

ISSN 2518-1726 (Online),

ISSN 1991-346X (Print)

**ACADEMIC SCIENTIFIC
JOURNAL OF COMPUTER SCIENCE**

**№2
2026**

ISSN 2518-1726 (Online),
ISSN 1991-346X (Print)



ACADEMIC SCIENTIFIC JOURNAL OF COMPUTER SCIENCE

3 (359)

JULY – SEPTEMBER 2026

**PUBLISHED SINCE JANUARY 1963
PUBLISHED 4 TIMES A YEAR**

ALMATY, NAS RK

Chief Editor:

MUTANOV Galimkair Mutanovich, doctor of technical sciences, professor, academician of NAS RK, (Almaty, Kazakhstan), <https://www.scopus.com/authid/detail.uri?authorId=6506682964>, <https://www.webofscience.com/wos/author/record/1423665>

EDITORIAL BOARD:

KALIMOLDAYEV Maksat Nuradilovich, (Deputy Editor-in-Chief), Doctor of Physical and Mathematical Sciences, Professor, Academician of NAS RK, Advisor to the General Director of the Institute of Information and Computing Technologies of the CS MES RK, Head of the Laboratory (Almaty, Kazakhstan), <https://www.scopus.com/authid/detail.uri?authorId=56153126500>, <https://www.webofscience.com/wos/author/record/2428551>

MAMYRBAEV Orken Zhumazhanovich, (Academic Secretary), PhD in Information Systems, Deputy Director for Science of the Institute of Information and Computing Technologies CS MES RK (Almaty, Kazakhstan), <https://www.scopus.com/authid/detail.uri?authorId=55967630400>, <https://www.webofscience.com/wos/author/record/1774027>

BAIGUNCHEKOV Zhumadil Zhanabaevich, Doctor of Technical Sciences, Professor, Academician of NAS RK, Institute of Cybernetics and Information Technologies, Department of Applied Mechanics and Engineering Graphics, Satbayev University (Almaty, Kazakhstan), <https://www.scopus.com/authid/detail.uri?authorId=6506823633>, <https://www.webofscience.com/wos/author/record/1923423>

WOICIK Waldemar, Doctor of Technical Sciences (Phys.-Math.), Professor of the Lublin University of Technology (Lublin, Poland), <https://www.scopus.com/authid/detail.uri?authorId=7005121594>, <https://www.webofscience.com/wos/author/record/678586>

SMOLARJ Andrej, Associate Professor Faculty of Electronics, Lublin polytechnic university (Lublin, Poland), <https://www.scopus.com/authid/detail.uri?authorId=56249263000>, <https://www.webofscience.com/wos/author/record/1268523>

KEILAN Alimkhan, Doctor of Technical Sciences, Professor (Doctor of science (Japan)), chief researcher of Institute of Information and Computational Technologies CS MES RK (Almaty, Kazakhstan), <https://www.scopus.com/authid/detail.uri?authorId=8701101900>, <https://www.webofscience.com/wos/author/record/1436451>

KHAIROVA Nina, Doctor of Technical Sciences, Professor, Chief Researcher of the Institute of Information and Computational Technologies CS MES RK (Almaty, Kazakhstan), <https://www.scopus.com/authid/detail.uri?authorId=37461441200>, <https://www.webofscience.com/wos/author/record/1768515>

OTMAN Mohamed, PhD, Professor of Computer Science Department of Communication Technology and Networks, Putra University Malaysia (Selangor, Malaysia), <https://www.scopus.com/authid/detail.uri?authorId=56036884700>, <https://www.webofscience.com/wos/author/record/747649>

NYSANBAYEVA Saule Yerkebulanovna, Doctor of Technical Sciences, Associate Professor, Senior Researcher of the Institute of Information and Computing Technologies CS MES RK (Almaty, Kazakhstan), <https://www.scopus.com/authid/detail.uri?authorId=55453992600>, <https://www.webofscience.com/wos/author/record/3802041>

USATOVA Olga Alexandrovna, PhD, Associate Professor, Chief Scientific Secretary of the Institute of Information and Computing Technologies of the National Academy of Sciences of the Republic of Kazakhstan (Almaty, Kazakhstan), <https://www.scopus.com/authid/detail.uri?authorId=57204581062>, <https://www.webofscience.com/wos/author/record/JCO-3058-2023>

KAPALOVA Nursulu Aldazharovna, Candidate of Technical Sciences, Head of the Laboratory cybersecurity, Institute of Information and Computing Technologies CS MES RK (Almaty, Kazakhstan), <https://www.scopus.com/authid/detail.uri?authorId=57191242124>,

KOVALYOV Alexander Mikhailovich, Doctor of Physical and Mathematical Sciences, Academician of the National Academy of Sciences of Ukraine, Institute of Applied Mathematics and Mechanics (Donetsk, Ukraine), <https://www.scopus.com/authid/detail.uri?authorId=7202799321>, <https://www.webofscience.com/wos/author/record/38481396>

MIKHALEVICH Alexander Alexandrovich, Doctor of Technical Sciences, Professor, Academician of the National Academy of Sciences of Belarus (Minsk, Belarus), <https://www.scopus.com/authid/detail.uri?authorId=7004159952>, <https://www.webofscience.com/wos/author/record/46249977>

TIGHINEANU Ion Mihailovich, Doctor of Physical and Mathematical Sciences, Academician, President of the Academy of Sciences of Moldova, Technical University of Moldova (Chisinau, Moldova), <https://www.scopus.com/authid/detail.uri?authorId=7006315935>, <https://www.webofscience.com/wos/author/record/524462>

Academic Scientific Journal of Computer Science

ISSN 2518-1726 (Online),

ISSN 1991-346X (Print)

Owner: «Central Asian Academic Research Center» LLP (Almaty).

Certificate № KZ77VPY00121154 on the re-registration of the periodical printed and online publication of the information agency, issued on 05.06.2025 by the Republican State Institution «Information Committee» of the Ministry of Culture and Information of the Republic of Kazakhstan

Subject area: *information and communication technologies*.

Currently: *included in the list of journals recommended by the CCSES MSHE RK in the direction of «Information and communication technologies»*.

Periodicity: 4 times a year.

<http://www.physico-mathematical.kz/index.php/en/>

© «Central Asian Academic Research Center» LLP, 2026

БАС РЕДАКТОР:

МУТАНОВ Ғалымқайыр Мұтанұлы, техника ғылымдарының докторы, профессор, ҚР ҰҒА академигі, (Алматы, Қазақстан), <https://www.scopus.com/authid/detail.uri?authorId=6506682964>, <https://www.webofscience.com/wos/author/record/1423665>

РЕДАКЦИЯ АЛҚАСЫ:

КАЛИМОЛДАЕВ Мақсат Нұрәділұлы, (бас редактордың орынбасары), физика-математика ғылымдарының докторы, профессор, ҚР ҰҒА академигі, ҚР ҒЖБМ ҒК «Ақпараттық және есептеу технологиялары институты» бас директорының кеңесшісі, зертхана меңгерушісі (Алматы, Қазақстан), <https://www.scopus.com/authid/detail.uri?authorId=56153126500>, <https://www.webofscience.com/wos/author/record/2428551>

МАМЫРБАЕВ Өркен Жұмажанұлы (ғалым хатшы), Ақпараттық жүйелер саласындағы техника ғылымдарының (PhD) докторы, ҚР ҒЖБМ ҒК «Ақпараттық және есептеу технологиялары институты» директорының ғылым жөніндегі орынбасары (Алматы, Қазақстан), <https://www.scopus.com/authid/detail.uri?authorId=55967630400>, <https://www.webofscience.com/wos/author/record/1774027>

БАЙГУНЧЕКОВ Жұмаіл Жаңабайұлы, техника ғылымдарының докторы, профессор, ҚР ҰҒА академигі, Кибернетика және ақпараттық технологиялар институты, Қолданбалы механика және инженерлік графика кафедрасы, Сәтбаев университеті (Алматы, Қазақстан), <https://www.scopus.com/authid/detail.uri?authorId=6506823633>, <https://www.webofscience.com/wos/author/record/1923423>

ВОЙЧИК Вальдемар, техника ғылымдарының докторы (физ-мат), Люблин технологиялық университетінің профессоры (Люблин, Польша), <https://www.scopus.com/authid/detail.uri?authorId=7005121594>, <https://www.webofscience.com/wos/author/record/678586>

СМОЛАРЖ Анджей, Люблин политехникалық университетінің электроника факультетінің доценті (Люблин, Польша), <https://www.scopus.com/authid/detail.uri?authorId=56249263000>, <https://www.webofscience.com/wos/author/record/1268523>

КЕЙЛАН Әлімхан, техника ғылымдарының докторы, профессор (ғылым докторы (Жапония)), ҚР ҒЖБМ ҒК «Ақпараттық және есептеу технологиялары институтының» бас ғылыми қызметкері (Алматы, Қазақстан), <https://www.scopus.com/authid/detail.uri?authorId=8701101900>, <https://www.webofscience.com/wos/author/record/1436451>

ХАЙРОВА Нина, техника ғылымдарының докторы, профессор, ҚР ҒЖБМ ҒК «Ақпараттық және есептеу технологиялары институтының» бас ғылыми қызметкері (Алматы, Қазақстан), <https://www.scopus.com/authid/detail.uri?authorId=37461441200>, <https://www.webofscience.com/wos/author/record/1768515>

ОТМАН Мохаммед, PhD, Информатика, Коммуникациялық технологиялар және желілер кафедрасының профессоры, Путра университеті Малайзия (Селангор, Малайзия), <https://www.scopus.com/authid/detail.uri?authorId=56036884700>, <https://www.webofscience.com/wos/author/record/747649>

НЫСАНБАЕВА Сауле Еркебұланқызы, техника ғылымдарының докторы, доцент, ҚР ҒЖБМ ҒК «Ақпараттық және есептеу технологиялары институтының» аға ғылыми қызметкері (Алматы, Қазақстан), <https://www.scopus.com/authid/detail.uri?authorId=55453992600>, <https://www.webofscience.com/wos/author/record/3802041>

УСАТОВА Ольга Александровна, PhD, қауымдастырылған профессор, ҚР ҒЖБМ "Ақпараттық және есептеу технологиялары институтының" бас ғалым хатшысы (Алматы, Қазақстан), <https://www.scopus.com/authid/detail.uri?authorId=57204581062>, <https://www.webofscience.com/wos/author/record/JCO-3058-2023>

КАПАЛОВА Нұрсұлу Алдажарқызы, техника ғылымдарының кандидаты, ҚР ҒЖБМ ҒК «Ақпараттық және есептеу технологиялары институты», Киберқауіпсіздік зертханасының меңгерушісі (Алматы, Қазақстан), <https://www.scopus.com/authid/detail.uri?authorId=57191242124>,

КОВАЛЕВ Александр Михайлович, физика-математика ғылымдарының докторы, Украина Ұлттық Ғылым академиясының академигі, Қолданбалы математика және механика институты (Донецк, Украина), <https://www.scopus.com/authid/detail.uri?authorId=7202799321>, <https://www.webofscience.com/wos/author/record/38481396>

МИХАЛЕВИЧ Александр Александрович, техника ғылымдарының докторы, профессор, Беларусь Ұлттық Ғылым академиясының академигі (Минск, Беларусь), <https://www.scopus.com/authid/detail.uri?authorId=7004159952>, <https://www.webofscience.com/wos/author/record/46249977>

ТИГИНЯНУ Ион Михайлович, физика-математика ғылымдарының докторы, академик, Молдова Ғылым Академиясының президенті, Молдова техникалық университеті (Кишинев, Молдова), <https://www.scopus.com/authid/detail.uri?authorId=7006315935>, <https://www.webofscience.com/wos/author/record/524462>

Academic Scientific Journal of Computer Science

ISSN 2518-1726 (Online),

ISSN 1991-346X (Print)

Меншіктеуші: «Орталық Азия академиялық ғылыми орталығы» ЖШС (Алматы).

Ақпарат агенттігінің мерзімді баспасөз басылымын, ақпарат агенттігін және желілік басылымды қайта есепке қою туралы ҚР Мәдениет және Ақпарат министрлігі «Ақпарат комитеті» Республикалық мемлекеттік мекемесі **05.06.2025 ж.** берген **№ KZ77VPY00121154** Куәлік.

Тақырыптық бағыты: *ақпараттық-коммуникациялық технологиялар*

Қазіргі уақытта: *«ақпараттық-коммуникациялық технологиялар» бағыты бойынша ҚР БҒМ БҒСБК ұсынған журналдар тізіміне енді.*

Мерзімділігі: жылына 4 рет.

<http://www.physico-mathematical.kz/index.php/en/>

© «Орталық Азия академиялық ғылыми орталығы» ЖШС, 2026

Главный редактор:

МУТАНОВ Галимканр Мутанович, доктор технических наук, профессор, академик НАН РК, (Алматы, Казахстан), <https://www.scopus.com/authid/detail.uri?authorId=6506682964>, <https://www.webofscience.com/wos/author/record/1423665>

Редакционная коллегия:

КАЛИМОЛДАЕВ Максат Нурадилович, (заместитель главного редактора), доктор физико-математических наук, профессор, академик НАН РК, советник генерального директора «Института информационных и вычислительных технологий» КН МНВО РК, заведующий лабораторией (Алматы, Казахстан), <https://www.scopus.com/authid/detail.uri?authorId=56153126500>, <https://www.webofscience.com/wos/author/record/2428551>

МАМЫРБАЕВ Оркен Жумажанович, (ученый секретарь), доктор философии (PhD) по специальности «Информационные системы», заместитель директора по науке РГП «Институт информационных и вычислительных технологий» Комитета науки МНВО РК (Алматы, Казахстан), <https://www.scopus.com/authid/detail.uri?authorId=55967630400>, <https://www.webofscience.com/wos/author/record/1774027>

БАЙГУНЧЕКОВ Жумадил Жанабаевич, доктор технических наук, профессор, академик НАН РК, Институт кибернетики и информационных технологий, кафедра прикладной механики и инженерной графики, Университет Сатпаева (Алматы, Казахстан), <https://www.scopus.com/authid/detail.uri?authorId=6506823633>, <https://www.webofscience.com/wos/author/record/1923423>

ВОЙЧИК Вальдемар, доктор технических наук (физ.-мат.), профессор Люблинского технологического университета (Люблин, Польша), <https://www.scopus.com/authid/detail.uri?authorId=7005121594>, <https://www.webofscience.com/wos/author/record/678586>

СМОЛАРЖ Анджей, доцент факультета электроники Люблинского политехнического университета (Люблин, Польша), <https://www.scopus.com/authid/detail.uri?authorId=56249263000>, <https://www.webofscience.com/wos/author/record/1268523>

КЕЙЛАН Алимхан, доктор технических наук, профессор (Doctor of science (Japan)), главный научный сотрудник РГП «Института информационных и вычислительных технологий» КН МНВО РК (Алматы, Казахстан), <https://www.scopus.com/authid/detail.uri?authorId=8701101900>, <https://www.webofscience.com/wos/author/record/1436451>

ХАЙРОВА Нина, доктор технических наук, профессор, главный научный сотрудник РГП «Института информационных и вычислительных технологий» КН МНВО РК (Алматы, Казахстан), <https://www.scopus.com/authid/detail.uri?authorId=37461441200>, <https://www.webofscience.com/wos/author/record/1768515>

ОТМАН Мохамед, доктор философии, профессор компьютерных наук, Департамент коммуникационных технологий и сетей, Университет Путра Малайзия (Селангор, Малайзия), <https://www.scopus.com/authid/detail.uri?authorId=56036884700>, <https://www.webofscience.com/wos/author/record/747649>

НЫСАНБАЕВА Сауле Еркебулановна, доктор технических наук, доцент, старший научный сотрудник РГП «Института информационных и вычислительных технологий» КН МНВО РК (Алматы, Казахстан), <https://www.scopus.com/authid/detail.uri?authorId=55453992600>, <https://www.webofscience.com/wos/author/record/3802041>

УСАТОВА Ольга Александровна, PhD, ассоциированный профессор, Главный ученый секретарь «Института информационных и вычислительных технологий» КН МНВО РК (Алматы, Казахстан), <https://www.scopus.com/authid/detail.uri?authorId=57204581062>, <https://www.webofscience.com/wos/author/record/JCO-3058-2023>

КАПАЛОВА Нурсулу Алдажаровна, кандидат технических наук, заведующий лабораторией кибербезопасности РГП «Института информационных и вычислительных технологий» КН МНВО РК (Алматы, Казахстан), <https://www.scopus.com/authid/detail.uri?authorId=57191242124>

КОВАЛЕВ Александр Михайлович, доктор физико-математических наук, академик НАН Украины, Институт прикладной математики и механики (Донецк, Украина), <https://www.scopus.com/authid/detail.uri?authorId=7202799321>, <https://www.webofscience.com/wos/author/record/38481396>

МИХАЛЕВИЧ Александр Александрович, доктор технических наук, профессор, академик НАН Беларуси (Минск, Беларусь), <https://www.scopus.com/authid/detail.uri?authorId=7004159952>, <https://www.webofscience.com/wos/author/record/46249977>

ТИГИНЯНУ Ион Михайлович, доктор физико-математических наук, академик, президент Академии наук Молдовы, Технический университет Молдовы (Кишинев, Молдова), <https://www.scopus.com/authid/detail.uri?authorId=7006315935>, <https://www.webofscience.com/wos/author/record/524462>

Academic Scientific Journal of Computer Science

ISSN 2518-1726 (Online),

ISSN 1991-346X (Print)

Собственник: *ТОО «Центрально-азиатский академический научный центр» (г. Алматы).*

Свидетельство о постановке на переучет периодического печатного издания, информационного агентства и сетевого издания № **KZ77VPY00121154**. Дата выдачи **05.06.2025**

Тематическая направленность: *информационно-коммуникационные технологии.*

В настоящее время: *вошел в список журналов, рекомендованных КОКШВО МНВО РК по направлению «информационно-коммуникационные технологии».*

Периодичность: *4 раза в год.*

<http://www.physico-mathematical.kz/index.php/en/>

© ТОО «Центрально-азиатский академический научный центр», 2026

CONTENTS

Akhmetova A.Zh., Kantureyeva M.A., Serimbetov B.A., Zhasuzakova M.Zh., Mustafayeva N.T. Identification of communities and active users in social networks.....	11
Akzhigitov D., Mukhtarkhanov M., Zhetpisbayeva A., Kulakayeva A., Zhetpisbayev K. Analysis of the mechanical structure of a 12U CubeSat nanosatellite and assessment of its mechanical stability.....	29
Alibekov A., Nurimanov A., Mukhametkali A., Kassym D., Varol H.A. Qolda-Math: a small vision-language model for mathematical reasoning in Kazakh.....	50
Alisheva S.S., Ospanova A.O., Shaimerdenova G.S., Shaimerdenova L., Zhidebayeva A.N. Detecting Kazakh language offensive comments in online content using deep learning algorithms.....	68
Askar G., Mutaikhan A., Aitkozhasyeva M., Nauryzbaikyzy G., Makhataeva Z. Retrieval beats fine-tuning: adapting a Kazakh small language model for historical question answering.....	86
Beissenova G.I., Shynzhigit B.B., Beisebayeva A.Zh., Kaldarova B.S., Lesbayev A.U. Deep neural network-based speaker authentication for secure access control systems.....	106
Bekenuly B., Baimakhanova A., Baiturganova V., Bolatbekova D. Comparison of CNN architectures for automated ACL injury classification from knee MRI.....	123
Bekbolat A., Razak A., Alimbayeva Z., Ozhikenova A. Multi-task deep learning for knee MRI pathology classification.....	139
Issakhov A., Akylbekkyzy A., Abylkassymova A. Predicting the spread of airborne viral infections in enclosed spaces.....	158
Kisseleva O., Abdramanov K., Naumenko V., Mukanova Z., Maintser D. Evaluation of serverless cold start optimization techniques.....	177
Koshkarov Ye.Zh., Amirgaliyev B.Ye., Mulesa O. Physics-aware masked pretraining for cross-basin well log foundation models.....	197
Kurmetkan T., Turysbek Zh., Mukhitova A., Nauryzbaeva A. Research on methods for reducing knowledge conflicts in large language models in Kazakh question-and-answer systems.....	214
Makatov Ye.K., Abdul Razaque, Ismukanova A.N., Salykova A.K., Glock E.S. Machine learning-based model for phishing URL detection in online social networks.....	233
Mimenbayeva A.B., Naizagarayeva A.A., Ospanova T.T., Koikelova D.K., Kudabayeva R.N. Spatial and climate data analysis for machine learning-based crop yield prediction in Northern Kazakhstan.....	255
Mukhamediev R.I., Abdoldina F.N., Gertsen Y.A. Economic opportunities and constraints of artificial intelligence adoption in Kazakhstan's resource-oriented economy.....	274

Mutanov G.M., Karyukin V.I., Ussatova O.A., Begimbayeva Y.Y., Baispay G.B. The assessment of blockchain readiness for academic document verification.....	289
Mussiraliyeva Sh., Bolatbek M., Baisylbayeva K. Classification of cyber-extremist ideologies in the Kazakh language considering multi-annotator disagreement.....	306
Omarov B.S., Sultan D.R., Rakhymberdiev D.S., Abibullayeva A.A., Toktarova A.B. Deformable attention-guided Transformer for brain tumor segmentation.....	326
Oralbekova D., Bekarystankyzy A., Yerimbetova A., Berzhanova U. Development of a streaming Kazakh speech recognition system using end-to-end models.....	346
Serek A., Shoiynbek A., Zheludkov M., Kuanyshbay D., Meraliyev B. Comparative analysis of machine learning methods for detecting fraud in text messages.....	365
Smakhanova A.K., Uskenbayeva R.K., Kassymova A.B., Kalpeyeva Zh.B., Konyrbaev N.B. Experimental research using models and methods of the university digital ecosystem.....	384
Sultanali A.E., Kazbekova G.N. Machine learning-based detection of AI-generated content in students' written works.....	401
Tashenova Zh., Yergazin A., Nurlybaeva E., Abdugulova Zh., Amanzholova Sh. Experimental evaluation of a hybrid IDPS with adaptive learning for low-latency 5G networks using edge computing.....	418
Telman D., Yerimbetova A., Daiyrbayeva E. Leakage-resistant fake news detection for Kazakh: temporal evaluation of lexical, Transformer and hybrid models.....	441
Zhanegizov A., Mamyrbayev O., Abdigappar M. Application of YOLO12 for road defect detection in GPR images.....	457

МАЗМҰНЫ

Ахметова А.Ж., Кантуреева М.А., Серимбетов Б.А., Жасұзақова М.Ж., Мұстафаева Н.Т. Әлеуметтік желілердегі қауымдастықтар мен белсенді қолданушыларды анықтау.....	11
Ақжігітов Д.Ж., Мұхтарханов М.О., Жетписбаева А.Т., Кулакаева А.Е., Жетписбаев К.У. 12U CubeSat үлгісіндегі наноспутниктің механикалық құрылымын талдау және оның механикалық орнықтылығын бағалау.....	29
Алибеков А., Нуриманов А., Мұхаметқали А., Қасым Д., Варол Х.А. Qolda-Math: қазақ тіліндегі математикалық пайымдауға арналған кіші көру-тілдік модель.....	50
Алишева С.С., Оспанова А.О., Шаймерденова Г.С., Шаймерденова Л., Жидебаева А.Н. Онлайн-контенттегі қазақ тіліндегі бейәдеп пікірлерді терең оқыту алгоритмдері арқылы анықтау.....	68
Асқар Ғ., Мұтайхан А., Айтқожаева М., Наурызбайқызы Г., Махатаева Ж. Retrieval fine-tuning-тен басым: тарихи сұрақтарға жауап беру үшін қазақша шағын тілдік модельді бейімдеу.....	86
Бейсенова Г.И., Шынжігіт Б.Б., Бейсебаева А.Ж., Қалдарова Б.С., Лесбаев А.У. Терең нейрондық желіге негізделген қауіпсіз басқару жүйелерінде сөйлеушіні аутентификациялау.....	106
Бекенұлы Б., Баймаханова А., Байтурганова В., Болатбекова Д. Алдыңғы крест тәрізді байлам жарақатын анықтауға арналған CNN архитектураларын салыстыру.....	123
Бекболат А., Разак А., Алимбаева Ж., Ожикенова А. Tize MPT патологиясын жіктеуге арналған көп тапсырмалы терең оқыту.....	139
Исахов А.А., Ақылбекқызы А., Абылкасымова А. Шектеулі кеңістіктерде әуе тамшыларымен вирустық инфекцияның таралуын сандық модельдеу.....	158
Киселева О., Абдраманов Қ., Науменко В., Мұқанова Ж., Майнцер Д. Serverless жүйелеріндегі суық іске қосуды оңтайландыру әдістерін эксперименттік бағалау.....	177
Кошкарров Е.Ж., Амиргалиев Б.Е., Мүлеса О. Бассейнаралық ұңғымалық диаграммалардың іргелі модельдерін физикалық тұрғыдан негізделген масқаланған алдын ала оқыту.....	197
Құрметқан Т., Тұрысбек Ж., Мұхитова А., Наурызбаева А. Қазақ тіліндегі сұрақ-жауап жүйелерінде үлкен тілдік модельдердегі білім қайшылықтарын азайту әдістерін зерттеу.....	214
Мақатов Е.К., Abdul Razaque, Исмуканова А.Н., Салыкова А.К., Глюк Е.С. Онлайн әлеуметтік желілердегі фишингтік URL-дарды анықтауға арналған машиналық оқыту негізіндегі модель.....	233

Мименбаева А.Б., Найзағараева А.А., Оспанова Т.Т., Қойкелова Д.К., Қудабаева Р.Н. Солтүстік Қазақстандағы дақылдар өнімділігін машиналық оқыту арқылы болжауда кеңістіктік және климаттық деректерді талдау.....	255
Мухамедиев Р.И., Абдолдина Ф.Н., Герцен Е.А. Қазақстанның ресурстық бағытталған экономикасына жасанды интеллектті енгізудің экономикалық мүмкіндіктері мен шектеулері.....	274
Мутанов Г.М., Карюкин В.И., Усатова О.А., Бегимбаева Е.Е., Байспай Г.Б. Білім беру құжаттарының түпнұсқалығын тексеруге блокчейн технологиясының дайындығын бағалау.....	289
Мусиралиева Ш., Болатбек М., Байсылбаева К. Көпаннынотаторлы келіспеушілікті ескере отырып қазақ тіліндегі киберэкстремистік идеологияларды жіктеу.....	306
Омаров Б.С., Сұлтан Д.Р., Рахымбердиев Д.С., Абибуллаева А.А., Тоқтарова А.Б. Ми ісіктерін сегменттеуде деформацияланатын зейін механизміне негізделген мультимодальды трансформер.....	326
Оралбекова Д.О., Бекарыстанқызы А., Еримбетова А.С., Бержанова У.Г. Интегралды модельдерді қолдану арқылы қазақ тіліндегі ағындық сөйлеуді тану жүйесін әзірлеу.....	346
Серек А., Шойынбек А., Желудков М., Қуанышбай Д., Мералиев Б. Мәтіндік хабарламалардағы алаяқтықты анықтауға арналған машиналық оқыту әдістерінің салыстырмалы талдауы.....	365
Смаханова А.К., Ускенбаева Р.К., Касымова А.Б., Кальпеева Ж.Б., Қоңырбаев Н.Б. Университеттің цифрлық экожүйесінің модельдері мен әдістерін қолдана отырып жүргізілген эксперименттік зерттеу.....	384
Сұлтанәлі А.Е., Казбекова Г.Н. Білім алушылардың жазбаша жұмыстарындағы AI-контентті машиналық оқыту арқылы анықтау.....	401
Ташенова Ж.М., Ергазин Ә., Нұрлыбаева Ә.Н., Абдуғулова Ж.К., Аманжолова Ш.А. Edge Computing көмегімен төмен кідірісті 5G желілеріне арналған бейімделгіш оқытуы бар гибриді IDPS-ті эксперименттік бағалау.....	418
Тельман Д., Еримбетова А., Дайырбаева Ә. Қазақ тіліндегі жалған жаңалықтарды анықтау: лексикалық, трансформерлік және гибридті модельдерді уақыт бойынша бағалау.....	441
Жанегизов А.С., Мамырбаев Ә.Ж., Абдіғаппар М.Д. YOLO12 моделін георадар кескіндеріне қолдану.....	457

СОДЕРЖАНИЕ

Ахметова А.Ж., Кантуреева М.А., Серимбетов Б.А., Жасузакова М.Ж., Мустафаева Н.Т. Выявление сообществ и активных пользователей в социальных сетях.....	11
Акжигитов Д.Ж., Мухтарханов М.О., Жетписбаева А.Т., Кулакаева А.Е., Жетписбаев К.У. Анализ механической конструкции наноспутника модели 12U CubeSat и оценка его механической устойчивости.....	29
Алибеков А., Нуриманов А., Мухаметкали А., Касым Д., Варол Х.А. Qolda-Math: маленькая мультимодальная модель зрения и языка для математических рассуждений на казахском языке.....	50
Алишева С.С., Оспанова А.О., Шаймерденова Г.С., Шаймерденова Л., Жидебаева А.Н. Выявление оскорбительных комментариев на казахском языке в онлайн-контенте с помощью алгоритмов глубокого обучения.....	68
Аскар Г., Мутайхан А., Айткожаева М., Наурызбайкызы Г., Махатаева Ж. Retrieval эффективнее fine-tuning: адаптация малой языковой модели казахского языка для ответов на исторические вопросы.....	86
Бейсенова Г.И., Шынжигит Б.Б., Бейсебаева А.Ж., Калдарова Б.С., Лесбаев А.У. Аутентификация личности по голосу на основе глубокой нейронной сети для систем безопасного управления доступом.....	106
Бекенулы Б., Баймаханова А., Байтурганова В., Болатбекова Д. Сравнение архитектур сверточных нейронных сетей для автоматической классификации разрыва ПКС по МРТ коленного сустава.....	123
Бекболат А., Разак А., Алимбаева Ж., Ожикенова А. Многозадачное глубокое обучение для классификации патологий колена по МРТ.....	139
Исахов А.А., Акылбеккызы А., Абылкасымова А. Прогнозирование распространения вирусной инфекции воздушно-капельным путем в замкнутых помещениях.....	158
Киселева О., Абдраманов К., Науменко В., Муканова Ж., Майнцер Д. Экспериментальная оценка методов оптимизации холодного запуска в serverless-системах.....	177
Кошкарар Е.Ж., Амиргалиев Б.Е., Мулеса О. Физически-осознанное маскированное предобучение для межбассейновых фундаментальных моделей каротажных диаграмм.....	197
Курметкан Т., Турысбек Ж., Мухитова А., Наурызбаева А. Исследование методов снижения конфликтов знаний в больших языковых моделях казахских вопросно-ответных систем.....	214
Макатов Е.К., Abdul Razaque, Исмуканова А.Н., Салыкова А.К., Глок Е.С. Модель обнаружения фишинговых URL в онлайн-социальных сетях на основе машинного обучения.....	233

Мименбаева А.Б., Найзагараева А.А., Оспанова Т.Т., Койкелова Д.К., Кудабаяева Р.Н. Анализ климатических данных для прогнозирования урожайности пшеницы методами машинного обучения в Северном Казахстане.....	255
Мухамедиев Р.И., Абдолдина Ф.Н., Герцен Е.А. Экономические возможности и ограничения внедрения искусственного интеллекта в ресурсно-ориентированной экономике Казахстана.....	274
Мутанов Г.М., Карюкин В.И., Усатова О.А., Бегимбаева Е.Е., Байспай Г.Б. Оценка готовности технологии блокчейн к проверке подлинности документов об образовании.....	289
Мусиралиева Ш., Болатбек М., Байсылбаева К. Классификация киберэкстремистских идеологий на казахском языке с учетом разногласий нескольких аннотаторов.....	306
Омаров Б.С., Султан Д.Р., Рахымбердиев Д.С., Абибуллаева А.А., Токтарова А.Б. Мультимодальный трансформер на основе механизма деформируемого внимания для сегментации опухолей головного мозга.....	326
Оралбекова Д.О., Бекарыстанкызы А., Еримбетова А.С., Бержанова У.Г. Распознавание казахской речи с применением интегральных моделей.....	346
Серек А., Шойынбек А., Желудков М., Куанышбай Д., Мералиев Б. Сравнительный анализ методов машинного обучения для выявления мошенничества в текстовых сообщениях.....	365
Смаханова А.К., Ускенбаева Р.К., Касымова А.Б., Кальпеева Ж.Б., Конырбаев Н.Б. Экспериментальное исследование с использованием моделей и методов цифровой экосистемы университета.....	384
Султанали А.Е., Казбекова Г.Н. Обнаружение AI-контента в письменных работах обучающихся методами машинного обучения.....	401
Ташенова Ж.М., Ергазин А., Нурлыбаева Э.Н., Абдугулова Ж.К., Аманжолова Ш.А. Экспериментальная оценка гибридной системы обнаружения и предотвращения вторжений (IDPS) с адаптивным обучением для сетей 5G с низкой задержкой с использованием граничных вычислений.....	418
Тельман Д., Еримбетова А., Дайырбаева Э. Обнаружение фейковых новостей на казахском языке: временная оценка лексических, трансформерных и гибридных моделей.....	441
Жанегизов А.С., Мамырбаев О.Ж., Абдигаппар М.Д. Применение модели YOLOv12 для анализа данных георадара.....	457

ACADEMIC SCIENTIFIC JOURNAL OF COMPUTER SCIENCE
ISSN 1991-346X
Volume 3.
Number 359 (2026). 177–196

<https://doi.org/10.32014/2026.2518-1726.458>

IRSTI 31.15.37
UDC 621.546.06

© **Kisseleva O.***, **Abdramanov K.**, **Vitaliy N.**, **Mukanova Z.**,
Maintser D., 2026.

Turan University, Almaty, Kazakhstan.
E-mail: o.kisseleva@turan-edu.kz

EVALUATION OF SERVERLESS COLD START OPTIMIZATION TECHNIQUES

Abdramanov Kuanysh — Turan University, Almaty, Kazakhstan,
E-mail: abdramanovkuanysh@gmail.com, <https://orcid.org/0009-0002-1645-0543>;
Kisseleva Olga — PhD, Turan University, Almaty, Kazakhstan,
E-mail: o.kisseleva@turan-edu.kz, <https://orcid.org/0000-0001-8637-1416>;
Vitaliy Naumenko — PhD, Turan University, Almaty, Kazakhstan,
E-mail: v.naumenko@turan-edu.kz, <https://orcid.org/0009-0005-9393-6733>;
Zhanna Mukanova — PhD, Turan University, Almaty, Kazakhstan,
E-mail: zhanna.mukanova.83@mail.ru, <https://orcid.org/0000-0001-6506-9007>;
Dmitriy Maintser — Turan University, Almaty, Kazakhstan,
E-mail: d.maintser@turan-edu.kz, <https://orcid.org/0009-0006-7024-9700>.

Abstract. Cold start latency remains one of the main challenges in serverless computing, particularly for Function-as-a-Service (FaaS) applications operating under dynamic workloads. Runtime initialization, dependency loading, and resource allocation increase response latency and reduce system performance. Although numerous cold start mitigation techniques have been proposed, comprehensive experimental comparisons under identical workload conditions remain limited. This study analyzes contemporary cold start optimization techniques and experimentally evaluates the effectiveness of warm-up, caching, and concurrency strategies for improving serverless application performance under different workload intensities. A controlled serverless test environment was developed using Python, Flask, Redis, and Locust. Cold starts were simulated through artificial initialization delays, while Redis was used for result caching and a thread pool enabled concurrent request processing. Performance was evaluated under moderate (100 users), intensive (300 users), and prolonged high-load scenarios using response time, throughput, request-processing errors, and P95/P99 latency metrics. The results show that optimization effectiveness depends on workload intensity. Under moderate workloads, performance differences were

minor. Under intensive workloads, warm-up and concurrency reduced response latency and improved system stability. Caching reduced the error rate in the short-term intensive scenario, although it introduced substantial additional latency. During prolonged testing, the average response time of the caching configuration decreased, indicating a more workload-dependent effect of cache population. Combined optimization provided balanced performance but did not consistently outperform individual techniques because of coordination overhead. The findings provide practical guidance for selecting cold start optimization strategies according to workload characteristics and support the design of adaptive, high-performance serverless applications and cloud services.

Keywords: serverless computing, Function-as-a-Service, cold start, warm-up, caching, concurrency, performance optimization

For citations: Kisseleva O., Abdramanov K., Vitaliy N., Mukanova Z., Maintser D. Evaluation of serverless cold start optimization techniques. Academic Scientific Journal of Computer Science, 2026. — No.3. — P. 177–196. DOI: <https://doi.org/10.32014/2026.2518-1726.458>

© Киселева О.*, Абдраманов Қ., Науменко В., Мұқанова Ж.,
Майнцер Д., 2026.

«Тұран» университеті, Алматы, Қазақстан.

E-mail: o.kisseleva@turan-edu.kz

SERVERLESS ЖҮЙЕЛЕРІНДЕГІ СУЫҚ ІСКЕ ҚОСУДЫ ОҢТАЙЛАНДЫРУ ӘДІСТЕРІН ЭКСПЕРИМЕНТТІК БАҒАЛАУ

Абдраманов Қуаныш — «Тұран» университеті, Алматы, Қазақстан,
E-mail: abdramanovkuanys@gmail.com, <https://orcid.org/0009-0002-1645-0543>;

Киселева Ольга — PhD, «Тұран» университеті, Алматы, Қазақстан,
E-mail: o.kisseleva@turan-edu.kz, <https://orcid.org/0000-0001-8637-1416>;

Науменко Виталий — PhD, «Тұран» университеті, Алматы, Қазақстан,
E-mail: v.naumenko@turan-edu.kz, <https://orcid.org/0009-0005-9393-6733>;

Мұқанова Жанна — PhD, «Тұран» университеті, Алматы, Қазақстан,
E-mail: zhanna.mukanova.83@mail.ru, <https://orcid.org/0000-0001-6506-9007>;

Майнцер Дмитрий — «Тұран» университеті, Алматы, Қазақстан,
E-mail: d.maintser@turan-edu.kz, <https://orcid.org/0009-0006-7024-9700>.

Аннотация. Суық іске қосу кідірісі serverless-есептеулердің негізгі мәселелерінің бірі болып қала береді, әсіресе динамикалық жүктеме жағдайында жұмыс істейтін Function-as-a-Service (FaaS) қосымшалары үшін. Орындау ортасын инициализациялау, тәуелділіктерді жүктеу және есептеу ресурстарын бөлу жауап беру уақытын ұзартып, жүйе өнімділігін төмендетеді. Суық іске қосуды азайтудың көптеген әдістері ұсынылғанымен, бірдей жүктеме жағдайларында олардың кешенді эксперименттік салыстырылуы әлі де шектеулі. Зерттеудің мақсаты - суық іске қосуды оңтайландырудың

заманауи әдістерін талдау және әртүрлі жүктеме деңгейлерінде serverless-қосымшалардың өнімділігін арттыру үшін алдын ала қыздыру (warm-up), кэштеу және параллельді орындау (concurrency) стратегияларының тиімділігін эксперименттік тұрғыдан бағалау. Зерттеу үшін Python, Flask, Redis және Locust негізінде бақыланатын serverless сынақ ортасы әзірленді. Суық іске қосу жасанды инициализация кідірісі арқылы модельденді, Redis нәтижелерді кэштеу үшін пайдаланылды, ал сұраныстарды параллель өңдеу ағындар пулы арқылы жүзеге асырылды. Жүйенің өнімділігі орташа жүктеме (100 пайдаланушы), қарқынды жүктеме (300 пайдаланушы) және ұзақ уақытқа созылған жоғары жүктеме жағдайларында жауап беру уақыты, өткізу қабілеті, сұраныстарды өңдеу қателері және P95/P99 кідіріс көрсеткіштері бойынша бағаланды. Нәтижелер оңтайландыру тиімділігінің жүктеме қарқындылығына тәуелді екенін көрсетті. Орташа жүктеме кезінде өнімділік айырмашылықтары шамалы болды. Қарқынды жүктеме кезінде алдын ала қыздыру және параллельді орындау жауап беру кідірісін азайтып, жүйенің тұрақтылығын арттырды. Кэштеу қысқа мерзімді қарқынды сценарийде қателер санын азайтты, алайда қосымша кідіріс туғызды. Ұзақ мерзімді тестілеу барысында кэштеу конфигурациясының орташа жауап беру уақыты қысқарды, бұл оның тиімділігі жүктеме сипаты мен кэштің толу деңгейіне тәуелді екенін көрсетті. Біріктірілген оңтайландыру теңгерімді өнімділікті қамтамасыз етті, бірақ үйлестіруге байланысты үстеме шығындар салдарынан жекелеген тәсілдерден тұрақты түрде жоғары нәтиже көрсете алмады. Алынған нәтижелер суық іске қосуды оңтайландыру стратегияларын жүктеме ерекшеліктеріне сәйкес таңдауға және бейімделетін жоғары өнімді serverless-қосымшалар мен бұлттық сервистерді жобалауға практикалық негіз береді.

Түйін сөздер: serverless-есептеулер; Function-as-a-Service; суық іске қосу; алдын ала қыздыру; кэштеу; параллельді орындау; өнімділікті оңтайландыру

© Киселева О.*, Абдраманов К., Науменко В., Муканова Ж.,
Майнцер Д., 2026.

Университет «Туран», Алматы, Казахстан.

E-mail: o.kisseleva@turana.edu.kz

ЭКСПЕРИМЕНТАЛЬНАЯ ОЦЕНКА МЕТОДОВ ОПТИМИЗАЦИИ ХОЛОДНОГО ЗАПУСКА В SERVERLESS-СИСТЕМАХ

Абдраманов Куаныш — Университет «Туран», Алматы, Казахстан,
E-mail: abdramanovkuanys@gmail.com, <https://orcid.org/0009-0002-1645-0543>;

Киселева Ольга — PhD, Университет «Туран», Алматы, Казахстан,
E-mail: o.kisseleva@turana.edu.kz, <https://orcid.org/0000-0001-8637-1416>;

Науменко Виталий — PhD, Университет «Туран», Алматы, Казахстан,
E-mail: v.naumenko@turana.edu.kz, <https://orcid.org/0009-0005-9393-6733>;

Муканова Жанна — PhD, Университет «Туран», Алматы, Казахстан,
E-mail: zhanna.mukanova.83@mail.ru, <https://orcid.org/0000-0001-6506-9007>;

Майнцер Дмитрий — Университет «Туран», Алматы, Казахстан,
E-mail: d.maintser@turana-edu.kz, <https://orcid.org/0009-0006-7024-9700>.

Аннотация. Проблема холодного запуска остается одной из ключевых проблем serverless-вычислений, особенно для приложений Function-as-a-Service (FaaS), функционирующих в условиях динамической нагрузки. Инициализация среды выполнения, загрузка зависимостей и выделение вычислительных ресурсов увеличивают время отклика и снижают производительность системы. Несмотря на наличие большого числа предложенных методов оптимизации холодного запуска, комплексные экспериментальные сравнения различных подходов при одинаковых условиях нагрузки остаются ограниченными. Цель исследования заключается в анализе современных методов оптимизации холодного запуска и экспериментальной оценке эффективности стратегий предварительного прогрева (warm-up), кэширования и параллельной обработки запросов (concurrency) для повышения производительности serverless-приложений при различных уровнях нагрузки. Для проведения исследования была разработана контролируемая экспериментальная среда с использованием Python, Flask, Redis и Locust. Холодный запуск моделировался посредством искусственной задержки инициализации, Redis применялся для кэширования результатов, а параллельная обработка запросов реализовывалась с помощью пула потоков. Производительность системы оценивалась при умеренной нагрузке (100 пользователей), интенсивной нагрузке (300 пользователей) и продолжительной высокой нагрузке по времени отклика, пропускной способности, количеству ошибок обработки запросов и показателям задержки P95/P99. Результаты показывают, что эффективность оптимизации зависит от интенсивности нагрузки. При умеренной нагрузке различия в производительности были незначительными. При интенсивной нагрузке предварительный прогрев и параллельная обработка снижали задержку отклика и повышали стабильность системы. Кэширование уменьшало количество ошибок в краткосрочном интенсивном сценарии, однако приводило к дополнительной задержке. При продолжительном тестировании среднее время отклика конфигурации с кэшированием снижалось, что свидетельствует о зависимости ее эффективности от характера нагрузки и степени заполнения кэша. Комбинированная оптимизация обеспечивала сбалансированную производительность, но не всегда превосходила отдельные методы из-за дополнительных расходов на координацию. Полученные результаты могут быть использованы для выбора стратегий оптимизации холодного запуска с учетом характеристик нагрузки и при проектировании адаптивных высокопроизводительных serverless-приложений и облачных сервисов.

Ключевые слова: serverless-вычисления; Function-as-a-Service; холодный запуск; предварительный прогрев; кэширование; параллельная обработка; оптимизация производительности

Introduction. The cold start problem becomes especially pronounced under variable or high workloads, when the system must rapidly scale by launching new function instances. Many simultaneous cold starts during traffic spikes lead to increased latency, higher error rates, and degradation of user quality of service. Analyses of real-world production workloads from major cloud providers indicate that the efficiency of serverless platforms largely depends on their ability to minimize the overhead associated with scaling and repeated function deployment.

A wide range of methods for mitigating cold start latency has been proposed in scientific literature. These include container pre-warming and maintaining pools of ready-to-use instances, runtime snapshotting techniques, optimization of function size and dependencies, intelligent caching mechanisms, predictive scaling based on workload forecasting, and advanced resource scheduling algorithms. The FaaSLight approach demonstrates the effectiveness of application-level code loading optimization, whereas CodeCrunch highlights the efficiency of function compression and cost-aware warm-up placement (Shiekhani et al., 2025; Roy et al., 2024). Attention has also been given to hybrid scheduling strategies that combine workload prediction with adaptive resource management.

However, a considerable portion of existing studies focus either on the theoretical description of architectural solutions or on the evaluation of individual optimization techniques in isolated environments. At the same time, there remains a limited number of studies providing direct experimental comparisons of multiple practical optimization strategies under identical high-load conditions using unified performance metrics. This creates a research gap related to the need for a comprehensive evaluation of the applicability of widely used optimization approaches in real-world serverless deployment scenarios.

The objective of this study is to provide a comprehensive analysis of cold start optimization methods in serverless computing and to experimentally evaluate the impact of warm-up, caching, and concurrency strategies on system performance under different workload intensities. To achieve this objective, the study reviews modern approaches for reducing cold start latency, develops an experimental testbed simulating a serverless environment, and conducts load testing with analysis of response time, throughput, stability, and request error metrics. The results obtained are intended for practical application in the design and optimization of high-load cloud services.

The novelty of this study lies in the unified experimental evaluation of three commonly used cold start mitigation strategies — warm-up, caching, and concurrency — under identical experimental conditions and workload generation parameters. Unlike studies focusing primarily on individual optimization mechanisms, the proposed evaluation considers their behavior across moderate, intensive, and prolonged high-load scenarios using a common set of performance indicators, including response latency, P95/P99 latency, throughput, and request-processing error rate. This enables a direct comparison of the practical effects and workload-dependent limitations of the considered strategies within the same experimental environment.

Literary review. The cold start latency problem in Function-as-a-Service (FaaS) environments remains one of the most actively studied topics in the field of serverless computing, as this factor significantly affects request latency, application scalability, and overall quality of service. According to contemporary research, cold start mitigation approaches can be classified according to their level of application into platform-level methods, application-level methods, and intelligent resource management strategies. Such a classification enables a systematic evaluation of the advantages and limitations of existing solutions.

Platform-Level Optimization Methods. At the infrastructure level, considerable attention has been devoted to reducing runtime initialization overhead. One of the most widely adopted approaches is function instance pre-warming, in which a subset of containers is maintained in an active state before incoming requests arrive. This mechanism eliminates the need for a full runtime initialization cycle and significantly reduces first-response latency. A more advanced extension of this concept is represented by runtime snapshot mechanisms, such as AWS Lambda SnapStart, which enable restoration of pre-initialized execution environments within milliseconds (Liu et al., 2023).

Another important direction involves the use of lightweight virtualization technologies. The adoption of microVMs, minimized containers, and specialized runtime environments reduces the overhead associated with creating isolated execution environments compared to traditional virtual machines. According to existing surveys, such solutions can reduce cold start latency from approximately 700 ms to 300 ms, corresponding to a reduction of more than 57% (Chaudhary et al., 2024).

Attention has been given to the CodeCrunch framework, which introduces a mechanism for compressing inactive but still “warm” functions in memory. This approach enables a larger number of warm instances to be stored within the same memory capacity, thereby reducing the probability of cold starts. As illustrated in Figure 1, the proposed compression mechanism improves warm-start availability while reducing the number of cold start events. The authors report (Roy et al., 2024) a reduction in average service time from 8.15 to 6.75 seconds, along with an approximately 48% decrease in the number of cold starts. In addition, the study investigates function placement across heterogeneous x86 and ARM architectures while considering the trade-off between execution performance and the cost of maintaining active instances.

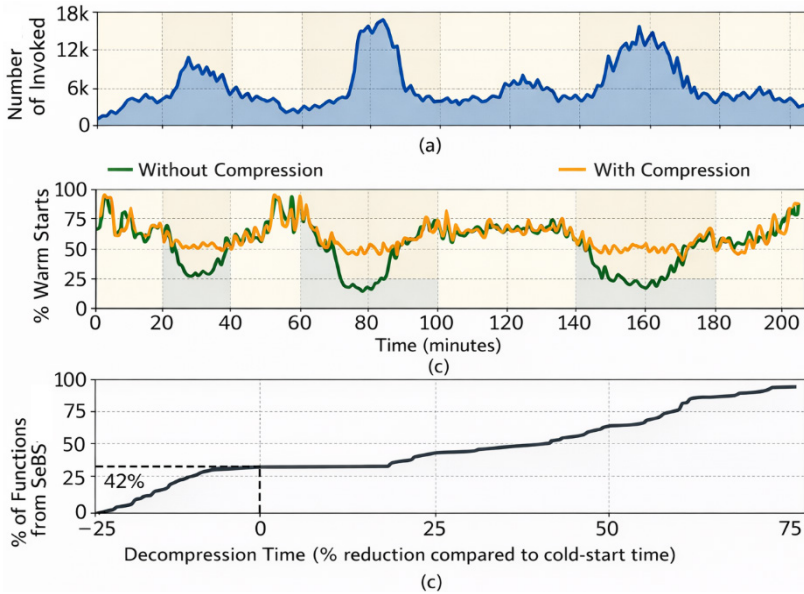


Figure 1 – Impact of In-Memory CodeCrunch Compression on Warm Starts

Application-Level Optimization Methods. Unlike platform-level solutions, application-level optimization methods are directly accessible to developers and do not require modifications to the cloud provider's infrastructure. One of the fundamental approaches is reducing the size of the function and its dependencies. Since the loading of libraries and user code constitutes a significant portion of cold start latency, minimizing the deployment package can substantially decrease startup time.

The FaaSLight study proposes an application-level static analysis approach that separates code into mandatory and optional components. During function initialization, only the critical code required for execution is loaded, while additional modules are imported on demand. According to the authors' results, the loading phase latency was reduced by up to 78.95%, while the overall response time decreased by an average of 19.21% (Shiekhani et al., 2025). This approach is particularly effective for functions with large dependency sets and complex application structures.

Another important optimization strategy is function fusion, in which multiple sequentially invoked functions are merged into a single execution module. This reduces the number of network transitions, repeated authentication procedures, and individual cold start events. Although the resulting function becomes larger in size, the total latency of the user workflow can decrease due to the reduction of distributed execution overhead (Liu et al., 2023).

Intelligent Resource Management and Scheduling. A separate category of approaches includes workload prediction and adaptive resource allocation

methods. Their primary objective is not to directly accelerate cold starts, but rather to reduce their frequency through proactive scaling mechanisms. The Hybrid Model study proposes a two-level scheduling architecture that combines global workload distribution across cluster nodes with local reuse of already warmed function instances. In Figure 2, the scheduling mechanism integrates predictive workload analysis with buffer-aware resource allocation to improve container reuse efficiency. To forecast incoming requests, the authors employ time-series analysis and machine learning techniques, including ARIMA and Random Forest models (Sharma, 2024).

According to the reported results, the application of the hybrid scheduling approach reduced SLA violations by up to 34%, thereby improving system stability under variable workloads (Sharma, 2024). However, the effectiveness of such systems strongly depends on prediction accuracy and the cost of maintaining reserved resources. Prediction errors may lead either to excessive resource consumption or to increased latency caused by an insufficient number of pre-initialized instances.

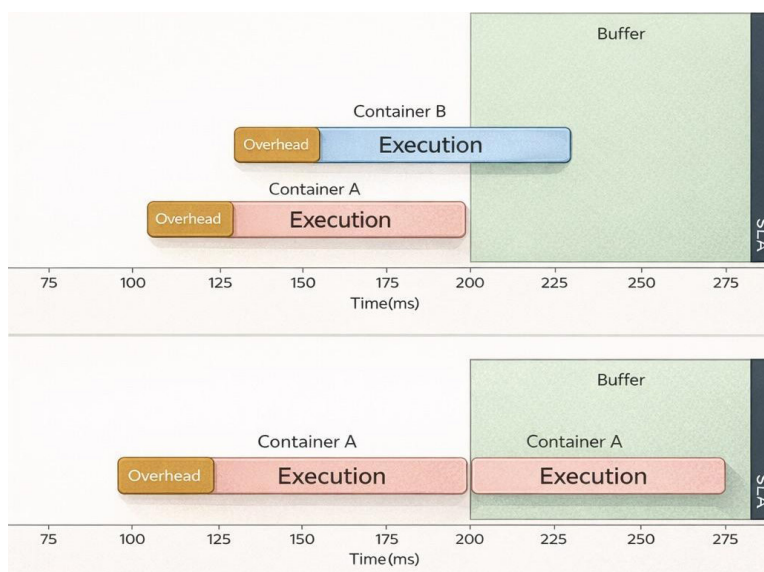


Figure 2 – Illustration of the Buffer-Aware Scheduling Algorithm

Production Workload Studies. Empirical studies of industrial workloads have made a significant contribution to understanding the behavior of serverless platforms. The *Serverless in the Wild* study demonstrates that real-world serverless systems are characterized by high request variability, short bursts of activity, and substantial heterogeneity in function execution times (Shahrad et al., 2020). These characteristics complicate the application of universal optimization mechanisms and highlight the need for adaptive multi-level resource management strategies.

Table 1 – Comparative analysis of optimization methods.

Method	The basic idea	Effectiveness	Overhead and limitations	Application level
FaaSLight	Separation of the code into mandatory and optional for deferred loading.	Average: 19.21%, up to 42.05%	Static code analysis is required; it may not be effective for languages with high dynamics.	Application
CodeCrunch Compression	Compression of the file system of "warm" functions in RAM.	On average, there are 48% fewer cold starts. The average service time is 6.75 seconds with compression versus 8.15 seconds without compression.	A decompression delay is added, which may exceed the gains from a warm start.	Platform
Heterogeneity of CodeCrunch	Choose between x86 and ARM architectures to balance keep-alive performance and cost.	The average service time is 7.87 seconds x86 and 8.4 seconds ARM.	It's a difficult planning task; not all functions benefit equally from a single architecture.	Platform
Hybrid Model Planning	Predictive load balancing based on SLA slack to maximize container reuse.	Reduction of SLA violations to 34%.	May increase the median delay due to queues; complexity and inaccuracy of forecasting.	Platform (Planner)
Lightweight execution environments	Minimizing the size and startup time of the runtime environment.	The use of lightweight runtimes reduced the cold start time to 57.14% (from 700 ms to 300 ms)	Requires deep changes in the platform; high complexity of development and support.	Platform

Experimental Methodology. To evaluate the practical effectiveness of widely used cold start mitigation techniques, an experimental study was conducted in a controlled software environment reproducing the core principles of a serverless architecture. The overall structure of the simulated environment is presented in Figure 3. The choice of a laboratory-based approach was motivated by the fact that commercial FaaS platforms typically abstract internal mechanisms related to container management, scaling policies, and resource allocation, making it difficult to accurately measure the impact of individual factors on system performance. The use of a local testbed ensured reproducibility of workload scenarios, controllability of function execution parameters, and comparability of results across different system configurations. Similar approaches are widely adopted in serverless performance studies where isolated evaluation of function behavior under varying workload intensities and runtime conditions are required.

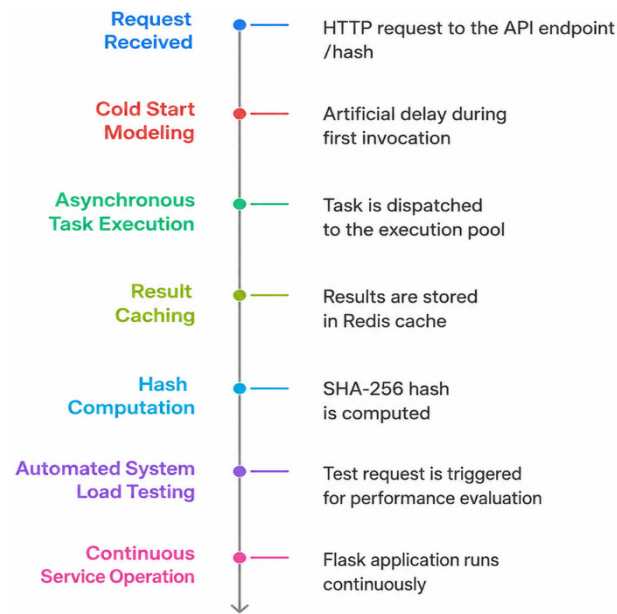


Figure 3 – Components and Operational Stages of the Simulated Serverless Architecture

The experimental environment was based on a server-side application implemented in Python. This technology stack was selected due to its suitability for rapid prototyping of distributed systems, extensive support for networking and multithreading libraries, and flexibility in simulating the lifecycle of serverless functions. The architecture of the testbed emulated the Function-as-a-Service model, where each incoming HTTP request triggered the execution of an independent processing function. To approximate the behavior of real-world serverless platforms, the system incorporated a conditional cold start mechanism. When no previously active function instance was available, the execution was delayed by an additional artificial latency of 2 seconds, simulating runtime creation, dependency loading, and runtime initialization. The value of 2 seconds was kept constant across all experimental configurations to ensure identical cold start conditions and to make the comparison between optimization strategies reproducible. This artificial delay represents a controlled experimental parameter rather than a universal cold start latency of a specific cloud platform.

Each experimental scenario was executed in five independent runs. The reported values represent the arithmetic means of the measurements obtained across the five repeated runs.

To analyze the impact of pre-warming, a separate system configuration was implemented in which the platform periodically initiated background function invocations to keep function instances in an active state. This mechanism reduced the probability of functions transitioning into an inactive state between user

requests and thereby decreased the number of cold starts. Such strategies are widely discussed in literature as one of the most direct approaches for reducing startup latency, although they introduce additional resources overhead associated with maintaining ready-to-use instances.

To evaluate the effect of result reuse, the experimental environment incorporated an external Redis cache. Upon receiving a request, the system first checked whether a previously generated response was already stored in the cache and, if a matching key was found, returned the cached result without re-executing the computational function. Redis was selected due to its widespread adoption of high-load web systems and its low in-memory data access latency. In the context of serverless applications, caching is considered an effective mechanism for reducing computational overhead, minimizing repeated function invocations, and stabilizing response times for recurring requests (Chaudhary et al., 2024; Jin et al., 2025)

Additionally, the study examined the influence of parallel request processing. For this purpose, the application employed a thread pool allowing multiple incoming requests to be processed concurrently without sequential blocking at the level of a single function. This approach models concurrency strategies commonly used in modern cloud environments to improve throughput and reduce queue waiting time under intensive workloads. Existing studies indicate that the impact of parallelism becomes particularly significant under burst scenarios and short-term spikes in user activity (Shahrad et al., 2020).

Load testing was performed using the Locust framework, which is designed for simulating large-scale user interaction with HTTP services. The selection of this tool was motivated by its flexibility in configuring request intensity, the number of virtual users, and the duration of testing scenarios. The experiments reproduced moderate-load and high-load conditions, as well as a prolonged stress-testing scenario with a continuous request stream. This testing methodology enables observation not only of the system's immediate behavior, but also of its long-term stability, which is especially important for serverless applications characterized by periodic workload spikes. Similar load-generation techniques are widely applied in studies of distributed cloud system scalability.

The workload was generated using a fixed number of concurrent virtual users. Two short-duration load scenarios were considered, corresponding to 100 and 300 concurrent users. In addition, a prolonged high-load scenario with 300 concurrent users was performed for 10 minutes. The same workload parameters were applied to each configuration to maintain comparable experimental conditions.

The comparative analysis was conducted between the baseline system configuration and variants with enabled warm-up, caching, concurrency, and combined optimization mechanisms. The workload dynamics and testing scenarios used during the experiments are illustrated in Figure 4. To evaluate effectiveness, the experiments recorded average response time, throughput, request-processing

errors, and high-percentile latency metrics, including P95 and P99. Since the request workload was controlled by Locust through the configured number of concurrent users, throughput was primarily used to characterize the achieved request-processing intensity, while response time, error rate, and high-percentile latency were used as the main indicators for comparing the configurations. The use of these metrics corresponds to common practices in cloud service performance evaluation, as it enables analysis not only of average values but also of service stability under peak workloads.

The selected methodology made it possible to investigate the influence of the most accessible and practically applicable cold start optimization methods in a reproducible environment, compare their behavior under different workload conditions, and establish an empirical basis for subsequent analysis of the experimental results.

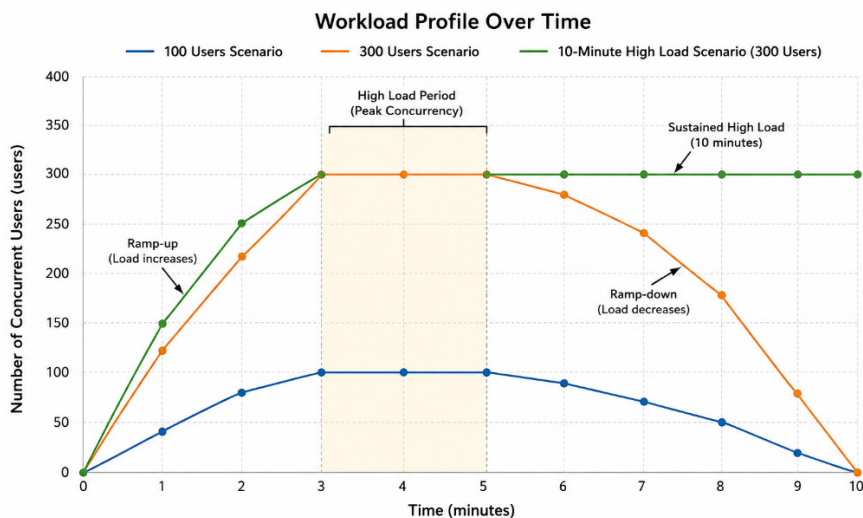


Figure 4 – Workload profile over time

Results. The load-testing experiments evaluated the behavior of the simulated serverless system under different optimization strategies and workload intensities. The comparison included the baseline configuration and configurations employing warm-up, caching, concurrency, and combined optimization mechanisms.

The obtained performance metrics are summarized in Figure 5.

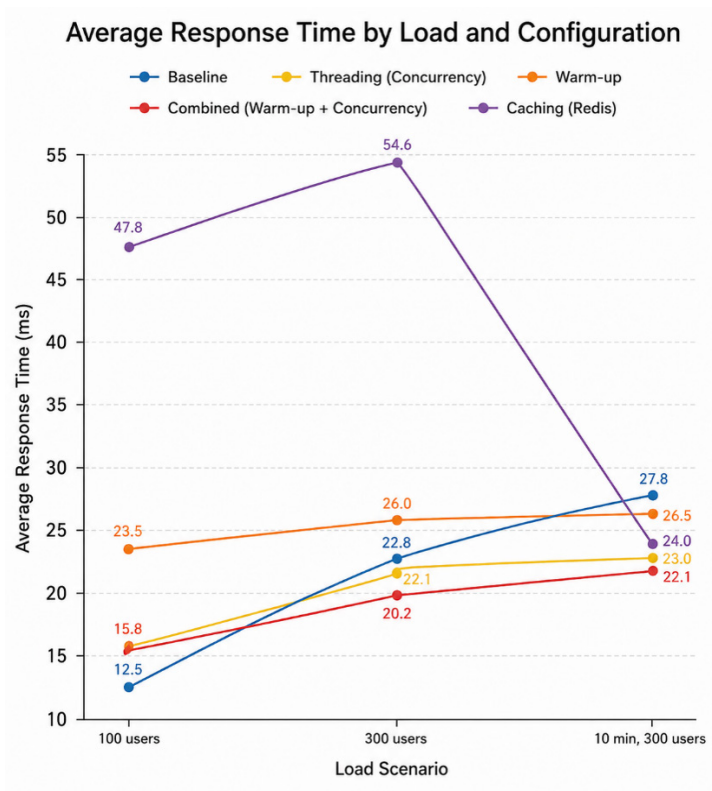


Figure 5 – Average Response Time Under Different Workload Conditions

Under the moderate workload scenario corresponding to 100 concurrent users, all configurations demonstrated stable operation. The baseline configuration achieved an average response time of 12.40 ms without request-processing errors and maintained a throughput of approximately 65 requests per second. The concurrency and warm-up configurations produced average response times of 15.57 ms and 15.49 ms, respectively. The combined optimization mode reached an average response time of 23.32 ms, while the caching configuration recorded the highest average response time of 47.77 ms.

After increasing the workload to 300 concurrent users, greater differences between configurations were observed. The baseline configuration achieved an average response time of 22.85 ms and recorded 14,309 request-processing errors, corresponding to an error rate of 25.43%. The warm-up configuration demonstrated the lowest average response time of 20.11 ms, followed by the concurrency configuration with 21.88 ms. The combined optimization mode reached 25.78 ms, whereas the caching configuration recorded the highest average response time of 54.56 ms. The caching configuration registered 7,901 request-processing errors, corresponding to an error rate of 19.21%. However, it also processed fewer requests

than the baseline configuration, with 41,127 completed requests compared with 56,266 for the baseline. Therefore, the lower absolute number of errors should be interpreted together with the lower number of processed requests.

The lower number of processed requests in the caching configuration can be associated with its substantially higher response latency under the 300-user workload. Since Locust generated the workload using concurrent virtual users, longer request processing reduces the number of request iterations that can be completed during the same observation period. Therefore, the difference in the number of processed requests should not be interpreted as a reduction in the offered workload, but as a difference in the number of completed request iterations under the same concurrency level.

The latency dynamics observed during prolonged high-load execution are presented in Figure 6.

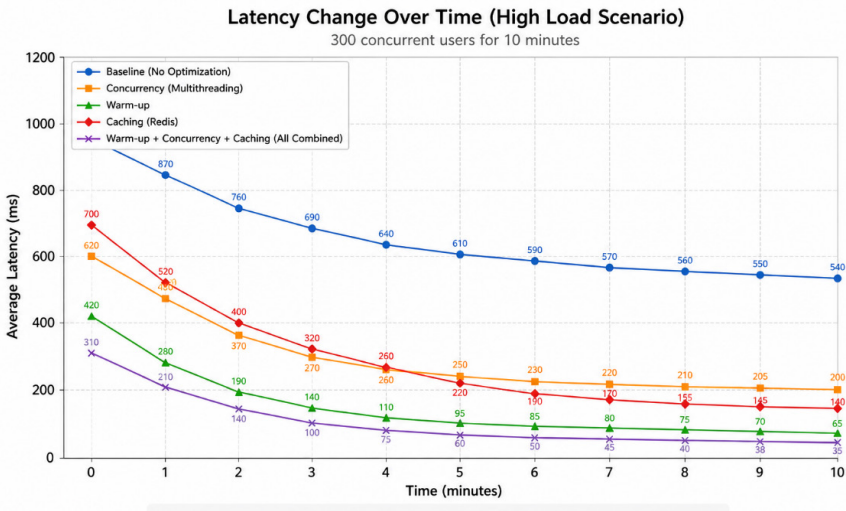


Figure 6 – Latency Change Over Time

The distribution of request-processing errors across the evaluated configurations is illustrated in Figure 7.

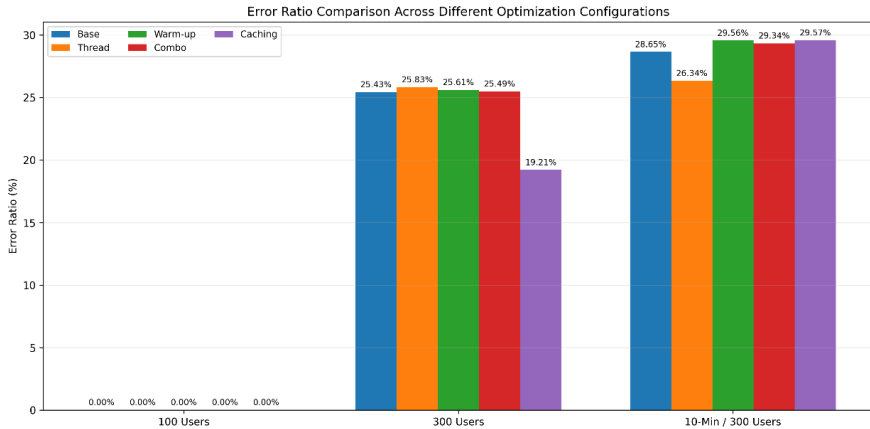


Figure 7 – Error Ratio Comparison Across Different Optimization Configurations

The prolonged stress-testing scenario demonstrated further differences between optimization strategies. The baseline configuration produced an average response time of 27.85 ms. The warm-up and concurrency configurations achieved lower average response times of 21.58 ms and 22.53 ms, respectively. The caching configuration demonstrated an average response time of 24.15 ms, while the combined optimization mode reached 26.27 ms.

The throughput comparison for all evaluated configurations is presented in Figure 8.

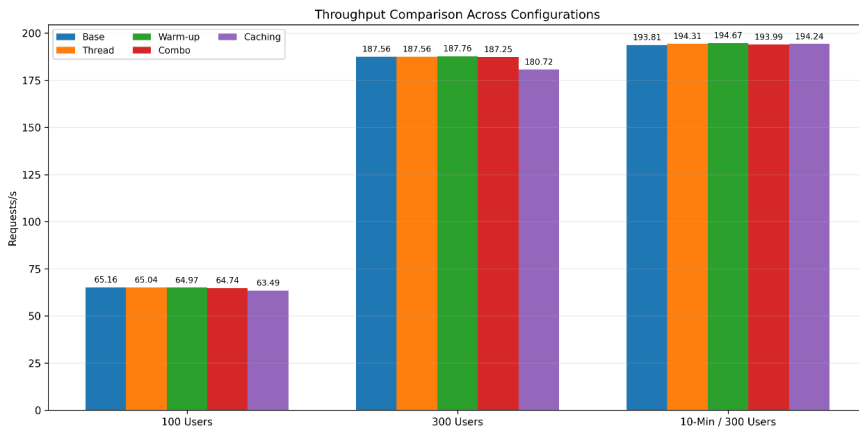


Figure 8 – Throughput Comparison Across Configurations

Table 2 – Load Testing Results

Scenario	Number of Requests	Average Response Time (ms)	p95 (ms)	p99 (ms)	Requests/s
base_100	21379	12.40	22	76	65.16
thread_100	21504	15.57	29	76	65.04
warm_up_100	19886	15.49	28	79	64.97
combo_100	20567	23.32	33	94	64.74
caching_100	15360	47.77	39	1200	63.49
base_300	56266	22.85	72	130	187.56
thread_300	56274	21.88	74	170	187.56
warm_up_300	56331	20.11	61	100	187.76
combo_300	56174	25.78	80	130	187.25
caching_300	41127	54.56	150	830	180.72
10min_base_300	116284	27.85	80	210	193.81
10min_thread_300	109556	22.53	73	120	194.31
10min_warm_up_300	116809	21.58	69	110	194.67
10min_combo_300	116398	26.27	81	130	193.99
10min_caching_300	116552	24.15	78	120	194.24

The combined optimization mode produced stable performance across all workload scenarios but did not achieve the lowest response time in every experiment. Figure 9 summarizes the P95 and P99 latency values measured under all testing scenarios.

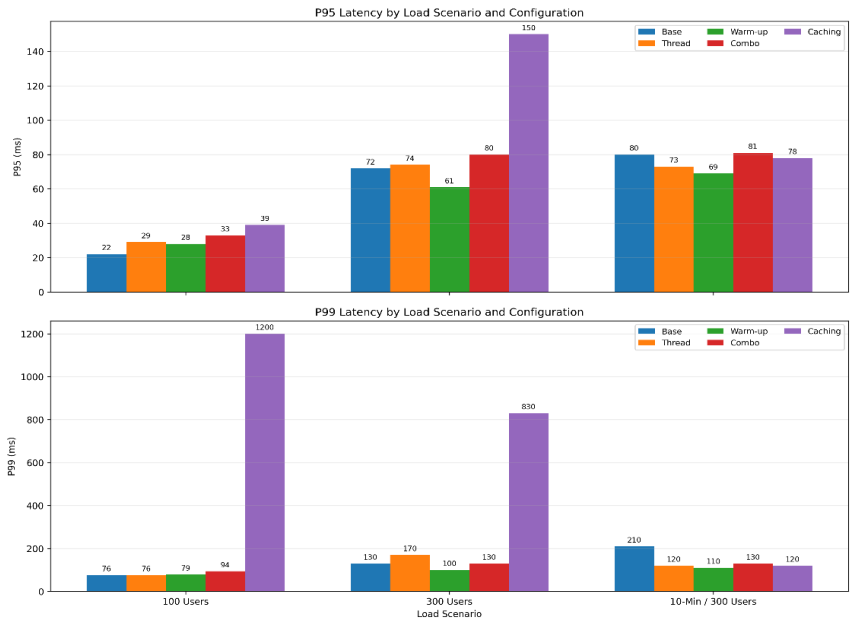


Figure 9 – P95 and P99 Latency by Load Scenario and Configuration

Discussions. The experimental results demonstrate that the effectiveness of cold start optimization strategies strongly depends on workload intensity and system operating conditions. Under moderate workloads, most evaluated configurations provided relatively stable performance, indicating that the baseline implementation offered sufficient computational resources without requiring additional optimization mechanisms. Consequently, the potential benefits of cold start mitigation remained limited when system resources were not saturated.

A different trend emerged under intensive workloads. The warm-up strategy consistently produced the lowest response times and improved system stability by reducing the number of cold start events during rapid scaling. These findings support previous studies emphasizing that maintaining pre-initialized function instances is one of the most effective approaches for minimizing startup overhead in Function-as-a-Service environments (Liu et al., 2023). Similarly, the concurrency strategy improved response time by reducing queue waiting time and enabling more efficient parallel request processing. Although its performance gains were slightly lower than those achieved with warm-up, concurrency demonstrated stable behavior under sustained workloads, which is consistent with observations reported by Shahrade et al. (2020).

The caching strategy exhibited a more complex performance profile. Under the short-term 300-user workload, caching reduced the absolute number of request-processing errors from 14,309 to 7,901 compared with the baseline configuration. However, the number of processed requests was also lower, and therefore the error rate provides a more appropriate basis for comparison. The caching configuration achieved an error rate of 19.21%, compared with 25.43% for the baseline. At the same time, its average response time increased to 54.56 ms. During prolonged testing, the average response time of the caching configuration decreased to 24.15 ms, indicating improved latency after the cache had been populated. However, its error rate did not demonstrate a comparable improvement. These results indicate that the benefits of caching depend on workload characteristics and cache utilization, while the additional cache-access overhead can negatively affect short-term response latency.

Another important observation concerns the combined optimization strategy. Although integrating multiple optimization mechanisms produced balanced performance across all scenarios, it did not consistently outperform the best individual approaches. The combined configuration did not consistently provide lower latency than the individual optimization strategies. In particular, the warm-up configuration demonstrated lower average response times under the intensive and prolonged workloads. This suggests that applying several optimization mechanisms simultaneously does not necessarily result in cumulative performance improvements. This finding indicates that combining optimization techniques does not automatically result in cumulative improvements and highlights the importance of selecting optimization mechanisms according to application characteristics rather than maximizing their number.

Overall, the obtained results indicate that the effectiveness of cold start mitigation strategies depends on workload characteristics and system operating conditions. In the experiments conducted, warm-up demonstrated the lowest response latency under intensive and prolonged workloads, while concurrency also provided lower latency than the baseline configuration. Caching showed benefits after cache population but introduced substantial latency overhead during the short-term intensive scenario. In the conducted experiments, warm-up provided the lowest response latency under intensive and prolonged workloads. Concurrency also reduced response latency relative to the baseline configuration, while caching demonstrated greater benefits after cache population but introduced substantial latency overhead in the short-term intensive scenario.

Conclusion. The conducted study was aimed at providing a comprehensive analysis of cold start optimization methods in serverless computing and an experimental evaluation of the practical effectiveness of warm-up, caching, and concurrency strategies under different workload conditions. The relevance of this problem is determined by the fact that cold start remains one of the key limitations of the Function-as-a-Service model, directly affecting request latency, service stability, and compliance with SLA requirements (Liu et al., 2023; Shahrads et al., 2020).

The analysis of the scientific literature demonstrated that existing cold start mitigation approaches can be categorized into platform-level, application-level, and orchestration-based approaches. Platform-level solutions, including lightweight runtime environments, snapshot mechanisms, and intelligent memory management, provide the most significant performance improvements, but require deep control over the provider's infrastructure. Application-level approaches, such as dependency reduction and function structure optimization, are easier to implement from the developer's perspective, although their effectiveness depends on the characteristics of a particular application. Predictive scaling and adaptive scheduling methods reduce the frequency of cold starts, but their efficiency is highly sensitive to workload prediction accuracy (Shiekhani et al., 2025; Roy et al., 2024).

The experimental results confirmed that the impact of optimization mechanisms strongly depends on the operational mode of the system. Under moderate workloads, the baseline configuration demonstrated acceptable performance, while the benefits of additional optimization mechanisms remained limited. However, with an increasing number of concurrent requests, the most significant positive effect was achieved by warm-up and concurrency strategies, which reduced average response time and improved request-processing stability. These findings confirm the importance of minimizing startup overhead and reducing waiting queues under intensive scaling conditions.

The caching strategy demonstrated a more complex effectiveness profile. In the short-term intensive scenario, caching resulted in the highest average response time of 54.56 ms, although the corresponding error rate was lower than that of

the baseline configuration. During prolonged testing, the average response time of the caching configuration decreased to 24.15 ms, which indicates that cache population can improve the response performance over time. However, the results do not demonstrate a consistent reduction in error rate during prolonged operation. Therefore, caching should be considered as a workload-dependent optimization mechanism whose effectiveness depends on the proportion of repeated requests and the overhead of accessing the external cache.

It should also be noted that the combined use of multiple optimization methods does not always lead to proportional performance improvements. In several scenarios, the additional coordination overhead between mechanisms partially compensated for the expected gains. Consequently, the effectiveness of comprehensive optimization depends not on the number of implemented techniques, but on how well they are aligned with the application architecture and workload profile.

The practical significance of this research lies in the fact that the obtained results may assist developers and cloud system architects in selecting appropriate optimization strategies for serverless applications. For services characterized by unpredictable workload spikes, warm-up and adaptive scaling mechanisms appear to be the most suitable approaches. For systems with a high degree of request repetition, caching provides substantial benefits. In scenarios involving intensive parallel processing, concurrency-oriented approaches become justified. The selection of an optimization strategy should therefore consider the workload characteristics, response-time requirements, and architectural properties of the target application.

Future research directions include extending the experimental environment to real cloud platforms, comparing different FaaS providers, applying machine learning techniques for dynamic optimization strategy selection, and analyzing the economic efficiency of cold start mitigation while considering the cost of reserved resources. Such developments would enable the transition from isolated engineering solutions toward intelligent self-adaptive performance management systems for serverless computing.

References

- Birajdar P.A., Harsha V., Satpathy A., Addya S.K. (2025) Adaptive workload management for enhanced function performance in serverless computing. In Proceedings of the 26th International Conference on Distributed Computing and Networking (ICDCN '25). — P. 276–277. DOI: <https://doi.org/10.1145/3700838.3703657> (in Eng.).
- Baresi L., Quattrocchi G., Ticongolo I.G. (2023) Dependency-aware resource allocation for serverless functions at the edge. In Service-Oriented Computing: 21st International Conference, ICSOC 2023. — P. 347–362. Springer. DOI: https://doi.org/10.1007/978-3-031-48421-6_24 (in Eng.).
- Chaudhary S., Somwanshi D.K., Rajawat V.S., Sharma M., Verma P. (2024) Cold start latency optimization in serverless computing: A comprehensive review of methods and emerging solutions. In Proceedings of the 6th International Conference on Information Management and Machine Intelligence (ICIMMI 2024). — P. 1–8. DOI: <https://doi.org/10.1145/3745812.3745825> (in Eng.).

Fuerst A., Sharma P. (2021) FaaS-Cache: Keeping serverless computing alive with Greedy-Dual caching. In Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems. — P. 386–400. DOI: <https://doi.org/10.1145/3445814.3446757> (in Eng.).

Hu Q., Li H., Nikougoftar E. (2025) Mitigating cold start problem in serverless computing using predictive pre-warming with machine learning. Computing, 107, Article 27. DOI: <https://doi.org/10.1007/s00607-024-01382-y> (in Eng.).

Jin L., Cai Z., Wang H., Zhang Z., Ma R., Guan H., Liu Y., Buyya R. (2025) Ephemera: Accelerating I/O-intensive serverless workloads using a dynamically provisioned in-memory file system. ACM Transactions on Architecture and Code Optimization, 22(3), Article 102. — P. 1–24. DOI: <https://doi.org/10.1145/3747846> (in Eng.).

Liu X., Wen J., Chen Z., Li D., Chen J., Liu Y., Wang H., Jin X. (2023) FaaS-Light: Application-level cold-start latency optimization for function-as-a-service in serverless computing. arXiv. DOI: <https://doi.org/10.48550/arXiv.2207.08175> (in Eng.).

Mahmoudi N., Khazaei H. (2022) Performance modeling of serverless platforms based on metrics. arXiv. DOI: <https://doi.org/10.48550/arXiv.2202.11247> (in Eng.).

Roh S., Jeong H., Boo H., Huh E. (2024) An efficient serverless-VM switching mechanism for cloud cost optimization. In Proceedings of the 9th International Conference on Intelligent Information Technology (ICIIT '24). — P. 482–486. DOI: <https://doi.org/10.1145/3654522.3654594> (in Eng.).

Roy R.B., Patel T., Garg R., Tiwari, D. (2024) CodeCrunch: Improving serverless performance via function compression and cost-aware warmup location optimization. In Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '24). — P. 85–101. DOI: <https://doi.org/10.1145/3617232.3624866> (in Eng.).

Shahrad M., Fonseca R., Goiri Í., Chaudhry G., Batur P., Cooke J., Laureano E., Tresness C., Russinovich M., Bianchini R. (2020) Serverless in the wild: Characterizing and optimizing the serverless workload at a large cloud provider. In Proceedings of the USENIX Annual Technical Conference (ATC 2020). — P. 205–218. DOI: <https://doi.org/10.48550/arXiv.2210.10331> (in Eng.).

Sharma A. (2024) Performance optimization techniques for serverless computing platforms. International Journal of Computer Engineering and Technology, 15(4). — P. 802–813. DOI: <https://doi.org/10.5281/zenodo.13464891> (in Eng.).

Shiekhani L., Wang H., Shi W., Liu J., Qiu Y., Gu C., Ding W. (2025) The hybrid model: Prediction-based scheduling and efficient resource management in serverless environments. Applied Sciences, 15(14), Article 7632. DOI: <https://doi.org/10.3390/app15147632> (in Eng.).

Silva P., Fireman D., Pereira T.E. (2020) Prebaking functions to warm the serverless cold start. In Proceedings of the 21st International Middleware Conference. — P. 1–13. DOI: <https://doi.org/10.1145/3423211.3425682> (in Eng.).

Yao X., Chen N., Yuan X., Ou P. (2023) Performance optimization of serverless edge computing function offloading based on deep reinforcement learning, 139. — P. 74–86. DOI: <https://doi.org/10.1016/j.future.2022.09.009> (in Eng.).

Publication Ethics and Publication Malpractice in the journals of the Central Asian Academic Research Center LLP

For information on Ethics in publishing and Ethical guidelines for journal publication see <http://www.elsevier.com/publishingethics> and <http://www.elsevier.com/journal-authors/ethics>.

Submission of an article to the journals of the Central Asian Academic Research Center LLP implies that the described work has not been published previously (except in the form of an abstract or as part of a published lecture or academic thesis or as an electronic preprint, see <http://www.elsevier.com/postingpolicy>), that it is not under consideration for publication elsewhere, that its publication is approved by all authors and tacitly or explicitly by the responsible authorities where the work was carried out, and that, if accepted, it will not be published elsewhere in the same form, in English or in any other language, including electronically without the written consent of the copyright-holder. In particular, translations into English of papers already published in another language are not accepted.

No other forms of scientific misconduct are allowed, such as plagiarism, falsification, fraudulent data, incorrect interpretation of other works, incorrect citations, etc. The Central Asian Academic Research Center LLP follows the Code of Conduct of the Committee on Publication Ethics (COPE), and follows the COPE Flowcharts for Resolving Cases of Suspected Misconduct (http://publicationethics.org/files/u2/New_Code.pdf). To verify originality, your article may be checked by the Cross Check originality detection service <http://www.elsevier.com/editors/plagdetect>.

The authors are obliged to participate in peer review process and be ready to provide corrections, clarifications, retractions and apologies when needed. All authors of a paper should have significantly contributed to the research.

The reviewers should provide objective judgments and should point out relevant published works which are not yet cited. Reviewed articles should be treated confidentially. The reviewers will be chosen in such a way that there is no conflict of interests with respect to the research, the authors and/or the research funders.

The editors have complete responsibility and authority to reject or accept a paper, and they will only accept a paper when reasonably certain. They will preserve anonymity of reviewers and promote publication of corrections, clarifications, retractions and apologies when needed. The acceptance of a paper automatically implies the copyright transfer to the Central Asian Academic Research Center LLP.

The Editorial Board of the Central Asian Academic Research Center LLP will monitor and safeguard publishing ethics.

Requirements for articles design for publication in the journal are available on the websites:

www.nauka-nanrk.kz

<http://physics-mathematics.kz/index.php/en/archive>

ISSN2518-1726 (Online),

ISSN 1991-346X (Print)

Managing Editor: *A.Shormakova*

Editors: *D.S. Alenov, T. Apendiev*

Computer layout: *G.D. Zhadyranova*

Signed for print: September 30, 2026

Format: 70×90 1/16. 26.5 printed sheets. Order No. 3.