Прогнозирование загрузки кластерной системы с использованием адаптивных смесей моделей

Ю.С. Артамонов

***Аннотация* — В этой работе мы рассматриваем подход к выбору высокопроизводительного окружения на основе прогнозирования загрузки узлов кластера. Цель работы – исследовать различные модели прогнозирования применительно к задаче прогнозирования загрузки кластера, выбрать наиболее удачные их конфигурации и выяснить, как эффективно применять эти модели совместно.**

В статье приведены результаты сравнения моделей EMMSP, моделей на основе нейронных сетей и адаптивных моделей для решения задачи прогнозирования загрузки узлов кластера. Рассмотрены параметры нейросетевых моделей: выбор функций активации, алгоритмов инициализации и обновления весов нейронов, а также кодирование дополнительных признаков для обучения сети на основе данных о дате и времени. Выполнено тестирование моделей адаптивной селекции и адаптивной композиции и продемонстрировано улучшение результатов прогнозирования по сравнению с моделями, на основе которых они построены. Обучение и тестирование моделей проведено на наборе данных загрузки кластера «Сергей Королев» за период с ноября 2013 года по декабрь 2016 года.

***Ключевые слова* — адаптивная композиция, адаптивная селекция, загрузка ресурсов, кластер, нейронная сеть, прогнозирование.**

I. Введение

В последнее время многие исследования посвящены прогнозированию загрузки различных вычислительных ресурсов, таких как ядра CPU [1], отдельные узлы кластера или облака [2]. Задача прогнозирования загрузки вычислительных ресурсов актуальна, от её эффективного решения зависит то, насколько быстро будут получены результаты вычислительных экспериментов, как будут использоваться ресурсы в будущем, а также как будут запланированы периоды их обслуживания и модернизации. Без прогнозирования доступности разделяемых ресурсов невозможно эффективное использование, например, кластерных систем, где пользователи кластера совместно используют узлы с различными характеристиками, занимая их частично или полностью своими вычислениями.

В работе [3] мы протестировали модель EMMSP для задачи прогнозирования загрузки вычислительных ресурсов и сделали выводы о применимости этой модели. Модель показала себя недостаточно хорошо, поскольку давала хороший прогноз только на специфических данных и участках истории загрузки, но вместе с тем мы продемонстрировали, что она может быть эффективно использована в простейших моделях адаптивной композиции с другими моделями. В этой работе рассматривается использование нейросетевых моделей для прогнозирования загрузки ресурсов кластера, сравнивается этот подход с продемонстрированным ранее и исследуется комбинация модели EMMSP и нейросетевой модели в виде моделей адаптивной селекции и адаптивной композиции.

II. Нейросетевые модели прогнозирования

Нейросетевые модели прогнозирования базируются на использовании нейронных сетей, которые можно обучить для решения задачи регрессии, где на основании некоторых входных параметров они должны выдать значение на выходе, аппроксимируя неизвестную функциональную зависимость выходных данных от входных.

Нейросетевые модели были использованы в работах [4] и [5] для прогнозирования загрузки различных по своей природе ресурсов: CPU серверов и электрических сетей. В обеих задачах нейросетевые модели показали хорошие результаты и были признаны эффективными и адекватными задаче прогнозирования. Учитывая эти результаты, рассмотрим, насколько хорошо нейросетевые модели подойдут для прогнозирования количества загруженных узлов кластера.

Принимая во внимание работы [6] и [7], применяющие нейронные сети для решения задач прогноза, мы выбрали для исследования модель многослойного персептрона (MLP) с двумя скрытыми слоями, для которой рассмотрели две конфигурации признаков: с дополнительными признаками даты и времени (T-DL MLP) и без них (DL MLP).

Структура MLP состоит из нейронов и связей между ними (рис. 1). Нейроны имеют заданную функцию преобразования – активации, а связи – веса. В этой работе мы использовали в качестве функции активации нейронов скрытого слоя функцию гиперболического тангенса.

Статья получена 21 марта 2017 г.

Ю.С. Артамонов, ассистент кафедры информационных систем и технологий Самарского университета (email: artamonov@about.me)

Результаты работы получены при частичной поддержке Программы Минобрнауки России по государственным заданиям высшим учебным заведениям и научным организациям в сфере научной деятельности (в рамках конкурса научных проектов, выполняемых научными коллективами исследовательских центров и (или) научных лабораторий образовательных организаций высшего образования) - проект № 9.1616.2017/ПЧ.

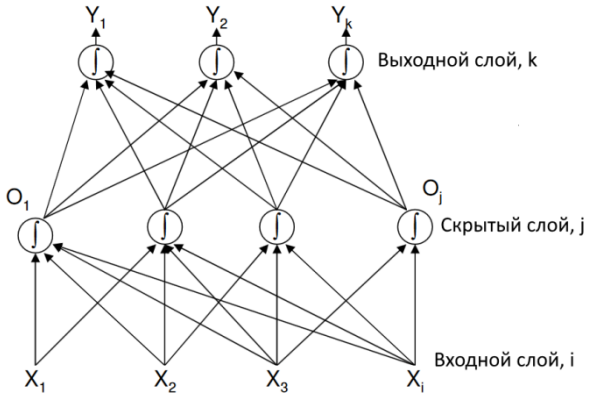

Рис. 1. Структура MLP с одним скрытым слоем

Обучение нейронной сети заключается в изменении весов связей между нейронами, и задача алгоритма обучения состоит в нахождении такой конфигурации весов всех связей, где будет минимизирована функция ошибки. В поставленной задаче прогнозирования используется функция ошибки – MSE (Mean Squared Error) в методе градиентного спуска, в качестве итоговой ошибки для сравнения моделей используется MAE (Mean Average Error). Использование ошибки MAPE невозможно, поскольку временные ряды включают в себя большое количество значений равных или близких 0.

Для обучения и тестирования моделей на основе MLP мы используем библиотеку DeepLearning4j, которая предоставляет инструментарий для работы с нейронными сетями различных конфигураций. В состав этой библиотеки включены наиболее популярные архитектуры нейронных сетей, алгоритмы обучения и оптимизации. Библиотека написана на языке Java и использует нативные расширения для вычислений на CPU и GPU, чтобы обеспечить высокую производительность [9]. Библиотека DeepLearning4j распространяется на условиях лицензии Apache License 2.0, что позволяет применять её в любых приложениях, в том числе коммерческих, а открытый исходный код позволяет привлечь большое количество исследователей и улучшить качество библиотеки.

III. Применение нейросетевых моделей прогнозирования

Для обучения нейронных сетей семейства MLP используется метод обратного распространения ошибки (backpropagation) с различными модификациями. Метод представляет собой итеративный градиентный алгоритм, который применяется для минимизации ошибки работы MLP и получения желаемых выходных значений. Суть метода заключается в распространении сигналов ошибки от выходов сети к её входам, обратно прямому распространению сигналов в обычном режиме работы [10].

Существенными параметрами метода и его модификаций являются:

- количество эпох обучения,
- коэффициент обучения,
- алгоритм инициализации весов связей,
- алгоритм обновления весов,
- алгоритм оптимизации,
- момент обучения.

Для сравнения методов обучения с различными модификациями нами были подобраны параметры обучения, представленные в таблице 1. Мы сравнивали обучение моделей T-DL MLP и DL MLP на задаче прогнозирования 12 точек загрузки кластера (1 точка – средняя загрузка группы узлов кластера за 1 час). При обучении и прогнозировании на вход нейронным сетям подавались только данные временных рядов. Оптимальное количество входов, подобранное экспериментально, равно 9 для DL MLP и для T-DL MLP с учётом дополнительных признаков равно 13.

Таблица 1. Экспериментально подобранные параметры обучения

	DL MLP	T-DL MLP
Количество эпох обучения	350	400
Коэффициент обучения	0,01	0,01
Момент обучения	0,9	0,9
Количество нейронов входного слоя	9	9 + 4
Количество нейронов скрытых слоёв	1ый: 20 2ой: 15	1ый: 20 2ой: 15

В процессе обучения на вход нейронной сети DL MLP подавались всевозможные наборы последовательных 9 значений ряда, при этом тестовые наборы шли в случайном порядке. Для каждого тестового набора на выходе нейронной сети формировалось 12 значений, которые сравнивались с 12 значениями из тестового набора. Параметры i = 9, k = 12. Для модели T-DL MLP кроме значений ряда нами были выделены 4 дополнительных признака на основе средней загрузки кластера с привязкой ко времени и дате: 2 признака, характеризующих час, с которого выполняется прогнозирование, и 2 признака, характеризующие день недели, в который выполняется прогноз.

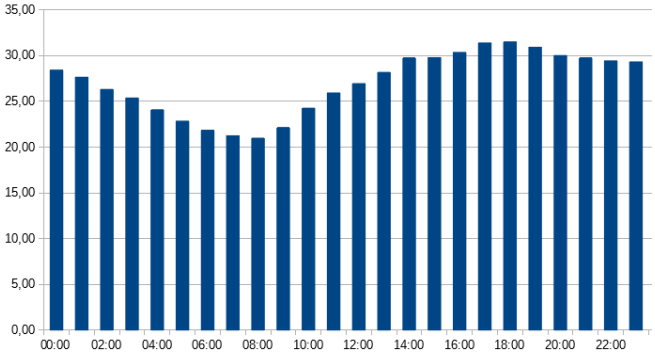

Рис. 2. Средняя загрузка кластера по часам (группы узлов QDR_TMP)

На рис. 2 представлен график средней загрузки кластера по часам, где видно, что загрузка возрастает на отрезке от 06:00 до 18:00 и затем спадает с 18:00 до 06:00. Такое распределение средней загрузки по часам позволило нам выделить 2 дополнительных признака для модели прогнозирования T-DL MLP: 1 – значение средней загрузки на момент, с которого требуется

выполнить прогноз, 2 – значение производной средней загрузки в этот момент. Эти два дополнительных признака однозначно описывают время, в которое требуется выполнить прогноз.

На рис. 3 представлен график средней загрузки кластера по дням недели, где мы можем видеть, что средняя загрузка принимает наименьшее значение в воскресенье и возрастает до своего максимального значения в среду. На основе этих данных мы можем выделить ещё 2 признака для модели T-DL MLP: 1 – значение средней загрузки в день недели, с которого требуется выполнить прогноз, 2 – значение производной средней загрузки в этот день. Два этих признака однозначно описывают день недели, с которого требуется выполнить прогноз.

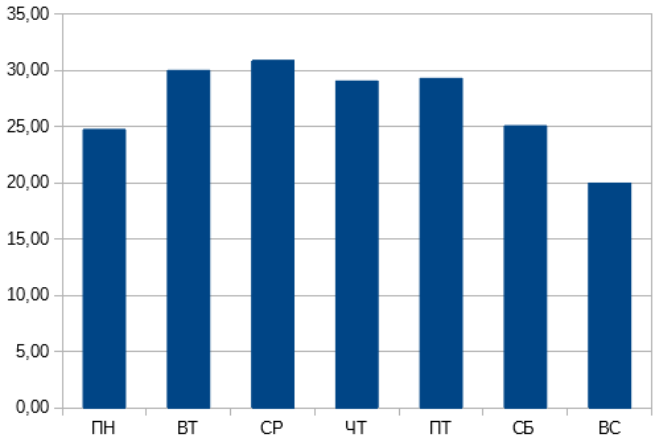

Рис. 3. Средняя загрузка кластера по дням недели (группы узлов QDR_TMP)

Пример таблицы кодирования дней недели приведён в таблице 2, где в колонках с кодами признаков приведены значения дополнительных признаков нормированные в диапазоне [-1, 1]. Таблицы кодирования времени и дня недели вычисляются один раз и используются для обучения модели и прогнозирования, их не требуется вычислять заново при последующем прогнозировании той же моделью, поскольку средние значения загрузки по времени и дню недели изменяются медленно и незначительно.

Таблица 2. Дополнительные признаки для модели T-DL MLP по дню недели (для группы узлов QDR_TMP)

	ПН	ВТ	СР	ЧТ	ПТ	СБ	ВС
Средняя загрузка	24,76	30,01	30,91	29,05	29,29	25,09	19,98
Код значения	-0,12	0,83	1	0,66	0,7	-0,07	-1
Код значения производной от загрузки	0,91	1	0,16	-0.37	0,03	-0,83	-1

Метод обратного распространения ошибки подвержен следующим проблемам:

- медленная сходимость,
- сходимость к локальным минимумам,
- переобучение.

Модификации метода обратного распространения ошибки с моментами и различными алгоритмами обновления весов связей, такими как Adadelta, позволяют бороться с приведёнными выше проблемами, ускорить обучение и уменьшить ошибку работы MLP.

В рамках исследования нейросетевых моделей мы рассмотрели различные конфигурации обучения нейронных сетей методом обратного распространения ошибки. В качестве параметров конфигурации в обучении выступали: алгоритм инициализации весов нейронов, алгоритм обновления весов нейронов и алгоритм оптимизации.

Мы протестировали 2 варианта инициализации весов: равномерным распределением (Uniform) и по методу Xavier. В качестве алгоритма обновления весов были протестированы: алгоритм Nesterov Accelerated Gradient (Nesterovs) [11] и адаптивная оценка моментов (Adam) [12]. Были опробованы 2 алгоритма оптимизации: линейный градиентный спуск (LGD) и стохастический градиентный спуск (SGD). Эти оптимизации и параметры метода градиентного спуска и алгоритма обратного распространения ошибки описаны в работе [13].

В тесте использовалась обучающая выборка длины 6000 точек и тестовая с длиной 1000 точек, данные выборки получены за период с 1 января 2015 года по 1 января 2016 года. Результаты тестирования моделей с различными параметрами обучения для решения задачи прогнозирования загрузки кластера представлены в таблице 2, значения ошибки RMSE (Root Mean Square Error) приведены для оценки разброса прогнозных значений.

В таблице 3 представлены результаты тестирования модификаций метода обратного распространения ошибки, 2 лучших результата для каждой модели выделены подчёркиванием. Из приведённых результатов мы можем сделать вывод, что наиболее эффективными модификациями метода обратного распространения ошибки для задачи прогнозирования загрузки кластера являются:

1) стохастический градиентный спуск с инициализацией весов равномерным распределением и обновлением весов алгоритмом ADAM – как для T-DL MLP, так и для DL MLP;
2) стохастический градиентный спуск с инициализацией весов равномерным распределением и обновлением весов по методу Нестерова с моментами – для T-DL MLP;
3) стохастический градиентный спуск с инициализацией весов по методу Xavier и обновлением весов алгоритмом ADAM – для DL MLP.

Результат моделей T-DL MLP и DL MLP отличается незначительно, что, вероятно, обусловлено особенностью тестовых данных. Однако мы видим, что дополнительные признаки, характеризующие время и дату, улучшают прогноз нейросетевыми моделями на основе многослойного персептрона.

Таблица 3. Значения ошибок RMSE и MAE в зависимости от конфигурации обучения нейронной сети

Алгоритм инициализации весов	Алгоритм обновления весов	Алгоритм оптимизации	T-DL MAE	DL MAE	T-DL RMSE	DL RMSE
UNIFORM	NESTEROVS	LGD	7,33	7,99	8,22	9,24
XAVIER	NESTEROVS	LGD	7,31	8,20	9,10	9,38
UNIFORM	NESTEROVS	SGD	7,12	7,64	8,08	8,94
XAVIER	NESTEROVS	SGD	7,86	7,70	8,62	8,97
UNIFORM	ADAM	LGD	8,02	7,83	8,94	9,13
XAVIER	ADAM	LGD	8,14	7,91	8,55	9,18
UNIFORM	ADAM	SGD	7,21	7,54	8,11	8,84
XAVIER	ADAM	SGD	7,79	7,56	8,71	8,85

IV. Адаптивные модели селекции и композиции

Пример прогнозирования данных загрузки кластера нейронной сетью с учётом признаков времени и даты приведен на рис. 4, моделью EMMSP – на рис. 5. Пунктиром показаны прогнозные значения ряда. Графики прогнозных значений были получены вычислением прогноза через каждые 12 точек.

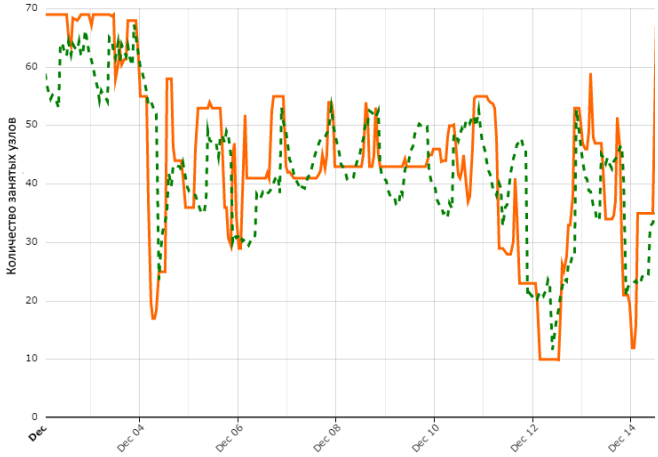

Рис. 4. Прогноз загрузки ресурсов при помощи модели T-DL MLP

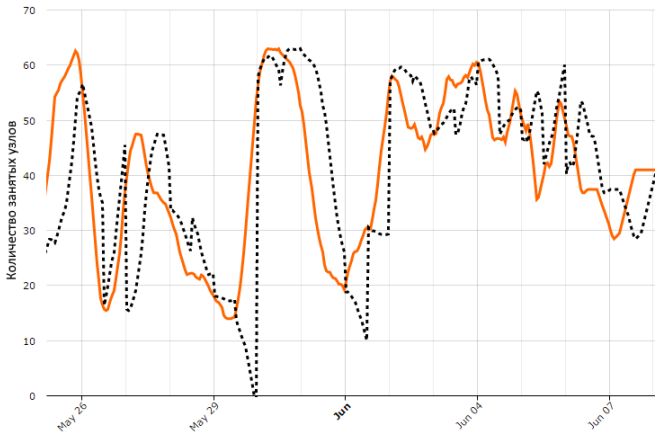

Рис. 5. Прогноз загрузки ресурсов при помощи модели EMMSP

Гипотеза 1. Для некоторого участка данных на основе прошлых прогнозов можно выбрать наиболее удачную модель.

Чтобы подтвердить эту гипотезу, мы выяснили, на какой средней длине участка данных эффективны модели EMMSP и T-DL MLP. Для модели EMMSP средняя длина отрезка данных, на котором она давала лучший результат, чем T-DL MLP, составила 27 точек, а для модели T-DL MLP - 36 точек. Поскольку на наших данных есть такие отрезки, на которых длительное время лучший результат показывает одна из моделей, то мы можем попробовать применить адаптивную селективную модель [14].

Пусть имеется n моделей прогнозирования:

$\hat{y}_{j,t+d}$ – прогноз j-ой модели на момент $t+d$; (1)

$\varepsilon_{jt} = y_t - \hat{y}_{jt}$ – ошибка прогноза в момент t; (2)

$\hat{\varepsilon}_{jt} := \gamma|\varepsilon_{jt}| + (1-\gamma)\hat{\varepsilon}_{jt}$ – экспоненциально (3)

сглаженная ошибка для оценки лучшей модели, где γ-экспериментально выбираемый коэффициент сглаживания.

Тогда, лучшая модель в момент времени t:

$j_t^* = argmin(\hat{\varepsilon}_{jt})$. (4)

И адаптивная модель прогнозирования может быть записана в виде:

$\hat{y}_{t+d} := \hat{y}_{j_t^*,t+d}$. (5)

Мы реализовали такую адаптивную модель, используя модели EMMSP и T-DL MLP как составляющие. Коэффициент γбыл выбран равным 0.75, его применение снижает частоту переключения активной модели, предотвращая неверный выбор модели в граничных условиях. Пример графика прогноза адаптивной селективной моделью показан на рис. 6.

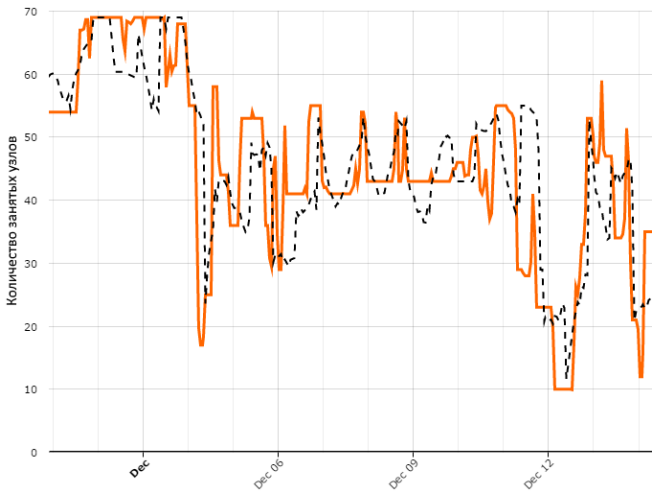

Рис. 6. Прогноз загрузки ресурсов при помощи модели адаптивной селекции

Гипотеза 2. Если сложить показания моделей с учётом некоторых весов, то можно получить лучший прогноз, чем даёт каждая из моделей.

Линейная комбинация моделей формируется как взвешенная сумма прогнозов:

$$\hat{y}_{t+d} = \sum_{j=1}^{n} w_{jt}\, \hat{y}_{j,t+d}, \quad (6)$$

$$\sum_{j=1}^{n} w_{jt} = 1. \quad (7)$$

Поскольку в нашей адаптивной композиции моделей участвуют всего две модели, то мы можем достаточно просто выполнить перебор весов w_{jt} от 0 до 1, не прибегая к процедуре адаптивного подбора веса [14]. В нашем случае единственный вес w_t является параметром модели и находится в процессе обучения на исторических данных. Пример графика прогноза адаптивной композицией моделей приведён на рис. 7.

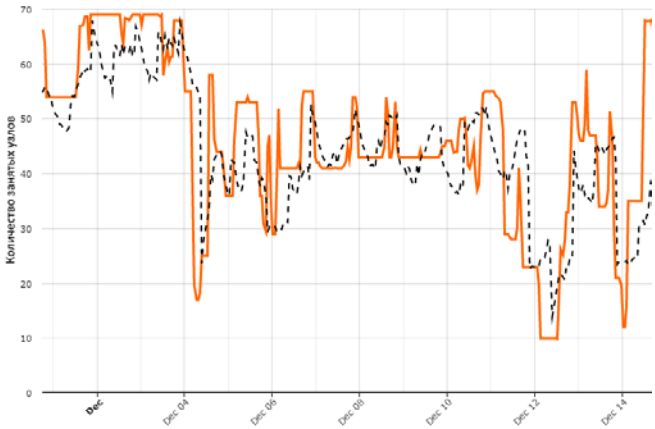

Рис. 7. Прогноз загрузки ресурсов при помощи модели адаптивной композиции

В качестве итоговой метрики ошибок выбрана средняя абсолютная ошибка (MAE), поскольку относительная ошибка прогноза (MAPE) не может быть использована в рядах, включающих значения близкие или равные нулю. По средним ошибкам MAE, приведённым в таблице 4, мы можем сделать вывод, что адаптивные модели не дают значительного улучшения прогноза, а модель адаптивной композиции не улучшает результат, работает хуже модели T-DL MLP. Однако, адаптивная селективная модель способна улучшить результат, поскольку выбирает модель из моделей с различной природой: EMMSP – детерминированный прогноз по прецедентам, T-DL MLP – нейросетевая модель, работающая по принципу чёрного ящика и подстраивающая свои веса для наилучшей аппроксимации всего набора данных.

Таблица 4. Значение ошибки MAE для различных моделей прогнозирования

Модель	EMMSP	T-DL MLP	Адаптивная селективная модель	Адаптивная композиция
MAE	8,7	7,12	6,8	7,5

Данные для тестирования были собраны в период с ноября 2013 года по декабрь 2016 года. Открытые данные мониторинга загрузки кластера «Сергей Королев» доступны в машиночитаемом формате JSON по адресу: http://templet.ssau.ru/wiki/открытые_данные.

Приведенные модели также реализованы в подсистеме аналитики [15] сервиса Templet Web [16] (http://templet.ssau.ru/app) и используются со следующими параметрами обучения:

- объём данных для обучения моделей – учитываются данные загрузки кластера за 6 месяцев;
- модели обучаются в режиме офлайн;
- повторное обучение моделей производится каждые 12 часов.

V. Заключение

На основании приведенных выше результатов, мы можем сделать вывод, что цели работы успешно достигнуты. Проанализированы две различные по своей природе модели прогнозирования, каждая из которых подходит для различных участков данных загрузки кластерной системы. На базе этих моделей реализованы и исследованы модели адаптивной селекции и композиции. Из этих двух моделей на основе адаптивных смесей нами выявлена одна модель – модель адаптивной селекции, которая даёт лучший результат, чем каждая из моделей в составе смеси.

Алгоритмы прогнозирования на основе нейросетевой модели с двумя скрытыми слоями и учётом информации о дате и времени (T-DL MLP), а также модель адаптивной селекции на базе моделей EMMSP и T-DL MLP интегрированы в сервис Templet Web, что позволяет пользователям оценить время, через которое будет запущена задача в пакетной системе PBS. Графики прогноза и истории загрузки кластера доступны зарегистрированным пользователям системы. В будущем планируется предоставить пользователям интерактивную подсказку о количестве доступных ресурсов и оценке времени запуска задачи на основе требований к кластеру (узлов, групп, лицензий на ПО), указанных в задаче на момент запуска.

Результаты прогнозирования загрузки кластера могут быть применены для решения нескольких типов задач:

− повышение эффективности использования кластера (энергоэффективность, повышение загрузки);

− выбор оптимальных окружений и параметров для

расчётов;

− планирование развития кластера и периодов его обслуживания.

− выделение простаивающих ресурсов для задач добровольных вычислений [17].

Помимо выбора окружения для проведения вычислительных экспериментов методы прогнозирования загрузки вычислительных ресурсов наиболее востребованы сейчас в облачных окружениях, где они могут позволить коммерческим компаниям снизить затраты на обслуживание серверов или же эффективно приспосабливаться к растущим требованиям клиентов.

В работе [18] описана технология, позволяющая задействовать простаивающие мощности кластеров и суперкомпьютеров для решения задач массивных вычислений. Приведённая нами модель прогнозирования может эффективно применяться для определения объёма свободных узлов кластера на следующие 12 часов, что необходимо для совместного использования ресурсов кластера с инструментом CluBORun. Прогнозирование загрузки кластера позволит исключить конкуренцию за ресурсы между пользователями кластера и механизмами загрузки простаивающих ресурсов, а так же верно выбрать длительность работы задач, запускаемых CluBORun.

Библиография

[1] S. Naseera, G.K. Rajini, P. Sunil Kumar Reddy “Host CPU Load Prediction Using Statistical Algorithms a comparative study” // International Journal of Computer Technology and Applications – 2016. – 9(12). – pp. 5577-5582.

[2] S. Di, D. Kondo, W. Cirne “Host load prediction in a Google compute cloud with a Bayesian model” // Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis. – IEEE Computer Society Press, 2012. – p. 21.

[3] Ю.С. Артамонов “Применение модели EMMSP для прогнозирования доступных вычислительных ресурсов в кластерных системах” // Известия Самарского научного центра РАН. – 2016. – том 18, № 4 (4). – С. 681-687.

[4] S. Naseera, G.K. Rajini, N. Amutha Prabha, G. Abhishek “A comparative study on CPU load predictions in a computational grid using artificial neural network algorithms” // Indian Journal of Science and Technology. – 2015. – Т. 8. – №. 35.

[5] K. Kalaitzakis, G. Stavrakakis, E.M. Anagnostakis “Short-term load forecasting based on artificial neural networks parallel implementation” // Electric Power Systems Research. – 2002. – Т. 63. – №. 3. – pp. 185-196.

[6] M. Chandini, R. Pushpalatha, R. Boraia “A Brief study on Prediction of load in Cloud Environment” // International Journal of Advanced Research in Computer and Communication Engineering. – 2016. – 5(5). – pp. 157-162.

[7] H.A. Engelbrecht, M. van Greunen “Forecasting methods for cloud hosted resources, a comparison” // Network and Service Management (CNSM), 2015 11th International Conference on. – IEEE, 2015. – pp. 29-35.

[8] С. Хайкин Нейронные сети. – М.: Вильямс, 2006. – 1104 с.

[9] Deeplearning4j: Open-source distributed deep learning for the JVM [Электронный ресурс]. URL: http://deeplearning4j.org (дата обращения: 01.01.2017)

[10] С. Осовский Нейронные сети для обработки информации. – М.: Финансы и статистика, 2002. – 344 с.

[11] Y. Nesterov Introductory Lectures on Convex Optimization A Basic Course – Springer, 2004. – 211 p.

[12] D.P. Kingma, J.L. Ba “ADAM: A Method for Stochastic Optimization” // arXiv: 1412:6980 [cs.LG], – 2014.

[13] S. Ruder “An overview of gradient descent optimization algorithms” //arXiv preprint arXiv: 1609.04747. – 2016.

[14] Ю.П. Лукашин Адаптивные методы краткосрочного прогнозирования временных рядов. – М.: Финансы и статистика, 2003. – 415 с.

[15] Ю.С. Артамонов, С.В. Востокин “Разработка распределенных приложений сбора и анализа данных на базе микросервисной архитектуры” // Известия Самарского научного центра Российской академии наук, т. 18, № 4(4), 2016. с.688-693.

[16] Ю.С. Артамонов, С.В. Востокин “Инструментальное программное обеспечение для разработки и поддержки исполнения приложений научных вычислений в кластерных системах” // Вестн. Сам. гос. техн. ун-та. Сер. Физ.-мат. науки, 19:4 (2015), С. 785–798.

[17] О.С. Заикин, М.А. Посыпкин, А.А. Семёнов, Н.П. Храпов Опыт организации добровольных вычислений на примере проектов OPTIMA@home и SAT@home // Вестник Нижегородского университета им. Н.И. Лобачевского, 2012, 5(2), С. 340-347.

[18] A. P. Afanasiev, I. V. Bychkov, M. O. Manzyuk, M. A. Posypkin, A. A. Semenov, O. S. Zaikin (2015). “Technology for integrating idle computing cluster resources into volunteer computing projects”. In Proc. of The 5th International Workshop on Computer Science and Engineering, Moscow, Russia (pp. 109-114).

Prediction of cluster system load using adaptive model mixture

Y.S. Artamonov

***Abstract* — In this paper, we examine the approach to choose a high-performance environment based on the prediction of the load of cluster nodes. The aim of the work is to investigate various prediction models in application to the task of forecasting the cluster load, choose the most successful model configurations and find out how to effectively apply these models all together.**

The article presents the results of comparison of EMMSP models, models based on neural networks and adaptive models for solving the task of forecasting the load of cluster resources. The following parameters of neural network models are considered: selection of activation functions, algorithms for initialization and updating of neuron weights, and coding of additional features for training the network on the basis of date and time data. The testing of adaptive selection models and adaptive composition was performed and improvement of the forecasting results was shown in comparison with the models on which they were based. Training and testing of the models was performed using the load dataset for the cluster "Sergey Korolev" for the period from November 2013 to December 2016.

***Keywords* — adaptive composition, adaptive selection, resources load, cluster, neural network, prediction.**