TEMPORAL TRIANGLES:
A STRATEGIC BENCHMARK FOR PROPERTY GRAPH DATABASE SELECTION

EXECUTIVE SUMMARY

As organizations increasingly rely on connected data to drive business decisions, selecting the right graph database technology becomes critical to success. This whitepaper introduces the Temporal Triangles (TT) benchmark, a powerful tool for evaluating graph database performance across diverse business applications.

By testing how databases handle temporally ordered patterns in connected data, the TT benchmark provides actionable insights for technology leaders making strategic database selection decisions. Our approach enables fair comparison of different graph database platforms while measuring performance across use cases spanning financial services, cybersecurity, supply chain management, and customer relationship data.

INTRODUCTION

THE CONNECTED DATA CHALLENGE

The exponential growth of connected data across industries has driven the adoption of property graph databases as essential tools for storing, querying, and analyzing complex relationships. As organizations increasingly rely on these systems for mission-critical applications, selecting the right platform becomes a high-stakes decision with significant business implications.

While several benchmarks exist for traditional relational databases, the unique characteristics of graph databases—particularly those handling time-based data—remain inadequately addressed by current evaluation methods.

WHY TRADITIONAL BENCHMARKS FALL SHORT

Property graph databases differ fundamentally from relational systems in their optimization for traversal operations and pattern matching over highly connected data structures. These specialized operations demand benchmarks that:

1. Test realistic workloads representative of actual business applications
2. Evaluate performance on pattern matching with temporal constraints
3. Provide meaningful comparisons across different vendor platforms
4. Scale predictably with increasing data volumes

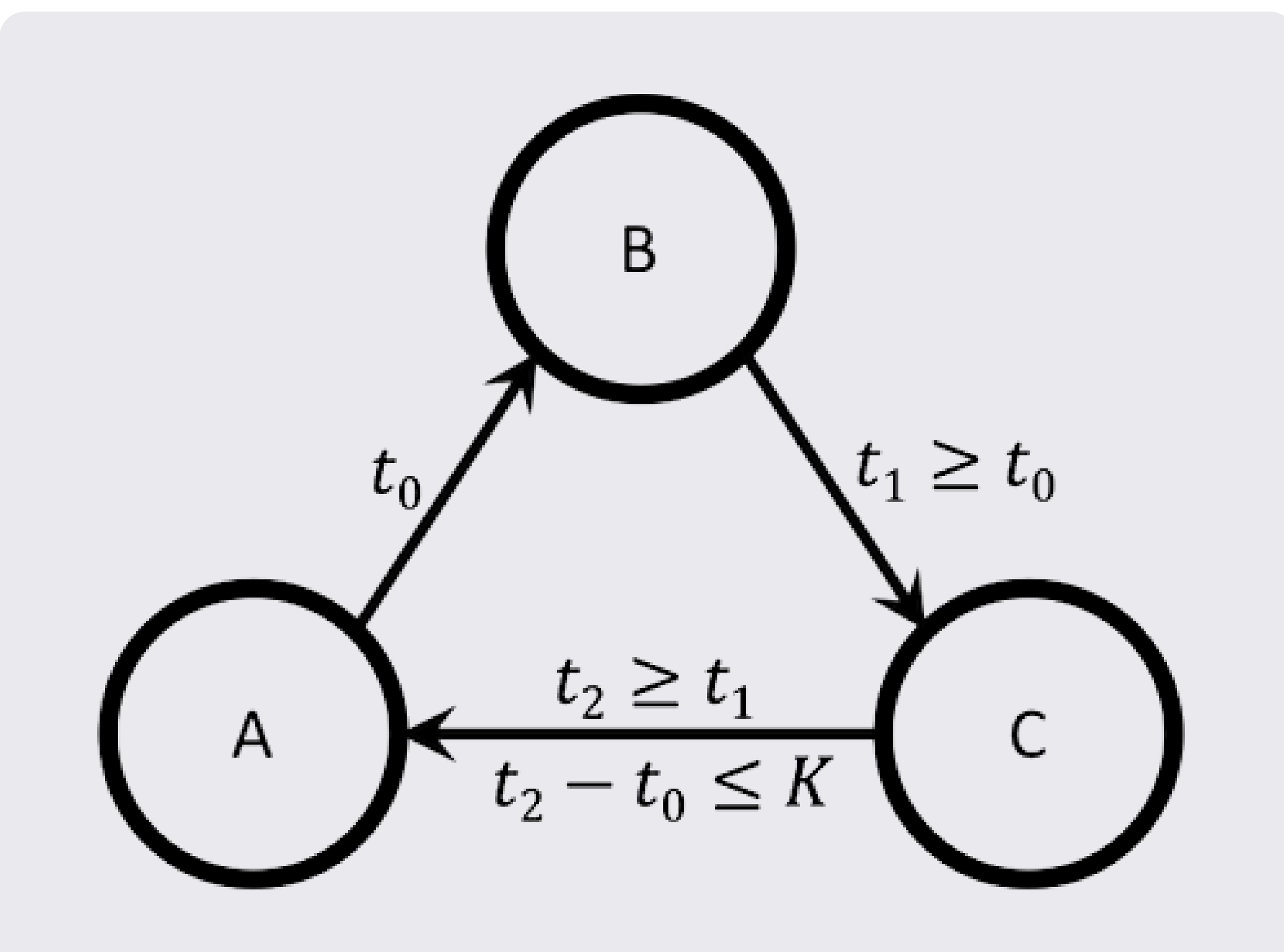
The Temporal Triangles benchmark addresses these needs by focusing on a pattern that appears naturally across multiple business domains.

THE TEMPORAL TRIANGLES BENCHMARK EXPLAINED

WHAT ARE TEMPORAL TRIANGLES?

The Temporal Triangles query identifies three-node cycles with sequentially ordered timestamps. In simple terms, it finds patterns where:

- Three distinct entities are connected in a cycle
- The connections between them have timestamps in ascending order
- The entire cycle completes within a specified time window



This pattern represents situations where actions or events flow from one entity to another and eventually return to the origin within a meaningful timeframe.

WHY THIS PATTERN MATTERS

What makes the Temporal Triangles pattern particularly valuable is that it transcends industry boundaries while maintaining consistent business relevance. Across diverse domains, the pattern represents:

1. **Causality chains:** Events that trigger subsequent related events
2. **Feedback loops:** Processes that eventually return to influence their origin
3. **Temporal dependency:** Activities that must occur in a specific sequence
4. **Event windows:** Related activities that occur within characteristic timeframes

This semantic consistency across domains means that a graph database optimized for TT queries can effectively serve diverse business use cases, making the benchmark a meaningful predictor of real-world performance.

BUSINESS APPLICATIONS

FINANCIAL SERVICES

In financial systems, Temporal Triangles often indicate patterns of interest:

- **Fraud detection:** Circular money flows are a classic indicator of potential fraud, identifying situations where funds move from Entity A to B to C and back to A within a suspiciously short timeframe
- **Money laundering:** Complex networks of transactions designed to obscure the source of funds often contain cyclical patterns
- **Market manipulation:** In trading networks, temporal triangles might indicate coordinated trading activities designed to manipulate prices

Business scenario: Entity A transfers funds to shell company B, which purchases assets through intermediary C, who then transfers equivalent value back to A, all within a 48-hour period—potentially indicating a wash trading scheme that compliance systems need to identify.

CYBERSECURITY

Security operations centers rely on identifying complex attack patterns:

- **Multi-stage attacks:** Many cyber attacks involve multiple stages that form temporal triangles—reconnaissance followed by exploitation followed by data exfiltration
- **Botnet communication:** Command servers contacting controllers, which instruct bots, which then report back to command servers
- **Data exfiltration:** Information moving through multiple systems before reaching its destination, often with verification steps completing a cycle

Business scenario: A potential data breach where server A communicates with server B, which then transfers data to external IP C, which subsequently sends a confirmation message back to server A—all within minutes. Detecting this pattern quickly can prevent significant losses.

SUPPLY CHAIN MANAGEMENT

Operational efficiency depends on understanding the flow of materials and information:

- **Production cycles:** Products moving through manufacturing and distribution systems often form cycles as they transform and transfer between entities
- **Transportation optimization:** Identifying temporal triangles in logistics networks can reveal inefficient routing or opportunities for consolidation
- **Quality feedback loops:** The flow of products through inspection, testing, and verification creates patterns that are valuable for analytics

Business scenario: Raw materials from supplier A are delivered to manufacturer B, who processes them into components for assembler C, who then sends quality feedback data back to supplier A—ideally completing within a 14-day production cycle.

CUSTOMER RELATIONSHIP MANAGEMENT

Understanding relationship formation patterns drives business growth:

- **Referral cycles:** Customer A refers prospect B, who becomes a customer and refers C, who then interacts with A
- **Influence mapping:** Identifying opinion leaders and tracking how their influence spreads through customer networks
- **Engagement cycles:** Tracking how marketing initiatives trigger customer actions that eventually lead to revenue events

Business scenario: At a conference, attendee A connects with attendee B on a professional network. B then connects with C, and finally, C discovers they share interests with A and connects back, completing the triangle. The timestamps of these connections fall within the 3-day conference window, indicating a potentially valuable business relationship cluster.

BENCHMARK IMPLEMENTATION

SYNTHETIC DATA GENERATION

The benchmark uses industry-standard RMAT graph generation to create synthetic datasets with realistic power-law degree distributions. This approach produces data with characteristics similar to real-world networks:

- Pronounced hubs and communities
- Small-world characteristics
- Realistic clustering patterns
- Scale-free properties

By controlling parameters like graph size, degree skew factor, timestamp range, and time window threshold, the TT benchmark provides a tunable, realistic, and computationally challenging test that measures both structural pattern matching capabilities and temporal filtering performance.

SCALE FACTORS

The benchmark defines standard scale factors to evaluate database performance across different data volumes:

Scale	**Vertices**	**Edges**	**Time Window**
XS	4,096	65,536	100
S	65,536	1,048,576	1,000
M	1,048,576	16,777,216	10,000
L	4,194,304	67,108,864	50,000
XL	16,777,216	268,435,456	100,000
XXL	67,108,864	1,073,741,824	200,000
XXXL	268,435,456	4,294,967,296	500,000

This graduated scale allows organizations to match benchmark conditions to their expected data volumes and growth projections. Having only 4 billion edges may not be large enough for some benchmarks. To generate larger datasets for reporting benchmark studies, simply using the SF as the size indicator is fine.

QUERY IMPLEMENTATION

The Temporal Triangles query can be implemented across various graph database platforms using their native query languages:

For Rocketgraph and Neo4j (Cypher):

```
MATCH (a)-[e1]->(b)-[e2]->(c)-[e3]->(a)
WHERE a <> b AND b <> c AND a <> c
  AND e1.timestamp <= e2.timestamp
  AND e2.timestamp <= e3.timestamp
  AND e3.timestamp - e1.timestamp < $threshold
RETURN a, b, c, e1.timestamp, e2.timestamp, e3.timestamp
```

For TigerGraph (GSQL):

```
CREATE QUERY TemporalTriangles(INT threshold) FOR GRAPH MyGraph {
  // Query implementation
  // (Detailed syntax available upon request)
}
```

For Apache TinkerPop (Gremlin):

```
g.V().as('a')
  .outE().as('e1').inV().as('b')
  .outE().as('e2').inV().as('c')
  .outE().as('e3').inV()
  .where(eq('a'))
  // Additional constraints
  // (Detailed syntax available upon request)
}
```

BENCHMARK METHODOLOGY

ENVIRONMENT STANDARDIZATION

To ensure fair comparison, all benchmark runs should be conducted on equivalent hardware configurations with standard specifications:

- **Hardware:** 16+ cores, 128GB RAM, NVMe SSD storage
- **Software:** Standardized OS, libraries, and runtime environments
- **Database configurations:** Documented settings for memory allocation, threading, indexing, and caching

PERFORMANCE METRICS

The benchmark measures several key performance indicators:

Primary Metrics

1. **Query latency:** Time to complete the query (milliseconds or seconds)
    - Mean, median, 95th percentile, standard deviation
    - Lower is better
2. **Throughput:** Number of temporal triangles processed per second
    - Calculated as (result count / execution time)
    - Higher is better
3. **Resource efficiency:** Ratio of results to resources consumed
    - Results per GB of RAM used
    - Results per CPU core-second
    - Higher is better

Secondary Metrics

1. **Scalability factors:**
    - Vertical scaling: Performance change when adding CPU/memory
    - Horizontal scaling: Performance change when adding nodes (distributed systems)
    - Data scaling: Performance change when increasing data volume
2. **Index efficiency:**
    - Index size relative to data size
    - Query performance improvement from indices
    - Index maintenance overhead
3. **Cost efficiency:**
    - Performance-per-dollar metrics
    - TCO considerations for on-premises vs. cloud deployments

BENCHMARKING PROTOCOL

A systematic execution process ensures consistency and comparability:

1. **Data Loading Phase:**
   - Measure loading time and throughput
   - Track index creation time
2. **Warm-up Phase:**
   - Execute standard queries to warm up the system
   - Run the benchmark query with varying parameters
3. **Execution Phase:**
   - Execute the TT query with standard parameters
   - Test different time window thresholds
   - Execute benchmark at each defined scale factor
4. **Measurement and Validation:**
   - Record execution time, result count, memory usage, CPU utilization
   - Verify result correctness and consistency

TARGET DATABASE SYSTEMS

The TT benchmark is designed to evaluate a wide range of property graph database systems, including but not limited to:

- Neo4j
- Rocketgraph
- TigerGraph
- Amazon Neptune
- JanusGraph
- ArangoDB
- Microsoft Azure Cosmos DB Graph API
- Oracle Graph
- OrientDB
- Memgraph

Each system offers unique features and architectural approaches, making comparative analysis valuable for understanding their relative strengths and weaknesses for temporal pattern matching workloads.

COMMON CHALLENGES AND MITIGATIONS

Effective benchmarking must address several common challenges:

1. **Caching effects:** Database caches dramatically affect performance
   Mitigation: Clear caches between runs or measure both cold and hot cache performance
2. **Query optimization variability:** Query planners may produce different plans across runs
   Mitigation: Force consistent query plans or record and report plan differences
3. **Result materialization:** Systems differ in how they materialize and return results
   Mitigation: Measure both query execution time and full result retrieval time
4. **Resource contention:** Other processes may affect benchmark performance
   Mitigation: Use dedicated, isolated environments for benchmark execution

PRACTICAL IMPLICATIONS FOR DECISION MAKERS

These findings have several practical implications for organizations selecting and optimizing graph database systems:

SYSTEM SELECTION GUIDANCE

1. **Performance baseline:** The TT benchmark provides a meaningful proxy for real-world performance across diverse use cases. Systems that perform well on this benchmark are likely to handle a variety of graph analytics workloads effectively.
2. **Cost-performance tradeoffs:** The resource utilization metrics enable more accurate cost-benefit analysis when selecting graph database systems, particularly for cloud-based deployments where resource consumption directly impacts operational costs.
3. **Scalability planning:** The benchmark's scaling dimensions help inform capacity planning and infrastructure requirements as data volumes grow.

OPTIMIZATION STRATEGIES

Based on benchmark findings, several optimization strategies can improve temporal query performance:

1. **Temporal indexing:** Creating effective indices on timestamp properties
2. **Query formulation:** Structuring queries to apply temporal constraints efficiently
3. **Memory management:** Optimizing cache usage for pattern matching operations
4. **Parallelization:** Leveraging multi-core and distributed processing capabilities

DEPLOYMENT CONSIDERATIONS

1. **On-premises vs. cloud:** Benchmark results can guide deployment model decisions
2. **Resource allocation:** Optimal CPU/memory ratios for different workloads
3. **Scaling approach:** Vertical vs. horizontal scaling tradeoffs

CONCLUSION

SUMMARY OF BENEFITS

The Temporal Triangles benchmark offers several significant advantages over existing graph database benchmarks:

1. **Cross-domain relevance:** Unlike domain-specific benchmarks, the TT query pattern appears naturally across diverse business functions, making it an ideal proxy for real-world performance.
2. **Balanced complexity:** The TT query combines structural pattern matching with temporal constraints, creating a benchmark that tests multiple database capabilities simultaneously.
3. **Tunable difficulty:** By adjusting parameters such as graph size and time window threshold, the benchmark can target different performance aspects and scale from small test environments to enterprise deployments.
4. **Standardized methodology:** The benchmarking methodology provides a rigorous framework for fair comparison, addressing common pitfalls in graph database evaluation.

NEXT STEPS

We invite technology leaders to adopt the Temporal Triangles benchmark as part of their database evaluation process:

Evaluate existing systems: Apply the benchmark to assess current database performance

Compare alternatives: Use standardized metrics to evaluate different vendor offerings

Plan for growth: Use scaling dimensions to project future requirements

Share experiences: Contribute to the community knowledge base around performance optimization

By establishing a common performance evaluation framework, organizations can make more informed decisions about graph database technology investments and ensure they select platforms capable of meeting both current and future business needs.

ABOUT ROCKETGRAPH

Rocketgraph enables enterprises and government agencies to discover the hardest-to-find insights without hiring a command center full of rocket scientists. Born out of a high-performance computing project at the Department of Defense, our graph analytics platform allows an analyst to use GenAI to do iterative analysis with the largest, most complicated datasets on the planet and get answers hundreds of times faster than traditional graph tools. Rocketgraph builds property graphs that scale to hundreds of billions of edges. Our government and enterprise customers build fine-tuned forecasts, detect sophisticated fraud schemes, monitor nefarious activity on the dark web, keep their networks secure, and answer their most challenging questions with Rocketgraph graph analytics.