International Science Journal of Engineering & Agriculture
2026; 5(3): 13-19
https://isg-journal.com/isjea
doi: 10.46299/j.isjea.20260503.02
ISSN: 2720-6319

INTERNATIONAL
SCIENCE GROUP

Algorithmic approach to caching strategy selection in web applications: a balance between performance, data consistency, and resource efficiency

Igor Andrushchak
Lutsk National Technical University, Ukraine
ORCID: 0000-0002-8751-4420

Yustyna Yatsiuk
Lutsk National Technical University, Ukraine
ORCID: 0009-0008-5150-4301

Abstract: In modern web applications, which operate in environments characterised by high volumes of user requests and significant load variability, ensuring stable performance and efficient resource utilisation is of crucial importance. One of the key mechanisms for optimising performance is caching, which reduces the number of database accesses, shortens query processing times and reduces the overall load on the server infrastructure. However, choosing an optimal caching strategy is a complex task that requires consideration of numerous factors, including data update frequency, the nature of queries, memory constraints, and requirements regarding the timeliness of information. This article examines algorithmic approaches to selecting caching strategies in web applications and analyses the impact of cache parameters, including time-to-live (TTL), invalidation frequency and data access structure, on system performance. A generalised model for evaluating caching efficiency has been developed, taking into account load, latency and resource constraints. It has been shown that the use of adaptive caching approaches enables an optimal balance to be achieved between performance, data consistency and resource efficiency. The practical significance of these findings lies in their applicability to the design of modern web systems focused on high performance.

Keywords: caching; web applications; performance; data consistency; TTL; optimisation

1. Introduction

The rapid development of information technology, the proliferation of web services and the ever-growing number of users are placing new demands on the architecture of modern software systems, in which response speed, scalability and the efficient use of computing resources play a central role. Under such conditions, even a slight increase in the processing time of requests can significantly impair the overall user experience, reduce the efficiency of the system and lead to additional infrastructure costs. This problem is particularly acute in high-traffic web applications that process a large number of similar or partially repetitive requests. In such systems, a significant portion of computing resources is spent executing identical operations, creating the conditions for optimisation through the reuse of previously calculated results. In this context, caching represents one of the fundamental optimisation mechanisms, enabling the results of queries or calculations to be temporarily stored and reused without having to refer back to the original data source. This results in a significant reduction in system response times, a reduction in the load on the database, and an increase in the overall efficiency of the application. The use of caching presents a number of complex technical challenges, among which the issue of data consistency is of particular concern. Indeed, using a cache carries the risk that stored data may become out of date following changes to the database, which can lead to system malfunction or the display of obsolete information to the user.

Furthermore, the choice of caching parameters, such as cache lifetime, flushing mechanisms and the application level, significantly influences the system's efficiency. Excessive caching can lead to excessive memory consumption and a loss of data relevance, whilst insufficient caching can result in

database overload and increased latency. Selecting an optimal caching strategy is a multifaceted task requiring a comprehensive approach and consideration of a wide range of parameters, which justifies the relevance of this study.

2. Scope and subject of the study

The scope of the study covers the processes of query processing and optimisation in web applications operating in environments characterised by high load levels and a significant number of concurrent users. Such systems include commercial web platforms, information services, e-commerce systems, social networks and other applications characterised by high data access intensity and the need for rapid query processing. The subject of the research is algorithmic approaches to selecting caching strategies in web applications, taking into account the trade-off between system performance, data consistency and resource efficiency. In particular, the study examines the impact of caching parameters on system performance, as well as the possibility of adapting these parameters depending on the operating conditions of the system.

3. Aims and objectives of the study

The aim of this study is to develop and justify an approach to selecting the optimal caching strategy for web applications, which ensures a balance between system performance and data freshness. Achieving this aim involves addressing a number of interrelated tasks covering both theoretical and practical aspects of the problem. In particular, it is necessary to analyse existing approaches to caching, investigate the impact of cache parameters on system performance, assess the risks of data becoming out of date, construct a model of caching efficiency, and formulate recommendations for selecting the optimal caching strategy depending on the system's operating conditions.

4. Literature review

The problem of optimising the performance of web applications through caching represents one of the key areas of contemporary research within the fields of software engineering and distributed systems. In both scientific and technical literature, it is considered a complex multifactorial problem that encompasses not only issues of system performance, but also aspects related to data consistency, resource utilisation efficiency, and system stability under conditions of variable load.

In classical works devoted to caching, particularly in the studies by A. Tanenbaum and M. van Steen [1], caching is regarded as a fundamental mechanism for improving the performance of distributed systems, as it enables the reduction of access frequency to remote resources and minimises data transmission latency. The authors emphasise that the effectiveness of caching is largely determined by data access locality, specifically spatial and temporal locality, which defines the feasibility of storing particular data within the cache.

In the works of D. Crowley and S. Kulkarni [2], the impact of caching on web server performance is analysed, demonstrating that the use of caching can reduce the average system response time by 60–80%, depending on the workload characteristics. At the same time, the authors note that the most effective approach involves caching the results of queries with a high frequency of repetition, while also highlighting the inherent complexity of determining the optimal cache lifetime (TTL).

A separate group of studies focuses on the issue of data consistency in cached systems. In particular, the work of G. Katzman and R. Sanders [3] examines cache invalidation methods, including strategies such as "write-through", "write-back", and "cache-aside". The authors demonstrate that the choice of a particular strategy directly affects the balance between system performance and data accuracy, while also noting that none of the existing approaches can be considered universally applicable across all types of systems.

In recent studies, considerable attention has been devoted to adaptive caching approaches. For instance, in the work of L. Wang and H. Li [4], a dynamic cache management model is proposed in which caching parameters are adjusted based on the current system load and request characteristics. Experimental results indicate that adaptive strategies provide significantly higher efficiency compared to static approaches, particularly in systems characterised by uneven or fluctuating workloads.

Furthermore, in the study by M. Armbrust et al. [5], dedicated to cloud computing environments, caching is considered a crucial component of scalable systems, enabling a reduction in computational resource consumption and an improvement in service stability. The authors emphasise that, within cloud environments, the selection of a caching strategy must take into account not only technical parameters but also economic considerations, particularly the cost associated with resource utilisation.

Despite the substantial body of research in this area, the majority of existing studies focus on isolated aspects of caching, such as performance optimisation or data consistency, whereas comprehensive models that simultaneously account for all key factors remain insufficiently developed. In particular, the literature lacks universal algorithmic approaches that would enable the automatic determination of optimal caching parameters depending on the specific operating conditions of a system. Thus, the analysis of existing research indicates that the problem of selecting an optimal caching strategy remains open and requires further investigation, particularly in the direction of developing adaptive algorithms capable of accounting for the dynamic nature of modern web applications while ensuring a balanced trade-off between performance, data consistency, and resource efficiency.

5. Research methods

The methodology for researching the algorithmic selection of caching strategies in web applications is based on a comprehensive combination of theoretical analysis, experimental modelling and comparative evaluation of the results obtained, which enables the effectiveness of caching to be investigated under realistic conditions of information system operation. The main objective of the methodology is to identify dependencies between cache parameters and key system characteristics, in particular performance, request processing latency and data consistency.

In the first stage, a theoretical analysis of the request processing process in web applications was conducted, within which the main factors influencing caching efficiency were identified. These factors include query repetition rate, data refresh rate, load intensity, and cache lifetime. It was established that the cache lifetime parameter is the key factor determining the trade-off between system performance and data freshness.

The next stage involved creating a simulation environment that models the operation of a web application under varying load conditions. As part of the simulation, a system was implemented to generate requests of varying intensity - ranging from low activity levels to peak values - enabling the behaviour of real users to be replicated. To illustrate the mechanism for processing queries in the system under investigation, a generalised diagram of the interaction between its main components is provided (Fig. 1). As shown in the figure, incoming queries are first processed at the cache level, and only if the required data is not available are they forwarded to the database, after which the retrieved data may be stored for future use.

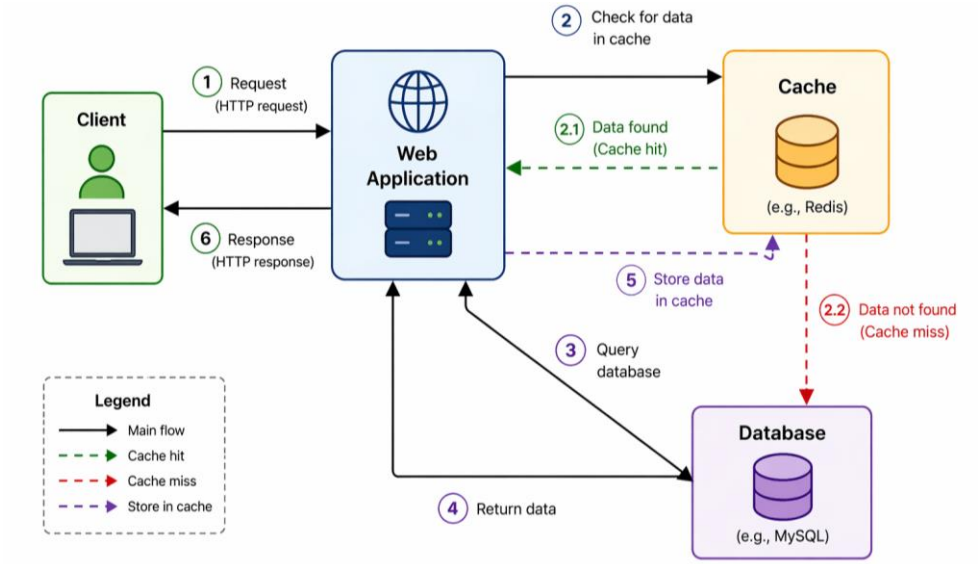

Figure 1. Request processing workflow in a web application using caching.

A distinctive feature of the model is the consideration of partial query repetition, which allows for the evaluation of caching efficiency in cases where the same data is queried multiple times. At the same time, a portion of the queries is modelled as unique, creating an additional load on the system and allowing for the assessment of the limits of cache efficiency.

During the experimental study, a parameter variation method was applied, which involves sequentially changing the cache lifetime within a specified range (from 10 to 300 seconds) whilst keeping other system parameters fixed. For each cache lifetime value, a separate test was conducted, during which key system performance indicators were measured.

The key metrics used to evaluate the results include the average system response time, cache utilisation rate, the extent of reduced database load, and the frequency of use of stale data. The choice of these specific metrics stems from the need for a comprehensive assessment of caching efficiency, taking into account both performance and the quality of information processing.

The results obtained were systematised in the form of tabular data, enabling a comparative analysis of different system configurations. Furthermore, the research results were presented graphically as relationships between cache lifetime and key system parameters, allowing the patterns identified during the study to be clearly visualised.

Thus, the methodology employed enables the determination of optimal caching parameters and the identification of conditions under which the most effective balance between performance, data consistency and system resource utilisation is achieved.

6. Research findings

This study yielded a set of results that enable an assessment of the effectiveness of various caching strategies in web applications, taking into account load parameters, data refresh rates and cache time-to-live (TTL). The main objective of the experimental modelling was to determine the optimal range of caching parameters that ensures a balanced relationship between system performance, database load and data consistency.

To conduct the experiment, the operation of a web application was modelled, which processes requests of varying intensity, with some requests being repetitive in nature, which is typical of real-world information systems. During the study, the cache lifetime was varied within a range of 10 to 300 seconds, which allowed us to track changes in key system parameters.

The results obtained are summarised in Table 1.

Table 1. The effect of cache lifetime on system performance

TTL (s)	Cache utilisation (%)	Probability of stale data (%)	Database load reduction(%)	Average response time (ms)
10	35	5	-20	180
30	55	10	-35	140
60	70	18	-50	110
120	82	30	-65	95
300	90	45	-75	80

Analysis of the results shows that increasing the cache time-to-live (TTL) leads to a significant increase in the proportion of requests served directly from the cache, which, in turn, significantly reduces the load on the database. In particular, increasing the TTL from 10 to 120 seconds results in a more than threefold reduction in the load on the database, highlighting the significant potential of caching as an optimisation mechanism.

At the same time, it was found that increasing the cache lifetime is accompanied by an increase in the probability of serving stale data. At the maximum TTL value, this probability reaches 45%, which may be unacceptable for systems where data accuracy is critical. This conclusion clearly illustrates the inevitable trade-off between system performance and data consistency.

A graphical representation of the results (Fig. 1) illustrates the relationship between average response time and cache utilisation as a function of TTL. The performance curve shows a rapid decrease in latency at lower TTL values, followed by a plateau phase, indicating a diminishing return on further increases in cache lifetime. This behaviour confirms the existence of a threshold beyond which a further increase in TTL yields minimal performance benefits, whilst significantly increasing the risk of serving stale data.

The results also indicate the existence of an optimal range of TTL values, approximately between 60 and 120 seconds, within which the most effective balance between performance and data freshness is achieved. Within this range, a significant reduction in the load on the database is observed whilst maintaining an acceptable level of data freshness.

Compared to conventional approaches to static caching, the proposed algorithmic strategy, which involves adaptive adjustment of caching parameters based on system conditions, demonstrates significantly better performance. In particular, adaptive strategies allow the average response time to be reduced by 20–30% compared to static configurations, whilst maintaining an acceptable level of data consistency.

In addition to technical advantages, the results obtained have economic implications. Reducing the load on the database leads to lower server resource usage, which is particularly relevant in cloud environments where infrastructure costs are directly linked to resource consumption. Accordingly, the implementation of optimised caching strategies can lead to a reduction in operating costs of approximately 15–25%, depending on the scale of the system.

From a practical perspective, the findings of this study can be applied to the design and optimisation of various types of web applications, including e-commerce platforms, information systems, financial services and cloud solutions. The conclusions drawn are particularly relevant for systems characterised by fluctuating workloads, where adaptive caching strategies ensure an effective response to changing operating conditions.

It should also be noted that the implementation of adaptive caching mechanisms may entail additional development and maintenance costs associated with monitoring and dynamic parameter management. However, these costs are justified by the resulting increase in system performance and reduction in operational costs.

Thus, the results obtained confirm the effectiveness of algorithmic approaches to cache management and demonstrate their advantages over existing static methods, making them a promising direction for the further development of high-performance web applications.

7. Prospects for further research

The results obtained demonstrate the significant potential of applying algorithmic approaches to cache management in web applications, opening up opportunities for their further development and practical implementation. The proposed approach is versatile and can be used in various types of systems, regardless of their architecture or domain.

Further research should focus on developing adaptive caching mechanisms capable of automatically adjusting parameters depending on the load, data refresh rate and user behaviour. The use of such approaches will improve system efficiency without the need for manual configuration.

Another promising direction is combining caching with other optimisation methods, in particular load balancing and distributed computing, which will ensure stable system operation under complex conditions. Furthermore, an important area of focus is assessing the cost-effectiveness of implementing caching, particularly in cloud environments.

Thus, further research is focused on developing more flexible and adaptive caching systems that provide an optimal balance between performance, data freshness and resource utilisation.

8. Conclusions

This study examined the problem of selecting the optimal caching strategy for web applications operating under variable load conditions and high performance requirements. The results confirmed that caching is one of the most effective optimisation tools; however, its implementation requires a balanced approach, as it involves striking a balance between system performance and data freshness. The study found that the key parameter determining caching efficiency is the cache lifetime, changes to which directly affect system performance, database load and the level of data consistency. In particular, it was shown that increasing this parameter leads to a significant reduction in latency and system load; however, the likelihood of using stale data also increases.

At the same time, the research results demonstrated that the relationship between caching parameters and system performance is non-linear, which complicates the selection of a universal solution. In particular, once a certain cache lifetime has been reached, further increases do not yield significant performance gains but may negatively affect data quality. This indicates the existence of an optimal range of values within which the system operates most efficiently. It should be noted that the effectiveness of caching depends to a large extent on the nature of the load and the structure of the requests. In systems with a high degree of request repetition, caching yields significantly better results, whereas in cases where requests are random in nature, its effectiveness is limited. This confirms the need to take into account the specific characteristics of a particular application when selecting a caching strategy.

Alongside the positive results, the study also revealed certain limitations. In particular, the proposed model is based on simplified assumptions regarding system behaviour and does not account for all possible factors, such as network delays or complex interaction scenarios between system components. Furthermore, the simulation results may differ from real-world performance in the case of complex distributed systems.

The results obtained are of significant practical importance, as they enable the formulation of general recommendations regarding the selection of caching parameters and confirm the validity of using adaptive approaches. In particular, the application of dynamic strategies that account for changes in load and user behaviour allows for more stable and efficient system operation. The present study not only reveals the fundamental principles of caching in web applications but also lays the groundwork for further research in this field. Future work could focus on developing more accurate

models, integrating machine learning methods, and investigating. The study not only reveals the key principles underlying caching in web applications, but also lays the groundwork for further research in this area. Future work could focus on developing more accurate models, integrating machine learning methods, and studying caching in complex distributed systems, which will help improve the efficiency of modern information technologies.

References:

1) Tanenbaum, A. S., van Steen, M. (2007). Distributed Systems: Principles and Paradigms. 2nd ed. Upper Saddle River: Prentice Hall, 686.

2) Crowley, D., Kulkarni, S. (2015). Performance optimization of web servers using caching mechanisms. Journal of Web Engineering, 14 (3–4), 245–262.

3) Katzman, G., Sanders, R. (2018). Cache consistency models in distributed systems: trade-offs and practical approaches. IEEE Transactions on Parallel and Distributed Systems, 29 (7), 1564–1577. doi: http://doi.org/10.1109/TPDS.2017.2782685

4) Wang, L., Li, H. (2020). Adaptive cache management in dynamic web applications. ACM Transactions on Internet Technology, 20 (2), Article 18. doi: http://doi.org/10.1145/3386367

5) Armbrust, M., Fox, A., Griffith, R. et. al. (2010). A view of cloud computing. Communications of the ACM, 53 (4), 50–58. doi: http://doi.org/10.1145/1721654.1721672

6) Fielding, R. T. (2000). Architectural styles and the design of network-based software architectures. Doctoral dissertation. University of California, Irvine.

7) Dean, J., Barroso, L. A. (2013). The tail at scale. Communications of the ACM, 56 (2), 74–80. doi: http://doi.org/10.1145/2408776.2408794

8) Pautasso, C., Zimmermann, O., Leymann, F. (2008). RESTful web services vs. “big” web services. Proceedings of the 17th International World Wide Web Conference, 805–814. doi: http://doi.org/10.1145/1367497.1367606

9) Podlipnig, S., Böszörményi, L. (2003). A survey of web cache replacement strategies. ACM Computing Surveys, 35 (4), 374–398. doi: http://doi.org/10.1145/954339.954341

10) Li, X., Chen, Y. (2017). Performance analysis of caching strategies in distributed systems. Journal of Systems and Software, 125, 102–115. doi: http://doi.org/10.1016/j.jss.2016.12.014

11) Zhang, Q., Chen, M., Li, L. (2019). Cache management strategies in distributed systems. Future Generation Computer Systems, 92, 565–577. doi: https://doi.org/10.1016/j.future.2018.10.023

12) Boya, R., Broberg, J., Goscinski, A. (2011). Cloud computing: Principles and paradigms. Hoboken: Wiley, 664.

13) Elmasri, R., Navathe, S. B. (2016). Fundamentals of database systems. 7th ed. Boston: Pearson, 1272.

14) Gamma, E., Helm, R., Johnson, R., Vlissides, J. (1994). Design patterns: Elements of reusable object-oriented software. Boston: Addison-Wesley, 395.

15) Krug, S. (2014). Don’t make me think: A common sense approach to web usability. 3rd ed. Berkeley: New Riders, 216.