УДК 004.382 EDN WZELFS
https://www.doi.org/10.47813/mip.7.2026.20.3008

Проактивное распределение ресурсов облачных вычислительных систем с использованием ансамблевого прогнозирования нагрузки

Р.А. Золотарев*, Г.В. Смородников

Санкт-Петербургский политехнический университет Петра Великого, Институт компьютерных наук и кибербезопасности, Высшая школа программной инженерии, Политехническая ул., 29, Санкт-Петербург, 195251, Россия

*E-mail: lorax710@gmail.com

Аннотация. Рассмотрена задача проактивного распределения виртуальных машин в облачной вычислительной системе с использованием краткосрочных прогнозов нагрузки. Предложена интегрированная архитектура, объединяющая сервис формирования прогнозов, хранилище временных рядов, скоринговый модуль выбора физического сервера, исполнительный компонент и механизм динамической миграции виртуальных машин. Для прогнозирования предусмотрено применение ансамблевых моделей, объединяющих рекуррентные и свёрточные методы анализа временных рядов с полносвязным агрегатором. Прогнозируемые значения используются совместно с текущими характеристиками CPU, RAM и другими параметрами при вычислении оценки пригодности физических узлов. Проведено экспериментальное исследование в среде CloudSim. Показано, что совместное применение прогнозирования и динамического перераспределения повышает среднюю загрузку CPU с 61 до 74 %, снижает долю перегруженных узлов с 17 до 2 %, а дисперсию загрузки CPU – с 140,2 до 47,2. При масштабировании до 500 физических серверов и 5000 виртуальных машин среднее время обработки заявки составило 142 мс. Полученные результаты подтверждают целесообразность интеграции прогнозной аналитики непосредственно в контур управления ресурсами облачной инфраструктуры.

Ключевые слова: облачные вычисления, распределение ресурсов, прогнозирование нагрузки, ансамблевое прогнозирование, виртуальные машины, скоринговый алгоритм.

Для цитирования: Золотарев, Р. А. Проактивное распределение ресурсов облачных вычислительных систем с использованием ансамблевого прогнозирования нагрузки / Р. А. Золотарев, Г. В. Смородников // Модернизация, Инновации, Прогресс (MIP ENGINEERING-VII-2026): материалы VII Всероссийской (национальной) научной конференции (Красноярск, 11–12 сентября 2026 г.). – Красноярск: Красноярский краевой Дом науки и техники Российского Союза научных и инженерных общественных объединений, 2026. – Т. 20. – С. 57-70. – https://doi.org/10.47813/mip.7.2026.20.3008. – EDN: WZELFS.

Proactive resource allocation in cloud computing systems using ensemble workload forecasting

R. A. Zolotarev*, G. V. Smorodnikov

Peter the Great St. Petersburg Polytechnic University, Institute of Computer Science and Cybersecurity, Higher School of Software Engineering, 29 Polytechnicheskaya St., Saint Petersburg, 195251, Russia

*E-mail: lorax710@gmail.com

Abstract. Proactive virtual machine allocation in a cloud computing environment using short-term workload forecasts is considered. An integrated architecture combining a forecasting service, a time-series database, a scoring-based physical server selection module, an execution service, and a dynamic virtual machine migration mechanism is proposed. The forecasting subsystem supports ensemble models combining recurrent and convolutional time-series processing methods with a fully connected aggregation model. Forecasted values are incorporated into the server suitability score together with current CPU and RAM utilization and other resource characteristics. The developed approach was evaluated using a CloudSim-based simulation environment. Experimental results demonstrate that combining forecasting with dynamic reallocation increases average CPU utilization from 61% to 74%, reduces the share of overloaded nodes from 17% to 2%, and decreases CPU load variance from 140.2 to 47.2. In scalability testing with 500 physical servers and 5000 virtual machines, the average request placement time

was 142 ms. The obtained results demonstrate that integrating predictive analytics directly into the cloud resource management loop enables more stable and efficient allocation while preserving acceptable online decision-making latency.

Keywords: cloud computing, resource allocation, workload forecasting, ensemble forecasting, virtual machines, scoring algorithm.

For citation: Zolotarev, R. A. Proactive resource allocation in cloud computing systems using ensemble workload forecasting / R. A. Zolotarev, G. V. Smorodnikov // Modernization, Innovations, Progress (MIP ENGINEERING-VII-2026): Proceedings of the VII All-Russian (National) Scientific Conference (Krasnoyarsk, September 11–12, 2026). – Krasnoyarsk: Krasnoyarsk Regional Science and Technology City Hall of the Russian Union of Scientific and Engineering Public Associations, 2026. – Vol. 20. – P. 57-70. – https://doi.org/10.47813/mip.7.2026.20.3008. – EDN: WZELFS.

1. Введение

Облачные вычисления являются одним из основных способов предоставления масштабируемых вычислительных ресурсов и позволяют динамически формировать виртуальную инфраструктуру в соответствии с изменяющимися требованиями пользователей [1]. Существенную роль в эффективности облачной платформы играет механизм размещения виртуальных машин на физических серверах. Неоптимальное размещение приводит одновременно к двум противоположным проблемам: часть физических узлов может быть перегружена, тогда как вычислительные мощности других серверов остаются недоиспользованными. Следствием становятся рост задержек, снижение стабильности, избыточные миграции виртуальных машин и повышение риска нарушения соглашений об уровне обслуживания.

Традиционные алгоритмы распределения ресурсов, включая Round Robin, First Fit и различные варианты статического скоринга, преимущественно принимают решения на основании текущего состояния инфраструктуры. Такой подход обеспечивает относительно небольшие вычислительные накладные расходы, однако обладает реактивным характером. Если нагрузка быстро изменяется, состояние сервера в момент размещения новой виртуальной машины может существенно отличаться от его состояния через несколько следующих интервалов времени. В результате корректное в текущий момент решение способно привести к локальной перегрузке узла уже в ближайшем будущем.

Одним из способов перехода от реактивного к проактивному управлению является использование прогнозирования нагрузки. В исследованиях методов прогнозирования облачных рабочих нагрузок показано, что предварительная оценка будущего потребления ресурсов может применяться для масштабирования, планирования, размещения задач и управления активными серверами [2-4]. Особое распространение получили методы глубокого обучения, ориентированные на обработку временных последовательностей. Рекуррентные сети LSTM позволяют учитывать долговременные зависимости [5], а временные и обычные свёрточные модели обеспечивают эффективное выделение локальных закономерностей.

Комбинированные архитектуры способны использовать преимущества нескольких типов моделей [6-9].

Однако высокая точность прогнозирования сама по себе не гарантирует повышения эффективности облачной инфраструктуры. Практическое значение приобретает способ включения прогнозируемой информации непосредственно в алгоритм принятия решений. Систематические исследования prediction-based scheduling указывают на необходимость совместного рассмотрения прогнозирования и планирования ресурсов, поскольку именно результирующие показатели инфраструктуры – загрузка узлов, количество перегрузок, равномерность распределения, задержка принятия решения – определяют эффект прогнозной подсистемы [7].

Предыдущие исследования авторов были посвящены отдельным составляющим рассматриваемой задачи. Исследовался скоринговый алгоритм выбора физических серверов с использованием текущих и прогнозируемых метрик [10], предложены ансамблевые LSTM- и TCN-модели краткосрочного прогнозирования [11], а также рассмотрена интеграция LSTM-прогнозирования с традиционными алгоритмами распределения ресурсов [12]. В настоящей работе эти направления объединяются в единый контур проактивного управления. Основное внимание уделяется не только прогнозу как самостоятельной задаче, но и его влиянию на конечные характеристики распределения ресурсов.

Целью исследования является разработка и экспериментальная оценка проактивного подхода к распределению виртуальных машин, при котором краткосрочные прогнозы нагрузки формируются ансамблевой прогнозной подсистемой и непосредственно учитываются адаптивной скоринговой функцией при выборе физического сервера. Дополнительно исследуется совместное влияние прогнозирования и фоновой миграции виртуальных машин, а также производительность решения при масштабировании моделируемого кластера.

2. Постановка задачи (Цель исследования)

Рассмотрим облачный кластер, состоящий из множества физических серверов

$$S = \{s_1, s_2, \dots, s_n\}.$$

Для каждого сервера s_iв момент времени tдоступен вектор текущего состояния

$$x_i(t) = \{C, P, U_i, (t), RAM_i(t) Disk_i(t) VM_i(t) \dots\},$$

а прогнозная подсистема формирует оценку будущего состояния

$$\hat{x}_i(t + \Delta t),$$

где Δtсоответствует выбранному горизонту прогнозирования.

При поступлении запроса на создание виртуальной машины vс требованиями CPU_v, RAM_vи $Disk_v$на первом этапе исключаются физические узлы, не удовлетворяющие минимальным ресурсным требованиям:

$$CPU_i^{available} \geq CPU_v,$$
$$RAM_i^{available} \geq RAM_v,$$
$$Disk_i^{available} \geq Disk_v.$$

В результате формируется множество допустимых серверов $S_f \subseteq S$. Для каждого элемента $s_i \in S_f$вычисляется интегральная оценка пригодности S_i, учитывающая как текущие, так и прогнозируемые характеристики. Целевой сервер определяется выражением

$$s^* = \arg\max_{s_i \in S_f} S_i. \qquad (1)$$

При этом необходимо одновременно обеспечить несколько критериев: повышать фактическую утилизацию вычислительных ресурсов, снижать число серверов, достигающих критического уровня загрузки, уменьшать неравномерность распределения и сохранять приемлемое время принятия решения.

Дополнительным требованием является отказоустойчивость прогнозного контура. Недоступность прогнозной подсистемы не должна блокировать размещение ВМ. В этом случае предусмотрен fallback-режим, в котором скоринговый алгоритм продолжает функционировать на основании текущих значений метрик. Такой режим был заложен в требования и реализован в программной системе.

3. Методы и материалы исследования

3.1. Архитектура проактивного управления ресурсами

Разработанные средства построены в виде распределённой микросервисной системы. Основными компонентами являются сервис симуляции облачной инфраструктуры, сервис прогнозирования, база временных рядов InfluxDB, сервис Allocation, исполнительный сервис Executor и фоновый планировщик миграций.

Для моделирования облачной инфраструктуры используется среда CloudSim [13], позволяющая воспроизводить работу физических серверов, виртуальных машин и механизмов управления вычислительными ресурсами. В рамках разработанного стенда CloudSim применяется для моделирования текущего состояния кластера, изменения нагрузки, размещения и миграции виртуальных машин. Прогнозный сервис получает временные ряды,

формирует будущие значения и записывает результаты в ту же логическую систему хранения. Благодаря этому Allocation при обработке новой заявки получает согласованный набор текущих и ожидаемых показателей. Аналогичная архитектура детально реализована в ВКР: после получения заявки Allocation извлекает текущие и прогнозируемые данные, фильтрует узлы, рассчитывает скоринг и передаёт выбранный сервер в Executor.

После выбора физического узла Executor инициирует фактическое размещение в симуляционной среде. Параллельно работает механизм Dynamic RA Scheduler, который периодически анализирует состояние кластера и при обнаружении неблагоприятного распределения инициирует миграции ВМ. Таким образом, начальное размещение виртуальных машин является проактивным, а фоновое перераспределение выполняет функцию дополнительной коррекции. Взаимодействие перечисленных компонентов в рамках единого контура прогнозирования, размещения и последующего перераспределения виртуальных машин представлено на рисунке 1.

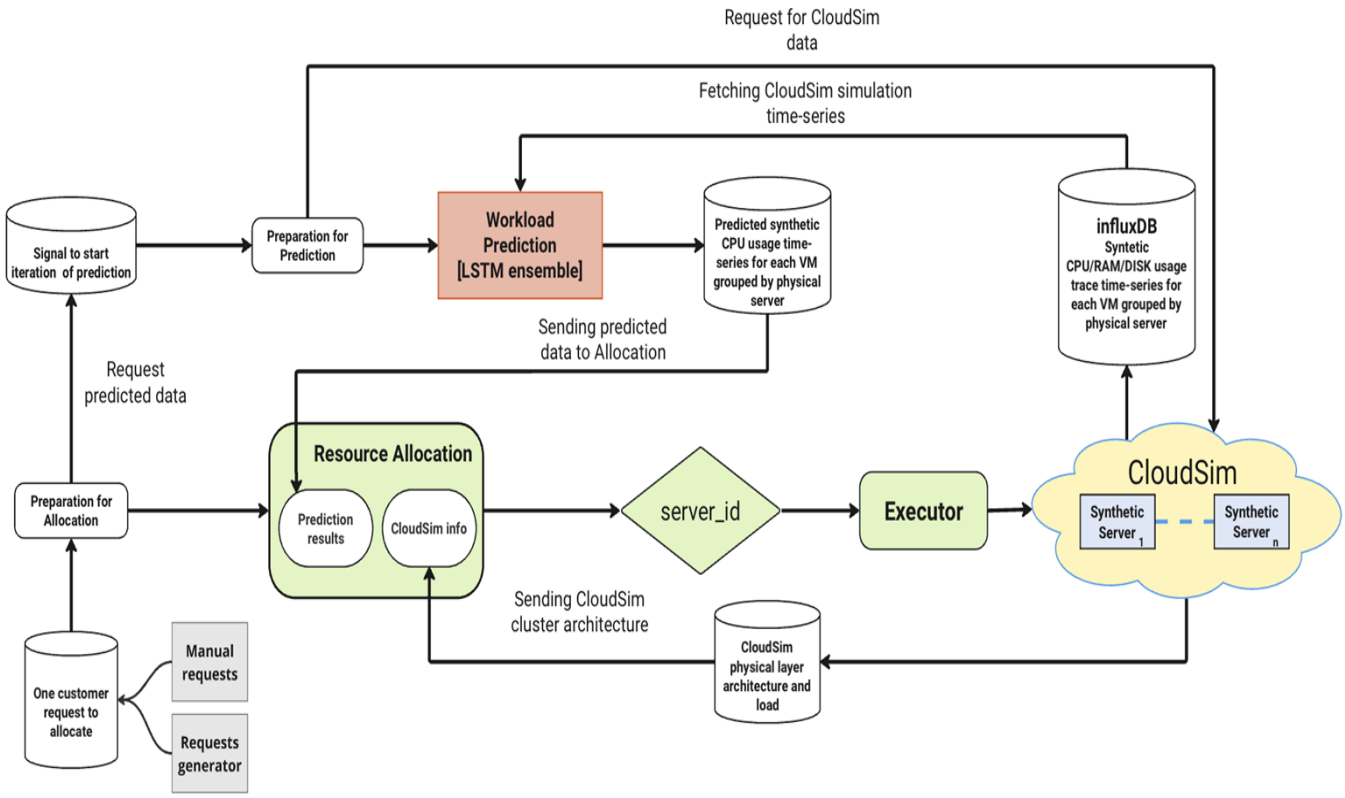

Рисунок 1. Архитектура проактивного распределения ресурсов с использованием прогнозной подсистемы.

3.2. Ансамблевое краткосрочное прогнозирование

Для формирования прогноза разработан отдельный сервис, поддерживающий взаимозаменяемые стратегии прогнозирования. Такой подход позволяет изменять модель без модификации логики Allocation и обеспечивает возможность экспериментировать с различными архитектурами в рамках единого API.

В универсальной ансамблевой схеме временной ряд загрузки одновременно обрабатывается рекуррентной моделью LSTM и временной свёрточной сетью TCN. Полученные предсказания объединяются и поступают на полносвязный агрегатор DNN/MLP, формирующий итоговое значение. Для каждой виртуальной машины используются 60 исторических отсчётов с шагом 5 минут, то есть информация за предыдущие пять часов; на выходе формируются 12 будущих значений, соответствующих часовому горизонту прогнозирования. Такая структура позволяет сочетать способность LSTM моделировать последовательные зависимости и возможность TCN выделять временные закономерности посредством каузальных расширенных свёрток. В ВКР данная схема описывается как массовое прогнозирование для 250 ВМ.

Сервис поддерживает и другие варианты стратегии. В частности, серверный сценарий использует несколько параллельных моделей для обработки матрицы «время × виртуальная машина», после чего их выходы агрегируются полносвязной сетью. Реализованный механизм Strategy позволяет динамически подключать новые методы прогнозирования без изменения управляющего контура. В фактической реализации предусмотрено формирование почасового прогноза в виде 12 точек с пятиминутным интервалом.

Для контроля прогнозов сервис содержит набор метрик MSE, RMSE, MAE, MAPE и sMAPE; в методической части дополнительно используется коэффициент детерминации R^2. В настоящей статье не вводятся искусственно полученные значения точности отдельных моделей: основной эксперимент направлен на оценку прикладного эффекта прогнозной информации в процессе распределения ресурсов. Такой подход позволяет отделить качество изолированной модели прогнозирования от эффективности замкнутого контура управления. Схема взаимодействия сервиса прогнозирования с хранилищем временных рядов, симуляционной средой CloudSim и сервисом распределения ресурсов представлена на рисунке 2.

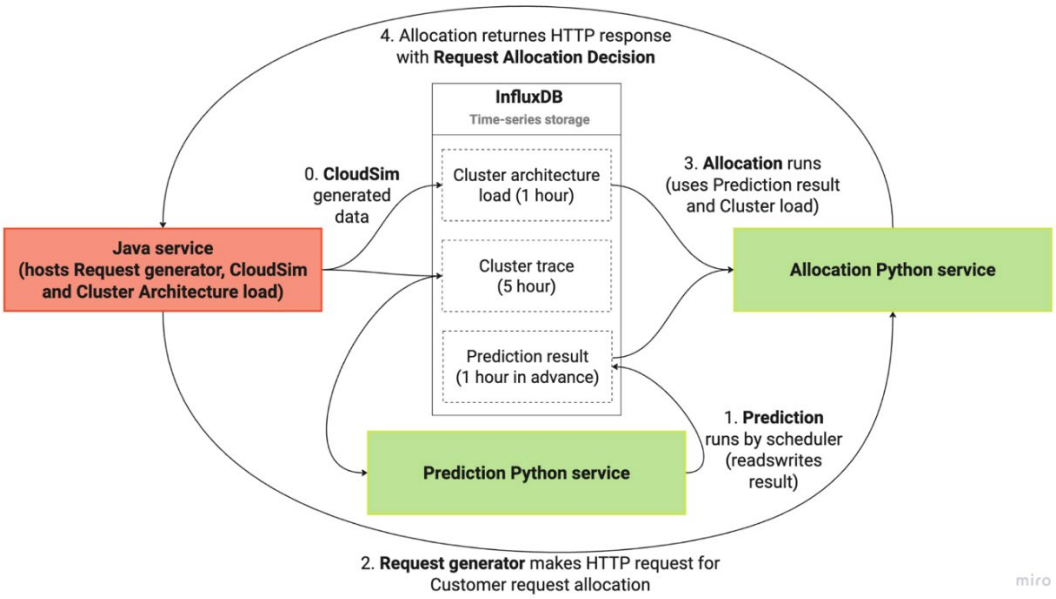

Рисунок 2. Схема взаимодействия подсистем прогнозирования и распределения ресурсов облачной системы.

3.3. Адаптивная скоринговая оценка серверов

После фильтрации физические узлы оцениваются по нескольким критериям. В качестве базовых используются загрузка CPU, использование оперативной памяти, дисковых ресурсов, количество размещённых ВМ и показатель надёжности. Исходные весовые коэффициенты составляют соответственно 0,25; 0,25; 0,15; 0,15 и 0,20. В отличие от статической взвешенной суммы веса адаптируются в зависимости от состояния всего кластера. Для критерия kопределяется коэффициент

$$factor_k = 1 + \bar{load}_k,$$

после чего нормализованный адаптивный вес вычисляется как

$$w_k^{adapt} = \frac{w_k \cdot factor_k}{\sum_{j=1}^{N} w_j \cdot factor_j}. \qquad (2)$$

При росте среднего потребления определённого ресурса его влияние на итоговую оценку автоматически увеличивается. Такой механизм позволяет без переключения алгоритма адаптировать стратегию к текущему профилю нагрузки.

Обобщённая итоговая функция пригодности представляется в виде

$$S_i = \sum_{k=1}^{N} w_k^{adapt} f_k(i) + \alpha EE_i + \beta FT_i + \gamma RF_i - P_i + B_i, \qquad (3)$$

где $f_k(i)$ – нормализованные оценки базовых ресурсов; EE_iхарактеризует эффективность использования ресурсов; FT_i – устойчивость к отказам; RF_i – соответствие свободных ресурсов требованиям новой ВМ; P_i – штрафы за приближение к критическим порогам; B_i – дополнительные бонусы для предпочтительных состояний узла.

Важным отличием алгоритма является включение прогнозной нагрузки в набор факторов, влияющих на score. Поэтому сервер с низкой текущей загрузкой, но ожидаемым значительным ростом CPU, может получить более низкую оценку, чем узел с сопоставимым текущим состоянием и стабильным прогнозом. Именно эта особенность обеспечивает переход от реактивного к упреждающему размещению.

3.4. Методика экспериментального исследования

Эксперименты проводились в изолированной симуляционной среде на основе CloudSim. В контур тестирования были включены сервис прогнозирования, InfluxDB, скоринговый модуль Allocation, Executor, механизм миграции и система регистрации метрик. Использовались модульный, функциональный и интеграционный уровни тестирования; для воспроизводимости сценариев применялись одинаковые начальные условия.

Для определения самостоятельного вклада прогнозирования и миграций было сформировано четыре режима:

1. без прогноза и без миграций;
2. с прогнозом, без миграций;
3. без прогноза, с миграциями;
4. с прогнозом и миграциями.

В основном эксперименте использовались 300 физических серверов и 1000 последовательных заявок на ВМ с различными ресурсными профилями. Для всех четырёх режимов применялась одинаковая последовательность запросов.

Дополнительно выполнено сравнение с Round Robin, BinPack, First Fit, Best Fit и Worst Fit. На заключительном этапе проводилось нагрузочное тестирование при увеличении размера инфраструктуры до 500 серверов и 5000 активных ВМ.

4. Полученные результаты

4.1. Влияние прогнозирования и миграций

В таблице 1 приведены результаты четырёх режимов работы системы.

Таблица 1. Влияние прогнозирования и миграции на характеристики облачного кластера.

Сценарий	CPU, %	RAM, %	Disk, %	Перегруженные узлы, %	Дисперсия CPU	Миграции
Без прогноза, без миграций	61	58	12	17	140,2	0
С прогнозом, без миграций	68	66	15	11	97,4	0
Без прогноза, с миграциями	65	62	15	9	83,1	238
С прогнозом и миграциями	74	70	17	2	47,2	266

Использование прогнозирования без миграций обеспечивает заметный положительный эффект: средняя загрузка CPU увеличивается с 61 до 68 %, доля перегруженных серверов уменьшается с 17 до 11 %, а дисперсия – со 140,2 до 97,4. Следовательно, прогнозная информация влияет на качество первоначального размещения ещё до применения корректирующих механизмов.

Миграции без прогнозирования обеспечивают более существенное снижение числа перегрузок, однако остаются реактивным механизмом: перемещение выполняется после формирования неблагоприятного состояния. Максимальный эффект наблюдается при совместном использовании обоих инструментов.

Относительно базового режима средняя загрузка CPU повышается на **13 процентных пунктов**, или приблизительно на **21,3 % относительно исходного уровня**. Средняя загрузка RAM увеличивается на 12 п.п. Доля перегруженных серверов уменьшается с 17 до 2 %, то есть в **8,5 раза**. Дисперсия загрузки CPU сокращается с 140,2 до 47,2 – приблизительно в **2,97 раза**, или на 66,3 %.

Дополнительные метрики подтверждают этот результат. Стандартное отклонение загрузки CPU уменьшается с 11,8 до 4,9 %, максимальная разница между наиболее и наименее загруженными узлами – с 56 до 25 %, а коэффициент вариации – с 0,193 до 0,064. Таким образом, повышение средней утилизации не сопровождается концентрацией нагрузки на отдельных узлах, а достигается одновременно с повышением равномерности.

4.2. Сравнение с классическими алгоритмами

Результаты сопоставления с распространёнными алгоритмами представлены в таблице 2.

Таблица 2. Сравнение алгоритмов распределения ресурсов.

Алгоритм	CPU, %	RAM, %	Среднее время, мс	Перегруженные узлы, %	Дисперсия CPU	CV CPU
Round Robin	60	58	34	15	130,4	0,187
BinPack (RAM)	66	72	48	10	91,2	0,138
First Fit	64	61	39	12	102,5	0,162
Best Fit (CPU)	67	69	49	9	85,3	0,128
Worst Fit (CPU)	65	63	46	10	93,2	0,142
Предложенная система	74	70	74	2	47,2	0,064

Сравнительный эксперимент показывает, что прогнозно-ориентированный скоринг имеет большие вычислительные накладные расходы по сравнению с простыми эвристиками: среднее время выбора составляет 74 мс против 34 мс у Round Robin. Однако эта величина остаётся значительно ниже заданного ограничения в 150 мс.

При этом выигрыш по эксплуатационным характеристикам существенно выше. По сравнению с Round Robin средняя загрузка CPU увеличивается на 14 п.п., доля перегруженных узлов уменьшается в 7,5 раза, а дисперсия – приблизительно на 63,8 %. Даже относительно более эффективного Best Fit средняя загрузка CPU выше на 7 п.п., количество перегруженных узлов ниже в 4,5 раза, а дисперсия CPU меньше приблизительно на 44,7 %.

4.3. Масштабируемость

Для оценки масштабируемости разработанных средств проводилось последовательное увеличение числа физических серверов и виртуальных машин в моделируемой облачной инфраструктуре. Основные результаты нагрузочного тестирования представлены в таблице 3.

Таблица 3. Результаты нагрузочного тестирования

Конфигурация	Интенсивность	Среднее время, мс	Максимальное время, мс
50 серверов / 500 ВМ	50 rps	45	70
100 серверов / 1000 ВМ	100 rps	72	108
200 серверов / 2000 ВМ	200 rps	98	145
300 серверов / 3000 ВМ	300 rps	116	170
400 серверов / 4000 ВМ	300 rps	131	192
500 серверов / 5000 ВМ	300 rps	142	205

До приблизительно 200 серверов наблюдается близкий к линейному рост времени обработки. Для более крупных конфигураций применяются пакетная обработка и мультипроцессинг. При 500 серверах и 5000 ВМ среднее время составляет 142 мс. Максимальное зарегистрированное значение достигает 205 мс, поэтому корректно говорить именно о выполнении ограничения по **среднему**, а не максимальному времени обработки.

Полученные показатели выявляют, что прогнозно-ориентированный подход может использоваться в режиме онлайн без неприемлемого роста задержки выбора физического сервера. Вместе с поддержкой fallback-режима это позволяет не ставить функционирование облачной инфраструктуры в жёсткую зависимость от доступности ML-подсистемы.

Следует отметить ограничение проведённого исследования: основной эксперимент выполнен в контролируемой среде CloudSim с воспроизводимыми синтетическими сценариями. Благодаря этому удаётся корректно сравнить алгоритмы при одинаковых входных условиях, однако полученные значения не следует автоматически переносить на конкретный промышленный дата-центр. Следующим этапом исследования должно стать воспроизведение эксперимента на реальных трассах Google Cluster, Alibaba Cluster Trace или на телеметрии эксплуатационного кластера, а также отдельная оценка связи между MAPE/R^2 прогнозной модели и конечными показателями распределения.

5. Выводы

Предложен подход к проактивному распределению ресурсов облачной вычислительной системы, объединяющий ансамблевое краткосрочное прогнозирование

нагрузки, адаптивную скоринговую оценку физических серверов и динамическое перераспределение виртуальных машин.

В отличие от реактивных схем прогнозируемое состояние инфраструктуры используется непосредственно при первоначальном выборе физического узла. Прогнозная подсистема реализована как заменяемый микросервис и поддерживает ансамблевую обработку временных рядов с параллельным использованием моделей разных типов и последующей агрегацией результатов. Это позволяет развивать методы прогнозирования независимо от логики распределения ресурсов.

Экспериментальное исследование подтвердило, что прогнозирование имеет положительный эффект даже без последующих миграций. Наибольший результат достигнут при объединении обоих механизмов: средняя загрузка CPU увеличилась с 61 до 74 %, доля перегруженных узлов снизилась с 17 до 2 %, а дисперсия загрузки уменьшилась почти в три раза. По сравнению с Round Robin, BinPack, First Fit, Best Fit и Worst Fit разработанная система демонстрирует наиболее низкую долю перегруженных узлов и наиболее равномерное распределение CPU при сохранении допустимой задержки принятия решения.

При масштабировании до 500 серверов и 5000 ВМ среднее время размещения составило 142 мс. Полученный результат подтверждает возможность применения прогнозной аналитики в контуре управления, работающим в реальном времени, при условии оптимизации вычислений и сохранения fallback-режима.

Перспективным направлением дальнейшей работы является использование вероятностных оценок и доверительных интервалов прогнозирования непосредственно в скоринговой функции. Это позволит учитывать не только прогнозируемую величину нагрузки, но и неопределённость прогноза. Дополнительный интерес представляет автоматический выбор прогнозирующей стратегии в зависимости от характеристик временного ряда, а также экспериментальное исследование системы на реальных трассах промышленной облачной инфраструктуры.

Список литературы

1. Armbrust, M. A View of Cloud Computing / M. Armbrust, A. Fox, R. Griffith [et al.] // Communications of the ACM. – 2010. – Vol. 53, No. 4. – P. 50–58. – https://doi.org/10.1145/1721654.1721672.
2. Kashyap, S. Prediction-based scheduling techniques for cloud data center’s workload: a systematic review / S. Kashyap, A. Singh // Cluster Computing. – 2023. – Vol. 26. – P. 3209–3235. – https://doi.org/10.1007/s10586-023-04024-8.

3. Masdari, M. A survey and classification of the workload forecasting methods in cloud computing / M. Masdari, A. Khoshnevis // Cluster Computing. – 2020. – Vol. 23. – P. 2399–2424. – https://doi.org/10.1007/s10586-019-03010-3.
4. Nawrocki, P. Data-Driven Adaptive Prediction of Cloud Resource Usage / P. Nawrocki, P. Osypanka, B. Posluszny // Journal of Grid Computing. – 2023. – Vol. 21. – Art. 6. – https://doi.org/10.1007/s10723-022-09641-y.
5. Hochreiter, S. Long Short-Term Memory / S. Hochreiter, J. Schmidhuber // Neural Computation. – 1997. – Vol. 9, No. 8. – P. 1735–1780. – https://doi.org/10.1162/neco.1997.9.8.1735.
6. Ahamed, Z. Technical Study of Deep Learning in Cloud Computing for Accurate Workload Prediction / Z. Ahamed, M. Khemakhem, F. Eassa, F. Alsolami, A. S. Al-Malaise Al-Ghamdi // Electronics. – 2023. – Vol. 12, No. 3. – Art. 650. – https://doi.org/10.3390/electronics12030650.
7. Patel, E. A hybrid CNN-LSTM model for predicting server load in cloud computing / E. Patel, D. S. Kushwaha // The Journal of Supercomputing. – 2022. – Vol. 78. – P. 10327–10356. – https://doi.org/10.1007/s11227-021-04234-0.
8. Simaiya, S. A hybrid cloud load balancing and host utilization prediction method using deep learning and optimization techniques / S. Simaiya, U. K. Lilhore, Y. K. Sharma [et al.] // Scientific Reports. – 2024. – Vol. 14. – Art. 1337. – https://doi.org/10.1038/s41598-024-51466-0.
9. Yuan, H. An Improved LSTM-Based Prediction Approach for Resources and Workload in Large-Scale Data Centers / H. Yuan, J. Bi, S. Li, J. Zhang, M. C. Zhou // IEEE Internet of Things Journal. – 2024. – Vol. 11, No. 12. – P. 22816–22829. – https://doi.org/10.1109/JIOT.2024.3383512.
10. Золотарев, Р. А. Скоринговый алгоритм распределения ресурсов в облачных системах на основе текущих и прогнозируемых метрик / Р. А. Золотарев, А. В. Самочадин, Р. Р. Попов, Г. В. Смородников // Перспективы науки. – 2025. – № 8(191). – С. 16–19. – URL: https://moofrnk.com/assets/files/journals/science-prospects/191/science-prospect-8%28191%29main.pdf (дата обращения: 08.09.2026).
11. Смородников, Г. В. Ансамблевые модели LSTM и TCN для краткосрочного прогнозирования нагрузки в облачных вычислительных системах / Г. В. Смородников, Р. А. Золотарев, А. В. Самочадин, Р. Р. Попов // Перспективы науки. – 2025. – № 8(191). – С. 32–35. – URL: https://moofrnk.com/assets/files/journals/science-prospects/191/science-prospect-8%28191%29main.pdf (дата обращения: 08.09.2026).
12. Smorodnikov, G. Elastic cloud resource allocation using short-term long short-term memory-based workload prediction / G. Smorodnikov, R. Zolotarev, A. Rykova, A. Sabutkevich, A.

Samochadin // Fourth International Conference on Optics, Computer Applications, and Materials Science (CMSD-IV 2024): Proceedings of SPIE. – 2025. – Vol. 13651. – Art. 136510J. – https://doi.org/10.1117/12.3060861.

13. Calheiros, R. N. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms / R. N. Calheiros, R. Ranjan, A. Beloglazov [et al.] // Software: Practice and Experience. – 2011. – Vol. 41, No. 1. – P. 23–50. – https://doi.org/10.1002/spe.995.