УДК 004.032
doi:10.31799/1684-8853-2022-4-12-19

Научные статьи
Articles

Оценка времени отклика среды для вычислений с интенсивным использованием данных

А. В. Горбунова[a], *канд. физ.-мат. наук, старший научный сотрудник, orcid.org/0000-0002-9183-0426, avgorbunova@list.ru*
В. М. Вишневский[a], *доктор техн. наук, профессор, orcid.org/0000-0001-7373-4847*
[a]*Институт проблем управления им. В. А. Трапезникова РАН, Профсоюзная ул., 65, Москва, 117997, РФ*

Введение: *объем цифровых данных непрерывно растет так же, как и потребность в их хранении и обработке в различных целях. Для проведения анализа данных используются высокопроизводительные вычислительные среды, связанные с методами распараллеливания, и, соответственно, приложения, интенсивно использующие данные. Отсутствие качественных инструментов оценки эффективности процесса параллельной обработки данных или задач приводит к избыточному выделению ресурсов.* ***Цель:*** *разработать математические модели сред для вычислений с интенсивным использованием данных и методы анализа их производительности, т. е. оценки среднего времени отклика системы на основе данных о производительности системы на уровне решения подзадач.* ***Результаты:*** *представлена математическая модель системы параллельных вычислений в виде системы массового обслуживания с параллельной обработкой заявок с различными вариантами архитектуры, в том числе с отличным от пуассоновского входящим потоком и неэкспоненциальным распределением времени обслуживания. В качестве метода анализа ее среднего времени отклика используется комбинация имитационного моделирования с одним из методов машинного обучения (искусственные нейронные сети). Эффективность метода подтверждается численными экспериментами и не зависит от типа входящего потока, типа распределения времени обслуживания заявок, а также от количества приборов в узлах системы. Погрешность аппроксимации среднего времени отклика не превышает 10 %, что позволяет оптимизировать общепринятую стратегию избыточного выделения ресурсов, значительно сократив их объем.* ***Практическая значимость:*** *представленные модели и метод их анализа могут быть использованы для эффективного планирования распределения ресурсов систем с интенсивным использованием данных.*

Ключевые слова *— приложения с интенсивным использованием данных, параллельные вычисления, система массового обслуживания, среднее время отклика, нейронные сети.*

Для цитирования: Горбунова А. В., Вишневский В. М. Оценка времени отклика среды для вычислений с интенсивным использованием данных. *Информационно-управляющие системы*, 2022, № 4, с. 12–19. doi:10.31799/1684-8853-2022-4-12–19
For citation: Gorbunova A. V., Vishnevsky V. M. Estimating the response time of a data-intensive computing environment. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2022, no. 4, pp. 12–19 (In Russian). doi:10.31799/1684-8853-2022-4-12–19

Введение

Согласно прогнозам одной из ведущих компаний в области цифровых технологий (IDC, International Data Corporation), к 2025 году глобальная сфера данных увеличится до 175 зеттабайт [1]. Это более чем в пять раз больше по сравнению с объемом цифровых данных, имевшихся по состоянию на 2018 год. При этом работа с большими данными стала доступной в основном благодаря развитию облачных технологий, а также появлению множества научных и технических приложений, пользующихся услугами облачных провайдеров для проведения вычислений и обработки данных [2–5].

Один из основных способов повышения производительности различного рода сервисов центров обработки данных заключается в распараллеливании вычислений [5]. К настоящему моменту разработано множество сред с параллельным подходом для ускорения работы приложений, интенсивно использующих данные [6].

В основе большинства сервисов центров обработки данных для проведения крупномасштабных вычислений находятся параллельные структуры. Они являются основным составным элементом процесса обработки данных систем параллельных и (или) распределенных вычислений. Однако в силу отсутствия хороших инструментов объективной оценки характеристик их производительности в настоящее время основной стратегией является избыточное выделение ресурсов, половина из которых простаивает большую часть времени [7]. Это приводит к значительному повышению затрат на содержание оборудования (серверов). Поэтому с точки зрения повышения производительности и эффективного использования ресурсов интерес представляет прогнозирование таких характеристик систем с интенсивным использованием данных, как ее среднее время отклика, в том числе в области высоких нагрузок.

В настоящей статье предлагается новый подход к оценке среднего времени отклика для высокопроизводительной вычислительной среды. Подход с использованием нейронных сетей (НС) позволяет довольно быстро и с приемлемой точностью получать оценки интересующих харак-

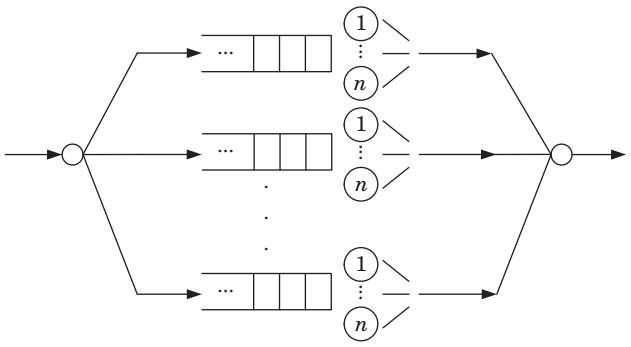
теристик. Преимущество подхода по сравнению с ранее известными заключается в универсальности, поскольку отсутствуют ограничения на архитектуру систем с параллельной обработкой заявок.

Система с параллельной обработкой заявок и известные методы ее анализа

В основе большинства центров обработки данных находятся параллельные структуры. Под параллельной структурой (системой) мы будем понимать систему массового обслуживания (СМО), каждый узел которой представляет собой самостоятельную СМО. При поступлении в систему задача (заявка) разбивается на части — независимые подзадачи (подзаявки), каждая из которых поступает на обслуживание на соответствующий узел (в подсистему). Задача считается выполненной после обработки последней из ее составляющих.

В классической СМО с параллельным обслуживанием заявок предполагается, что в каждой подсистеме имеется один обслуживающий прибор. Мы же расширим эту систему до самого общего случая, когда в каждой СМО может находиться по $n \geq 1$ приборов, т. е. каждая подсистема будет представлять собой СМО типа $G|G|n$ (рис. 1). Таким образом станет возможен анализ характеристик системы в условиях выделения дополнительных ресурсов. Результаты исследования позволят избежать избыточности в их выделении.

Одной из первых работ, посвященных анализу рассматриваемой СМО с подсистемами вида $M|M|1$, является статья [8]. Здесь было получено точное выражение для оценки среднего времени отклика системы в случае с двумя подсистемами $M|M|1$, $K = 2$. В большинстве следующих работ по данной тематике различными методами были получены только аппроксимации среднего времени отклика для $K > 2$. С подробным обзором публикаций можно ознакомиться, например, в [9, 10].



■ ***Рис. 1.*** Математическая модель системы с параллельной обработкой задач с подсистемами вида $G|G|n$

■ ***Fig. 1.*** Mathematical model of a system with parallel processing of tasks with subsystems of the form $G|G|n$

Среди основных приближенных методов исследований СМО с параллельным обслуживанием заявок с K подсистемами вида $M|M|1$ или $M|G|1$ можно перечислить матрично-геометрический подход; интерполяцию на основе данных, полученных в крайних случаях высоких и слабых входных нагрузок; подход с использованием элементов теории порядковых статистик; эмпирический подход (построение аналитических формул на основе данных, полученных с помощью симуляции) [8, 11–14].

Заметим, что точных решений для среднего времени отклика не существует даже в случае экспоненциального входящего и обслуживающих потоков при $K > 2$. Сложность анализа объясняется зависимостью времен пребывания подзаявок в подсистемах в силу их общих моментов поступления. Поэтому аналитический подход строится, как правило, на предположении об отсутствии этой зависимости. В результате применение полученных оценок ограничивается либо числом подсистем K, либо конкретным типом распределения обслуживающего потока, либо недостаточной точностью приближения. Это сужает область применения полученных аналитических выражений, в частности к современным системам, особенно если речь идет о распределениях с тяжелыми хвостами. Что касается случая с подсистемами более общего вида, т. е. $G|G|1$, то исследований на эту тему крайне мало, и появились они в основном в последнее время. Среди недавних работ, посвященных данной тематике, стоит отметить [15–19].

Система массового обслуживания с параллельным обслуживанием заявок является естественной моделью для многих реально существующих систем в различных областях, где осуществляется параллельная обработка заданий в целях повышения производительности. В данном случае речь идет не только о телекоммуникационных системах, но и о производственных сферах (сборка заказов, логистика и т. д.). Поэтому, несмотря на некоторое снижение активности исследований в этом направлении, их актуальность все еще велика [14].

Подход к исследованию СМО с параллельной обработкой заявок с использованием НС

Нейронные сети получили широкое распространение благодаря возможности их применения к решению слабо формализуемых задач.

Кроме того, они являются одним из основных инструментов анализа больших данных. Сложно в настоящее время назвать область, в которой НС или другие методы машинного обучения не нашли бы свое применение.

Математическая модель, которая используется для описания функционирования вычислительных сред с интенсивным использованием данных, представляет собой СМО. Методы машинного обучения и НС в частности к решению сложных задач теории массового обслуживания стали применяться относительно недавно, хотя данная перспектива была предсказуема. Под сложными задачами в области теории очередей понимаются задачи, решение которых невозможно получить с помощью известных аналитических методов, либо решение настолько сложно, что фактически получение численных результатов с помощью разработанных алгоритмов трудно реализуемо даже с учетом возможностей современных вычислительных машин. Обзор публикаций по применению методов машинного обучения к решению задач в области теории очередей можно найти в статье [20].

Идея применения НС базируется на возможности с их помощью решать задачу прогнозирования, т. е. задачу аппроксимации функции нескольких переменных. Понятно, что искомые оценки среднего времени отклика зависят, например, от значений таких параметров, как нагрузка системы, количество подсистем K и приборов в них, интенсивностей для входящего потока и времени обслуживания на приборах. Следовательно, задачу нахождения оценок времени отклика можно рассмотреть как задачу аппроксимации функции в зависимости от перечисленных параметров.

Для интерполяции функции необходим набор значений входных параметров и соответствующих им истинных значений аппроксимируемых величин. Получить истинные значения можно несколькими способами. Например, с помощью имитационного моделирования или с помощью точного аналитического решения, которое в данном случае отсутствует.

Необходимость в комбинации НС с имитационным моделированием или алгоритмическим решением возникает из-за значительных временных затрат, которые требуются при использовании только симуляции или только алгоритма. Поэтому мы ограничиваем количество входных данных, для которых необходимо использовать имитационное моделирование или вычислительный алгоритм. После чего на полученном наборе обучаем НС, которая позволит прогнозировать характеристики для любых промежуточных значений входных параметров без ограничений на их количество за минимальное время, сопоставимое со временем, необходимым для проведения расчетов по простой аналитической формуле.

В зависимости от доступной мощности вычислительной системы, программной среды для моделирования, уровня загрузки системы и т. д. время получения одного значения с помощью имитационного моделирования может варьироваться от нескольких десятков секунд до нескольких минут. При этом результат прогноза нейросети для заданного множества значений входных параметров выдается практически мгновенно. Таким образом, в зависимости от объема промежуточных значений, который ничем не ограничивается, время, затрачиваемое на оценивание искомых характеристик, значительно сокращается. Так, например, если количество промежуточных данных совпадает с количеством исходных, то общее время (с учетом имитационного моделирования данных для обучения нейросети) на оценку некоторой характеристики будет примерно в два раза меньше времени, потраченного на получение того же количества оценок, но только с помощью имитационной модели; если промежуточных данных в два раза больше исходных, то время сокращается в три раза и т. д.

Для обучения НС существует множество алгоритмов, большинство из которых реализовано в различных программных средах в виде готовых функций, как, например, в Python или MatLab. Поэтому наряду с написанием авторского программного кода одного из известных алгоритмов обучения можно воспользоваться готовым решением, что значительно ускоряет процесс, не требуя при этом слишком глубокого погружения в различные аспекты методов обучения нейросетей. При этом, как правило, в процессе любого обучения имеющаяся выборка исходных данных (имитационного моделирования) разбивается на тренировочную (и валидационную) выборку, на которой происходит непосредственное обучение нейросети одним из методов, и тестовую, на которой проверяется работоспособность обученной сети. Для более качественного анализа авторы проверяли работоспособность нейросети не только на тестовых данных, но и на существенном количестве промежуточных входных данных.

Среди основных преимуществ описанной методики можно выделить универсальность, так как имитационное моделирование иногда бывает единственно возможным способом анализа сложных систем. При этом оно является ресурсоемким инструментом. Благодаря применению нейросетей этот недостаток удается устранить.

Математическая модель параллельной вычислительной среды с интенсивным использованием данных

Более детально опишем архитектуру СМО с параллельным обслуживанием заявок, которую будем использовать для моделирования процесса функционирования параллельной вычислительной среды. Система состоит из K подсистем типа $G|G|n$ (см. рис. 1). Поступающая в систему задача, или в терминах теории очередей заявка, расщепляется на K подзаявок, каждая из которых встает в очередь в соответствующей подсистеме. Обслуживание в подсистемах происходит в порядке поступления подзаявок (дисциплина First In First Out, FIFO). В каждой подсистеме находится одинаковое число приборов n, $n \geq 1$. Выбор многолинейной системы позволит проанализировать повышение производительности узлов за счет выделения дополнительных ресурсов, поскольку таким образом моделируются дополнительные реплики серверов. Исходя из результатов исследования [17], ограничимся случаем с тремя репликами серверов, т. е. будем проверять точность предложенного подхода для $1 \leq n \leq 3$.

В качестве распределения времени обслуживания рассмотрим усеченное распределение Парето с плотностью распределения вида

$$f(x) = \frac{\alpha L^{\alpha} x^{-\alpha-1}}{1-(L/H)^{\alpha}},\ 0 \leq L \leq x \leq H,\ \alpha > 0.$$

Усеченное распределение Парето является трехпараметрическим. Параметр α — параметр формы, а параметры L и H фактически являются минимальным и максимальным значением случайной величины с данным распределением.

В статье [17] на основе эмпирических данных для поисковой системы Google предложены следующие значения параметров распределения времени обслуживания: $\alpha = 2{,}0119$, $L = 2{,}14$, $H = 276{,}6$, т. е. минимальное время обслуживания составляет 2,14 мс, а максимальное — 276,6 мс. Соответственно имеем, что среднее время обслуживания примерно равно 4,22 мс, дисперсия — 22,34, а коэффициент вариации CV, представляющий собой отношение корня из дисперсии к среднему значению случайной величины, будет составлять примерно 1,22.

Для распределения входящего потока рассмотрим несколько вариантов. Поскольку в отдельных исследованиях допускается предположение о пуассоновском характере входящего потока (CV = 1) для центров обработки данных, то не будем исключать экспоненциальное распределение для времени между соседними поступлениями заявок [17]. Также рассмотрим еще два типа распределения для входящего потока с коэффициентом вариации, отличным от единицы. В частности, для распределения Эрланга известно, что его коэффициент вариации всегда меньше единицы. Поэтому рассмотрим распределение Эрланга с плотностью вида

$$g(x) = \beta^2 x e^{-\beta x},\ x \geq 0, \beta > 0$$

и с CV $\approx$ 0,7. Также рассмотрим распределение с тяжелым хвостом (CV > 1), а именно гамма-распределение с плотностью

$$p(x) = \frac{\gamma^k}{\Gamma(\gamma)} x^{k-1} e^{-\gamma x},\ x \geq 0, \gamma > 0, k > 0,$$

для которого CV = 2.

Численный эксперимент

Проверим работоспособность и качество аппроксимации предложенного подхода с использованием НС. Первый вариант архитектуры высокопроизводительной вычислительной среды — это СМО с параллельной обработкой заявок с пуассоновским входящим потоком с интенсивностью λ, обратно пропорциональной среднему времени между соседними поступлениями заявок. Время обслуживания имеет усеченное распределение Парето с плотностью распределения $f(x)$ со средним значением $b = 4{,}22$ мс. Фактически каждая подсистема представляет собой СМО вида $M|G|n$.

Для обучения нейросети будем использовать входные данные, где коэффициент загрузки $\rho = \lambda b/n$ принимает значения на отрезке [0,1; 0,9] с шагом 0,1, число подсистем K меняется от двух до 24, а число приборов n — от одного до трех. В качестве выходных данных нейросети будет выступать среднее время отклика $E[R_K]$. Значения выходных параметров были получены с помощью имитационного моделирования, проведенного в программной среде Python.

Разобьем входные данные на два множества, соответствующие уровню слабой и высокой загрузок, т. е. для $\rho \in [0{,}1; 0{,}5]$ и $\rho \in [0{,}6; 0{,}9]$, и проведем обучение НС на каждом из этих двух наборов. При этом набор данных разбивается в соотношении 80 и 20 % на обучающую и тестовую выборки. По меркам НС нельзя сказать, что количество наборов данных для обучения велико. Тем не менее проверим точность работы предложенного подхода в данных условиях.

В качестве структуры нейросети выбран двухслойный персептрон с двумя скрытыми слоями по 10 нейронов в каждом с логистической функцией активации $\varphi(x) = 1/(1 + e^{-x})$. Обучение бу-

дем проводить методом обратного распространения ошибок или методом Адама в программной среде Python, где проводилось имитационное моделирование.

Для объективной проверки качества прогноза обученных НС будем использовать абсолютно незнакомые ей промежуточные данные. Для НС, обученной в области низких нагрузок, рассмотрим значения для прогноза $\rho \in [0{,}15;\ 0{,}55]$, а для НС, обученной в области более высоких нагрузок, — для $\rho \in [0{,}65;\ 0{,}85]$ с шагом 0,1 в обоих случаях.

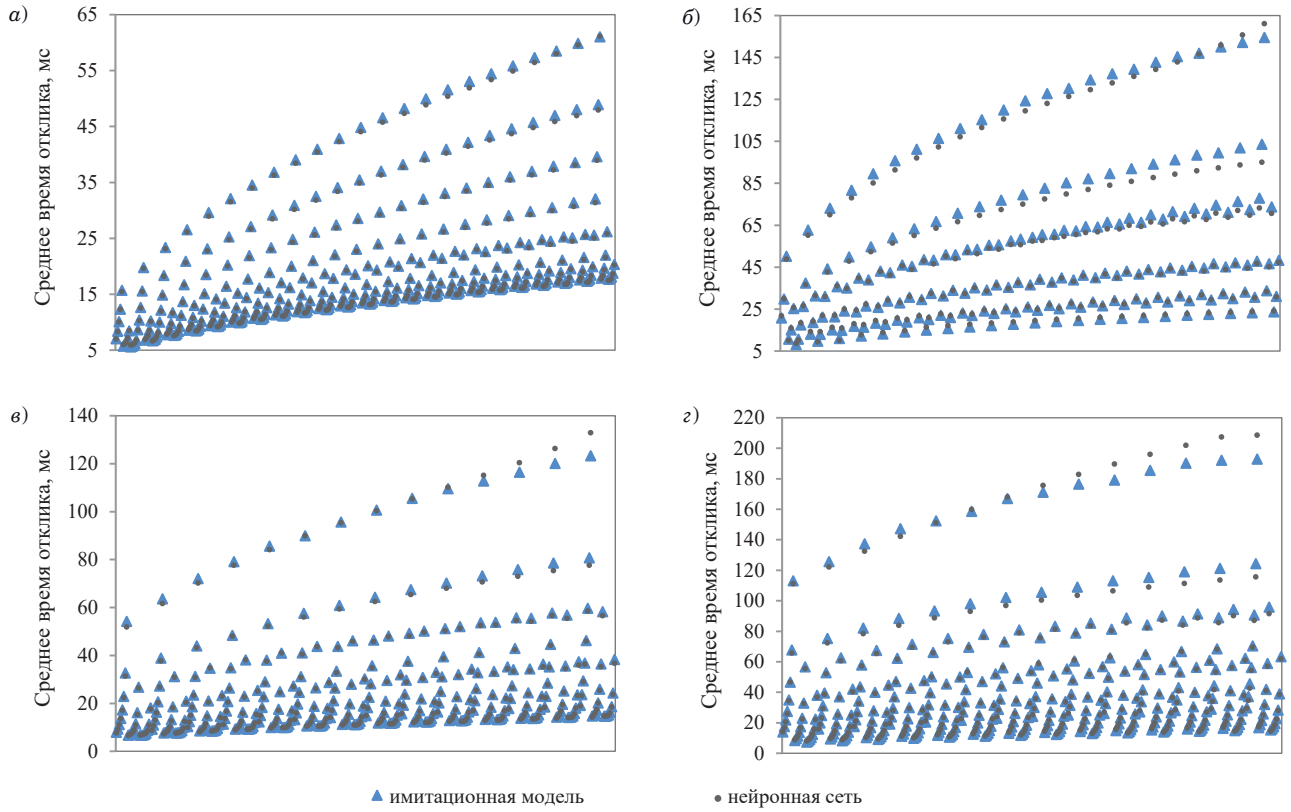
На рис. 2, *а–г* представлено отклонение оценок исследуемых характеристик от их истинных значений, полученных с помощью имитационного моделирования. Для большей детализации проанализируем относительную погрешность аппроксимации, а также ее среднее

$$\text{MAPE} = \frac{1}{n}\sum_{j=1}^{N}\left|\frac{y_j^* - y_j}{y_j}\right| \cdot 100\%,$$

максимальное и минимальное значения, где y_j^* — оценка исследуемой характеристики (математического ожидания времени отклика), полученная с помощью аналитических формул, а y_j — реальное значение оцениваемой характеристики, полученное в результате имитационного моделирования системы, $j = 1, \ldots, N$, N — количество наборов данных в выборке, предназначенной для оценки погрешности аппроксимации.

Примерно 99 % ошибок аппроксимации среднего времени отклика для $\rho \in [0{,}15;\ 0{,}55]$ не превышает 3 % (см. рис. 2, *а*). Для $\rho \in [0{,}65;\ 0{,}85]$ результат прогноза среднего времени отклика нейросетью оказался несколько хуже. В этом случае количество ошибок аппроксимации, не превышающее 7 %, составляет примерно 92 % от их общего количества (см. рис. 2, *б*). При этом максимальная ошибка аппроксимации не превышает 10 %.

Теперь проанализируем случай с распределением Эрланга для входящего потока с плотностью распределения $g(x)$. Для этого рассмотрим



■ ***Рис. 2.*** Сравнение результатов прогнозирования среднего времени отклика с результатами имитационного моделирования для СМО: *а* — с пуассоновским входящим потоком, $\rho \in [0{,}15;\ 0{,}55]$; *б* — с пуассоновским входящим потоком, $\rho \in [0{,}65;\ 0{,}85]$; *в* — с распределением Эрланга для входящего потока, $\rho \in [0{,}15;\ 0{,}85]$; *г* — с гамма-распределением для входящего потока, $\rho \in [0{,}15;\ 0{,}85]$

■ ***Fig. 2.*** Comparison of the results of predicting the average response time with simulation results for QS: *а* — with Poisson input, $\rho \in [0{,}15;\ 0{,}55]$; *б* — with Poisson input, $\rho \in [0{,}65;\ 0{,}85]$; *в* — with Erlang distribution for the incoming flow, $\rho \in [0{,}15;\ 0{,}85]$; *г* — with gamma distribution for the incoming flow, $\rho \in [0{,}15;\ 0{,}85]$

меньший набор входных данных для обучения, т. е. коэффициент загрузки принимает значения на отрезке [0,1; 0,9] с шагом 0,1, как и раньше, число приборов n от одного до трех, а число подсистем K меняется от двух до 16. Параметр распределения Эрланга определяется выражением $\beta = 2n\rho/b$.

После обучения нейросети строим соответствующие оценки для аналогичных промежуточных значений данных, т. е. для $\rho \in [0{,}15; 0{,}85]$ с шагом 0,1, $K = 2, \ldots, 16$ и $n = 1, 2, 3$. Так, для среднего времени отклика (см. рис. 2, *в*) примерно 98,5 % ошибок не превышает 5 %. Максимальная погрешность приближения $E[R_K]$ не превышает 8 %, что является приемлемым, особенно учитывая, что практически 99 % погрешностей находятся в пределах 5 %, о чем свидетельствуют их низкие средние значения (1,8 %).

Если интервалы между соседними поступлениями заявок имеют гамма-распределение $p(x)$ с параметрами $k = 0{,}25$ (CV = 2) и $\gamma = kn\rho/b$, то после обучения нейросети на тех же входных данных, что и в случае с распределением Эрланга, получим следующий результат для аналогичных значений промежуточных данных. Для среднего времени отклика (см. рис. 2, *г*) примерно 96 % относительных ошибок от их общего количества не превышает 6 %. При этом максимальная относительная погрешность приближения не превышает 10 %.

Если анализировать уровень снижения загрузки в случае выделения дополнительных серверов, то ситуация выглядит следующим образом. Вне зависимости от типа распределения для входящего потока из трех рассмотренных, при наличии в каждом узле двух серверов ($n = 2$) вместо одного происходит снижение среднего времени отклика примерно в два раза, что было ожидаемо. Если число серверов $n = 3$, то среднее время отклика уменьшается в среднем на 65 % по сравнению с $n = 1$. Расчет происходил для уровня высокой загрузки $\rho \in [0{,}6; 0{,}9]$, поскольку именно в этом случае выделение дополнительных ресурсов для повышения качества обслуживания является актуальным вопросом.

Резюмируя, можем сделать следующие выводы. Для 93 % входных данных погрешность аппроксимации среднего времени отклика не превышает 5 %. Поскольку мы рассматривали абсолютные значения относительной ошибки, то компенсация погрешности приближения в размере 5 % может в случае положительных оценок фактически привести к избыточному выделению ресурсов в размере 10 %. Однако по сравнению с принятым избыточным выделением ресурсов в размере 50 % от их общего объема для современных центров обработки данных экономия 40 % является серьезным преимуществом.

Обсуждение

В работе для оценки среднего времени отклика используются НС, однако предложенный подход не ограничивается только НС, можно применять и другие методы машинного обучения: градиентный бустинг, случайные деревья, бэггинг и др. При этом все сложности, связанные с обучением в данном случае НС, отражаются и на результатах прогнозирования. В частности, речь идет о выборе конкретного алгоритма обучения, подборе архитектуры НС, определении количества данных, необходимых для обучения, и пр. Тем не менее результаты численного эксперимента позволяют говорить о приемлемом качестве прогнозирования благодаря доступности широкого инструментария для обучения нейросетей в различных программных средах, что значительно снижает трудности и временные затраты. Кроме того, открываются возможности для оценки других характеристик рассматриваемой системы, например моментов времени отклика более высокого порядка.

Заключение

В работе предложен подход для оценки производительности систем с интенсивным использованием данных. С помощью комбинации имитационного моделирования с НС были получены оценки для среднего времени отклика. Преимущество подхода по сравнению с ранее известными заключается в универсальности, поскольку отсутствуют ограничения на архитектуру систем с параллельной обработкой заявок. Скорость работы обученной НС сопоставима с проведением вычислений по простой аналитической формуле в противовес имеющимся сложным вычислительным алгоритмам. При этом качество аппроксимации является довольно высоким и не зависит от архитектуры используемой математической модели.

Финансовая поддержка

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-29-06043.

Литература

1. **Reinsel D., Gantz J., Rydning J.** *IDC Report: The Digitization of the world from edge to core. IDC white paper*. Framingham, MA, International Data Corporation, 2018. 28 p.

2. **Han L., Ong H. Y.** Parallel data intensive applications using MapReduce: a data mining case study in biomedical sciences. *Cluster Computing*, 2015, vol. 18, pp. 403–418. doi:10.1007/s10586-014-0405-9
3. **Shanthi Thangam M., Vijayalakshmi M.** Data-intensive computation offloading using fog and cloud computing for mobile devices applications. *2018 Intern. Conf. on Smart Systems and Inventive Technology (ICSSIT)*, 2018, pp. 547–550. doi:10.1109/ICSSIT.2018.8748812
4. **Pandey A., Wang S., Calyam P.** Data-intensive workflow execution using distributed compute resources. *2019 IEEE 27th Intern. Conf. on Network Protocols (ICNP)*, 2019, pp. 1–2. doi:10.1109/ICNP.2019.8888119
5. **De Oliveira D. C., Liu J., Pacitti E.** *Data-intensive workflow management: For clouds and data-intensive and scalable computing environments*. Morgan & Claypool Publishers, 2019. 180 p. doi:10.2200/S00915ED1V01Y201904DTM060
6. **Khalid M., Yousaf M. M.** A comparative analysis of big data frameworks: An adoption perspective. *Applied Sciences*, 2021, vol. 11, no. 22, article number: 11033. doi.org/10.3390/app112211033
7. **Alesawi S., Nguyen M., Che H., Singhal A.** Tail latency prediction for datacenter applications in consolidated environments. 2*019 Intern. Conf. on Computing, Networking and Communications (ICNC)*, 2019, pp. 265–269. doi:10.1109/ICCNC.2019.8685505
8. **Nelson R., Tantawi A. N.** Approximate analysis of fork/join synchronization in parallel queues. *IEEE Transactions on Computers*, 1988, vol. 37, pp. 739–743. doi:10.1109/12.2213
9. **Thomasian A.** Analysis of fork/join and related queueing systems. *ACM Computing Surveys (CSUR)*, 2014, vol. 47, pp. 17:1–17:71. doi:10.1145/2628913
10. **Горбунова А. В., Зарядов И. С., Самуйлов К. Е., Сопин Э. С.** Обзор систем параллельной обработки заявок. *Discrete and Continuous Models and Applied Computational Science*, 2017, т. 25, № 4, с. 350–362. doi:10.22363/2312-9735-2017-25-4-350-362
11. **Varma S., Makowski A. M.** Interpolation approximations for symmetric fork-join queues. *Performance Evaluation*, 1994, vol. 20, pp. 245–265. doi:10.1016/0166-5316(94)90016-7
12. **Varki E., Merchant A., Chen H.** *The M/M/1 fork-join queue with variable subtasks*. https://www.cs.unh.edu/~varki/publication/2002-nov-open.pdf. (дата обращения: 24.03.2022).
13. **Zertal S., Harrison P.** Queueing models of RAID systems with maxima of waiting times. *Performance Evaluation*, 2007, vol. 64, pp. 664–689. doi:10.1016/j.peva.2006.11.002
14. **Sethuraman S.** *Analysis of fork-join systems: Network of queues with precedence constraints*. CRC Press, 2022. 104 p.
15. **Qiu Z., Perez J. F., Harrison P. G.** Beyond the mean in fork-join queues: Efficient approximation for response-time tails. *Performance Evaluation*, 2015, vol. 91, pp. 99–116. doi:10.1016/j.peva.2015.06.007
16. **Wang W., Harchol-Balter M., Jiang H., Scheller-Wolf A., Srikant R.** Delay asymptotics and bounds for multitask parallel jobs. *Queueing Systems*, 2019, vol. 91, pp. 207–239. doi:10.1007/s11134-018-09597-5
17. **Gorbunova A. V., Lebedev A. V.** Bivariate distributions of maximum remaining service times in fork-join infinite-server queues. *Problems of Information Transmission*, 2020, vol. 56, no. 1, pp. 73–90. doi:10.1134/S003294602001007X
18. **Nguyen M., Alesawi S., Li N., Che H., Jiang H.** A black-box fork-join latency prediction model for data-intensive applications. *IEEE Transactions on Parallel and Distributed Systems*, 2020, vol. 31, no. 9, pp. 1983–2000. doi:10.1109/TPDS.2020.2982137
19. **Gorbunova A. V., Lebedev A. V.** Response time estimate for a fork-join system with Pareto distributed service time as a model of a cloud computing system using neural networks. *Communications in Computer and Information Science*, 2022, vol. 1552, pp. 318–332. doi:10.1007/978-3-030-97110-6_25
20. **Вишневский В. М., Горбунова А. В.** Применение методов машинного обучения к решению задач теории массового обслуживания. *Информационные технологии и вычислительные системы*, 2021, № 4, с. 70–82. doi:10.14357/20718632210407

UDC 004.032
doi:10.31799/1684-8853-2022-4-12-19

Estimating the response time of a data-intensive computing environment

A. V. Gorbunova[a], PhD, Phys.-Math., Senior Researcher, orcid.org/0000-0002-9183-0426, avgorbunova@list.ru
V. M. Vishnevsky[a], Dr. Sc., Tech., Professor, orcid.org/0000-0001-7373-4847
[a]V. A. Trapeznikov Institute of Control Sciences of RAS, 65, Profsoyuznaya St., 117997, Moscow, Russian Federation

Introduction: The amount of digital data is constantly growing as well as the need for its storage and processing for various purposes. To conduct data analysis, high-performance computing environments associated with parallelization methods, and, accordingly, data-intensive applications are used. The lack of quality tools for evaluating the effectiveness of the process of parallel data processing or tasks leads to excessive allocation of resources. **Purpose:** To develop mathematical models of data-intensive computing environments and methods for their performance analysis, i.e., for estimating the average system response time based on the data on system performance at the level of subtask solving. **Results:** We present a mathematical model of a parallel computing system in the form of a queueing system

with parallel query processing on various architectures, including non-Poisson input flow and non-exponential service times. As a method for analyzing the average response time, we use a combination of simulation modeling with one of the machine learning methods (artificial neural networks). The effectiveness of the method is confirmed by numerical experiments and depends neither on the type of input flow, nor on the type of distribution of query service times, nor on the number of servers in the nodes of the system. The approximation error of the average response time does not exceed 10%, which makes it possible to optimize the generally accepted resource allocation, significantly reducing the amount of the resources. **Practical relevance:** The presented models and the method of their analysis can be used for efficient planning and allocation of resources for data-intensive systems.

Keywords — data-intensive applications, parallel computing, queueing system, average response time, neural networks.

For citation: Gorbunova A. V., Vishnevsky V. M. Estimating the response time of a data-intensive computing environment. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2022, no. 4, pp. 12–19 (In Russian). doi:10.31799/1684-8853-2022-4-12-19

Financial support

The reported study was funded by RFBR, project No. 19-29-06043.

References

1. Reinsel D., Gantz J., Rydning J. *IDC Report: The Digitization of the world from edge to core. IDC white paper*. Framingham, MA, International Data Corporation, 2018. 28 p.
2. Han L., Ong H. Y. Parallel data intensive applications using MapReduce: a data mining case study in biomedical sciences. *Cluster Computing*, 2015, vol. 18, pp. 403–418. doi:10.1007/s10586-014-0405-9
3. Shanthi Thangam M., Vijayalakshmi M. Data-intensive computation offloading using fog and cloud computing for mobile devices applications. *2018 Intern. Conf. on Smart Systems and Inventive Technology (ICSSIT)*, 2018, pp. 547–550. doi:10.1109/ICSSIT.2018.8748812
4. Pandey A., Wang S., Calyam P. Data-intensive workflow execution using distributed compute resources. *2019 IEEE 27th Intern. Conf. on Network Protocols (ICNP)*, 2019, pp. 1–2. doi:10.1109/ICNP.2019.8888119
5. De Oliveira D. C., Liu J., Pacitti E. *Data-intensive workflow management: For clouds and data-intensive and scalable computing environments*. Morgan & Claypool Publishers, 2019. 180 p. doi:10.2200/S00915ED1V01Y201904DTM060
6. Khalid M., Yousaf M. M. A comparative analysis of big data frameworks: An adoption perspective. *Applied Sciences*, 2021, vol. 11, no. 22, article number: 11033. doi.org/10.3390/app112211033
7. Alesawi S., Nguyen M., Che H., Singhal A. Tail latency prediction for datacenter applications in consolidated environments. 2*019 Intern. Conf. on Computing, Networking and Communications (ICNC)*, 2019, pp. 265–269. doi:10.1109/ICCNC.2019.8685505
8. Nelson R., Tantawi A. N. Approximate analysis of fork/join synchronization in parallel queues. *IEEE Transactions on Computers*, 1988, vol. 37, pp. 739–743. doi:10.1109/12.2213
9. Thomasian A. Analysis of fork/join and related queueing systems. *ACM Computing Surveys (CSUR)*, 2014, vol. 47, pp. 17:1–17:71. doi:10.1145/2628913
10. Gorbunova A. V., Zaryadov I. S., Samouylov K. E., Sopin E. S. A survey on queuing systems with parallel serving of customers. *Discrete and Continuous Models and Applied Computational Science*, 2017, vol. 25, no. 4, pp. 350–362 (In Russian). doi:10.22363/2312-9735-2017-25-4-350-362
11. Varma S., Makowski A. M. Interpolation approximations for symmetric fork-join queues. *Performance Evaluation*, 1994, vol. 20, pp. 245–265. doi:10.1016/0166-5316(94)90016-7
12. Varki E., Merchant A., Chen H. *The M/M/1 fork-join queue with variable subtasks*. Available at: https://www.cs.unh.edu/~varki/publication/2002-nov-open.pdf (accessed 24 March 2022).
13. Zertal S., Harrison P. Queueing models of RAID systems with maxima of waiting times. *Performance Evaluation*, 2007, vol. 64, pp. 664–689. doi:10.1016/j.peva.2006.11.002
14. Sethuraman S. *Analysis of fork-join systems: Network of queues with precedence constraints*. CRC Press, 2022. 104 p.
15. Qiu Z., Perez J. F., Harrison P. G. Beyond the mean in fork-join queues: Efficient approximation for response-time tails. *Performance Evaluation*, 2015, vol. 91, pp. 99–116. doi:10.1016/j.peva.2015.06.007
16. Wang W., Harchol-Balter M., Jiang H., Scheller-Wolf A., Srikant R. Delay asymptotics and bounds for multitask parallel jobs. *Queueing Systems*, 2019, vol. 91, pp. 207–239. doi:10.1007/s11134-018-09597-5
17. Gorbunova A. V., Lebedev A. V. Bivariate distributions of maximum remaining service times in fork-join infinite-server queues. *Problems of Information Transmission*, 2020, vol. 56, no. 1, pp. 73–90. doi:10.1134/S003294602001007X
18. Nguyen M., Alesawi S., Li N., Che H., Jiang H. A black-box fork-join latency prediction model for data-intensive applications. *IEEE Transactions on Parallel and Distributed Systems*, 2020, vol. 31, no. 9, pp. 1983–2000. doi:10.1109/TPDS.2020.2982137
19. Gorbunova A. V., Lebedev A. V. Response time estimate for a fork-join system with Pareto distributed service time as a model of a cloud computing system using neural networks. *Communications in Computer and Information Science*, 2022, vol. 1552, pp. 318–332. doi:10.1007/978-3-030-97110-6_25
20. Vishnevsky V. M., Gorbunova A. V. On the application of machine learning methods to solving problems queuing theory. *Journal of Information Technologies and Computing Systems*, 2021, no. 4, pp. 70–82 (In Russian). doi:10.14357/20718632210407