УДК 004.052:004.75

SEMANTIC OBSERVABILITY IN STREAM PROCESSING: RUNTIME CORRECTNESS VERIFICATION FOR ONLINE MARKET MICROSTRUCTURE FEATURE ENGINEERING

Egor Kolychev[1] (master student)
Scientific supervisor – PhD, Associate Professor Ivan Perl[1]
[1]ITMO University
507367@niuitmo.ru

Abstract
Stream processing systems are widely used to calculate market microstructure metrics in real time, but most observability tools monitor only infrastructure indicators and therefore cannot reveal semantic correctness errors in the computed values. This work proposes a three-level observability model for such computations. The model separates infrastructure monitoring, computation-state monitoring, and semantic monitoring based on invariants derived from formal metric definitions. Using Order Flow Imbalance as the main example, the study shows that the mathematical specification of a metric can be transformed into runtime checks for spread validity, quote-update consistency, and agreement between matched trade volume and buy/sell volume. A taxonomy of streaming operators for financial metrics is also introduced, distinguishing windowed aggregates, order-dependent state machines, temporal joins, online estimators, and path-dependent volume-time operators. The proposed approach is evaluated on an OFI streaming pipeline with injected duplicate, reordered, lost, and frozen events. The experiments demonstrate that standard infrastructure indicators may remain normal while semantic invariants reveal data quality failures with constant per-event overhead. The model can support real-time financial analytics systems that require correctness guarantees beyond latency and throughput monitoring.
Keywords
Stream processing, observability, semantic correctness, runtime verification, market microstructure, Order Flow Imbalance.

СЕМАНТИЧЕСКАЯ НАБЛЮДАЕМОСТЬ В ПОТОКОВОЙ ОБРАБОТКЕ: ПРОВЕРКА КОРРЕКТНОСТИ ВО ВРЕМЯ ВЫПОЛНЕНИЯ ДЛЯ ПОТОКОВОГО ФОРМИРОВАНИЯ ПРИЗНАКОВ РЫНОЧНОЙ МИКРОСТРУКТУРЫ

Колычев Е. И.[1] (магистрант)
Научный руководитель – кандидат технических наук, доцент Перл И. А.[1]
[1]Университет ИТМО
507367@niuitmo.ru

Аннотация
Системы потоковой обработки широко используются для расчета метрик рыночной микроструктуры в реальном времени, однако большинство средств наблюдаемости контролирует только инфраструктурные показатели и поэтому не выявляет семантические ошибки в вычисленных значениях. В работе предложена трехуровневая модель наблюдаемости для таких вычислений. Модель разделяет инфраструктурный мониторинг, мониторинг состояния вычислений и семантический мониторинг, основанный на инвариантах, выводимых из формальных определений метрик. На примере дисбаланса потока заявок показано, что математическая спецификация метрики может быть преобразована в проверки времени выполнения: корректность спреда, согласованность обновлений котировок и соответствие объема исполненных сделок суммарным объемам

покупок и продаж. Также предложена таксономия потоковых операторов для финансовых метрик, включающая оконные агрегаты, порядково-зависимые автоматы состояний, временные соединения, онлайн-оцениватели и путь-зависимые операторы с окнами по объему. Подход проверен на потоковом конвейере расчета OFI при внесении дубликатов, перестановок, потерь и заморозки событий. Эксперименты показывают, что стандартные инфраструктурные индикаторы могут оставаться в норме, тогда как семантические инварианты выявляют нарушения качества данных с постоянными накладными расходами на событие. Модель применима при проектировании систем финансовой аналитики, которым требуются гарантии корректности сверх мониторинга задержки и пропускной способности.
Ключевые слова
Потоковая обработка, наблюдаемость, семантическая корректность, проверка времени выполнения, рыночная микроструктура, дисбаланс потока заявок.

Stream processing systems are increasingly used to compute market microstructure metrics in real time: Order Flow Imbalance (OFI), Volume-Synchronized Probability of Informed Trading (VPIN), Trade Flow Imbalance, and others [1]. These metrics require maintaining state between events, strict processing order, and window semantics, making them sensitive to failures typical of distributed systems: event duplication, reordering, and data loss. Stanford, Kallas, and Alur demonstrated that advances in stream processing performance have not been matched by adequate attention to computation correctness: an error in a streaming application immediately produces wrong results for downstream consumers [2]. Existing observability tools – OpenTelemetry, Prometheus, consumer lag monitoring – track infrastructure metrics (latency, throughput, error rates) but cannot detect a situation where all system indicators are normal while computed values are incorrect. Batch validation tools (Great Expectations, dbt tests) operate post-hoc on materialized data. Academic Stream Runtime Verification systems (RTLola, MonPoly) provide formal guarantees over event streams but have not been applied to domain-specific financial streaming computations [3]. Thus, a gap exists between three fields – stream processing, financial mathematics, and runtime verification – each solving an adjacent but non-overlapping problem.

This work proposes a three-level observability model for streaming computations. Level 1 (infrastructure) covers standard monitoring: latency, throughput, consumer lag, implemented via OpenTelemetry and Prometheus. Level 2 (computation) covers state store size, checkpoint duration, and watermark progress – partially available in Apache Flink and Kafka Streams as built-in metrics. Level 3 (semantic) introduces domain-specific correctness invariants derived from the mathematical definition of the computed metric – absent from existing systems and constituting the main contribution of this work. The key insight is that a formal metric specification defines not only the computation algorithm but also a set of runtime-checkable invariants. For OFI [1], defined as the sum of contributions from consecutive changes in best bid/ask prices in the order book, the definition yields invariants: (a) in the absence of quote update events, the OFI increment equals zero; (b) the spread (ask minus bid) is non-negative at all times; (c) the total buy and sell volume matches the total matched trade volume. Violation of any invariant indicates data corruption: event duplication, reordering, or message loss. The PGVal study confirms that even exactly-once semantics does not guarantee semantic correctness: Flink silently drops late events in join operations, and Kafka Streams discards all late events in multi-stream topologies [4].

To systematize semantic monitoring requirements, a taxonomy of microstructure metrics by streaming operator type is proposed: (1) windowed commutative aggregates, order-insensitive within a window (Trade Flow Imbalance, Realized Volatility); (2) order-dependent state machines where each event's contribution depends on the previous state (OFI); (3) join operators with temporal shift requiring delayed reference lookups (Spread Decomposition); (4) online estimators based on rolling regression (Kyle's Lambda); (5) path-dependent operators with volume-time windowing where bucket boundaries depend on cumulative volume (VPIN). Each class exhibits a specific set of semantic invariants and different sensitivity to ordering violations and data loss. The taxonomy enables determining the minimum set of runtime checks for each computation type.

Experimental verification is conducted on an OFI streaming pipeline. Under controlled fault injection scenarios – duplicate quotes from a feed provider, event reordering due to network disruption, reprocessing after Kafka partition recovery, and frozen data feeds – infrastructure metrics remain within normal bounds while semantic invariants detect correctness violations. Invariant checking executes with constant per-event overhead, ensuring minimal additional cost for high-throughput systems.

The proposed semantic observability model detects correctness violations invisible to standard infrastructure monitoring. The approach is technology-agnostic: invariants are derived from the metric's mathematical definition and applicable to architectures of varying scale – from lightweight pipelines to distributed frameworks. The results can be applied in designing online financial market analytics systems requiring real-time correctness guarantees. Future work includes developing adaptive invariants calibrated to market regime from historical data, and formalizing automatic invariant derivation from arbitrary streaming metric specifications.

Literature

1. Cont R., Kukanov A., Stoikov S. The Price Impact of Order Book Events [Electronic resource]. – arXiv:1011.6402v3 [q-fin.TR], 2011. – 26 p. – URL: https://arxiv.org/pdf/1011.6402.
2. Stanford C., Kallas K., Alur R. Correctness in Stream Processing: Challenges and Opportunities // CIDR 2022: 12th Annual Conference on Innovative Data Systems Research. – Santa Cruz, CA, USA, 2022. – 2 p. – URL: https://www.cidrdb.org/cidr2022/papers/p103-stanford.pdf.
3. Bartocci E., Falcone Y. (eds.). Lectures on Runtime Verification: Introductory and Advanced Topics [Electronic resource]. – Cham: Springer, 2018. – Lecture Notes in Computer Science. – Vol. 10457. – VII, 233 p. – DOI: 10.1007/978-3-319-75632-5. – URL: https://inria.hal.science/hal-01762298/file/rv-book-editor-version.pdf.
4. Tahir J., Mayer R., Doblander C., Jacobsen H.-A. How Reliable Are Streams? End-to-End Processing-Guarantee Validation and Performance Benchmarking of Stream Processing Systems // Proceedings of the VLDB Endowment. – 2024. – Vol. 18, No. 3. – P. 585–598. – DOI: 10.14778/3712221.3712227.