

DOI: <https://doi.org/10.60797/COMP.2026.12.1> EDN: XPUMUK**COST-EFFICIENT STRATEGIES FOR MANAGING TRAFFIC SPIKE PATTERNS IN HIGH-LOAD SYSTEMS. ALTERNATIVES TO HORIZONTAL SCALING: A MULTI-FACTOR ANALYSIS OF COST-EFFICIENT APPROACHES TO HANDLING SPIKE LOADS IN EVENT-DRIVEN SYSTEMS**

Research article

Redko D.A.^{1,*}¹ Ada Support Incorporated, Toronto, Canada

* Corresponding author (dredko[at]outlook.com)

Suggested: 18.05.2026; Accepted: 08.10.2026; Published: 29.10.2026

Abstract

This paper examines cost-efficient strategies for managing traffic spike patterns in high-load systems with event-driven and stream-oriented architectures. It is demonstrated that relying on horizontal scaling cannot be considered a universal response, as it is constrained by both technical and economic limitations. A classification of spike patterns specific to event-driven systems is proposed, and the reasons for their critical impact are identified. Based on a literature review, alternative approaches to handling spike loads are systematized. An original model for assessing spike severity and an integral indicator for strategy selection are developed. It is concluded that, under spike load conditions, the highest effectiveness is achieved through a staged decision-making process, in which horizontal scaling is used as one of several available tools, while the choice of response is determined by spike characteristics and economic feasibility.

Keywords: traffic spike patterns, event-driven systems, stream-oriented architectures, horizontal scaling, load shedding, admission control, predictive auto-scaling, vertical scaling, SLO, cost-efficiency.

COST-EFFICIENT СТРАТЕГИИ УПРАВЛЕНИЯ SPIKE-ПАТТЕРНАМИ ТРАФИКА В ВЫСОКОНАГРУЖЕННЫХ СИСТЕМАХ. АЛЬТЕРНАТИВЫ ГОРИЗОНТАЛЬНОМУ МАСШТАБИРОВАНИЮ: МУЛЬТИФАКТОРНЫЙ АНАЛИЗ COST-EFFICIENT ПОДХОДОВ К ОБРАБОТКЕ SPIKE-НАГРУЗОК В EVENT-DRIVEN СИСТЕМАХ

Научная статья

Редько Д.А.^{1,*}¹ Эйда Саппорт, Торонто, Канада

* Корреспондирующий автор (dredko[at]outlook.com)

Предложена: 18.05.2026; Принята: 08.10.2026; Опубликовано: 29.10.2026

Аннотация

В статье рассматриваются cost-efficient стратегии управления spike-паттернами трафика в высоконагруженных системах с event-driven и stream-oriented архитектурой. Показано, что реакция через горизонтальное масштабирование не может считаться универсальной, поскольку она связана с техническими и стоимостными ограничениями. Предложена классификация spike-паттернов для event-driven систем и раскрыты причины их критичности. На основе анализа литературы систематизированы альтернативные подходы к обработке spike-нагрузок. Разработана авторская модель оценки тяжести всплеска и интегральный показатель выбора стратегии. Сделан вывод о том, что при spike-нагрузках наибольшую результативность обеспечивает поэтапное принятие решений, в котором горизонтальное масштабирование используется как один из допустимых инструментов, а выбор реакции определяется параметрами всплеска и экономической целесообразностью.

Ключевые слова: spike-паттерны трафика, event-driven системы, stream-oriented архитектуры, горизонтальное масштабирование, сброс нагрузки, контроль доступа, прогнозируемое автоматическое масштабирование, вертикальное масштабирование, SLO, экономическая эффективность.

Introduction

The widespread adoption of event-driven and stream-oriented architectures has reshaped the requirements for load management in high-load systems. In particular, such systems are characterized by the continuity of event streams, dependence on queue processing rates, and high sensitivity to short-term spikes in incoming traffic. At the same time, the cost of cloud resources continues to play an independent role in engineering decision-making, as an imbalance between performance and cost often renders a technically viable solution economically inefficient. In the survey by V. Cardellini, F. Lo Presti, M. Nardelli, and G. Russo, it is rightly noted that stream processing systems rely on a wide range of runtime solutions, including resource reallocation, scaling, load shedding, migration, and execution parameter reconfiguration. The breadth of these approaches indicates the absence of a single universal solution to traffic spikes, while the choice of response must take into account the characteristics of the operating environment and the load profile [1].

Moreover, the operational practice of cloud-based stream processing services reveals another important issue. Automatic scaling is often perceived as a convenient mechanism for lowering the barrier to entry; however, its cost may significantly deviate from a conditionally optimal scenario. As emphasized in the study by Y. Wang, B. Lyu, and V. Kalavri, cloud platforms

tend to scale up rapidly to meet performance requirements but reduce resources much more slowly after the peak has passed. This creates an asymmetry that results in paying for underutilized resources and gives rise to the so-called *non-expert tax*, i.e., the costs incurred when using auto-scaling without expert tuning [2]. As a result, the problem arises of determining which strategies for handling spike loads are truly cost-efficient for event-driven systems and under what conditions they are preferable to horizontal scaling.

The aim of this study is to systematize alternatives to horizontal scaling, develop a model for assessing spike severity, and formulate an algorithm for selecting a response strategy to spike loads.

Research methods and principles

The research materials consist of international scholarly publications focused on runtime management of stream processing systems, handling of bursty workloads, load shedding, admission control, predictive auto-scaling, vertical scaling, and the evaluation of the performance–cost trade-off in cloud environments. The study employs methods of analysis and synthesis of scientific literature, comparative analysis of approaches to handling spike loads, as well as structural and functional analysis of response mechanisms to traffic spikes in event-driven systems. Based on the reviewed sources, cost-efficient strategies are systematized, and an original framework for their evaluation is proposed. To support the proposed framework, a calculation-based validation was additionally performed. Public Azure Functions Trace 2019 data were used to justify the relevance of event-driven spike scenarios, while published PASS benchmark results were used to calculate and compare cost-efficiency indicators for common auto-scaling strategies.

Main results

A traffic spike pattern can be defined as a sharp deviation of incoming flow intensity from the baseline level, occurring over a short time interval and capable of disrupting the event processing regime. In event-driven systems, the criticality of such deviation is determined by the impact of increased input flow on queue length, processing time, consumer latency, event acknowledgment time, and compliance with latency SLOs. In the study by L. Lu, L. Cherkasova, V. de Nitto Personè, N. Mi, and E. Smirni on overload management under bursty workloads, it is shown that increased burstiness affects blocking queues, response time, and the probability of session interruption. The authors propose an admission control policy that accounts for the level of burstiness in regulating access; in other words, according to the authors, spike load should be evaluated based on its impact on service parameters rather than on the value of input intensity [3].

Based on the above, a classification of spike patterns is proposed (Fig. 1).

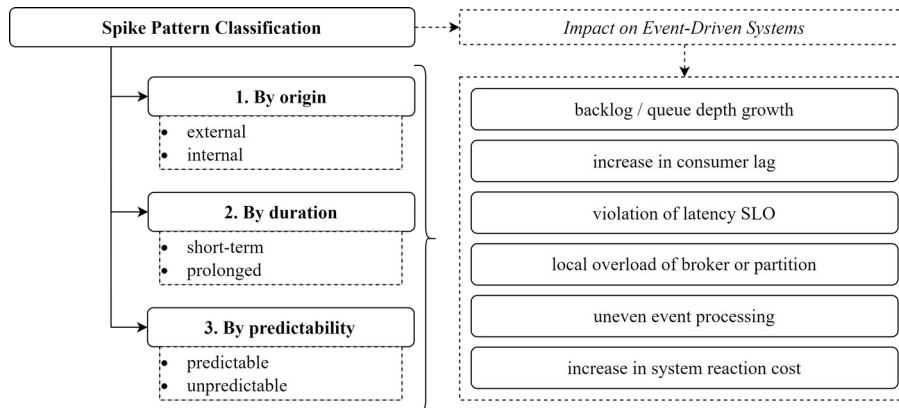


Figure 1 - Classification of spike patterns and the main reasons for their critical impact on event-driven systems

DOI: <https://doi.org/10.60797/COMP.2026.12.1.1>

As can be observed, the criticality of spike patterns for event-driven systems is relatively high. At the same time, for the purpose of a formalized assessment of the identified patterns, it is proposed to introduce a spike severity index (1):

$$SI = w_1 \text{peakbase} + w_2 T_{\text{burst}} T_{\text{ref}} + w_3 Q_{\text{growth}} Q_{\text{max}} + w_4 L_{\text{slo}} L_{\text{max}} \quad (1)$$

where:

- SI is the integral assessment of spike severity;
- peak is the peak input flow intensity;
- base is the baseline intensity;
- T_{burst} is the spike duration;
- T_{ref} is the reference duration;
- Q_{growth} is the queue growth;
- Q_{max} is the maximum allowable backlog;
- L_{slo} is the deviation of latency from the target value;
- L_{max} is the maximum allowable deviation;
- w₁ + w₂ + w₃ + w₄ = 1.

In the proposed interpretation, a moderate spike corresponds to a low SI value (up to 0.8), a significant spike to an intermediate range (0.8–1.2), and a critical spike to a high index value (above 1.2).

At the same time, it is important to emphasize why the exclusive use of horizontal scaling cannot be considered universal. While horizontal scaling remains relevant for high-load systems, its universalization leads to a critical limitation: under spike load conditions, additional infrastructure capacity is not provisioned instantaneously, as time is required for instance startup, warm-up, load redistribution, and, in some cases, state migration between computational nodes. Another issue is that the response to a spike may be excessive. After the spike subsides, the system continues to retain a portion of resources that no longer provide comparable utility. In the study by I. Fé, R. Matos, J. Dantas, C. Melo, et al., focused on the trade-off between performance and cost in auto-scaling, particular attention is given to the necessity of evaluating scaling through the joint consideration of performance and cost. In other words, resource expansion cannot be reduced to a purely technical action, as the economic cost of the response must be analyzed alongside the probability of maintaining the required service level [4].

For event-driven systems, this issue is further complicated by dependencies on queues, operator state, and inter-node interactions:

- 1) increasing the number of instances raises coordination overhead, network communication costs, and state synchronization requirements;
- 2) under short spikes, the deployment time of horizontal scaling may approach the duration of the spike itself;
- 3) under prolonged spikes, an excessive response leads to overshoot, after which the reduction in the number of instances occurs significantly more slowly than their increase.

For systems with strict SLO constraints, such asymmetry implies a higher cost of configuration errors, as misconfiguration of scaling thresholds, cooldown parameters, or the interpretation of incoming metrics may shift the solution from a resource-efficient mode to a state of resource overprovisioning without a meaningful improvement in service quality. Thus, the following typology of cost-efficient strategies can be proposed as alternatives to horizontal scaling (Table 1).

Table 1 - Typology of cost-efficient strategies as alternatives to horizontal scaling

DOI: <https://doi.org/10.60797/COMP.2026.12.1.2>

Approach	Mechanism	Application scenario	Limitations	Cost-efficiency
Horizontal scaling	Adding new instances	Sustained increase in load	Startup latency, overshoot, inter-node overhead	Moderate for short spikes, higher for sustained ones
Load shedding	Dropping part of the incoming flow	Critical backlog growth, time constraints	Partial data loss	High for short critical spikes
Admission control / throttling	Limiting incoming flow	Early-stage load increase, risk of overload	Depends on accurate threshold definition, restricts part of requests	High for moderate and significant spikes
Prediction-based / predictive scaling	Proactive resource adjustment	Predictable and recurring spikes	Depends on forecast accuracy	High for regular patterns
Vertical scaling	Increasing resources of existing instances	Short sudden spikes	Limited by node capacity	High for short-term spikes

Source: compiled by the author based on [5], [6], [7], [8]

Based on Table 1, it is proposed to use an integral cost-efficiency assessment for strategy selection (2):

$$CEs = w1 \cdot (1 - Cs) + w2SLOs + w3 \cdot (1 - Rs) + w4 \cdot (1 - Ts) + w5 \cdot (1 - Os) \quad (2)$$

where:

- CE_s is the integral assessment of strategy *s*;
- Cs is the normalized implementation cost of the strategy;
- SLOs is the normalized degree of SLO compliance;
- Rs is the normalized risk of data loss, degradation of processing quality, or partial failure;
- Ts is the normalized stabilization time after the spike;
- Os is the normalized operational complexity of implementation and maintenance.

The optimal strategy is the one for which the value of CE_s is maximized (3):

$$s^* = \operatorname{argmax} CE_s \quad (3)$$

The proposed model makes it possible to evaluate multiple strategies within a single scale. The set of compared options should include the following (Fig. 2): controlled load shedding, admission control, prediction-based provisioning, predictive auto-scaling, vertical scaling, and horizontal scaling.

Figure 2 - Comparative analysis of cost-efficiency strategies for managing spike patterns

DOI: <https://doi.org/10.60797/COMP.2026.12.1.3>

Thus, for short-term and sharply pronounced spikes, high effectiveness is often achieved by admission control or vertical scaling, as their effects manifest more rapidly; for predictable recurring spikes, predictive auto-scaling may have an advantage. Under critical queue growth and severe time constraints, controlled load shedding may be acceptable if the reduction in processing completeness is compensated by maintaining an acceptable level of service, whereas for sustained demand, horizontal scaling remains appropriate.

The decision-making algorithm for spike loads is presented in Fig. 3.

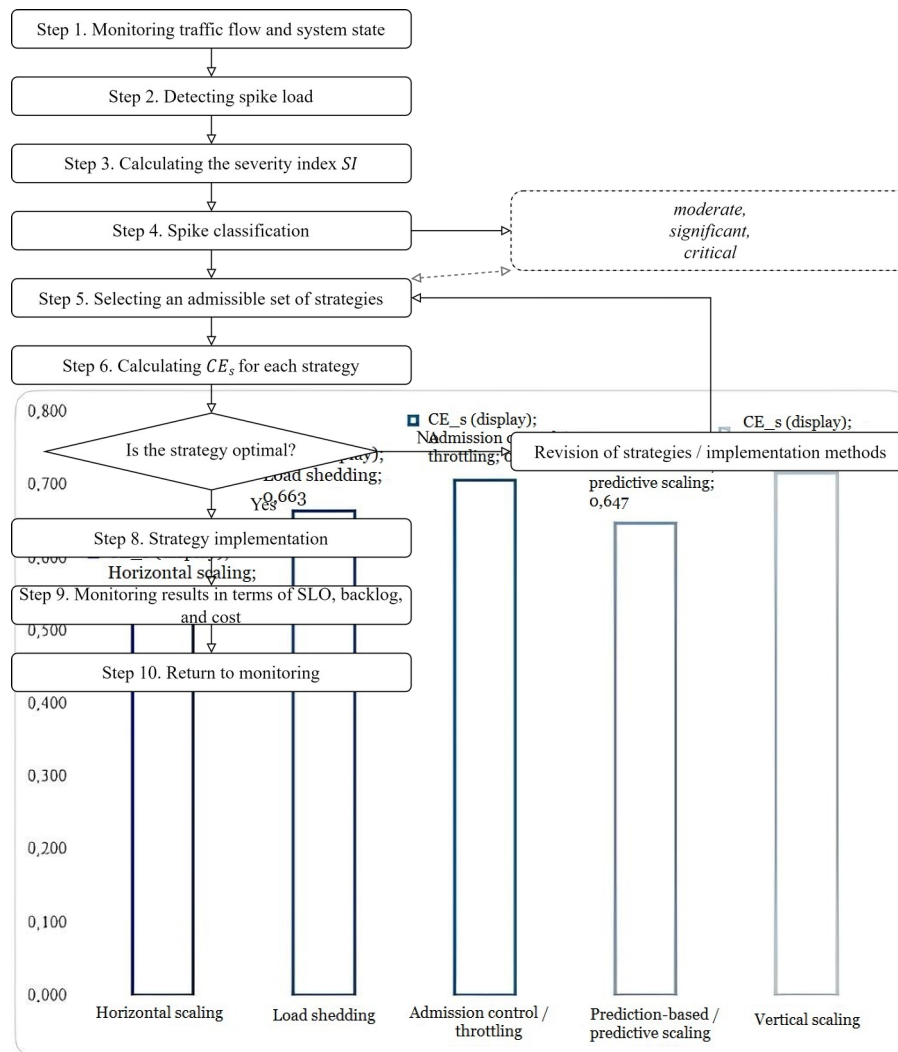


Figure 3 - Decision-making algorithm for spike loads

DOI: <https://doi.org/10.60797/COMP.2026.12.1.4>



Accordingly, the algorithm is based on applying the proposed model with the determination of optimal cost parameters; after implementing the selected measure, backlog, latency, and the cost of the response are monitored. If critical symptoms persist, the procedure is repeated with updated input parameters until optimal values are achieved.

To reduce the purely theoretical nature of the proposed model, a compact calculation-based validation was carried out. Azure Functions Trace 2019 was selected as an empirical reference point because it reflects real serverless/event-driven workloads and contains invocation counts in 1-minute intervals, trigger types, execution time, and memory usage. This ensures that the proposed model is evaluated against a relevant class of real event-driven workloads rather than an abstract traffic profile [9].

For the numerical validation of the cost-efficiency indicator, the published PASS benchmark data were used. The evaluated strategies include threshold-based rules, target tracking, and a queuing-theory-based strategy. Since the PASS results contain QoS guarantee rate for average response time, QoS guarantee rate for TP999 tail latency, and resource cost, the following compact empirical specification was used:

$$CE_{test} = 0,35 * QoS_{avgRT} + 0,40 * QoS_{TP999RT} + 0,25 * (1 - C_{norm}),$$

where C_{norm} is the resource cost of a strategy normalized by the maximum resource cost among the compared methods. The results show that the queuing-theory-based strategy has the highest cost-efficiency score, despite not having the lowest resource cost. This can be attributed to its higher QoS guarantee rate, especially for TP999 tail latency, which is critical for high-load systems with strict SLO requirements. Therefore, the validation supports the main argument of the study: the response to spike load should not be selected only by resource expansion logic, but through the joint assessment of QoS preservation, response cost, and overload dynamics.

A sensitivity check was also performed by varying the weights in CE_{test} within a $\pm 15\%$ range and normalizing their sum to one. The ranking of strategies remained stable: the queuing-theory-based strategy retained the highest score, target tracking remained in the middle position, and threshold-based rules retained the lowest score. This indicates that the result is not determined by an arbitrary choice of a single coefficient and that the proposed indicator can be used as a preliminary decision-support tool.

Conclusion

Thus, the conducted study demonstrates that managing spike patterns in high-load event-driven systems requires moving away from the simplified assumption that an increase in incoming traffic should automatically trigger horizontal scaling. The proliferation of stream-oriented architectures, together with stricter SLO/SLA requirements and rising cloud resource costs, makes such a response economically vulnerable. Cost-efficient management strategies serve as alternatives to horizontal scaling and may take various forms, on the basis of which an original approach is proposed.

The calculation-based validation provides supporting evidence for the practical applicability of the proposed model. Using published benchmark data, it was shown that the integral cost-efficiency indicator produces a differentiated ranking of response strategies and remains stable under moderate variation of weighting coefficients. Therefore, the proposed approach can be used as a preliminary decision-support framework for selecting cost-efficient responses to spike loads in event-driven and stream-oriented systems.

Конфликт интересов

Не указан.

Рецензия

Все статьи проходят рецензирование. Но рецензент или автор статьи предпочли не публиковать рецензию к этой статье в открытом доступе. Рецензия может быть предоставлена компетентным органам по запросу.

Conflict of Interest

None declared.

Review

All articles are peer-reviewed. But the reviewer or the author of the article chose not to publish a review of this article in the public domain. The review can be provided to the competent authorities upon request.

Список литературы на английском языке / References in English

1. Cardellini V. Runtime adaptation of data stream processing systems. The state of the art / V. Cardellini, F.L. Presti, M. Nardelli et al. // ACM Comput Surv. — 2022. — Vol. 54, № 11. — DOI: 10.1145/3514496.
2. Wang Y. The non-expert tax. Quantifying the cost of auto-scaling in cloud-based data stream analytics / Y. Wang, B. Lyu, V. Kalavri // Proceedings of the International Workshop on Big Data in Emergent Distributed Environments. — 2022. — P. 1–7. — DOI: 10.1145/3530050.3532925.
3. Lu L. Awaiting. Efficient overload management for busy multi-tier web services under bursty workload / L. Lu, L. Cherkasova, V. de Nitto Personè et al. // Web Engineerin. — 2010. — Vol. 618. — DOI: 10.1007/978-3-642-13911-6_6.
4. Fé I. Performance-cost trade-off in auto-scaling mechanisms for cloud computing / I. Fé, R. Matos, J. Dantas et al. // Sensors. — 2022. — Vol. 22, № 3. — DOI: 10.3390/s22031221.
5. Tatbul N. Staying FIT. Efficient load shedding techniques for distributed stream processing / N. Tatbul, U. Cetintemel, S. Zdonik // Proceedings of the 33rd International Conference on Very Large Data Bases. — 2007. — P. 1598–1601.
6. Ashraf A. Prediction-based VM provisioning and admission control for multi-tier web applications / A. Ashraf, B. Byholm, I. Porres // J. Cloud Comp. — 2016. — Vol. 5. — Art. 15. — DOI: 10.1186/s13677-016-0065-9.
7. Guo Y. Predictive auto-scaling system for large-scale enterprise web applications / Y. Guo, J. Ge, P. Guo et al. // Proceedings of the ACM Web Conference 2024. — 2023. — P. 2747–2758. — DOI: 10.1145/3589334.3645330.
8. Peri A. Vertical scaling can save time. Optimizing container scheduling to handle sudden bursts / A. Peri, M. Tsenos, V. Kalogeraki // Proceedings of the 19th ACM International Conference on Distributed and Event-based Systems. — 2025. — P. 86–97. — DOI: 10.1145/3701717.3730549.



9. Shahrads M. Serverless in the wild. Characterizing and optimizing the serverless workload at a large cloud provider / M. Shahrads, R. Fonseca, I. Goiri et al. // Proceedings of the 2020 USENIX Annual Technical Conference. — 2020. — P. 205–218.