



Международный журнал информационных технологий и энергоэффективности

Сайт журнала: <http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004.89

## ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И ИХ ФУНКЦИОНИРОВАНИЕ В АСПЕКТЕ ПРОГНОЗИРОВАНИЯ ИХ СОСТОЯНИЯ

**Васильев А.В.**

*ФГБУО ВО «МИРЭА - РОССИЙСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ», Москва, Россия, (119454, г. Москва, просп. Вернадского, 78, стр. 4.), e-mail: light7591@gmail.com*

Вычислительные системы все более и более используются во всех сферах повседневной жизни. Для обеспечения бесперебойного их функционирования необходимо знать как принципы их композиции, так и уметь разрабатывать методы, направленные на поддержание их бесперебойной работы. Целью данного исследования является описание принципов функционирования вычислительных систем и методов, которые могут быть направлены на поддержание их бесперебойной работы. В работе использовались общенаучные методы: анализ теоретических источников, сбор информации, описание. Необходимость поддержания бесперебойной работы вычислительной системы очевидна ввиду того, что убытки от её простоя или потерь данных на ней могут быть весьма велики. Методы и средства поддержки вычислительной системы в рабочем состоянии очень важны и в перспективе всегда будут таковыми.

Ключевые слова: Вычислительные системы, программное обеспечение, нейросети, метрики, временные ряды, прогнозирование.

## COMPUTATIONAL SYSTEMS AND THEIR FUNCTIONING IN THE ASPECT OF FORECASTING THEIR CONDITION

**Vasilyev A.V.**

*MIREA - RUSSIAN TECHNOLOGICAL UNIVERSITY, Moscow, Russia (119454, Moscow, avenue. Vernadsky, 78, b. 4), e-mail: light7591@gmail.com*

Computational systems are increasingly being used in all spheres of everyday life. To provide their faultless functionality it is necessary to know the principles of their composition as well as be able to develop methods directed to keeping their faultless work. The purpose of the given research is description of functioning of computational systems and methods that can be directed to provide their faultless work. General scientific methods were used in the work: analysis of theoretical sources, collection of information, description. The necessity of keeping the faultless work of a computational system is obvious due to that the losses of its being idle or its data loss can be quite high. Methods and means of keeping a computational system in working condition are very important and in perspective will be always so.

Keywords: Computational systems, software, neural networks, metrics, time series, forecasting.

### Введение

Вычислительные системы широко применяются в повседневной деятельности для решения самых разнообразных задач. Обычно под вычислительной системой подразумевается ПК (PC, персональный компьютер). В более широком смысле под термином “вычислительная

система” подразумеваются как персональные компьютеры (предназначенные для одного пользователя), так и специализированные компьютеры для решения узкого круга задач (например, мэйнфреймы или суперкомпьютеры). Иногда также используется словосочетание “компьютерная система”[1, 2].

Вычислительные системы (ВС), используемые в настоящее время, выполняют множество задач, не связанных напрямую с вычислением (математикой). Следует, однако, делать различие между понятиями “ЭВМ” (электронно-вычислительная машина) и компьютером. Первое понятие символизирует, как следует из названия, то, что в качестве узлов, исполняющих работу ЭВМ (т.е. её функциональных узлов), используются электронные компоненты. Под понятием же компьютера могут скрываться принципы не обязательно электронного характера – он может быть основан на биологических, механических, квантовых и других принципах. Термин “компьютер” также предполагает возможность внесения поправок (изменений) в программу, исполняемую на нем, что, в свою очередь, доступно не для всех видов ЭВМ.

Существуют математические модели вычислительных систем. Они используются, как правило, для какой-то конкретной цели. К математическим моделям вычислительных систем относятся автомат фон Неймана, абстрактный автомат, конечный автомат (с памятью или без), а также универсальная машина Тьюринга и машина Поста. Целями применения (использования) данных моделей вычислительных систем являются соответственно исследования возможностей создания самовоспроизводящихся машин, построения дискретных моделей более сложных автоматов для работы с символами (последовательностями символов), применение в анализаторах (синтаксических и лексических) и тестирования программного обеспечения на основе моделей (более широко в данных методах используется теория конечных автоматов). Универсальная же машина Тьюринга и машина Поста являются алгоритмически эквивалентными и обе предназначены для формализации самого понятия алгоритма и решения задач, касающихся алгоритмической разрешимости. Математические модели вычислительных систем также могут делиться на подмодели (к примеру, не- и детерминированные конечные автоматы).

Вычислительные системы строятся по специальной архитектуре (например, архитектура фон Неймана, Гарвардская архитектура), которая подвержена изменениям в соответствии с видами задач, решаемых на вычислительной системе, построенной на ее основе. Изменение (оптимизация) архитектуры вычислительной системы осуществляется для наиболее реалистичного математического моделирования исследуемых явлений. Результат задачи, выполняемой на вычислительной системе, предоставляется пользователю посредством устройств вывода, каковыми могут быть мониторы (дисплеи), проекторы, индикаторы и т.п..

Современные вычислительные системы могут быть цифровыми (т.е. предназначенными для работы с дискретными численными (а также символьными) переменными, а также аналоговыми (для работы с потоками данных, поступающими непрерывно). Цифровые компьютеры в настоящее время обладают намного большим диапазоном применения, чем аналоговые (которые используются, как правило, для некоторых специальных целей). Остальные подходы к построению вычислительных систем являются в настоящее время либо используемыми в узком кругу (т.е. нацеленными на узкоспециализированные задачи), либо являются экспериментальными (пробными) решениями.

Цифровые компьютеры основаны на той или иной системе счисления. В настоящее время используется двоичная система счисления, переход к которой позволил значительно упростить внутреннее устройство компьютеров, а также периферийного оборудования (и более упрощенно реализовать арифметические и логические операции в компьютере). В общем и целом выбор системы представления данных не меняет основные принципы, в соответствии с которыми работает компьютер – любой компьютер способен работать в качестве другого (т.е. эмулировать его). Производительность же многих компьютеров, как правило, определяется скоростью, с которой они могут осуществлять чтение (запись) данных в память. Код программы, исполняемой на вычислительной системе, как правило, хранится в той же памяти, что и данные (в соответствии с архитектурой фон Неймана). Данное решение используется, как правило, в большинстве компьютерных систем, в отличие от микро-ЭВМ (контроллеров) и сигнальных процессоров.

Основной (фундаментальной) особенностью компьютеров является способность исполнять некий изменяемый набор инструкций (программу). При помощи вычислений компьютер способен обрабатывать информацию по определенному алгоритму. Как правило, обработка информации при этом сводится к применению простой алгебры логики. Однако следует иметь в виду, что компьютеры могут решить не любую математическую задачу.

Применение компьютеров в настоящее время охватывает как просто вычисления, так и работу с базами данных (СУБД), управление различными устройствами, простое хранение информации и её пересылка по всевозможным каналам связи, работу в Интернете, игры, моделирование сложных биологических, метеорологических процессов (для этого используется специальный класс вычислительных систем – суперкомпьютеры) и многое другое. Широко распространены распределенные вычисления, во время которых множество слабых компьютеров работают над отдельными частями какой-то сложной задачи, а также применения вычислительных систем при работе с искусственным интеллектом (т.е. для решения таких задач, где отсутствует чётко определенный более или менее простой алгоритм). Фактически в настоящее время компьютер превратился в главный информационный инструмент при решении всевозможных задач[3].

## **1. Постановка проблемы**

В процессе эксплуатации вычислительных систем происходят, помимо бесперебойной их работы, сбои, которые влияют на возможность их использования, корректность обработки ими данных и правильность выдаваемых ими результатов. Таким образом, системы и программное обеспечение, способные каким-то образом предсказывать состояние вычислительной системы, являются очень важными объектами как для разработки, так и для исследований в целом.

Простыми методами для того или иного контроля (диагностики) состояния вычислительной системы являются как специальное программное обеспечение, созданное именно для такой цели, так и аппаратные методы, направленные на то же самое. Рассмотрим, к примеру, персональные компьютеры (ПК), как один из классов вычислительных систем. К первой категории можно отнести, к примеру, специальное программное обеспечение для контроля за состоянием ПК (программы, по функциональности похожие на HWMonitor, Reliability Monitor и т.п. а также множество фирменного программного обеспечения,

поставляемого, как правило, с компьютерным оборудованием той или иной компании или же доступного для загрузки с сайта этой же компании). Ко второй же категории можно отнести специальные платы для того или иного разъема на материнской плате ПК или устройства для подсоединения к тому или иному порту ПК. Данные устройства и платы, как правило, снабжены той или иной индикацией или имеют встроенный динамик, что позволяет пользователю проконтролировать правильность загрузки ПК (похожие аппаратные средства уже имеются, как правило, на самой материнской плате для выполнения той же самой функции, и таким образом, в случае выхода их из строя их работу на себя может взять внешнее устройство). Но очевидно, что данный подход хотя и является одним из способов того или иного контроля за ПК, он позволяет лишь оценить то, как успешно загрузилась система и/или наблюдать как за её состоянием в целом, так и за состоянием её отдельных компонентов в настоящий момент, но никак не прогнозировать её состояние, восстанавливать систему в случае сбоя и т.п.. Таким образом, необходим, оправдан, заслуживает внимания и актуален более глубокий подход к данному вопросу.

## **2. Описание изучаемого предмета статьи**

Вычислительные системы всё более и более используются во всех сферах деятельности.

Кратко вычислительную систему можно разделить на[4]:

1. Аппаратное обеспечение – в случае персонального компьютера, например, им являются материнская плата, жесткий диск, процессор и оперативная память. Также в данную категорию могут быть отнесены периферийные устройства и корпус ПК с блоком питания.

2. Программное обеспечение. В данную категорию может быть отнесена операционная система(ОС). Как правило, ею является Windows или Linux. Также в данную категорию могут быть отнесены различные прикладные приложения для работы в указанных ОС.

Анализ соответствующих статей показал, что для прогнозирования (а следовательно, и для предсказания сбоев вычислительной системы) можно задействовать нейросети[5, 6]. Нейросети (NN, НС) построены по принципу биологических нейронных сетей. Каждый нейрон способен передавать информацию другим нейронам через связи. Каждая связь имеет свой вес, что, безусловно, влияет на выходной сигнал. Нейронные сети могут иметь разнообразную архитектуру, что и определяет сферу их применения.

## **3. Цель работы**

Приведение обзора систем и программного обеспечения (с демонстрацией реальных примеров), которое может быть задействовано для цели прогнозирования состояния вычислительной системы.

## **4. Методы исследования**

В работе были использованы такие методы исследования, как анализ научной литературы, электронных ресурсов и анализ авторских статей по теме исследования.

## **5. Результаты исследования**

Путем прогнозирования конечных состояний, получившихся исходов и т.п. может быть задействование искусственных нейронных сетей (ANN, ИНС) [5, 6] (при этом может быть дан прогноз как на состояние вычислительной системы, так и систем, отличных от неё). При этом необходимо задействовать обучение рассматриваемой нейронной сети на поиск сбоев в системе. Прогнозированием состояния системы в данном случае будет являться оценка фактических и аккумулированных показателей работы вычислительной системы, т.е. в данном случае нейронная сеть будет работать с какими-то данными, характеризующими состояние данной системы (показателями данной системы) и производить оценку состояния данной системы (т.е. делать прогноз) на основе этих данных.



Рисунок 1 – Обнаружение сбоев при использовании нейросети

При построении нейронных сетей, направленных на поиск (предсказание) аномалий, происходящих в работе вычислительных систем, широко применяется подход, основанный на предиктивном обслуживании (англ. predictive maintenance) [7, 8]. Сущность рассматриваемого подхода состоит в нахождении (обнаружении) неполадок, способных привести к отказу или серьезному ухудшению работы системы, с использованием (задействованием) искусственного интеллекта. Рассмотрим существующие системы мониторинга состояния вычислительных систем, основываясь на данном подходе.

Вычислительная система, будучи разработанной, прежде всего доводится до стадии эксплуатации. Конечному пользователю крайне необходимо, чтобы рассматриваемая система работала без каких-либо сбоев (либо с возможно минимальным их числом и при этом чтобы данные сбои не влияли на работу системы и были как можно более незаметны для самого конечного пользователя). Для цели предсказания состояния вычислительной системы задействуются определенные программы отслеживания (мониторинга) её состояния. Данные инструменты снимают показания с работающей системы (метрики), предоставляют возможность определения (диагностики) её состояния и нахождения причины произошедших сбоев (если они произошли). Данный процесс имеет название мониторинга системы (программной или аппаратной)[9]. В качестве простого метода, направленного на предотвращение остановки вычислительной системы, дестабилизации ее состояния и т.п. возможно, к примеру, определенным способом или методом настроить данное используемое программное обеспечение таким образом, чтобы оно срабатывало при отклонении какого-либо из наблюдаемых показателей от нормы (например, при уменьшении свободного места на жестком диске системы, за которой осуществляется наблюдение). Обычно средства мониторинга вычислительных систем уже так или иначе включают в себя подобные инструменты, что с успехом позволяет определять уже произошедшие в вычислительной

системе различные сбои, остановки, отклонения от нормы и т.п., или же позволяет определять потенциальные состояния (симптомы) будущих каких-либо неполадок в системе, но в целом и целом предсказание сбоев в вычислительной системе остается для средств мониторинга сложной задачей. Большинство потенциальных отказов системы, как правило, остается незамеченными и поэтому как никогда является актуальным поиск методов, направленных на как можно более успешное определение каких бы то ни было отказов вычислительных систем. Предикативное обслуживание, упомянутое выше, несомненно является одним из таких методов.

Метрики, позволяющие отслеживать состояние рассматриваемой вычислительной системы – это различные показатели системы, а также среды, в которой эксплуатируется данная система. При сборе данных с метрик устанавливается метка момента времени, когда метрики были получены. Данные метрик в анализе называются временными рядами (англ. time series)[10]. С вычислительной системы, находящейся в эксплуатации, возможно снятие тысяч метрик, формирующих пространство метрик (или же временных рядов, которые являются в данном случае многомерными)[11]. В общем же случае понятие временного ряда применимо не только к вычислительным системам.

Реализацией (осуществлением) предикативного обслуживания является задача нахождения аномалий в рассматриваемых временных рядах[12]. Аномалией в данном случае будет являться некое отклонение данных (показателей) программной системы, к примеру, замедление скорости обработки запроса того или иного вида или падение числа обработанных обращений при неизменном количестве (уровне) сессий клиентов, возникающих при обращении к вычислительной системе[13].

Теперь, когда начальные сведения о характеристиках, используемых в тех или иных инструментах, направленных на прогнозирование состояния вычислительной системы, даны, имеет смысл более подробно и по возможности максимально полно перейти к рассмотрению данных инструментов.

Для того, чтобы рассмотреть показатели вычислительной системы в ходе её мониторинга каким-либо программным средством, требуется приложение или приложения, способные обеспечить значительную загрузку данной вычислительной системы в ходе её работы с ним (с ними). В качестве такого приложения, необходимого для демонстрации методов прогнозирования состояния вычислительной системы, может быть рассмотрена программная система “Web-Консолидация”[14] как источник метрик для последующих анализов и пример в целом как приложения, которое способно значительно нагрузить вычислительную систему. Рассматриваемая программная система является определенно сложной и для нее собирается значительное количество метрик, которыми определенно являются показатели операционной системы, под которой собственно выполняется код (к примеру, загрузка CPU в процентах, состояние виртуальной памяти), показатели сети, сервера (опять же загрузка CPU, памяти и т.п.), а также собственные метрики программного комплекса по тем или иным определяющим подсистемам. Для снятия метрик с системы, как правило, используется определенное программное обеспечение, специальным образом созданное именно для таких целей. Одним из таких программных средств является graphite-clickhouse[15].

Данное программное обеспечение может обмениваться данными с пользовательским интерфейсом grafana, используемым для мониторинга состояния вычислительных систем.

Считывание же метрик с рассматриваемого приложения “Web-Консолидация” можно осуществить при помощи отдельного программного обеспечения – jmxtrans[16]. В этом случае схема работы всего программного обеспечения (graphite-clickhouse и jmxtrans) может быть представлена как некий комплекс, работающий согласованно и вместе.

Следует помнить, что пользовательские приложения (как предназначенные для стационарных ПК, так и мобильные), как правило, подвержены частой смене версий, что, безусловно, влияет на метрики данного приложения. Они также могут иметь различные особенности реализации[17], малый (как правило) процент аномалий, пробелы в считывании показателей системы (например, если сервер перегружен, получить от него какую-либо информацию трудно или совсем невозможно[18]). Все это совершенно точно и определенно влияет на обучение нейронной сети, рассматриваемой для прогноза состояния вычислительной системы и/или восстановления текущего состояния вычислительной системы в рабочее.

На исходе (результате) прогноза или ресторации окна потока метрик, актуального в данный момент, высчитывается отклонение от результата, полученного от программной системы, работающей должным образом[20]. В случае, если разница между полученными метриками программной системы и нейронной сети является слишком большой (т.е. в самом общем случае, если разница между моделью и фактическими показателями слишком велика)[20, 21], можно определенно сделать вывод об аномальности (неправильности) отрезка данных, с которым происходит работа в настоящий момент. Для использования нейронных сетей таким образом существует ряд проблем, а именно:

1. В общем случае, требуется обладать актуальной моделью, как для обучения[22, 23, 24] рассматриваемой нейронной сети, так для успешного обнаружения ошибок[25]. Изменения сезонности, направления и т.п. способны вызвать ложные срабатывания. Необходимо четко определенное время, когда используемая модель потеряет свою актуальность.

2. Ни в коем случае нельзя оставлять в стороне поиск и предотвращения ложных срабатываний[25, 26]. Определенно требуется максимально возможная минимизация их количества[27, 28]. В противном случае ложные прогнозы и/или срабатывания будут напрасно отнимать много времени у персонала, осуществляющего надзор за вычислительной системой. Как некоторые из вариантов борьбы с данной проблемой, здесь может помочь введение модели детектирования злоупотреблений[25] или введение некоего порога срабатываний.

Помимо данного программного обеспечения, следует отметить специализированный программный комплекс Prometheus, который непосредственно предназначен для мониторинга состояния как вычислительных систем, так и различных других объектов[29]. Он представляет собой базу данных временных рядов и при этом позволяет присоединение различных инструментов для расширения своего функционала. Prometheus работает посредством извлечения метрик с использованием HTTP-вызовов к конечным точкам, введенным заранее в его конфигурационные файлы. У данного комплекса есть заметное отличие от остальных баз временных рядов, заключающееся в его самостоятельной сборке метрик (технология Pull)[30].

Дать описание абсолютно всех программных комплексов и т.п., имеющих отношение к мониторингу состояния вычислительных систем, не представляется возможным, поэтому приведены лишь некоторые из них и кратко упомянуты технологии, лежащие в их работе.

С целью детектирования аномалий во временных рядах может быть применена рекуррентная нейронная сеть (англ. RNN, Recurrent Neural Network)[31, 32] с памятью LSTM (англ. long short-term memory). Данный тип нейронной сети способен запоминать прошлую информацию и учитывать ее при предсказаний будущих показателей (значений) [33]. Применение данной нейронной сети показано на рисунке 8. Из рисунка видно, что в целом рекуррентная нейронная сеть справляется с поиском аномалии на рассматриваемом участке времени. LSTM-нейронные сети в целом хорошо приспособлены к прогнозированию временных рядов (если это возможно для рассматриваемого ряда) и выдают достаточно достоверные сведения [34]. Но в целом использования одной технологии рекуррентной нейронной сети будет явно недостаточно (иными словами, она применима к малому количеству метрик). Тем не менее, возможно в целом её использование в качестве дополнительного (вспомогательного) средства поиска аномалий.

С целью прогнозирования отказов применяются автоэнкодеры, или автокодировщики (англ. auto-encoders)[35, 36]. По своей сути автокодировщик - искусственная нейронная сеть со специальной архитектурой, позволяющая применить обучение без учителя (англ. unsupervised learning)[36]. Простейшая форма автокодировщика – нейронная сеть, похожая на многослойный перцептрон (англ. multilayer perceptron, MLP) и имеющая входной слой (уровень входа), выходной слой (уровень выхода) и один или несколько слоев, соединяющие их[36, 37]. Фундаментальный же принцип функционирования автокодировщика – получение на выходном слое отклика, максимально похожего на входной[36, 37, 38]. Как правило, для избежания слишком тривиального выполнения данной задачи ее усложняют путем добавления каких-то ограничений относительно промежуточных слоев (меньшая размерность, ограничение активных нейронов промежуточного слоя). Для прогнозирования отказов автокодировщик обучается на нормальных данных и затем находит что-то аномальное в подаваемых данных. При этом автокодировщик обучается осуществлять реконструкцию, в отличие от MLP, своих собственных значений  $X$  (входное значение), а не значений  $Y$  (выходное), заданных входом  $X$ [39], т.е. в данном случае как раз и происходит прогнозирование отказов (аномалий).

Как упоминалось выше, необходимо всяческим образом предотвращать ложные срабатывания комплекса мониторинга вычислительной системы [27, 28]. Обычно механизм сокращения ложных срабатываний основан на неких паттернах, классифицируемых администратором системы и обнаруженными, как правило, при помощи нейросетей. Основной принцип сокращения ложных срабатываний в данном случае заключается в сравнении случая, возникшего в ходе работы вычислительной системы, с паттернами, собранными и классифицированными администратором, и на основе сравнения определяется принадлежность данного случая к ложным или приводящим к сбоям в работе вычислительной системы. Для сравнения двух соответствующих временных рядов может быть применён алгоритм динамической трансформации временной шкалы (англ. dynamic time warping, DTW)[40]. В совокупности с нейронными сетями LSTM работа всей схемы прогнозирования



(предсказания) отказов теоретически может быть представлена в виде схемы, показанной на рисунке 2.

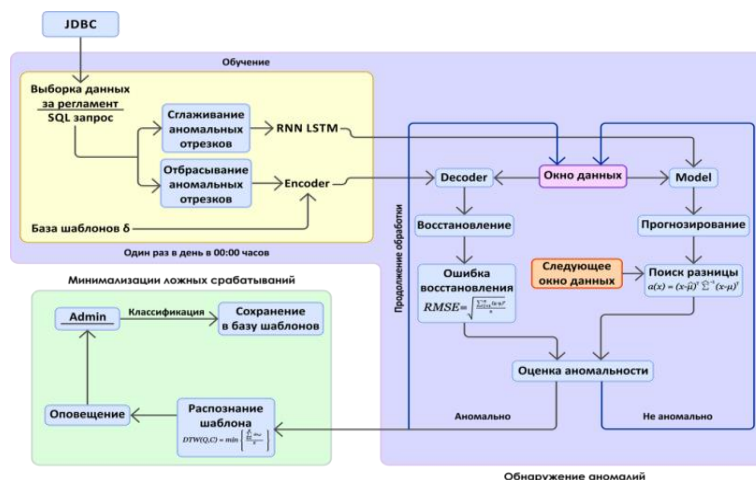


Рисунок 2 – Анализ метрик и схема предсказания отказов на его основе.

На данной схеме расположены несколько блоков (поиск аномальных отрезков, или сегментов, в поступающих данных мониторинга вычислительной системы, т.е. метриках, а также, собственно, и сам механизм минимизации ложных срабатываний). Данные могут подаваться из некоторой базы данных, в которую их сохранит graphite (или другое средство мониторинга вычислительной системы). Примеры работы всего комплекса (примеры аномальных метрик) во время функционирования вычислительной системы показаны на рисунке 3. Порог аномальности (процент, выше которого загрузка CPU считается чрезмерной) при этом выделен экспериментально администратором. Порог аномальности может быть легко отрегулирован в соответствии с требованиями к устойчивости вычислительной системы.



Рисунок 3 – Пример работы автокодировщика.

Из рисунка видно, что метрика (временной ряд) находилась на одном и очень низком уровне долгое время. Автокодировщиком было отнесено данное состояние вычислительной системы в аномальные.

До сих пор подходы к обеспечению работоспособности вычислительной системы рассматривались с точки зрения оценивания показаний её функционирования и последующего принятия решений в соответствии с данными показателями, задействуя при этом различные

инструменты для этого (средства мониторинга вычислительной системы, сборщики метрик, нейросети, методы анализа информации и т.д.). Следует, тем не менее, упомянуть, что как минимум средства мониторинга могут быть встроены в крупные программные комплексы, осуществляющие, к примеру, контроль за параллельными вычислениями, что исключает необходимость пользоваться сторонними программами. Таким образом, применение дополнительного программного обеспечения, направленного на поддержание работы нескольких вычислительных систем, выполняющих параллельные вычисления с помощью рассматриваемых комплексов, в той или иной степени теряет смысл (это определяется лишь тем, насколько эффективные средства для поддержки работы вычислительных систем, работающих параллельно, встроил в данный комплекс его разработчик). Самым очевидным примером, хотя и не связанным с параллельными вычислениями, является операционная система Windows и её диспетчер задач (англ. task manager), позволяющий осуществлять, хотя и очень ограниченно, мониторинг компьютера, на котором установлена сама Windows.

## **6. Выводы**

Таким образом, существующие системы (аппаратные и программные), направленные на прогнозирование состояния вычислительной системы, можно приближённо разделить на следующие условные категории:

1. Аппаратные средства (специализированные платы с индикацией, динамиком и т.п., возможно, какие-либо внешние средства). Иногда примитивные системы, осуществляющие, к примеру, информирование пользователя о правильности загрузки компьютера, могут быть встроены на материнской плате этого компьютера.

2. Программы для мониторинга состояния вычислительной системы на основе каких-то показателей (температура CPU, загрузка подсистем и т.п). В эту категорию входят, например, Task Manager, grafana, HWMonitor, Reliability Monitor.

3. Программы, необходимые для снятия метрик с функционирующей вычислительной системы и работы с ними (graphite-clickhouse, jmxtrans), специализированные программные комплексы, направленные на это (Prometheus).

4. Нейросети и технологии, связанные с ними (RNN, LSTM, un- и supervised learning, dynamic time warping, predictive maintenance, автокодировщики).

## **Список литературы**

1. Essentials of computer architecture. Douglas Comer. Second edition. Chapman and Hall/CRC. 2017. – p.511
2. Introduction to computing systems. From bits & gates to C/C++ and beyond. Yale Patt, Sanjay Patel. Third edition. McGraw-Hill Higher Education. 2019. – 800 p.
3. “Quantum computing for everyone”. Chris Bernhardt. The MIT Press. 2020. – p.216
4. Безнос О.С. Конфигурации, выбор и обоснование персонального компьютера целевого назначения/ Фундаментальная и прикладная наука: новые вызовы и прорывы/Безнос О.С., Притыка М.Ю.-Сборник статей Международной научно-практической конференции. 2020.

5. Любимова Т.В., Горелова А.В. Решение задачи прогнозирования с помощью нейронных сетей/Любимова Т.В., Горелова А.В. Международный научный журнал “Инновационная наука”, 2015.
6. Angela Meyer Early fault detection with multi-target neural networks/Angela Meyer -Bern University of Applied Sciences. arXiv:2106.08957. 2021.
7. Amruthnath Nagdev, Gupta Tarun. A research study on unsupervised machine learning algorithms for fault detection in predictive maintenance/Amruthnath Nagdev, Gupta Tarun. – ResearchGate.net, April 2018.
8. Susto, Gian Antonio. Machine learning for predictive maintenance: A multiple classifier approach/ IEEE Transactions on industrial informatics. 11 (3); 812-820. Susto, Gian Antonio, 2015.
9. Шамин И.М. Мониторинг IT-систем и сетевых устройств/Шамин И.М.\\ Международный студенческий научный вестник, 2019.
10. Евстигнеев И.П. Прогнозирование временных рядов/ Научная электронная библиотека elibrary.ru. eLibrary ID: 49338123, Евстигнеев И.П. 2022.
11. David Betancourt. Deep learning for high-dimensional time series/Towards Data Science, David Betancourt, Ph.D. 2019.
12. Колесников И.Н., Финогеев А.Г. Проактивный мониторинг событий на основе предиктивного анализа временных рядов/Cyberleninka.ru. Колесников И.Н., Финогеев А.Г., 2020.
13. Alexander Geiger, Dongyu Liu, Sarah Alnegheimish, Alfredo Cuesta-Infante, Kalyan Veeramachaneni. TadGAN: Time series anomaly detection using generative adversarial networks/ Alexander Geiger, Dongyu Liu, Sarah Alnegheimish, Alfredo Cuesta-Infante, Kalyan Veeramachaneni. arXiv.org, 2020.
14. <http://www.krista.ru/products/webconsolidation>. НПО “Криста”. ПК “Web Консолидация”/. НПО “Криста”. Электронный ресурс. 2022.
15. <http://www.github.com/go-graphite/graphite-clickhouse>. Graphite cluster backend with ClickHouse support. Электронный ресурс. 2022.
16. <http://www.github.com/jmxtrans>. Metrics exporter. Электронный ресурс. 2022.
17. Md Saidur Rahman, Foutse Khomh, Alaleh Hamidi, Jinghui Cheng, Giuliano Antoniol, Hironori Washizaki. Machine learning application development:practitioners’ insights/ Md Saidur Rahman, Foutse Khomh, Alaleh Hamidi, Jinghui Cheng, Giuliano Antoniol, Hironori Washizaki. arXiv.org, 2021.
18. Lamisha Rawshan, Jobaer Khan, Asif Imran . Identifying overloaded servers and managing dynamic placement of virtual machines in cloud/ Lamisha Rawshan, Jobaer Khan, Asif Imran. Researchgate.net, 2016.
19. В. Mehlig. Machine learning with neural networks/ В. Mehlig. arXiv.org. 2019.
20. Савицкий Д.Е., Дунаев М.Е., Зайцев К.С. Выявление аномалий при обработке потоков данных в реальном времени/Савицкий Д.Е., Дунаев М.Е., Зайцев К.С.. Cyberleninka.ru, 2022.
21. Саенко И.Б., Котенко И.В., Аль-Барри М.Х. Применение искусственных нейронных сетей для выявления аномального поведения пользователей центров обработки данных/ Саенко И.Б., Котенко И.В., Аль-Барри М.Х.. Cyberleninka.ru. 2022.

22. Stefano Mangini, Francesco Tacchino, Dario Gerace, Daniele Bajoni, Chiara Macchiavello. Quantum computing models for artificial neural networks/ Stefano Mangini, Francesco Tacchino, Dario Gerace, Daniele Bajoni, Chiara Macchiavello. arXiv.org, 2021.
23. Nikolaus Kriegeskorte, Tal Golan. Neural network models and deep learning—a primer for biologists/ Nikolaus Kriegeskorte, Tal Golan. arXiv.org. 2019.
24. Patrick L. Combettes, Jean-Christophe Pesquet, Audrey Repetti. A variational inequality model for learning neural networks/ Patrick L. Combettes, Jean-Christophe Pesquet, Audrey Repetti. arXiv.org, 2022.
25. Aditya Pandey, Abhishek Sinha, Aishwarya PS. Intrusion detection using sequential hybrid model/ Aditya Pandey, Abhishek Sinha, Aishwarya PS. arXiv.org, 2019.
26. Зуев В.Н. Обнаружение аномалий сетевого трафика методом глубокого обучения/ Зуев В.Н. cyberleninka.ru, 2020.
27. Christian Tomani, Florian Buettner. Towards trustworthy predictions from deep neural networks with fast adversarial calibration/ Christian Tomani, Florian Buettner. arXiv.org, 2020.
28. Xu Ji, Razvan Pascanu, Devon Hjelm, Andrea Vedaldi, Balaji Lakshminarayanan, Yoshua Bengio. Predicting unreliable predictions by shattering a neural network/ Xu Ji, Razvan Pascanu, Devon Hjelm, Andrea Vedaldi, Balaji Lakshminarayanan, Yoshua Bengio. arXiv.org, 2021.
29. <http://www.prometheus.io>. Prometheus - from metrics to insight. Электронный ресурс. 2022.
30. Xunhui Zhang, Yue Yu, Georgios Gousios, Ayushi Rastogi. Pull request decision explained: An empirical overview/ Xunhui Zhang, Yue Yu, Georgios Gousios, Ayushi Rastogi. arXiv.org, 2021.
31. Gabor Petnehazi. Recurrent neural network for time series forecasting/ Gabor Petnehazi, arXiv.org, 2019.
32. Hansika Hewamalage, Cristoph Bergmeir, Kasun Bandara. Recurrent neural network for time series forecasting: current status and future directions”. Hansika Hewamalage, Cristoph Bergmeir, Kasun Bandara. arXiv.org, 2019.
33. Malhotra Pankaj, Vig Lovekesh, Shroff Gautam, Agarwal Puneet. Long short-term memory networks for anomaly detection in time series/ Malhotra Pankaj, Vig Lovekesh, Shroff Gautam, Agarwal Puneet. ESANN, researchgate.net, 2015.
34. Обрубов М.О., Кириллова С.Ю. Применение LSTM-сети в решении задачи прогнозирования многомерных временных рядов/ Обрубов М.О., Кириллова С.Ю.. Cyberleninka.ru, 2021.
35. Chong Zhou, Randy C. Paffenroth. Anomaly detection with robust deep autoencoders/ Chong Zhou, Randy C. Paffenroth. Researchgate.net, 2017.
36. Umberto Michelucci. An introduction to autoencoders/ Umberto Michelucci. Researchgate.net, 2022.
37. Dor Bank, Noam Koenigstein, Raja Giryes. Autoencoders/ Dor Bank, Noam Koenigstein, Raja Giryes. arXiv.org, 2020.
38. Sai Krishna, Thulasi Tholeti, Sheetal Kalyani. How to boost autoencoders?/ Sai Krishna, Thulasi Tholeti, Sheetal Kalyani. arXiv.org, 2021.

39. Гурина А.О., Гузев О.Ю., Елисеев В.Л. Обнаружение аномальных событий на хосте при помощи автокодировщика/Гурина А.О.,Гузев О.Ю.,Елисеев В.Л.. International journal of open information technologies,ISSN: 2307-8162, vol.8, no.8, Cyberleninka.ru, 2020.
40. Vivek Mahato, Pdraig Cunningham. A case-study on the impact of dynamic time warping in time series regression/ Vivek Mahato, Pdraig Cunningham. arXiv.org, 2020.

## References

1. Essentials of computer architecture. Douglas Comer. Second edition. Chapman and Hall/CRC. 2017. – p.511
2. Introduction to computing systems. From bits & gates to C/C++ and beyond. Yale Patt, Sanjay Patel. Third edition. McGraw-Hill Higher Education. 2019. – 800 p.
3. “Quantum computing for everyone”. Chris Bernhardt. The MIT Press. 2020. – p.216
4. Beznos O.S. Configurations, Choice and Justification of a Personal Computer of Target Purpose / Fundamental and Applied Science: New Challenges and Breakthroughs / Beznos O.S., Prityka M.Y. - Collection of Articles of the International Scientific and Practical Conference. 2020.
5. Lyubimova T.V., Gorelova A.V. Solving the Forecasting Problem with the Help of Neural Networks / Lyubimova T.V., Gorelova A.V. International Scientific Journal "Innovative Science", 2015.
6. Angela Meyer Early fault detection with multi-target neural networks/Angela Meyer -Bern University of Applied Sciences. arXiv:2106.08957. 2021.
7. Amruthnath Nagdev, Gupta Tarun. A research study on unsupervised machine learning algorithms for fault detection in predictive maintenance/Amruthnath Nagdev, Gupta Tarun. – ResearchGate.net, April 2018.
8. Susto, Gian Antonio. Machine learning for predictive maintenance: A multiple classifier approach/ IEEE Transactions on industrial informatics. 11 (3); 812-820. Susto, Gian Antonio, 2015.
9. Shamin I.M. Monitoring of IT systems and network devices / Shamin I.M.\\ International Student Scientific Bulletin, 2019.
10. Evstigneev I.P. Forecasting of Time Series/ Scientific Electronic Library elibrary.ru. eLibrary ID: 49338123, Evstigneev I.P. 2022.
11. David Betancourt. Deep learning for high-dimensional time series/Towards Data Science, David Betancourt, Ph.D. 2019.
12. Kolesnikov I.N., Finogeev A.G. Proactive monitoring of events based on predictive analysis of time series/Cyberleninka.ru. Kolesnikov I.N., Finogeev A.G., 2020.
13. Alexander Geiger, Dongyu Liu, Sarah Alnegheimish, Alfredo Cuesta-Infante, Kalyan Veeramachaneni. TadGAN: Time series anomaly detection using generative adversarial networks/ Alexander Geiger, Dongyu Liu, Sarah Alnegheimish, Alfredo Cuesta-Infante, Kalyan Veeramachaneni. arXiv.org, 2020.
14. <http://www.krista.ru/products/webconsolidation>. NGO "Krista". PC "Web Consolidation"/. NGO "Krista". Electronic resource. 2022.
15. <http://www.github.com/go-graphite/graphite-clickhouse>. Graphite cluster backend with ClickHouse support. Electronic resource. 2022.
16. <http://www.github.com/jmxtrans>. Metrics exporter. Electronic resource. 2022.

17. Md Saidur Rahman, Foutse Khomh, Alaleh Hamidi, Jinghui Cheng, Giuliano Antoniol, Hironori Washizaki. Machine learning application development: practitioners' insights/ Md Saidur Rahman, Foutse Khomh, Alaleh Hamidi, Jinghui Cheng, Giuliano Antoniol, Hironori Washizaki. arXiv.org, 2021.
18. Lamisha Rawshan, Jobaer Khan, Asif Imran . Identifying overloaded servers and managing dynamic placement of virtual machines in cloud/ Lamisha Rawshan, Jobaer Khan, Asif Imran. Researchgate.net, 2016.
19. B. Mehlig. Machine learning with neural networks/ B. Mehlig. arXiv.org. 2019.
20. Savitskiy D.E., Dunaev M.E., Zaitsev K.S. Detection of anomalies in processing data flows in real time/Savitskiy D.E., Dunaev M.E., Zaitsev K.S. Cyberleninka.ru, 2022.
21. Sayenko I.B., Kotenko I.V., Al-Barry M.Kh. Application of artificial neural networks to detect anomalous behavior of data center users/ Sayenko I.B., Kotenko I.V., Al-Barry M.Kh. Su berleninka.ru. 2022.
22. Stefano Mangini, Francesco Tacchino, Dario Gerace, Daniele Bajoni, Chiara Macchiavello. Quantum computing models for artificial neural networks/ Stefano Mangini, Francesco Tacchino, Dario Gerace, Daniele Bajoni, Chiara Macchiavello. arXiv.org, 2021.
23. Nikolaus Kriegeskorte, Tal Golan. Neural network models and deep learning—a primer for biologists/ Nikolaus Kriegeskorte, Tal Golan. arXiv.org. 2019.
24. Patrick L. Combettes, Jean-Christophe Pesquet, Audrey Repetti. A variational inequality model for learning neural networks/ Patrick L. Combettes, Jean-Christophe Pesquet, Audrey Repetti. arXiv.org, 2022.
25. Aditya Pandey, Abhishek Sinha, Aishwarya PS. Intrusion detection using sequential hybrid model/ Aditya Pandey, Abhishek Sinha, Aishwarya PS. arXiv.org, 2019.
26. Zuev V.N. Detection of network traffic anomalies by the method of deep learning/ Zuev V.N. cyberleninka.ru, 2020.
27. Christian Tomani, Florian Buettner. Towards trustworthy predictions from deep neural networks with fast adversarial calibration/ Christian Tomani, Florian Buettner. arXiv.org, 2020.
28. Xu Ji, Razvan Pascanu, Devon Hjelm, Andrea Vedaldi, Balaji Lakshminarayanan, Yoshua Bengio. Predicting unreliable predictions by shattering a neural network/ Xu Ji, Razvan Pascanu, Devon Hjelm, Andrea Vedaldi, Balaji Lakshminarayanan, Yoshua Bengio. arXiv.org, 2021.
29. <http://www.prometheus.io>. Prometheus - from metrics to insight. Электронный ресурс. 2022.
30. Xunhui Zhang, Yue Yu, Georgios Gousios, Ayushi Rastogi. Pull request decision explained: An empirical overview/ Xunhui Zhang, Yue Yu, Georgios Gousios, Ayushi Rastogi. arXiv.org, 2021.
31. Gabor Petnehazi. Recurrent neural network for time series forecasting/ Gabor Petnehazi, arXiv.org, 2019.
32. Hansika Hewamalage, Cristoph Bergmeir, Kasun Bandara. Recurrent neural network for time series forecasting: current status and future directions". Hansika Hewamalage, Cristoph Bergmeir, Kasun Bandara. arXiv.org, 2019.
33. Malhotra Pankaj, Vig Lovekesh, Shroff Gautam, Agarwal Puneet. Long short-term memory networks for anomaly detection in time series/ Malhotra Pankaj, Vig Lovekesh, Shroff Gautam, Agarwal Puneet. ESANN, researchgate.net, 2015.

34. Obrubov M.O., Kirillova S.Y. Application of the LSTM network in solving the problem of forecasting multidimensional time series. Cyberleninka.ru, 2021.
  35. Chong Zhou, Randy C. Paffenroth. Anomaly detection with robust deep autoencoders/ Chong Zhou, Randy C. Paffenroth. Researchgate.net, 2017.
  36. Umberto Michelucci. An introduction to autoencoders/ Umberto Michelucci. Researchgate.net, 2022.
  37. Dor Bank, Noam Koenigstein, Raja Giryes. Autoencoders/Dor Bank, Noam Koenigstein, Raja Giryes. arXiv.org, 2020.
  38. Sai Krishna, Thulasi Tholeti, Sheetal Kalyani. How to boost autoencoders?/ Sai Krishna, Thulasi Tholeti, Sheetal Kalyani. arXiv.org, 2021.
  39. Gurina A.O., Guzev O.Yu., Eliseev V.L. Detection of anomalous events on the host using an autoencoder/Gurina A.O., Guzev O.Yu., Eliseev V.L. International journal of open information technologies, ISSN: 2307-8162, vol.8, no.8, Cyberleninka.ru, 2020.
  40. Vivek Mahato, Pdraig Cunningham. A case-study on the impact of dynamic time warping in time series regression/ Vivek Mahato, Pdraig Cunningham. arXiv.org, 2020.
-