

# **ПОВЫШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ С ДИНАМИЧЕСКОЙ АРХИТЕКТУРОЙ С ПОМОЩЬЮ НЕЙРОННЫХ МЕТОДОВ ПРОГНОЗИРОВАНИЯ**

**Алексеев Н.А., Мазанка Р.М.**

*Институт телекоммуникационных систем КПИ им. Игоря Сикорского  
E-mail: alexeyev@its.kpi.ua, roman.mazanka@outlook.com*

## **Distributed Computing Systems with Dynamic Architecture Productivity Increasing Using Neural Networks Methods**

The one of progressive ways to solve the problem of constantly growing data processing is to build high-performance distributed environments which include the number of personal devices having tangible computational power in total. Nowadays a lot of so-called volunteer computations implementations are known but still there's a challenge of effective tasks distribution between such client devices because of its non-stable participation. The work is dedicated to increasing of distributed computing systems with dynamic architecture productivity using neural network methods.

Для решения задач, которые требуют больших вычислительных мощностей зачастую создаются распределенные вычислительные системы. Принцип их работы заключается в том, что они делят сложную задачу на несколько подзадач, которые делегируются участникам этой системы для выполнения, после чего участники отчитываются центру управления, и результат отдается конечному пользователю. Таким образом, основной целью создания распределенных вычислительных систем является повышение эффективности вычислений путем распределения задачи и выполнения ее несколькими машинами одновременно. В современном мире количество устройств, которые имеют осязаемую вычислительную мощность, превышает количество пользователей сети Интернет, поэтому идея объединения этих устройств для решения вычислительных задач приобретает все большую популярность. Внедрение ГПСК в корпоративной среде позволяет привлечь ресурсы, которые простаивают, для выполнения высокопроизводительных вычислений, тем самым, частично перекрывая необходимость использования мощных компьютеров.

Сегодня существует множество готовых решений для построения распределенной вычислительной системы. Чаще всего необходимость в такой системе возникает во время разработки научных проектов, следовательно, решения для построения вычислительных систем зачастую разрабатываются «под ключ» и не являются универсальными. Таким образом, построение корпоративной вычислительной системы на основе данных систем является сложной задачей. Для выявления наиболее универсального программного обеспечения определим основные требования, которые оно должно удовлетворять: поддержка нод, подключенных путем беспроводного соединения, наличие клиента для наиболее популярных платформ (\*NIX-based, Microsoft Windows, Mac OS), высокая степень универсальности, а также поддержка динамической архитектуры.

В ранее проведенных исследованиях было определено, что из имеющегося программного обеспечения наиболее полностью удовлетворяет требованиям разработка Висконсинского университета в Мадисоне (англ. University of Wisconsin–Madison, штат Висконсин, США) HTCondor.

HTCondor – это специализированная система управления нагрузкой рабочих мест. Она поддерживает построение очередей, приоритезацию задач, политики планирования, мониторинг

и управление ресурсами [1]. Если машина-исполнитель более недоступна – система мигрирует данную задачу на другую машину (во время активности система делает контрольные точки, поэтому промежуточный прогресс сохраняется). Данная система построена на платформе UNIX, и имеет агенты для других платформ, однако, под другими платформами имеются некоторые ограничения. Согласно данным W3Schools, Microsoft Windows является лидирующей операционной системой (приблизительно 78,5% в 2016 году) [2], поэтому имеет смысл рассматривать ограничения HTCCondor относительно данной операционной системы. Для Microsoft Windows они следующие:

- контрольная точка процесса и миграция не поддерживаются вследствие отсутствия окружения задания;
- grid-типе должен быть condor;
- доступ к сетевым папкам, для которых необходима аутентификация типа Kerberos, не осуществляется [3].

Очевидно, что наиболее критичное ограничение – это недоступность сохранения промежуточного результата и продолжения выполнения задачи на другой машине при утрате связи. Таким образом, если на рабочем узле выполняется задача, и с этим узлом на какой-то промежуток времени теряется связь, система вынуждена отдать эту задачу другому узлу, а текущий прогресс будет утерян. Поскольку в системе HTCCondor присутствует архитектурное ограничение в операционной системе Microsoft Windows, для увеличения эффективности вычислений было предложено изменение приоритета узлов, подключенных путем беспроводного соединения с учетом следующих показателей:

- учет качества сигнала беспроводной сети на узле;
- учет длительности подключения узла к текущей беспроводной сети.

Каждому узлу, который может выполнять задачи, выставляется соответствующий рейтинг, который влияет на место в очереди. Таким образом, при изменении рейтинга машины имеется возможность изменить алгоритм распределения задач в распределенной вычислительной системе.

Рассмотрим первый показатель, который предложено включить в зависимость при определении рейтинга машины. При низком уровне сигнала беспроводной сети существует высокая вероятность потери связи с другими узлами системы (в т.ч., с центром управления), соответственно, результат выполнения делегированной задачи может быть утерян с большой долей вероятности. Более приоритетными машинами для выполнения задач являются те, у которых наблюдается максимальный уровень сигнала беспроводной сети.

Для получения вероятности отключения узла как функции от контекстной информации пользователя было предложено создание нейронной сети, которая способна прогнозировать время доступности узла. Выбор нейронной сети как основы метода принятия решения обоснован способностью дообучаться, увеличивая точность своих прогнозов по мере поступления данных. Таким образом, нейронная сеть позволит оценить, через какое время будет предпологаемое отключение, и, чем больший показатель будет получен, тем больший коэффициент будет использован для выставления рейтинга текущему устройству. Согласно теории нейронных сетей, если функция определена на конечном множестве точек, то трехслойный перцептрон способен ее аппроксимировать (рис. 1). В качестве алгоритма обучения предлагается выбрать алгоритм обратного распространения ошибки. Данный алгоритм имеет возможность переобучения, что является преимуществом для системы оценки длительности сессии.

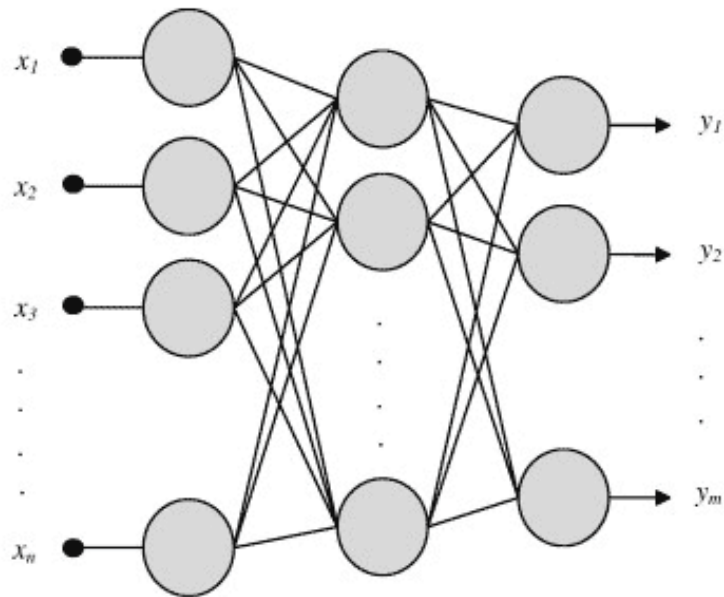


Рис. 1. Структура предлагаемой нейронной сети.

Количество весовых коэффициентов в такой сети можно вычислить по формуле (1).

$$l_w = \sum_{i=1}^{N_L-1} N_i N_{i+1}, \quad (1)$$

где  $N_L$  – количество слоев в нейронной сети,  $N_i$  – количество нейронов на  $i$ -ом слое.

Для обучения нейронной сети и получения начальных значений перед подключением ее к HTCCondor предлагается использовать беспроводную сеть Института телекоммуникационных систем. Данные начальные результаты будут справедливы для высших учебных заведений Украины, имеющих сходный режим работы.

Рассмотрев HTCCondor - как систему для использования в гетерогенной вычислительной сети с динамической архитектурой, можно сделать вывод, что для эффективного использования узлов, не сохраняющих промежуточные результаты вычислений, узлы, которые подключены путем беспроводного соединения следует рассматривать отдельно и учитывать (в их рейтинге) уровень сигнала беспроводной сети и оценку нейронной сети, которая базируется на длительности подключения узла и контекстной информации пользователя, а использование нейронной сети указанной конфигурации позволит более точно осуществлять распределение задач между вычислительными узлами.

## Литература

1. What is HTCCondor? [Электронный ресурс] – Режим доступа: <https://research.cs.wisc.edu/htcondor/description.html>. / – Электрон. текстовые данные (дата обращения: 08 марта 2016).
2. OS Platform Statistics – [Электронный ресурс] – Режим доступа: [https://www.w3schools.com/browsers/browsers\\_os.asp](https://www.w3schools.com/browsers/browsers_os.asp) – Электрон. текстовые данные (дата обращения: 26.03.2017).
3. Microsoft Windows – [Электронный ресурс] – Режим доступа: [http://research.cs.wisc.edu/htcondor/manual/v8.7/7\\_2Microsoft\\_Windows.html](http://research.cs.wisc.edu/htcondor/manual/v8.7/7_2Microsoft_Windows.html) – Электрон. текстовые данные (дата обращения: 26.03.2017).