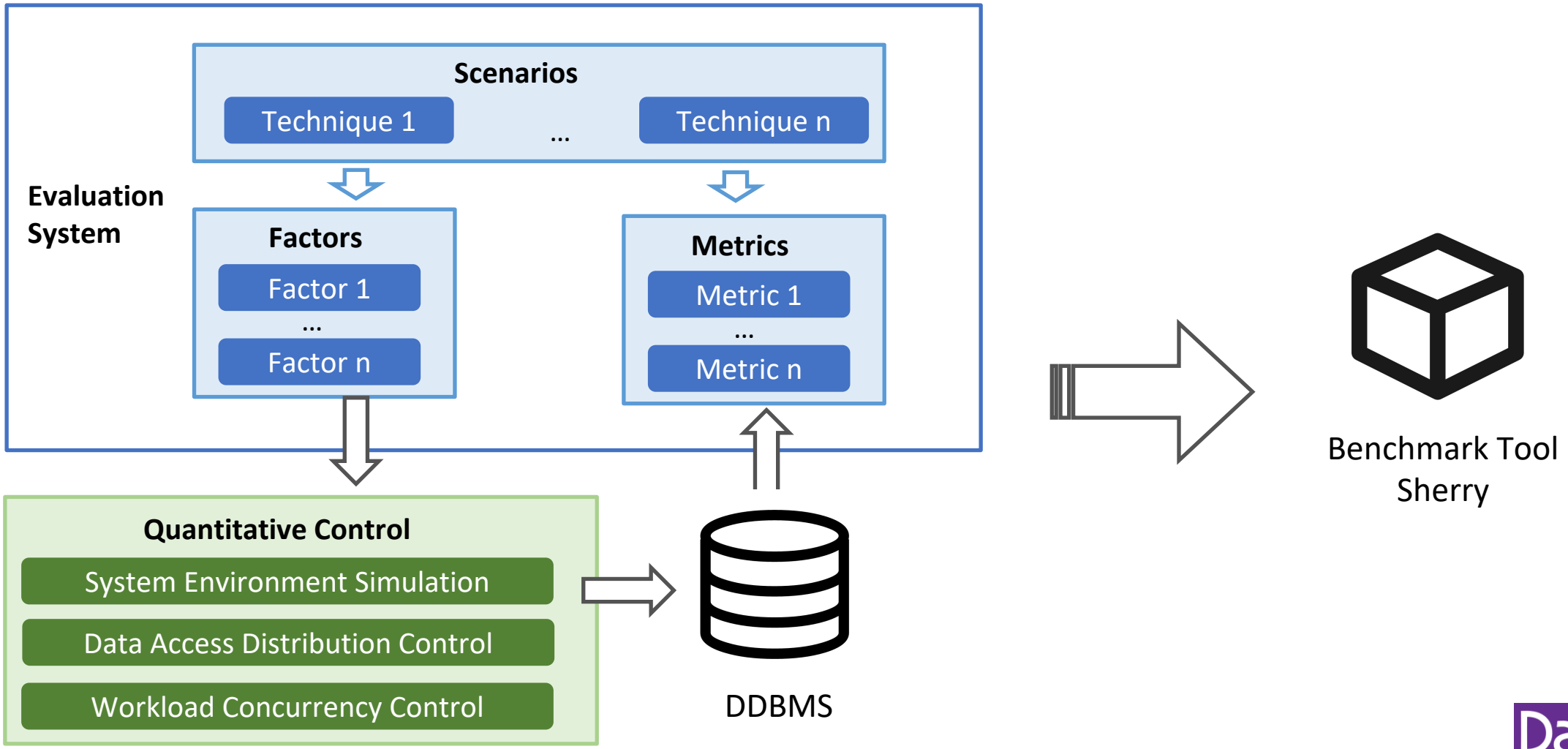
- Distributed lock management
- Distributed clock management



Availability

- Fault recovery

Contributions



Design of Evaluation System

Scenarios and Factors

- **Scalability-1. Distributed Transaction**
  - Challenge: Remote data access latency, atomic commit latency
  - Techniques & factors
    - Increase single-node transactions → transaction participant number
    - Reduce commit latency → operation number, operation read/write ratio
- Scalability-2. Dynamic Data Scheduling
- Consistency-1. Distributed Lock Management
- Consistency-2. Distributed Clock Management
- Availability. Fault Recovery

Design of Evaluation System

Scenarios and Factors

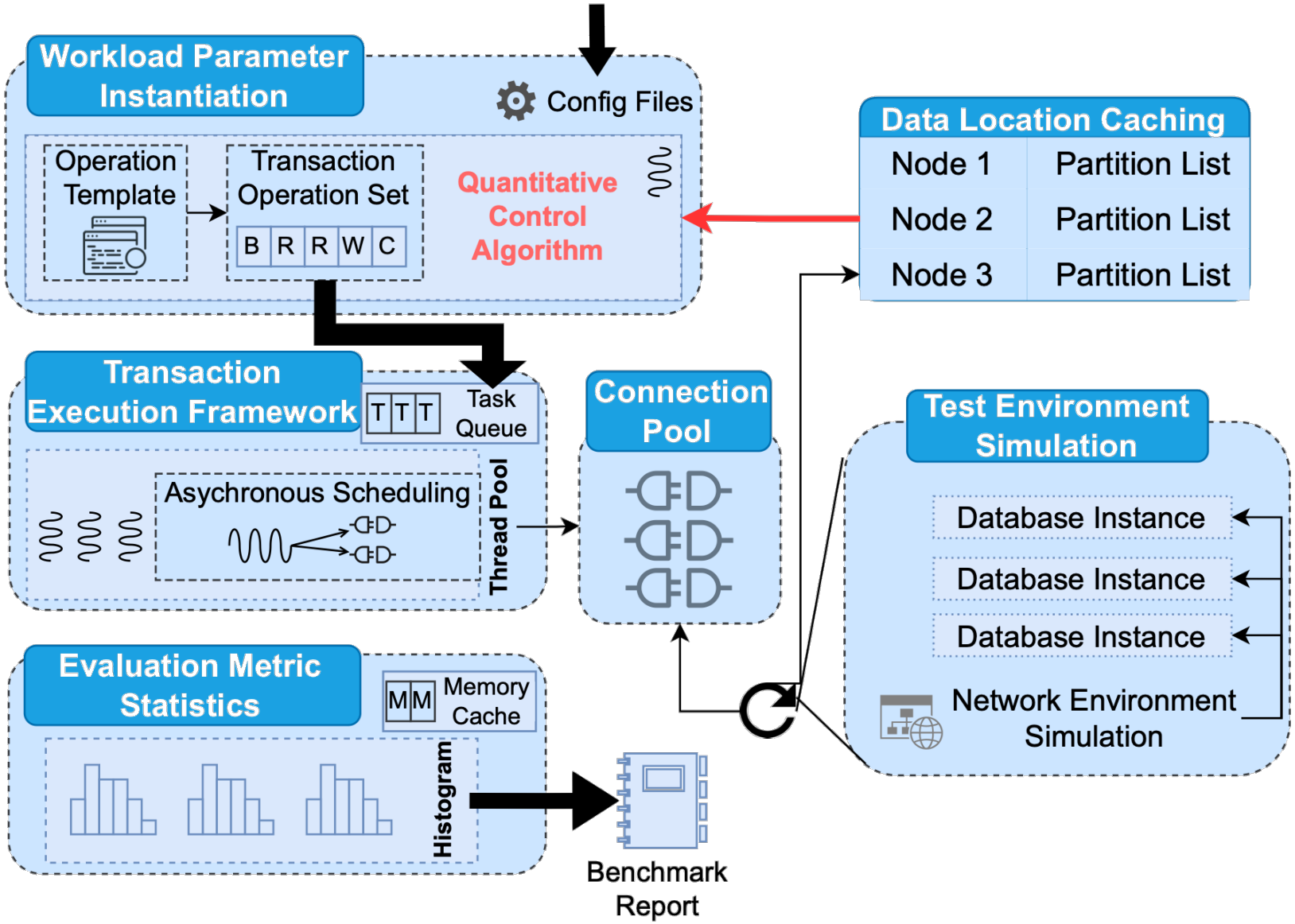
- Scalability-1. Distributed Transaction
- Scalability-2. Dynamic Data Scheduling
- **Consistency-1. Distributed Lock Management**
  - Challenge: Lock contention, Distributed deadlock
  - Techniques & factors
    - Reducing lock holding time → #conflicting transaction, #transaction operation
    - Global deadlock detection → deadlock cycle length, #deadlock cycle
- Consistency-2. Distributed Clock Management
- Availability. Fault Recovery

Design of Evaluation System

Metrics

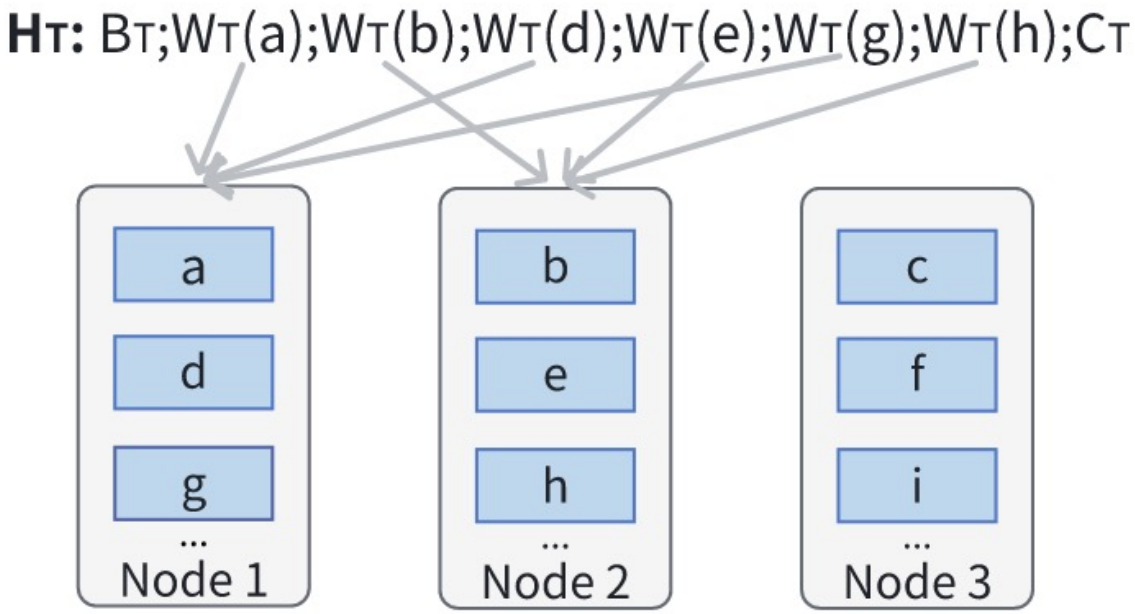
- **General Performance Metrics**
  - Throughput: average transactions per second
  - Latency: average response time
- **Metrics for Specific Scenarios**
  - Distributed Transaction
    - Read/commit performance: response time of read/commit operations
  - Dynamic data scheduling
  - Distributed lock management
    - Deadlock detection effectiveness: deadlock detection time/success rate
  - Distributed clock management
  - Fault Recovery

Framework of Sherry



Quantitative Control of Workload

Data Access Distribution Control

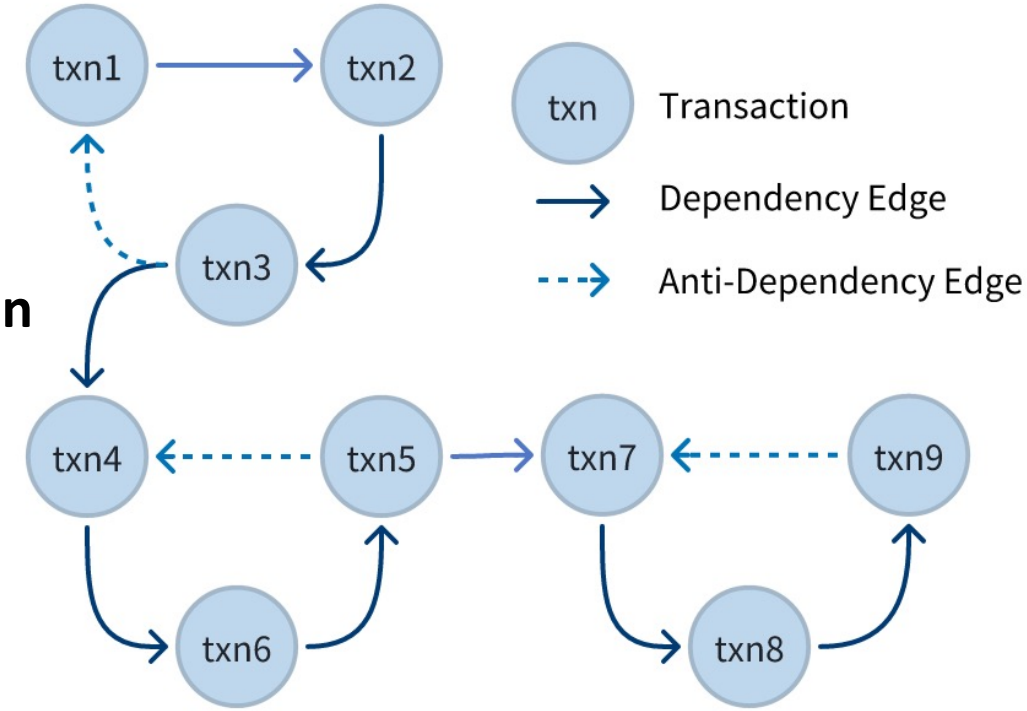


- **Data Placement Caching Strategy**
  - Monitor real-time throughput and scheduling status
  - **Periodically update** cache from the database
  - Ensure cache accuracy within an error threshold

Quantitative Control of Workload

Workload Concurrency Control

- **Transaction Dependency Graph Generation**
  - Put a back arrow **periodically** to setup deadlock length and deadlock number
- **Single-threaded Asynchronous Transaction Execution Framework**
  - **Non-blocking** transaction scheduling
  - I/O multiplexing for multi-transaction **execution on a single thread**



Effectiveness Evaluation

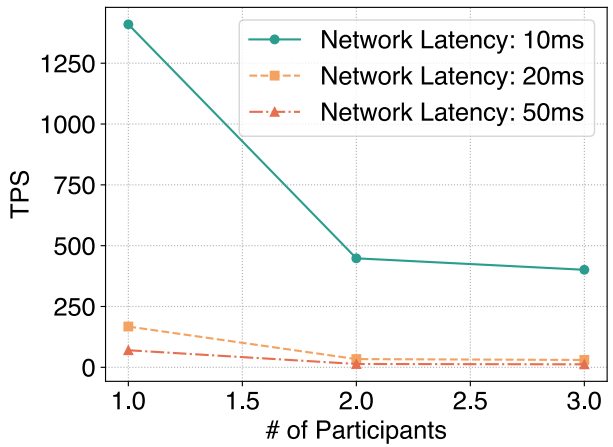
Environment and Settings

- **Database cluster environment**
  - 3-node OceanBase cluster (v4.1)
  - 8-core Intel(R) Xeon(R) Gold 6240R CPU @ 2.40GHz
  - 32GB physical memory, 64GB SSD storage
- **Benchmark configuration**
  - Isolation levels:
    - Conflict scenarios: read-commit
    - Non-conflict scenarios: snapshot isolation
  - Data volume: 90,000,000 rows
  - Execution settings: average of 5 runs, each lasting 5 minutes

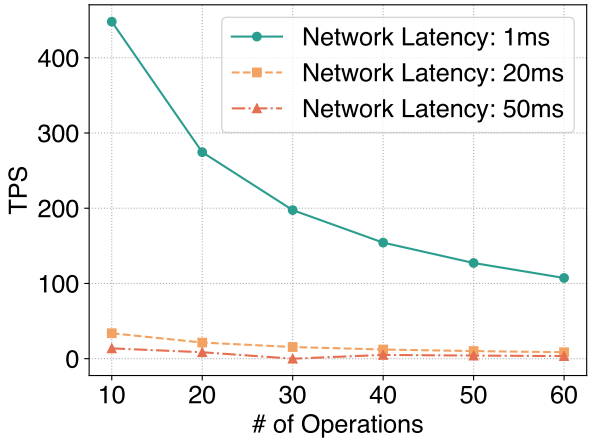
Selected Result Analysis

Distributed Transaction

- **Impact of participant number**
  - Single-node latency is much lower than distributed transactions
  - Distributed commit latency remains stable
- **Impact of operation number**
  - Read/write latency increases with more operations
  - Commit operation time stays stable



Impact of participant number

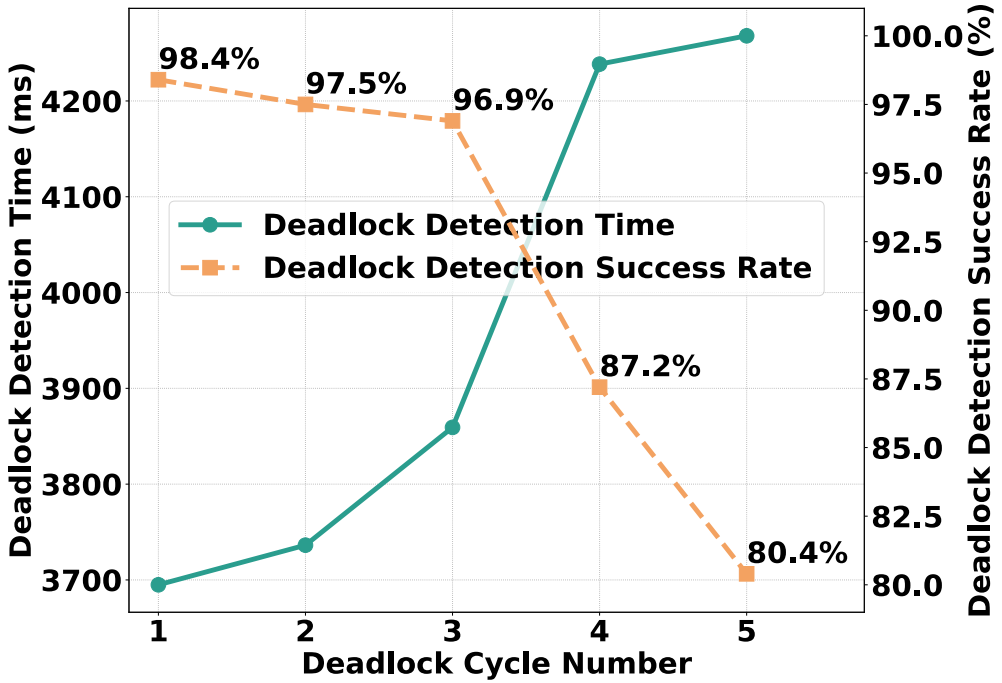


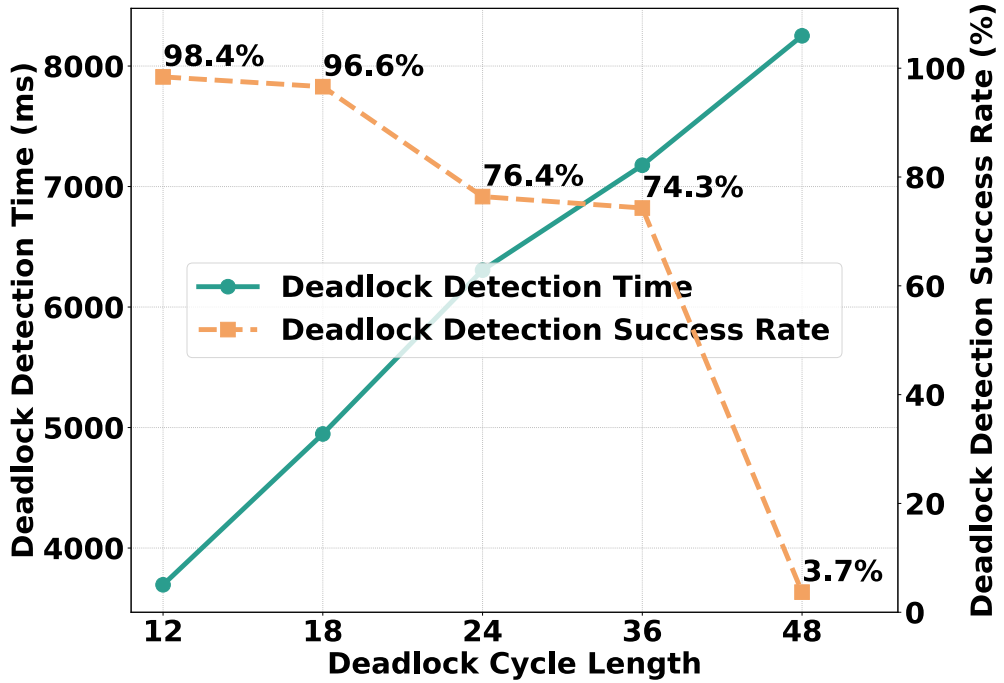
Impact of operation number

Selected Result Analysis

Distributed Lock Management

- **Impact of deadlock cycle length/number**
  - Detection time grows linearly with cycle length
  - Longer cycles reduce detection success
  - Cycle length impacts more than cycle number





Summary

- Propose an evaluation framework for scalability, consistency, and availability with five scenarios and metrics
- Design a workload generation algorithm for distribution and concurrency control
- Develop the Sherry benchmark tool
- Validate Sherry with various experiences

Thanks for Listening!
Q&A

DaSE
Data Science
& Engineering